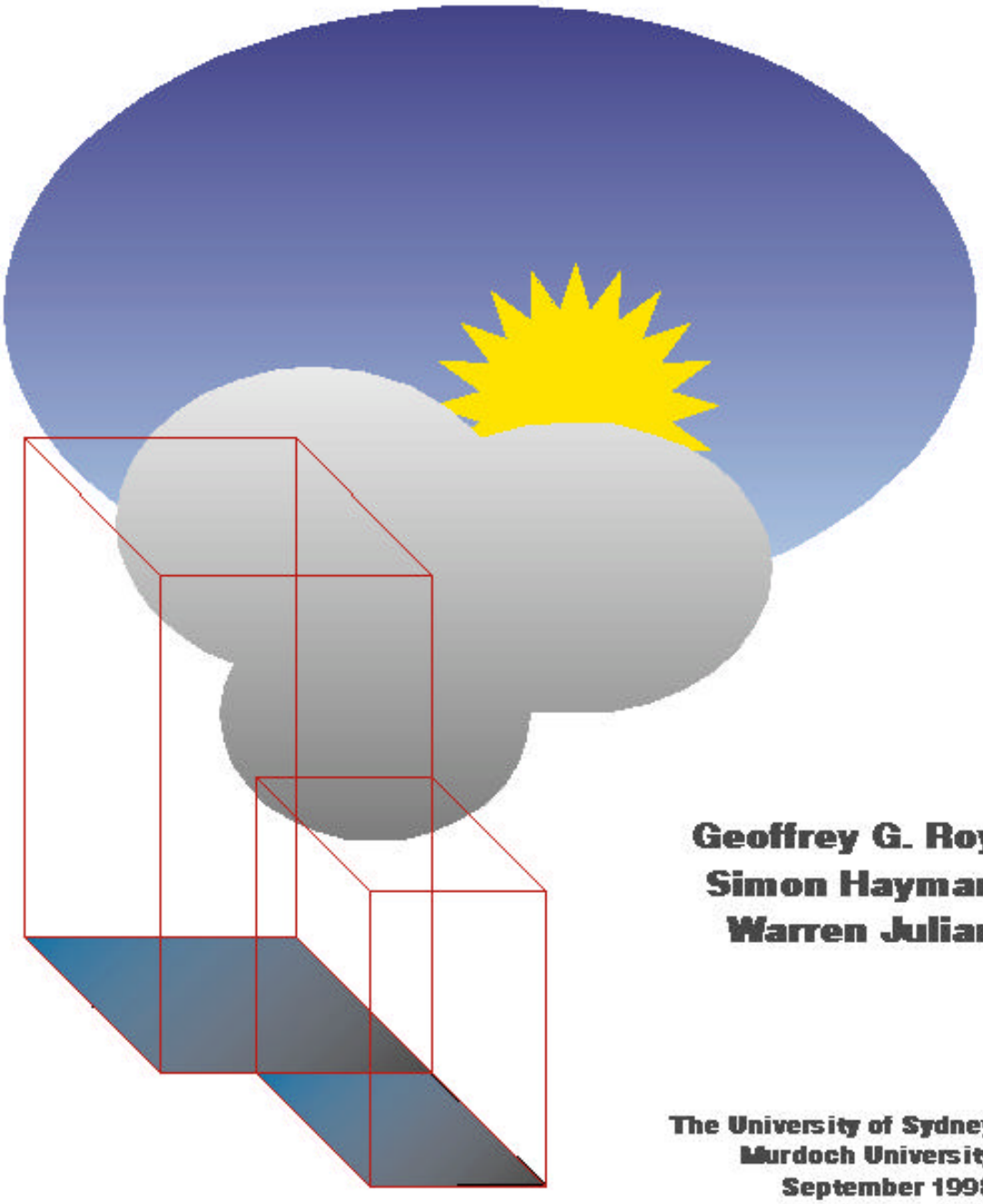
Sky Modelling from Digital Imagery



Geoffrey G. Roy
Simon Hayman
Warren Julian

The University of Sydney
Murdoch University
September 1998

ARC Project No. A89530177
Automated sky luminance and cloud cover estimation for improved sky modelling in different climate zones

Final Report

Sky Modelling from Digital Imagery

Geoffrey G. Roy
School of Engineering,
Murdoch University,
Murdoch, WA 6150

Simon Hayman
Department of Architectural and Design Science
The University of Sydney
Sydney, NSW 2006

Warren Julian
Department of Architectural and Design Science
The University of Sydney
Sydney, NSW 2006

The University of Sydney
Murdoch University

Contents

1 Introduction

1.1 Background

The sky, and the light it provides, is a fundamental part of our earthly existence. Our particular interest in this project is concerned with those properties of the sky which have a direct bearing on the availability of natural light for use in the built environment. Most buildings use natural daylight in some way, though some more than others. The balance between the use of natural and artificial lighting arises from many concerns, viz:

- Adequate lighting levels to allow the required activities to take place with safety and comfort in the building.
- Control of indoor climates (e.g. temperature, glare)
- Aesthetic considerations in the design of indoor spaces.
- The costs associated with modifying the indoor climate to meet required standards
- The energy consumption in achieving these standards.

A key element in meeting these concerns comes from the levels of available light from the sky, and how this varies throughout the day and year.

Skies change continuously, making their analysis and representation a particularly challenging problem. Current practice in building design most often relies on the use of a small number of "standard" skies from which "typical" daylight conditions can be studied. These standard skies are rarely seen in practice and often do not represent the characteristics of real skies in any specific location.

The study, and measurement of sky properties, has been the subject of on-going research for some time. For example, the time-based measurement of radiant flux from the sun, and sky, for the purpose of estimating heat loads on buildings is well established and such data forms the core knowledge for the thermal analysis of buildings. While it is also possible to record the integrated totals of incident luminous flux from the sky with some reliability, the highly directional nature of light makes this integrated information less than useful in studying the penetration of daylight into a particular façade aperture of a building.

Opto-mechanical scanners have been the prime source of information about the distribution of luminance across the sky dome. These (like the Krochmann PRC scanner) allow us to measure the luminance (average) over a small section of the sky (approx 10°). This information enables the luminance distribution to be mapped.

With the availability of high resolution CCD cameras we now have the capability to capture information about the sky at much finer resolutions. This opportunity opens up a new range of ways of analysing and documenting sky conditions.

1.2 Project Objectives

At its inception this project had the following objectives:

1) To evaluate, and demonstrate, the viability of using digital images of the sky to extract luminance information.

2) To attempt to extract cloud coverage information from the digital images.
3) To devise an image processing system that could analyse these images in real time (i.e. before the next image was recorded) and convert them to an efficient format for long term storage.
4) To save these models as a permanent record of the luminance characteristics of the sky so that they can be accessed by third partly simulation packages.
5) To prove the operations of this system through data collected at two locations in Australia representing quite different climatic conditions (i.e. Sydney and Darwin).

A number of events have not allowed us to achieve these objectives in full, viz:

- The funds provided for the project were less than requested and were insufficient to allow us to carry out the field trials and prove the whole system in full operation.
- We have experienced some problems with the operation of the digital camera which has caused some delays in building the physical connection between the camera and the controlling computer so that the computer and camera can operate unattended.

Otherwise, we have been able to achieve objectives 1 to 4, and the results of this work is reported herein.

1.3 Key Related Research

Research into properties of clouds, including cloud coverage, has been considerably boosted in recent years with the ready availability of satellite imagery. We now have a extensive range of satellite based data recordings of various environmental parameters of the earth. Smith et al [1994] present a typical set of results from such work, there are many other like this. Satellite images provide an excellent means of recording cloud properties over sections, or the whole, earth and most probably provides one of the best approaches for recording meteorological conditions. Such images are now regularly used in meteorological forecasting and often appear in television weather presentations.

Our particular interest, however, is from the point of view of the observer on the ground rather that trying to develop and overall description of clouds from an observer in space. We are concerned with the daylight availability in the built environment. This implies that our interest in focussed on a particular building in a particular location.

The most developed work for terrestrial based imagery has probably being done in the marine Physical Laboratory at the University of California (Shields et al, [1990], Shields [1998]). This group has developed (and sells) a complete package consisting of a camera, computer and related software which will record sky images and process this data to compute cloud coverage statistics. They also collect radiometric data from the digital images. Their Whole Sky Imager is designed to operate in both daylight and night-time conditions.

In some ways this project will complement this work, but with some important differences. We hope to demonstrate improvements in data recovery (both cloud

cover and radiometric data) as well as offering a data representation scheme which will retain the important characteristics of the digital images in a very compact and reusable form. In other ways we have not been able to demonstrate the same level of integrity (in terms of instrumentation calibration and reliability, and a field-ready package ready to deploy) due to limited resources. We are also targeting a much less sophisticated hardware environment, and thus potentially less expensive.

Davis et al [1992] have also reported research in the area of cloud edge extraction from coloured photographs. They have used colour photographs as their data source, with the usual problems of dealing with colour distortions inherent to film processing.

Nakamura and Oki [1975] and Robins et al [1984] have undertaken work towards extracting luminance values from photographic images, though there are always problems in calibration and controlling the optical properties of photographic images.

Much of the work presented in this report builds on earlier work done on the SDF (Standard Digital Form) modelling process. This scheme is intended to allow the effective representation and manipulation of complex luminance models of the sky. Descriptions of this work can be found in Roy, Ruck, Reid, Winkelmann and Julian [1995], Roy, Reid and Winkelmann [1995], Roy, Ruck, Reid and Julian [1995] and Roy [1998]. The SDF concept provides tools for the creation and manipulation of digital models of the sky luminance, their efficient storage and retrieval, and access to them through third-party software packages.

1.4 Summary of Achievements

The availability of high quality digital images of the sky has opened up a new range of opportunities for sky modelling and analysis. The core objectives for this project were to:

- Establish the viability of using CCD images to measure luminous characteristics of the sky dome.
- Quantify cloud coverage.
- Capture the sky conditions in a compact form suitable for long term storage and retrieval with minimal information loss.
- To demonstrate the operations of a prototype system in two different climate zones.

The last of these objectives has not been realized due to a shortfall in funding from the original proposal.

The specific achievements, however, have been:

1. The demonstration that a standard CCD camera can be calibrated to give quite respectable results for the determination of sky luminance levels.
2. The development of a number of modelling techniques to segment a digital image of the sky dome into cloud and sky components. These are based on empirical methods using colour information from the image in an intuitive way, and on the use of analogue neural networks to classify the image using a set of sample points which have been pre-classified. These show that it is possible to achieve a segmentation with some degree of reliability for images taken after sunrise and before sunset. We have been able to account to some extent for the presence of of

direct solar light, but in some cases the sun and the circumsolar region is miss-classified.

3. We have developed a method for cleaning up the resulting image using a convolution mask to remove odd miss-classified pixels. This step makes the subsequent processing more reliable.
4. The cloud edges have been able to be represented by polylines, suitably smoothed to give a level of accuracy required by the user (i.e. it is user selectable).
5. From this model we can construct an SDF model of the luminance distribution as well as the cloud coverage. This model reduces the data requirements of the original image (3.2 Mbytes) to somewhere in the range 20 to 50 kbytes for long term storage and retrieval.
6. We can demonstrate the complete processing of an image in less than 2 minutes using a medium speed micro computer.

We believe that the processes described in this report demonstrate the viability of using digital images from CCD cameras for the recording and analysis of skies for both luminance and cloud coverage.

2 CCD Calibration

2.1 Digital RGB imaging systems

A digital image is an array of pixels in which each pixel is defined by numerical components that define its grey level or colour. Three components are generally sufficient for image encoding although a fourth colour - black - is also used either as an applied colour, as in printing, or by default in imaging technology as it represents the 'off' state of the pixel. Due to the need to develop efficient digital imaging systems, with pixels able to detect any of thousands of colours, most imaging technology relies on a simple definition systems such as directly using the red, green and blue (RGB) output of the imaging device.

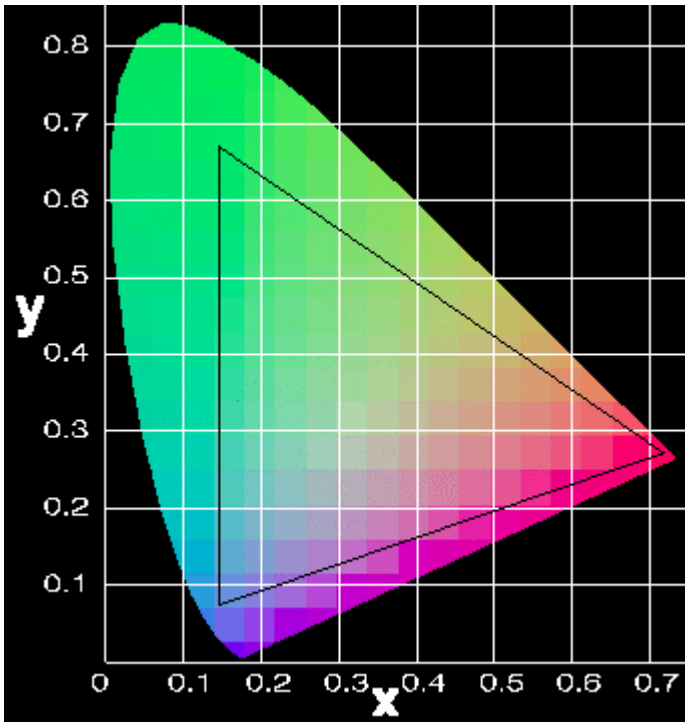


Figure 2.1 - CIE Chromaticity Diagram with typical RGB mixing boundary

The full range of possible colours and luminances can be specified using a system such as the CIE Chromaticity method [CIE, 1986] where the definition of a colour is based upon a mix of three ideal primaries that are theoretical rather than real. Any definition system that uses real primaries, such as the RGB system in the Nikon E2 camera, cannot cover the full range of colours as shown in Figure 2.1. In this particular application of imaging technology to cloud edge detection, however, the colour discrimination required is largely between tones of blue and white which are located close to the centre of the CIE diagram and, hence well within the bounds of a typical RGB system.

The other limitation of much digital imaging is its restricted capability in measuring the full range of luminances experienced under a typical sky which can be over four orders of magnitude. Furthermore, luminance is not directly measured in such systems but is derived from a weighted sum of components. For example, in a RGB system

the green component (G) carries most of this information as it peaks close to the peak of the V(λ) function whereas the red (R) and blue (B) components peak at the low ends of this luminous efficiency function. The weightings to produce a luminance related function, V, from displays conforming to the international standard for linear RGB [ITU, 1990] are:

$$V = 0.2125R + 0.7154G + 0.0721B$$

The weakness of this system is that although it simulates perceived brightness level relatively well in the image, it achieves this by scaling the luminances relative to the brightest object in the field which is likely to be white (this corresponds to the CIE Lightness function [CIE, 1986]). It will not give an absolute luminances because 'white' for RGB system will always result in an V value of 1. Additional information is required to estimate absolute luminances. Consequently, an imaging system based upon traditional photographic technology has advantages.

Cameras use a combination of aperture size and shutter speed to control the amount of light falling on the film to suit the sensitivity of the film used. These variables can be combined to give an 'exposure value' that will be proportional to, at least, the average luminance distribution of the scene photographed. A digital imaging system that has this capability may, therefore, be able to be calibrated to provide absolute measurements of luminance.

2.2 Digital camera systems

A number of attempts have been made to use photographs as an objective technique for cloud measurement. In the context of sky luminance measurement, the most well known is the work of Nakamura and Oki [1975] but others have also used this method, eg. Robbins et al [1984]. Nakamura and Oki's work was based on optical density measurements, using a microdensitometer, of the film grain in black and white images. The dynamic range of the film was found to range from approximately 0.1 to 10,000 cdm^{-2} with a claimed accuracy of ±5%. Luminance measurements of the fine detail of cloud structures were able to be measured but this came at a price as it required approximately a week to create the iso-diagram for each image. Such slow processing means that there are possible advantages in digital imaging.

A number of systems have been developed for cloud analysis based upon relatively cheap video cameras but these are often restricted by low resolution imaging, limited dynamic range in the image and lack of a suitable lens that can capture the whole sky (the widest field of view for C mount lenses used by many video cameras is 135 degrees). However, given very extensive resources Shields et al [1990] have developed an automatic cloud determination system with 1/3 degree resolution for sky radiances.

Most photographic analyses have used the Nikon Nikkor 8mm f/2.8 full-field lens as it provides a full 180 degree field of view. The lens optics produce an image in equiangular projection so that it is easy to measure altitudes and azimuths as equal distances on the image correspond to equal angles. Unfortunately, this K mount lens has not been available, until recently, on digital imaging systems. Kodak have produced, in association with Nikon, a digital camera 'back' (DS2000) which is able

to take the full range of Nikon lenses. Moreover, the digital imaging array or CCD (charge coupled device) only captures about 40 percent of the normal 35mm frame. Consequently, it is unsuitable for cloud measurements as the Nikon 8mm lens uses the full height of the 35mm format. Recently, Nikon in association with Fuji film, have produced the E2 series of digital cameras (Figure 2.2) that overcomes this limitation. An additional set of optics is provided in the camera where the film plane would be in a conventional camera which focuses the image down onto a 1200 by 1000 pixel CCD.

Key	Description
1	Aperture Dial
2	Remote Terminal
3	White Balance Window
4	Lens Mounting Index
5	Lens Release Button
6	Camera Strap Eyelet
7	Focus Mode Selector
8	Tripod Socket
9	Command Dial
10	LCD panel
11	Eyepiece Ring
12	Main Dial (L, S, C, SD, PB)
13	Shutter Release Button
14	Main Dial Lock-Release Button
15	Accessory Shoe
16	SENS (Sensitivity Selector) Button
17	Exposure Compensation Button
18	Metering System Button
19	MODE (Exposure Mode) Button

Figure 2.2 - Nikon E2 digital camera

2.3 Problems of Nikon E2 camera

2.3.1 Data Transfer

The Nikon E2 camera stores its images on a image memory card (PCIMCA) that can be removed and read by a separate card reader attached to a computer. Card capacity

is dependent on the image resolution used but would be severely limited in the field if images were taken at 15 minute intervals as it would probably need to be changed daily. The provision of a digital terminal which provides digital output and input control should have solved this problem. It, however, is officially an unsupported feature of the camera but the camera is equipped with a Send Data setting for the shutter (the option SD on the Main Dial located by reference number 12 in Figure 2.2). Furthermore, it was known that the port is used by Nikon engineers for repair diagnostics. Unfortunately, the port does not appear to use a standard data protocol and there was great difficulty in getting this information as Nikon Japan were not forthcoming and conflicting information was received from Nikon distributors around the world. As a result, an extended attempt was made to reverse engineer the communication protocol, and the camera to computer interface cable, but this was only partially successful. The communication sequence was able to be initiated but no data was transmitted by the camera. It, therefore, appears that a non-standard proprietary protocol is in place.

It is possible to view and capture images via the video output terminals provided on the camera but these do not carry any archive information on the date and time that the image was taken and the camera settings used. Although this is adequate for part of the system of cloud edge detection described in the following chapters it prevents luminance calibration of the images.

2.3.2 File Structure

The image files that the Nikon E2 produces are in JPEG format that can be read by any suitable graphics application. The images are automatically labelled with the time and date of exposure as the file names, eg. AP960415092638, in 2 digit year/month/date/hour/minute/second format. However, only the Nikon Browser package supplied with the memory card reader appeared to be able to provide the shutter speed, aperture setting and image 'quality', ie. compression level setting. Unlike most JPEG files this information was not stored in the 'Comments' field but transpired to be encoded with the image data. As a result, it was necessary to investigate the data structure of the JPEG files to identify those bytes that carried this data and provide a translation table.

To achieve this, a number of fixed images were taken varying one of the camera variables (shutter, aperture and quality) at a time and the header of the image, as an octal dump, was scanned for variations between images. It was found that this information, as well as the date and time stamp, was stored between byte addresses 0000645 and 0000710 with the following mappings:

address	octal	mode	resolution
645	1	basic	1 byte/pixel
	2	normal	2 bytes/pixel
	3	fine	4 bytes/pixel

Table 2.1 - Image quality mapping

address	octal	decimal	shutter speed (seconds)
655	36	30	1/8
	43	35	1/10
	50	40	1/15
	55	45	1/20
	62	50	1/30
	67	55	1/45
	74	60	1/60
	101	65	1/90
	106	70	1/125
	113	75	1/180
	120	80	1/250
	125	85	1/350
	132	90	1/500
	137	95	1/750
	144	100	1/1000
	151	105	1/1500
	156	110	1/2000

Table 2.2 - Shutter speed mapping

address	octal	decimal	f-stop
665	67	55	6.7
	74	60	8
	101	65	9.5
	106	70	11
	113	75	13
	120	80	16
	125	85	19
	132	90	22
	137	95	27
	144	100	32

Table 2.3 - Aperture mapping

It should be noted that the tables show actual values experimentally determined. Depending on the lens used, the camera is capable of longer shutter speeds and wider apertures than noted above. It is be expected that the above regular mapping can be extrapolated.

Once the mappings were established an application was written to extract this information from the data structure and place it into the JPEG Comments field in translated form so that any JPEG capable software could view it.

2.4 Luminance calibration of Nikon E2 camera

To generate a luminance function from the image RGB values in this experimental work, the values were compared with measured luminances to establish a calibration. A two stage analysis is, however, needed in practice due to the limited dynamic range of luminance reconstruction. Firstly, a comparison must be made between measured values of sky luminance and computed luminances and secondly, a comparison with exposure values (the combination of shutter speed and aperture).

The technique used was to capture sky images with the E2 camera mounted on its back pointing to the sky zenith. At same time as the image was captured, 150 measurements were made of zenith luminance using a Krochmann Sky Luminance Scanner. These images were, at first, imported into Photoshop[1] where an overlay corresponding to the IDMP sky luminance grid [Tregenza, 1987] was used to locate the 10 degree zenith measurement zone. The RGB values of 5 points within this region were taken and averaged if there was less than 5% variation between them otherwise they were discarded. If the simultaneous zenith luminance measurements also had less than 5% variation an average was taken otherwise these measurements too were discarded.

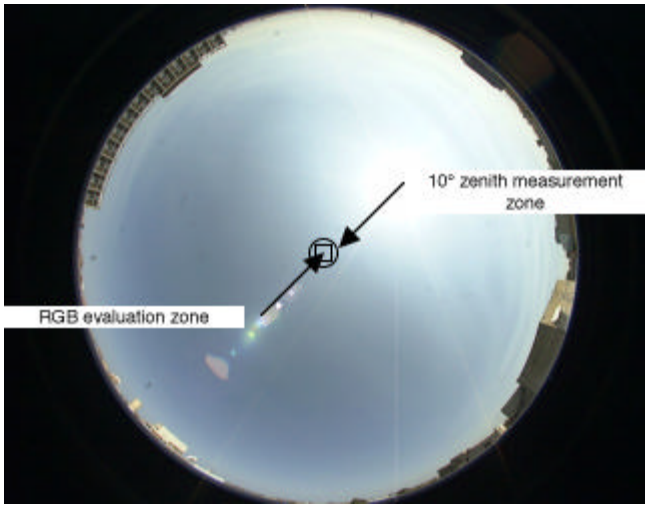


Figure 2.3 - Zenith RGB and luminance measurement zones

To increase the accuracy of this process the analysis was later carried out on a larger number of pixels, within the measured zenith region, by extracting the relevant pixels directly from the image data structure. A square array of 55 by 55 pixels, inscribed within the 10° measurement field, was used yielding a set of 3025 RGB values for each image. The results of a typical single image are shown in Figure 2.4 as a cumulative frequency distribution of RGB values.

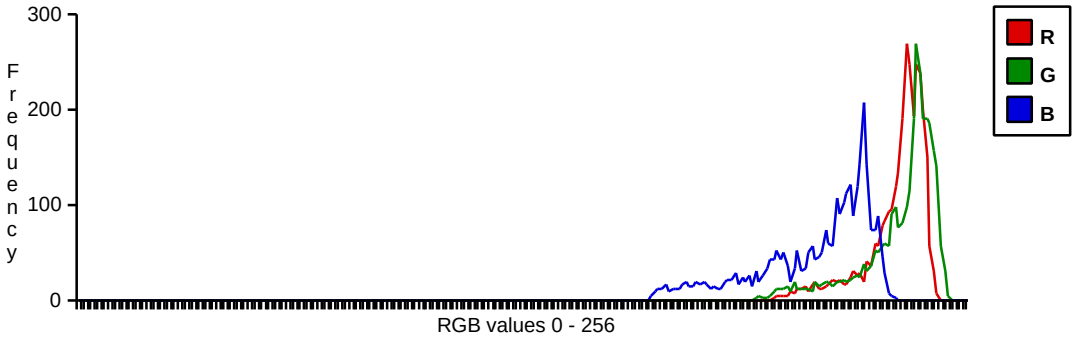


Figure 2.4 - Typical frequency distribution of RGB values from zenith image

[1] Adobe Photoshop, a package that supports image manipulation.

This data was then compared with the simultaneous measured luminance data by transforming the RGB data into a luminance function, V, using the relationship given in Section 2.1 above. A standard function [Kauffman, 1987] combining film (S, ISO number) and shutter speed (T, in seconds) and aperture (f, f-stop number) was used to calculate exposure value, Ev:

$$Ev = \frac{179}{200}\pi \times S \times \frac{T}{f^2}$$

A user defined curve fitting program, MacCurveFit [Rainer, 1998], was used to evaluate the calibrated luminance function L using the relative image luminance function, V, and the exposure settings for the image, Ev:

$$L = \frac{V^{2.4}}{2901.978\ Ev}$$

with L being measured in Cd/sqm.

The performance of this computed luminance function was then compared to the measured luminance data as shown in Figure 2.5. The correlation (r^2) of 0.965 is very high and exhibits good residual balance.

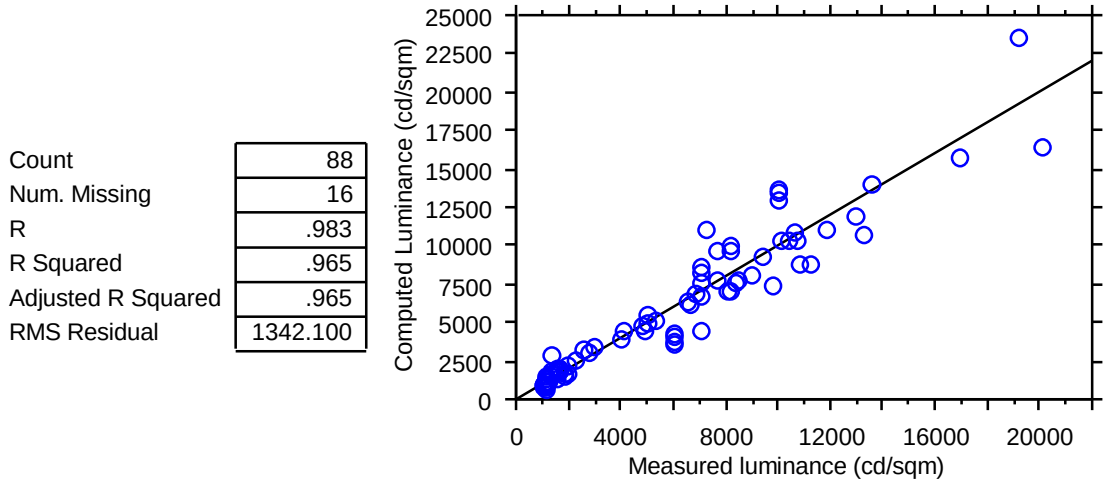


Figure 2.5. Comparison between measured luminance and the calibrated luminance function

2.5 Discussion

A little care should be taken with the computed luminances produced by the luminance function above as they are not measured using the correct spectral efficiency distribution which should be taken into account if absolute luminance measurement is required. It could be possible to introduce a compensatory filter to match the spectral response of the CCD to the V(λ) function into the filter turret of the Nikon full field lens but this could degrade the image colour for cloud detection.

The dynamic range the computed luminance function appears to be valid up to 30,000 cdm^{-2} which is a definite improvement over the photographic techniques, as used by

Nakamura and Oki,. However, it still falls short of the maximum sky luminances with a clear sky in the circumsolar area of the order of 60,000 cdm^{-2} (excluding direct sunlight). To extend the range it may be possible to develop a system of comparison between, at least, a pair of images taken at different manual exposures rather than relying upon the camera's automatic. or semi-automatic exposure function. For practical field use this will be dependent upon the problem of digital communication with the camera being solved.

Luminance is, however, only one part of the problem of cloud detection and measurement. The other issues is the ability of the system to distinguish cloud from sky. Here the RGB information must be used and 'colour' discrimination functions developed such as the red to blue ratio and is the subject of the following sections.

3 Sky Segmentation

3.1 Introduction

One of the basic tasks to be faced concerns the determination of the location and amount of cloud coverage in each sky image. Clouds are just not clouds, they have a wide range of properties and classification types. For this project, however, isolating the clouds by type and detailed property is not the ambition. Our ambition is much less – just to segment the sky into clear and cloudy parts.

The prime interest is to model the luminance characteristics of the sky (which is not simply related to the presence of clouds) and to characterise the sky in terms of the cloud coverage as typically estimated in meteorological measurements (ie. the fraction of the sky dome covered with cloud). If we know the location of clouds in the sky then it is possible to compute the proportion of coverage at a given time, and to perform various statistical analyses over time. For example, given a lengthy sequence of sky images we could produce cloud distributions, both spatially and temporally. Such information will provide designers with valuable information on the effectiveness, or impact, of daylight availability from different directions. A location which experiences dominate morning cloud, for example, should indicate to the building designer that daylight availability from eastern-oriented windows may be less than otherwise might be otherwise expected using standard sky models.

The essential task is to be able to process a digital image of the sky dome, and to describe the edges of the clouds that appear in that image. We then assume that all parts of the sky dome that fall within these edges are classified as cloud, the remainder being clear sky. Apart from accounting for the generally present horizon effects (land forms, vegetation and other buildings which appear in the sky image) the only major problem concerns the sun itself. We will be proposing some techniques to minimise the impact of the sun in our analysis, though as expected we will not be able to eliminate its impact completely[2].

In the work completed so far we also make some other assumptions, and limit the applicability of our analysis techniques, in particular:

- We assume that the images to be processed are captured after sunrise and before sunset, by a sufficient time interval to eliminate the need to consider sunrise/sunset colour change effects. Naturally image capturing devices which record in the visible spectrum[3] are only effective in periods of daylight.
- We assume that for the purposes of cloud coverage estimation the amount of sky occluded by the horizon effects is small. While we can estimate the proportion of horizon obstructions, it is not possible to know, from the digital image, whether there is clear sky or cloudy sky behind the obstruction.
- We have not considered the impact of atmospheric conditions where the sky is not visible (e.g. fog).
- We have not considered the impact of heavily polluted skies where the presence of particulate matter and other chemical components (eg NOx) which can change

[2] We have chosen not to use an occulator to mask the sun as this would add a level of complexity to our system that we feel is not required.
[3] Our camera is intended for daylight photography and will not have the sensitivity to capture images during non-daylight hours.

the colour characteristics of the sky. While it is probable that our analytical techniques could be extended to handle such situations, we have not been able to pursue this area of research in this project.

In essence our interest has been limited to daylight hours in locations with low levels of atmospheric pollution.

The task of segmenting the sky into clear sky and cloud regions can be approached in a number of ways, depending on the availability of measuring devices (eg. radar, infra-red imagery, visible light imagery, etc). We are limiting our interest to what can be recorded in daylight hours using a "conventional" digital camera.

Davis et al [1992] have shown that with a quite simple approach of working from coloured photographs a reasonable attempt can be made to isolate out the cloud patches in an image of the sky. They used images scanned from photographs, which in itself generates a range of uncertainties including the response of the film, the distortions introduced in film processing as well as the response of the scanning device. Even so they have been able to demonstrate some success.

Shields et al [1990] have developed a much more sophisticated approach using a CCD camera, with ND filters to extend the dynamic range of the recorded images, and red and blue filters to selectively record the image in these wavelengths. They use a Photometrics Slow Scan CCD which is cooled to –35°C for optimal performance, and will allow images to be recorded in very low light levels, and probably during the night time.

3.2 Basic Concepts

While the presence of clouds can be detected by various means (including radar, and other non-visible radiation detectors) we are limiting our interest to the visible light spectrum that can be recorded on a CDD imaging device. For most of the day the sky is characterised by blue coloured clear sky and white/grey coloured clouds. It is our hypothesis that the basic segmentation problem can be approached from an analysis of the colour information contained in the digital image of the sky.

At first sight this raises a number of interesting questions which could impact on the results of the following analyses. We have explored two approaches which are detailed in the following sections, though both use the same colour information from the digital image.

The digital image we have is 1000 by 1280 pixels of 24 bits depth with the effective image being a circle about 1000 pixels in diameter. At each pixel we have 256 levels of information for each of the red, green and blue (RGB) light levels. We accept the fact that the RGB values are intended to have a similar response to that of the human eye (since the camera we use is designed to take normal photographs) – and so look "realistic" to the viewer. A typical image is shown in Figure 1.1.

Figure 1.1: A Typical Digital Image

The image we have to work with is thus circular resulting from the fish-eye (180°) lens which produces an equi-angular transformation of the sky dome onto the image plane of the CCD. While we can work with this full image, it would appear that such detail is not necessary to demonstrate the operations of our analytical tools (and probably not necessary in most practical applications). We will thus (by default) reduce the working images to 500x500 or 250x250 pixels. We know that there are limits to this resolution reduction – at some point the resolution is too coarse to estimate luminance levels as well as to effectively segment the sky from the cloud patches. It might be expected that images down to about 100x100 pixels will still give useful results, though we have not explored this thoroughly.

3.3 Colour Spaces

The task of sky segmentation is essentially quite simple at face value – we argue that anything that is "blue" must be sky, the rest "cloud". To gain some understanding of the colour characteristics of the sky we have taken pixel samples from a set of images covering clear, partly cloudy and fully overcast skies – classifying these as cloud, sky, sun and other (horizon obstruction). We can examine these data points in RBG (red, green, blue) colour space as shown in Figure 1.2:

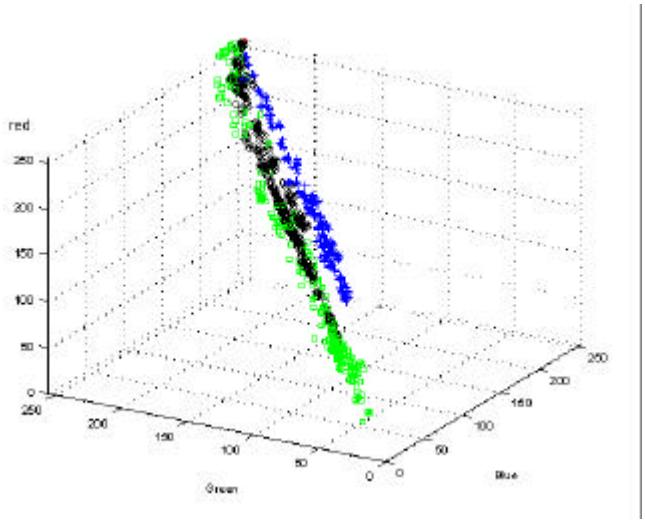


Figure 1.2: Scatter plot of pixel samples in RGB colour space (sky – blue, black – cloud, sun – red, other – green)

The sky, cloud and other classified points all lie in a region close to the principle diagonal. Note that all the "sun" points are all coincident at [255,255,255] and are not visible in Figure 1.2.

Taking the same data and plotting it in HSV (hue-saturation-value) colour space we get Figure 1.3:

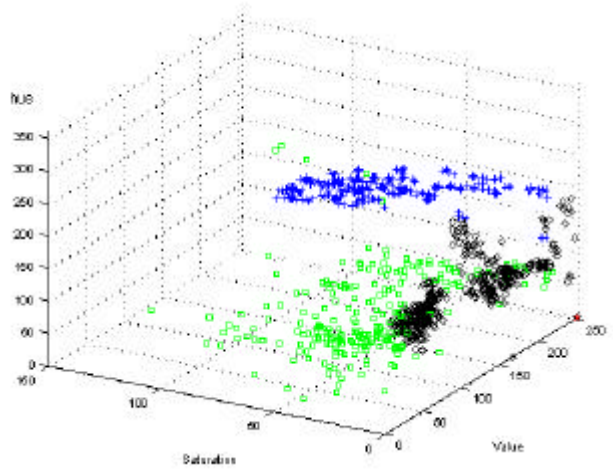


Figure 1.3: Scatter plot of pixel samples in RGB colour space (sky – blue, black – cloud, sun – red, other – green)

In this case we see a more clearly defined partitioning of the data sets, especially the sky from the rest.

If we look at the HSV colour model as shown in Figure 1.4 we see that the region which likely to best represent clouds will be along the Value axis from some where towards the botton (tip) of the HSV cone up towards the top (base of the cone) where we have the dark-greys, greys, light-greys and whites. In the HSV model red is usually located at a Hue angle of 0°, green at 120° and blue at 240°, with the secondary colours placed mid-way between the primaries. Those parts of the image that are likely to be clear sky will thus tend to have a hue value around 240° (± some deviation) and will have a non-zero saturation. Cloud, we assume, will have less, or

little saturation, and will tend to be identified by points along the Value axis, from the dark-greys to the whites.

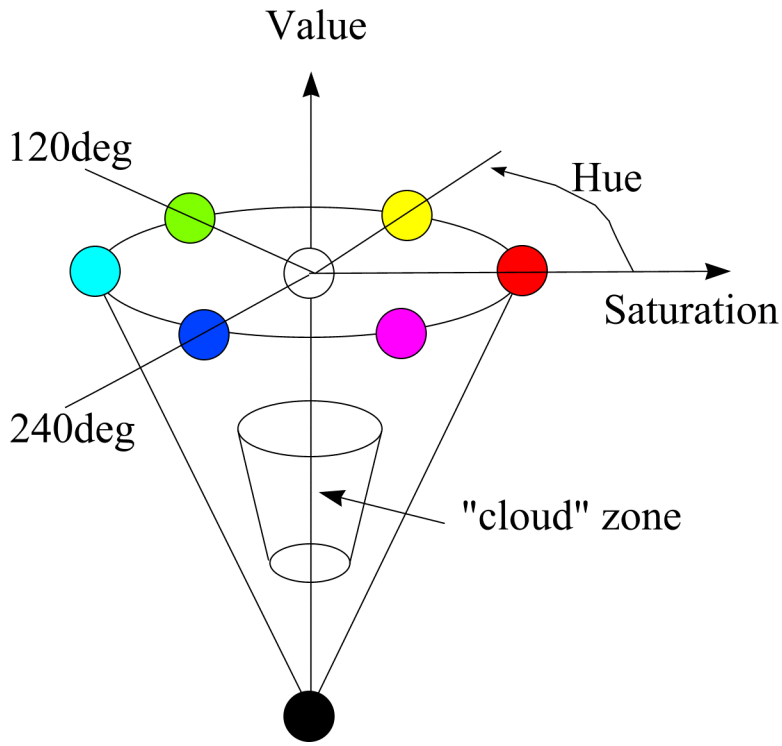


Figure 1.4: The HSV Colour Model

The HSV model thus seems to offer some possibilities for describing, and then partitioning, the colour space into the cloud and sky components. In RGB space this is not so obvious. Though, since there is a mapping from one to the other (see below), it could be done – but it is not so easy to think in RGB terms.

It is usual to scale the Value and Saturation axes from 0 to 100% (or 0 to 255 for our digital images) and the Hue from 0 to 360°. From the digital image we actually have RGB values. The conversion from RGB to HSV is done as follows (as a C language function):

```
void rgbTOhsv(integer red,integer green,integer blue,
              integer *hue,integer *sat, integer *val)
{
    integer m;
    *val = max3(red,green,blue);
    m = min3(red,green,blue);
    if (*val != 0)  *sat = (*val - m)*255 / *val;
    else            *sat = 0;
    if (*sat != 0)
    {
        if      (red   == *val) *hue = (green - blue)*60/(*val - m);
        else if (green == *val) *hue = 120 + (blue - red)*60/(*val - m);
        else if (blue  == *val) *hue = 240 + (red - green)*60/(*val - m);
        if (*hue < 0) *hue = *hue + 360;
    }
    else *hue = 0;
}
```

where *max3* and *min3* return the maximum and minimum, respectively, of three integer values. The Hue value is determined by the dominant RGB component and

the saturation from the deviation of this component from the minimum of the RGB values. The conversion algorithm is based on that proposed by Foley and Van Dam [1983].

3.4 An Empirical Approach

To develop a simple empirical approach we need to undertake some steps to classify the colour space. This process is largely a matter of experimentation, and some trial-and-error. Clearly there are some obvious issues:

1) *The response of CCD in camera, both for intensity and colour.* Since we only have one camera to work with we will make the assumption that our empirical model will be for that camera only – we would expect that other CCD devices will require a detailed analysis and may have different parameters.

2) *Any losses, distortions which may occur in getting the image from camera to computer.* The images from the camera are compressed using a JPEG algorithm after exposure. This may produce some errors, but it thought they will not be significant for our analysis.

3) *The exposure setting on the camera (shutter speed and aperture).* For the sequence of images used for the calibration and testing of these models for segmentation the exposure setting varies (by the camera's automatic exposure algorithm). If we assume that this adjustment process has been designed (perhaps "optimised") to use the effective dynamic response range of the CCD then it is possible to argue that the changing exposure settings are tending to cancel out the potential variations in light quanta actually reaching the CCD so that the colour rendition is retained. If, for example, the camera had no exposure adjustments then high levels of light falling on the CCD will quickly exceed its response range and the recorded saturation values will fall (as the CCD saturates).

4) *The dynamic range of the CCD is limited.* We expect, and can observe, that the digital images from the camera have a limited dynamic range. From the camera calibration described earlier in this report we have observed that the CCD is limited to a dynamic range of about 30,000 Cd/sqm. It is possible to extend the dynamic range with the use of multiple images with different exposure settings, but we have not taken this step here. For the purposes of sky segmentation we are not convinced that such a step is absolutely necessary, though it may improve the results of the algorithm in some circumstances. For example, if we have lots of close-to-white clouds, or a very large and bright circumsolar region.

Davis et al [1992] have proposed that the segmentation can be effectively achieved from a partitioning of the HSV colour space on the saturation dimension alone. They suggest that two thresholds on the saturation value (or 54% and 47%) can be used. That is, a value above 54% would result in a sky classification, and below 47% cloud. Presumably in between values would be "uncertain". This approach is not clearly demonstrated in the data from our camera as shown in Figure 1.3. In this Figure it would appear that the Hue dimension is a better discriminator.

Shields [1990] have proposed that the segmentation is best achieved from the red/blue ratio. A value above some threshold results in cloud classifications while a value below is considered sky. From our own data, we can see how this ratio looks in Figure 1.5 (this is a replot of the sky and cloud data points, only, from Figure 1.2).

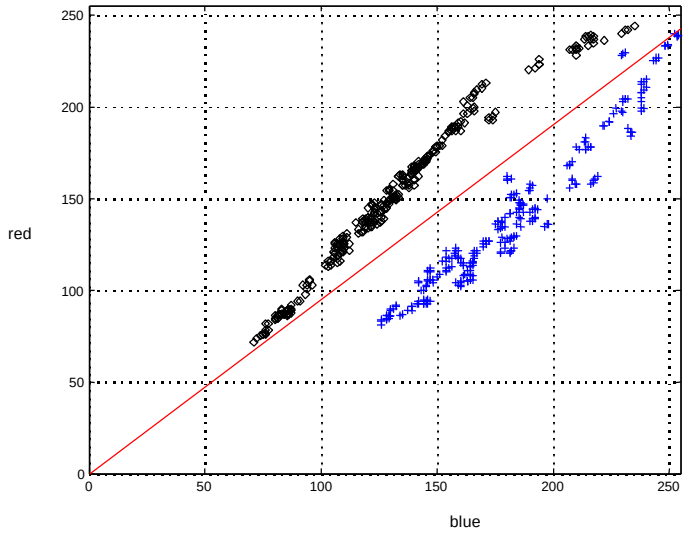


Figure 1.5: Plot of red vs blue dimensions of colour for sky (blue) and cloud (black) pixels

From this diagram we might conclude that a red/blue ratio of about 0.95 (the red line) would segment this data set rather well. There are many data points close to the divide line, so we will need to see how this works on a variety of full images.

Given these indications, and our own experimentation, we have implemented two empirical segmentation filters, viz:

- An HSV filter – which does a primary partitioning on the basis of hue value
- An RGB ratio filter – which does a primary partitioning on the basis of red/blue ratio.

In keeping with the KISS[4] principle we have been able to use a very simple partitionings of the colour space and get quite respectable results. Our initial investigations suggest the following for any pixel in the image:

HSV Filter

The HSV filter is defined as follows:

```
if (hue > 120 &&  hue < 300 && sat > 10) return sky;
else if (val > 200 && sat < 10 && inside_corona) return sun;
else return cloud;
```

The first condition isolates that part of the HSV space which has some "blue" colour – we will call that "sky". The second condition selects values which have high Values and low Saturations (ie the "whites"). If such points are also located where we expect the sun to be located (using the algorithm proposed by Roy et al [1989]), then we will classify the point as being "sun". The third condition defines all other points to be of type "cloud".

[4] Keep It Simple, Stupid!

RGB ratio Filter

The RGB ratio filter is defined as follows:

```
ratio = red/blue;
if (ratio < 0.95) return sky;
else if (val > 200 && sat < 10 && inside_corona) return sun;
else return cloud;
```

The first condition defines a sky classification on the basis of the red/blue ratio alone. The second condition has been included to isolate the sun, as we also included in the HSV filter (this condition could have been written in RGB terms giving the same result).

Both algorithms classify each pixel in the image into one of three possibilities (for the moment we are ignoring other objects that will appear in the image – like horizon obstructions). We have assumed that any high Value pixels are likely to the "sun" if the pixel falls within a region where we expect to find the sun. For clear skies we can then assume that the "sun" is really "sky" for the purposes of estimating cloud coverage. On the other hand we will expect to have situations where very bright clouds actually mask the sun and we will most probably be classified incorrectly.

Our models for segmentation are quite simple and no doubt open to all sorts of criticism, but before we dismiss them entirely we should see how they perform. From the sets of images we have acquired we have chosen a sample of six to demonstrate the performance of the various algorithms. These cover a range of conditions from clear skies through to fully overcast, with both "thin" and "dense" clouds. The results of applying the above empirical algorithms are shown in Figures 1.8(a) to (f), at the end of this section. The regions of cloud are shown in white, and sky in black.

We can make a number of initial observations:

1) Both the HSV and RGB ratio filters give very similar results.
2) The segmentation seems, in general, quite reasonable with clear sky and cloud regions being effectively segmented.
3) The presence of the sun, and its corona, does cause some problems. The black disk results when the algorithm detects that the high Value pixels fall within a region where the sun is expected for that time of day (the second condition in the algorithm described above). Where the sun is actually visible, in samples (a) and (b) the sun is effectively masked, though parts of the corona are not.
4) Where we have very bright, and thin, cloud coverage, as in sample (c), the algorithms conclude that the sun is visible.
5) The images with dense clouds, samples (d), (e) and (f) result in good segmentations, including sample (e) which would be best described as a partially cloudy sky.
6) The cloud coverage in sample (c) is quite well handled, though clearly the segmentation becomes less effective where the cloud is rather "thin" (and we might even debate if it should be classified as "cloud" or "sky").
7) At this stage the horizon obstructions are classified as "cloud". We will return to this issue later.

8) We see a number of image artefacts resulting from the penetration of direct sunlight into the lens system of the camera. These internal reflections are probably unavoidable if we allow direct sunlight in the image. We would view these as not generating significant errors in the following computations.

As we might expect the algorithms being used contains several parameters which have been arrived at from experimentation in an attempt to get good results across a wide range of sky conditions. It will always be the case that improvements may be possible given further fine tuning. Clearly, the parameters used here are dependent on a particular CCD (and perhaps camera) and the exposure settings, and the local atmospheric conditions. In another situation the algorithm will need to be re-calibrated to match the camera and CCD.

We are confident, however, that the concept of relatively simple empirical models can be applied and will give acceptable results for a range of camera and CCD devices. Local calibration and algorithm refinement will necessary in each case.

3.5 Neural Network Approach

Another approach, and perhaps the most obvious one to try, involves the application of neural network concepts. Analogue Neural Networks (ANNs) can be readily applied to pattern recognition tasks. In this case the task is quite simple. We have the data for each pixel of the digital image as RGB values, we have the exposure settings from the camera, and we know (by human judgment) whether a pixel should be classified as cloud or sky. For this model we will extend our classification task to also include the sun as well as the horizon obstructions.

The ANN we have chosen to use is a conventional feed-forward back-propagation network consisting of 4 input nodes, a hidden layer of nodes and one output node. The input nodes will correspond to:

1) The red value of the pixel in the range 0 to 255, scaled to the range –1 to +1
2) The green value of the pixel in the range 0 to 255, scaled to the range –1 to +1
3) The blue value of the pixel in the range 0 to 255, scaled to the range –1 to +1
4) The exposure value for the image defined as the ratio of T/f^2, where T is the shutter speed (secs) and f is the aperture setting. The amount of light entering the lens of the camera, and thus arriving on the CCD, is proportional to the time the lens is open and inversely proportional to the square of the aperture. The ratio T/f^2 is thus constant for a given amount of light entering the lens. This ratio is typically (for the images we have made) in the range 2.7 to 11.1 x 10^{-6}. These values have thus been scaled in the range –1 to +1 where –1 will represent a T/f^2 of 0 and +1 a value of $20x10^{-6}$.

The number of nodes in the hidden layer is of course open to debate and some experimentation. Clearly we need a sufficient number of nodes to provide enough degrees of freedom for the solution searching process, but not too many to "over fit" the network on the data provided for training. We suspect the number of nodes in the hidden layer will be in the range 3 to 5 or 6.

The output node will have a value in the range –1 to +1 which we will segment as shown in Table 3.1:

Output value	Interpretation
-1.0	Horizon obstruction
-0.25	Sky
+0.25	Cloud
+1.0	Sun

Table 3.1: Classification of the Output Node for the ANN

The data for the training (and testing) of the ANN was obtained by extracting representative sets of small sample patches from a number of images, and recording the RGB values and the exposure value. Approximately the same number of sample points were made for each type of output response (ie. sun, sky, cloud and obstruction). Various data sets have been tried (from small, with a few hundred points, to quite large, with many thousands of sample data points). The results seem much the same.

From the collected data, the ANN was trained on half the data points, and the tested on the other half.

The ANN model used appears like that shown in Figure 1.5:

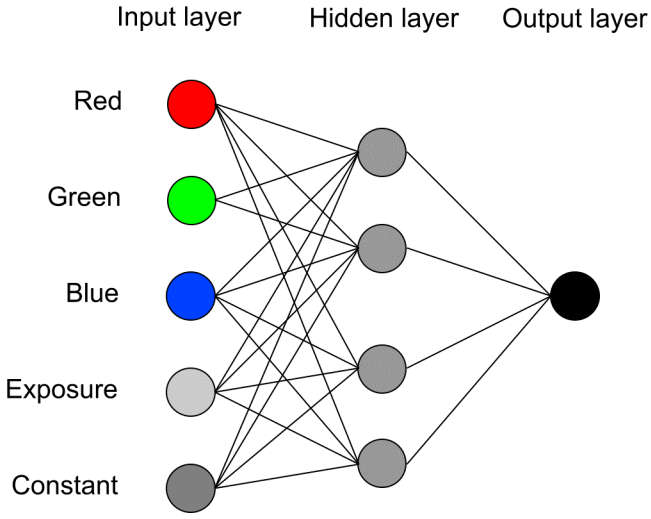


Figure 1.5: The ANN Structure

In this case we have the four input nodes plus an extra node for the constant, and three hidden nodes, plus the extra one to represent the constant. Each node is assumed to have a tanh response function.

For a smallish data set of some 590 sample data points we get the following results from the solution process. The solution converges quite quickly as shown in Figure 1.6, though from observation there are many possible solutions.

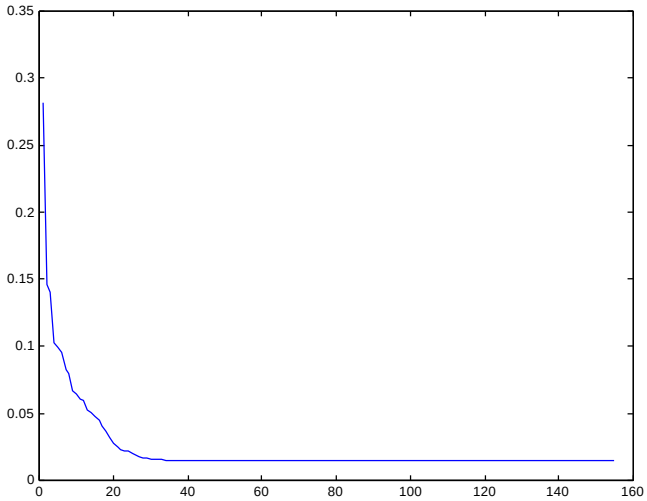


Figure 1.6: Convergence of ANN solution

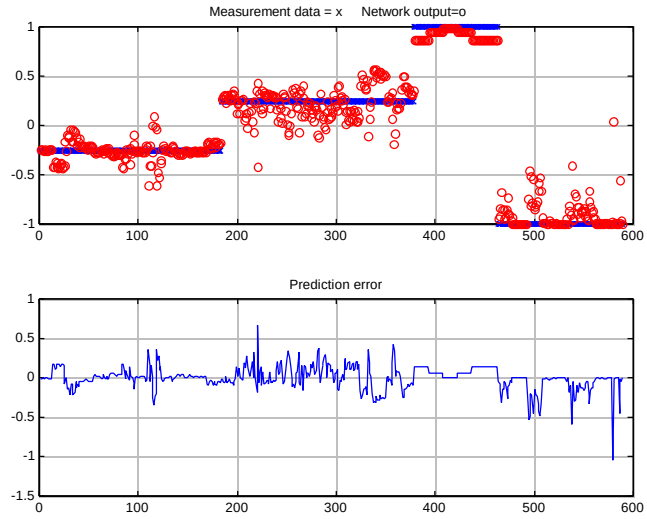


Figure 1.7: Performance of ANN on Test Data Set.

Figure 1.7 shows a typical result from an analysis of the test data set. We can see that, in the main, the ANN is able to classify the data points (the red circles are the predicted values, the blue dots are the actual test set classifications). A prediction error in excess of about 0.25 will generate an incorrect classification. The total number of errors out of 590 test points is 35 in this case.

The results do not look very much different as the number of hidden nodes is increased, though they do get noticeable worse for a number less than 3.

These results look promising, so we should now apply the ANN segmentation to the sample images. The results are shown in Figures 1.8(a) to (f). Here we now see a comparison with the empirical models described earlier. The results are similar, but there are some differences in individual images especially in (c), (d) and (e) where the ANN model has not correctly classified some cloud regions. Though in the case of sample (c) it has done a better job on the rather “thin” cloud regions.

The ANN model has similar problems with the sun's corona region as with the empirical models, as well as classifying regions of bright cloud that fall in the region of the sun's corona.

The ANN also provides a formal segmentation of the horizon effects – as they have been included in the ANN model. The result of this can be seen in Figure 1.8 where the horizon objects are shaded black (i.e. assume sky).

3.6 Comparing the Empirical and ANN Models

On the sample images analysed in Figure 1.8 all filter models appear to offer some potential. Both ANN and the two empirical filters are based on the premise that the sky/cloud segmentation can be derived from the colour information from the pixels in the digital image. Both use some additional information of the position of the sun to discriminate cloud from clear sky conditions where the sun is exposed to view.

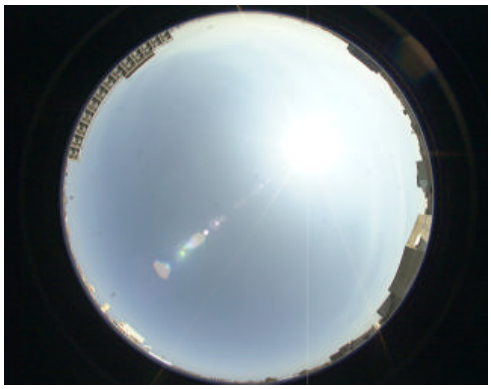
The empirical models probably include, or have the potential to include, more real information on how we (humans) discriminate clouds from sky, even though the model used here is very simple. While further analysis of the empirical model may be able to make it even better it is of interest to recognise that a rather simple model can give quite good results. We recognise that the RGB ratio filter gives much the same results as the HSV model, though we do not have the same intuitive "feel" for how it works[5].

The ANN model, on the other hand, has been derived from a more quantitative analysis and so is more automated and less subject to individual value judgements (though the basic classification of the data for both training and testing is still a manual task with a degree of subjective assessment). The results of the ANN filter model are generally quite good, but perhaps not as good as the two empirical models. We accept that improvements may be possible with better approaches to selecting training and test data sets – which are central to the ANN training process.

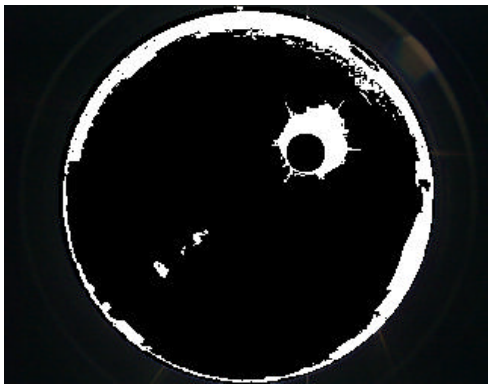
From the results shown in Figure 1.8 the empirical models are giving marginally better results, though we do not propose that this will always be the case. In other situations (CCDs, cameras and locations) the reverse might be true. It is clear, however, that both approaches offer some potential.

For all segmentation filters, but perhaps more so for the ANN model, the classification process produces some local variations and the regions of cloud (and sky) are often dotted by classification errors. We need to attend to this problem; firstly to "correct" the classification and secondly to assist in the subsequent cloud analysis. If we do not try to "tidy up" the segmented images our cloud analysis may become unnecessarily complex – caused by large numbers of small patches of cloud and/or sky. We will look at this problem in the next Section.

[5] The use of the red/blue ratio is based on the proposition that the light eminating from clouds has a larger red component than that for clear sky.

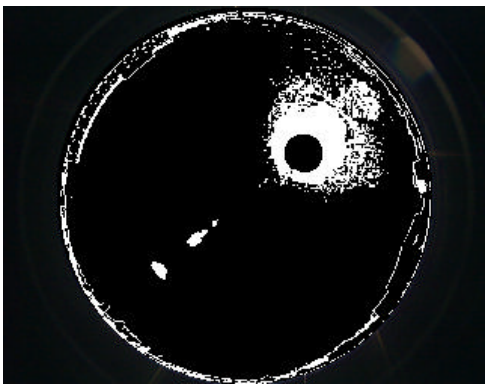


(a) 1998_02_24_14_01



HSV Filter

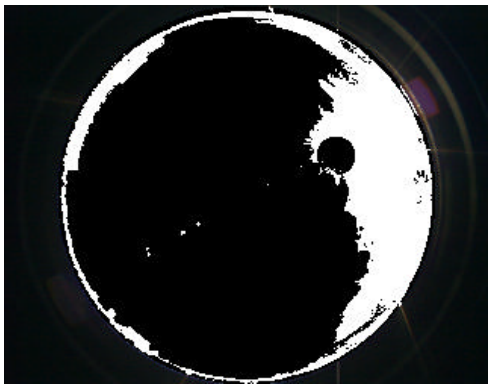
RGB ratio Filter



ANN Filter

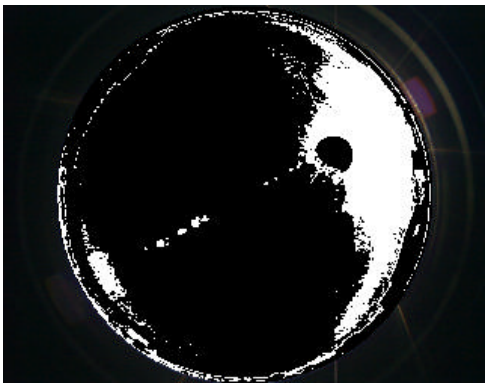
Figure 1.8(a): Comparison of Empirical and ANN Segmentation Algorithms

(b) 1998_03_03_14_59



HSV Filter

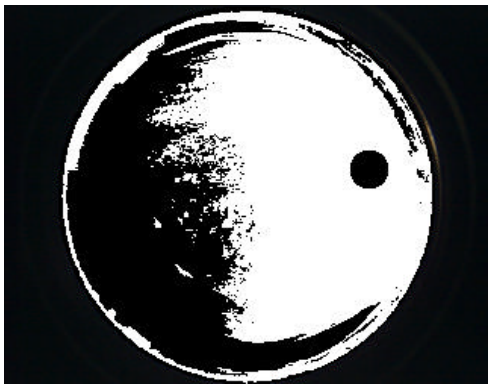
RGB ratio Filter



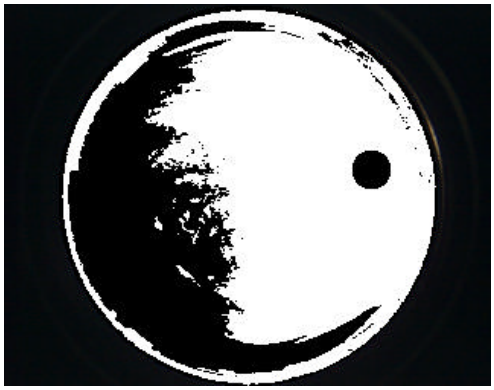
ANN Filter

Figure 1.8(b): Comparison of Empirical and ANN Segmentation Algorithms

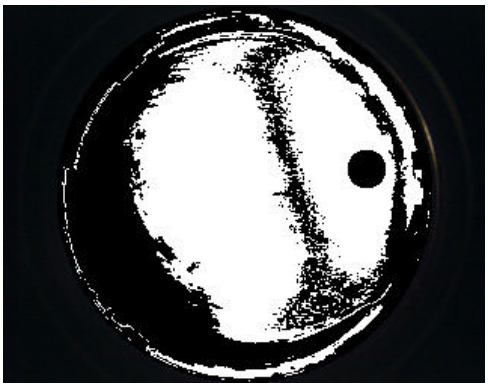
(c) 1998_03_03_16_01



HSV Filter

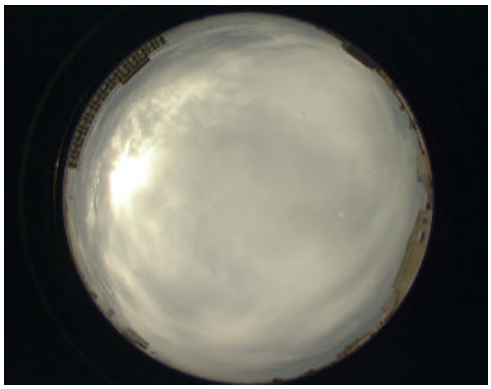


RGB ratio Filter

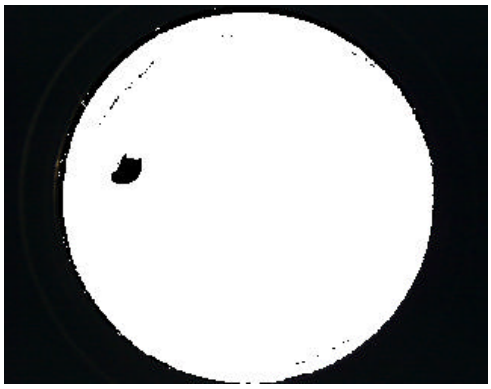


ANN Filter

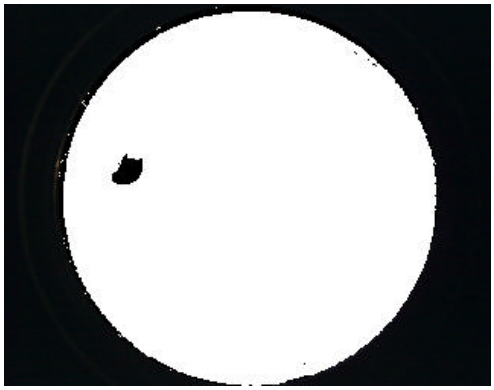
Figure 1.8(c): Comparison of Empirical and ANN Segmentation Algorithms



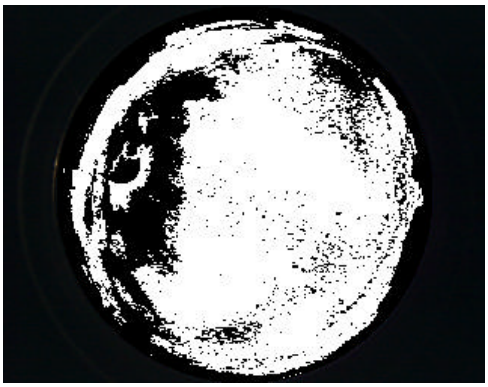
(d) 1998_03_05_08_15



HSV Filter



RGB ratio Filter



ANN Filter

Figure 1.8(d): Comparison of Empirical and ANN Segmentation Algorithms

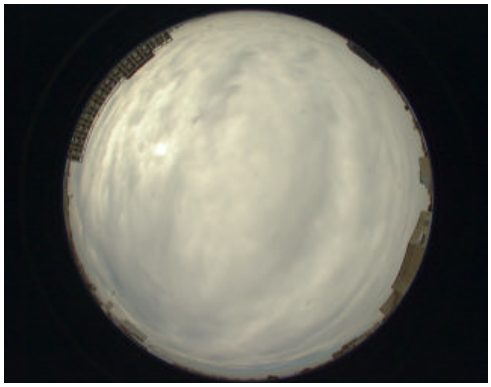
(e) 1998_03_05_14_45

HSV Filter

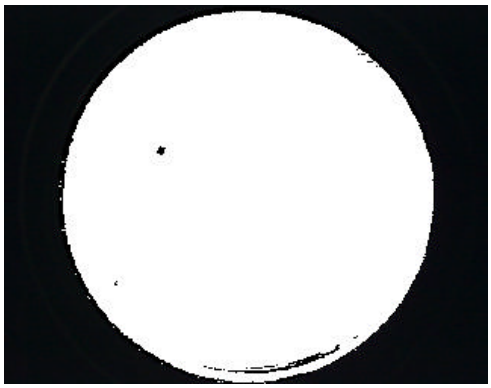
RGB ratio Filter

ANN Filter

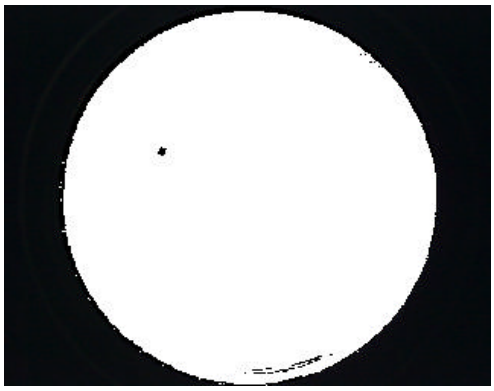
Figure 1.8(e): Comparison of Empirical and ANN Segmentation Algorithms



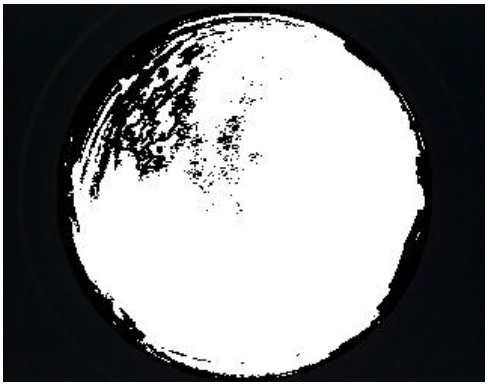
(f) 1998_03_06_09_30



HSV Filter



RGB ratio Filter



ANN Filter

Figure 1.8(f): Comparison of Empirical and ANN Segmentation Algorithms

4 Image Correction and Tidying

4.1 Introduction

From the segmentation process we get an image with all pixels classified as cloud or sky. Given the nature of the data there are bound to be errors in this process. Individual pixels will be misclassified resulting in many, often small, patches of sky within cloud regions and cloud within sky regions. Of course, some of these results may be correct as there is nothing to say that we cannot have many small patches of cloud and/or sky.

Essentially, if we have a single cloud pixel amongst a region of sky, or vice versa, then we might conclude that this pixel has been misclassified at best, or can be ignored for the purposes of simplifying the image. We thus need to impose a set of rules for making such judgements so that the whole image can be suitably processed and not unduly distorted.

Our goal is then to tidy up the image, reducing the number of cloud patches to a minimum consistent with maintaining the best possible overall classification.

4.2 The Convolution Process

To perform this tidying-up process we have adopted an approach using a convolution mask which is passed over the whole image. This mask makes an assessment about each pixel, and will determine if the classification of the pixel should remain or be changed. The convolution mask works on a voting principle. For any given pixel, if there are sufficient numbers of neighbouring pixels that have a common value (cloud or sky) then it is most probable that the selected pixel should also have the same value.

Figure 4.1 shows a typical example from an image after segmentation, with cloud pixels being shown as white. The edge of the cloud is not always well defined and we have pixels with apparently incorrect classifications.

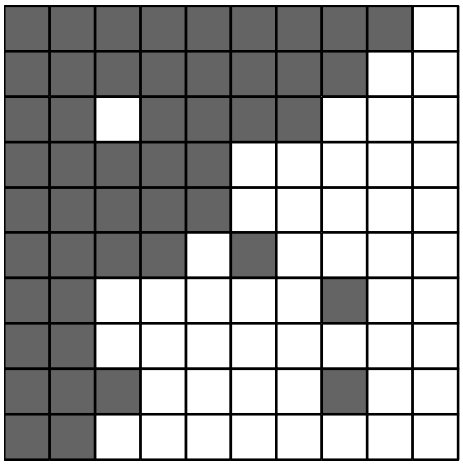
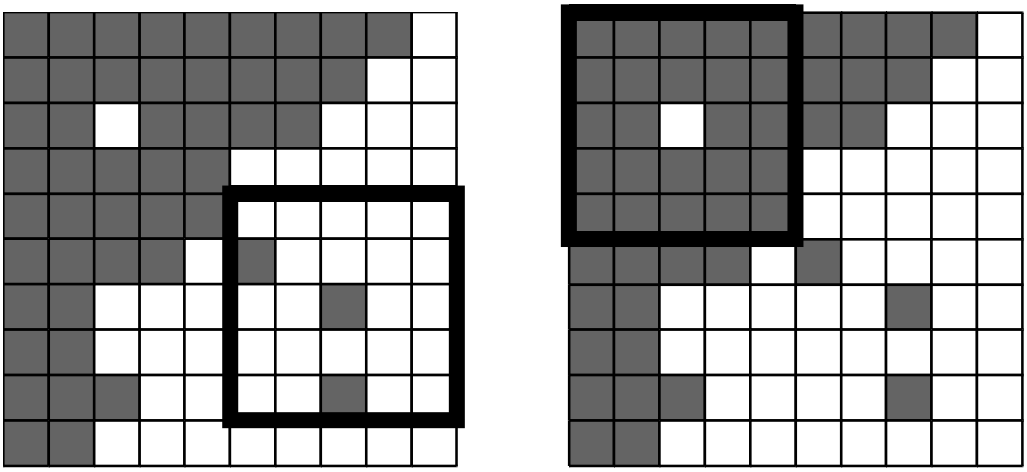


Figure 4.1: Typical Result After Segmentation

A convolution mask is a square of some (usually fixed) number of pixels which is placed over each pixel in the image and we focus on the central pixel. In Figure 4.2a

we have a 5x5 mask with a sky pixel at its centre. Within the mask we see that most of the pixels are in fact cloud, so we then suggest that it is most likely that the pixel at the centre should also be of type cloud. The converse is true in Figure 4.2b



(a) *(b)*

Figure 4.2: Placing the Convolution Mask on the Segmented Image

In these cases the conclusion is reasonably obvious, but that shown in Figure 4.3 it is not so clear. In this case we must be more careful about the decision taken.

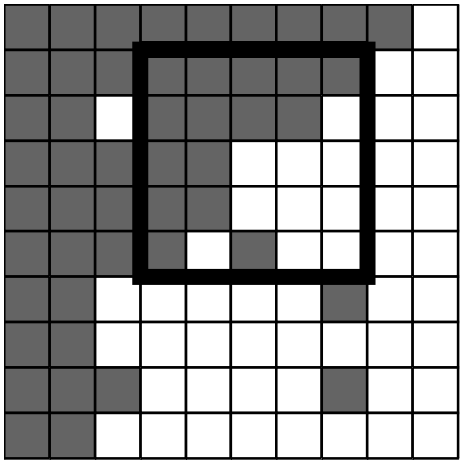


Figure 4.3: Placing the Mask Near a Cloud Edge

In applying the convolution mask we must therefore have a decision making strategy. We will do this by considering the voting pattern for each pixel being considered (i.e. the pixel at the centre of the mask). We also need to consider the size of the mask (3, 5, 7 etc).

The mask size must be a odd number of pixels so that the centre pixel is defined for each position of the mask. If the mask is too small we will not get a "big" enough picture of the local conditions (i.e. mix of cloud and sky pixels), if too large the mask becomes ineffective as the decision rules become less well defined. The computational cost also increases with the square of the size of the mask.

It has been our experience that a 5x5 mask works quite effectively on images of 500x500 pixel resolutions. For much lower resolutions (say less than 100x100) a 3x3 mask may be better suited.

For each position of the mask, we can count a cloud pixel as 1 and a sky pixel as 0. The cell at the centre can thus receive a vote somewhere between 0 and 25 (if we count the central cell itself in the voting. If the vote is 25 then there are no debates, the cell is, and remains, a cloud type. If the vote is 0 then cell is, and remains, a sky type. We must however decide on all intermediate possibilities. After some experimentation we have adopted the rules given in Table 4.1

Vote value	Decision
<= 7	sky
> 7 && <= 16	No change
> 16	cloud

Table 4.1: Decision Rules for the Convolution Mask

This scheme has divided the voting range into 3 parts, the low-range where we expect that the pixel is most likely sky, an upper-range where we expect the pixel to be most likely cloud, and a middle range where we don't have enough confidence to change the pixel from its current value. These cut-off values are quite arbitrary, though we have found they work quite well in practice.

The convolution mask is passed across the whole image, processing each centre pixel in sequence. The mask always operates on the existing image, a new image is created to contain the modified pixel values.

The convolution process may have the effect of moving the cloud edges if the voting level decision points are not well chosen. It is not too difficult to imagine that the edge between sky and cloud could easily be shifted under the convolution process. In fact it will naturally be shifted by the convolution, but we need to be careful not to make the shift always in one direction (say towards or away from the cloud mass). If this happened then we would be adding a distortion to the image perhaps increasing or decreasing the size of the cloud patches.

We will be able to see the impact of applying the convolution mask by repeatedly applying the mask. This will magnify the effect, if the mask is making significant changes to the position of the cloud edges. We might also contemplate applying the mask several times if we can see a further improvement to the image.

Figure 4.4 shows the effect of changing the upper and lower decision points in the convolution process on a synthetic image. When the upper value is set to 12 and the lower value to 11 (i.e. effectively having no "undecided" interval) we see that the edge changes further from the original as each convolution is imposed. The third case, (c), is where changes are allowed only in quite extreme situations (upper = 20, lower = 3). No changes are observed in this case. The middle option, (b), is our recommendation (i.e. upper= 16, lower = 7). Here we see some small changes, but they appear not to extend with subsequent convolutions.

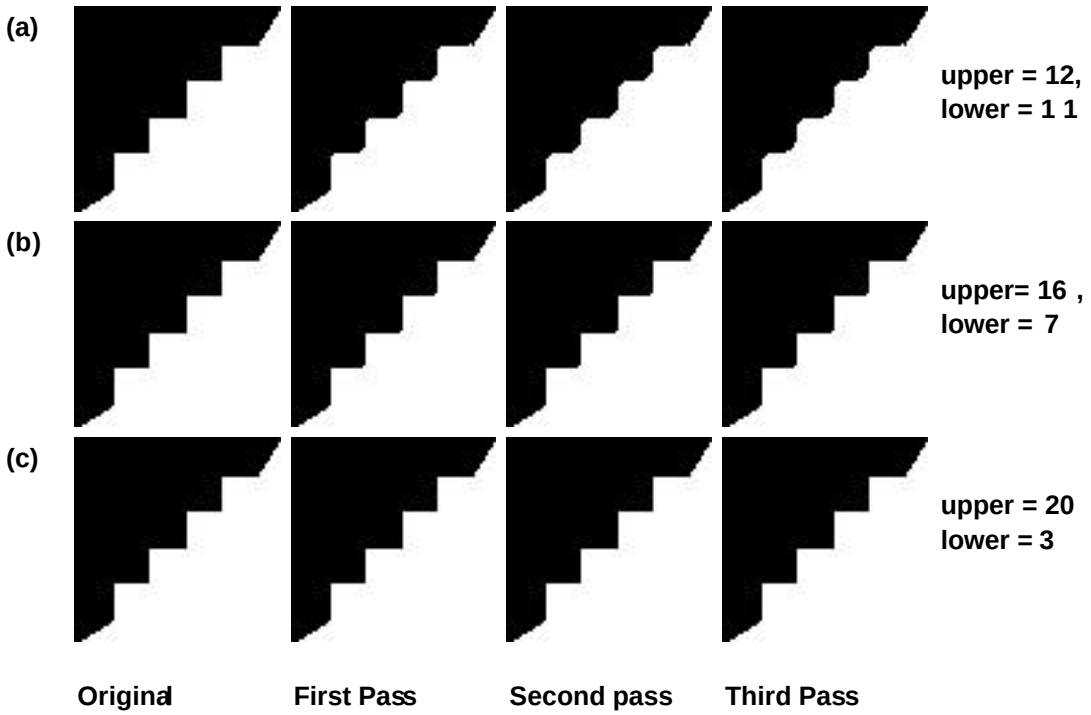


Figure 4.4: Effect of Changing Convolution Mask Decision Points

While the (c) option in Figure 4.4 would seem the best in terms of controlling edge altering effects in the convolution it will do little in tidying up the image from odd sky or cloud pixels, which we know we get a lot of in most images. We have thus chosen the decision points of 16 and 7 as proposed earlier.

The decision rules suggested above appear work well in real images. We can test them by repeatedly applying the convolution and looking at the results. Some results are shown (magnified a little) in Figure 4.5.

The first thing to note is that the first pass has the most dramatic effect. Subsequent passes cause little change, and as we can see in this example the cloud edges appear quite stable under several convolutions. This is what we want. It is also of interest to know that the first pass achieves most of the desired effect, probably eliminating the need for further processing. It would be our view that a single convolution is sufficient to tidy the image for the cloud edge detection.

4.3 The Results

Some typical results for complete digital images are shown in Figures 4.6 and 4.7. The (a) Figure shows the raw segmented image and (b) Figure the image after the convolution mask has been processed across the image. We see here the desire effect. Many of the small clusters of pixels have been eliminated and the cloud edges are generally much tidier.

Segmented Image

First pass convolution Second pass convolution Third pass convolution

Figure 4.5: Repeated Application of the Convolution Mask

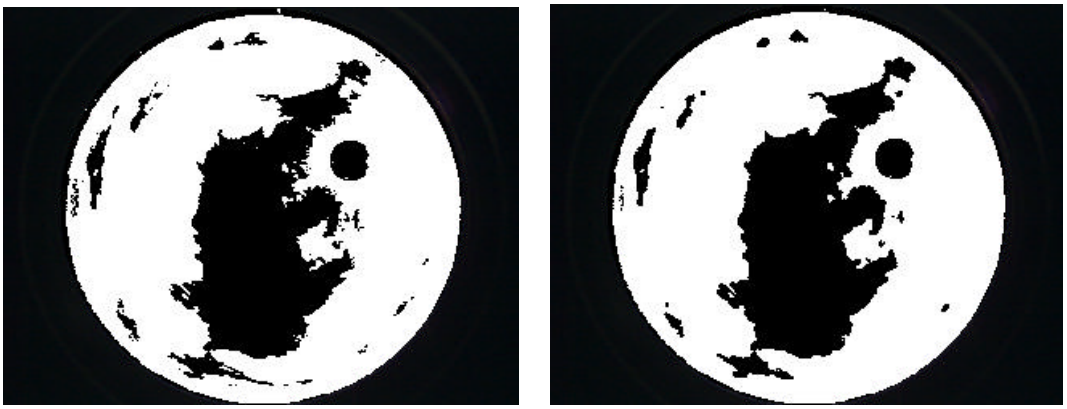


Figure 4.6: (a) After Segmentation (b) After Convolution

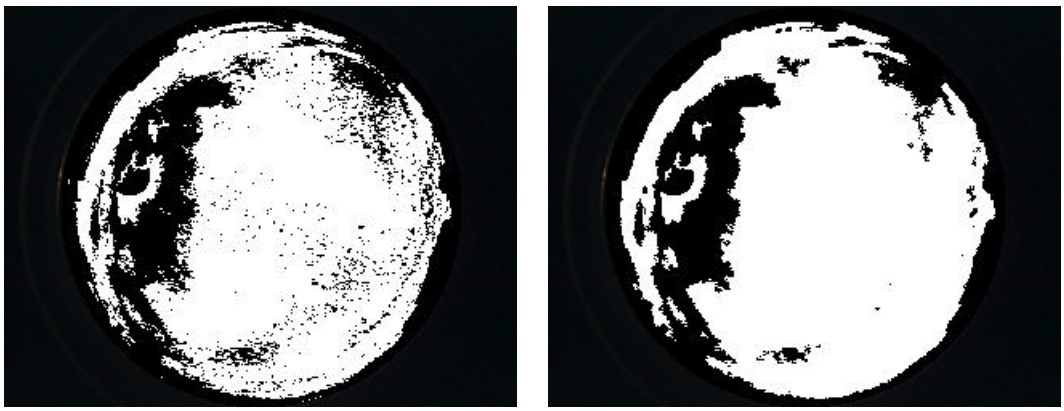


Figure 4.7: (a) After Segmentation, (b) After Convolution

We can thus conclude that this convolution process is achieving the desired goals of tidying up the image without having significant effects on the edges, apart from generally smoothing them.

4.4 Horizon Effects

In an ideal situation the digital images should be free from horizon effects, i.e. the image covers the full sky dome from horizon to horizon. Apart from very special locations, this is unlikely to be the case. Clearly, if we are focussing on sky conditions then we need to choose locations that maximise the visibility of the sky dome. In other situations we may wish to deliberately include the horizon effects where, for example, we are focussing on the daylighting conditions for a particular facade of a particular building in a particular location. In this case we would wish to represent as accurately as possible (including luminance properties) the impact of the horizon effects (vegetation, buildings etc).

For this project our interest is primarily on the sky dome, and our camera location (on the roof of the School of Architecture at the University of Sydney) provides a reasonably good view of the complete sky dome. Figure 1.1, included earlier in this report, shows the objects visible around the horizon. We have measured to total area of these effect, they amount to about 3% of the area of the sky dome.

As indicated previously we need to make some assumptions on how to handle this effect. We need to either assume that we remove that part of the dome from the analysis, or assume that behind the obstructions we have cloud or clear sky. There is no obvious solution to this problem. It is most likely that the decision to be take will depend on local conditions and the scale of the possible errors introduced by the horizon effects. There are many possible strategies, for example, here are a few:

- Assume clear sky behind the obstructions
- Assume cloud behind the obstructions
- Remove the obstruction area from the cloud coverage calculation, thus estimated the cloud coverage on the “visible” portion of the sky dome.
- Apply some “intelligent” decision based on the sky conditions near the edges of the obstruction.

We do not support any particular point of view, but leave that to the local context and the purpose for which the skies are being analysed to determine the best way to go. We are interested, however, in showing how we can use the data from the digital image to assist with defining the location of the horizon effects.

In the previous chapter we indicated that the neural network segmentation model included an output specification for horizon obstructions. Given that the colour of the horizon obstructions could be anything, we would suggest that the simple segmentation schemes (HSV filter and RGB ratio filter) are not going to leed to useful results. We have found however that the ANN model does a reasonable job at identifying the obstructions, though it is not perfect. We can see the effect of applying the ANN filter in Figure 4.8 where we have now shown the pixels classified as horizon obstructions coloured green.

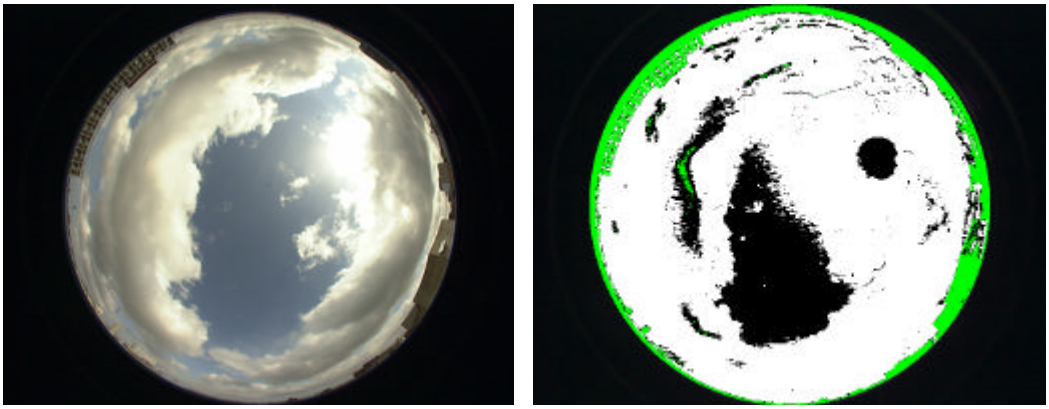


Figure 4.8: Segmentation showing horizon obstructions from ANN
(a) original image, (b) segmetation of horizon obstructions

Comparing the digital image with the segmented image shows a reasonable identification for the horizon obstructions, but it is by no means perfect. We also know that the results may vary considerably with lighting conditions (bright sunlight, dull overcast etc). We are face with two strategies at this point:

- Try to refine the segmentation process to identify the obstructions
- Use a more pragmatic approach on the basis that for a given location the position horizon effects do not change (at least we can assume this, outside major neighbourhood developments).

We have in fact adopted the second approach. It works like this.

Using a segmented image like that shown in Figure 4.8(b) as a starting point, and with the assistance of a paint program[6], create an image which includes the horizon obstructions designated by a colour on the appropriate pixels (green for convenience). The use of the paint program enables us to make corrections to the automatically generated image and to clean up the image where the ANN segmentation has failed to correctly discriminate the obstructions. We view this pragmatic approach as being reasonable and effective given that once constructed we can use the same mask on all images captured at that location (at least until there is some change in the horizon obstructions)

We have done this for the Sydney location. The mask can be loaded as required and, depending on the masking strategy (assume sky, cloud, etc.) then the image being processed can be modified accordingly. We would normally load the mask after completing the convolution process. An example is shown in Figure 4.9.

[6] Any image manipulation program that will allow pixels to be painted a set colour, will do.

Figure 4.9: Effect of Imposing an Horizon Mask
(a) Segmented image prior to imposing mask, (b) after imposing mask

In this case we assume that the sky behind the horizon obstruction is clear sky for the purposes of computing cloud coverage. The area of the mask is about 3% of the sky dome area.

5 Outlining the Clouds

5.1 Introduction

The segmentation process produces an estimate of which pixels in the digital image are cloud and which are sky. The convolution process tidies up the image to remove odd cloud/sky pixels in regions where the other classification dominates. Our starting point is thus a "clean" image classified as sky or cloud. While the complete image provides a clear description of the cloud coverage we need to undertake some further analysis to:

- provide a concise way of describing where the clouds are, and to
- compute the cloud coverage over the whole sky on in designated regions of the sky dome.

In the context of SDF modelling we will represent the clouds by sets of polylines describing the outer edge of each cloud patch as a set of vectors across the sky dome. This offers an efficient way of describing the complex boundary conditions, including clouds with "sky holes". We can also use the same geometric representation to compute the areas of the cloud patches, and thus the total cloud coverage.

5.2 The Algorithms

Given a relatively clean segmented image the cloud outlining algortithm proceeds as follows:

```
Mark the outer boundary of the image with sky pixels.
For each pixel in image:
       If the pixel is at a cloud/sky edge
              Mark pixel (e.g. change colour to red)
              Move to next edge pixel and mark it as an edge pixel
              Add pixel to edge list
       Until back at starting pixel
       Mark all cloud pixels inside edge as processed (by a flood fill)
Until all pixels are processed.
```

Figure 5.1 shows this algorithm in process. The outer edges of the image is marked as sky pixels. The first edge found (processing the image from the top left , across then down) has been found and marked. The next edge has been found, and will then be flood filled (only the cloud pixels). Finally the hole in the cloud will be found and then marked. Once all the pixels are marked we have found all the cloud edges.

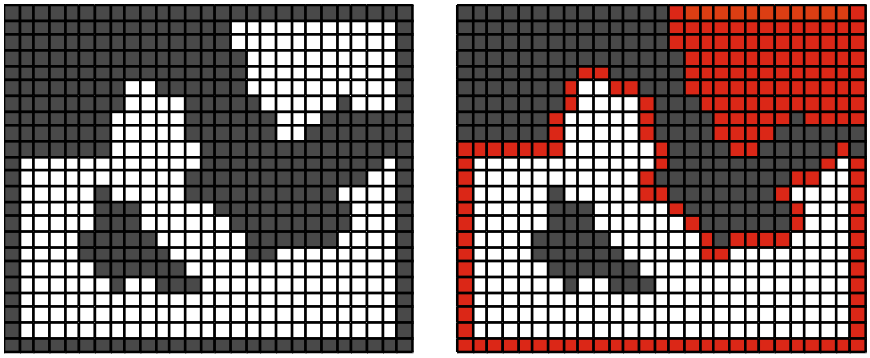


Figure 5.1: Marking the Cloud Edges

Even with a relatively clean image we will end up with quite complex edge descriptions, containing several hundred (if not thousand) edge segments. Each edge segment will be just one (or $\sqrt{2}$) pixel long. While this might be considered "accurate" it is most likely quite unnecessary in terms of the accuracy of the image and the segmentation process. We will end up with edge polyline definitions with far too much detail. The next step will simplify, or smooth, the edges to some acceptable level.

Figure 5.2 shows what we are trying to achieve. The first image shows the edge that we might get from the above process, the second image shows reasonable approximation with many fewer edge segments and little loss of detail.

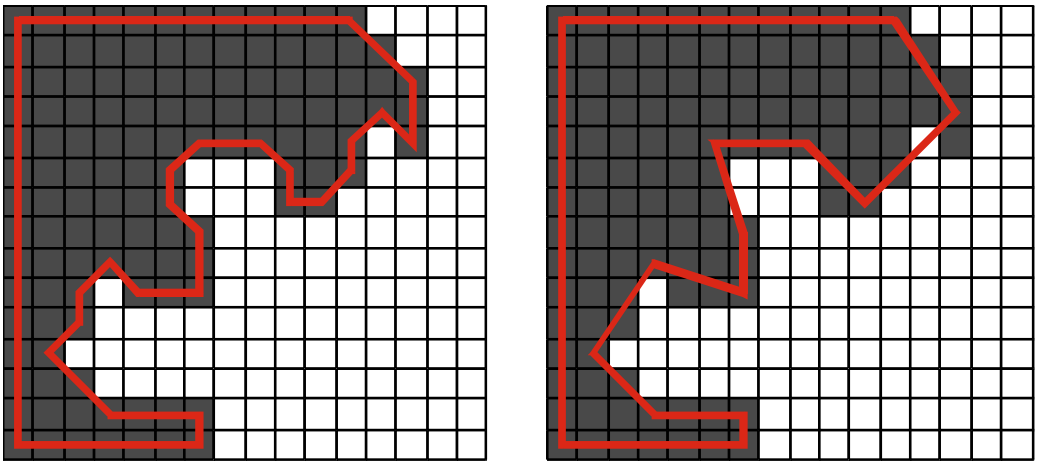


Figure 5.2: Smoothing the edge polyline

Consider the edge, A-B-C-D-E, definition in Figure 5.3. From the starting point (A) we have the choice of following the edge to B, or cutting off a part of the shape and go straight to C. If the area contained within the triangle ABC is sufficiently small then we might decide to make the approximation and drop the vertex C from the edge definition. We might even continue on and make the new edge go to D or E if the area of the polygon cut off is small. Clearly if the edge A-B-C-D-E is a straight line then the approximation A-E will result.

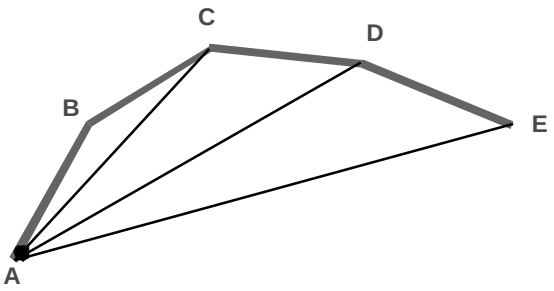


Figure 5.3: The Edge Smoothing Process

The decision to drop a vertex will depend on the area of the polygon shape to be cut off by the approximation. We compute the number of pixels in this triangular shape at each step, and using a "reasonable" value we can control the level of approximation required for the given application. This parameter will be called the "pruning factor".

Another form of simplification concerns the presence of degenerate polygons (e.g. polygons with 2 sides). It would seem sensible to eliminate such polygons. We might also consider also eliminating polygons with 3 or 4 sides. In the context of cloud edges these will probably only appear due to artefacts of the image analysis process. From experience we have typically selected a degenerate limit of 4, this means that polygons with 3 or less sides are ignored. The value can, however, be set to suit user requirements.

Once the outlines of the clouds are known we can compute the area of each polygon shape (on the sky dome) using the functions from the SDF modelling package, see Roy [1998]. Summing these for each cloud patch gives the estimated total cloud coverage.

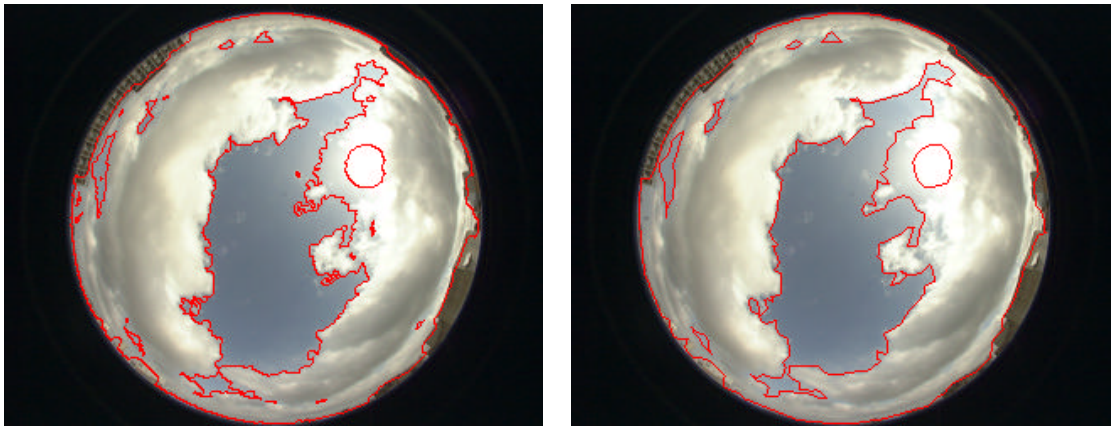
5.3 Some Results

To demonstrate the operation of the above algorithm and various values for the pruning factor we have some results in Figures 5.4 (a) to (d), with the cloud outlines shown in red.

The (a) option shows the result when the pruning factor is set to 0, i.e. no pruning is done and we see a very detailed edge definition with a large number (1576) edges and 27 different polygons being found. The (b) option shows what happens when the pruning factor is set at 50 (ie. the maximum area of polygon pruned is 5 square pixels in area[7]). The number of edges is less (356) and we have removed most of the very small polygon shapes. The (c) and (d) options show results for a pruning factor of 200 and 400 respectively. At a level of 400 we see some obvious approximations appearing – but this still may be acceptable, depending on what accuracy of results is required, or can be justified from the previous image processing.

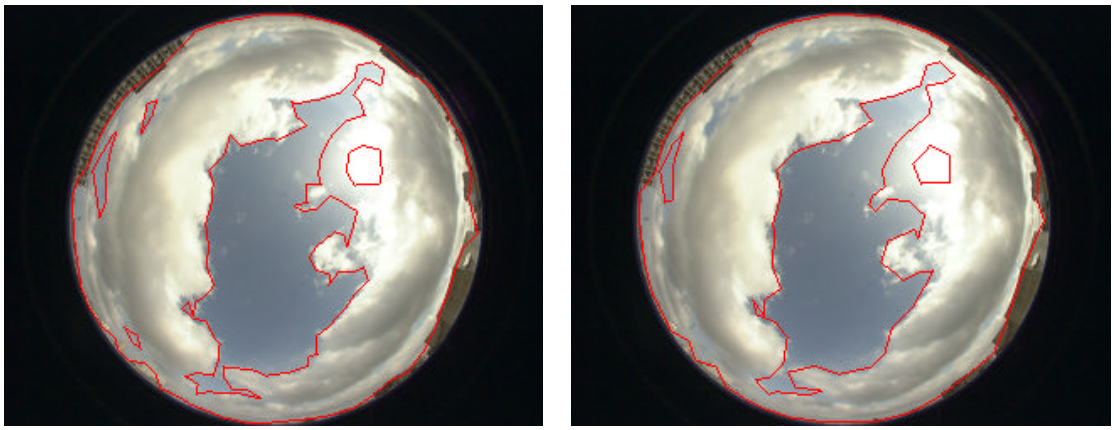
In some ways the required accuracy will depend on the cloud coverage computations. The results are given in the captions in Figure 5.4. It would seem that that the results are quite simular even up to a pruning factor of 400. It would be expected, however, that for much larger values the error in the cloud coverage could increase significantly.

[7] The pruning factor is measured in 10ths of a square pixel.



(a)
Pruning factor = 0
Degenerate limit = 4
Number of polygons = 27
Number of edges = 1576
Cloud coverage = 65%

(b)
Pruning factor = 50
Degenerate limit = 4
Number of polygons = 8
Number of edges = 356
Cloud coverage = 66%



(c)
Pruning factor = 200
Degenerate limit = 4
Number of polygons = 6
Number of edges = 176
Cloud coverage = 66%

(d)
Pruning factor = 400
Degenerate limit = 4
Number of polygons = 4
Number of edges = 117
Cloud coverage = 67%

Figure 5.4: Example results show cloud edge analysis for varying pruning factors.

6 Luminance Mapping

6.1 Introduction

The second major interest in this project is to extract luminance information from the digital images. The calibration of the CCD is given elsewhere in this report. The results of this calibration gives us an empirical relationship between the RGB values at a pixel and the luminance level in Cd/sqm. We must accept that this calibration can only give us an approximate estimate of luminance limited by:

- The response of the CCD to light in the visible spectrum.
- Variation in response across the CCD.
- The dynamic range of the CCD is considerably less than that of the human eye, though perhaps not too far short of the luminance range of typical skies – providing we ignore the solar region and its immediate corona and some "white" clouds which are reflecting high levels of light. Never-the-less these issues are significant.

The attractive feature of the CCD is the resolution it gives us. Opto-mechanical sky scanners do not allow us to measure the luminance levels at high resolutions. The Krochmann PRC scanner, for example, measures the average luminance over a region of about 10° of the sky dome. Measurements at this level of detail have formed the basis of most luminance measurements and sky modelling work in recent years. The opportunity to analyse skies at much finer details is thus an interesting prospect.

Berutto and Fontoynont [1995] have demonstrated the use of CCD imagery to map luminances for interior building spaces. They have shown that it is possible to obtain reasonable calibrations for their (monochrome) CCD camera, though it seems a somewhat different result than we have found for our camera (as described in Section 2). They were also concerned about the projections accuracy of the CCD and relating the pixel positions to points in the 3-D space of the objects being imaged. We are less concerned with this issue as the sky dome maps onto the CCD image plane with an equi-angular projection. There may well be optical distortions, but we have not detected that they are significant for our purpose.

The first task is to be able to produce a luminance map of the sky dome, i.e. transform the digital image into a luminance value (e.g. Cd/sqm) which can then be directly used for further lighting and daylighting analysis.

6.2 Image Analysis

Our digital images are available with 1000 pixels of resolution, though we typically process them at lower resolutions (500 or 250, for example). We have found this resolution reduction quite acceptable while reducing the computation times (and data storage requirements) quite substantially. The camera calibration is given by the empirical relationships (as previous presented in Section 2):

$$L = \frac{V^{2.4}}{2901.978 \cdot Ev}$$

$$V = 0.2125 \cdot R + 0.7154 \cdot G + 0.0721 \cdot B$$

$$Ev = \left(\frac{179p}{200} \right) \cdot S \cdot \frac{T}{f^2}$$

where:

- R, G and B are the RGB values of the pixel (in the range 0 to 255).
- V is called the RGB brightness.
- Ev the exposure value.
- S is the film speed (equivalent) of the CCD (ISO800 in our case).
- T is the shutter speed (seconds)
- f is the aperture stop value for the lens.
- L is the luminance in Cd/sqm.

A luminance map can now be quite simply produced by applying this transformation to each pixel in the image. Some example results are shown in Figure 6.1. In each case the luminance map is scaled (linearly) from black (0) to white (20,000 Cd/sqm).

The grey-scale shading used is quite arbitrary as we know that we should use a non-linear scaling if we are to try to model the response of the eye. This however would create problems in trying to see the results on the screen or on paper as we add additional response/representation issues, each of which have their own problems. We judge it best to use a simple linear scaling so that we can optimise the display of the luminance variations across the image. That is, we can see a little more clearly the luminance mapping, though not necessarily the real perceived luminance distribution.

The limited dynamic range of the CCD (about 30,000 CD/sqm) does give rise to some concern. One approach to increasing this range is to take multiple images with different camera exposure settings. Unfortunately we have not had the resources in this project to pursue this investigation. The process is quite simple once the camera calibration is done. If at the longest exposure, a pixel is close to saturated, then the next image is examined and the its calibration used. It is also possible to use information from multiple exposures to estimate an “average” value even if the pixel is not saturated.

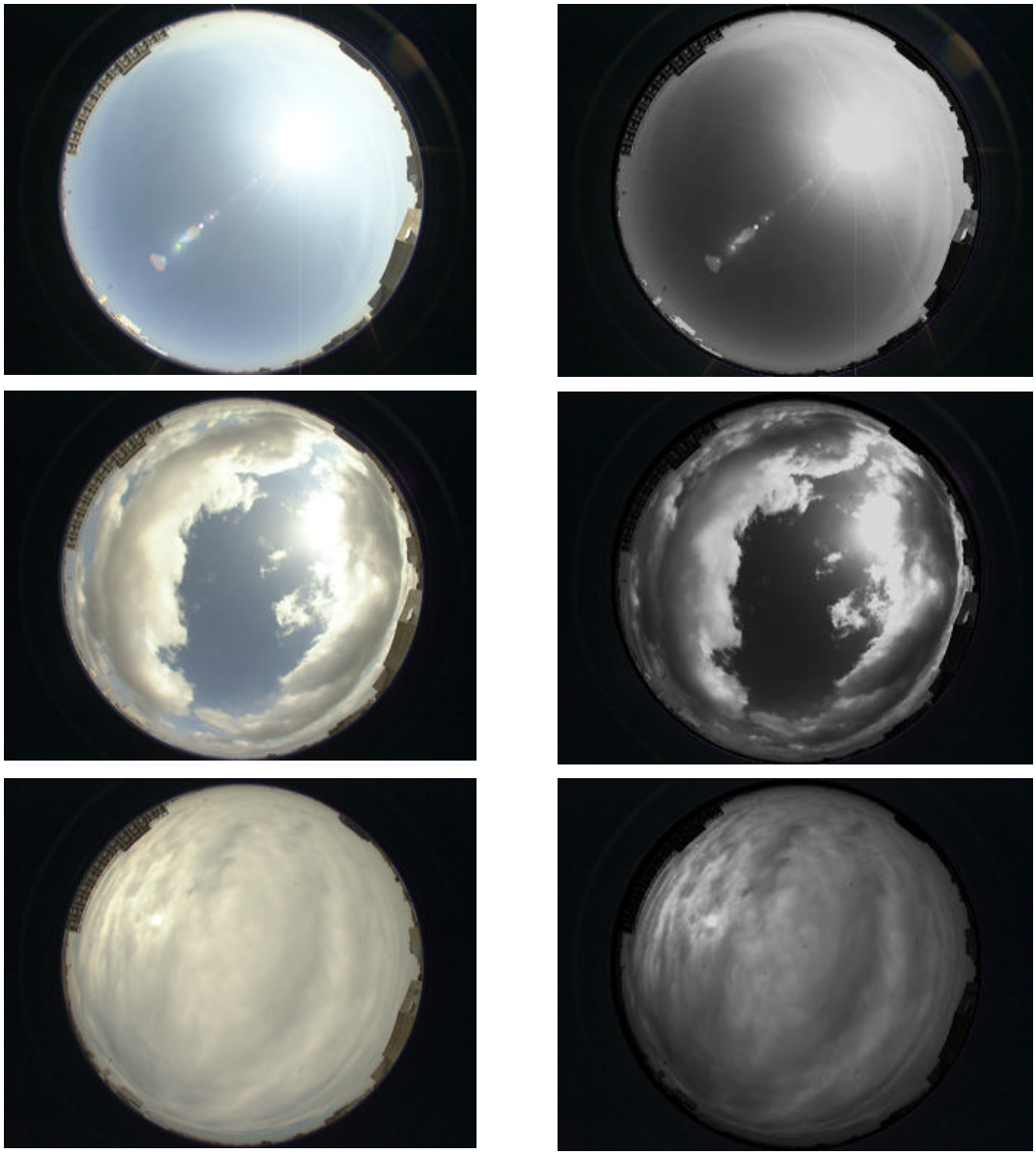


Figure 6.1: Some example luminance maps
(left) digital image ,(right) luminance map

7 Modelling in Standard Digital Form (SDF)

7.1 Introduction

The detail of SDF is given elsewhere, Roy [1998], though it can be summarised as follows:

- An SDF model is representation of the luminance distribution of the sky dome using iso-luminance contours.
- Within the SDF modelling framework a set of functions is provided to create, save, retrieve, manipulate and analyse a sky model.
- SDF models are represented within a user's computer program as a pre-defined data structure composed of sets of named parameters. These parameters include the essential luminance data (as iso-luminance contours), edge information (to describe cloud edges and horizon effects) as well as a sets of user defined parameters which can contain any data required to be stored in the SDF model.
- Persistent SDF models exist as disk files using a native binary format. An SDF files can contain one or many SDF models which can be selectively retrieved as required by the user.

The SDF model provides a relatively efficient way of capturing most of the important aspects of the sky image in a way that it can be readily analysed in third party application programs. The SDF concept was designed to provide an efficient way of storing and retrieving quite complex luminance distributions with minimal loss of information. This compression process become particularly important when working from digital images. A 1000x1000 digital image is typically about 3.2 Mbytes in size (without compression), whereas a good quality SDF model representing the same data is likely to be in the range 10 to 50 kbytes. There is always some loss of information, but we believe that it is minimal. The compression parameters can be set by the user, thus allowing SDF models with varying accuracies to be created.

Once in SDF format, sky models can be efficiently stored and retrieved and analysed using a range of SDF functions. In particular there are functions to compute the luminance level at any azimuth, altitude position, as well as integrating luminance values over regions of the sky dome. These are the properties that are most often required from a sky luminance model.

7.2 Building SDF Models from Digital Images

7.2.1 Choosing the Grid Size

The SDF modelling process builds the contour model from a triangular grid of points. This grid is triangulated according to the Delaunay principles, as modified and implemented by Reid[1997]. This ensures that the resulting triangles are well conditioned, i.e. as close to equilateral as possible. The accuracy of the resulting model will naturally depends on the size of this grid.

The analysis grid is defined as a set of approximately equally spaced points over the sky dome. The grid spacing is defined by an angular measure at the horizon. If this measure is *Gs* (degrees) then the first set of grid points is located at an altitude angle

of Gs/2 (i.e. just above the horizon) and at an azimuth interval of Gs. The altitude angle increases in steps of Gs, and the azimuth angle by:

```
Az_interval = Gs*(90/abs(90 - Altitude));
```

plus an extra point at the zenith (Altitude = 90°). This results in an approximately equally spaced grid over the sky dome. Table 7.1 shows how the number of grid points relates to the grid space (Gs):

Gs	No of grid points
3°	1800
5°	648
10°	164
12°	113

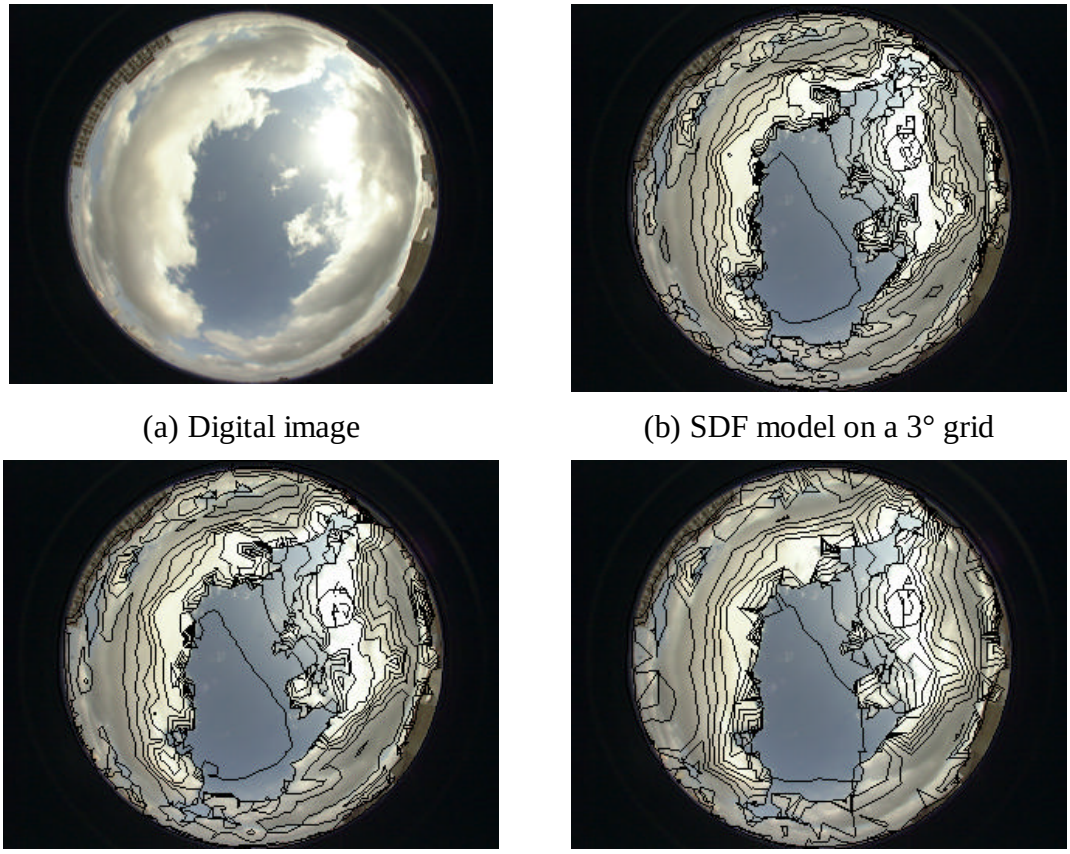
Table 7.1: Numbers of grid points for grid space size

The resulting distribution of grid points is quite similar to that proposed by Tregenza [1987], (for 145 points, but not quite equally spaced) for sampling using opto-mechanical sky scanners like the Krochmann PRC scanner. In our case the grid is uniformly spaced and can be automatically generated from a user selected Gs parameter. For a digital image we have the choice of varying grid size down to the single pixel level. This is most probably unnecessary and would simply create a very complex model at marginal gains in accuracy.

From our experimentation, a 3° grid would seem to be quite fine enough. Even coarser grids may be considered to give acceptable results. Figure 7.1 shows the results for three grid sizes (3°, 5° and 10°).

A 500x500 pixel image has some 200,000 pixels, so we probably need to apply some averaging to derive the luminance values at each grid point. This will also help minimise the impact of any pixel-to-pixel response variations. We have chosen to average the luminance across a 3x3 pixel cell centred at the grid point. Obviously, a too-large averaging cell will reduce the resolution of the resulting luminance model. The models shown in Figure 7.1 have been build using an image sample averaged over a 3x3 pixel cell at each grid point to derive the luminance values to build the SDF model.

Given that the contours are drawn with straight line segments, the 3° model gives a rather detailed contour map. The 10° grid shows some artefacts which have resulted from the larger sized triangles used for constructing the contours. We suggest that the 5° grid would be a reasonable result, though the ultimate test should be based on the errors in extracting luminance values rather than the “appearance” of the model.



(a) Digital image (b) SDF model on a 3° grid

(c) SDF model on a 5° grid (d) SDF model on a 10° grid

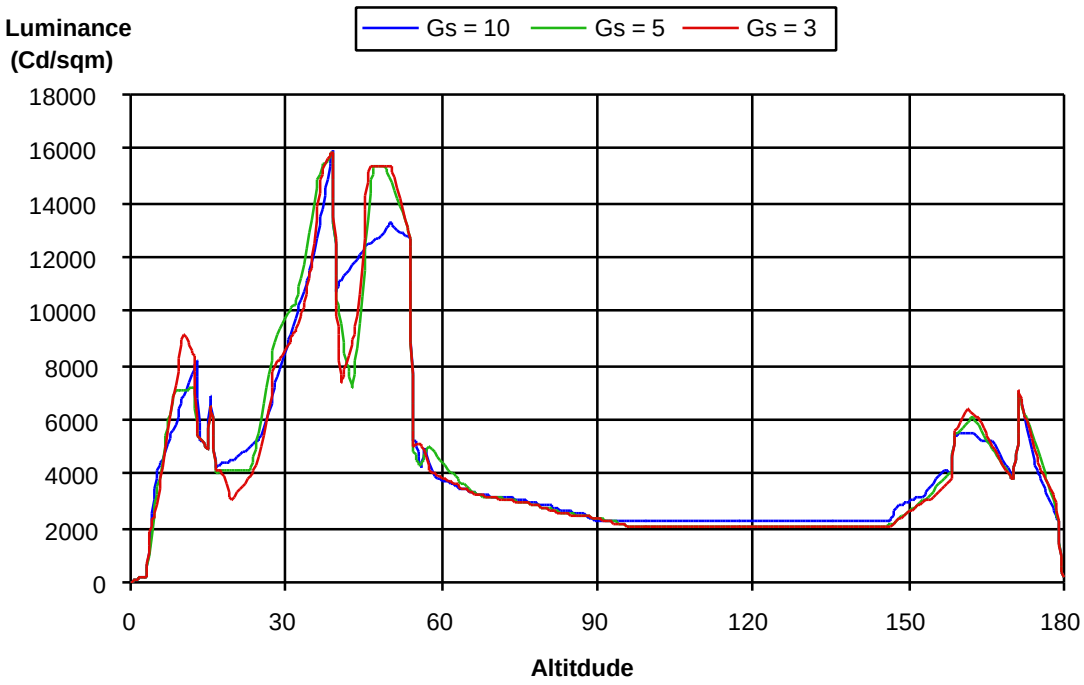
Figure 7.1: Example SDF models on difference grid sizes (each with 8 contour intervals)

Figure 7.2 shows some experimental results. Figure 7.2(a) is for a partially overcast sky (the same as in Figure 7.1), Figure 7.2(b) is for a totally clear sky and Figure 7.2(c) is for a fully overcast sky. The plots show the luminance levels recovered from the SDF models (in Cd/sqm) for varying grid size (Gs) values, 3°, 5° and 10° respectively.

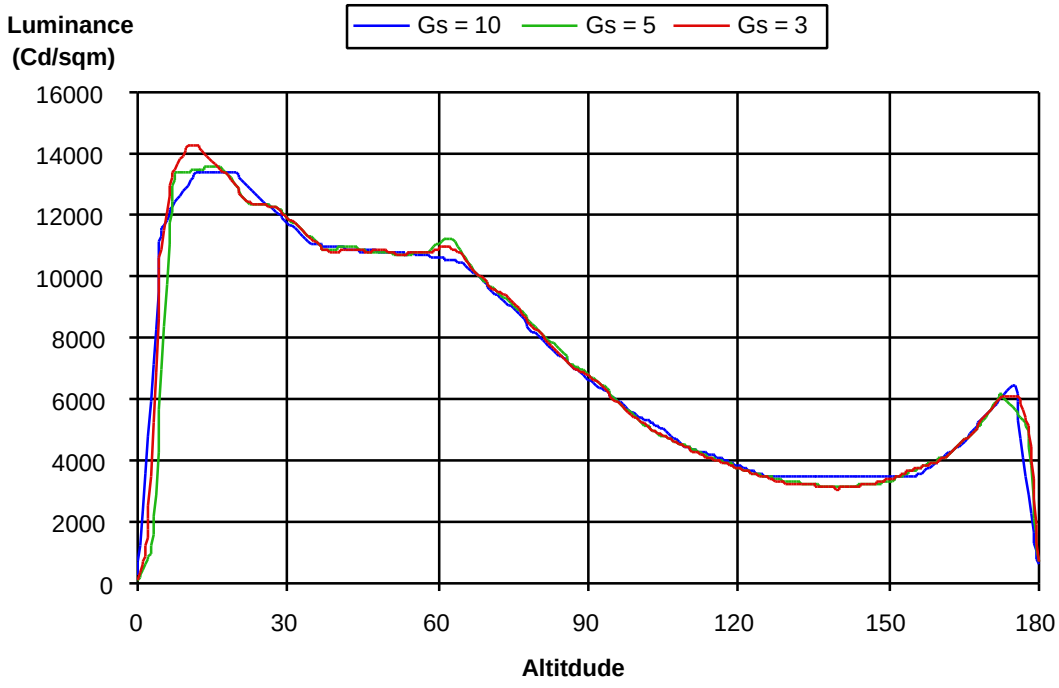
In each case the same image is used and shows the plot of the computed luminance along a vertical plane through the sky dome at an azimuth of 0° for altitudes from 0° to 180°.

If we take the finer (3°) grid as being the most accurate, then we see that the 5° grid is not too different in most cases while the 10° grid shows up some differences, especially in regions where the luminance values are varying rapidly in the partially cloudy sky in (a). These results demonstrate pretty well what was expected. We might therefore conclude that a grid size of 5°, or less, will give reasonable results in most cases.

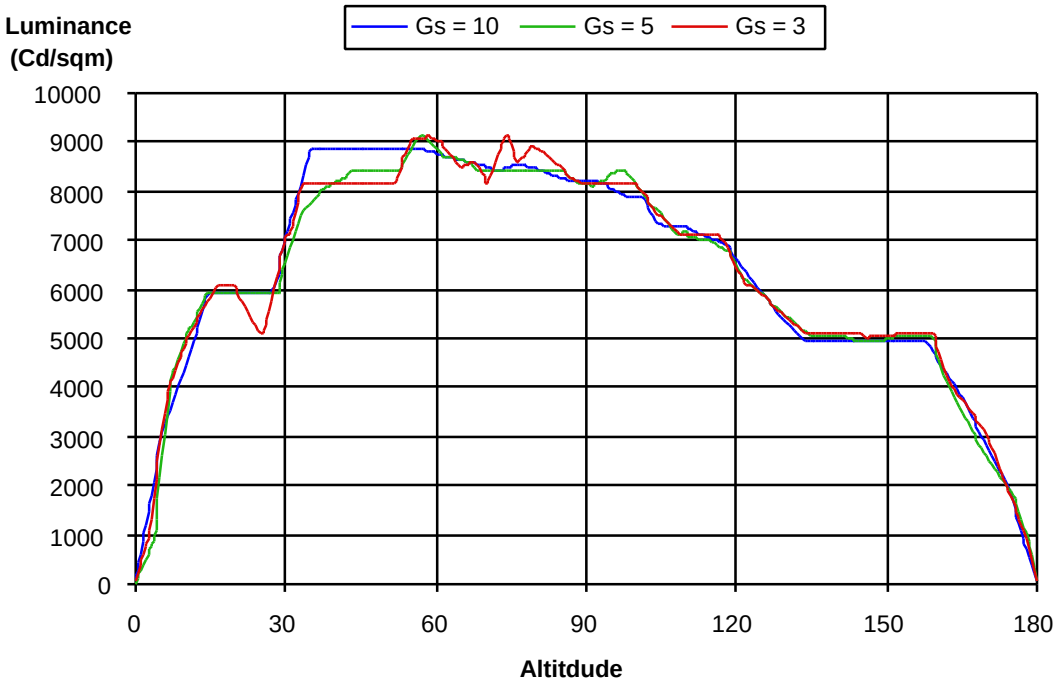
The interpolation scheme used in the SDF model is linear across the flat patches created from the triangulation generated from the iso-luminance. This explains the "flat" sections of the graphs.



(a) Partially Overcast Sky (Azimuth = 0°)



(b) Fully Clear Sky (Azimuth = 0°)



(c) Fully Overcast Sky (Azimuth = 0°)

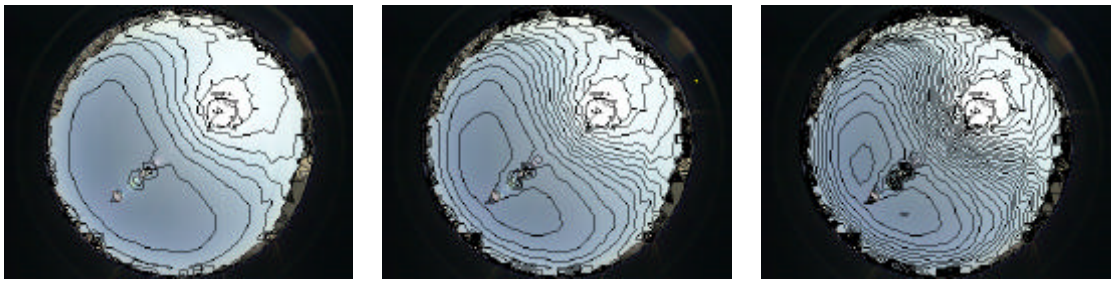
Figure 7.2: Comparing Computed Luminances for varying grid spacing

7.2.2 Choosing the Contour Intervals

The other major influence in building the SDF models is the choice of contour intervals. We can expect that as the number of contours increases (for a given range of luminance values) we will get better results. The choice of contour intervals is, in fact, a complex task as we have two conflicting objectives, viz:

- We would like to have more contours in regions of rapid change so that the interpolation will have smaller errors.
- We may want to have more closely spaced contours in regions of the sky that have a small range of values if we need to retain the more subtle variations in luminance.

Figure 7.3 shows three models with increasing numbers of equally spaced contours:



(a) SDF model with 8 contours

(b) SDF model with 16 contours

(c) SDF model with 32 Contours

Figure 7.3: Effect of varying number of contours in SDF model

Here we see that where the luminance gradients are greatest, the contours are more closely spaced (across the sky dome). As a result we have a significant part of the sky, especially in Figure 7.3(a), where there are few contours. This can be avoided by using more contours, but at the expense of increasing the complexity of the model and having, perhaps, too many contours in other parts of the model.

While "equally" space contours are useful for visualising luminance distributions (like those shown in Figure 7.3), there is nothing in the SDF framework that requires equally spaced contours.

In another context (representing models of bi-directional transmittance of light through prismatic glazing materials) we have found that we can have quite large variations of data values to deal with. For example, perhaps more than 95% of the bi-directional response has quite low levels of transmittance, and 5% having transmittance values several hundred times greater. In this case we need to be more "cunning" in choosing a contouring strategy. Luminance distributions for skies, however, tend to be relatively more "uniform" as seen in Figure 7.2.

The SDF framework offers two contouring strategies, viz:

- Equally spaced contours: where a the number of contours is chosen and the range of luminance values is divided equally by this number to give the contour levels. This strategy results in more contours being located in regions of rapid change. This can be seen in Figure 7.3.
- Adaptive contour intervals: where the user indicates a minimum number of intervals, and additional (intermediate) intervals are added if the distribution of data points across the set of intervals is sufficiently distorted. So, for example, if one interval has many more data points than the others, that interval is subdivided into two intervals. This process is repeated until an acceptable distribution is achieved. This strategy thus tries to match the interval to the distribution of data, see Roy [1998].

From experience with sky luminance models, it is suggested that this second option is not of great interest as the distribution of data points is typically uniform enough not to require the adaptive process. Figure 7.4 shows some comparative results for a partly cloudy sky.

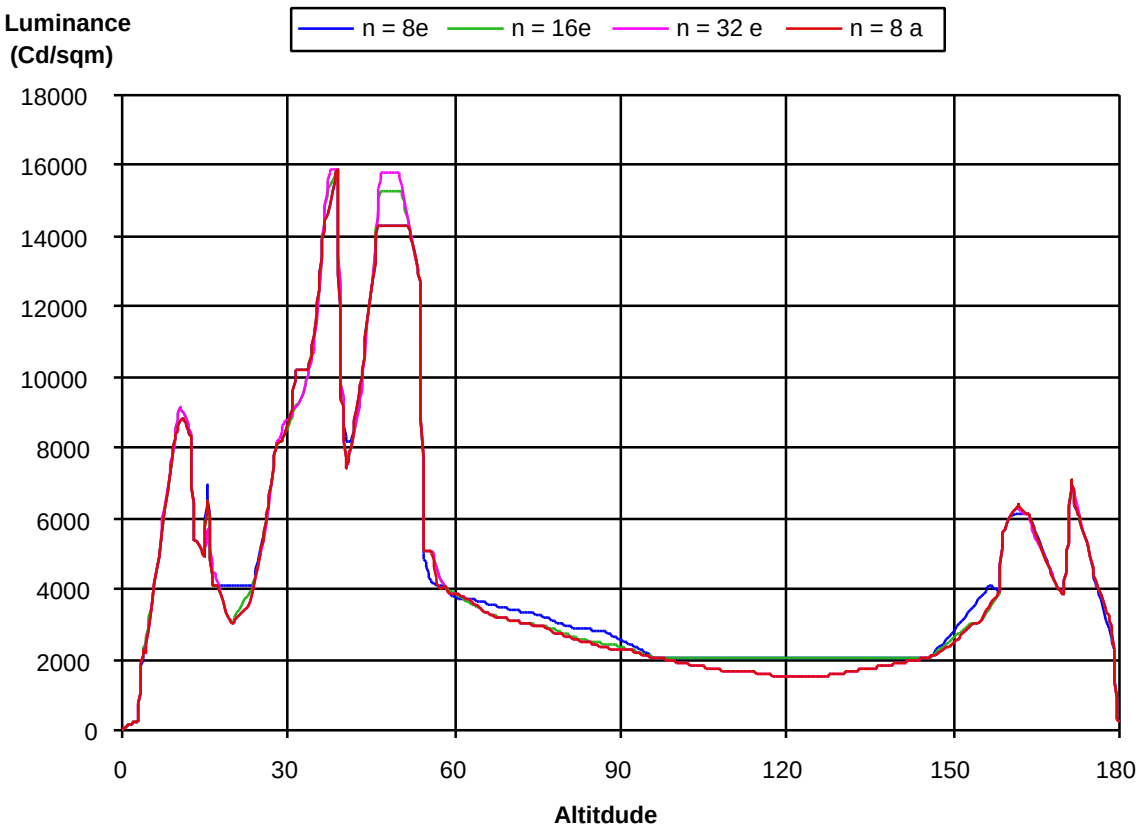


Figure 7.4: Effect of different contouring strategies

In Figure 7.4 we have equally spaced contour models (8e, 16e and 32e), with increasing numbers of contours (8, 16 and 32 respectively) . We can see some differences. The model with more contours is better able to represent the peak values, but there is not a lot of improvement. For the adaptive model (8a), the actual number of contours in the model is 15, after specifying that no contour interval should have more the 150% of the average number of data points per interval. As we expect in regions of low gradient we have been able to add more detail. Otherwise, the 8a model follows the 16e and 32e models quite closely.

For these results, and from other experiments we have done, it is fairly clear that 16 equally spaced contours will usually give acceptable results for most sky models.

7.2.3 Size of models

The final concern with the SDF models is their size, i.e. how much disk space they will occupy. If we were to fully implement an on-line sky monitoring system, then we will need to be able to store the SDF models as a permanent record of the digital images. Firstly, we know that the size of the SDF models vary (approximately) linearly with the number of contours and also linearly (or less) with the number of grid points. We can thus alter one or the other without running into quadratic complexity problems. Table 7.2 gives some typical results (for the partly overcast sky).

Grid Size	Number of Contours	Model Size (kbytes)
3	16	56
5	16	40
10	16	36
3	8	32
3	16	56
3	32	100

Table 7.2: SDF model sizes for varying grid sizes and numbers of contours

Given that the original images are approx 3.2 Mbytes (1000x1000) or 800 kbytes for a 500x500 image, the effective compression is significant. We naturally loose colour information, but retain luminance information with considerable detail. If we take a reasonable model specification (5° grid and 16 contours) then a full day's data (of images every 10 minutes for 10 hours, for example) will require about 30 Mbytes of disk storage.

8 Operational Modelling

8.1 Introduction

The final outcome of this project was to demonstrate an operational prototype of a real-time image processing system that would enable the capture and processing of sky images at relatively frequent intervals (say every 10 minutes). These images would be processed to a form that will retain all the essential luminance information that would enable the sky models to be used in third party simulation packages.

We have not been able to field test this concept due to limited resources provided for the project, though the individual components are operational (as shown in the previous sections of this report). In this section we will provide an overview of the operational context of the data collection and modelling system.

8.2 Image Analysis Process

We have described the image analysis processing steps in detail in previous sections of this report. The complete process is portrayed in Figure 8.1.

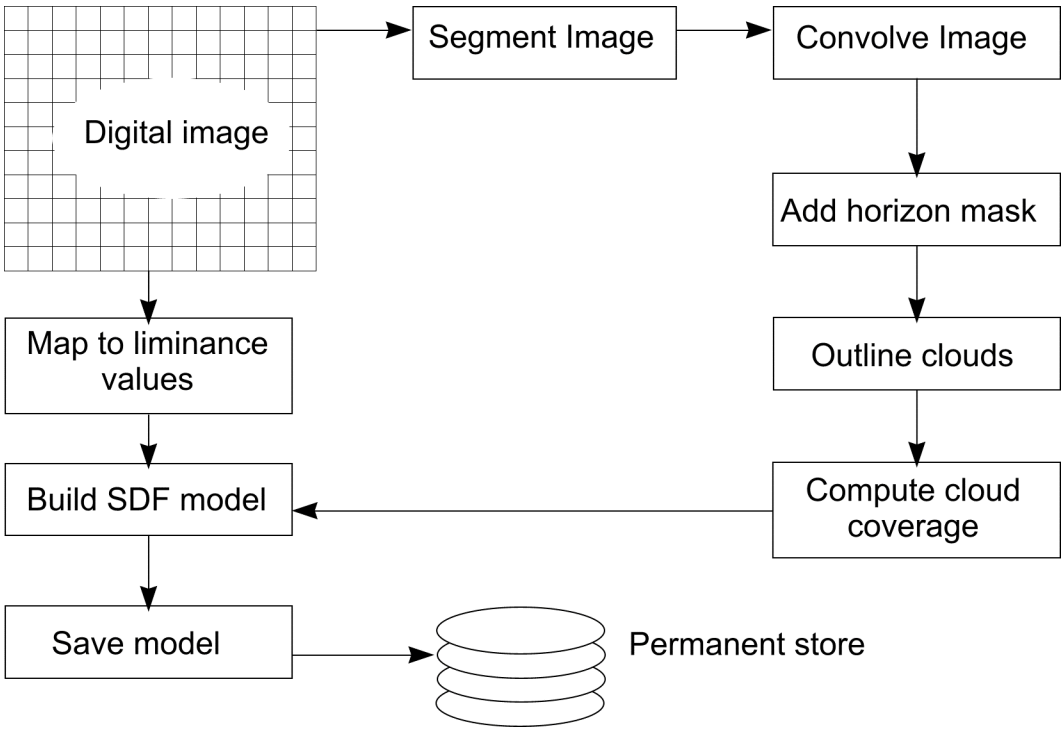


Figure 8.1: The Image Processing Sequence

The image processing requires two parallel phases. The computation of cloud outlines, and the building of the SDF model. The former follows the segmentation of the image into cloud and sky regions, then cleaning up the image and finally computing the cloud outlines. The luminance map and the cloud outlines are used to build the SDF model and to compute the cloud coverage. If we ignore the potential discontinuities caused by cloud (and other horizon objects) then the SDF model can be build from just the luminance map. We can also include in the SDF model the

edge information from the cloud edges. The SDF interpolation process can be better tuned to the luminance distribution if the edge information is included.

The SDF modelling framework allows users to define their own parameters which can be stored in, and retrieved from, SDF models. It is therefore possible to include in the SDF model a range of data values that might be related to each sky model and the image from which is was derived. One obvious parameter is the cloud coverage which might be stored as a percentage coverage (of the sky dome). Other parameters that might be included could be:

- Cloud coverage in different segments of the sky dome.
- Average luminance of the sky dome
- Luminance on the horizontal plane, or any other plane
- Image exposure parameters
- Other local conditions that can be recorded from other measuring devices, eg. radiation sensors.

8.3 Hardware Configuration

The proposed hardware set up is shown in Figure 8.2. Images are captured by the CCD in the digital camera which is equipped with an equi-angular projection fish-eye lens. The camera is controlled by the computer which will trigger the camera at the required intervals, and down load the bitmap image from the camera.

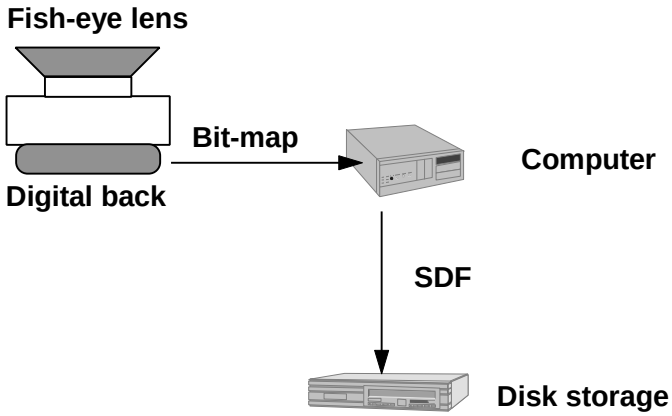


Figure 8.2: Hardware Configuration

Each image is processed as it is received, and the converted into an SDF model. This model is saved onto the hard disk. The full digital image is not retained. The SDF models can be retrieved from the controlling computer at appropriate intervals.

The time taken to process each image is typically much less than the 10 minute interval. A 500x500 image can be fully processed on a 180 Mhz machine in less than 2 minutes. The approximate break down of the processing phases are shown in Table 8.1.

Phase	Approx Time (secs)
Segmentation	9
Convolution	42
Outlining Clouds	17
Building SDF	10

Table 8.1: Approximate processing times

The processing times will naturally depend in the grid size and the number of contours – but not too greatly. The size of the image is more important as most of the processing time is taken in the convolution phase which will increase with the square of the image size.

Given that faster computers are becoming available all the time these times can only get less. It is therefore judged that the image processing time is not likely to be a problem, allowing images to be processed at quite close intervals.

9 References

Berrutto V and Fontoynont, M. [1995], "Applications of CCD cameras to Lighting research: Review and Extensions to the measurement of Glare Indices", Proceedings, 23rd Session CIE, New Delhi, pp192-195.

Commission Internationale de L'Éclairage [1986], "Colorimetry", 2nd edition, Publication No 15.2, CIE, Vienna.

Davis, G. B., Griggs D. J. and Sullivan G. D. [1992], "Automatic Estimation of Cloud Amount Using Computer Vision", Journal of Atmospheric and Oceanic Technology, Vol 9, Feb, pp81-85.

Foley, J. D. and Van Dam, A, [1983], "Fundamentals of Interactive Computer Graphics", Addison-Wesley, p615.

International Television Union (ITU) [1990], "Basic parameter values for the HDTV standard for the studio and for international programme exchange", ITU-R recommendation BT709, ITU, Geneva.

Kaufmann, J (ed), [1987], "IES Lighting Handbook – Applications", IESNA, New York, 11.24

Nakamura H and Oki M, [1975], "Measurement of luminance distribution under various sky conditions by orthographic projection camera", Proceedings CIE Conference, London, 493 - 502.

Rainer, K, [1998], "MacCurveFit", Kevin Rainer Software, MtWaverley.

Reid G. [1997], "Interpolation and Contouring of Discontinuous Discrete Data", MSc Thesis, Department of Computer Science, The University of Western Australia.

Robins C. L., Hunter K. C. and Cannon T. [1984], "Mapping sky and surface luminance distribution using a flux mapper", Energy and Buildings, 6, 247-252.

Roy G. G., Ruck N., Reid G., Winkelmann F. and Julian, W [1995], "The Development of Modelling Strategies for Whole Sky Spectrums Under Real Conditions for International Use", ARC Project A989131897 Report, The University of Sydney and Murdoch University, Sept.

Roy, G. G. [1998], "SDF User Guide", School of Engineering, Murdoch University.

Roy, G. G., Rodrigo M. and Wong K. K. [1989], "A Note on Solar Declination and the Equation of Time", Architectural Science Review, Vol 32, pp43-51

Roy, G. G., Ruck, N. and Winkelmann, F. [1995], "New Modelling Technologies for Sky Luminance", ", Proceedings, 23rd Session CIE, New Delhi, pp327-328

Roy, G. G., Ruck, N., Reid, G. and Julian, W [1995], "Sky Luminance: Standard digital form for modelling", Lighting Research Technology, Vol 27, No 3, pp161-167.

Ruck. N. Roy, G. G. and Reid, G.. [1993], "Modelling the Sky – A Standard Digital Form", proceedings, 3rd International Conference, International Buildings Performance Association, Adelaide, 16th-18th Aug, pp525-531.

Shields, J. E., Koehler, T. L., Karr M. E. and Johnson R. W. [1990], "Automated Clod Cover & Visibility Systems for Real Time Applications", Optical Systems Group, Marine Physical Laboratory, San Diego, CA, Technical Note No. 217.

Shields, J. [1998], "Whole Sky Imager Theoretical Discussion", http://www-mpl.ucsd.edu/people/jshields/index.html, Marine Physical Laboratory, Scripps Institution of Oceanography, University of California, San Diego.

Smith W. L. Jr., Garber, D. P., Ayers J. K. and Doelling D. R. [1995], "Cloud Properties Deribed from GOES-& for Spring 1994 ARM Intensive Observing Period Using Version 1.0.0 of ARM Satellite Data Analysis Program", NASA Reference Publication 1366, Langley Research Center, Hampton, Virginia.

Tregenza, P. R, [1987], "Subdivision of the sky hemisphere for luminance measurements", Lighting Research & Technology, 19, 1, 13-14.

10 DSM User Manual

DSM (Digital Sky Modeller) is a software package implemented as a part of this project. It was developed as a means of testing and demonstrating the concepts embodied in this report. It could form the basis for a production system for the real-time collection and analysis of skies.

The core components of DSM are the SDF modelling framework which facilitate the underlying computational capabilities to represent a sky and to allow it to be manipulated in various ways. Full details of SDF can be found elsewhere (Roy et al [1995] and Roy [1998]). The core SDF libraries are written in the C language and are readily adapted to most hardware platforms (e.g. Unix, Windows, Macintosh). DSM is also written in C, but is currently only available for the Macintosh OS.

DSM has been developed to be a test-bed for concept testing and so provides a wide range of options, and parameter settings, to the user. These can be used to test out various image processing concepts, as well as calibrations from external sources (CCD calibration, neural network coefficients, etc). It could form the basis for the development of a production system which would facilitate the on-line capture, processing and storage of sky images on a continuous basis.

DSM - Digital Sky Modeller

Version 1.2

User Guide

10.1 Overview

DSM provides a tool box for the analysis of digital images of the sky. The purpose of this analysis is to be able to estimate both cloud cover information and luminance distributions. The digital image is converted into Standard Digital Form (SDF). SDF provides an efficient mechanism for representing and storage of complex luminance distributions over the sky dome. The luminance map is represented by a map of iso-luminance contours.

In its current form DSM provides a set of tools for each stage of the image analysis. These can be applied manually by the user. It is intended that for the complete analysis of an image, from digital image to saved SDF model will be done automatically by stringing together a sequence of the basic operations.

DSM assumes that the digital image is available as a PICT formatted file taken with a fish-eye lens which as an equi-angular projection onto the image plane. The resulting circular image of the sky is centrally located in the rectangular image area.

10.2 Menus

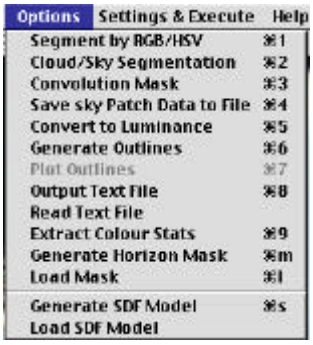
10.2.1 The File Menu

New	Creates a new image widow (not used in DSM)
Open	Opens an image file in PICT format
Close	Closes the currently active image window
Save	Save currently active image as a PICT file
Save As...	Save the currently active image as a new PICT file
Page Setup	Configures printer
Print Selection	Prints current selection
Print	Prints whole image
Quit	Quit DSM

10.2.2 The Edit Menu

Undo	Not implemented
Cut	Cuts out the currently selection (careful! not tested)
Copy	Copies current selection to clipboard
Paste	Not operating
Paste Special	Not operating
Clear	Not operating
Select All	Not operating
Crop to Selection	Crops the current image down to the selection
Show ColourMap	Displays current colour map (not operating)

10.2.3 The Options Menu



Segment by RGB/HSV	Test only - not in use
Cloud/Sky Segmentation	Perform segmentation of image into cloud and sky (and other components, including sun and horizon obstructions) according to currently selected method (see Analysis Options dialog)
Convolution Mask	Convolve image to tidy up cloud edges and image noise
Save Cloud Patch to File	Test only - not in use
Convert to Luminance	Converts the current image to a monochrome luminance map according to the range set in the Analysis Options dialog.
Generate Outline	Takes a convolved image and generates cloud edges
Plot Outlines	Draws the could edges on the current image
Extract Colour Stats	Extracts a table of colour statistics from the current image. The result is saved as a file containing a frequency distribution (100 bins) for the occurrence of R, G, B, H, S, V values for the image (HSV values are computed from RGB), or

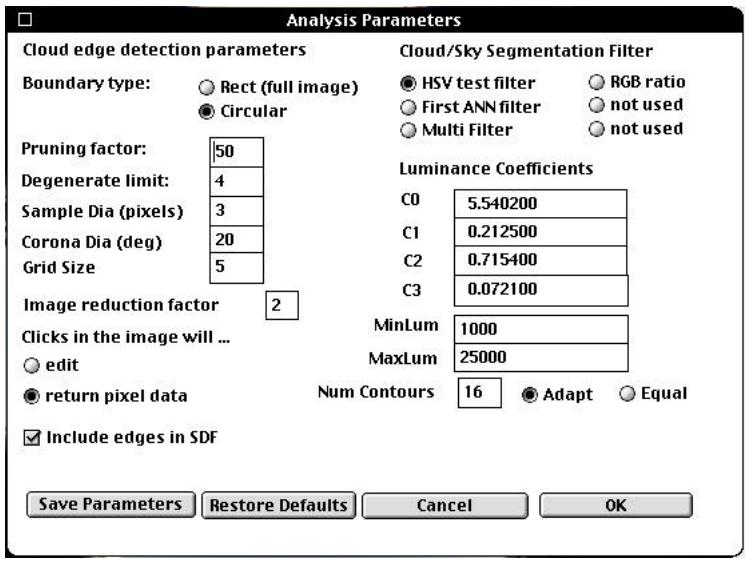
	Saves a file of RGB and HSV values for the sample points resulting from a series of "sample point" operations. (depends on release configuration)
Generate Horizon Mask	For an image processed to identify horizon obstructions, convert to a black/white image where the horizon obstruction appears in white.
Load Mask	Loads a mask into the currently image window, converting mask area to black (ie sky).
Generate SDF Model	Creates, and saves, and SDF model for the current image. Contour map is plotted.
Load SDF Model	Loads an SDF model and plots contour map into current image.

10.2.4 The Settings & Execute Menu

Analysis Options	Brings up Analysis Options dialog
Location and Time Data	Brings up Location and Time Data dialog
Neural Net Params	Brings up the dialog to insert/edit the neural network weighting coefficients
>>>>> **SDF run** <<<<<	Initiates a full image analysis from original PICT image to final SDF model
Save Parameters	Saves the current parameter settings into the configuration file
Restore Parameters	Restores the previously saved parameter settings

10.3 Dialogs

10.3.1 Options Dialog

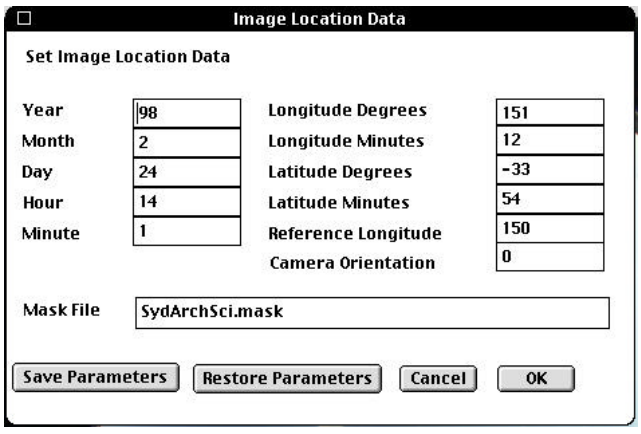


This dialog allows a range of image processing option to be set, saved as the default and restored as necessary.

Boundary Type	If "Rect" is selected whole rectangular image will be processed. If "Circular" is selected a circular region in the centre of the image will be processed.
Pruning Factor	The pruning factor is used in cloud edge detection. Cloud edges are smoothed to the point that the error, measured as the area of the image (in tenths of a pixel area), which is included, or excluded, from the polygon shape by making a straight line segment between two points along the edge.
Degenerate Limit	The smallest number of sides on a edge polygon (typically 3 or 4) before the polygon is eliminated.
Sample Dia	The diameter (in pixels) of the image sample region when sampling data from an image.
Corona Dia	The assumed diameter of the corona (in degrees)
Grid Size	The size of the triangulation grid used to sample the digital image, measured in degrees at the horizon.
Image reduction factor	This factor is used to load images at a size different from the image files. Values are typically 1, 2 or 4. If the original image is 1000x1280 pixels, a factor of 4 will produce working images of 250x320.
Click in the image will.	The mouse operates in two modes inside the current image: In Edit mode, a click and drag can define a region which can be copied, clipped etc. In Return Pixel Data mode, a click will report the colour and luminance values at the clicked point and the average over a small circular region (the size of which is set by the Sample Dia value). In this latter mode the data samples are retained and can be saved to a text file using the *Extract Colour Stats* option on the *Options* menu (release specific).
Include edges in SDF	Checking this option will cause the edges generated in the image from cloud analysis to be included in the SDF model. If these edges are not included, the SDF model will have only one edge – the edge defining the horizon boundary.
Cloud/Sky Segmentation Filter	There are currently four operational segmentation models: (a) HSV Filter - uses a filter based on HSV values.

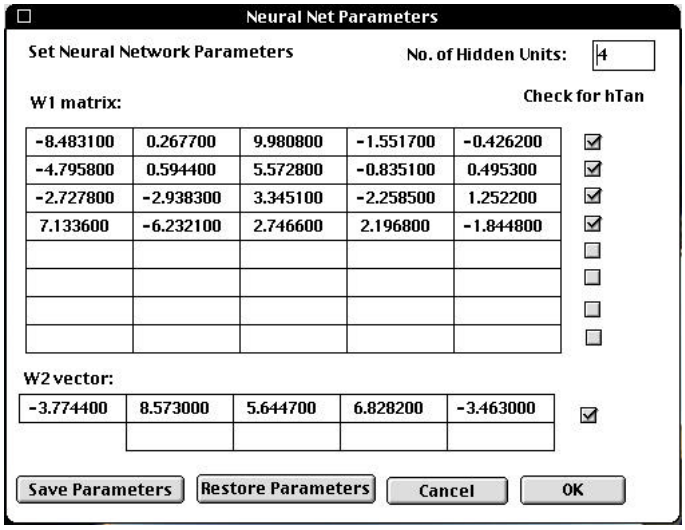
	(b) ANN Filter - based on a neural network . (c) Multi Filter - intended to be used to isolate horizon obstructions and to build an horizon mask. Not to be used to generate SDF model directly. (d) RGB ratio filter – based on the red/blue ratio.
Luminance Coefficients	These coefficients convert RGB values (0-255) to luminance (cd/sqm. C0 is the ratio $T/f^2 \times 10^6$ for the image (T is exposure time in seconds, and f the aperture). The other coefficients, (C1, C2 and C3) are applied to the red, green and blue components of the pixel values.
MinLum	Used for the Luminance map option to set the black luminance level
MaxLum	Used for the Luminance map option to set the white luminance level.
Num Contours	The number of contour intervals inserted into the SDF model (generally in the range 16 to 32).
Adapt	Sets the contouring mode to adaptive.
Equal	Chooses equally spaced contours.
Save Parameters	Saves the currently set parameters (including those from the Location & Time Data dialog) in the DSM configuration file (DSM.cfg).
Restore Defaults	Restores the defaults as stored in the configuration file.

10.3.2 Location & Time Data Dialog



This dialog allows the image time and location parameters to be pre-set. If the camera is not pointed accurately to true north (ie the top of the camera points north so that east is on the left of the image and west on the right) then this field can be set to add a correction (degrees). The mask file name will be used to load the local mask file when needed.

10.3.3 Neural Network Parameters Dialog



This dialog allows the user to set the neural network weighting coefficients. The number of hidden nodes can vary from 1 to 8. There are 4 input nodes (red, green and blue values), and the camera exposure as the ratio T/f^2 (x 10^6).

The coefficients are generated elsewhere (using a ANN simulator). Either linear or tanh node response functions can be used.

The W1 matrix is for the weights from the input nodes to the hidden layer in the order as described in Figure 1.5. The W2 vector is for the weights from the hidden layer to the single output node.

10.4 Operations

There is an implied sequence of operations to process an image. All processing steps operate on the *current* image. The current image is the one in the active (top most) window. Any required data for a processing step comes from this current image. In most cases the processing step creates a new image (leaving the original untouched). Upon launching DSM, the menu bar appears together with a console widow where various progress reports are displayed (and some debugging information).
The basic steps in image processing are as follows:

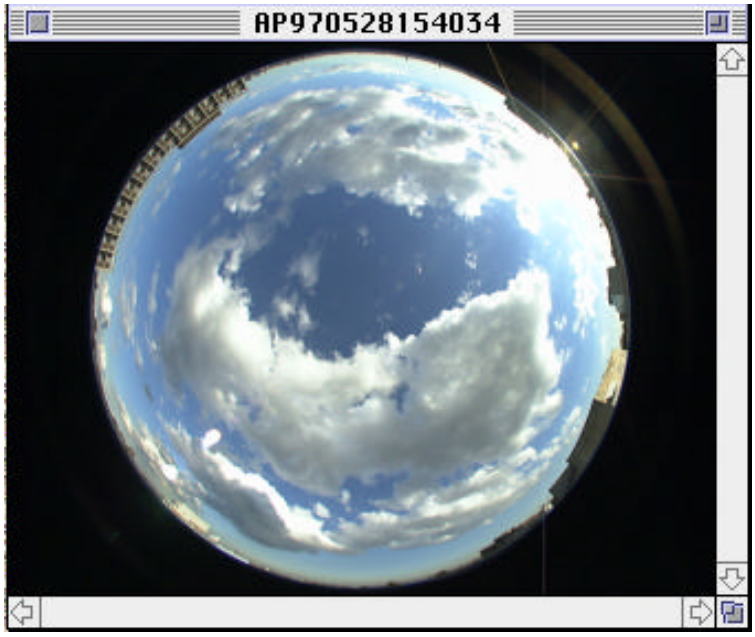
1. Load image from PICT file
2. Set time and location parameters in Location & Time Data dialog (this may be set as the default is the same image is being processed several times).
3. Set the required options in the Analysis Options dialog (generally, Circular image, HSV segmentation, and others as required)
4. Choose Cloud/Sky Segmentation - to segment sky into cloud and sky parts
5. Choose Convolution - clean up image and remove noise
6. Loads horizon mask (location specific).
7. Choose Generate Outlines - to compute cloud edges
8. Choose Generate SDF Model - to build SDF model, compute cloud coverage (as a % of the whole sky) and to save the SDF model into a file.

The steps 4 to 8 can also be done automatically in sequence from the **SDF run** option on the **Settings & Execute** menu.

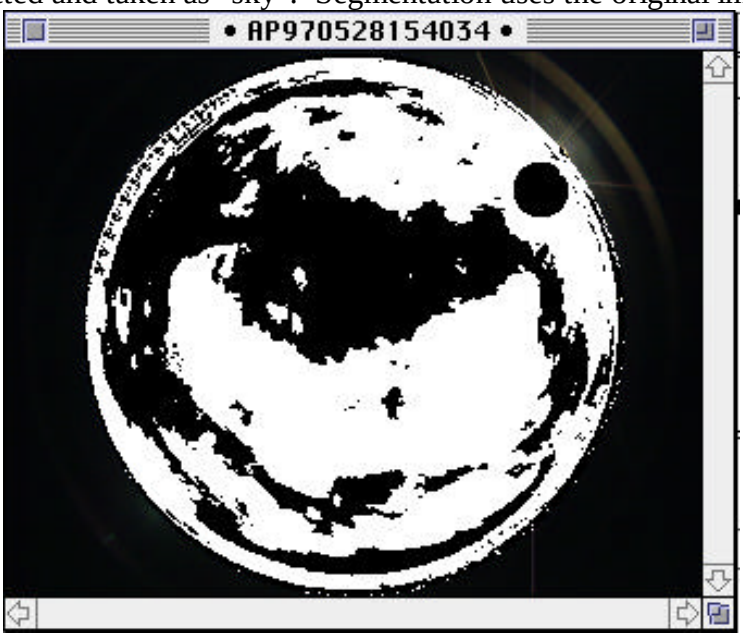
The images from the Nikon E2 camera are 1000x1280 pixels. The processing of images of this size takes some time (and lots of RAM), so it is best to reduce the size to at least half, and perhaps a quarter size. Images of 500x640 contain plenty of detail for most purposes.

The following sequence of images show this process at work:

The original image (quarter size: 320x250 pixels)

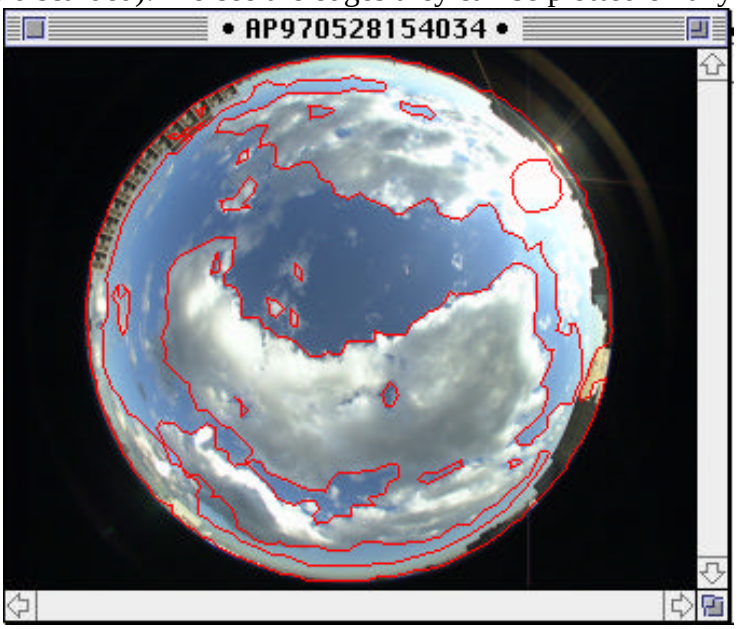


The segmented image, white is cloud. Note that the segmentation shows that the sun has been detected and taken as “sky”. Segmentation uses the original image as input.

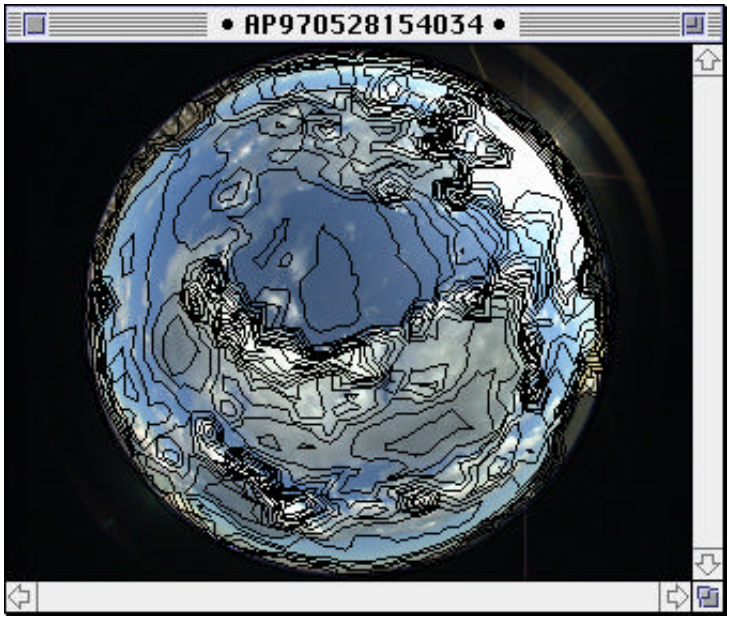


The convolved image showing edge smoothing and removing of much of the noise. Convolution uses the segmented image as input.

This image shows the cloud edges plotted on the original image. Cloud edge detection used the convolved image as input (the output in this case is a flooded image which is discarded). To see the edges they can be plotted on any image.



The SDF model has been computed (and saved) and the contours plotted. The SDF model requires the original image (to get at the luminance data) as input.


• AP970528154034 •