Cloud Cameras at the Pierre Auger Observatory

BY

Michael G. Winnick, BSc (Hons)

A dissertation submitted

In partial fulfillment of the requirements for the degree of

Doctor of Philosophy

University of Adelaide

School of Chemistry and Physics

June 2010

Chapter 4

Image processing and cloud detection

The Auger infrared cloud cameras provide images of the night sky but do not automatically decide which parts of these images are actually cloud. A survey of literature reveals much in the way of cloud detection algorithms, but most are only applicable to satellite-based measurements, or require measurements of the the infrared flux at multiple wavelengths. In this chapter different image processing techniques are discussed, developed, applied and evaluated with respect to images recorded by the Auger infrared cloud cameras.

There are several image processing techniques that are capable of detecting clouds within cloud camera images. Firstly is the method of edge detection, whereby different intensity gradients within an image are used to segment the sky into different regions. And secondly is the thresholding approach, whereby some intensity threshold is chosen for an image, above which any pixels are considered to be cloud and below, considered to be clear sky.

In order to facilitate manual processing of the large quantity of images recorded by the cloud cameras, a GUI program called PACMan (Pierre Auger Cloud Manager) has been created. Images that are processed are then converted and compressed into a format suitable for the cloud camera database and subsequent analysis.

4.1 Edge detection

Edge detection is a process by which localized sharp changes in intensity within an image are identified. As clouds are brighter in the infrared than clear night sky, there is usually a strong contrast in intensity between clouds and normal clear sky. Thus, edge detection algorithms provide an opportunity to identify clouds within infrared images on cloudy nights. The lack of distinguishing gradients on completely overcast nights makes it sometimes unreliable however.

There are a wide variety of edge detection algorithms currently available - as they have multiple applications within machine learning, medicine, industry and image processing. Generally an algorithm involves some form of pre-processing of a particular image - filtering out noise, or enhancing edges in an effort to improve the efficiency of the algorithm. Then once edges are located, they may be used to segment the image into different regions - in the case of the cloud camera images, cloudy and clear night sky regions.

4.1.1 The Canny edge filter

The primary algorithm investigated in this study is the Canny edge detection algorithm [47] due to its multiple variables and subsequent adaptability to different situations. This algorithm sets out to optimize the detection of a step edge obscured by Gaussian noise. In his creation of this algorithm Canny wanted to fufill three goals: good detection, good localization and one response to one edge. That is, he wanted a detector with a low probability of falsely identifying or missing an edge, and that only has a single response to a single edge that is located close to the real location of the edge. After formulating these criteria mathematically he suggested that the optimal filtering function may be approximated through a Gaussian filter.

A Gaussian filter is a weighted averaging filter, whereby the signal from the pixels surrounding a particular pixel j are weighted and averaged to provide a new signal for pixel j. The weights W are calculated using a Gaussian function such as that seen in equation 4.1:

$$W(x,\sigma) = \frac{exp(-x^2/(2\sigma^2))}{\sqrt{2\pi\sigma^2}} \tag{4.1}$$

where x is the distance between the pixel being weighted and pixel j, and σ is the standard deviation of the Gaussian function.

Filtering an image can help suppress the detection of any edges that are the result of fluctuation in the intensity of an object (noise), as opposed to the boundary between two objects. Increasing the σ value will reduce sensitivity to both noise and actual edges within an image, and may decrease the accuracy of edge localization. To improve processing time the filter is usually applied as a convolution mask (much smaller than the image). An example of the application of a Gaussian filter may be seen in figure 4.1, where much of the speckly noise is removed.

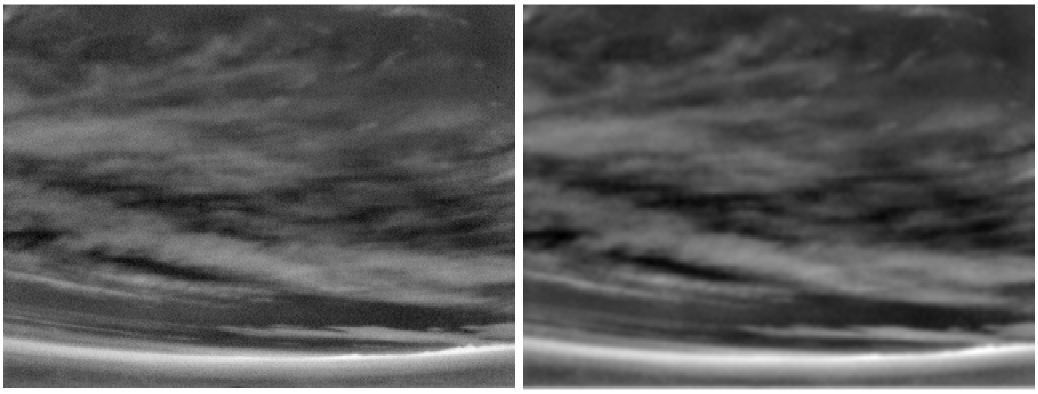


Figure 4.1: *The result of a Gaussian filter applied to a cloud camera image with $\sigma = 1.0$. **[Left]** Original image. **[Right]** Gaussian filtered image.*

After application of a Gaussian filter the next stage of the Canny edge detection algorithm is to calculate the edge strength and direction within the image. One way of doing this is through the application of a Sobel filter [168]. The Sobel filter consists of a pair of convolution masks, one G_x estimating the gradient horizontally across the image, and the other G_y, vertically. They take the form:

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

e.g for a pixel at row j and column i with intensity $S_{(j,i)}$:

$$\begin{aligned} G_x = &\ 2(S_{(j,i+1)} - S_{(j,i-1)}) + (S_{(j-1,i+1)} - S_{(j-1,i-1)}) + (S_{(j+1,i+1)} - S_{(j+1,i-1)}) \\ G_y = &\ 2(S_{(j+1,i)} - S_{(j-1,i)}) + (S_{(j+1,i-1)} - S_{(j-1,i-1)}) + (S_{(j+1,i+1)} - S_{(j-1,i+1)}) \end{aligned}$$

The two parameters G_x and G_y may be combined to estimate the overall magnitude G and direction of the gradient θ (from the vertical), as seen in equations 4.2 and 4.3:

$$G^2 = G_x^2 + G_y^2 \tag{4.2}$$

$$\theta = \begin{cases} \tan^{-1}(G_x/G_y) & G_y \neq 0 \\ 90.0^\circ & G_y = 0 \end{cases} \tag{4.3}$$

The angle θ is generalized to one of four different directions: horizontally, vertically or either of the two diagonal directions across the image. For example an angle $\theta = 3^\circ$ or 86° would be generalized to the vertical and horizontal directions respectively. Other operators exist that may calculate a superior measure of horizontal and vertical gradients, such as the Scharr filter [161], which suggests that the -1, 1, -2 and 2 within G_x and G_y be replaced by -3, 3, -10 and 10 respectively.

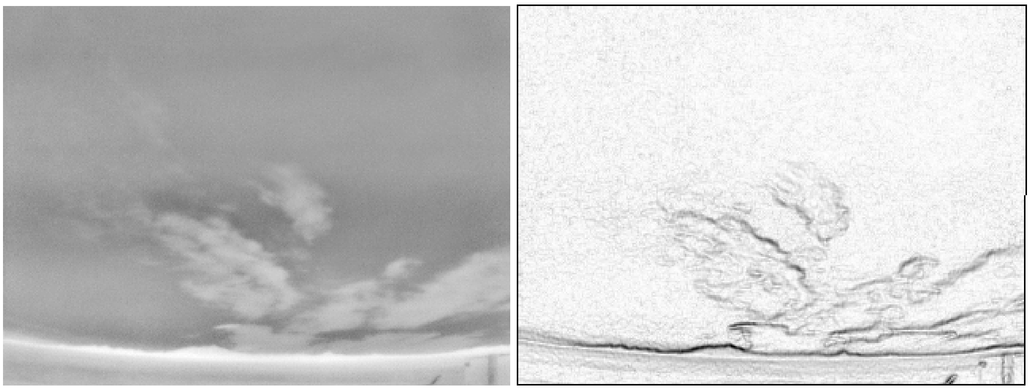


Figure 4.2: *Images demonstrating effect of the Sobel filter. Note that a Gaussian filter has not been applied in this example. [**Left**] Original image. [**Right**] Sobel gradient image.*

An example of the Sobel filter in action may be seen in figure 4.2, where the Sobel gradient G from equation 4.2 has been calculated for all but the outermost boundary of pixels (since the convolution mask is 3x3). Darker regions of the right-most image correspond to regions of strong gradient in the left-most image and so indicate the local presence of edges. Note the presence of edge gradients even in the absence of cloud in the upper half of the image. This is due to the noise inherent within the image that will be suppressed through application of a Gaussian filter. The regions of strongest (darker) gradient in the rightmost image correspond with the edges of the Earth and cloud within the left-most image.

The next stage of Canny's edge detection algorithm is to locate all of the edges within the image using non-maximum suppression. Non-maximum suppression is a process that locates pixels that may be potentially identified as being edges. Gradients are expected to be at their strongest closest to the position of edges within an image. For each pixel i with a gradient magnitude and direction associated with it, the gradient magnitude of pixels to either side of it along the direction of the gradient at i are considered. If the neighbouring gradient magnitudes are both less than the gradient of pixel i, then pixel i is identified as being a potential edge. This results in the identification of potential edge 'ridges' with the width of a single pixel within the image.

Two thresholds (the upper T_{upper}, and the lower T_{lower}) are then used to identify those of the identified edges that are 'true' edges, and those that are the result of noise within the image. If the gradient magnitude of a particular pixel (that survived non-maxima suppression) is above T_{upper} then it is marked as an edge. Due to the presence of noise, the magnitude of the gradient of an edge in an image may vary in different parts of an image. To identify those edges which happen to fall under the upper threshold T_{upper}, a secondary lower threshold T_{lower} is used which marks any pixels with a gradient above T_{lower} as an edge if it borders a pixel that has already been identified as an edge (and survived the non-maxima suppression stage).

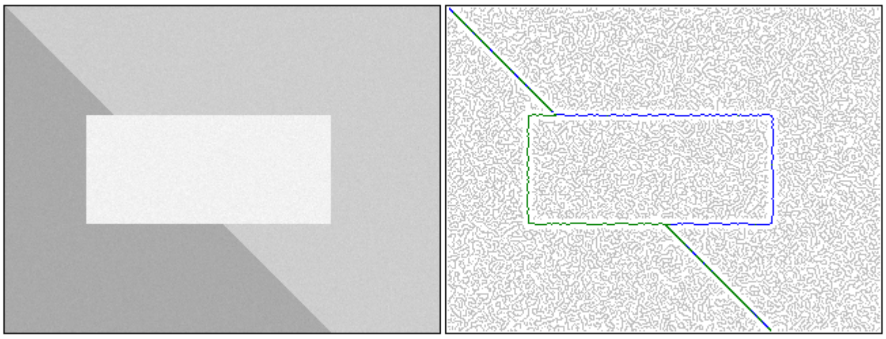


Figure 4.3: *Images demonstrating application of Canny's edge detection algorithm.* ***[Left]*** *A test image with a random Gaussian fluctuation added to simulate noise.* ***[Right]*** *An image showing edges detected (using Sobel filter) within the left-most image. Grey lines correspond to edges identified by non-maxima suppression but failed the thresholding stage. Green lines are edges identified by the non-maxima suppression that passed the initial upper threshold cut. Blue lines are edges that failed the upper threshold cut but passed the lower threshold.*

This hysteresis thresholding is demonstrated within figure 4.3. On the left is shown an image with different regions of intensity represented by varying shades of grey, with random Gaussian variation simulating noise. On the right the results of the Canny filter are shown - those pixels that survived the non-maxima suppression stage are marked by a grey, green or blue colour. Green pixels are those edges that had a gradient above T_{upper}, and blue pixels are those that had a gradient above T_{lower} and bordered a pixel already identified as an edge. Grey pixels are those pixels that passed non-maxima suppression but that did not have a sufficiently strong gradient to pass the thresholding stage, in this case they are primarily the result of the Gaussian noise introduced into the image. The edges between the darker lower left region and the central bright region are picked up by the upper threshold, but not those between the central and top right regions. The lower threshold T_{lower} however allows these less apparent edges to be picked up without any false edge detection due to Gaussian noise (since none of the Gaussian related noise is attached to the edges identified by the upper threshold).

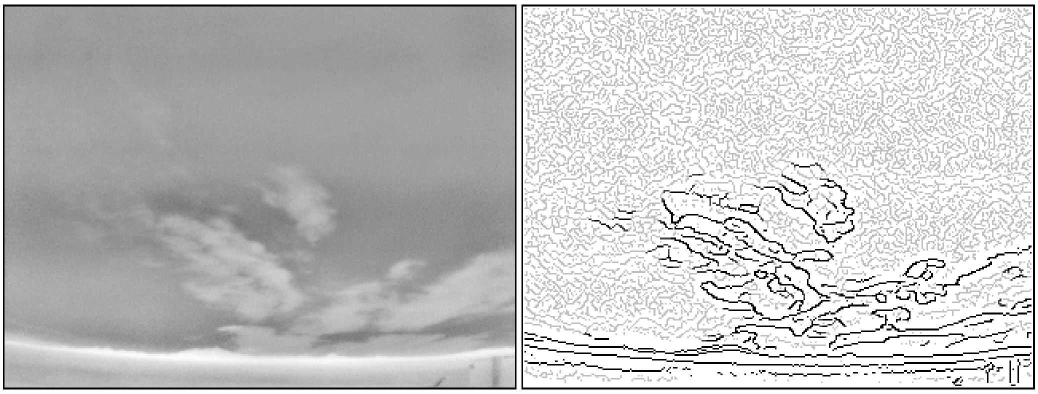


Figure 4.4: *Edges found in a cloud camera image using Canny's edge filter.* ***[Left]*** *Image recorded by the Los Leones cloud camera.* ***[Right]*** *Edges found by Canny's edge filter after using a 1σ Gaussian filter. Darker lines indicate edges found using arbitrary thresholding levels, and light grey lines indicate pixels that passed non-maxima suppression.*

An application of Canny's edge detection algorithm using arbitrary Gaussian and thresholding values to a cloud camera image may be seen in figure 4.4. The light grey lines indicate identified potential edges that did not pass the hysteresis thresholding stage, while the black lines are edges that did. Note that in a few places the shape of the cloud described by the edge lines is not complete. That is, an edge that is obvious (to the human eye) in the raw image to the left will not be detected in the right, no matter the thresholding level used, as the non-maxima suppression phase did not detect a potential edge there. This is likely due to the sometimes complicated structure of the cloud. Thus at best, the Canny edge filter (with an appropriate level of Gaussian filtering and threshold level) is likely best to function as a rough indicator of cloud presence and position.

4.1.2 Application of Canny's edge filter

With three different variables available, a great deal of customization is available for images on an individual basis, but given the large number of images it is prudent to try an optimitize a generic set of input values. There is currently no general rule for determining these parameters, therefore the degree of Gaussian filtering required must be worked out uniquely for this application, along with the gradient magnitude values for the two thresholding levels.

Non-maximum suppression was applied to a full month's (the Los Leones cloud camera in March 2005) worth of data with various levels of Gaussian filtering and the gradient magnitude of all the edges identified in the sky recorded. These images have already been processed manually (see section 4.3) to locate all of the cloud within them. This allows those edges resulting from noise and real edges (associated with a boundary between cloud and clear sky) to be identified within this set of data. An edge is said to be associated with noise if it is more than 3 pixels away from a cloud/clear sky boundary in the manually processed data.

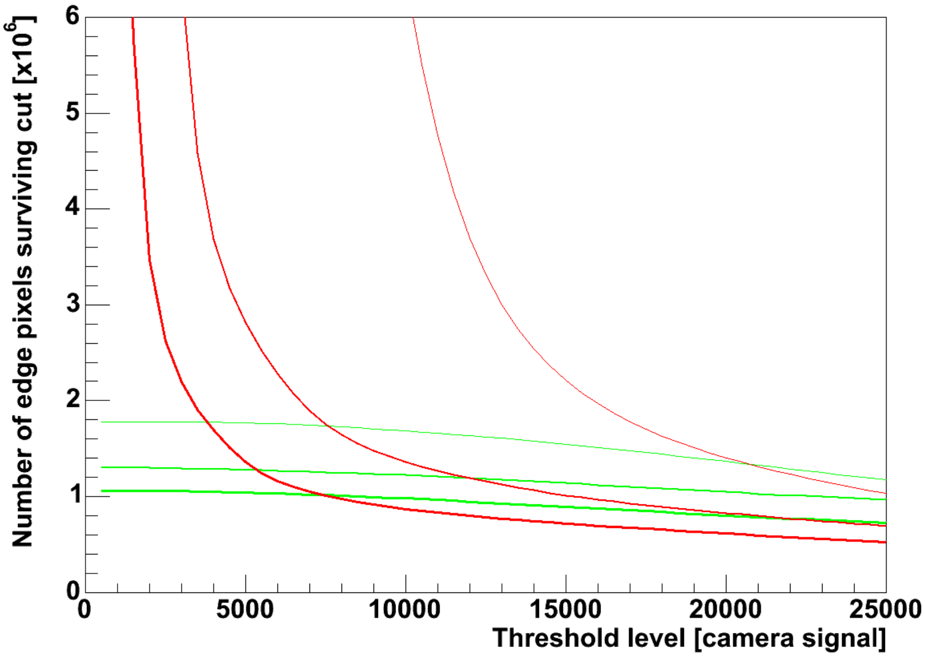


Figure 4.5: *Plot showing the number of non-maximum suppressed edge pixels with a gradient magnitude above some threshold level for a month's worth of cloud camera data (Los Leones, March 2005). Red lines indicate edge pixels identified that are not associated with real edges and green lines indicate edge pixels associated with real edges. Different line thickness' correspond to different levels of Gaussian filtering preprocessing σ =0, 1 and 2 from thinnest to thickest line respectively.*

The distribution of the real and noise edge gradients seen by the Los Leones cloud camera in March 2005, is represented in figure 4.5. The green lines indicate the number of actual edge pixels surviving a particular gradient threshold cut and the red lines indicate the number of noisy edge pixels surviving that same cut. Different line thickness' indicate different degrees of Gaussian filtering, the thinnest representing no Gaussian filtering, the thickest indicating filtering with $\sigma = 2.0$, and the third where $\sigma = 1.0$.

The thresholding levels and level of Gaussian filtering must be chosen such that the number of true edges identified is maximized. Increasing the level of Gaussian filtering decreases the number of edges detected in the non-maxima suppression stage. In the case of the sample data set in figure 4.5, increasing the filtering level from σ =0-1-2 decreases the total number of noisy edges detected from 37.2 million to 32.7 and 18.2 million respectively. At the same time the number of true edges detected drops from 1.8 million to 1.3 and 1.1 million respectively. Note that on a fractional basis though, a larger fraction of the real edges (28%) are lost than the noisy ones (12%) for a $\sigma = 1.0$ filter. Whereas for a $\sigma = 2.0$ filter a larger fraction of the noisy ones are lost (51%) as opposed to the real ones (39%).

At high threshold levels, more true edge pixels are being identified than noisy edge ones. While at lower threshold levels more noisy edges are being detected than real ones. The upper thresholding level is crucial in initially identifying potential cloud edges therefore it must be as low as possible, but not to the extent where more noise is being detected than real edges. A good tradeoff would appear to use the crossing points between the number of noisy and true edges (20800, 11600 and 7500 for σ =0, 1, 2 respectively) as the upper threshold, as any lower threshold will identify more noisy edges than real ones. The number of noisy edges drastically increases at lower thresholds, but the number of real edges identified also increases at the same time. A good lower threshold was chosen to be the point where 80% of the noisy pixels (14800, 7400 and 4000 for σ =0, 1, 2 respectively) are cut, as below this level the rate of noisy edge introduction appears to drastically increase. These threshold levels may be expected to vary between different cameras and camera flat-fieldings.

In order to evaluate the best level of Gaussian filtering to use it is prudent to examine the effect of the thresholding upon a sample of images (see figures 4.6 and 4.7). Three types of images are examined within these figures - a cloudy image, a clear night sky image and an image of a completely overcast night sky. Gaussian filters with σ = 0, 1 and 2 were applied with the associated threshold levels described in the previous paragraph. Dark lines indicate the location of detected edges, while light grey lines indicate the position of edges that did not pass the thresholding level.

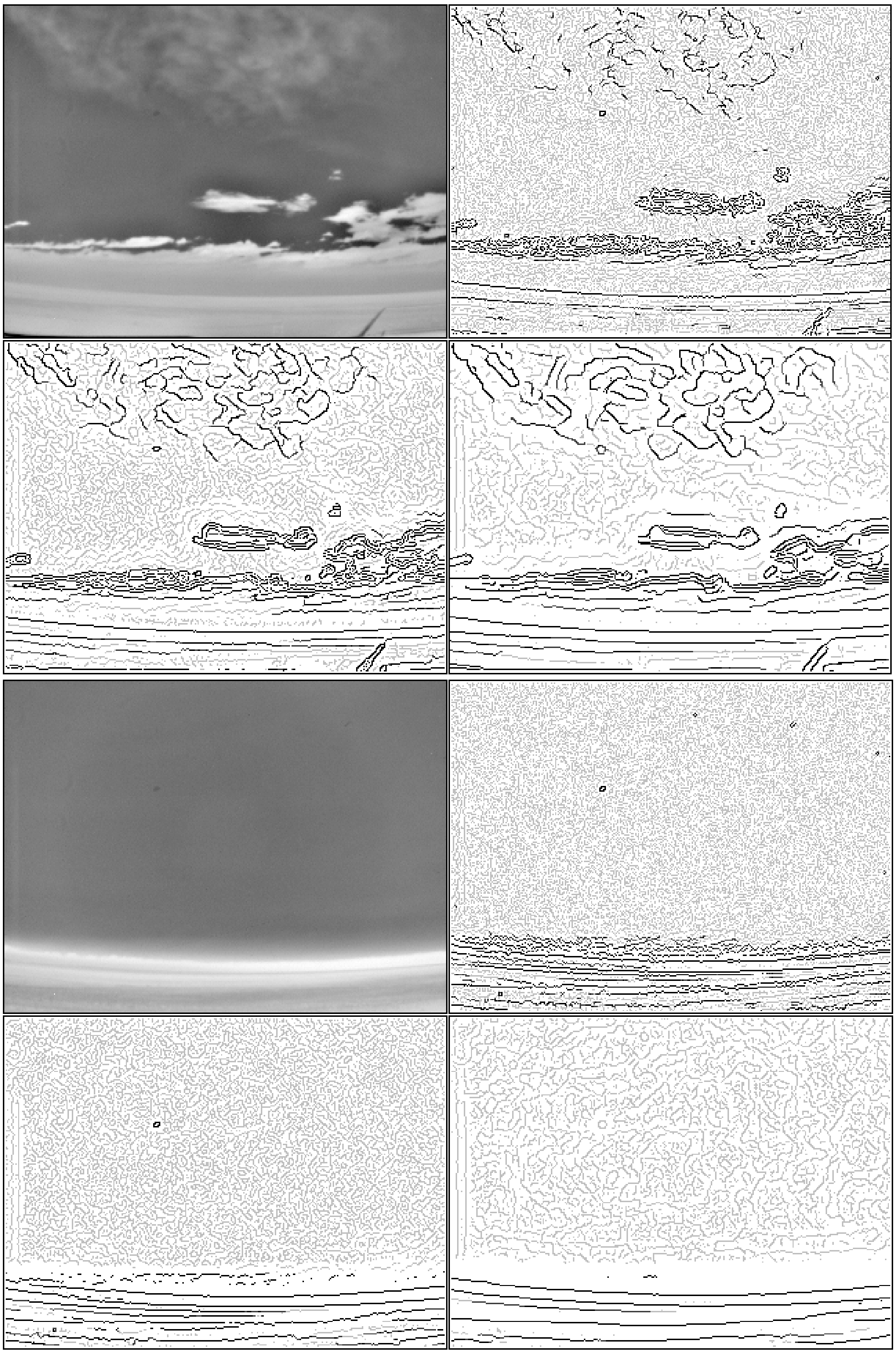


Figure 4.6: *[**Top left**] Raw cloud camera image. [**Top right**] Edges detected with no Gaussian filtering. [**Bottom left**] Edges detected with $\sigma = 1$ Gaussian filtering. [**Bottom right**] Edges detected with $\sigma = 2$ Gaussian filtering.*

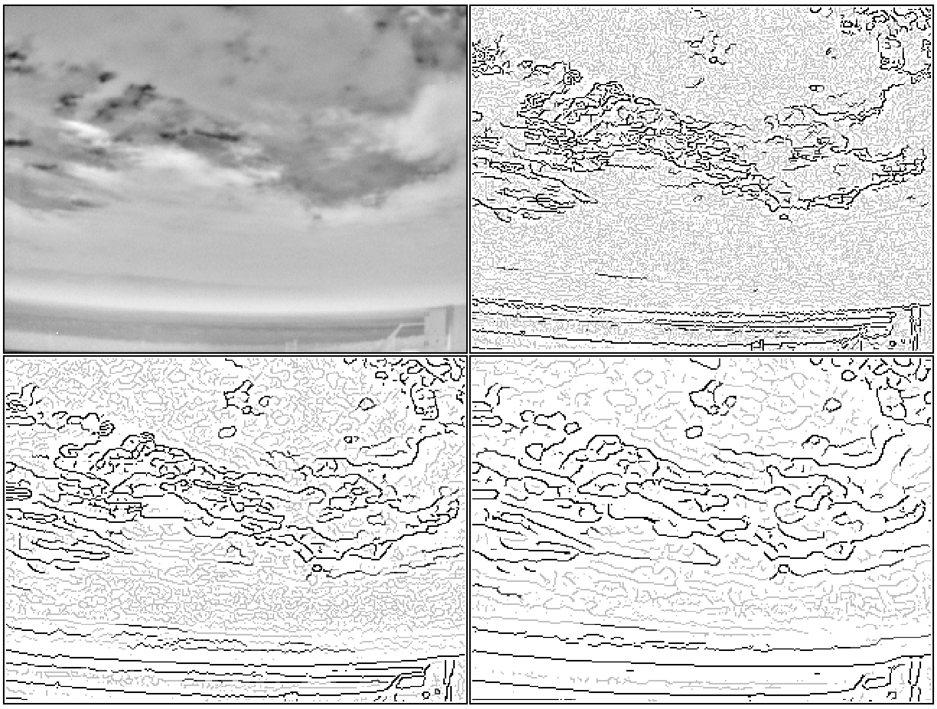


Figure 4.7: ***[Top left]*** *Raw cloud camera image.* ***[Top right]*** *Edges detected with no Gaussian filtering.* ***[Bottom left]*** *Edges detected with $\sigma = 1$ Gaussian filtering.* ***[Bottom right]*** *Edges detected with $\sigma = 2$ Gaussian filtering.*

The location of cloud boundaries are generally detected by the Canny edge filter at all three Gaussian filtering levels. This suggests, in these cases at least, that they were good choices to have made. An exception may be found in the case of the $\sigma = 0$ filtered case of the partially cloudy conditions (top-most right image in figure 4.6) where the defined threshold levels do not adequately describe the faint, higher altitude, cloud in the top of the image. The defined thresholds at higher levels ($\sigma = 1$ and 2) of Gaussian filtering detects the faint cloud structure much more readily however, suggesting that a degree of Gaussian filtering enhances the detection prospects for faint cloud.

Gaussian filtering also has the effect of significantly decreasing the number of false noise edges detected by Canny's filter. This may be observed in the clear sky parts of the cloudy and completely clear images, and in the cloudy (away from the edges) portion of the overcast image. The number of these light grey 'noisy' edges is reduced with increasing degrees of Gaussian filtering, also tending to appear more often in connected contours, similar to those edges correlated with real cloud boundaries, but with a lower associated gradient magnitude. This occurs as the Gaussian filter averages out and removes the localized noisy signal fluctuations responsible for the extra edges in the unfiltered image.

The number of edges detected that are associated with real boundaries are also reduced through an increased level of Gaussian filtering. In the non-Gaussian filtered images, many edges are detected in the neighbourhood of a boundary between cloud and clear sky (top image in figure 4.6), and the boundaries between the two cloud layers in figure 4.7. This is due to the presence of "auras" (see section 3.5.1) within the image creating extra edge gradients between the aura and the cloud or clear sky. Application of a Gaussian filter averages out the pixels in the aura region, reducing both the rate of incidence and gradient magnitude of such edges. This (using no Gaussian filter) makes the detection of the 'exact' size and shape of a cloud more difficult, but improves the response of the algorithm to strong (those with an aura) edge fragments.

In the case of the cloud camera images, using the Canny filter with no Gaussian filtering actually improves the detection prospects for strong cloud fragments, but has a low response to the fainter, higher altitude cloud fragments. Application of a Gaussian filter suppresses much of the noise associated with the image cloud or clear sky background, improving detection of fainter cloud and slightly reducing the response of Canny's filter to strong cloud fragments. The key therefore is to select the minimal Gaussian filtering level required to bring out faint cloud structure to maximise Canny's filter's response to cloud camera images. In the case of this study, this would take the form of a $\sigma = 1.0$ level of filtering, through presumably there may exist some optimal level of filtering such as $\sigma = 0.8$ or 1.3, but this is beyond the ability of this study to predict.

The dual thresholding levels used in figures 4.6 and 4.7 worked very well, but were derived using data that had already been processed. No generalized scheme for estimating the optimal thresholding levels within each image for Canny's filter currently exists. Therefore effective application of Canny's algorithm requires a degree of human supervision, due to the sensitivity of the edge finding algorithm and the diversity of cloud types that may appear within cloud camera images.

4.2 Image segmentation

Thresholding is an image segmentation process by which some signal level is chosen, and any pixels with a signal above this level are assigned to one group, and any below are assigned to another group. Clouds appear brighter in the infrared than clear night sky, therefore a good choice of a threshold will divide a cloud camera image into its cloudy and clear sky components (cloud being those pixels with a signal strength above some threshold).

Image segmentation (thresholding) is a common technique used in image processing and a variety of algorithms are available to calculate an optimal threshold for a cloud camera image. In this study two algorithms are examined, Otsu's thresholding algorithm, and the K-means cluster analysis. Such algorithms require examples of both cloud and clear night sky occuring within an image to effectively calculate a threshold however, and typically fail when presented with very isothermal scenes, such as on completely clear or overcast nights. The apparent brightness of clear night sky also varies with elevation. So to maximize effectiveness, these kinds of thresholding algorithms must often be applied multiple times at different elevations in a single image. The relationship between the clear sky signal and ground based measurements of atmospheric parameters (see section 3.4.1) may also be used to derive a threshold for a cloud camera image.

4.2.1 Otsu's thresholding algorithm

The Otsu thresholding algorithm was proposed in 1979 [139] by Otsu, as a means to choose a threshold level to segment a greyscale image into two components. He postulated that at the optimal threshold, the between-class variance of the two groups of pixels (one group consisting of those pixels with signal above that threshold, and the other below) will be maximized. The between-class variance ($\sigma^2_{between}$) is a weighted measure of the difference between the means of the two different groups and the mean of the overall signal distribution within that image, as seen in equation 4.4:

$$\sigma^2_{between} = w_B(\mu - \mu_B)^2 - w_A(\mu - \mu_A)^2 \tag{4.4}$$

where w_B and w_A are the fraction of pixels below and above that threshold respectively, and μ, μ_B and μ_A are the total mean, mean of the pixels below the threshold and the mean above the threshold respectively.

Application of this algorithm is simply through the evaluation of the between-class variance $\sigma^2_{between}$ for each possible threshold. The best threshold is chosen to be the threshold which has the highest between-class variance i.e.

1. Calculate the mean signal strength of all pixels.

2. Divide the pixels into two groups according to some threshold level.

3. Calculate the mean signal strength and fraction of pixels in each individual group.
4. Calculate the between-class variance using equation 4.4.
5. Repeat steps 2-4 until all potential thresholds have been evaluated.
6. Pick the optimal threshold as that threshold which had the largest between-class variance.

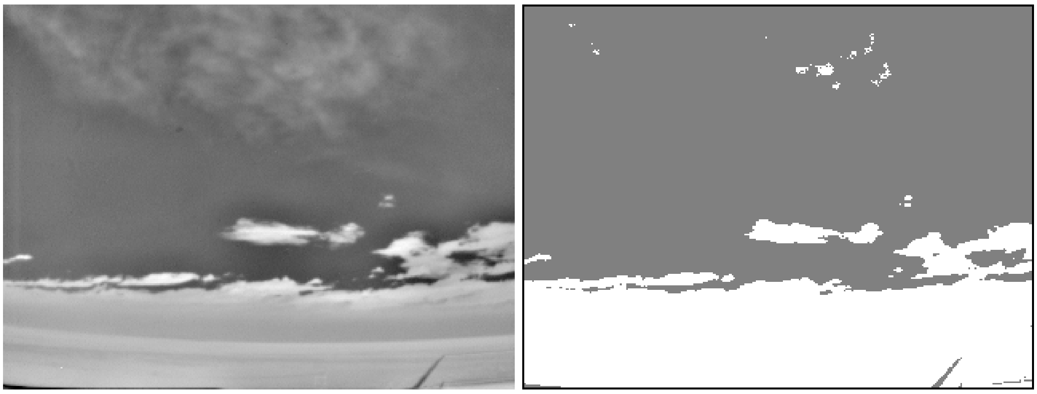


Figure 4.8: *Figure shows a cloud camera image (on the left) that has had the Otsu thresholding algorithm applied to it (seen on the right). White represents regions identified as being cloudy, and dark grey, as regions identified as clear night sky.*

Figure 4.8 shows a cloud camera image that has had a threshold level calculated by Otsu's algorithm and applied. On the left may be seen the original raw cloud camera image, and on the right is the thresholded version of that image, white representing pixels above the threshold (thought to be cloud) and dark grey as pixels below (thought to be clear sky). In this particular image, much of the brighter cloud banks close to the horizon (along with the ground) are identified as being cloud, but the higher elevation 'hazy' cloud is missed by the thresholding. This is because the threshold derived by Otsu's algorithm is strongly influenced by the strong signal of the horizon cloud, and has been set too high for detection of the fainter higher altititude cloud.

One way of overcoming this is through the use of a localized threshold. Instead of using signals from the whole image to derive a threshold, signals from parts of the image are used to derive a series of localized thresholds for different sections of the image. This sometimes increases the effectiveness of the automatic threshold detection for cloud. Within figure 4.9 is shown an image that has been thresholded by a series of localized thresholds. In this case, the localized thresholds describe the cloud conditions more effectively than the universal threshold in figure 4.8. Each row of pixels has an individual threshold derived from all the pixels within 50 rows of that row, therefore at most, only 101 rows of pixels (32320 pixels) are being used at once to derive an individual threshold. As the brighter cloud close to the horizon is not being involved in calculation of the threshold for the top half of the image, a lower threshold is derived that accurately captures the structure of the lighter cloud. Such a thresholding approach is more computationally expensive than the calculation of a single threshold.

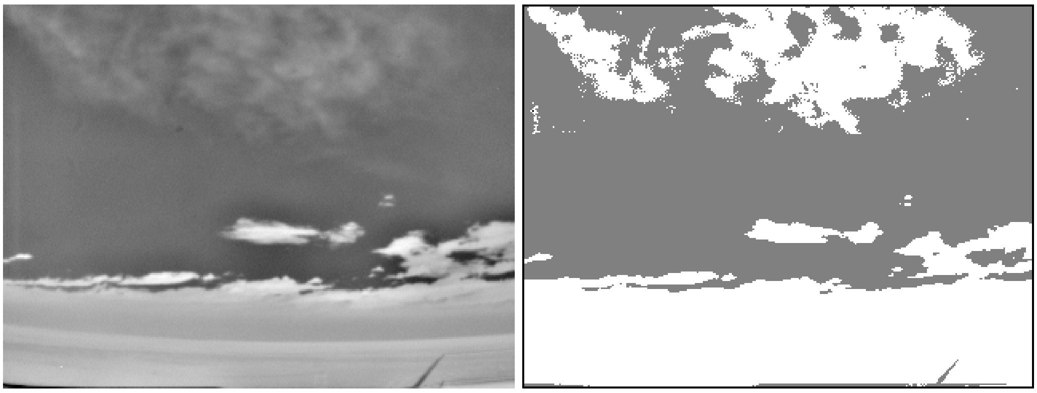


Figure 4.9: *Figure shows a cloud camera image (on the left) that has been thresholded using localized thresholds (seen on the right). White represents regions identified as being cloudy, and dark grey, as regions identified as clear night sky.*

Success of this automatic thresholding approach requires that Otsu's algorithm has had the opportunity to sample distributions of both clear sky and cloud. The algorithm implicitly assumes that there exists, in the first place, two distributions of signal to segment. Applying the algorithm to a single distribution will result in that single distribution being split into two. In the context of the cloud camera images, if a completely clear or completely cloudy image is segmented through this approach, then there will be some misidentification of clear and cloudy sky. An example of this is within figure 4.10 where the same localized Otsu thresholding approach used on figure 4.9 was applied to an almost completely clear night sky. Note in the upper regions of the image that large quantities of clear sky are being identified as cloud. This is because there is no actual cloud present in this part of the image for Otsu's algorithm to segment the clear sky against (no cloudy pixels within 50 rows of this upper image portion), therefore it adopts an incorrect

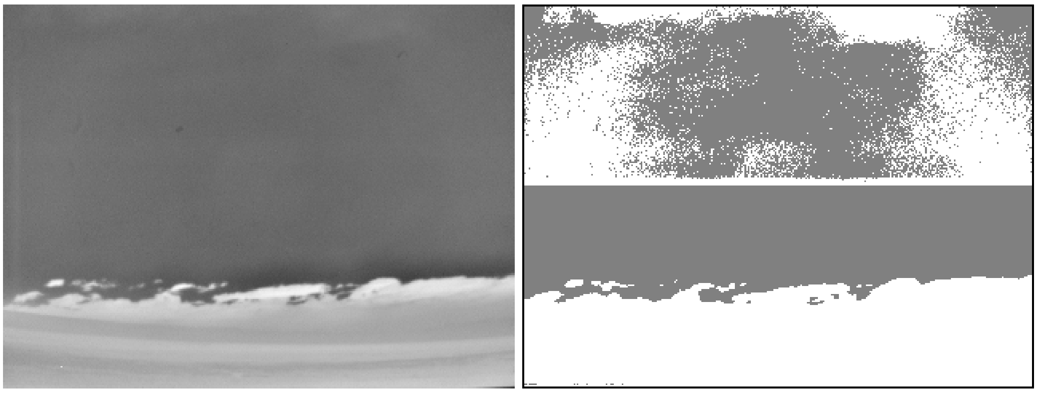


Figure 4.10: *Figure shows a cloud camera image (on the left) that has been thresholded using localized thresholds (seen on the right). White represents regions identified as being cloudy, and dark grey, as regions identified as clear night sky. The 'white line' seen across the centre of the right-most image is the result of the pixels closer to the horizon being brighter than those at higher elevations, and Otsu's algorithm subsequently identifying them as cloud.*

threshold at some level within the clear sky distribution, instead of above it. The lower portion of the image is thresholded correctly, as both clear sky and cloudy pixels are being used in the thresholding process.

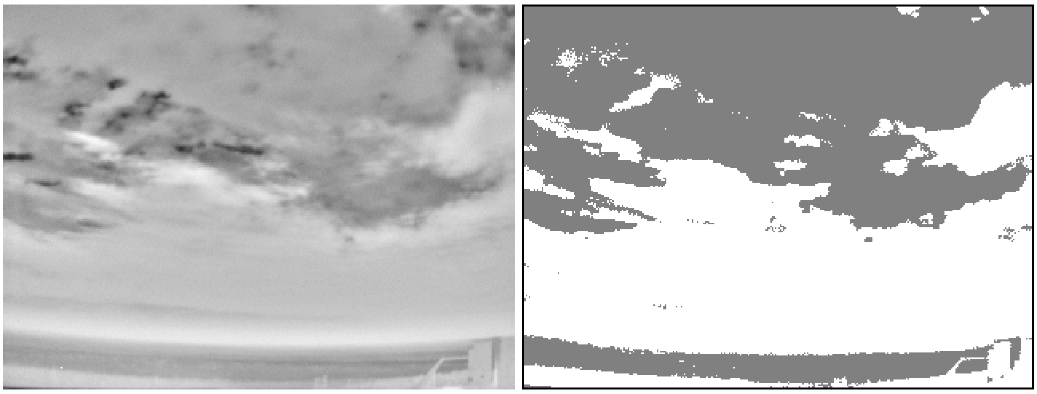


Figure 4.11: *Figure shows a cloud camera image (on the left) that has been thresholded using localized thresholds (seen on the right). White represents regions identified as being cloudy, and dark grey, as regions identified as clear night sky.*

Another problem with the automatic thresholding technique occurs when multiple cloud layers are present. As seen in figure 4.11, Otsu's algorithm has correctly identified the brighter, lower elevation cloud layer as cloud, but the dimmer (but still cloudy) upper portion of the image has been segmented as clear sky as it is the lower intensity distribution.

On its own at least, automatic segmentation is unreliable for the detection of cloud and clear sky within cloud camera images. While the algorithm is capable of distinguishing cloud and clear sky if both cloudy and clear pixels are being considered, accuracy is lost when presented with images that are completely clear, overcast or have multiple cloud layers, each with different intensity distributions. Thus a generic application of an automatic thresholding algorithm will not accurately locate the position of clouds in some types of image.

4.2.2 K-means cluster analysis

K-means cluster analysis is a process by which some number of data points are sorted into some number (‘K’) of clusters [129]. In the case of the cloud camera images, this algorithm aims to partition a particular image into different distributions of intensity, such as cloud or clear sky. Unlike the automatic thresholding technique described in section 4.2.1, this algorithm may partition cloud camera images into several different classes, as opposed to just two. As one of the problems with the automatic thresholding approach was the chance that clouds of differing intensities may not be detected, K-means cluster analysis may provide a superior method of cloud detection. An example of an image that has been segemented in this way may be seen in figure 4.12.

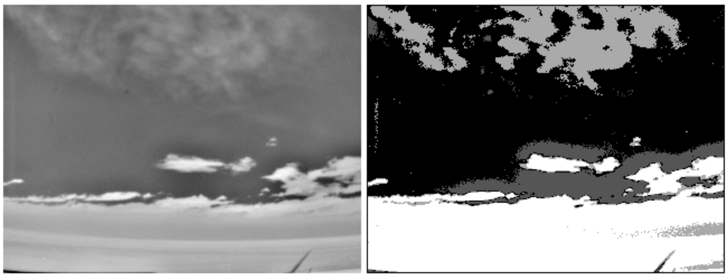


Figure 4.12: *An example of a cloud camera image being seperated into 4 different partitions using the K-means clustering algorithm. On the left is the original image, while on the right is the segmented image in which each partition is indicated by a different shade of colour.*

This algorithm functions first by randomly (or heuristically) assigning each pixel within an image to one of the 'K' partitions that the image will be attempted to be segmented into. A series of means are then calculated for each parameter the pixels are being segmented upon, for example if the pixel signal is just being considered, then the mean signal is calculated for each partition from the signals of the partitioned pixels. The variance (the square of the difference between the mean and the pixel intensity) of each pixel from these means is then evaluated, and the pixel reassigned to whichever partition it has the least variance with. Then the means are recalculated and the process continues until pixels are no longer being reassigned to different partitions i.e.

1. Decide how many segments to partition the image into.
2. Assign (either randomly or heuristically) each pixel in the image to one of the partitions.
3. Calculate the mean signal strength of the pixels in each partition.
4. Calculate the variance between each pixel's signal and the mean signal of each partition.
5. Reassign each pixel to the partition with which it has the least variance.
6. Repeat steps 3-5 until no pixels are being reassigned.

The K-means algorithm is guaranteed to converge, but the solution arrived at by the algorithm is not necessarily unique as it depends upon the initial seeding of the partitions. Randomly assigning the initial seeding of the partitions sometimes results in different partitions being calculated by the algorithm. An example may be seen in figure 4.13, where an image undergoing the algorithm has resulted in two different partioned images. In the bottom left image, the faint cloud in the upper regions of the image is clearly identified as a seperate intensity distribution, while in the bottom right image, the cloud is counted as being part of the clear sky distribution. Both these results were initially seeded with a random distribution.

Also worth noting is that the algorithm does not always succeed in partitioning the image into the number of partitions initially set by a user, resulting in fewer partitions than demanded. This occurs as some partitions merge due to lack of diversity in pixel signal distributions. While ultimately depending upon the initial distribution of partitions, it is rare to obtain more than 4 or 5 partitions in the cloud camera images.

Different strategies exist that can reduce or eliminate the degeneracy of the segmentation solution for an image. Some of these include a more

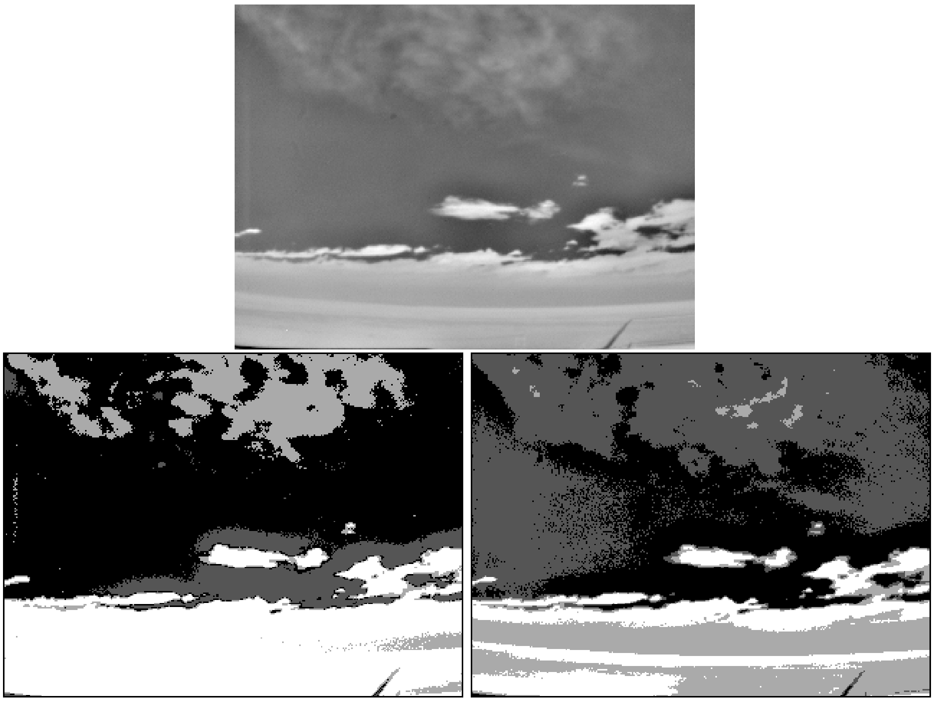


Figure 4.13: *Image on the top has been grouped into 4 partitions using the k-means algorithm. Two different results achieved may be seen on the bottom, with different colours representing different partitioned regions.*

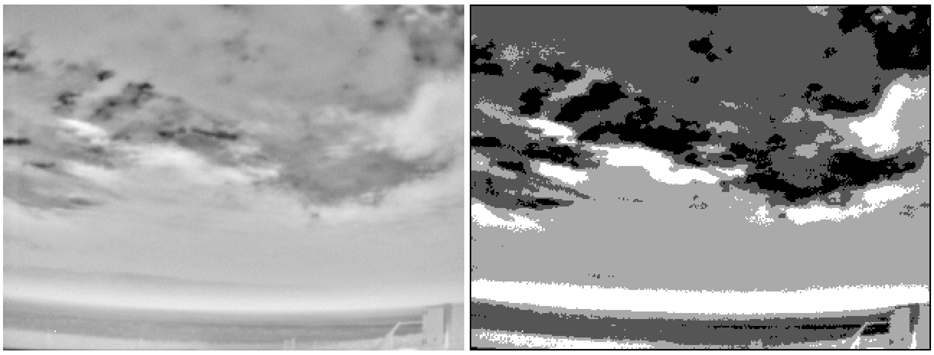


Figure 4.14: *On the left is a cloud camera image of overcast cloud conditions which has been segmented into 4 different regions using the k-means algorithm. On the right may be seen the result of the algorithm with different shades of grey denoting different regions.*

refined approach to defining the initial assignment of partitions [22]. Or repeating the algorithm several times upon a single image, and selecting the result that best satisfies some quality criterion. Ultimately in the case of the cloud camera images however, even if an 'optimal' partitioning result is achieved there still remains the problem of identifying which partitions are cloud, and which are clear sky. This problem is demonstrated in figure 4.14 where an overcast sky has been segmented into 4 regions using the K-means algorithm. All of the regions identified, are regions that could be classified as cloudy, including the partition corresponding to the darkest regions of the image. This clustering algorithm, on its own at least, cannot be relied upon to consistently correctly identify cloudy regions and suffers from the possibility of degenerate solutions being arrived at - depending upon the initial seeding conditions.

4.2.3 Atmospheric parameter based thresholding

Within section 3.4 it was suggested that it is possible to predict the signal registered by the camera with parameterizations of the forms seen in equation 4.5 or equation 4.6.

$$S = A + T_{sky} \sum_{i=1}^{\infty} B_i \left(\frac{(T_{chop}^4 - T_{sky}^4)}{T_{sky}^4} \right)^i \tag{4.5}$$

$$S = A + B(T_{chop}^4 - T_{sky}^4) \tag{4.6}$$

where S is the camera signal output, T_{chop} is the effective temperature of the camera chopper, T_{sky} is the effective temperature of the sky, and A, B and B_i represent the fitted parameters. With these equations it is possible to create a parameterization that predicts the signal registered by the camera when viewing clear night sky. Such parameterizations are unique to different cameras and flat fielding calibrations, however once developed would help an automated thresholding procedure to be developed.

In order to create a parameterization for clear sky signal, a sample of clear sky signals must be obtained, the effective temperature of the clear sky known, and the temperature of the chopper must be known. Most of the cloud cameras do not have an internal temperature sensor installed, therefore the chopper temperature approximation of '$T_{chop} \approx T_{amb} + 6.8$' from section 3.4 must be used. The identification of clear night sky examples and temperatures is more problematic however.

In his PhD thesis Maghrabi [130] suggested that the effective temperature of clear sky may be parameterized as a function of ambient temperature and water vapor pressure, as seen in equation 4.7.

$$T_{clear} = a + bT_{amb} + c\sqrt{\epsilon_0} \tag{4.7}$$

where T_{clear} and T_{amb} are the clear sky and ambient temperatures in degrees Celsius, ϵ_0 is the ground level water vapor pressure in mb and a, b and c are the fitted parameters. A generalized form of this parameterization was calculated by Maghrabi [130] from data collected in several locations around the Earth at a zenith angle of 0°, but application to the cloud cameras requires a prediction of sky temperatures at multiple non-zenith angles and a unique parameterization developed for the Auger Observatory site to maximize precision.

The cloud cameras are viewing a region of sky close to the horizon. This means that extrapolation of the relevent effective sky temperatures from the temperature directly overhead the camera is difficult, as the $\ln(\sec\theta)$ ($\theta =$ zenith angle) approximation suggested by Maghrabi [130] breaks down close to the horizon. Therefore a different parameterization of the form in equation 4.7 must be developed, that is valid for elevations close to the horizon, to successfully describe the relevent effective sky temperatures.

Over a some limited range of temperatures (T) it is reasonable to approximate T^4 as some linear function of T. Applying such an approximation to equation 4.6 implies that the clear sky signal S_{clear} registered by a cloud camera is proportional to the difference between the chopper and clear sky temperatures, as seen in equation 4.8.

$$S_{clear} = a + b\,(T_{chop} - T_{clear}) \tag{4.8}$$

where S_{clear} is the clear sky signal, T_{chop} and T_{clear} are the chopper and clear sky temperatures respectively, and a and b are fitted parameters. Such a linear fit to clear sky data is not as successful as the fit suggested in equation 4.6, having an RMS fit of 240, compared to that of equation 4.6, which has a fit of 217 (using the data sample in section 3.4.1). It is, however, not a poor fit either. Substituting equation 4.7 into equation 4.8 reveals a simple linear parameterization in equation 4.9 between clear sky signal, ambient temperature and water vapor pressure (ϵ_0 in mb).

$$S_{clear} = A + BT_{amb} + C\sqrt{\epsilon_0} \tag{4.9}$$

where A, B and C are fitted parameters. Fitting this parameterization sidesteps the problem of having to directly calculate the effective clear sky temperature at low elevations, and only requires the selection of clear sky signals and the availability of ambient temperature and relative humidity measurements by on-site weather stations. The problem of selecting clear sky examples remains however.

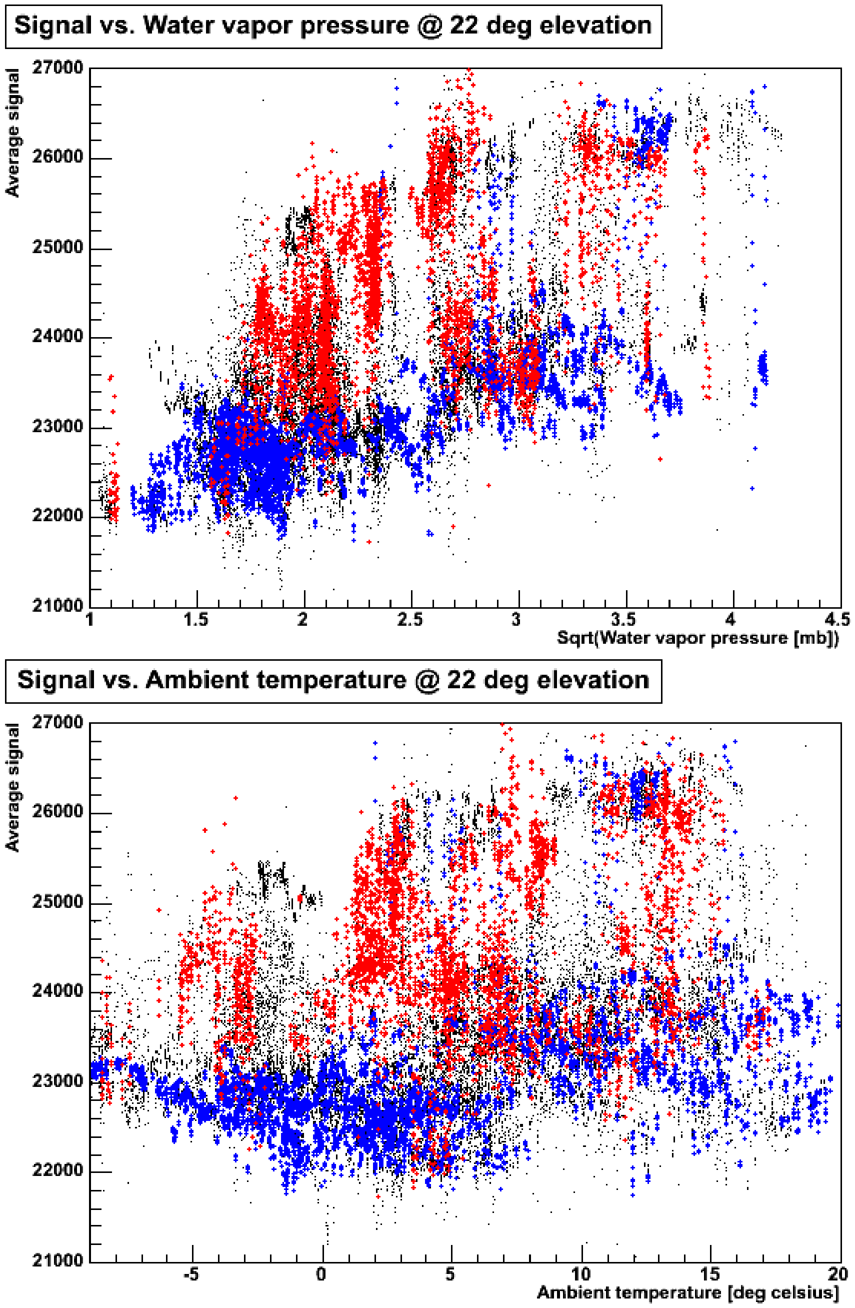


Figure 4.15: *Mean signal recorded between 21.5 and 22.5 degrees of elevation within an image, plotted against the square root of the measured water vapor pressure at that time* ***[Top]*** *and ambient temperature* ***[Bottom]****. Los Leones images, February to October 2007 were used. Red marks indicate measurements that occured while the average cloud fraction measured by all LIDARs was greater than 90%, and blue marks while the cloud fraction was 0%. Black marks are all other measurements.*

The mean signal within Los Leones cloud camera images between February and October 2007 have been plotted against ambient temperature and the square root of the water vapor pressure in figure 4.15. This signal is the mean of the pixel signal strengths within a particular image that lie within half a degree of 22° elevation from the horizon. The ambient temperature is the average temperature measured by all the weather stations at that particular point in time. Weather stations do not directly measure water vapor pressure, but it may be extrapolated from measurements of relative humidity and ambient temperature with equation 4.10:

$$\epsilon_0 = \frac{H}{100}.exp\left(\frac{17.67T_{amb}}{T_{amb}+243.5}\right) \tag{4.10}$$

where ϵ_0 is the water vapor pressure (mb), H is the relative humidity (%) and T_{amb} is the ambient temperature (°C). Red markers indicate measurements that were taken at overcast (>90% LIDAR cloud coverage) times and blue markers indicate measurements that were taken at very clear (0% LIDAR cloud coverage) times.

Generally speaking, signals recorded on LIDAR-identified clear nights have a lower average signal at 22° elevation than signals recorded at the same elevation on cloudy nights. This is because on cloudy nights cloud is more likely to be seen than clear night sky, which increases the infrared flux and subsequent signal recorded by the camera. There appears to be some overlap between the signals associated with clear and cloudy nights as some signals identified as being clear are located in amongst groups of signals that have been identified as cloud, and vice versa. This is because the cloud conditions detected by the LIDAR sites above the fluorescence detectors are not necessarily correlated with the cloud conditions observed by the cloud cameras over the surface detector array. If a parameterization based upon equation 4.9 is to be created, then those signals that are associated with clear sky conditions must be identified.

Identifying signals associated with clear night skies requires more than just LIDAR information, as LIDAR cloud fraction measurements are not always a measure of cloud conditions within a cloud camera's field of view. Another way of identifying cloudy conditions is through the RMS of the mean signal recorded at a particular elevation. If both cloud and clear sky are being observed at a particular elevation, then the RMS of the signals at that elevation are going to be greater than if just clear sky or cloud fills that particular part of the sky. This is because a section of sky containing both cloud and clear night sky will have more variance in recorded pixel signals than would otherwise be the case. Therefore by removing measurements with a high RMS, the quality (in terms of an increase in the fraction of measurements coming from a purely clear night sky) of a set of signal measurements will increase.

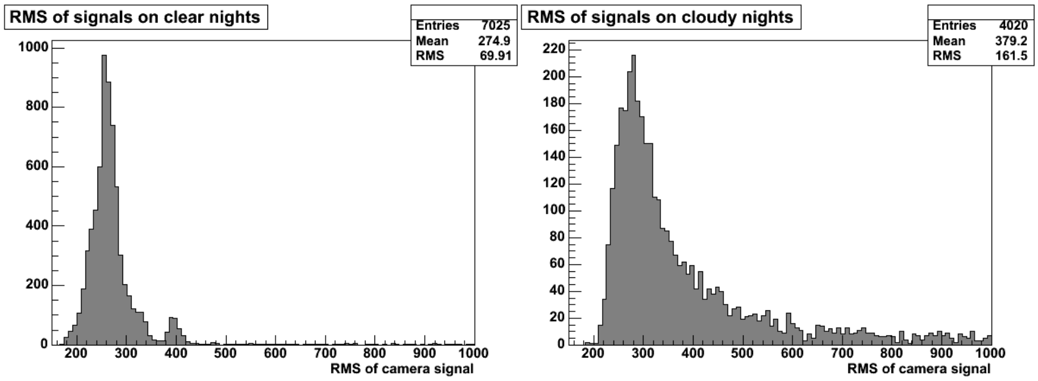


Figure 4.16: *Distributions of the RMS of signals taken at 22° elevation by the Los Leones cloud camera between February and October in 2007.* ***[Left]*** *RMS of signals recorded on LIDAR measured cloud free nights.* ***[Right]*** *RMS of signals on very cloudy (> 90% LIDAR cloud coverage) nights.*

This is demonstrated in figure 4.16, where the distribution of the RMS of signals at 22° elevation in images recorded by the Los Leones cloud camera between February and October 2007 are shown. Signals on clear nights typically have an RMS below around 350, while the RMS of signals on cloudy nights have a much wider distribution of RMS values, with a significant proportion of its distribution above 350. Both distributions have a similar peak in their RMS values between 250 and 300, this is likely due to completely clear or completely cloudy conditions within the section of sky being considered. Parts of both distributions away from this peak are likely due to mixed cloud conditions at the 22° elevation within the image. Therefore a cut on RMS will remove those signals that are partly contaminated by cloud, but will leave signals that are either completely cloud or clear sky based.

Figure 4.17 shows the distribution of the mean signals of the measurements in the left-most plot in figure 4.16 that have an RMS less than 345. The number 345 comes from the mean of that distribution added to the RMS of that distribution, and acts as an almost arbitrary threshold between the signals thought to be from a completely clear/overcast sky and a partially cloudy sky. The signal distribution in figure 4.17 appears to contain two different distributions, a large one centered at ≈22800, and a small one at ≈26400. The large distribution is thought to consist of signals from completely clear skys, or from very faint, high altitude cloud, while the smaller

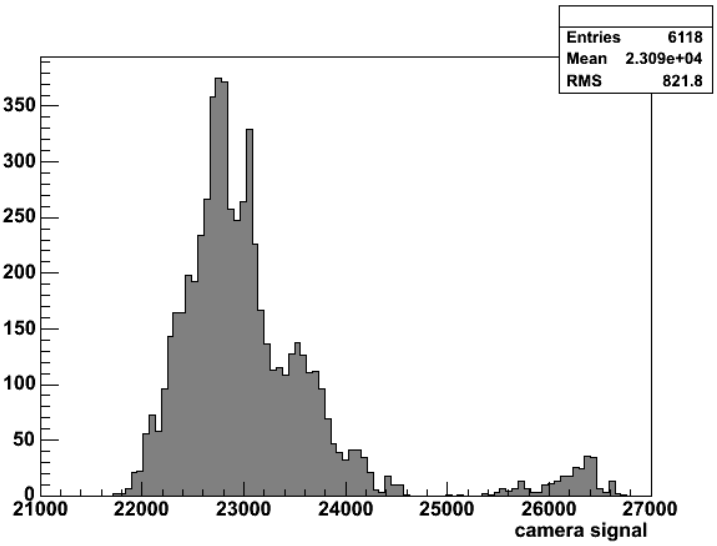


Figure 4.17: *Distribution of the mean signals at 22° within images recorded by the Los Leones cloud camera between February and October 2007, that occurred on clear nights (0% LIDAR cloud coverage), and that have an RMS of less than 345.*

distribution is thought to contain signals from completely overcast examples of the sky. Another cut, this time based upon signal strength is therefore advisable in order to remove any overcast signals. Due to the LIDAR cloud fraction cut (0%) already applied, most of the signals are going to be of the completely clear variety. Applying a cut based upon the mean and RMS of the figure 4.17 distribution will therefore likely preserve the clear signals as the mean is being weighted primarily by clear signals already.

Multiple parameterizations of the form in equation 4.9 must be created in order to successfully predict the signal expected from clear sky, as the effective temperature of clear sky (and hence camera signal) is expected to vary with zenith angle. Figure 4.18 demonstrates fits made to the 2007 Los Leones (February to October) cloud camera data at elevations ranging from 3 to 29°. In each image, within 0.5° of each degree of elevation, the mean pixel signal is calculated along with its RMS and the ambient temperature (Kelvin) and water vapor pressure at the time of the image recording. At each individual elevation, this data then underwent the LIDAR, RMS and signal strength cuts described earlier, in order to remove any cloud contaminated measurements. Equation 4.9 was then fitted to the surviving clear sky data at each elevation. The top-left, top-right and bottom-left plots indicate the value of the fitted A, B and C parameters respectively, at different elevations. The bottom-right plot indicates the residuals of equation 4.9 fitted to the selected data at that elevation.

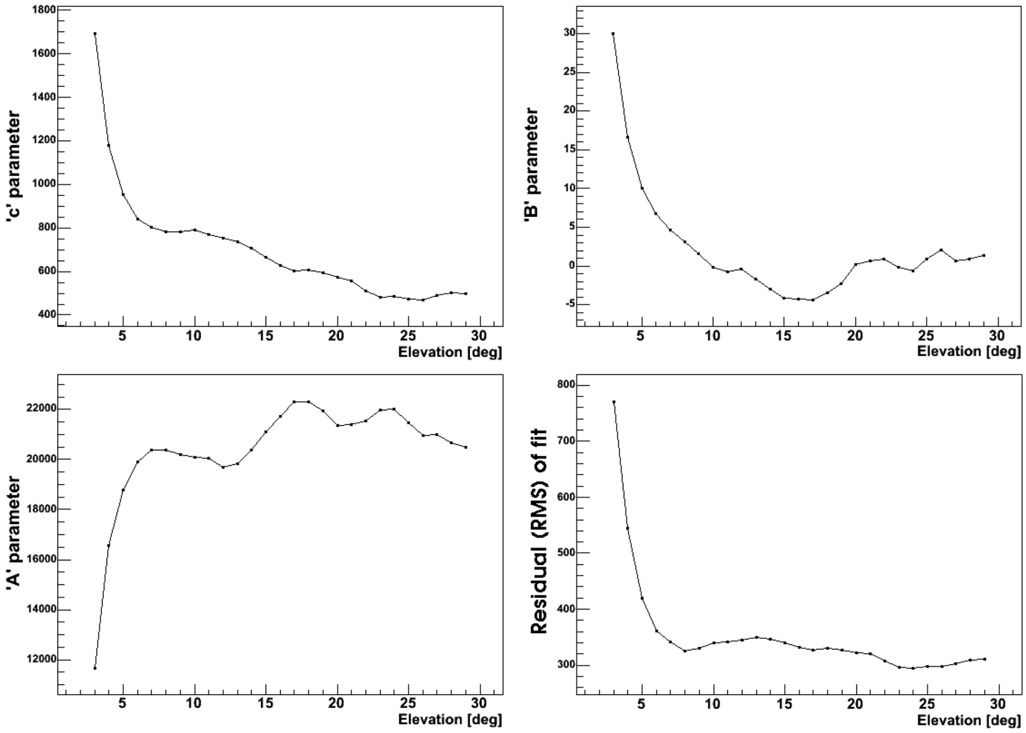


Figure 4.18: *Plots describing the parameters and fitted RMS of equation 4.9 to data from Los Leones between February and October in 2007.* ***[Top Left]*** *Variation of fitted C parameter with elevation.* ***[Top Right]*** *Variation of fitted B parameter with elevation.* ***[Bottom Left]*** *Variation of fitted A parameter with elevation.* ***[Bottom Right]*** *Variation of the residuals (RMS) of the fit with elevation.*

The C parameter, represented in the top-left plot of figure 4.18 represents the fitted constant associated with the square root of the water vapor pressure. It is shown to increase approximately linearly from 29° of elevation down to 6°, below which it rapidly increases. This constant may be thought of as representative of the effect the estimated water vapor pressure has upon the camera signal. At progressively lower elevations, the optical thickness of the clear sky night sky increases due to the increased water vapor content seen along the line of sight. Therefore the effect of water vapor upon camera signal is expected to increase at lower elevations, as is observed. Below elevations of 6° the rate at which water vapor contributes to the camera signal if much higher than at higher elevations. This is likely due the to optical depth, at elevations below 6°, becoming large enough become an effective black body - changing the response of the camera, as discussed in section 3.4.3.

The B parameter, shown in the top-right plot represents the fitted constant associated with the ambient temperature. Like the parameter associated with water vapor pressure (C), the clear sky signal dependence upon temperature increases at lower elevations. Once again, this is likely due to the increasing optical thickness at lower elevations better approximating a black body, and thus increasing the emission of the infrared flux with respect to temperature..

The A parameter, represented in the bottom-left plot of figure 4.18 represents the signal a particular parameterization predicts when faced with a clear sky with an effective temperature of 0 Kelvin and no water vapor pressure ($\epsilon_0 = 0$). This constant may be thought of a rough measure of the dependence of temperature and water vapor pressure (through their B and C parameters) on determining the signal at a particular elevation. At low elevations, the B and C parameters take on greater values, increasing their contribution to the predicted signal. Since a larger proportion of the signal is being determined by water vapor and temperature, then the fitted parameter A must decrease.

The final plot in the bottom-right corner of figure 4.18 shows the residuals of the fitted parameterization compared with the fitted data. Above 6° of elevation the RMS tends to range between values of 400 and 440, while at lower elevations the fit gets progressively worse with larger residuals. This is likely because, at lower elevations, regions of sky very far away from the Pierre Auger Observatory are being examined. The increased distance from the observatory means that the cloud conditions being observed by the LIDARs are less likely to be correlated, which results in an increase in the chance that cloudy samples are surviving the quality cuts and contaminating the sample of ‘clear sky’ examples the parameterization is being fitted to.

Once a parameterization is fitted for a particular elevation level it may be used to estimate the signal strength expected to be seen for clear sky, and compare this estimate against the actual signal. In order to effectively use it as a cloud detection method however it must be used to set some particular signal level, above which a pixel will be considered to be cloud, and below which it is considered to be clear sky. As each parameterization has an RMS associated with its residuals of the fit it may be useful to use multiples of this RMS as a way of setting this threshold.

The Los Leones 2007, February to October data used to calculate the fits in figure 4.18 were processed (all cloud present in images identified) manually, allowing an evaluation of the success rate of a particular threshold level to take place. The parameterization at 22 degrees of elevation in figure 4.18 takes the form of equation 4.11, with an associated fitted RMS of σ =307.

$$S_{clear} = 21533 + 0.932T_{amb} + 511.2\sqrt{\epsilon_0} \tag{4.11}$$

where S_{clear} is the predicted clear sky signal, T_{amb} is the local ambient temperature (in Kelvin) and ϵ_0 is the ground level water vapor pressure (in mb). In the tops plots of figure 4.19 are shown histograms showing the difference (in terms of multiples of the fitted RMS σ) between the clear sky predicted signal S_{clear} and the signals measured by pixels (between elevations 21.5-22.5°) identified as cloudy or clear by the manual processing.

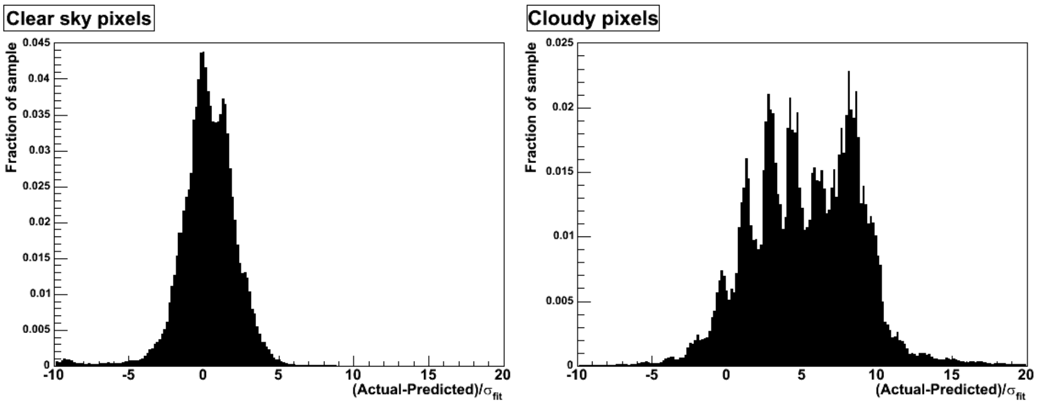


Figure 4.19: *Processed data was gathered from data recorded by the Los Leones cloud camera in 2007, between February and October, and the difference between the signal measured by each camera pixel, and the predicted clear sky signal recorded. **[Left]** The distribution of signal differences for pixels identified by previous analysis as being clear. **[Right]** The distribution of signal differences for pixels identified by previous analysis as being cloud.*

The left-most plot in figure 4.19 shows the distribution of the difference between predicted clear sky and actual pixel signal strengths, for pixels identified as being free of cloud. The distribution shows a strong peak centered at approximately 0σ, which is consistent with the idea that the LIDAR, RMS and signal quality cuts mentioned earlier have ensured that the parameterization in equation 4.11 was fitted to mostly clear sky data. At approximately 2σ there is another, smaller, peak that is likely the result of very high altitude cirrus cloud of a very similar intensity to the clear night sky. Due to its similarity it was likely mis-identified as being clear sky in the manual processing process. Practically speaking, if such cloud is missed in an analysis it is generally unlikely to have any effect on any analysis, as this cloud layer would usually be above the extensive air shower being observed by the fluoresecence detector. There is also a distribution of pixels falling below -5σ of the predicted clear sky signal. These are likely pixels associated with the 'auras' sometimes observed around bright cloud banks (see section 3.5.1) that are an imaging artefact of the scene differencing process in the camera.

In the right-most plot of figure 4.19 is shown the differences between the signals of pixels identified as being cloud and the signal expected for a clear night sky at 22° of elevation. Most of the pixels identified as cloud, have signal intensities above the clear sky predicted signal. The peak and distribution around and below 0σ, are likely clear sky pixels that have been misidentified by cloud during the manual processing of the data. Above 0σ are what appear to be 5 separate peaks. These are likely from very overcast periods in the data, where there was a significant number of cloudy pixels being recorded.

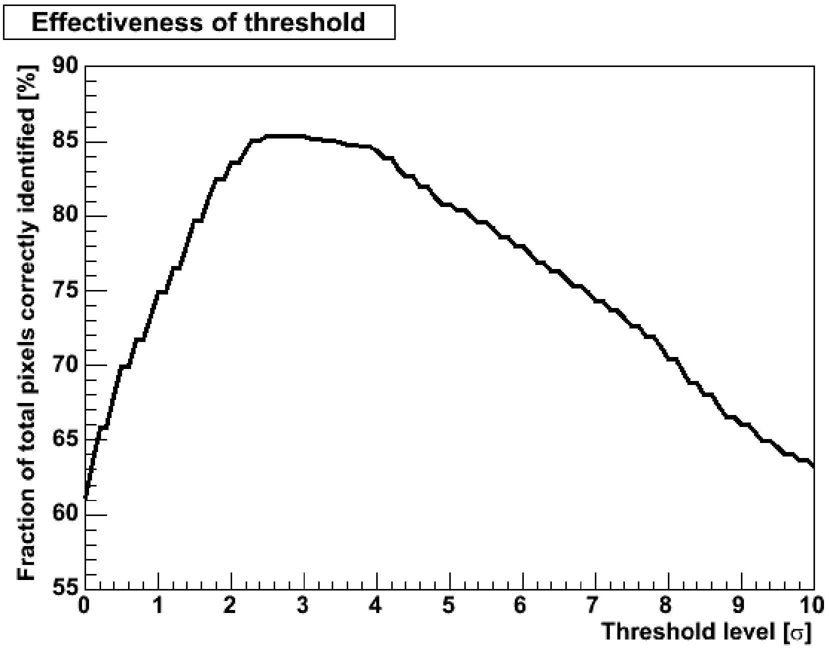


Figure 4.20: *Plot of threshold cut vs fraction of all pixels (from figure 4.19) correctly identified at different thresholds.*

The effectiveness of a particular threshold cut may be evaluated using the data in figure 4.19. A threshold is defined is terms of some multiple of the fitted residual RMS of equation 4.11 added to the predicted clear sky signal. Pixels with signal strengths above the threshold are considered to be cloud, and those below are considered to be clear sky. In figure 4.20 is shown the fraction of pixels (from the sample of data used for figure 4.19) that is correctly identified for a given threshold level. A cloudy pixel is said to be identified correctly if its signal strength is above the threshold, while a clear pixel is correctly identified with a signal below the threshold. According to this figure, the best threshold is at a level 2.7σ above the predicted clear sky signal strength, with a success rate of approximately 85%.

There are a few reasons why the apparent success rate of this thresholding process is not higher. Firstly is the presence of high altitude cirrus cloud with a similar signal intensity to clear night sky. This parameterization does not appear to have the precision necessary to distinguish between cloud and clear sky in this situation, therefore the success rate is being negatively affected by those cirrus, high altitude cloudy pixels being identified as clear sky. The second reason is to do with the temperature and weather readings used to create the clear sky signal parameterization. There is no guarantee that the selection of water vapor pressures and ambient temperatures that survive the quality cutting process (mentioned earlier), are representative of the entire range of weather parameters that may be encountered during camera operation. This means that the parameterization may have to extrapolate the clear sky signals for temperatures or water vapor pressures much different to that used to fit the parameterization in the first place. This reduces the accuracy of the clear sky signal estimation, and so makes the thresholding process less effective than it might otherwise be.

Also contributing to inaccuracy is the fact that the ground level water vapor pressure is not a perfect predictor of the total atmospheric water vapor content above ground level. The parameterization ultimately uses an estimation of the vertical distribution of water vapor above the Earth's surface. Without a perfect description of the vertical water vapor profile the effective emissivity of the atmosphere, and thus camera clear sky signal, cannot be known.

It is therefore unlikely that a single threshold level can be sufficient for a complete set worth of data, though the 2.7σ level mentioned here may be a useful starting level. In a practical application of this thresholding process, it may be necessary to adjust the threshold level for some sets of images as a set of mostly overcast images will tend to favor a lower threshold, while a set consisting mostly of clear sky images will tend to favor a higher threshold, and hence σ level.

4.3 Cloud detection algorithm evaluation

There are multiple ways to apply the different techniques described in the previous section to detect cloud within a cloud camera's infrared images. Some algorithms are more effective than others. In this section, different algorithms will have their effectiveness evaluated using a sample of the Los Leones cloud camera data (February to June, 2007) consisting of 17580 images that have been manually scanned for cloud. Algorithms will be described, applied to the test set of data, and the accuracy evaluated by the fraction of (clear or cloudy) pixels that are in agreement with the manually processed version of the image. In the final summary section the algorithms are discussed and compared against one another.

4.3.1 Segmentation: Weather dependent threshold

Within section 4.2.3 is discussed a weather dependent threshold. Several months worth of data is gathered, and with the aid of LIDAR and quality cuts, a selection of clear night sky signals are identified and parameterized as a function of ambient temperature and ground-level water vapor pressure. A threshold using this parameterization may be determined as some multiple of the RMS of the residual from the fit of the parameterization to the data. Pixels with signal strengths above this threshold are considered to be cloud, and pixels below as clear sky.

1. For several months worth of cloud camera data (from the same camera and with the same flat-field calibration) calculate the mean signal strength and RMS in each image, for each elevation between 3-29°.

2. Record the average fraction of cloud measured by the LIDARs when each cloud camera image was taken.

3. Using LIDAR cloud fraction, RMS and signal strength cuts determine which mean signal strengths are from a cloud-free night sky.

4. For each elevation fit a parameterization of the form described in section 4.2.3, and record the RMS of the residuals from each fit.

5. Determine a threshold for each individual pixel within a particular image as being the closest (in terms of elevation) parameterization plus some multiple of the fitted residual RMS for that parameterization.

6. If a pixel is above this threshold, consider it as being cloud, if below, then as clear sky.

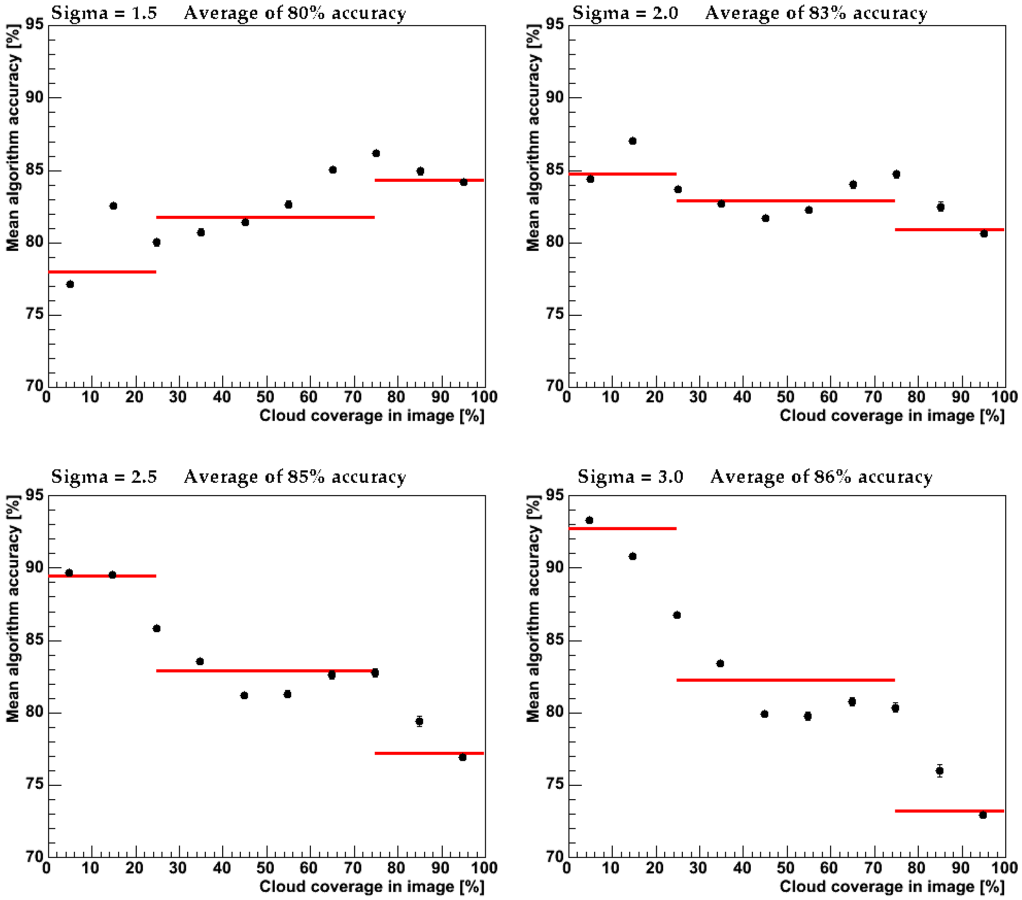


Figure 4.21: *Figure shows the accuracy of cloud detection for various threshold levels. Black marks indicate the mean accuracy within a 10% width bin. The red lines indicate the mean accuracy during clear (<25%), partially (25-75%) and overcast (>75%) cloud conditions.*

Figure 4.21 shows the accuracy of this algorithm with thresholds at various multiples of residual RMS (sigma σ) above the paramaterization. An overall mean accuracy of 80% at a threshold of 1.5σ improves with larger values of σ up until a mean accuracy of 86% at 3.0σ. Below 2.0σ this thresholding method is more effective on cloudy nights, but above at higher threshold levels, the algorithm is more effective on clear nights.

This improvement with higher threshold levels is likely due to a bias in the data set towards clear nights. With an increasing sigma threshold level, clear nights are more and more likely to be correctly identified and overcast ones less so. This is also evident in the shift in the accuracy in overcast conditions at different threshold levels - at 1.5σ there is nearly 85% accuracy, at 3.0σ this has dropped down to 76%. A shift in the opposite direction may be observed for clear night images. This apparent improvement with higher levels of σ is simply the result of a larger portion being of clear sky conditions, as a threshold increases the number of pixels classified as clear sky will increase, which naturally favors clearer night-time conditions.

A better measure of the algorithm's effectiveness is at whatever threshold level the response from clear, partially clouded and overcast cloud conditions is the same. If either clear or overcast conditions are being clearly favored by a particular threshold level then it implies that the algorithm's effectiveness is a result of some numerical threshold level, as opposed to the parameterization's ability to predict clear sky signal. In the case of this particular data set, a threshold 2σ best approximates a level response implying that the true effectiveness of this algorithm is approximately 80-83%.

4.3.2 Segmentation: Otsu's algorithm

Otsu's algorithm was discussed in subsection 4.2.1, and is a method for automatically choosing some threshold, based upon minimizing the inter-class variance between the two classes of pixels that would result at a particular threshold level.

Three applications of the algorithm are discussed here. In the first application, the algorithm is applied to the entire image - combining information from both pixels viewing the Earth and pixels viewing the night sky. This sometimes has the effect of missing fainter cloud in the night sky if the ground appears sufficiently bright, therefore the algorithm is also evaluated using just pixels from the sky (the top-most 180 rows in the image) in the second algorithm described here. The third algorithm simulates an estimated distribution of clear sky signals and adds it to the data being evaluated by Otsu's algorithm. This provides an improvement to the accuracy of Otsu's algorithm on completely overcast days, as the addition of a guaranteed clear sky distribution allows better assignment of a threshold in such conditions.

Otsu's algorithm applied to the whole image

Otsu's algorithm provides a method for estimating a threshold, based upon the between-class variance between different sets of pixels in an image. Pixels with a signal intensity above this threshold are classified as being cloud, and those below as being clear sky. A step-by-step guide to the algorithm is shown below.

1. Calculate the mean signal strength of all pixels in the image.
2. Divide the pixels into two groups according to some threshold level.
3. Calculate the mean signal strength and fraction of pixels in each individual group.
4. Calculate the between-class variance using equation 4.4.
5. Repeat steps 2-4 until all potential thresholds have been evaluated.
6. Pick the optimal threshold as that threshold which had the largest between-class variance.
7. Assign any pixels in the image with a signal strength above the optimal threshold as being cloud, and any below as clear sky.

Figure 4.22 shows the results of application of this algorithm to the test set of data. Accuracy of the algorithm is evaluated by determining the fraction of pixels that are correctly identified as cloudy or clear sky (according to the manually twice-checked test set). The histogram on the left suggests a generally high degree of success with a mean success rate of approximately 79%, with most of the images having more than 70% of their pixels correctly classified as being clear or cloudy. On the right is a plot describing the mean accuracy for images depicting different cloud conditions. For mostly clear images with less than 25% of the pixels being cloud the mean accuracy is nearly 92%. For partially cloudy conditions, with between 25% and 75% of the pixels being cloud, the algorithm performs slightly less well with a mean accuracy of 83%. During overcast conditions, with a greater than 75% cloud coverage, the mean accuracy is much poorer with a a mean accuracy of 55%.

This algorithm has an impressive accuracy during very clear conditions, but at increasingly overcast conditions it becomes much poorer. This due to the apparent brightness of the land within the image field of view.

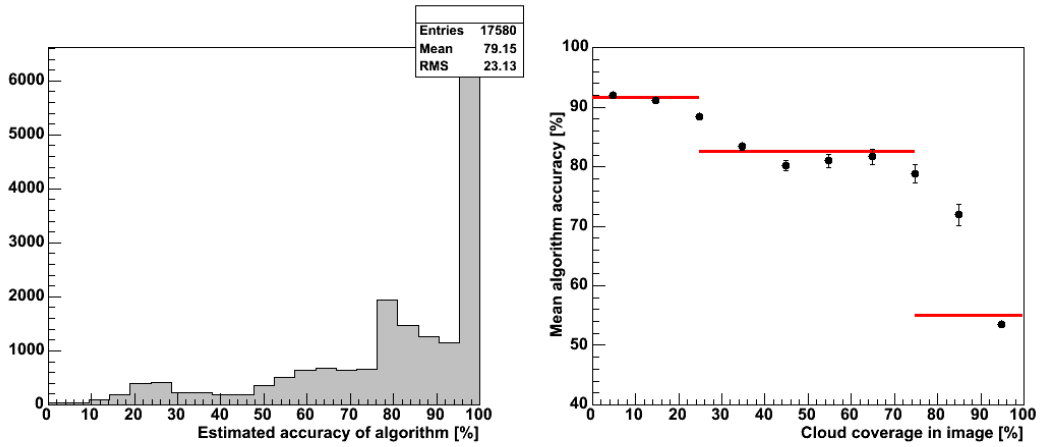


Figure 4.22: *Figure shows the accuracy in cloud detection of Otsu's algorithm being applied to images in the test set of data. **[Left]** Histogram of the algorithm's accuracy pertaining to the test set of 17580 cloud images. **[Right]** Plot showing the mean algorithm accuracy in different cloud conditions. Black marks indicate the mean accuracy within a 10% width bin with standard error on the mean. The red lines indicate the mean accuracy during clear (<25%), partially (25-75%) and overcast (>75%) cloud conditions.*

The high signal from the bright pixels associated with the landscape will generally contrast very strongly with the much dimmer clear night sky thereby allowing Otsu's algorithm to arrive at a successful threshold, due to the presence of two well defined signal distributions. Fainter cloud appearing in the image may be missed however, as there is greater similarity with the clear sky distribution than with the much brighter landscape.

As the fraction of cloud in the image increases the size of the 'bright' distribution of pixels in the image increases and there is less contrast between the cloud and clear sky signal distributions. This results in an increase in the derived threshold, that begins to start identifying the dimmer regions of cloud as clear sky. In completely overcast conditions there is no contrast available between the clear sky and cloud, therefore the threshold is set somewhere within the cloudy pixel distribution, and much the cloud is subsequently falsely classified as being clear sky and the accuracy of the algorithm drops.

The performance of Otsu's algorithm in a determining a threshold that successfully segments clear and cloudy regions is dependent upon the availability of information on both the signal intensity expected by clear sky and cloud. The presence of the landscape always simulates a bright cloud distribution, but without a clear sky distribution the threshold determined by Otus's algorithm may be an overestimate.

Otsu's algorithm applied to the sky

In this algorithm, Otus's algorithm as applied as normal, but this time only using the pixels that make up the sky region. The presence of the landscape within the image may sometimes cause a threshold to be set too high, failing to recognize the presence of fainter cloud in the night sky. By ignoring the landscape the algorithm will become more sensitive to the presence of such clouds, and set a correspondingly lower threshold.

1. Calculate the mean signal strength of pixels in the top 180 rows of the image.

2. Divide the pixels in the top 180 rows into two groups according to some threshold level.

3. Calculate the mean signal strength and fraction of these pixels in each individual group.

4. Calculate the between-class variance using equation 4.4.

5. Repeat steps 2-4 until all potential thresholds have been evaluated.

6. Pick the optimal threshold as that threshold which had the largest between-class variance.

7. Assign any pixels in the image with a signal strength above the optimal threshold as being cloud, and any below as clear sky.

Figure 4.23 shows the results of this algorithm applied to the test set of data. The histogram on the left demonstrates a low accuracy in correctly identifying pixels, with a mean accuracy of 57%, and most of the images having less than 80% of their pixels correctly identified. The right-most plot describes the mean accuracy for images under different cloud conditions. For mostly clear images with less than 25% of the pixels being cloud, the mean accuracy is 48%. For partially cloudy conditions, with between 25% and 75% of the pixels being cloud, the algorithm has a mean accuracy of 83%. During overcast conditions, with a greater than 75% cloud coverage, the mean accuracy is 66%.

The performance of this algorithm is very poor during very clear and overcast conditions. This is likely due to the lack of the availability of both clear and cloud intensity distributions under these conditions. Without

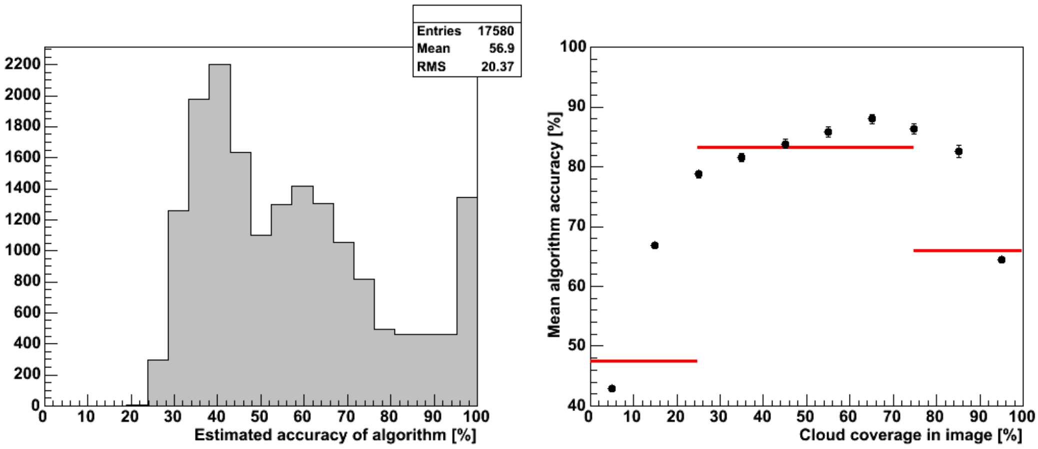


Figure 4.23: *Figure shows the accuracy in cloud detection of Otsu's algorithm being applied to the sky regions of images in the test set of data.* ***[Left]*** *Histogram of the algorithm's accuracy pertaining to the test set of 17580 cloud images.* ***[Right]*** *Plot showing the mean algorithm accuracy in different cloud conditions. Black marks indicate the mean accuracy within a 10% width bin with standard error on the mean. The red lines indicate the mean accuracy during clear (<25%), partially (25-75%) and overcast (>75%) cloud conditions.*

examples of both distributions, the threshold calculated by Otsu's algorithm will either be too low or high, resulting in a high error rate.

This algorithm performs more poorly than when it is applied using information from all the pixels in an image. This is because while the algorithm may be more sensitive to fainter cloud conditions, these cloud conditions are not common in actual cloud camera data. The algorithm performs at its best in mixed cloud coverage conditions, likely because it is here where both the clear sky and cloudy conditions are approximately equally represented.

Otsu's algorithm with false clear sky distribution

One of the drawbacks to the previous algorithms operating during very overcast conditions is the lack of a substantial distribution of signals associated with a clear night sky, with which to establish a good threshold against. The work done in section 4.2.3 has shown it is possible to estimate the clear sky signal through the use of fitted parameterizations. In this algorithm, a simulated distribution of 25600 clear sky signals is created from these parameterizations, and is added to the distribution of signals from a particular image before Otsu's algorithm calculates a threshold. This ensures that at least a quarter of the signals being evaluated by the algorithm always represent a clear sky signal

1. For several months worth of cloud camera data (from the same camera and with the same flat-field calibration) calculate the mean signal strength and RMS in each image, for each elevation between 3-29°.

2. Record the average fraction of cloud measured by the LIDARs when each cloud camera image was taken.

3. Using LIDAR cloud fraction, RMS and signal strength cuts determine which mean signal strengths are from a cloud-free night sky.

4. For each elevation fit a parameterization of the form described in section 4.2.3, and record the RMS of each fit.

5. Randomly pick one elevation, and evaluate the parameterization to yield the clear sky signal for the image in question.

6. Randomly draw a number from a Gaussian distribution with an RMS and a mean equivalent to the parameterization picked in step 5.

7. Repeat steps 5-6, 25600 times.

8. Calculate the mean signal strength of all the image pixels and randomly generated clear sky signals.

9. Divide all the pixel and random signals into two groups according to some threshold level.

10. Calculate the mean signal strength and fraction of signals in each individual group.

11. Calculate the between-class variance using equation 4.4.

12. Repeat steps 9-11 until all potential thresholds have been evaluated.

13. Pick the optimal threshold as that threshold which had the largest between-class variance.

14. Assign any pixels in the image with a signal strength above the optimal threshold as being cloud, and any below as clear sky.

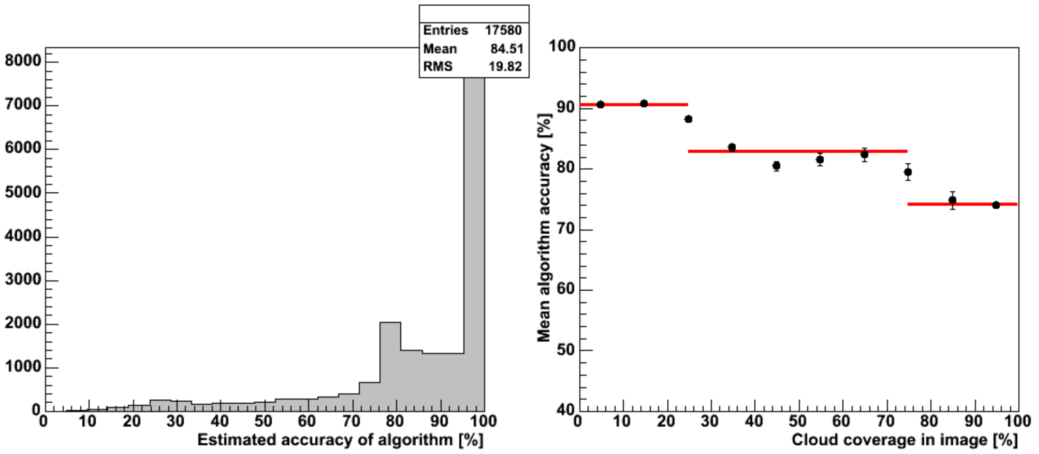


Figure 4.24: *Figure shows the accuracy in cloud detection of Otsu's algorithm (with a false clear sky distribution) being applied to the sky regions of images in the test set of 17580 cloud images.* ***[Left]*** *Histogram of the algorithm's accuracy pertaining to the test set of data.* ***[Right]*** *Plot showing the mean algorithm accuracy in different cloud conditions. Black marks indicate the mean accuracy within a 10% width bin with standard error on the mean. The red lines indicate the mean accuracy during clear (<25%), partially (25-75%) and overcast (>75%) cloud conditions.*

Figure 4.24 demonstrates the cloud/clear sky classification accuracy through the use of Otsu's algorithm with an added fake clear sky distribution of signals. The overall algorithm efficiency is nearly 85%, with it working most effectively under clear sky conditions with a mean accuracy of 90%. Under partially cloudy and overcast cloud conditions the algorithm functions less effectively, with mean accuracies of 83% and 74% respectively.

This algorithm showed an increase of nearly 6% in the overall accuracy from the approach where Otsu's algorithm was just applied to image data, indicating that this approach is superior. The mean accuracy of during clear sky and partially cloudy conditions are very similar to that of the plain application of Otsu's algorithm, the mean accuracy during overcast conditions, however, show a substantial improvement of approximately 20%. Artificially adding a clear sky distribution of pixels successfully increased the ability of Otsu's algorithm to pick an appropriate threshold level during overcast cloud conditions, where there would otherwise be no such distribution available to provide a substantial contrast against the clouds.

A higher degree of accuracy was likely not achieved due to the width of the false clear sky signal distribution created. Imprecision in the fitted parameterization (because it is an approximation) leads to the false clear sky signals being drawn from a greater range of values than would be seen in reality. Fainter, higher altitude cloud, will therefore sometimes have signal distributions similar to that of the broad simulated distribution. This reduces the effectiveness of the fake clear sky signals as contrasts, and thus the threshold derived by Otsu's algorithm.

4.3.3 Segmentation: K-means cluster analysis

The K-means algorithm may sort information into different clusters based upon several different variables. In this application, however, the only sorting variable being considered is the signal intensity of each pixel. Applying the K-means algorithms allows an image to be divided into several different partitions, each of which may be assigned as being either clear or cloud sky. In these cloud detection algorithms, the K-means algorithm is used to segment images into three different partitions. The number three was chosen as there are typically three distributions of pixels within an image belonging to the landscape/bright cloud, faint cloud and clear night sky.

Initial seeding of the clusters is carried out randomly, therefore the final assignment of pixels to different partitions may vary if the algorithm is applied to the same image multiple times. In order to account for this, each image is run through the algorithm several times, and the final partition arrangement chosen that has the least variance between pixels and their partition means.

Images cannot always be segmented into three partitions (in the test set of data 47% of images cannot be segmented this way). This occurs in very featureless images, such as very low altitude overcast cloud conditions or very clear nights. If such an image is encountered in these algorithms, a simple Otsu threshold is calculated and applied to the image to segment it into cloudy and clear sky regions, as the failure to segment implies only one or two strong distributions of pixels within the image.

Three partitions, favoring cloud

In this algorithm, the K-means algorithm is used to segment images into 3 partitions. The two partitions with the highest mean signal strength are classified as being cloud (this includes the landscape), while the partition with the lowest mean signal strength is classified as being clear sky. Images are processed by the K-means algorithm several times, to find the partition structure which minimizes the variance between the mean signal of a partition and its constituent pixels, as this is taken to be the 'best-fit' case. Images that cannot be segmented in this way, have Otsu's algorithm applied to them to determine some threshold, above which pixels are considered to be cloudy.

1. Randomly assign each pixel in the image to one of three partitions.
2. Calculate the mean signal strength of the pixels in each partition.
3. Calculate the variance between each pixel's signal and the mean signal of each partition.
4. Reassign each pixel to the partition with which it has the least variance.
5. Repeat steps 3-5 until no pixels are being reassigned.
6. Check that all three partitions have members, if not go back to step 1.
7. Calculate the mean pixel intensity of each partition.
8. Calculate the total variance between each pixel's intensity and its corresponding cluster's mean intensity.
9. Repeat steps 1-8, 10 times.
10. If cannot partition this image into three clusters, apply the algorithm in section 4.3.2 and end.
11. Use the partition with the lowest variance, calculated in step 8.
12. Assign the partition with the lowest mean intensity as clear sky, and the two highest as cloud.

Figure 4.25 shows the effectiveness of this K-means segmentation algorithm when it's applied to the test set of data. Only 9246 of the images were successfully segmented into three regions, the rest were segemented using Otsu's algorithm (see section 4.3.2). The overall mean effectiveness of the algorithm was approximately 71%. During clear sky and partially cloudy conditions, the algorithm performed similarly with a mean accuracy of around 77%. During overcast cloud conditions however, the mean effectiveness dropped to approximately 60%.

The drop in accuracy during overcast cloud conditions is likely due to the algorithm's assumption that there is always a clear sky distribution of signals within the image. The partition with the lowest mean intensity is always being assigned as clear sky, in the case of overcast conditions (where there is not always a clear sky signal distribution) this will result in an erroneous classification of this partition. The effectiveness of the algorithm

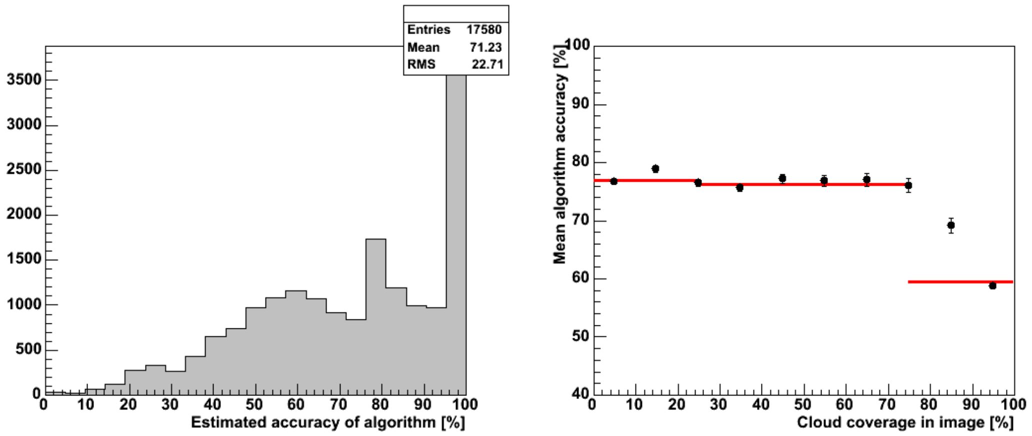


Figure 4.25: *Figure shows the accuracy in cloud detection using the K-means segmentation algorithm (favoring cloud) being applied to images in the test set of 17580 cloud images.* ***[Left]*** *Histogram of the algorithm's accuracy pertaining to the test set of data.* ***[Right]*** *Plot showing the mean algorithm accuracy in different cloud conditions. Black marks indicate the mean accuracy within a 10% width bin with standard error on the mean. The red lines indicate the mean accuracy during clear (<25%), partially (25-75%) and overcast (>75%) cloud conditions.*

during less cloudy nights is conversely negatively affected by the presumption that there are always two signal distributions associated with cloud. On completely clear nights there is generally only two distributions, one from the landscape and one from the clear sky itself. If the K-means algorithm successfully segments the image, it is possible that two of the partitions will be associated with the clear sky. Since two partitions are guaranteed to be classified as cloud, then there will always be a mis-classification of clear sky pixels as cloudy pixels in this situation.

Three partitions: favoring clear sky

Once again, much like the previous algorithm, an image is segmented into three different partitions. In this case, however, only the partition with the highest mean signal strength is classified as being cloud. The other two with lower mean signal intensities are classified as being clear night sky.

1. Randomly assign each pixel in the image to one of three partitions.

2. Calculate the mean signal strength of the pixels in each partition.

3. Calculate the variance between each pixel's signal and the mean signal of each partition.
4. Reassign each pixel to the partition with which it has the least variance.
5. Repeat steps 3-5 until no pixels are being reassigned.
6. Check that all three partitions have members, if not go back to step 1.
7. Calculate the mean pixel intensity of each partition.
8. Calculate the total variance between each pixel's intensity and its corresponding cluster's mean intensity.
9. Repeat steps 1-8, 10 times.
10. If cannot partition this image into three clusters, apply the algorithm in section 4.3.2 and end.
11. Use the partition with the lowest variance, calculated in step 8.
12. Assign the partition with the highest mean intensity as cloud, and the two lowest as clear sky.

Figure 4.26 shows the accuracy of this K-means segmentation algorithm when its applied to the test set of data. As with the previous algorithm, those images not segmented into three regions were segmented using Otsu's algorithm (see section 4.3.2). The overall mean effectiveness of the algorithm was approximately 77%. During clear sky conditions, the algorithm performed well with a mean accuracy of around 92%, and poorly during overcast cloud conditions with a mean accuracy of 58%.

This clear sky favoring algorithm performs very similarly to the cloud favoring algorithm in mixed cloud conditions, but performs much ($\approx$ 10%) better and much worse in clear and overcast conditions respectively. The improvement in clear sky conditions comes about because there is generally more spread in the signals associated with clear sky signals, than with the signals registered for the landscape. Therefore a segmentation into three partitions is likely to have two associated with the clear sky signals and one with the landscape, and thus by assigning more partitions as clear sky the image is more likely to have clear sky distributions correctly classified. The accuracy for overcast cloud conditions gets worse, as in such conditions there is little if any clear sky distribution present in the image. Therefore all three partitions are likely to consist of cloudy sky or landscape distributions, and by classifying more partitions as clear sky, the fraction of pixels incorrectly identified as clear sky will increase.

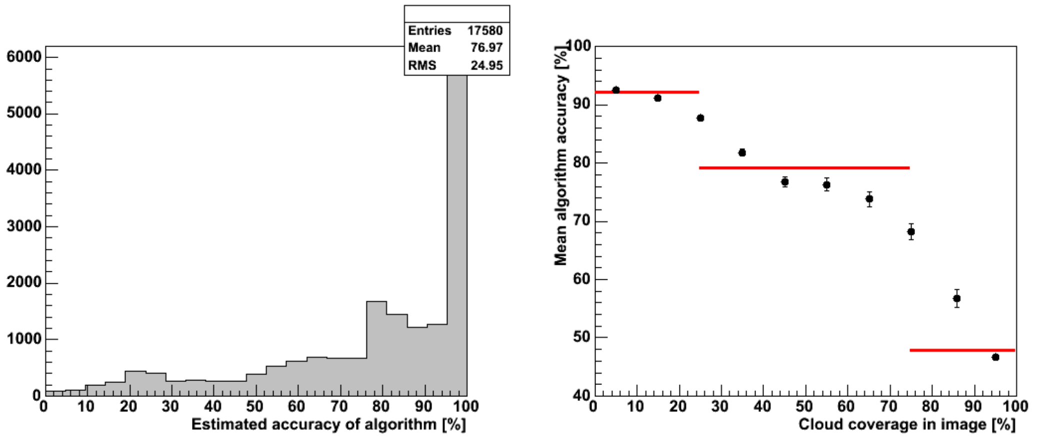


Figure 4.26: *Figure shows the accuracy in cloud detection using the K-means segmentation algorithm (favoring clear sky) being applied to images in the test set of 17580 cloud images.* ***[Left]*** *Histogram of the algorithm's accuracy pertaining to the test set of data.* ***[Right]*** *Plot showing the mean algorithm accuracy in different cloud conditions. Black marks indicate the mean accuracy within a 10% width bin with standard error on the mean. The red lines indicate the mean accuracy during clear (<25%), partially (25-75%) and overcast (>75%) cloud conditions.*

4.3.4 Segmentation: Edge based

In this algorithm, Canny's edge detection algorithm is used in conjunction with Otsu's thresholding algorithm to detect cloud through their boundary with clear night sky. Canny's edge detection algorithm is used to locate and define the position of edges within the image. These edge contours initially have a width of one pixel, but are expanded into larger 'edge regions' by assigning any neighbouring pixels as being edges too. Each 'edge region' is then thresholded using Otsu's algorithm, to identify which side of the original edge contours are cloud, and which are clear sky. This last step is demonstrated in the right-most image in figure 4.27, where white and black are cloudy and clear sky pixels (identified through Otsu's threshold algorithm) respectively. Grey pixels are pixels that do not belong to any 'edge regions', and are later classified (seen in the bottom image in figure 4.27) depending upon the surrounding edge region pixels.

1. Smooth the selected image with a gaussian weighted averaging filter.

2. Work out the signal gradient and direction across each pixel in the filtered image using a Sobel filter.

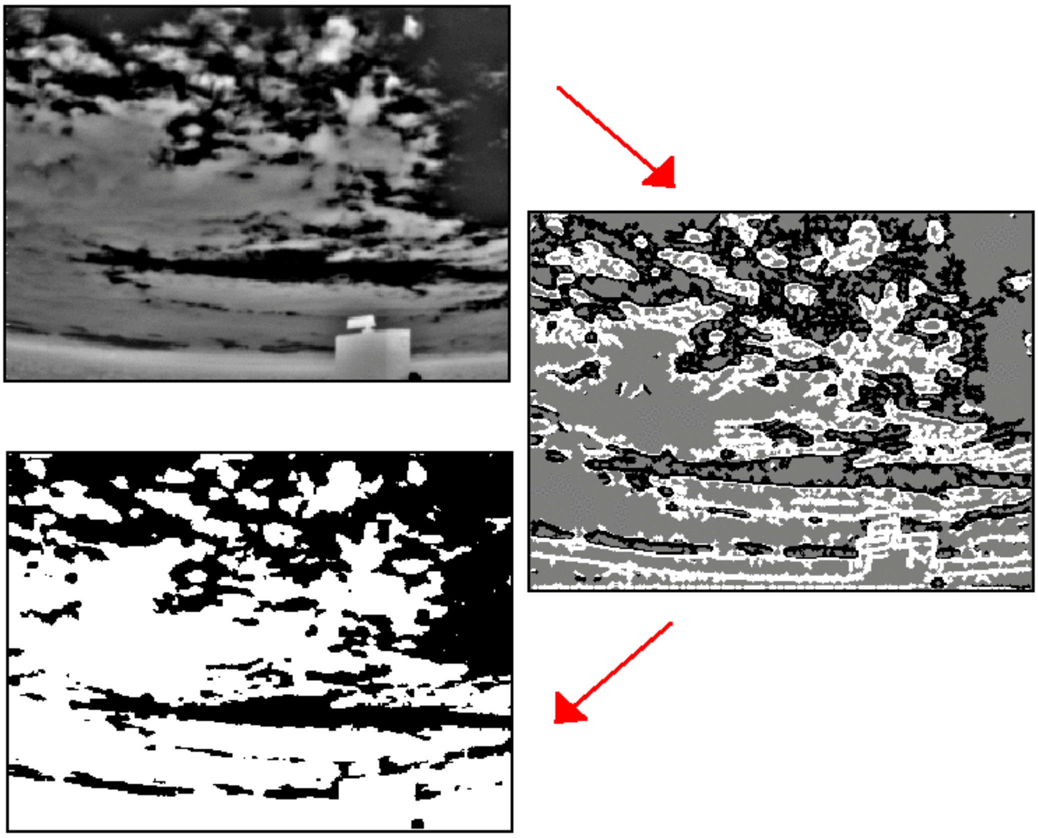


Figure 4.27: *An overview of the edge-based thresholding process. The raw top-most image has Canny's edge finding algorithm applied to it, locating regions of pixels around the boundary between cloud and clear sky in the image. Each identified edge region is then thresholded, as is seen in the right-most image. Grey indicates pixels not part of an edge region, and white and black indicate pixels from the 'edge regions' that have been identified as cloud or clear sky respectively by Otsu's thresholding algorithm. For each region of grey pixels, the number of cloudy and clear pixels along its boundaries are evaluated and used to classify the region as either cloud or clear sky.*

3. Use non-maxima supression and hysteresis (see section 4.1.1) to locate edges within the image.

4. Expand any edges identified into ‘edge regions’ by marking every non-edge pixel adjacent to an edge pixel as an edge pixel as well.

5. For each ‘edge region’, apply Otsu’s algorithm to determine a signal intensity threshold in that region.

6. Segment each individual ‘edge region’ based upon the calculated threshold (see the right-most image in figure 4.27).

7. For each region of non-edge pixels, calculate the fraction of edge pixels surrounding that region that have been indentified as cloud.

8. If there are more pixels identified as cloud surrounding a non-edge region, than as clear sky, then assign that entire region as being cloud, or vice versa.

This algorithm requires four different user-defined variable to function correctly: the level of Gaussian filtering, and the upper and lower thresholds for the non-maxima suppression, the hysteresis process and the creation of the edge regions. This gives the algorithm a great deal of flexibility in its response to different types of images. For the purposes of evaluation, a cloudy image was chosen at random from the test set of data and a set of variables chosen such that this algorithm correctly identified all the cloud in that image. These same variables were then used to evaluate the effectiveness of this algorithm for the entire test set of data, the results of which may be seen in figure 4.28.

Figure 4.28 shows the performance of the edge-based thresholding algorithm when applied to the test set of data. The algorithms had a mean accuracy of 58% on the test data, but most of the images were either very successful (>95%) or very poor (<5%) is terms of cloud detection. During overcast cloud conditions, the algorithm performed most poorly, with an average detection accuracy of 40%. In clear, or partially cloudy conditions however, the mean accuracy was around 65%.

A large fraction of the images that underwent this algorithm either had most of their cloud detected accurately, or most of their cloud missed - as seen in the histogram in figure 4.28. As there are some four different variables in the algorithm (Gaussian filtering level, lower and upper threshold and the size of the edge regions), and only one set of variables were used in this study, then this could be interpreted to mean that either the algorithm will perform very well, or very poorly. Those images that were well suited to the settings

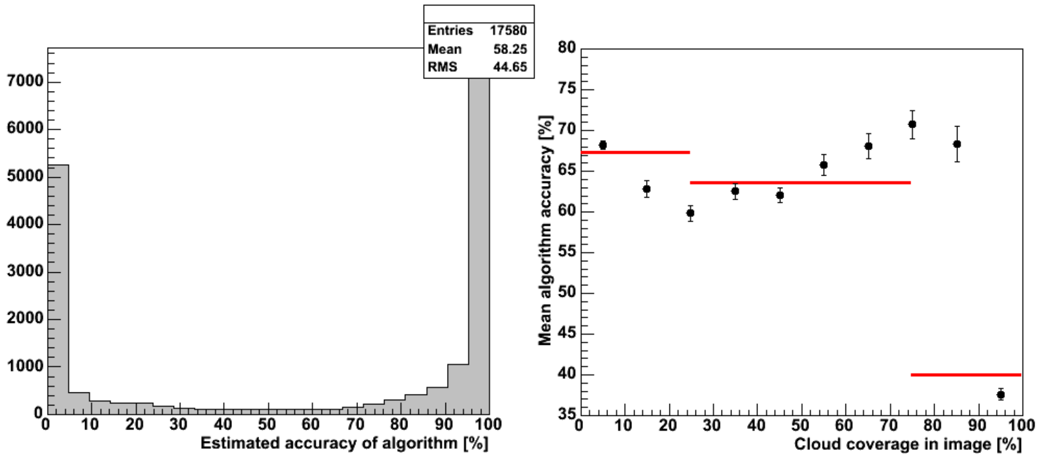


Figure 4.28: *Figure shows the accuracy in cloud detection using the edge detection based segmentation algorithm (favoring clear sky) being applied to images in the test set of 17580 cloud images. [**Left**] Histogram of the algorithm's accuracy pertaining to the test set of data. [**Right**] Plot showing the mean algorithm accuracy in different cloud conditions. Black marks indicate the mean accuracy within a 10% width bin with standard error on the mean. The red lines indicate the mean accuracy during clear (<25%), partially (25-75%) and overcast (>75%) cloud conditions.*

of the study had a higher fraction of their cloud identified successfully, while those images to which the settings were poorly suited, had very little of their cloud identified correctly. Only a relatively small fraction of images had an intermediate level of success between 10-90%.

This suggests that this algorithm has the potential to be very effective in successfully describing cloud conditions within the cloud camera images. The catch is, however, that a user must first identify the right variables for the algorithm to work with a particular image. A poor choice of thresholds and filtering levels may be too sensitive to noise within the image and register it as cloud, or may not be sensitive enough, and incompletely define the shape of any cloud present. Despite the potentially very good results offered by the algorithm, its potential for extremely poor results makes it unreliable for unsupervised processing of large quantities of images.

Another interesting result is the poor cloud detection (40%) provided by the algorithm during overcast cloud conditions. This is likely due to the lack of strong edge gradients in such conditions. If the sky is completely overcast, and the landscape at a similar intensity to the cloud, then there may be no areas of contrast over which Canny's edge detection algorithm may detect an edge, and so hence detect any cloud. In the cases where appropriate edge gradients do exist, then lower intensity cloud regions are going to be incorrectly identifed as clear sky. Clearly this algorithm is not robust under such conditions.

4.3.5 Summary

The effectiveness of the cloud detection algorithms upon the test sample of data is summarized within Table 4.1. Each algorithm has listed beside it its effectiveness in clear sky, partially cloudy and overcast conditions. Also noted is the overall mean success of the algorithm for the test set of data. To aid the eye, those values above 80% are marked in blue, while those below are marked in red. Values above 90% are additionally displayed in a bold text font.

Algorithm	**A**	**B**	**C**	**Overall**
Weather dependent threshold	85%	83%	81%	83%
Otsu's algorithm (whole sky)	**92%**	83%	55%	79%
Otsu's algorithm (part of sky)	48%	84%	66%	57%
Otsu's algorithm (false clear sky)	**90%**	83%	74%	85%
K-means cluster analysis (cloud bias)	77%	77%	59%	71%
K-means cluster analysis (clear sky bias)	**92%**	79%	48%	77%
Edge detection	67%	64%	40%	58%

Table 4.1: *Cloud detection accuracy of algorithms upon a test set of images. A, B and C represent the accuracy in clear, partially clouded and overcast conditions respectively. 'Overall' represents the mean accuracy of the algorithm for all the images in the test set. An algorithm accuracy above 80% is represented by the color blue, while any values below are marked red.*

Generally speaking, as shown within Table 4.1, the strongest performing algorithms were those utilizing the weather dependent information - the weather dependent threshold and Otsu's thresholding algorithm with the false clear sky distribution. Both of these algorithms achieved greater than 80% effectiveness in most of the different cloud conditions. The only algorithm with a 80% cloud detection effectiveness during overcast conditions was also the weather dependent threshold. The poorest performing algorithms, with no values above 80%, were the edge detection and K-means (cloud bias) algorithms.

An interesting aspect to Table 4.1 is the generally poor response of the cloud detection algorithms during completely overcast cloud conditions.

With the exception of Otsu's (false clear sky) algorithm, all of the processes were successful in identifying clouds and clear sky 49-66% of the time. As the pixels are either viewing the landscape or overcast night sky, there are no pixels with associated clear sky signals that the cloudy pixels may be contrasted against. With the exception of the weather dependent threshold and Otsu's (false clear sky) algorithms (which happen to be the most effective algorithms), all of the detection algorithms require some sort of contrast within the image. In overcast conditions, with its absence, the result is lower intensity cloud being erroneousy identified as clear night sky.

This highlights the necessity for examples of both cloudy and clear sky pixels within cloud camera images for the effective application of cloud detection algorithms, however the situation is not as severe in the case of clear night sky conditions. This is evidenced by the majority of the algorithms achieving detection successes in excess of 80%. Despite the lack of cloud in these conditions, the landscape of the Earth is present and able to provide a contrast in intensities due to it usually bright intensity with respect to the clear night sky. This point is further illustrated by the Otsu's (part of sky) algorithm, where Otsu's threshold was calculated just using pixels viewing the night sky. Without the landscape present the clear night sky was partially segmented into cloud.

The predictive nature of the clear sky parameterization discussed in sections 4.2.3 and 4.3.1 is clearly invaluable to the successful detection of cloud in the cloud camera images, as it ensures that at least an estimation of the clear sky brightness is always available. Two applications of this parameterization were examined in this study - the weather dependent threshold and Otsu's (false clear sky) algorithm. Of the two algorithms Otsu's was, on average, slightly more effective, with a mean accuracy of 85% - performing better than the raw Otsu's (whole sky) algorithm's approach, due to its increased (but still below 80%) success in overcast cloud conditions. The weather dependent threshold had a mean accuracy of 83%, but a more consistent response across a variety of cloud conditions.

Despite its slightly inferior on-average performance, the weather dependent threshold is probably the superior choice of the two clear sky parameterization algorithms. Neither algorithm provides a consistent 90% accuracy in their cloud detection, therefore neither algorithm may be completely relied upon to successfully describe cloud conditions. So the choice of algorithm falls to the one that requires the least manual intervention on the part of a human operator. The weather dependent threshold requires the choice of some threshold (in terms of σ-variation from the fitted parameterization), giving it a great deal more flexibility than the more rigid Otsu's (false clear sky) algorithm. A human user encountering an image which has been unsuccessfully processed by the latter algorithm will have to spend time picking another algorithm, or some threshold level for the weather dependent threshold. In the former case however, even if unsuccessful, a human user will know what settings were used in the weather dependent threshold and thus be able to pick a more successful thresholding level quite easily. Therefore of the two, the weather dependent threshold is more desirable for large scale processing applications.

4.4 Pierre Auger Cloud Manager (PACMan)

The PACMan (**P**ierre **A**uger **C**loud **Man**ager) GUI[1] was created to facilitate processing of the large numbers of images recorded by the cloud cameras. PACMan uses files with the .irp filename extension which contain (in a compressed format) images, along with most of the information required to detect cloud in those images. Several options are available within PACMan that can not only initially create files of the .irp format, but also process images contained within and convert them into a format suitable for uploading into the cloud camera database. Manual supervision is still required for much of the processing procedure, however the graphical nature of PACMan is designed to make this process as painless as possible.

4.4.1 The .irp file format

The PACMan program stores images from the cloud cameras along with relevant information in files with an .irp filename extension. These files consist of a header containing information such as image pointing direction and image recording time, followed by the actual stored image data itself.

File structure

The header of a .irp file consists of several integer and character arrays stored in binary format, that not only provide generic information about the images contained within, but also the location of different images within the file. In order from the start of an .irp file, these arrays take the form:

[1] Graphical User Interface

File id	Identifies the file as being of .irp type	char[3]="irp"
Image num.	Total number (N) of raw images stored in file.	unsigned int
Eye	Which cloud camera the images belong to.	unsigned char
Type	What number in a set an image is from.	unsigned char[N]
Group	What group an image is assigned to.	unsigned char[N]
Position	Position of an image within the file.	unsigned long int[N]
Time	Time in GPS seconds of image capture.	unsigned long int[N]
Elevation	Elevation of the image pointing direction.	double[N]
Azimuth	Azimuth of the image pointing direction.	double[N]

After this initial header information, information describing each raw image is added sequentially to the file along with a 'processed' version of that image. Each image is stored via a series of arrays described below. The actual use of the arrays is described in the following sections regarding image compression.

K	Length of 'key chain' array.	unsigned short int
I	Length of 'keys' array.	unsigned short int
Processed data		unsigned char[9600]
Raw data		unsigned char[76800]
Key chain		unsigned short int[K]
Keys		unsigned short int[I]

Compression of processed data

Processed data is simply a binary white/black image indicating which pixels within an image are cloud and which are of a clear night sky. The most obvious method for compression is to represent each pixel in an image as a single bit (either a 0 or 1). There are 8 bits in 1 byte, therefore it is possible to represent 8 pixels of a processed image with a single char variable, and thus the 76800 pixels of a cloud camera image with 9600 bytes (9600 char variables).

Pixels are read from the processed image starting at its top left corner and moving right until the end of the row is reached, at which point pixels start being read on the left-hand side again one row below. If these pixels are considered in blocks of 8, then they may be represented by a single number between 0 and 255 by converting from binary i.e.

$$\begin{array}{|c|c|c|c|c|c|c|c|}\hline a & b & c & d & e & f & g & h \\ \hline\end{array} \quad = \quad 128a + 64b + 32c + 16d + 8e + 4f + 2g + h$$

where a, b, c, d, e, f, g and $h = 1$ if that particular pixel is cloudy, or 0 if it is of clear night sky. e.g.

$$\begin{array}{|c|c|c|c|c|c|c|c|}\hline 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ \hline \end{array} \quad = \quad 202$$

Each of these numbers may be stored via a char variable - allowing a the entire processed image to be stored using only 9.6 KB of hard disk.

Compression of raw data

Lossless compression of raw cloud camera images within the PACMan framework is achieved through representation of the signal for each pixel by a single 1 byte char variable, as opposed to the 2 byte unsigned short integer required to record each pixel's data ordinarily. This compression is a form of "delta compression", which stores pixel signals as the difference between it and its neighbour. Much like the case of the processed image data in the previous section, the image is considered as a one dimensional array of numbers starting at the top left corner of the image moving to the right, and starting on the left-hand side one row below upon reaching the right-hand side of the image.

1. The difference Δ between the signal recorded by a pixel and the signal recorded by the previous pixel is calculated for all (except the very first) pixels in the image.

2. The 255 most frequently appearing values of Δ are assigned a number between 0 and 254.

3. A new pixel array ('Raw Data') is created which has had its original pixel values substituted for values between 0 and 254 (from step 2), and 255 for any pixel values which were not assigned a number in step 2. The very first pixel in the image is always assigned 255.

4. Other one-dimensional arrays - the 'keys' and 'key chains' arrays of unsigned short int variables are created. The 255 entries in the 'key chain' array correspond to the signals identified in step 2. And entries within the 'keys' array correspond to the signal of pixels marked as 255 in step 3.

The hard disk space required to store a raw cloud camera image uncompressed is 153.6 KB. With this compression algorithm, the hard disk requirement becomes:

$$[\text{storage}] = 76.8 + \frac{255 + k}{500} \text{ KB}$$

where k is the number of pixels marked as 255. The extent of compression is dependent upon how many of the pixels within an image have an identical Δ value. Note that the effectiveness of this compression algorithm depends upon how many identical signals are recorded within a particular image. Typically, this algorithm reduces the storage requirement for a raw cloud camera image to 55-65% of the original requirement.

Decompression using this algorithm involves using the Raw Data array and subsituting any of the entries marked 255 contained therein as the value stored in the 'keys' array e.g. the tenth entry of Raw Data marked as 255 is equated as the tenth entry in the 'keys' array. Those 'Raw Data' entries marked as a non-255 value have their Δ values looked up using the 'Key Chain' array, which is then added to the value of the previous entry in the 'Raw Data' array to calculate the actual original value of that pixel.

4.4.2 Usage

The PACMan GUI is a program used to modify and access information stored in .irp files. Not only can the raw and processed cloud camera images within .irp be accessed and viewed, but options exist to sort and process them. PACMan is a tool to aid in the processing of large quantity of data recorded monthly by all four of the cloud cameras at the Pierre Auger Observatory.

Creation of the .irp file format

Files with the .irp extension readable by PACMan are created from the raw data recorded by the cloud cameras. Cloud cameras provide a single file (of .irb format) for each night's observations. Multiple .irb files are combined to form a single .irp file, usually containing a record of a month's data from a single eye. PACMan extracts and stores within the .irp file format the raw image data, the time images were taken, the pointing direction of the images, and which eye the images were recorded from. Creation of an .irp file requires users to provide a list of the .irb files to be combined (in the form of an ASCII text file) and to then follow various prompts by PACMan.

The .irb data contains both images captured in the field-of-view scans and the full sky mosaic scans. Usually just images captured in the field-of-view scans are added to any .irp files being created, but the option exists to use mosaics images as well. A synthetic set of field-of-view images may be created from several of the images in the mosaic image set. This is achieved by calculating the pointing direction of all of the pixels of the images of the mosaic image set and comparing them with the average pointing direction of the pixels in the field-of-view image set. For each of the pixels in

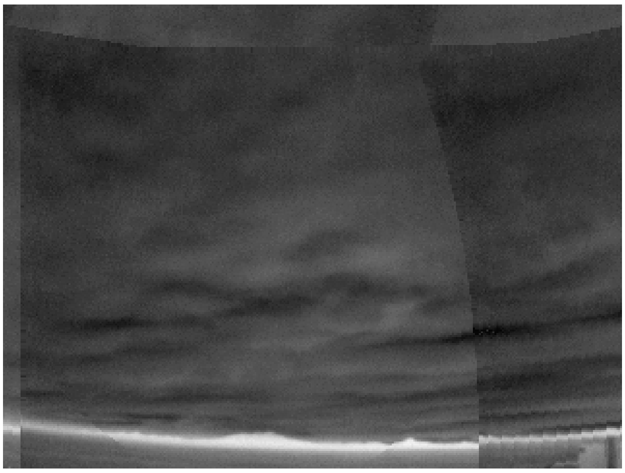


Figure 4.29: *A 'field-of-view' image created from a mosaic image set.*

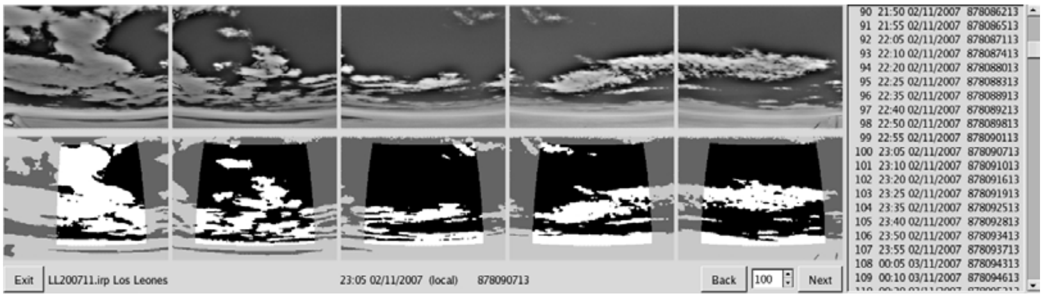
the average field-of-view image set those pixels within the mosaic set within around 0.2 degrees of that pixel are averaged to give pixel values for the mosaic field-of-view image set. Such image sets sometimes have apparent image discontinuities (as seen in figure 4.29) located at the boundaries of those mosaic images that were used, which may sometimes complicate image processing.

A single month's data from a single eye typically requires 350-550 Mb of storage space in the .irp style format.

Viewing images

PACMan displays images through use of the GTK C++ graphics library. Images within the file may be inspected on a set-by-set or individual basis to evaluate the effectiveness of the application of a particular processing algorithm.

Within figure 4.30 is an example of an image set being displayed by PACMan. The five raw images are at the top, while their processed counterparts (showing the pixels identified as cloud as being light grey/white) are just below. The areas of highest contrast in the image are those parts of the images that are *estimated* to be within the fluorescence detector's field-of-view, but not overlapping any of the neighbouring images (it is not precise, and is only meant to be taken as a rough guide!). On the right-hand side is a scrollable list showing all of the image sets contained in the .irp file, and allowing a user to easily find one in particular. This viewing format allows an operator to quickly assess the effectiveness of the application of a particular cloud detection algorithm, and determine an alternative strategy if necessary.



```
LL200711.irp Los Leones          23:05 02/11/2007  (local)     878090713
```

```
 90  21:50 02/11/2007  878086213
 91  21:55 02/11/2007  878086513
 92  22:05 02/11/2007  878087113
 93  22:10 02/11/2007  878087413
 94  22:20 02/11/2007  878088013
 95  22:25 02/11/2007  878088313
 96  22:35 02/11/2007  878088913
 97  22:40 02/11/2007  878089213
 98  22:50 02/11/2007  878089813
 99  22:55 02/11/2007  878090113
100  23:05 02/11/2007  878090713
101  23:10 02/11/2007  878091013
102  23:20 02/11/2007  878091613
103  23:25 02/11/2007  878091913
104  23:35 02/11/2007  878092513
105  23:40 02/11/2007  878092813
106  23:50 02/11/2007  878093413
107  23:55 02/11/2007  878093713
108  00:05 03/11/2007  878094313
109  00:10 03/11/2007  878094613
```

Figure 4.30: *A full field-of-view set of images stored in .irp file format, being displayed by PACMan.*

Processing images

The processing of the cloud camera images refers to the application of a particular cloud detection algorithm (discussed in the previous sections) to those images. Because of its adaptability, and reliability in all types of cloud conditions, the weather dependent threshold has been chosen as the favored algorithm.

Processing of a set of images is therefore carried out through several phases. First of all, a set of clear sky parameterizations (at different elevations) must be created, through the fitting of suspected clear sky signals with the measured ambient temperature and ground level water vapor pressure. A simple system for the selection of such data is detailed in section 4.2.3.

The value of these parameterizations at the time of a particular image recording is then calculated (using ambient temperature and ground level water vapor pressure measurements made by the Pierre Auger weather stations) to estimate the clear sky brightness at different elevations. A threshold is then calculated as some multiple of the residual RMS of that fit above the predicted clear sky brightness level for each image. Pixels at elevations greater (or below) than the highest elevation parameterization use the closest parameterization available. Pixels above the threshold are taken as cloud, and those below as clear sky. PACMan will perform this process automatically, for a user supplied set of parameterizations.

The user then quickly scans the set of images using PACMan (seen in figure 4.30), determining which sets of images have had their cloud successfully identified. PACMan may then be used to adjust the thresholding level of different sets of images to more optimal levels if necessary. Images recorded at similar times will tend to respond best to the same level of thresholding, therefore it is rarely necessary to adjust each individual set of images, rather, several hours worth of images may be corrected by a single adjustment at a time.

The lack of a single reliable cloud detection algorithm means that manual scanning by a human is required to successfully process the data. PACMan provides an easy-to-use interface, that provides a relatively simple way of processing large numbers of cloud camera images that minimizes the hassle associated with this process.

When an operator is satisfied that all of the cloud within an .irp file has been identified, they may then use PACMan to convert the cloud camera images into a format compatible with the Auger fluorescence detectors. This format provides an interpretation of the cloud camera data which estimates the level of cloud within the field of view of each cloud camera pixel.

4.5 Cloud camera database

The cloud camera database is one of the atmospheric databases maintained by the Pierre Auger observatory. Cloud camera information is stored via a cloud index calculated for each individual FD (fluorescence detector) pixel. Due to the quantity of data recorded by the cloud cameras all collected information is compressed before being uploaded into the database.

4.5.1 Format

Cloud camera images, in their raw processed format, do not provide much information about the cloud conditions that may be affecting a particular shower. This is because there is no explicit correlation between the regions of sky being viewed by the fluorescence detector, and the regions of sky recorded in the image. In addition the quantity of cloud camera images available, makes storage of this information in its native format expensive. Therefore a less storage-demanding and more useful format for the processed cloud camera images has been developed.

Work done by Dr Andrew Smith has provided an estimate (after correction of camera lens aberration effects) of the pointing direction of each pixel in a particular cloud camera image, based upon the overall pointing direction of the camera. The pointing direction of each of the fluorescence detector's pixels has been well measured in the past. Because the cloud cameras are located very close to the detectors (they're mounted on the roof), the pointing direction of the fluorescence detector pixels may be compared directly with the pointing direction of pixels in the cloud camera.

The full viewing angular width of the fluorescence detector pixels is approximately 1.5°. Any cloud camera pixels with a pointing angle within 0.75° of the pointing direction of a pixel (for its associated fluorescence detector) are therefore taken to be representative of the cloud conditions in that pixel's field of view. The amount of cloud within that FD pixel is simply taken as the fraction of those representative cloud camera pixels that have been found to be viewing cloud. This cloud fraction is then further simplified into a cloud index using the following table.

Fraction of cloud (%)	Cloud Index
0-10	0
10-30	1
30-50	2
50-70	3
70-90	4
90-100	5

A full set of 5 cloud camera images are required to calculate the cloud index for each of a fluorescence detector's 2640 pixels. Most of the pixels will have a cloud index of 0, or 5, as it is only if a particular pixel is looking at the boundary between cloud and clear sky, that it will see a mixed cloud fraction.

Each set of 2640 cloud indices, extracted from a set of cloud camera images, is then assigned a validity time. As images are recorded every 5 minutes, this validity will typically extend 2.5 minutes before and after the time the images were recorded. This validity time may be extended up to 10 minutes if there happens to be no closer (in time) set taken.

4.5.2 Compression

Each cloud index is stored in the Pierre Auger database with an associated validity time (in the form of two numbers representing the lower and upper bound of validity time in terms of gps seconds), and information about which pixel in which camera it corresponds to. The large quantity of cloud camera data that must be stored makes it worthwhile to attempt to reduce the amount of storage space required to contain all the data. One way of doing this is to increase the validity times of some of these indices, such that other identical indices do not need to be stored.

If a particular pixel has a cloud index that is identical to a cloud index in a bordering validity period, then only one database entry is made -

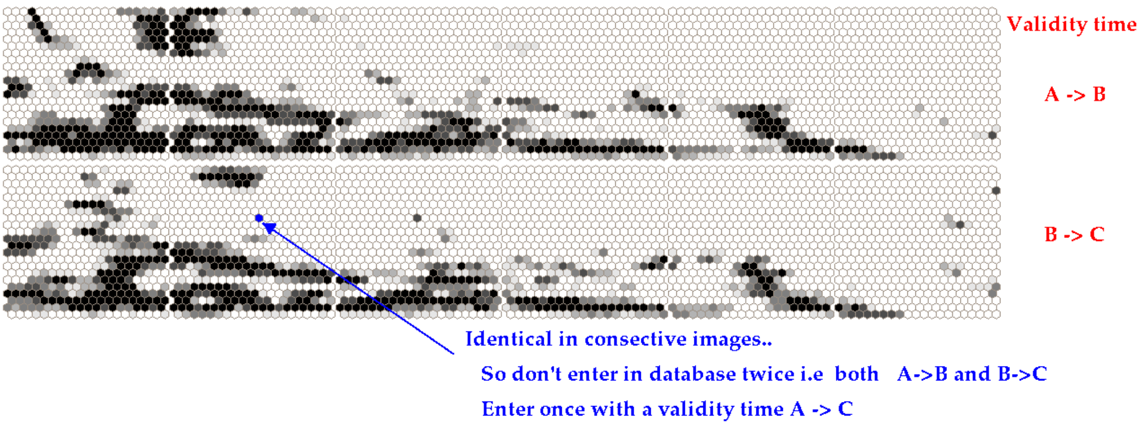

Validity time
A -> B
B -> C
Identical in consective images..
So don't enter in database twice i.e both A->B and B->C
Enter once with a validity time A -> C

Figure 4.31: *Figure demonstrating the compression of validity times. A pixel (highlighted in blue) has the same cloud index in both sets of images. Instead of making two entries with two validity periods, a single entry is made, combining both validity times into one. Other pixels in this example would undergo this same process.*

with an extended validity time covering both cloud index measurements, thus reducing the required storage space. An example of this is shown in figure 4.31 where a particular pixel has the same cloud index in two neighbouring sets of images.

Chapter 5

Accuracy of cloud camera data

The accuracy of information stored in the cloud camera database depends upon two main factors. Firstly the rate-of-capture of images by the cameras: the slower the capture rate, the less certain we are of cloud conditions between successive captures. Secondly is simply how successful the current methodology is at identifying cloud in the cloud camera images, particularly on very clear and very overcast nights. Unfortunately the lack of other independent measurements of cloud conditions above the array makes it difficult to quantify the effectiveness of the current analysis of cloud camera data.

5.1 Rate of image capture

The rate of image capture is a measure of how often the cloud cameras record cloud conditions across their corresponding FD (fluorescence detector) field of view. If images are recorded with a high rate of image capture, then cloud information is less likely to be missed, but at the same time more processing time must be spent on the images and more storage space made available. With a low rate of capture, less time processing needs to be spent but there is a correspondingly lower efficiency of recording cloud information. Images are currently recorded every 5 minutes, and it is important to check the efficiency of this so that it may be revised if necessary.

Cloud camera efficiency depends upon how often it encounters a cloud with enough speed that it fails to record spatial information of that cloud's motion. This failure is demonstrated within figure 5.1. Within the figure a set of images (image set 1) are taken showing a fragment of cloud just about to leave an FD pixel, and by the time image set 2 is taken, this same fragment of cloud has travelled across an entire pixel and has just entered a third pixel, meaning the cloud camera has no record of the fragment of cloud

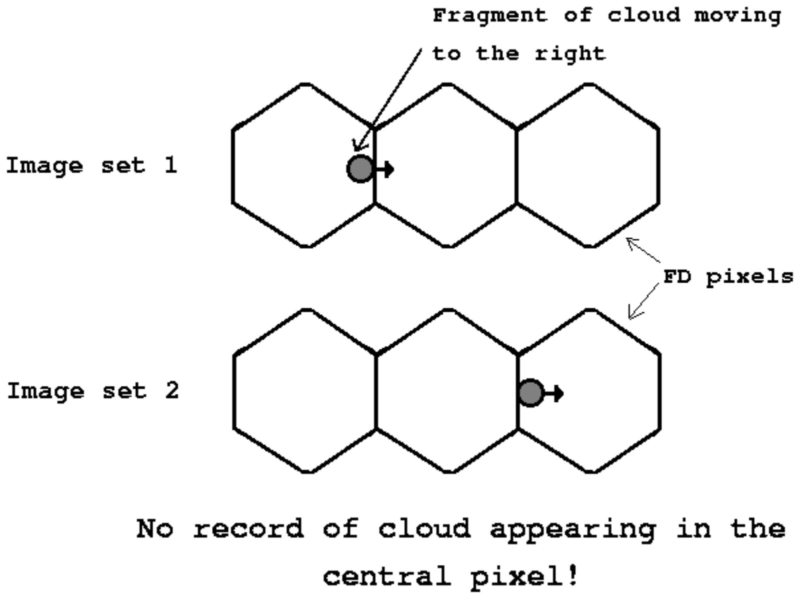

Fragment of cloud moving to the right
Image set 1
FD pixels
Image set 2
No record of cloud appearing in the central pixel!

Figure 5.1: *Figure demonstrating a loss of cloud information. In image set 1, cloud is about to leave the left-most pixel (moving to the right). By the time the second set of images is recorded in set 2, the cloud has crossed the middle pixel and has reached the right-most one.*

being in the central pixel. Thus the cloud camera fails when the angular velocity of cloud is such that it can pass through an entire FD pixel's field-of-view in between image sets i.e. when:

$$\Delta T \leq \Delta t$$

where ΔT is the time cloud takes to cross a pixel and Δt is time between image sets being recorded.

Working out the apparent angular speed of cloud is complicated by the direction of cloud movement with respect to the FD pixel. A cloud fragment moving at a particular speed perpendicular to a pixel's pointing direction is going to cross that pixel faster than if it were moving toward or away from the FD pixel. To calculate the crossing time of a cloud fragment, the distance that fragment must travel to cross an FD pixel's field of view (approximated by an ellipse with radii x, y_0 and y as seen in figure 5.2) is first estimated. Assuming that the cloud is moving parallel to the Earth's surface at a constant speed and a consistent direction (represented by the red arrow S in figure 5.2), then using the parameters described in figure 5.2, the distance (D) cloud must travel to cross the FD pixel's field of view may be estimated using equations 5.1 and 5.2. The ellipse in figure 5.2 approximates an overhead view of the shape of a pixel's field-of-view at a height H, and a cloud fragment's speed and direction of travel relative to this footprint. Once the distance is worked out, calculation of the crossing time (ΔT) is simply equation 5.3.

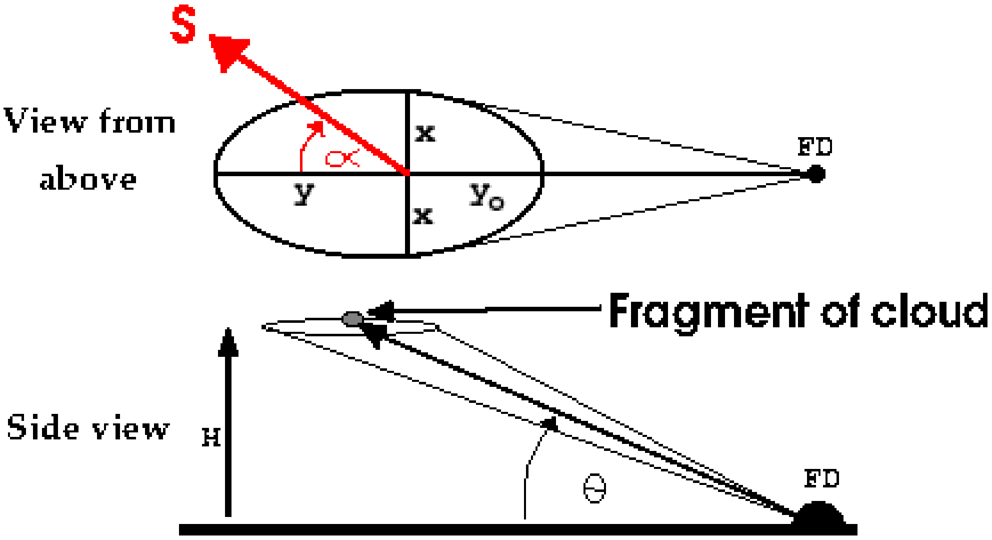


Figure 5.2: *Diagram demonstrating the geometry used to work out the crossing time of cloud. H is the height of the cloud fragment, θ is the elevation of the FD pixel looking at the cloud, S and α describe the speed and direction of the cloud fragment and x, y, $y_\circ$ describe the shape of the 'footprint' of the FD pixel's field of view at the level of a fragment of cloud.*

$$D = \frac{1}{\sqrt{\frac{\cos^2(\alpha)}{y^2} + \frac{\sin^2(\alpha)}{x^2}}} + \frac{1}{\sqrt{\frac{\cos^2(\alpha)}{y_0^2} + \frac{\sin^2(\alpha)}{x^2}}} \tag{5.1}$$

$$\begin{aligned} x &= \tan(0.75°)\tfrac{H}{\sin(\theta)} \\ y &= H(\tfrac{1}{\tan(\theta-0.75°)} - \tfrac{1}{\tan(\theta)}) \\ y_0 &= H(\tfrac{1}{\tan(\theta)} - \tfrac{1}{\tan(\theta+0.75°)}) \end{aligned} \tag{5.2}$$

$$\Delta T = \frac{D}{S} \tag{5.3}$$

Over 100 weather balloon flights have been undertaken above the Auger array over several years and different seasons [111]. These balloon flights provide details of wind speed and direction above ground level and thus an overview of the conditions experienced by clouds forming within the field of view of the cloud cameras (see figure 5.3). Generally speaking, the wind direction coincides with direction of cloud travel higher than 700m above ground level, with the speed of clouds less than, or equal to the prevailing wind speed - with larger banks of clouds moving more slowly [53][118]. Therefore assuming that cloud travel speed is equivalent to wind speed and direction will allow a conservative estimate of cloud movement to be made. Cloud height information from Auger LIDAR sites, complete the picture of the distribution of clouds above the array [135].

NOTE:
This figure is included on page 156
of the print copy of the thesis held in
the University of Adelaide Library.

Figure 5.3: *Wind data [111] recorded by 109 weather balloons above the Auger array. Each line represents a single flight.*

Figure 5.3 describes the wind speed and direction above the Pierre Auger Observatory recorded by 109 individual weather balloon flights. In the left-most plot is shown measured wind speed vs balloon height. Windspeed is suppressed close to the Earth's surface, due to friction with the ground. It increases to a maximum at approximately 9km above the Earth, corresponding to the location of the tropopause above the observatory, before beginning to decrease at higher altitudes. Wind direction is described in the right-most plot and has little consistency close to the Earth's surface, but at higher altitudes ($>$ 1.5 km) is mostly in a westerly direction.

The efficiency of a particular FD pixel is estimated through a combination of information gathered by weather balloons and LIDAR. For an Auger fluorescence detector pixel facing directly north at an elevation of 20°, the distance a fragment of cloud must travel to cross a pixel's field of view (D from equations 5.1 and 5.2), was calculated for each weather balloon measurement of wind direction and corresponding height. The related wind speed of each associated weather balloon measurement was assumed to be

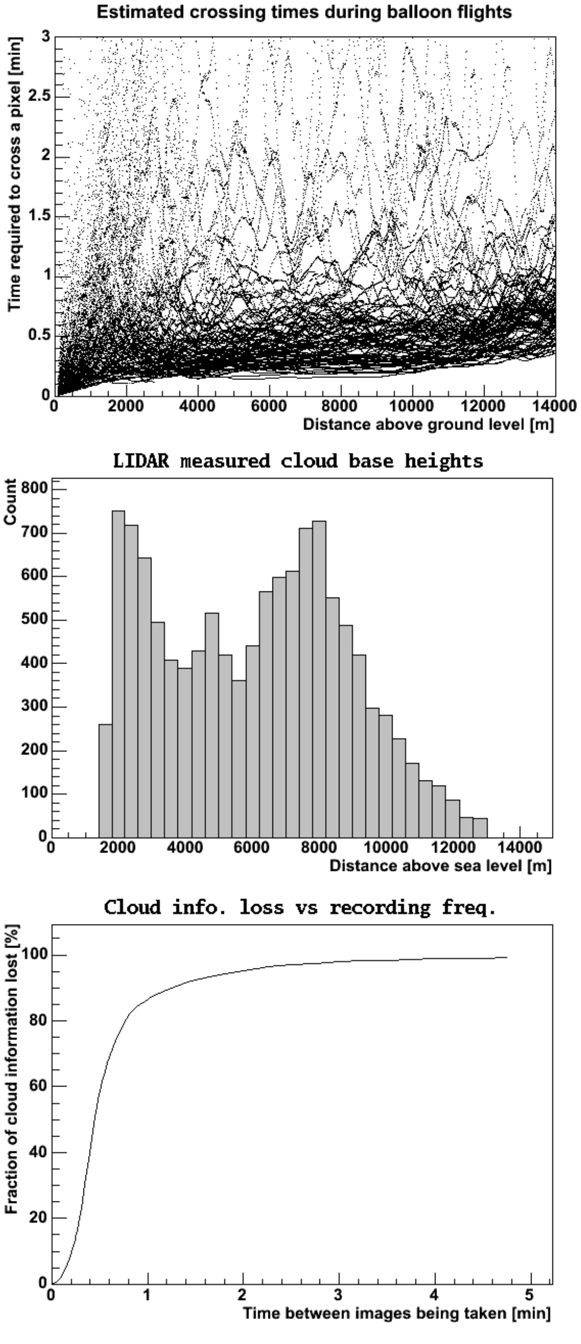


Figure 5.4: *Results for a pixel facing North at* 20° *elevation.* ***[Top left]*** *Expected crossing times for a fragment of cloud vs height during the weather balloon flights.* ***[Top right]*** *Distribution of cloud heights measured by the Auger LIDAR.* ***[Bottom]*** *The failure rate of the cameras for a* ***small fragment*** *of cloud vs image taking frequency.*

equal to the speed of any cloud present, and used with the calculated value of D and equation 5.3 to estimate the time ΔT a fragment of cloud takes to cross the pixel's field of view. The calculated crossing times are shown in the top plot of figure 5.4 and the observed distribution of cloud height by the LIDAR and CLF is shown in the plot below that. Each crossing time value is weighted by the frequency with which cloud has been observed at that associated height. The fraction of the measurements above a particular fixed crossing time is taken to be the cloud detection efficiency of the pixel, if cloud camera images are recorded that far apart from each other e.g. the weighted fraction of measurements with crossing times above 1 minute is taken to be the cloud detection efficiency if images are recorded every minute. This efficiency for different image capture rates is shown in the bottom plot of figure 5.4.

According to figure 5.4 there is over a 95% chance of missing cloud information by only taking pictures every 5 minutes, however it is important to remember that this is a pessimistic estimate. Figure 5.4 does not take into account the size of the cloud, the fact that cloud moves more slowly than the prevailing windspeed. Also note that the efficiency in this plot is only representative of a single pixel.

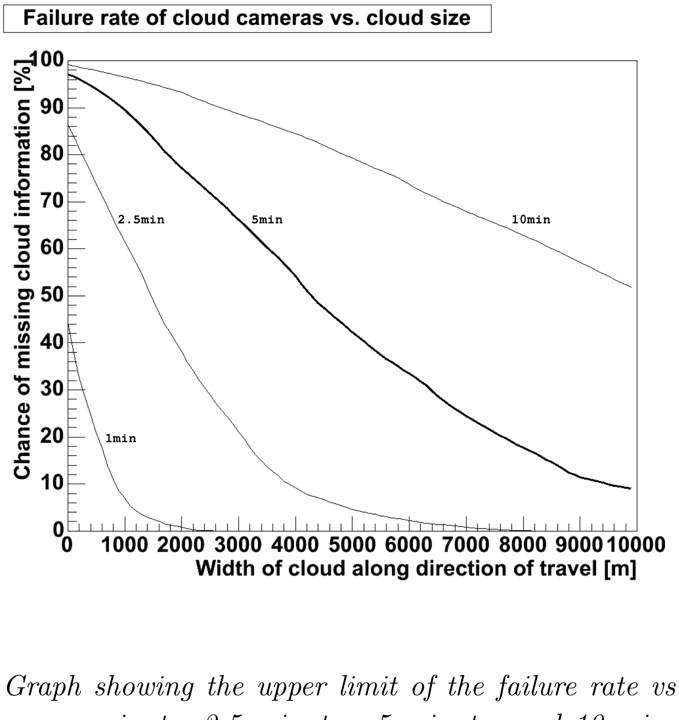


Figure 5.5: *Graph showing the upper limit of the failure rate vs cloud size, for images taken every minute, 2.5 minutes, 5 minutes and 10 minutes, for a pixel facing North at 15° elevation.*

Figure 5.5 demonstrates how the efficiency of the cameras starts to improve once the proper dimensions of a cloud are taken into account. The calculation was carried out simply by adding some cloud width to the crossing distance D, and calculating the crossing times, weighting them by the cloud height distribution then calculating the efficiency for that particular cloud width. The efficiency for several different image taking intervals are shown in figure 5.5. These still overestimate the failure rate however, as the dimensions of the cloud are highly simplified (only 1 of the 3 cloud dimensions are considered) and cloud speeds are still exaggerated. A more complete understanding of the types of cloud conditions encountered above the observatory is necessary to improve the accuracy of the estimate - but it is not unreasonable to treat figure 5.5 as an upper limit.

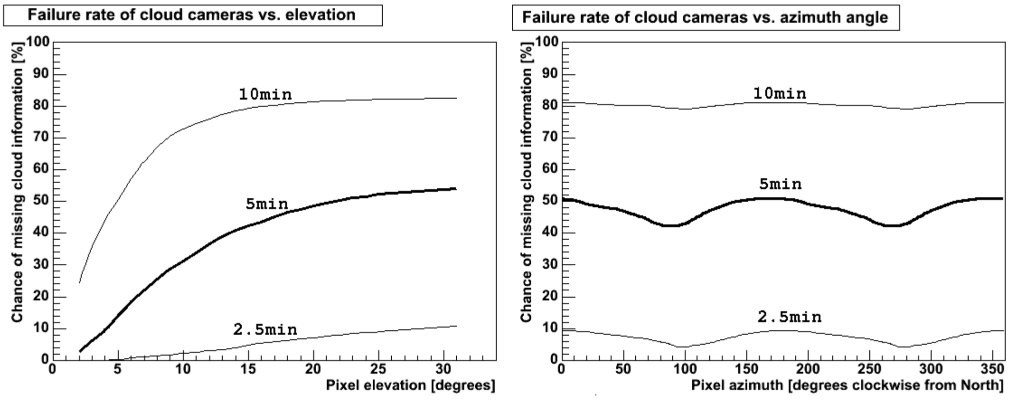


Figure 5.6: *Upper limit of the chance of failure by the cloud cameras (at various image taking frequencies) for a bank of cloud of width 5000m vs pixel pointing direction.* ***[Left]*** *Failure rate vs pixel elevation for a pixel pointing North.* ***[Right]*** *Failure rate vs azimuth angle for a pixel at 15° elevation.*

Efficiency of the cloud cameras also depends upon the pointing direction of the pixel. Different pixels will therefore have different failure rates as can be seen in figure 5.6. Pixels at lower elevations will have lower failure rates as they are looking at much larger swathes of sky, therefore any cloud within the pixel will take much longer to traverse that pixel's field of view. Pixels pointing East or West will also have a lower failure rate because (as can be seen in figure 5.3) most of the cloud will be travelling East/West across the array meaning they have to travel further (several times the distance for pixels facing North or South) to traverse a pixel's field of view than they otherwise would (due to the shape of the pixel's field-of-view). The high wind speeds measured by the balloons involved minimize such fluctuations of efficiency with azimuth angle to approximately 10% however.

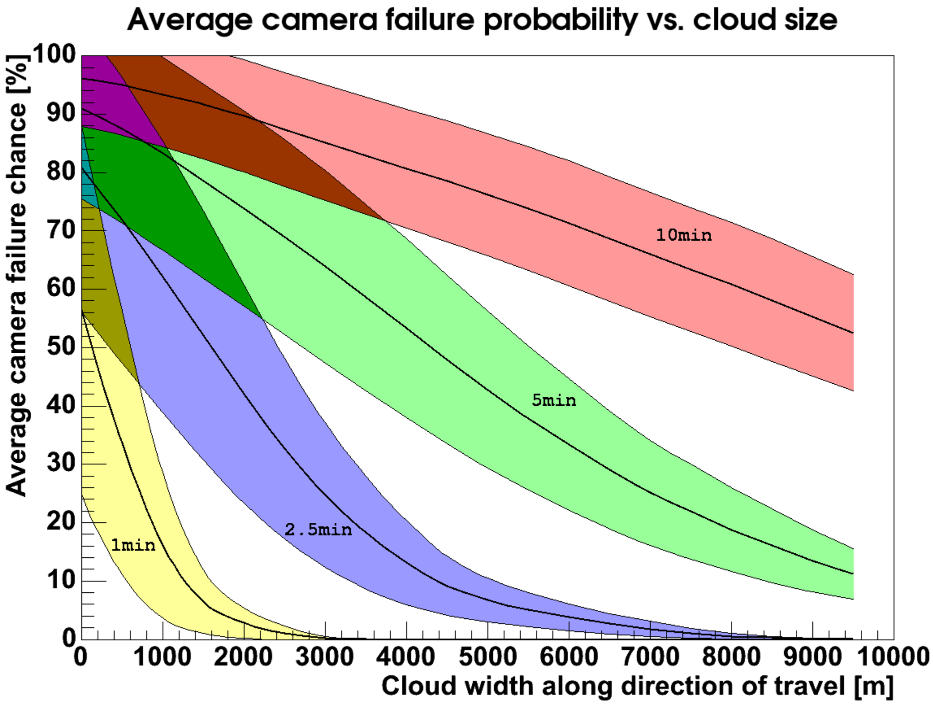


Figure 5.7: *Average chance of failure of pixels in the Los Leones cloud camera. Plot shows the average failure probability of cloud detection in FD pixels for different cloud sizes and for different image taking frequencies. Central thick lines represent the average probability for a particular image recording frequency, while the lighter outside lines are the RMS of the failure probability of the pixels.*

An overall upper limit of the failure rate for a particular camera and image taking frequency can be calculated by simply taking the average failure rate of all the fluorescence pixels at the detector site associated with that camera. as can be seen in figure 5.7. Variation of the mean failure rate is minimal (less than 2%) between different cameras, with the Coihueco and Los Morados sites having the lowest failure rate due to their East/Westery viewing direction. Substantial variations of the failure rate with azimuth angle are very localized and no larger than 10% (see figure 5.6). Overall the plot indicates very poor performance by the cloud cameras taking pictures every 5 minutes - only dropping below a 20% failure rate for clouds of 'width' greater than 7.5km.

Three factors are not taken into account in this analysis, however.

First is the actual speed of the cloud which tends to decrease with cloud size and with the vertical thickness of the cloud, which would increase the effective crossing time across the pixel. With these two factors accounted for we would observe a suppression of the failure rate for all cloud sizes, but especially so for larger cloud sizes (due to greater thickness and lower cloud speed [53][118]). The failure rate is quite high for small cloud sizes (<1km width) however (even if images are recorded every minute), so it is likely that even if the two neglected factors were accounted for we would still observe very high failure rates for small fragments of cloud. Thus care must be taken when using cloud camera data to work out the position of small fragments (particularly at higher elevations) of cloud some time before or after a particular image is taken. And thirdly this analysis only studies the ability of the cameras to always extrapolate the position of cloud between image recordings - it is important to note that efficiency also depends upon the difference between the time of interest and the time of image taking i.e. if one is interested in the cloud conditions 30 seconds before a particular image set is taken, the cloud detection efficiency is going to be better than if the time of interest is 2 minutes earlier.

The cloud cameras currently record images every 5 minutes which, according to this study, has a much higher failure rate that if images were taken more frequently. Recording images more regularly would result in a decrease in the cloud detection failure rate, but also result in a corresponding increase in the amount of data that must be both stored and processed. Therefore a balance must be struck between efficiency and practicality.

5.2 Comparison between cloud cameras

The four cloud cameras have some overlap in the volume of space they are recording cloud conditions for. Therefore one of the most obvious checks to perform is to compare the result obtained from different cameras at the same time. Clouds come in a variety of shapes and sizes and may be difficult to identify when viewed from different angles unless the height of that particular cloud is known, therefore information from the cloud cameras can only be compared in a very general sense. Cloud height information allows an estimate of the fraction of cloud seen above the Auger surface detector array by a particular cloud camera. This in turn allows a comparison between different cloud cameras for the fraction of cloud observed at a particular time.

If the height of a bank of cloud observed by a cloud camera is known, we can estimate that cloud's location above the Earth's surface. This is achieved by identifying the 'footprint' of space parallel to the Earth's surface looked at by a fluorescence detector pixel at some particular height. The shape of this footprint is described in figure 5.8 and with equations 5.4, and its location over the Earth's surface calculated using a particular height and the pointing direction of the relevent FD pixel.

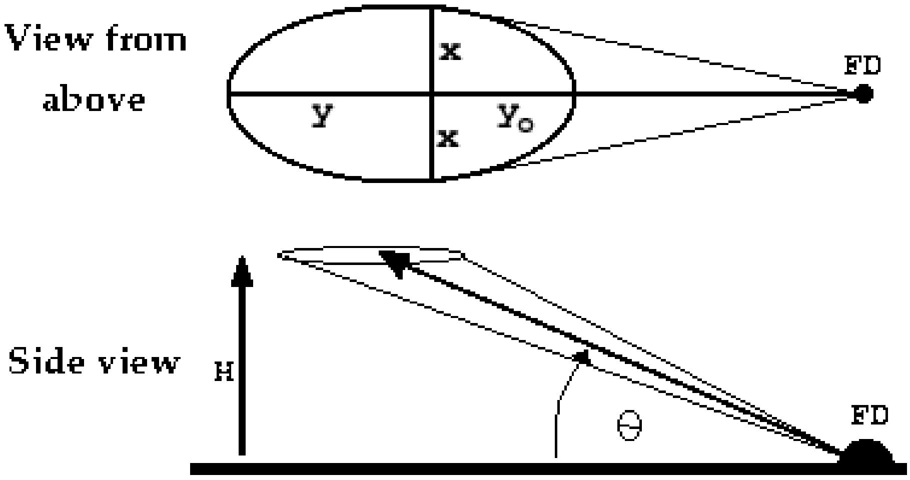


Figure 5.8: *Diagram demonstrating the geometry used to work out the region of sky seen by an FD pixel at a particular height H. θ is the elevation of the FD pixel, and x, y, $y_\circ$ describe the shape of the 'footprint' of the FD pixel's field of view at a height H above the fluorescence site..*

$$\begin{aligned} x &= \tan(0.75°)\tfrac{H}{\sin(\theta)} \\ y &= H(\tfrac{1}{\tan(\theta-0.75°)} - \tfrac{1}{\tan(\theta)}) \\ y_0 &= H(\tfrac{1}{\tan(\theta)} - \tfrac{1}{\tan(\theta+0.75°)}) \end{aligned} \tag{5.4}$$

Using cloud height information from the CLF or LIDAR, these footprints in conjunction with a cloud index calculated from cloud camera information allow a birds-eye view of the cloud coverage above the observatory's surface detector array to be created (assuming that the cloud height is uniform across the array). Figure 5.9 shows two such examples of extrapolating cloud coverage information from cloud camera data, from two different cameras at a particular point in time. The lowest cloud height measured at this point in time was combined with the cloud coverage recorded by the cloud cameras to create a birds-eye view of cloud coverage. The individual cloud index (normally varying between 0 and 5) of each pixel is broken down into simply whether that pixel sees no cloud (having an index of 0), or sees some cloud (having a non-zero index). To help ensure consistency when comparing between different cloud cameras, only the area between the four fluorescence detector sites (neglecting each detector's dead zone) has had its cloud coverage mapped.

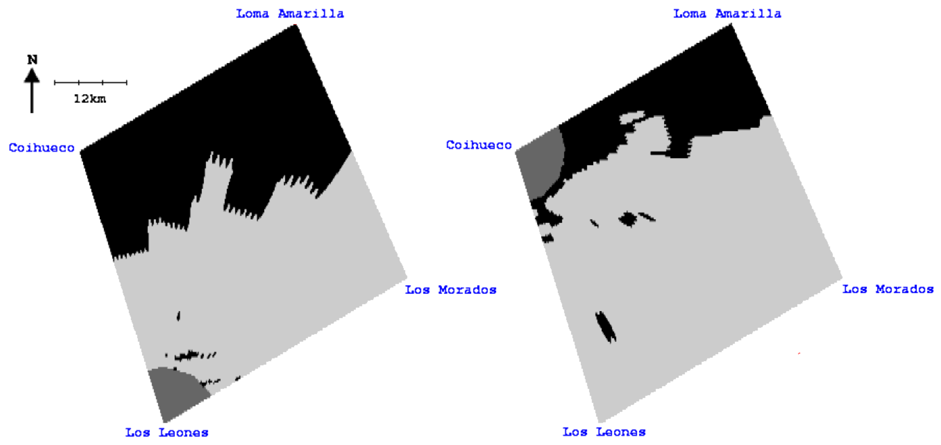


Figure 5.9: *Bird's eye view of the cloud coverage seen by the Los Leones **[Left]** and Coihueco **[Right]** cloud cameras at a particular point in time. Black and light grey represents clear and cloudy sky respectively. The dark patch of grey in front of the Los Leones and Coihueco sites is the dead zone in front of that particular eye which lacks cloud information.*

Within figure 5.9 may be seen a pair of cloud measurements made independently that agree quite well, however this method of comparison cannot be expected to always be as successful for two reasons - even if all the cloud cameras involved have identified cloud conditions quite accurately. Firstly, the cloud cameras are taking pictures of cloud cover from the side-on, not from directly below. This means that some of the cloud identified by the cloud cameras comes from the vertical thickness of the cloud, as opposed to the effective cloud coverage about the surface detector array. Since cloud thickness is neglected in this analysis, the estimated cloud fraction is an overestimate. Secondly the cloud height is estimated from CLF and LIDAR measurements within relatively localized areas, and the assumption is made that the height of any cloud seen is constant across the whole region being considered. In reality, cloud banks can occur at multiple heights and this has dire consequences for this analysis since if the cloud height is mis-identified, then the cloud location is also going to be misidentified - which obviously negatively affects the accuracy of this analysis.

Figure 5.10 is the result of a comparison between the cloud frac-

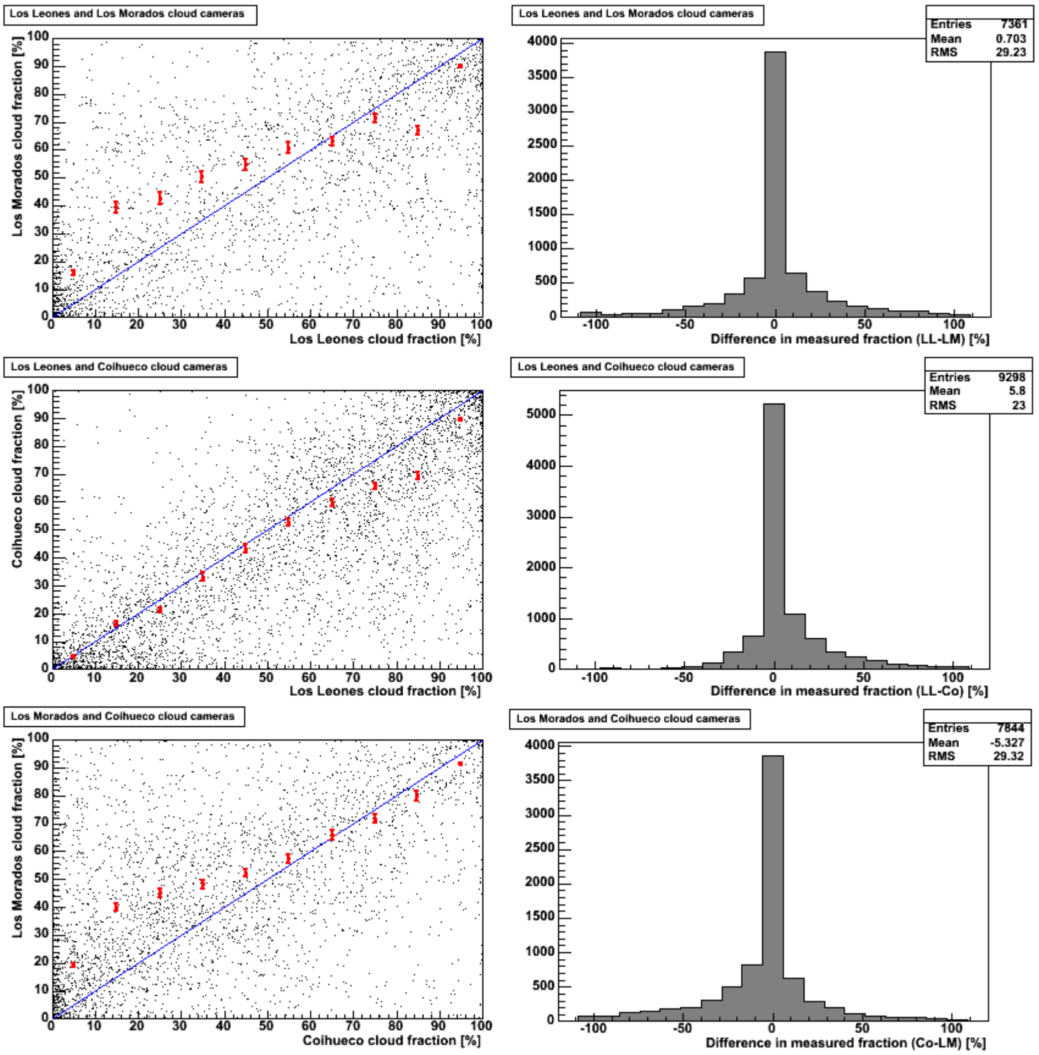


Figure 5.10: *Plots comparing the fraction of cloud seen by different cloud cameras at the same time.* ***[Left column]*** *Black dots represent measurements of cloud fractions, blue line indicates perfect agreement between the measurements and the red dots and bars are the mean and standard errors respectively.* ***[Right column]*** *Overall distribution of differences in measured cloud fractions.*

tions measured by different cloud cameras at the same time. As different cloud cameras record images at the same time, this allows a comparison between the fraction of cloud seen by different cameras to be made. All available comparisons between cloud camera measurements were made from data with cloud height measurements up until the end of 2007, and used in this study. A total of 73610, 92980 and 78440 images were used in the comparison between Los Leones and Los Morados, Los Leones and Coihueco, and Los Morados and Coihueco respectively. In the left-hand column the mean cloud fraction (with standard error) measured by one camera is compared with the cloud fraction measured by another camera, and in the right-hand column the distribution of the disagreement of measured cloud fractions is represented. Overall the comparison between the Los Leones and Coihueco cameras show the best agreement, with the majority of the disagreement between Los Morados andLos Leones/Coihueco occuring in the <45% cloud fraction range. Los Leones also appears to be overestimating the cloud fraction measured by Los Morados and Coihueco at >75% fractions.

Overall there is good agreement between the independently measured cloud fractions, 72% of the measurements made by the Los Leones and Los Morados cameras are within 20% of agreement with each other. Similarly, 78% and 71% of the measurements between Los Leones/Coihueco and Los Morados/Coihueco are within 20% agreement with each other respectively. The overall distribution (shown on the right-hand side of figure 5.10) of the differences between the fraction of cloud measured is centred around, or close to zero for all the different comparisons. This implies an overall good agreement between cloud camera measurements, as there is little, if any, overall systematic difference between them.

There appears to be, in some cases, some variation between cloud fraction measurements between cameras under certain cloud conditions. There is some indication of a bias of Los Morados observing more cloud on clear nights than the Los Leones and Coihueco cameras, and Los Leones observing more cloud on average than the other cameras in very cloudy conditions. A possible explanation for Los Morados is that of the three cameras, it is the only one actually pointing towards the Andes mountain range to the West of the observatory. Thus if cloud height is underestimated, the Los Morados camera is more likely to see extra cloud above the mountains appearing in an otherwise clear day. Los Leones is slightly puzzling however, it is pointing more towards the mountain range than the Coihueco camera so it may be seeing more cloud from that (similar to Los Morados). There is no increase in measured cloud fraction for the more cloudy nights from the Los Morados camera (which agrees well with Coihueco) however. There may be some sort of bias in the analysis of the Los Leones data where cloudy conditions are being overestimated - assuming of course that it is not the Los Morados and Coihueco data!

	Los Leones	Los Morados	Coihueco
Los Leones	-	69%	90%
Los Morados	74%	-	78%
Coihueco	71%	61%	-

Table 5.1: ***Agreement between cameras on clear nights:*** *e.g. 90% of times thought to have less than 10% cloud cover by Los Leones, also had less than 10% cloud cover measured by the Coihueco cloud camera.*

	Los Leones	Los Morados	Coihueco
Los Leones	-	78%	78%
Los Morados	80%	-	74%
Coihueco	87%	82%	-

Table 5.2: ***Agreement between cameras on overcast nights:*** *e.g. 80% of times thought to have greater than 90% cloud cover by Los Morados, also had greater than 90% cloud cover measured by the Los Leones cloud camera.*

Another area of interesting comparison is how successfully the cloud cameras are differentiating between cloudy and clear sky conditions. Due to the nature of the cloud cameras, they are very poor are differentiating between featureless scenes (i.e. clear or overcast nights), therefore it is interesting to see how often this may lead to disagreement between cameras. Within tables 5.1 and 5.2 may be seen the results of a comparison which describes how often different cloud cameras agree about clear and overcast conditions. Cloud fractions were calculated using the same method used in figure 5.10. On average, about 26% of the time the cloud cameras appear to disagree about conditions being clear, and about 20% of the time while conditions are overcast. The lower disagreement rate for overcast conditions is likely because completely overcast conditions are sometimes easier to confirm due to the texture of the cloud than completely clear conditions which always lack such texture.

5.3 Comparison with LIDAR data

Each fluorescence detector has a LIDAR installed nearby that can provide independent measurements of cloud information. Every LIDAR provides not only a measurement of the height of any nearby cloud, but also the fraction of sky filled by cloud above each fluorescence detector. With cloud height information the fraction of cloud above the array seen by a cloud camera may be estimated and compared with the fraction of cloud seen by the LIDAR. The LIDAR and cloud cameras are in this case deriving their cloud fractions from different areas of the sky, therefore an exact correlation between the two is not expected. Even so there does appear to be some correlation between the two measurements, particularly on very clear and very overcast days.

The LIDARs are 351nm pulsed Nd:YLF lasers stationed at each of the four fluorescence detectors [9]. Laser light is backscattered by the atmosphere, and received by three mirrors at the LIDAR. This scattering information not only helps determine the distribution of aerosols in the atmosphere at a particular time, but also may be analysed to provide information on cloud conditions [135]. Monthly models of the molecular atmosphere above the Auger site are used to estimate and subtract the expected signal Rayleigh scattered by the LIDAR laser. As clouds appear as strong echoes of the laser signal they appear quite strongly in the resulting subtracted profile. This information may then be used to estimate the height of any cloud, its optical thickness and the fraction of cloud in the sky. Measurements are taken and are available on an hourly basis.

The most obvious method of comparison is simply to compare the fraction of cloud measured by the LIDARs with the fraction of cloud seen by the cloud cameras. Using the method described in section 5.2, the lowest cloud height measured at a particular time (from either CLF or LIDAR) is used to determine the position of clouds observed by the cameras, which is then used to determine the fraction of cloudy sky between the four fluorescence detector sites. Cloud fractions reported by the LIDAR are only from sky within 45° of the zenith above the FD site (not regions covered by that site's cloud camera). If multiple cloud layers are detected, the average fraction of each layer is averaged across a whole hour for a single measurement. Thus a perfect relationship between the two cloud fraction measurements is not to be expected.

Figure 5.11 contains a comparison between the average fraction of cloud measured by the LIDARs at a particular point in time and the fraction of cloud observed by each particular cloud camera at that same time. Coihueco shows the best agreement between the two measurements, while the Los Leones and Los Morados cloud cameras - while appearing to agree

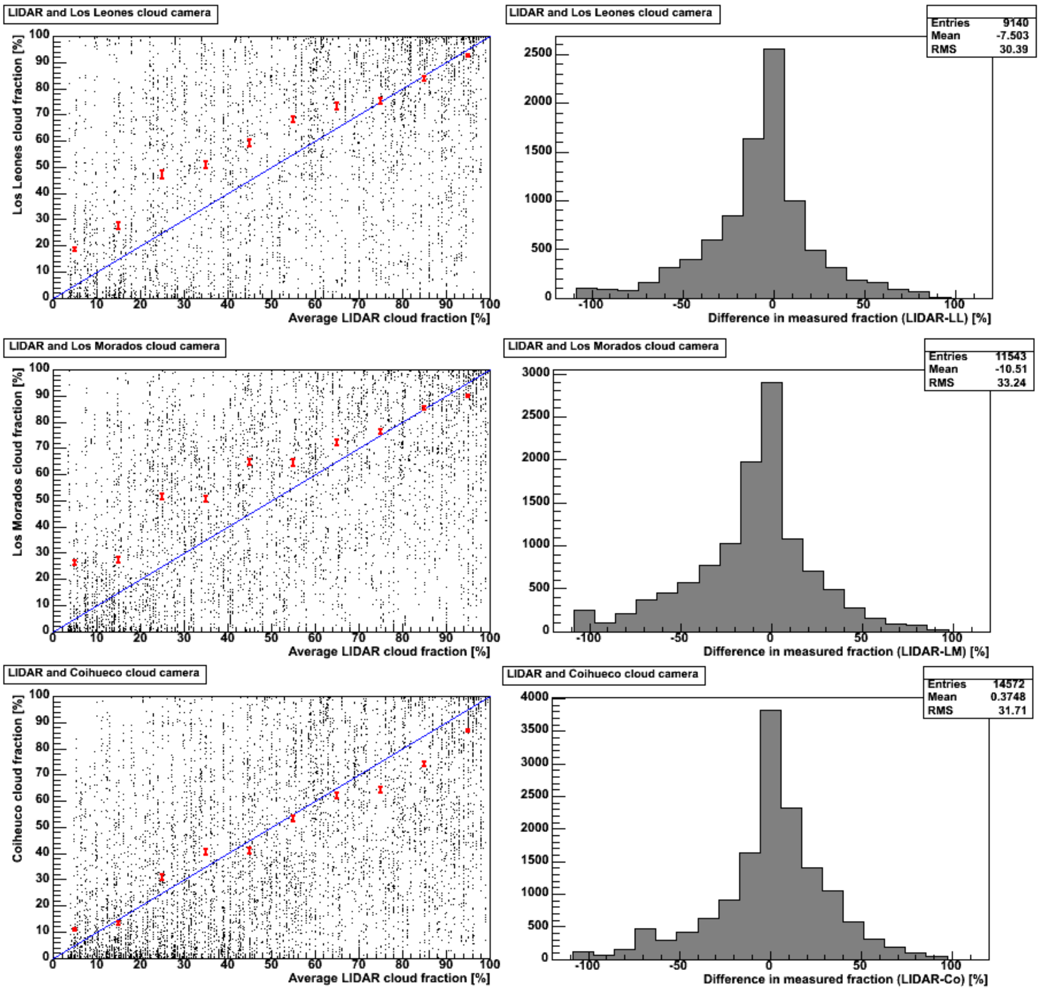


Figure 5.11: *Plots comparing the fraction of cloud seen by cloud cameras and the LIDARs at the same time.* ***[Left column]*** *Black dots represent measurements of cloud fractions, blue line indicates perfect agreement between the measurements and the red dots and bars are the mean and standard errors respectively.* ***[Right column]*** *Overall distribution of differences in measured cloud fractions.*

well for cloud fractions >70%, tend to observe more cloud than the LIDAR on clearer nights. Overall agreement between the LIDAR and cloud cameras is not as good as between the cloud cameras themselves (see section 5.2) with only 61%, 56% and 58% (Los Leones, Los Morados and Coihueco respectively) of the cloud camera fractions lying within 20% of the lidar data.

Within section 5.2 it was noted that Los Morados tended to report more clouds on clear days than the other two cameras, this also is the case when comparing it against LIDAR measurements in figure 5.11. This is likely due to Los Morados pointing more directly towards the Andes mountains and thus observing more clouds which are counted in the cloud fraction estimate when the cloud height happens to be underestimated. Comparison with the LIDAR also suggests that Los Leones experiences the same phenomenon (probably due to the same reasons, given it is also partly pointing towards the Andes), but unlike Los Morados this result was not evident in the comparison with other cloud cameras - though it was noted that Los Leones appeared to be recording extra cloud on cloudy days. One explanation may be that it is Coihueco and Los Morados that has been underestimating the cloud on cloudy days (instead of Los Leones overestimating), this would explain the slight tendency of the LIDAR to record more cloud than Coihueco on cloudy days in figure 5.11 - but no such tendency is evident in the Los Morados comparison. Overall, on average, Coihueco shows good agreement with the LIDAR cloud fraction measurements - better than the other two cameras perhaps because, unlike the others it is not pointing at the Andes and is thus consistently viewing the same type of sky as the LIDAR.

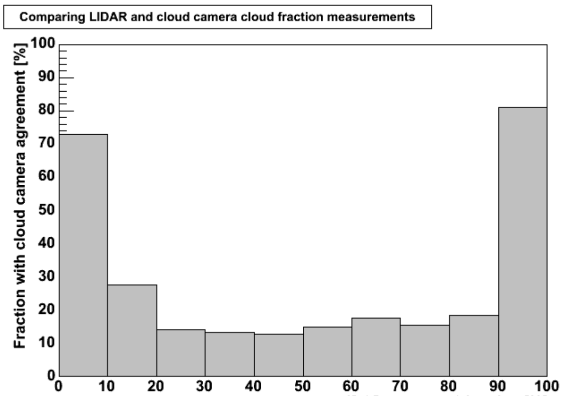


Figure 5.12: *Plot demonstrating agreement between LIDAR measured cloud fraction and cloud cameras. They are said to be in agreement if the difference between the LIDAR and the cloud camera measured cloud fraction is less than 10%.*

The strongest agreement between LIDAR and camera measurements occurs during time periods when the night sky is either very clear, or very overcast - as seen in figure 5.12. A likely explanation for this is simply because clouds are usually a localized phenomena. If a LIDAR records a partial fraction of cloud there is no guarantee that the rest of the sky is as cloudy, perhaps a single cloud just happened to be passing overhead at the time. On the other hand, if there is a large amount of cloud structure present at a particular time then the LIDAR is likely to record a very high cloud fraction, that is correlated with the cloud structure in the area at large. With the converse being true for no clouds being recorded by the LIDAR - no clouds detected implying either clear or very sparsely cloud populated skies.

5.4 Comparison with CLF data

Another instrument that can detect the presence of clouds is the CLF (Central Laser Facility). The CLF primarily studies aerosol content within the atmosphere, but the strong scattering of its laser light off clouds means it is capable of detecting and measuring the height of any clouds directly above the facility. Using cloud height information measured by the CLF, cloud seen by a cloud camera can be checked to see if it lies above the CLF - allowing a cross-check between the two instruments.

The CLF is a 355nm pulsed YAG laser positioned within the center of the Auger surface detector array [75][76]. Light scattered from the laser fired into the atmosphere can (among other things) be used to measure the abundance of aerosols over the Auger observatory at a particular time and can help calibrate the observatory's fluorescence detectors. During FD operation the CLF laser is fired 200 times every hour and the results averaged to provide hourly measurements of the aerosol extinction coefficient and the vertical aerosol optical depth (VAOD). Clouds may also be detected by the CLF - appearing as sharp steps in the measured VAOD profile. The lowest base height of any observed cloud structure above the laser is recorded in a database.

Using the method described in section 5.2, cloud detected by the CLF can be cross-checked with cloud identified by the cloud cameras. Cloud height is provided by the CLF database only once an hour and is the lowest cloud height recorded by the CLF during that hour. This height is used to estimate the position of cloud above the surface detector array (see figure 5.13). If cloud cameras identify cloud within a 2km lateral distance from the CLF during any hour then the CLF and cloud camera are said to be in agreement. Increasing the range of agreement from 2km to something higher

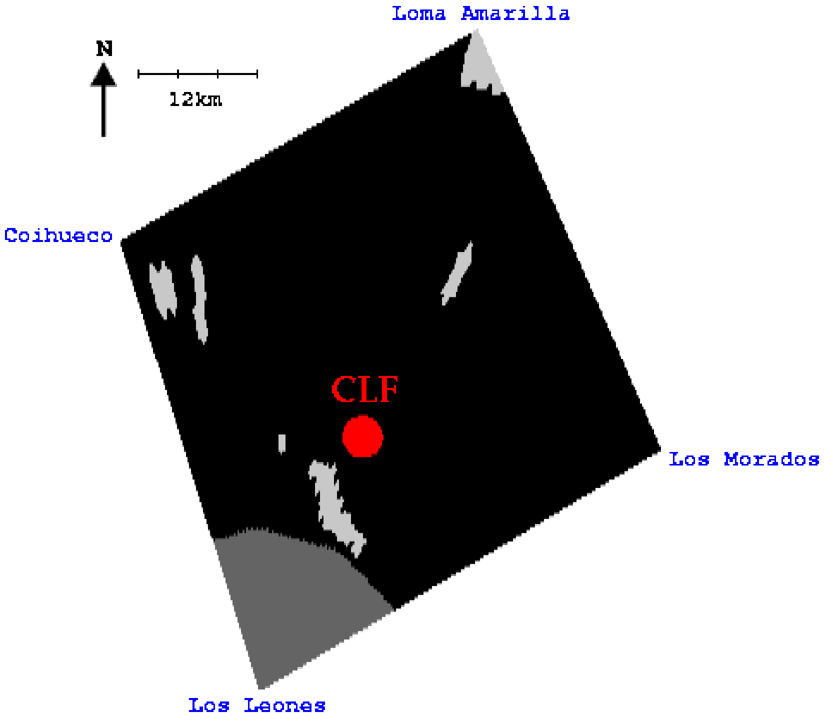


Figure 5.13: *Bird's eye view of the area between the four fluorescence detectors. Cloud height information has been used to estimate the position of cloud (cloud and clear sky marked by light grey and black respectively) seen by the cloud cameras over the array. The location of the CLF is marked in red.*

will result in better correlation for obvious reasons.

Figure 5.14 demonstrates how often a correlation is seen between cloud cameras and the CLF. For all three cloud cameras there is good agreement at least 90% of the time while looking at clouds at lower altitudes. At heights greater than around 8000m however, the correlation steadily drops off to around 40%, or 20% in the case of Coihueco.

One possible reason for the drop in cloud camera/CLF agreement with cloud height is the loss of resolution associated with higher cloud altitudes. With increasing cloud heights, information from each individual pixel is being assigned to an ever increasing area, while the region above the CLF being considered remains the same. Thus the decision about whether the CLF agrees with the cloud camera or not is being based upon a smaller and smaller portion of the cloud camera's data. Before dropping in agreement above cloud heights of 8000m, there is on average approximately 90% agree-

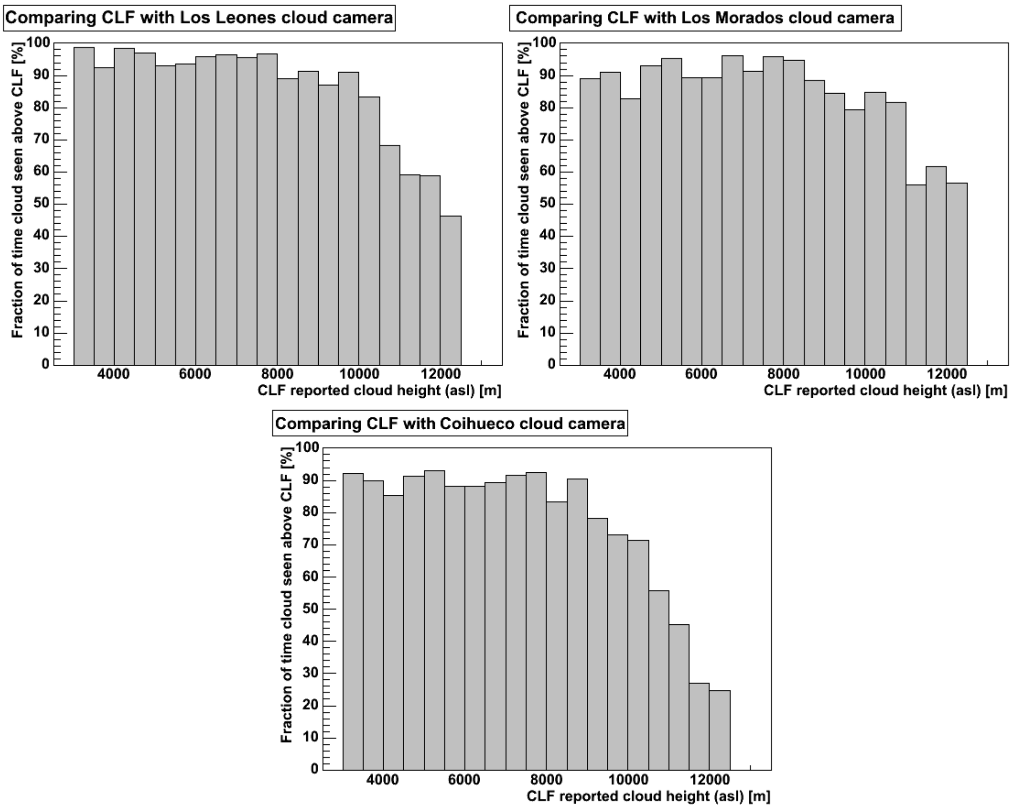


Figure 5.14: *Plots describing the frequency at which the CLF and cloud cameras agree on the presence of cloud above the CLF, as a function of the cloud height reported by the CLF. If the cloud camera detects cloud above the CLF while the CLF is reporting the presence of cloud (in the form of a reported cloud height), then the cloud camera is said to agree with the CLF, for that particular CLF measurement.*

ment between the cloud cameras and the CLF, suggesting good accuracy in the analyzed cloud camera data. Another possible reason for the poor cloud camera and CLF agreement for higher altitude clouds may be due to their lower infrared flux (see section 3.2.2). Higher altitude cirrus cloud is optically thinner than lower altitude stratus cloud, and is often easily mistaken for a clear night sky by the cloud cameras.

Another way of looking at the agreement between CLF and cloud camera is in terms of cloud fraction measured by LIDAR in figure 5.15. In order to remove the fall-off in correlation with cloud height observed in figure 5.14, all CLF measurements where the cloud height has been above 8000m have been ignored in figure 5.15. The result indicates agreement between the

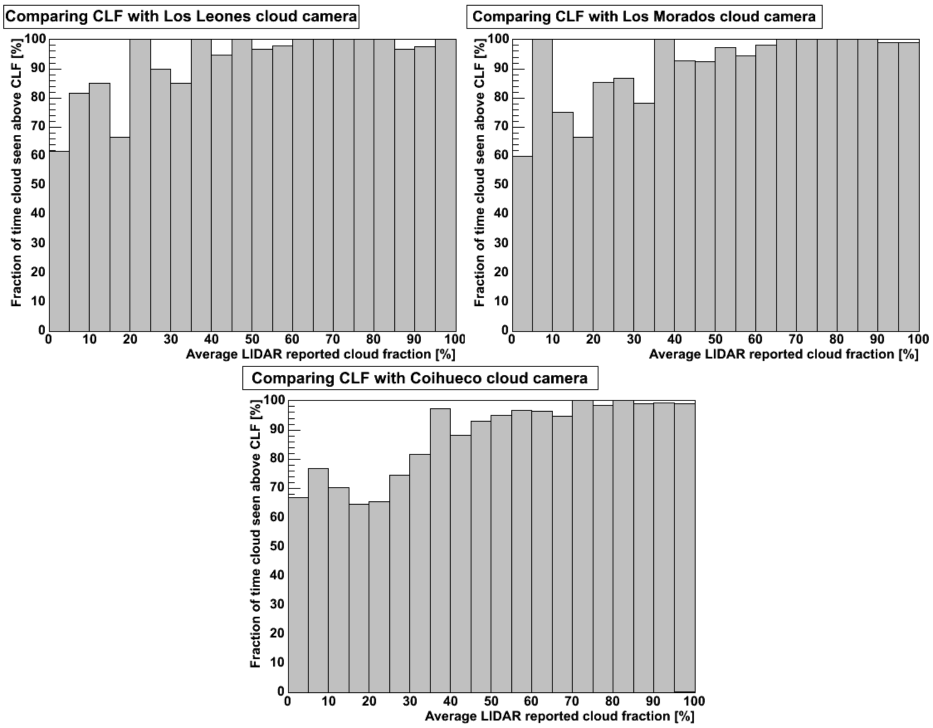


Figure 5.15: *Plots describing frequency at which the CLF and cloud cameras agree on the presence of cloud above the CLF, depending upon the average fraction of cloud present according to LIDAR measurements. CLF height measurements >8000m above sea level have been neglected in these plots.*

CLF and cloud cameras in this analysis varies according to the amount of cloud present, with at least 60% agreement on very clear nights, and close to 100% agreement on very cloudy nights.

The stronger agreement between the instruments on cloudy nights is likely due simply to the fact that more cloud is being observed by the cloud cameras. Therefore during cloudy conditions, even if the cloud camera analysis misses a particular piece of cloud above the CLF, there is a stronger chance (than on a clearer night) that there is going to be another bank of cloud observed passing above the CLF which will trigger agreement in this analysis. The weakest agreement between the cameras and CLF occurs in very clear conditions with only approximately a 60% (66% in Coihueco's case) agreement rate, which may be taken as a pessimistic lower estimate on the sensitivity of the cloud camera analysis. It is important to remember that (as seen in section 5.1) the movement of smaller clouds is more difficult to record with the cloud cameras, and that on clearer nights we are more likely to encounter smaller clouds (since if they were bigger we would have a larger cloud fraction by definition). Thus the cloud cameras may have seen the cloud detected by the CLF, but due to its movement it was not actually recorded as being over the CLF. With more regular cloud height measurements from the CLF it will be possible to overcome this (since cloud camera images could be correlated much better in time with the CLF) and gain a much better estimate of the sensitivity of the cloud cameras.

Chapter 6

Applying cloud camera data to events

Being able to provide a record of cloud conditions during the measurement of extensive air showers is the ultimate goal of the cloud cameras. Within this chapter is described the means by which cloud camera data is combined with cloud height information to assess the location of any cloud with respect to a particular extensive air shower. Several years of cloud camera data are now available for analysis, and are used in this chapter to study the effect of a cloud camera based quality cut upon extensive air showers.

6.1 Extensive air showers and the position of cloud

Information from a cloud camera is strictly directional - the camera reveals the direction in which cloud is seen, but not how far away it is. Thus the position of any observed cloud, from cloud camera data alone, cannot be known with respect to any particular extensive air shower. In order to determine the position of a cloud with respect to an extensive air shower, the cloud's height must be known (or estimated), then combined with information from the cloud camera.

A cloud is said to be obscuring an air shower (in a particular fluorescence detector pixel) if the distance between it and the fluorescence detector is less than the distance between the detector and the axis of the shower i.e.

$$D_{axis} \geq \frac{H}{\sin(\theta)} \tag{6.1}$$

where D_{axis} is the distance between the FD pixel and the extensive air shower axis, H is the measured cloud base height[1] above the detector and θ is the elevation of the FD pixel in question.

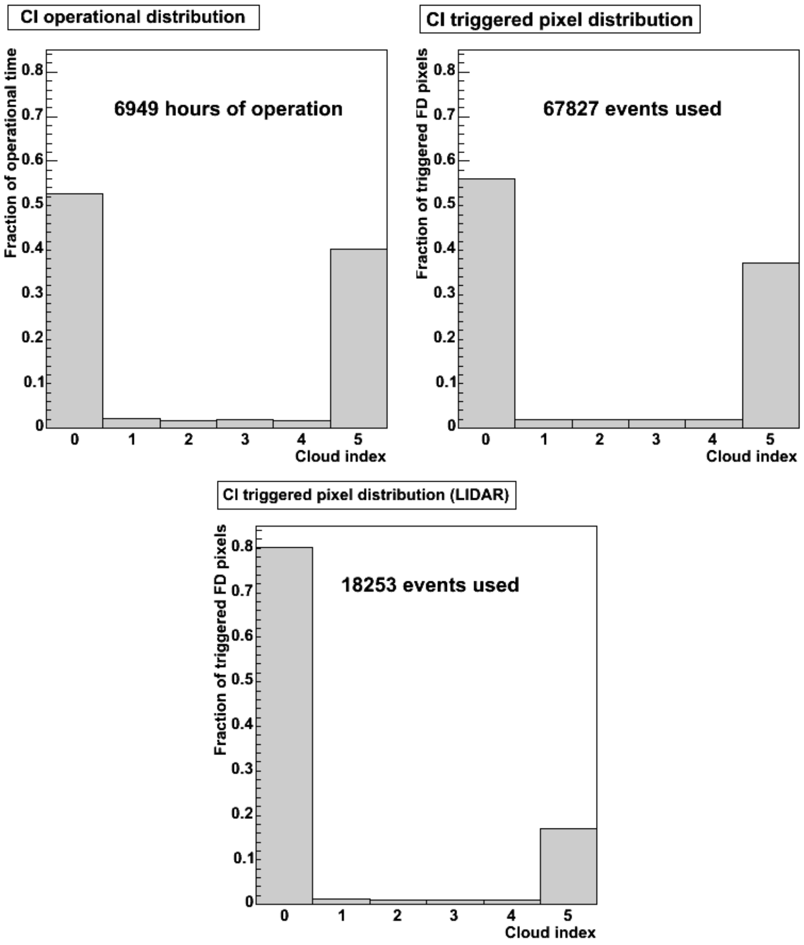


Figure 6.1: *Plots showing distribution of cloud indices.* ***[Top left]*** *Distribution seen by cameras for all time.* ***[Top right]*** *Distribution of cloud indices associated with pixels triggered by extensive air showers (no substantial quality cuts applied).* ***[Bottom]*** *Distribution of cloud indices that are actually (according to LIDAR cloud height information) obscuring an extensive air shower. Note that a cloud index of 0 indicates that a pixel's view of a shower is not obscured.*

Figure 6.1 describes the fraction of triggered FD pixels reported for each possible cloud index value. The plots were constructed out of hybrid

[1] The measured cloud base height is taken as the lowest cloud height measured by the CLF or LIDAR at a particular time.

event and cloud camera data up until the end of 2006. In order to ensure no bias in the data toward cloud-free events (by other quality cuts) only the geometry cuts (in appendix A) were used to select events. A cloud index of 0 corresponds to no cloud within that particular pixel, and an index of 5 indicates that a particular pixel's field of view is completely filled by cloud.

In the top-left plot may be seen the overall distribution of cloud indices recorded by all the cloud cameras before the end of 2006. The top-right plot is the distribution of cloud indices seen by the cloud camera in the triggered pixels of the selected 67827 events. The bottom plot is the distribution of actual obscuration for the 18253 events that had both cloud camera and LIDAR cloud height information available. If a non-zero cloud index was reported by the cloud camera but the LIDAR implied that the cloud was behind the shower, then the index was converted to a 0.

As different cloud heights are sometimes reported by different LIDAR stations, only the lowest reported cloud height at a particular time is used. The LIDARs are measuring the cloud height only from clouds directly above each fluorescence detector, therefore it cannot be known with absolute certainty if a particular bank of cloud in the cloud camera's field-of-view has the same cloud height as that reported. By using only the lowest measured cloud height, the analysis is made as conservative as possible.

The plots in figure 6.1 demonstrate the necessity of combining cloud height information with cloud camera measurements. Distributions of cloud indices in triggered pixels during events and the overall distribution of recorded indices are very similar, with a slight bias towards an index of 0 for the triggered pixels. This is because fluorescence light from extensive air showers is more likely to be seen by the Auger detectors when there is no obscuring cloud.

Overall, nearly half of the triggered pixels in the top-right plot of figure 6.1 have some form of cloud being seen by the cameras and any quality cut made by the cloud camera based on this data would have catastrophic consequences for the final number of showers left for analysis. Using cloud height information drastically cuts down the fraction of obscured pixels (as seen in the bottom plot) - with only 20% or so of the overall triggered pixels being flagged as being obscured by cloud. Clearly using cloud camera information on its own is extremely conservative when applying quality cuts, as the vast majority ($\approx$ 80%) of triggered pixels are not seen to be obscured by cloud when cloud height information is taken into account.

6.2 Availability of cloud camera data

In this section the availability of cloud camera data on a month-by-month basis is presented. Geometry quality cuts were used to select events between 2004 and 2007, which were then checked for the availability of cloud camera data. Unfortunately cloud camera coverage is not as extensive as it could be due to hardware problems in the camera systems.

From data recorded between the start of 2004 and the end of 2007 events were selected using basic geometry cuts described in appendix A. Each of these 84249 events were then cross-checked with cloud camera data to see if the cloud camera has data available during that particular event. The fraction of events that had cloud camera data are presented in figure 6.2 on a month-by-month basis.

The Los Leones, Los Morados and Coihueco cloud cameras came online in April 2004, November 2005 and April 2005 respectively. Hardware technical difficulties have occurred all throughout the operation of all cameras - accounting for the less than perfect coverage for many months. Severe hardware problems are responsible for the complete lack of usable data recorded in September/October 2006 and January 2007 by the Los Leones camera, September/October 2006 for the Los Morados camera and August/September 2007 for the Coihueco cloud camera.

Hardware difficulties usually occur as a result of problems with the cabling connecting the camera to the computer within the fluorescence detector building and usually take the form of the pan-and-tilt device (that the camera is mounted upon) failing to respond, or corrupted images being received by the camera control computer. As a whole, 64%, 70% and 69% of the selected events (measured by Los Leones, Los Morados and Coihueco respectively) have cloud camera data available.

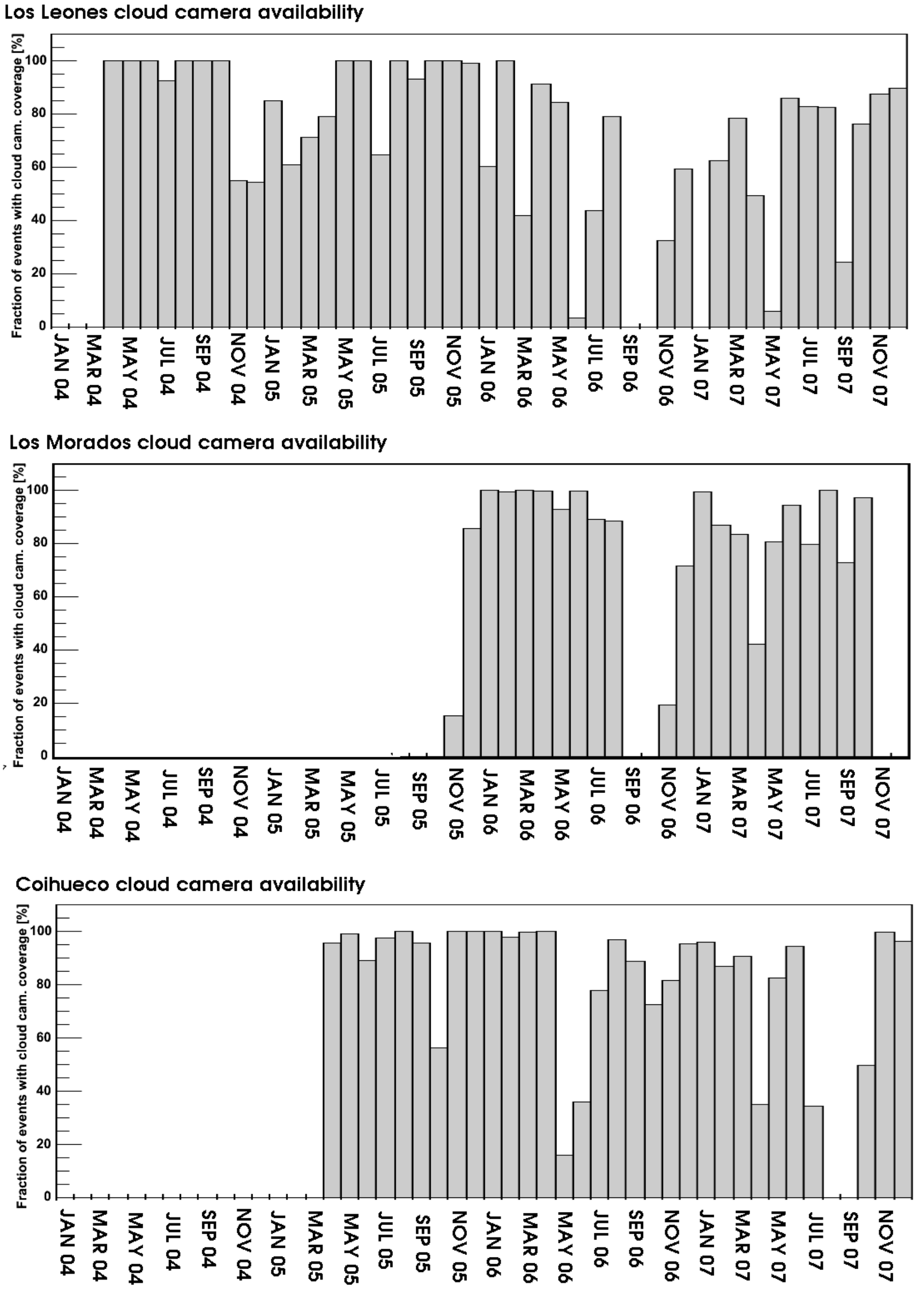


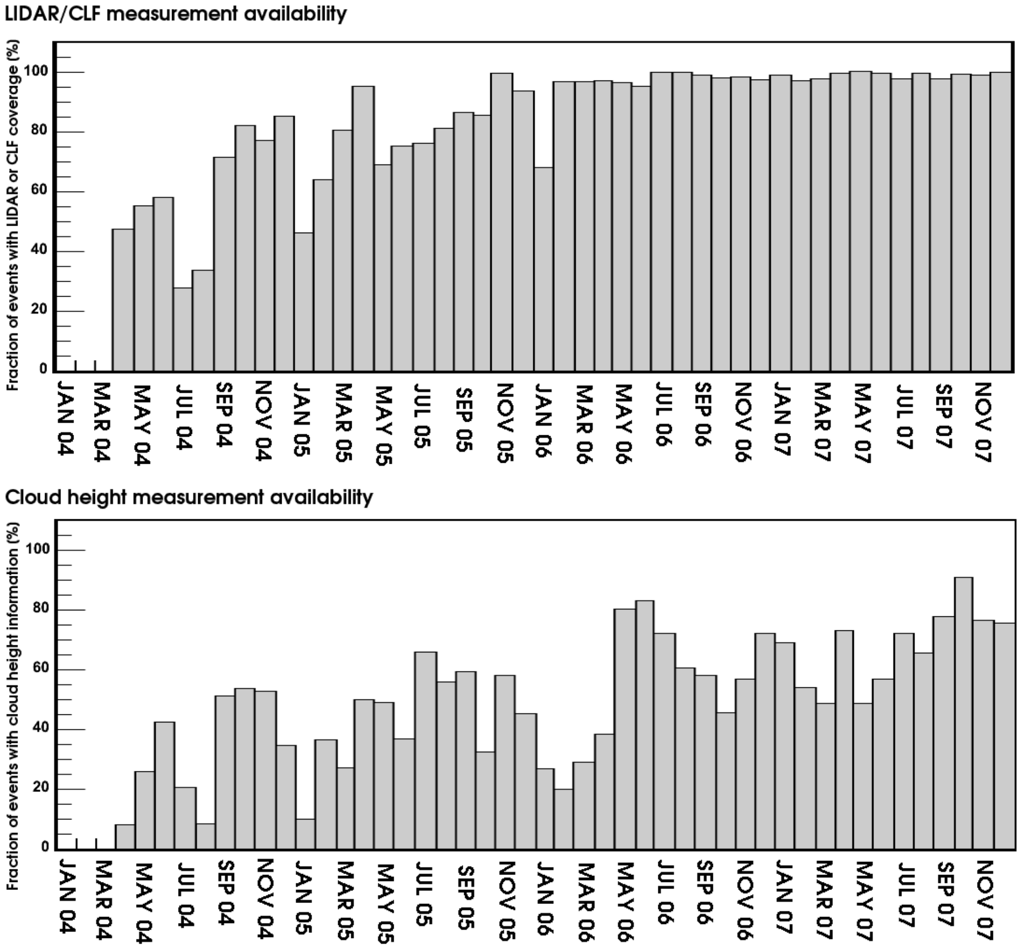
Figure 6.2: *Plots showing the fraction of selected events seen by a particular fluorescence detector that have cloud camera data available on a month-by-month basis between the beginning of 2004 and the end of 2007. Cloud camera data recorded in 2008 and 2009 is currently still being processed.*

6.2.1 Availability of cloud height information

Cloud height information is important in that it allows the cloud cameras to more accurately determine the position of any cloud above the Auger observatory. LIDAR or CLF measurements are available for the vast majority of events selected for this study, however clouds are not always detected by these instruments. This may be due to a lack of cloud above the array, or because none of the cloud present happens to be above the CLF or LIDAR sites. Thus the actual availability of cloud height information is less than would first appear. Cloud height information measured at other times may possibly be used to increase the number of showers with cloud height data.

As in the previous section, 84249 events were selected according to the basic geometry cuts described in appendix A between 2004 and the end of 2007. The CLF and LIDAR instruments were then checked for any available measurements, the result of which may be seen in the upper plot in figure 6.3. Coverage is presented on a month-by-month basis and is available for nearly every selected event after November 2005. Often the cloud camera reports seeing cloud, yet neither the CLF or LIDAR detect cloud, despite being operational - simply because none of the observed cloud happens to be in a detectable position for either instrument. If no cloud is detected, then the LIDAR/CLF data will not return a valid cloud height measurement. The bottom plot of figure 6.3 shows the actual cloud height availability for all the selected events which is, on average, about 60%.

Application of the cloud camera database usually requires both cloud camera and cloud height information to be available. The fraction of selected events that have enough information available to determine cloud obscuration is represented in figure 6.4. Some events that only have cloud camera data do not need cloud height information because no clouds were detected by the cameras, which is why some months in figure 6.4 have coverage despite the lack thereof in figure 6.3.



LIDAR/CLF measurement availability

Cloud height measurement availability

Figure 6.3: *Plots showing the availability of measurements made by the CLF or LIDAR, and the availability of actual cloud height measurements for selected hybrid events. [**Top**] Fraction of events that have a measurement made by the CLF or LIDAR available. [**Bottom**] Fraction of events that actually have a cloud height* (< 13*km*) *measured at the time.*

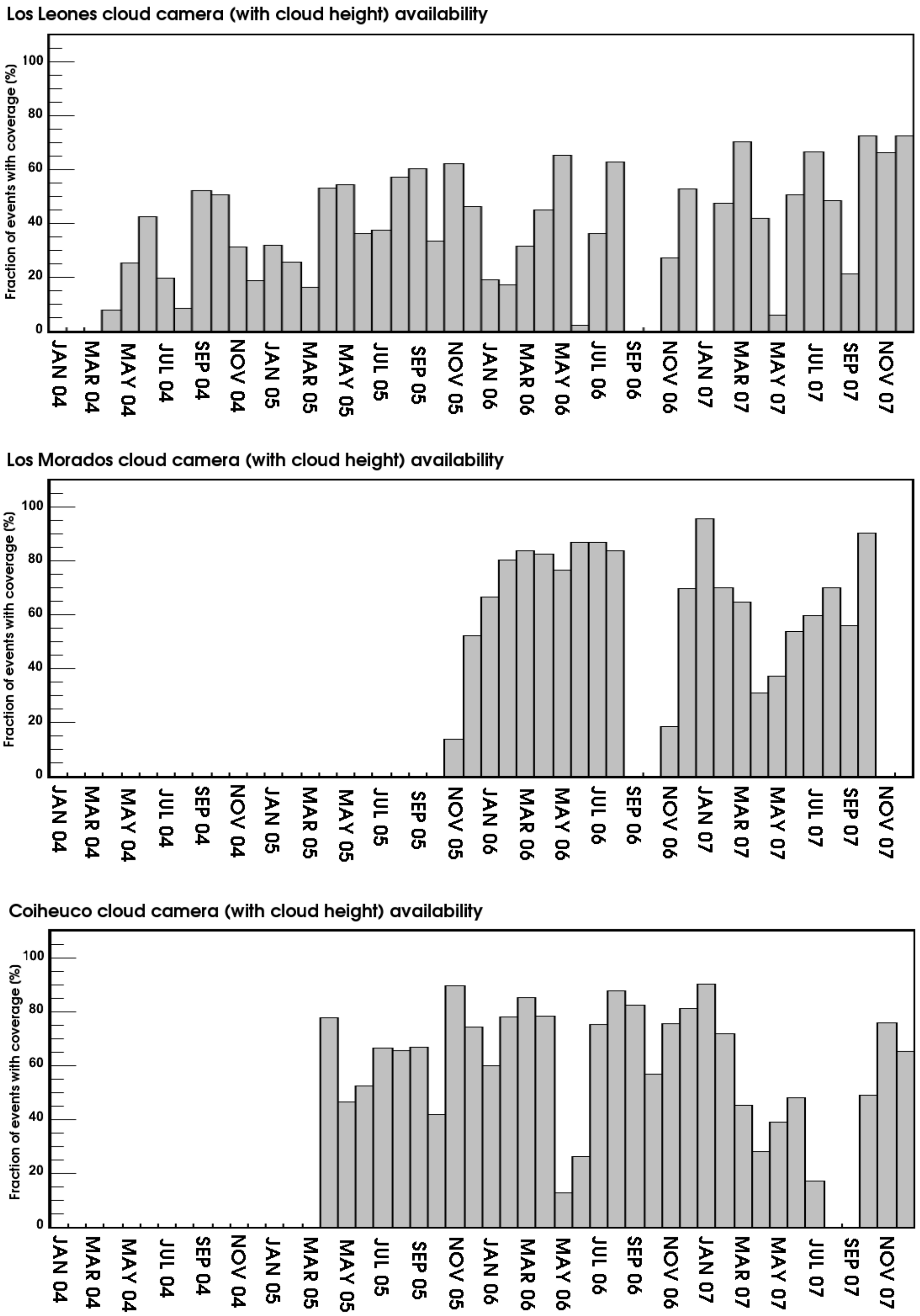


Figure 6.4: *Plots describing the availability of sufficient cloud camera and height data for selected hybrid events.*

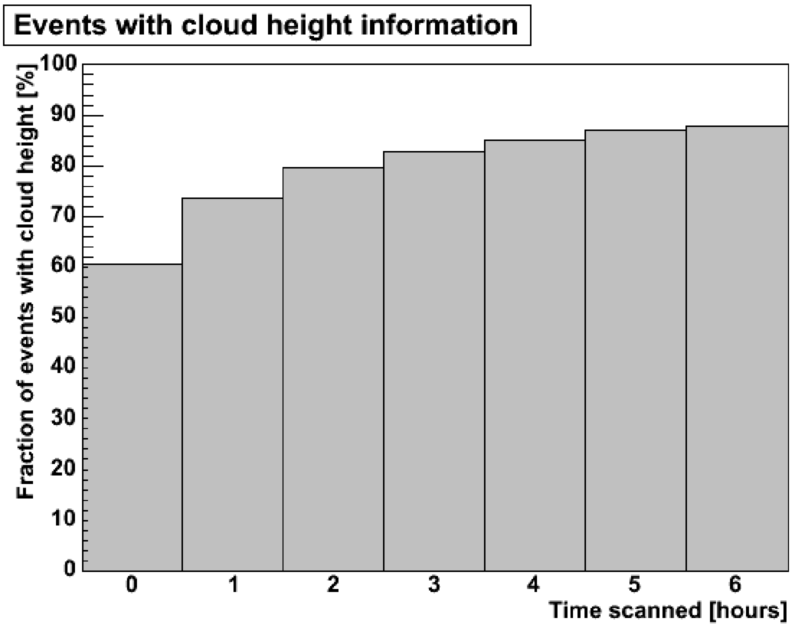


Figure 6.5: *Plot shows the fraction of events with cloud height information if cloud height information is used from neighbouring periods of time. For example: approximately 74% of selected events have cloud height information within 1 hour of the event.*

Cloud height measurement coverage may be extended by using other measurements made at around the same time in question. Often, even if no cloud measurement is available for a particular period of time, a measurement is available in a neighbouring period as seen in figure 6.5. By using cloud height information from within 6 hours of an event, the fraction of events with cloud height coverage increases approximately 25%. Cloud conditions can vary quite significantly over the course of the night however, so some sort of safety margin may have to be introduced when using cloud heights measured several hours distant - depending on how rapidly cloud height can change.

6.3 Cloudy event removal by quality cuts

Clouds may obscure or scatter extra light from an extensive air shower towards a fluorescence detector. This results in unusual ‘dips’ or ‘bumps’ in the measured shower longitudinal profile that may cause that measurement to be discarded by existing quality cuts. In this section, a sample of events is taken and checked for cloud contamination using cloud camera and LIDAR information. The effectiveness of some existing quality cuts is then evaluated by studying how the fraction of cloud obscured events changes with the introduction of the cuts.

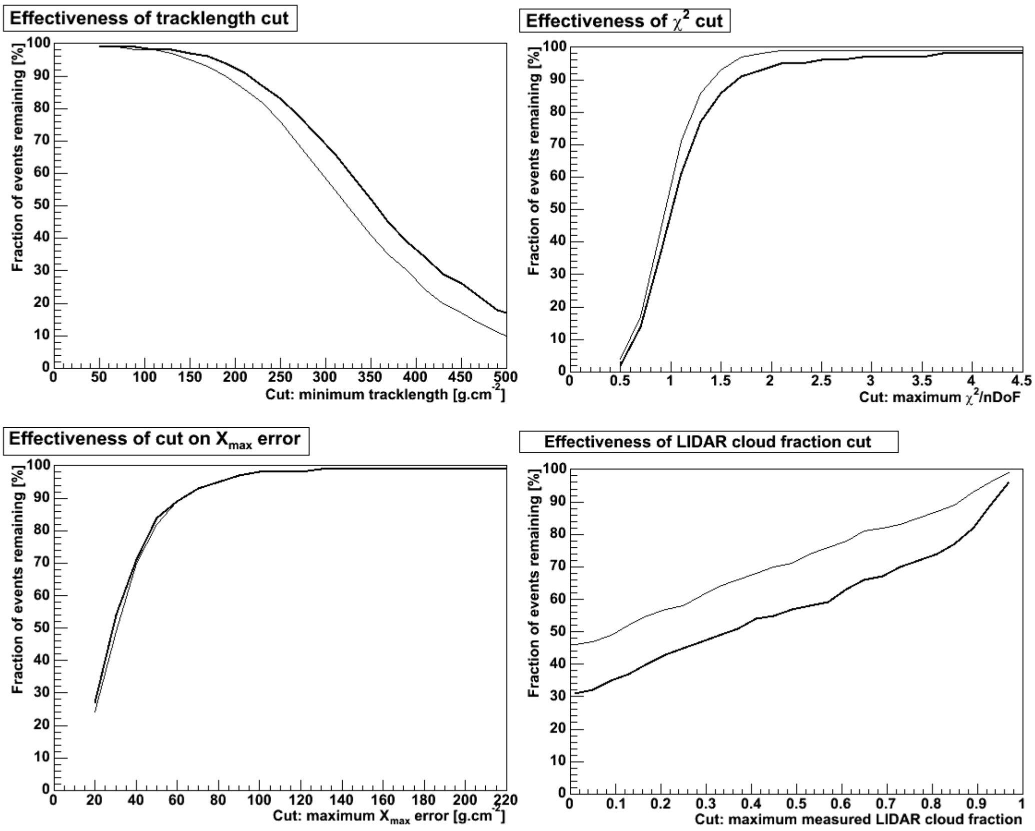


Figure 6.6: *Plots describing the fraction of clear and cloud obscured events remaining after applying different levels of quality cuts. The thick line represents the fraction remaining of cloud obscured events, and the thin line indicates the fraction of cloud-free events remaining.*

Events were selected using the geometry cuts specified in appendix A then used in conjunction with the cloud camera database (and LIDAR) to determine which events were obscured by cloud. The attrition of both cloudy and clear events with increasingly severe quality cuts is shown in figure 6.6, where the fraction of the total events remaining after a particular cut is displayed. Four types of quality cut are examined: firstly is the tracklength cut where the total length of the measured shower longitudinal profile (in g/cm^2) is required to be above some value. Next are the χ^2 and X_{max} error cuts, where the $\frac{\chi^2}{nDoF}$ and X_{max} error of a fitted Gaisser-Hillas profile are required to be below some value. Finally there is the LIDAR cloud fraction cut, where the average (from all available measurements at a particular time) fraction of cloud measured by LIDARs is determined and used to remove events. In addition to the geometry cuts, quality cuts of χ^2/nDoF < 2.5, X_{max} error < 50 g/cm^2 and tracklength > 320 g.cm^{-2} are applied e.g. for the analysis of tracklength, the geometry cuts are applied along with the tracklength and χ^2/nDoF quality cuts.

The tracklength quality cut, in the top-left of figure 6.6, shows a decreasing fraction of events remaining with an increasingly strict (bigger) cut on tracklength. In the range 100-250 g.cm^{-2}, clear events are more likely to be cut than cloud obscured events, but with stricter cuts both types of events are affected equally by this quality cut. This is likely due to the detection of more lower energy showers at larger distances from the fluorescence detector on clear nights, compared with cloudy nights. These low energy and distant showers appear with apparently small tracklengths as they are relatively dim from the perspective of the fluorescence detector, and so will not trigger as many FD pixels as a closer or more energetic shower would. The initial bias towards clear night showers being removed by the tracklength cut, is likely due to the removal of this extra low tracklength population of showers.

With the χ^2/nDoF cut, cloudy showers are more likely to be removed than clear night showers with an increasingly severe (lower) cut. The χ^2/nDoF cut removes showers which are poorly fitted by the Gaisser-Hillas function. The bumps and dips that may be introduced by cloud within a measured longitudinal profile make it more difficult for the Gaisser-Hillas function to accurately described a particular profile. Thus the presence of cloud interfering with an air shower may sometimes manifest itself through a poor χ^2/nDoF fit. This is demonstrated in the top-right plot of figure 6.6. A stricter (lower χ^2/nDoF value) results in a greater fraction of cloud obscured events being cut than clear events. Below a cut of approximately 1.5 this effect diminishes however. This is likely because those events that had their interference by cloud manifest itself in the form of a poor χ^2/nDoF value, have all been cut by this point. Those cloud obscured events that remain all have a χ^2/nDoF value similar to that of clear events, and so, are no longer preferentially cut.

The X_{max} error quality cut removes events that have a larger uncertainty in the reconstructed X_{max} value. It is clear from the bottom-left plot of figure 6.6 that this quality cut does not substantially favor the removal of either clear, or cloud obscured events. A large X_{max} error value is a symptom of a poorly fitting Gaisser-Hillas profile that is, in turn, a symptom of interference by clouds. A χ^2/nDoF cut of 2.5 has already been applied to this set of data however, that has removed most of the poorly fitting, and hence, most of the cloud obscured events already. With the presence of a χ^2/nDoF cut already in place, this quality cut has little effect on removing cloud obscured events from a set of measured extensive air showers.

Cuts via LIDAR measured cloud fraction (seen in the bottom-right of figure 6.6) are effective at preferentially removing cloud obscured events. Cuts down to a LIDAR cloud fraction of 0.8 are seen to preferentially remove cloud obscured events. Below this level the effect is not observed to increase and the cloud fraction cut always removes approximately 16% more obscured events than clear events. The lack of a direct correlation between the fraction of cloud observed by the LIDAR and the actual cloud coverage over the array is likely responsible for the lack of an increase in cut effectiveness at lower cloud fractions. Of the quality cuts, this quality cut is the most effective at preferentially removing cloudy events.

According to the cloud camera data, application of the profile quality cuts mentioned here has limited effectiveness at removing cloud contaminated events. The tracklength cut tends to preferentially remove cloud-free events, while the χ^2/nDoF cut tends to preferentially remove cloud-contaminated events in this sample of data. Application of both these quality cuts may, to some, degree cancel each other out, thereby limiting their effectiveness. The LIDAR cloud fraction cut however, shows a relatively strong tendency to preferentially remove cloud affected events.

6.4 Cloud induced X$_{max}$ measurement bias

Overcast cloud conditions would tend to introduce a bias towards measurement of deeper developing showers by the Auger fluorescence detectors. A number of events were selected using basic geometry, quality and cloud coverage cuts to determine the magnitude and nature of this bias. The magnitude of this bias is apparent when comparing the mean reconstructed X_{max} values for events recorded during clear and cloudy night sky conditions. This has consequences for some mass composition studies, as cloud conditions can make the measured shower sample appear to be of a lighter mass composition (more penetrating) than it otherwise would be.

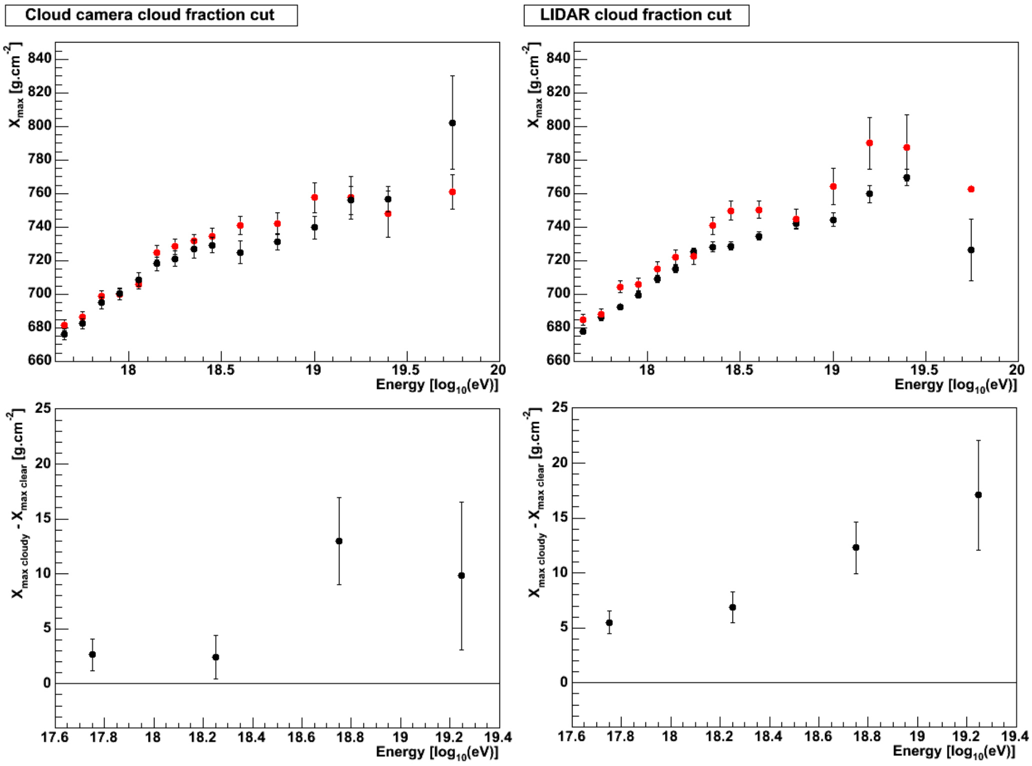


Figure 6.7: *Plots demonstrating difference in mean X_{max} values during clear and overcast conditions.* ***[Top]*** *Red markers indicated the mean X_{max} value derived for a particular energy on cloudy (>80% cloud coverage) nights and black markers are the mean values derived on clear (<20% cloud coverage) nights.* ***[Bottom]*** *Plots showing the difference in mean X_{max} for cloudy and clear conditions for several broad regions of energy.*

Events were selected between the beginning of 2006 and end of 2007 that pass all the quality cuts described in appendices A and B. For each event the average fraction of cloud (calculated using the method in section 5.2) seen by all available cloud cameras was calculated, along with the mean cloud fraction of any available LIDAR cloud measurements. Events without a cloud fraction measurement from either type of instrument were ignored, along with any event that did not occur during very clear (<20% cloud fraction) or very overcast (>80% cloud fraction) conditions. Only events that are thought to be either very clear or very overcast were used in this study. The mean value (with associated standard error) of the measured X_{max} of the selected clear and overcast events are plotted seperately in figure 6.7 as a function of their reconstructed energy.

Cloud fractions measured by the cloud cameras are used in the left column of the figure 6.7 plots, while LIDAR data are used in the right column plots. Both types of measurements suggest a tendency towards deeper mean X_{max} values during cloudy conditions when compared with clear night-sky conditions. The cloud camera and LIDAR cloud fraction measurements describe mean X_{max} shifts of different magnitude. Measurements made with the cloud cameras indicate an overcast sky shows a mean shift of 9 g.cm^{-2}, while similar measurements made by the LIDAR indicate a mean shift of 13 g.cm^{-2}.

The apparent deepening of the mean X_{max} value during very cloudy conditions is a consequence of the lower probability of shallower showers being successfully observed. With large quantities of cloud present, shallower showers are likely to have much of their development obscured by cloud resulting in either a complete failure of observation or a very poor measurement that is subsequently discarded by quality cuts. Deeper developing showers however, are likely to have more of their profile unobscured and subsequently well measured. Therefore after quality cuts, any surviving measurements are more likely to be of deeper developing showers resulting in a bias towards a deeper mean X_{max} value.

The bias towards deeper showers has consequences for mass composition analysis, as events with a proton primary particle tend to have deeper developing showers than heavier primary particles of an equal energy. Thus cloudy conditions may introduce a bias in fluorescence air shower measurements towards showers of lighter compositions. The bias appears (the standard error is greater for the cloud camera measurements) to be larger when using LIDAR cloud fraction information than the cloud camera one, likely because the cloud camera sometimes has difficulty resolving overcast and clear sky conditions and thus the corresponding sample is likely to have some cases where cloud conditions are mis-identified.

Currently, this bias is being accounted for via a quality cut made on the LIDAR measured cloud fraction [162], as the database containing cloud camera data was not available at the time of that analysis. This involves vetoing any measured event that occurred during a >20% (LIDAR measured) cloud fraction measurement. The cut removes approximately 20% of the events from the data sample and was chosen as a good compromise between improving data quality, while maintaining statistics. This is a conservative cut in terms of shower profile quality, as the lack of LIDAR cloud spatial information means that it does not take into account whether the detected clouds are actually interfering with a particular shower or not. Measurements from the cloud camera give the spatial information for such interference to be detected however.

6.5 Pixel level veto

The cloud camera based pixel veto attempts to improve the reconstructed parameters of measured air showers that have been affected by cloud. Cloud camera data is used to identify grammage bins[2] within a measured longitudinal profile that are obscured by cloud and remove them to improve the accuracy of reconstruction. With the removal of particular grammage bins from a profile, the fitted Gaisser-Hillas function (and so hence reconstructed energy and X_{max}) changes. The overall effectiveness of the technique is limited however, as an increase in uncertainty of parameters reconstructed from a pixel vetoed shower is observed.

6.5.1 Methodology

The pixel veto involves removing grammage bins from a measured extensive air shower that have been, according to the cloud camera, obscured by cloud. Excessive light or lack thereof measured from a shower due to cloud interference affects the fit made by the Gaisser-Hillas (GH) function to the longitudinal profile of a shower and so its reconstructed energy and X_{max}. By removing cloud obscured parts of the shower from consideration, the goodness-of-fit of the GH profile may be improved, and hence the reconstructed energy and depth of maximum.

Fluorescence and Cherenkov light from extensive airs showers may be scattered by cloud. Therefore the measured longitudinal profile of an air shower passing through, or behind, cloud may contain unusual peaks or dips due to cloud scattering either extra light towards or away from the Auger fluorescence detectors. These abnormalities sometimes manifest in a form of a poor χ^2 fit for the GH function in the profile, therefore the most badly affected showers are vetoed by the common $\frac{\chi^2}{nDoF} < 2.5$ quality cut. By using the cloud camera to locate and remove the anomalous regions of the shower, the χ^2 of the fitted GH function may be improved and so increase the number of showers available for analysis.

The veto is carried out by identifying any grammage bins contaminated by cloud and removing them. Within the *FdApertureLightFinder* class

[2] A 'grammage bin' is a measurement (with associated error) of the energy deposited by an extensive air shower at a particular depth. It is calculated by combining whatever measurements were made by the fluorescence detector around that depth.

of the Auger Offline reconstruction software, changes were made to the *FindLightFlux* and *FindZeta* functions. For each value of zeta[3] (ζ) tested in the *FindZeta* function, the contribution from any grammage bins with a ζ angle that includes any obscured pixels are ignored. The best ζ angle identified by the modified *FindZeta* function is then used in the *FindLightFlux* function, where any grammage bins that contain cloud obscured pixels are discarded.

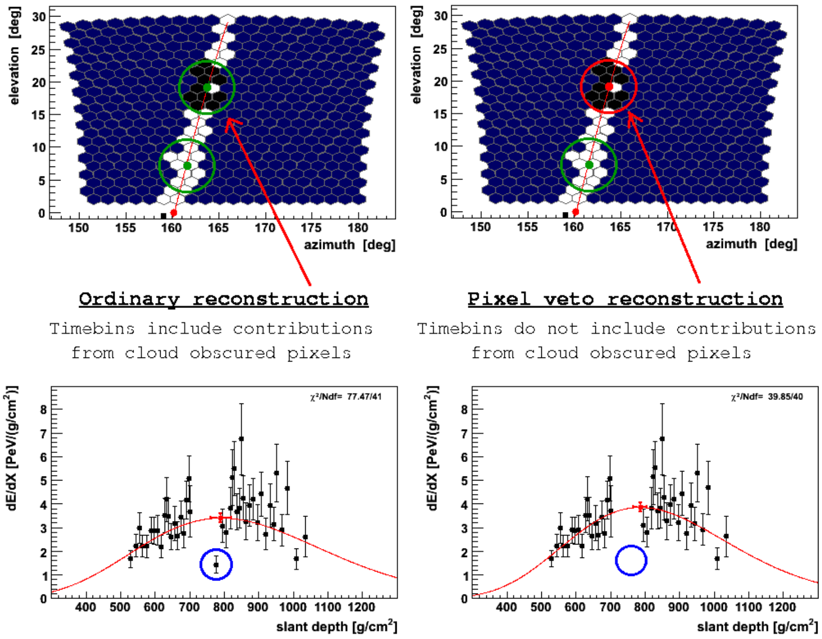


Figure 6.8: *Demonstrating the method of the pixel veto. Top plots demonstrate triggered pixels for the event. Blue are untriggered pixels, white are triggered pixels and black are triggered pixels that are obscured by cloud. Green dots surrounded by green circles represent pixels sampled for a particular grammage bin, while red ones are grammage bins that have been discarded due to the veto.*

This process is demonstrated in figure 6.8 where a dip belonging to a cloud obscured region of an air shower is identified and removed by the pixel veto process. It is important that whole grammage bins are removed in the veto process - not just the contribution from cloud obscured pixels. If only the contribution from cloud pixels is ignored then some grammage

[3] The zeta angle determines which FD pixels are included in the final reconstruction of an extensive air shower. A small angle means that only the pixels close to the shower detector plane of the extensive air shower are used in the analysis. A larger angle means pixels pointing further away from the shower detector plane are also used.

bins will be underestimating the amount of light in the shower, because they would only be measuring part of the light which would otherwise be seen if the cloud were not present. This results in dips in the profile remaining, and in some cases becoming worse.

The pixel veto is a conservative cut on cloud conditions due to potential variation of the height of different clouds at a given time. Cloud height information is gathered from the CLF and LIDAR sites from a relatively (when compared with the size of the surface detector array) small area of sky. Unless a particular cloud detected by the camera is being directly observed by either LIDAR or CLF, its height is being estimated from measurements of other distributions of cloud. The pixel veto analysis uses only the lowest cloud height reported at a particular time in an attempt to be conservative, since if multiple cloud heights are being measured, there is no way of knowing which cloud in the cloud camera image corresponds to which measured cloud height. This means that sometimes a shower that falls in front of a bank of cloud will be erroneously marked as being cloud obscured, as the CLF or one of the LIDARs happened to record a cloud height much lower than that of the cloud in question.

6.5.2 Shift in reconstructed parameters

Applying the cloud camera based grammage bin veto will cause a shift in the reconstructed parameters of a shower if parts of that shower are found to be obscured by cloud. In this section the grammage bin veto is applied to a sample of events in 2006 and 2007 that passed the appendix A geometry quality cut. By examining the shift in parameters as a result of an application of the grammage bin veto, an appreciation may be gained for the added uncertainty as a result of the veto.

Hybrid events were selected from the period between the start of 2006 and the end of 2007. A total of 74748 events were selected using the geometry quality cuts seen in appendix A. Using the grammage bin veto described in section 6.5.1, these events were re-processed to remove any cloud contamination reported by the LIDAR and cloud cameras. 52609 of the 74748 events had cloud camera and cloud height information, and of these, only 11989 were reported as having parts of the shower track obscured. Only 1226 of the cloud obscured events passed the both the geometry (appendix A) and profile (appendix B) quality cuts before and after application of the cloudy pixel veto. Removing parts of a shower with the veto, and reconstructing an event results in a shift of the reconstructed parameters as seen in figure 6.9, which shows the shift of several parameters for those cloud obscured events that passed the quality cuts both before and after the veto process.

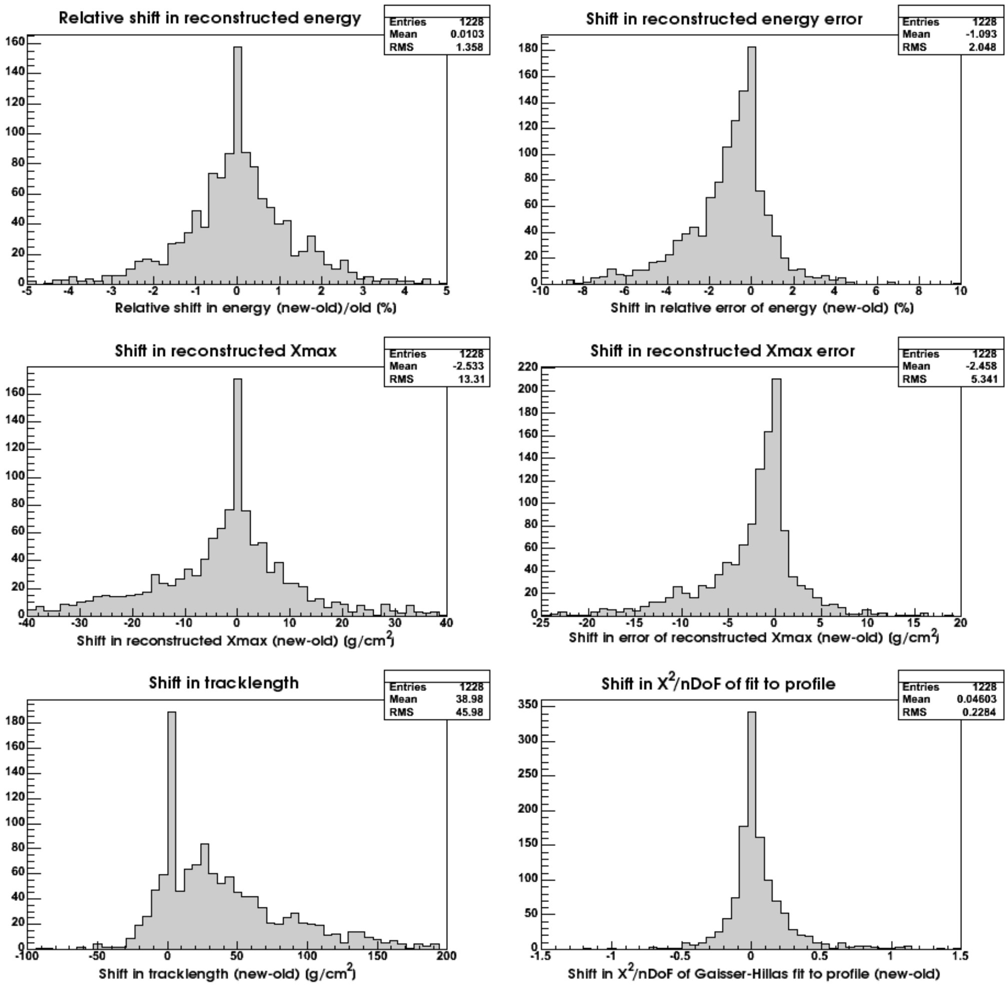


Figure 6.9: *Plots showing shifts in several reconstructed parameters as a result of the grammage bin veto. Events with cloud camera and cloud height data available, that passed the quality cuts described in appendices A and B both before and after the grammage bin veto was applied, were used in these plots.*

Energy is calculated from an FD measurement of a shower by integrating the Gaisser-Hillas function fitted to the energy deposit profile. For most of the events used in figure 6.9 there is less than a 1.5% shift in energy as a result of the cloud camera grammage bin veto. This is to be expected, as a large shift in energy would only be expected if the fitted Gaisser-Hillas function saw a large shift in the fitted $(\frac{dE}{dX})_{max}$ (energy deposited at X_{max}) value. A large shift would only occur if the fit was, in the first place, being strongly influenced by a strong well defined peak or dip in the profile. Any strong modulation in the profile would result in a higher $\frac{\chi^2}{nDoF}$ value that is likely to have already been cut via quality cuts already applied to this set of events.

An average increase of 1% is observed in the error of the reconstructed energy (in the top-right plot of figure 6.9) This is likely due to the tendency for fewer points to be available during the Gaisser-Hillas fit, sometimes leading to a less well defined profile. The lack of a distinct positive offset in this distribution indicates that the cloud veto generally does not substantially decrease the error in the reconstructed shower energy.

The reconstructed depth of shower maximum (X_{max}) is also affected by the grammage bin veto. X_{max} shows a slight tendency to increase after applying the cloud veto, as seen in figure 6.9. This is to be expected as a shower is more likely to be vetoed in its shallower regions, since a deeper veto requires the shower to be of higher energy or the presence of cloud lower in the atmosphere. Cutting the shallower part of a shower usually deepens the fitted X_{max}, while cutting the deeper part results in a shallower fitted X_{max}.

Much like the shift in reconstructed energy error, a slight increase in the uncertainty of the reconstructed X_{max} value is observed with the cloudy pixel veto. This is, once again, due the fewer points available for the fitting of a profile.

Tracklength will decrease or (occasionally) very slightly increase as a result of the grammage bin veto, as seen in figure 6.9. Grammage bins removed by the veto on either side of the shower profile will obviously decrease the observed tracklength of the shower. Removing grammage bins within the center of the profile will sometimes slightly increase the tracklength of the shower however. This change is a result of a stage in the reconstruction (the *CombineAndFillFluxes* function in the *FdApertureLightFinder* class) of the shower where grammage bins with low signal are combined until a large enough signal/noise ratio is achieved to create new grammage bins. If the *CombineAndFillFluxes* function encounters a hole in the profile (such as that introduced by this cloud camera veto) then this combination process stops and begins anew on the other side of the hole. The atmospheric depth assigned to each particular grammage bin depends upon the effective depth of the grammage bins used to create it. Thus introducing a hole means different grammage bins are being combined together, which results in the final grammage bins appearing in subtly different spots which will sometimes result in an apparent slight increase in total shower tracklength. This is also why the quantity of energy deposited in each grammage bin may shift slightly with the veto, or why the same number of grammage bins remain even after cutting out data with the veto - as seen in figure 6.9.

Figure 6.9 shows that the goodness-of-fit of a GH function fitted to the longitudinal profile of an extensive air shower can be both improved and damaged by the cloud camera based grammage bin veto. A decrease indicates that grammage bins have been removed that improved the fit of the Gaisser-Hillas function to the profile. An increase indicates that grammage bins that were relatively well fitting were cut while poorer fitting grammage bins were left alone.

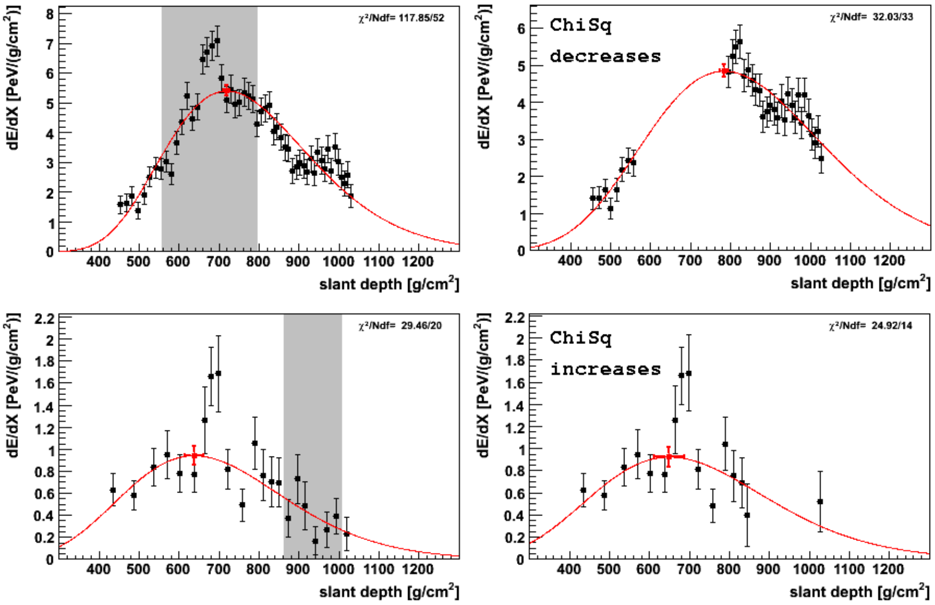


Figure 6.10: *Plots demonstrating how the χ^2 (of an air shower's Gaisser-Hillas fit) may be modified through the application of the grammage bin veto. Grey areas within the plots represent areas of the shower which are obscured according to the cloud camera and LIDAR. [**Top**] Decreasing χ^2 of Gaisser-Hillas fit. [**Bottom**] Increasing χ^2 of Gaisser-Hillas fit.*

The removal of cloud obscured grammage bins is demonstrated in figure 6.10, where in the top plot a selection of poorly fitting grammage bins were removed, resulting in a better fit. In the bottom plot, a selection of well fitting grammage bins were removed while a small bump located around the X_{max} is ignored. A slight increase after the grammage bin veto does not necessarily mean that there exists some cloud effect that the veto has missed however, as even in well measured air showers randomly removing a good fitting point (relative to the others) may cause an increase if it is slightly more poorly fitting than the others. It is to be expected that in some instances the grammage bin veto will incorrectly remove non-cloud affected shower regions as it is a conservative cut, as mentioned earlier. Another possibility is that since clouds are not stationary with time, images taken by the cloud camera may have been 'out-of-date' and conditions may have been clearer than suggested by the cloud camera.

6.5.3 Effectiveness

There are several different applications of the cloud camera based grammage bin veto. In order to investigate these applications, events recorded during 2006 and 2007 were selected and had the cloudy pixel veto applied to them. Firstly the deviation of vetoed grammage bins from the fitted GH profile are examined, to see if the 'bumps' and 'dips' really are being removed by the veto. Secondly those events that pass both the geometry (appendix A) and profile (appendix B) quality cuts and that are confirmed clear of cloud contamination are examined. Thirdly of interest are those events which initially appear to be of good quality (passing both the geometry and profile cuts) but later degrade in quality after the veto is applied. Lastly are those events which at first are of poor quality (initially failing both quality cuts), but show an improvement after application of the veto and passing the cuts.

Effectiveness at removing poorly fitting grammage bins

A selection of events from 2006 and 2007 were examined in order to study the goodness-of-fit of those grammage bins that are being removed by the veto. These events pass both the geometry and profile cuts descibed in appendix A and B with the exception of the χ^2/nDoF cut. The χ^2 cut is not used, so as to attempt to preserve any bumps or dips in the profiles. In addition, only those events that were affected by the cloudy pixel veto (having grammage bins removed from their analysis) were considered in this study.

Figure 6.11 describes the individual goodness of fit of grammage bins left and grammage bins cut as a result of the cloud camera veto. The goodness of fit of each individual grammage bin is described by its deviation, which is simply the difference between it and the predicted Gaisser-Hillas fit divided by the uncertainty associated with that grammage bin - as seen in equation 6.2.

$$Deviation = \frac{\frac{dE_{grammage}}{dX} - \frac{dE_{predicted}}{dX}}{\sigma} \tag{6.2}$$

where $dE_{grammage}$ and $dE_{predicted}$ are the measured energy deposit, and the predicted energy deposit (through the Gaisser-Hillas fit) for the grammage bin in question. σ is the error in the energy deposit in that grammage bin.

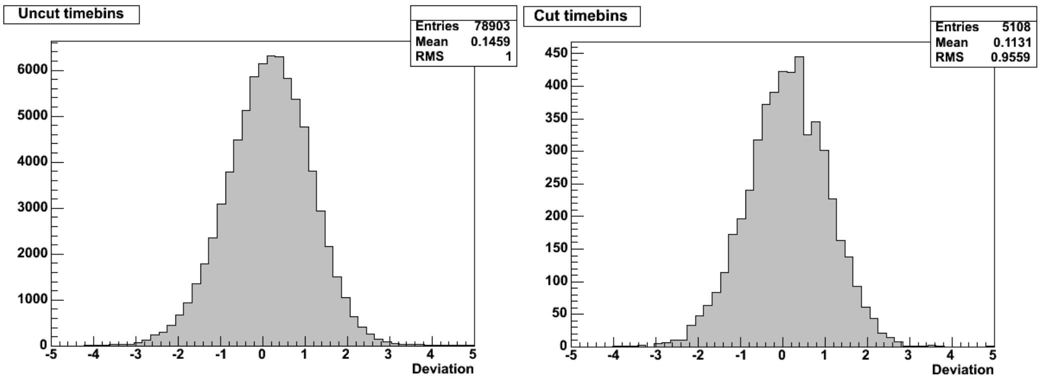


Figure 6.11: *Plots describing the deviation from the Gaisser-Hillas fit of grammage bins from events (obscured in some way by cloud) selected from March-June 2006 that survived the quality cuts in appendices A and B. **[Left]** Showing distribution for grammage bins that were cut. **[Right]** Showing distribution for grammage bins that were not cut in events obscured by cloud.*

The plots in figure 6.11 describe the deviation of grammage bins that have both been preserved, and cut by the cloudy pixel veto. Overall about 6% of the grammage bins in cloud affected events were removed by the veto. Both distributions are approximately centered at zero, as is expected since they were used to derive the Gaisser-Hillas fit they are being compared against. Both distributions also have approximately the same mean and RMS values. This suggests that there is no substantial difference between the two distributions, and indicates that the cloudy pixel veto is removing a substantial number of well-fitting (and presumably) cloud free pixels.

This ineffectiveness is likely due to the conservative nature of the veto. The cloud camera veto assumes all cloud coverage is at the lowest cloud height measured at the time, which means that occasionally the cloud camera is going to remove a perfectly good piece of the shower profile because it underestimated its height - an example of this is the bottom event in figure 6.10 in the previous section. Even when a bump is successfully targeted by the veto, often some well-fitting grammage bins are also removed as a side effect - also evidenced in figure 6.10, in the top-most profile.

Events that no longer pass profile quality cuts after application of the veto

Of interest are those events which were initially considered to be 'good' by the profile quality cuts (appendix A), but later fail those same quality cuts after application of the cloudy pixel veto. Given the apparent inefficiency of the cloud veto at identifying cloud obscured track pixels, it is expected that a number of events will have perfectly well fitting GH profile grammage bins removed by the veto.

Of those showers that passed the geometry quality cuts (appendix A) selected during 2006 and 2007, 8447 events had reconstructed energies exceeding 10^{18}eV. Of these, 5598 (66%) had enough information from the cloud cameras and LIDAR to apply the cloud veto, 3836 of these events were determined to be cloud-free and 1762 partially obscured by cloud. Of the cloud-free events, 1510 (40%) passed the profile quality cuts (appendix B), and 791 (45%) of the cloud obscured events passed the same cuts. After application of the pixel veto only 466 (27%) of the cloud obscured events still passed the profile quality cuts.

A greater fraction of the cloud obscured events initially passed the profile quality cuts, than the non-cloud obscured events. This suggests one of three things: firstly, that the effects of cloud upon a measured shower profile are not significant. Secondly, that the cloud cameras are inefficient at identifying cloud affected events. Or thirdly that the profile quality cuts used are ineffective at removing cloud contaminated events.

The most likely explanation is a combination of the inefficiency of the cloud veto at identifying cloudy events, and the ineffectiveness of cloudy event removal by existing quality cuts. As mentioned in the previous section, the veto uses a conservative estimate for the cloud height that may result in identified cloud being thought closer than it actually is. In figure 6.9 it was shown that (with the exception of the LIDAR cut which is not used here) the usual quality cuts are not particularly effective at removing cloud contaminated events - although this may tie back in with the inefficiency of the veto, as cloud camera data was used to arrive at this conclusion.

A total of 325 events from the sample of events that initially passed the appendix A and B quality cuts (and were $> 10^{18}$eV) subsequently failed those same cuts after applying the cloud camera grammage bin veto. Most of these failed because the search for the best zeta value failed during reconstruction (presumably because too much of the shower was removed). Those that succeeded in finding a best zeta value typically failed due to either the tracklength or X_{max} error quality cut.

The profiles of five of those events that failed the profile quality cuts after the cloudy pixel veto, are shown in figure 6.12. It is interesting to note that none of the profiles exhibit any obvious bumps or dips typically associated with cloud intereference.

It is clear that application of the cloudy pixel veto can remove enough information to cause an otherwise well-measured event to fail subsequent quality cuts. This is due to fewer grammage bins being available, resulting in a shortening of the observed tracklength, and a more poorly fitted Gaisser-Hillas profile. Application of the cloudy pixel veto to all events within a set of data may remove some cloud obscured events, but will also remove some perfectly good ones at the same time.

Events that fail profile quality cuts, but pass after application of the veto

A total of 61 events during 2006 and 2007 initially failed the appendix A and B quality cuts, but later passed after application of the cloudy pixel veto. Typically these events failed the profile quality cuts due to a poor χ^2/nDoF fit or large X_{max} error. Application of the cloud veto, in these cases, results in an improved Gaisser-Hillas fit to an extent where they now pass those quality cuts.

Three of these events are shown in figure 6.13. All three events initially failed the X_{max} error cut, and one failed the $\frac{\chi^2}{nDoF}$ cut. Of the events, only the top-most one appears to have a had a cloud induced effect removed (the apparent Cherenkov splash at 830 g.cm^{-2}), and it is also this one which shows the most marked improvement in $\frac{\chi^2}{nDoF}$ - with a decrease from 4.6 down to 0.7.

The large decrease of $\frac{\chi^2}{nDoF}$ with the veto is a result of several grammage bins which strongly disagreed with the fitted GH profile being removed. For the other events only a slight (0.6 and 0.1 for the middle and bottom events respectively) improvement in the $\frac{\chi^2}{nDoF}$ observed. This makes sense, because in those two of the three events the parts of the shower which were removed tended to agree relatively well with the Gaisser-Hillas fit already, but in the case of the cloud affected event, a portion of the shower that disagreed quite strongly with the fit was removed - resulting in the superior change in $\frac{\chi^2}{nDoF}$ value.

Evidently, shower profiles may be improved to an extent where they can start passing quality cuts (in some cases) through the removal of some grammage bins. To evaluate whether a ‘bad’ part of the shower is actually

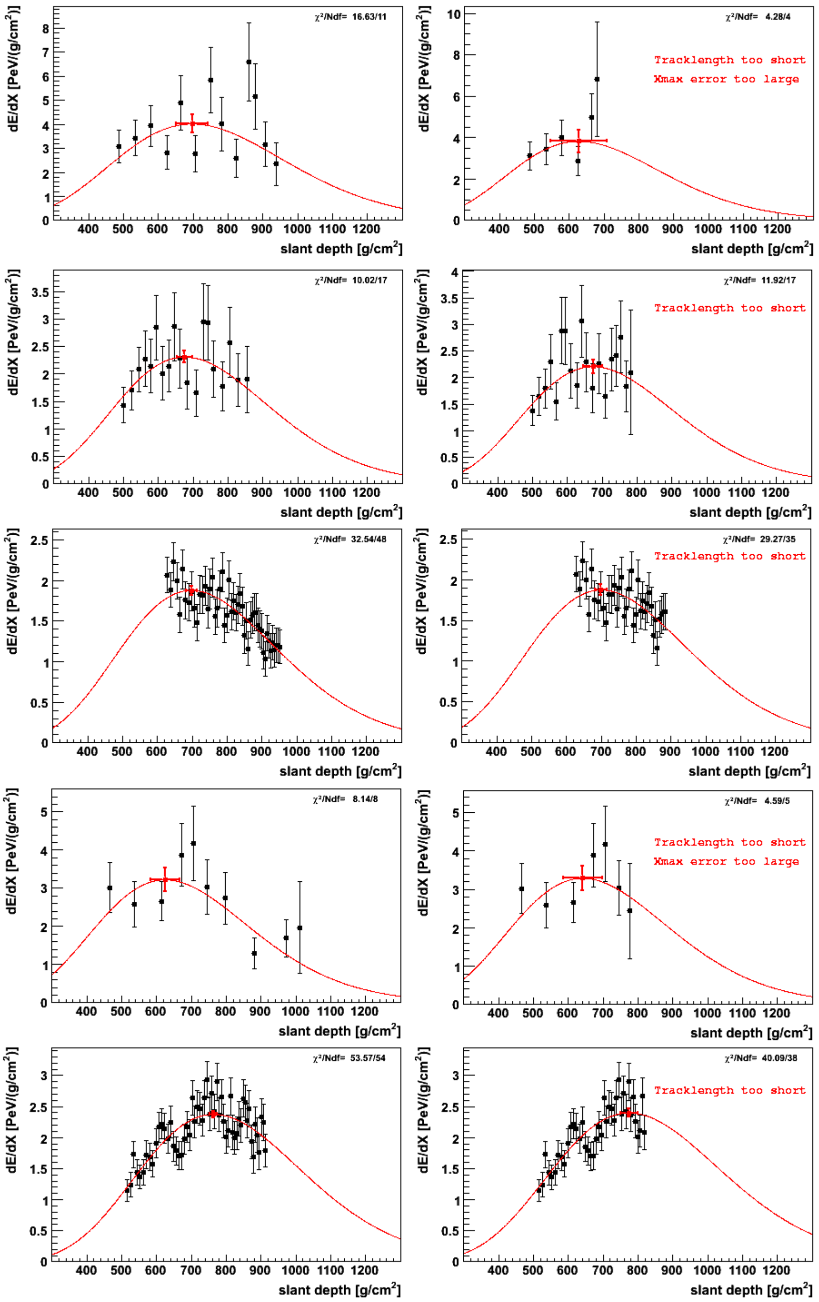


Figure 6.12: *Events* $> 10^{18}$ *eV selected from March-June 2006 that initially passed the quality cuts in appendices A and B, but later failed those same cuts after undergoing the cloud veto. Plots on the left are the normal reconstructed profiles, plots on the right are the same profiles reconstructed, but with the cloud veto applied.*

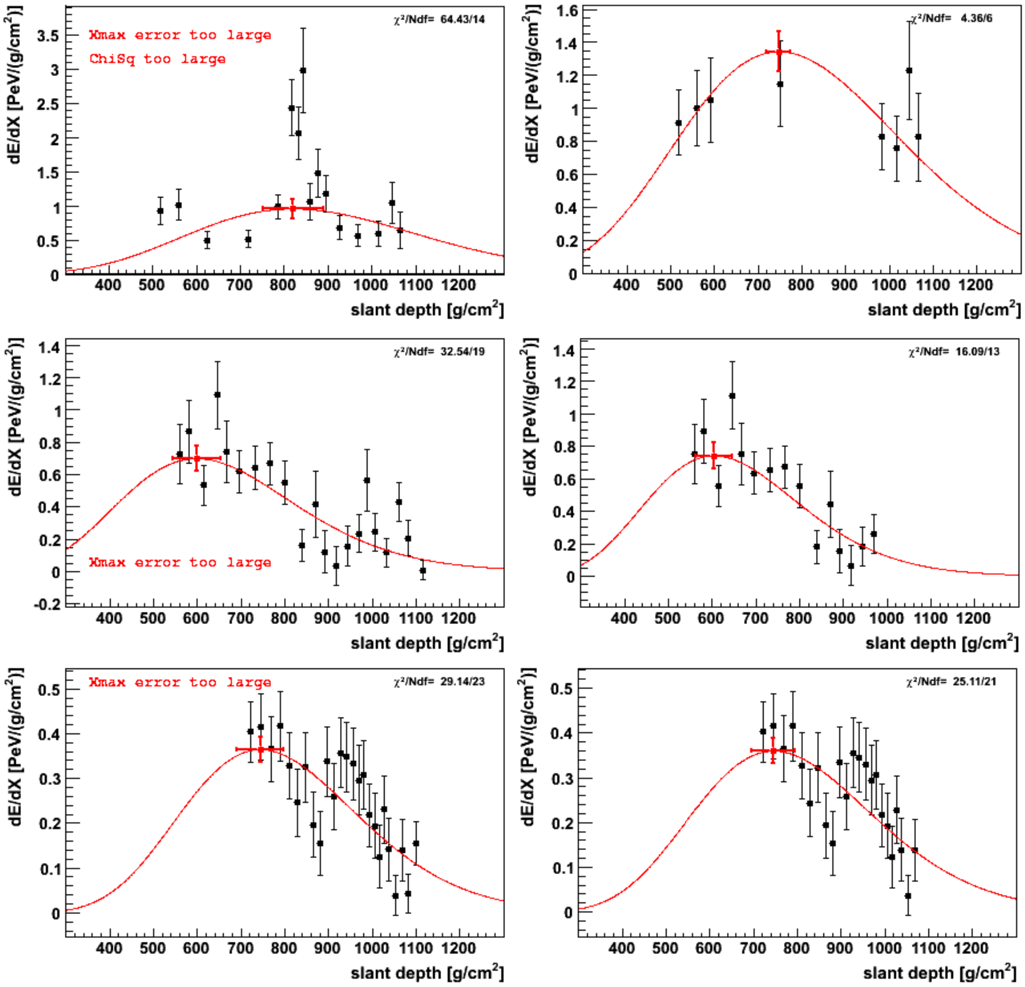


Figure 6.13: *Three events during 2006 and 2007 that initially did not pass the quality cuts in appendix A and B, but later passed those same cuts after undergoing the cloud veto. Plots on the left are the normal reconstructed profiles, plots on the right are the same profiles reconstructed, but with the cloud veto applied.*

being removed by the veto the change in $\frac{\chi^2}{nDoF}$ might be examined, as a large improvement corresponds to poorly fitting grammage bins being removed, and a slight improvement to the removal of well fitting grammage bins.

A significant number of grammage bins that agree well with a fitted Gaisser-Hillas profile are being removed by the cloud camera based grammage bin veto cut. This is to be expected, given the (necessarily) conservative nature of the cut. It is observed however, that if these bins are incorrectly (they are not actually obscured by cloud) removed it can either lead to an apparent improvement or worsening in event quality that may affect whether it passes a given series of quality cuts. Thus the cloud pixel veto cannot be completely trusted in 'improving' event reconstruction quality. A measure of the degree of trust that can be associated with an application of the grammage bin veto to a particular event is the change in $\frac{\chi^2}{nDoF}$. A small change suggests the veto is being conservative, while a large change may correspond to a correct application. If the grammage bin veto is to be used its effectiveness must be considered.

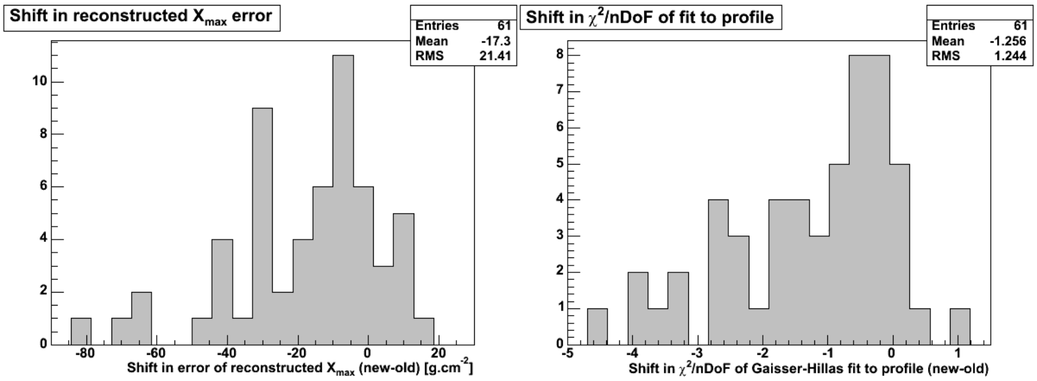


Figure 6.14: *Histograms showing the shift in the χ^2/nDoF parameters and estimated error of the reconstructed X_{max} and χ^2/nDoF, for events that only passed the appendix A and B quality cuts after the cloudy pixel veto.*

Within figure 6.14 is shown the shift in the fitted GH χ^2/nDoF and estimated error in reconstructed X_{max} for those events that initially failed the appendix A and B quality cuts, only to pass them after the cloudy pixel veto was applied. The histograms show that for the majority of events there is an improvement in in the quality of Gaisser-Hillas profile fit. Those events that show a decrease in the quality of fit but manage to pass the quality cuts generally had their tracklength slightly improved (through the process explained in section 6.5.1) by the veto, and already had good X_{max} error and

χ^2/nDoF values.

It may be prudent to evalulate the effectiveness of the cloudy pixel veto on an event-by-event basis by the improvement it delivers to the fit of the GH function to the measured longitudinal profile. A quality cut based upon the shift in the error of the reconstructed X_{max} or χ^2/nDoF may served this purpose. For example, only apply the cloudy pixel veto to those events which show a minimum of a 30 g.cm^{-2} improvement in the X_{max} error, or an improvement of at least 2 to the χ^2/nDoF value.

If such a quality cut were to be applied in conjunction with the cloudy pixel veto approximately 15 events during 2006 and 2007 may be 'salvaged' with the quality cuts described here. This would give less than a 1% increase to the size of the data sample available for analysis.

6.6 Summary

In this chapter was discussed the practical application of cloud camera information to FD measured extensive air shower data. Cloud camera data only provides directional information, it does not measure how far away a particular bank of cloud is from a fluorescence detector. This means that the cloud cameras cannot determine if an observed cloud is in front of, or behind a particular air shower. If the height of a bank of cloud is known, this limitation may be overcome and the actual location of a bank of cloud determined.

Cloud height measurements are available from the LIDARs at each FD site and the CLF, however these measurements are only performed upon the sky directly above each instrument. There is no guarantee that cloud will be above any of these instruments - or even if there is, that it will be at the same height as another bank of cloud seen by the cloud camera. Approximately 60% of events have a cloud height measurement immediately available. It is suggested here that the fraction of events with a cloud height measurement may be increased by using cloud height measurements taken at other times i.e. with measurements taken hours before or after a particular event is observed. Using measurements taken within an hour of an event would increase the events with an associated cloud height to 74%, and within six hours to 88%. More work is required to determine how stable, and thus how accurate these cloud heights would be however.

Cloud camera data combined with cloud height information was used to study the effectiveness of current quality cuts at removing cloud affected events. The fraction of events obscured by cloud (according to the cloud cameras) was calculated as a function of different quality cuts. It was found that while, generally speaking, more cloudy events are removed than clear ones, there are diminishing returns with stricter quality cuts.

Also examined was the shift in mean X$_{max}$ value due to overcast cloud conditions. As the higher altitude portions of air showers are often obscured on overcast nights, there is a tendency to miss the shallower air showers on such occasions. This results in a slight increase in the reconstructed mean X$_{max}$ parameter on overcast nights. LIDAR data indicates an average increase of 13 g.cm^{-2} on overcast nights, while the cloud cameras show a similar result with an average increase of 9 g.cm^{-2}.

Examined in this chapter was a proposed application of cloud camera data to the removal of cloud-obscured FD pixels from the reconstruction of extensive air showers. It was found that while such a 'pixel veto' sometimes greatly improves the reconstruction of an event (removing apparently anomalous parts of a measured longitudinal profile) it also often vetoes apparently good measurements - resulting in a poorer reconstruction of that event. This is likely due to the conservative nature of the cloud height estimate. Because of this, the 'pixel veto' is not recommended to be a part of the standard reconstruction procedure.

If the pixel veto were to be applied to FD data, then it is recommended that some sanity check be in place to minimise any adverse effects. Those events that benefited most from the pixel veto also showed the greatest improvement in their fitted χ^2/nDoF error. Therefore it is advised that the pixel veto only be applied to those events that show such a marked improvement in some quality dependent parameter - such as (for example) at least a shift of -2 in χ^2/nDoF or -30 in X$_{max}$ error (from figure 6.14). Such an addition to the reconstruction procedure is estimated to result in an increase of less than 1% of usable events however.

The cloud camera data is likely to be most useful as an aid in other studies where clouds may play a role. For example, in the past cloud camera data has been used to help determine a limit to the UHE photon flux measured by Auger. Cloud information is also important in the search for more exotic physics. 'Double-bang' showers with two depth of maximums are expected to result from tau neutrino interactions or miniature black hole production in the atmosphere. As cloud affected events may mimic such interactions the cloud cameras may provide a useful aid in the search for such exotic events.

Chapter 7

Conclusion

Successful and reliable identification of clouds within images recorded by the Pierre Auger cloud cameras, require that some estimate of the clear sky background be available. Without an expected clear sky background signal to contrast an image against any cloud detection algorithm will be relying upon any contrasts within the image to identify cloud of interest. This is less important during clear night sky conditions, as the landscape often provides a similar signal intensity to that of cloud. During overcast cloud conditions however, there exists no contrast within the image, making detection of that cloud extremely unreliable without an estimated clear sky signal. One potential way around this is to develop a detection algorithm based upon the texture of cloud within the image, though this approach is not examined within this dissertion.

Parameterizations giving some estimate of the clear sky temperature are available, and have been studied extensively in the past. The cloud cameras are non-radiometric however, which makes direct application of these parameterizations problematic. It was found, however, that it is possible to create an expression that approximates the signal expected for a clear night sky as a function of ambient temperature and effective blackbody sky temperature. This allowed the development of (with the help of one of the aforementioned clear sky temperature parameterizations) a parameterization for the estimated clear sky signal as a function of ambient temperature and ground level water vapor pressure.

The use of this clear sky signal parameterization allowed the creation of two cloud detection algorithms, both of which had superior performance when compared with the more conventional edge detection and thresholding approaches examined here. Neither algorithm had the reliability to provide unsupervised detection of cloud within the images. To ensure high standards, the decision was made to manually check each of the approximately five hundred thousand images, making sure that each image had its cloud detected correctly.

In order to faciliate processing of the large number of cloud camera images a graphical user interface (GUI) was created. This GUI stores images and the location of any cloud located within them. It has integrated into it multiple cloud detection algorithms that allow a user to easily test different strategies for detection of cloud within a particular set of images.

The infrared cloud cameras are the only instrument currently in operation at the Pierre Auger Observatory that provides information about the cloud conditions above the whole array. LIDARs installed at each of the fluorescence detectors and a CLF (Central Laser Facility) are only able to provide information on localized cloud conditions. The limited LIDAR and CLF data may be used as a rough cross-check for the cloud camera data however, and this is examined in chapter 5. A comparison between the cloud cameras and LIDAR/CLF showed strong agreement during very overcast, and very clear sky conditions. Such a comparison is very rough however, and the lack of other comparisons makes it impossible to more accurately quantify how successful the cloud cameras and analysis were at detecting cloud.

Previous studies with the LIDAR have shown a bias towards the detection of deeper penetrating showers during overcast cloud conditions, and cloud camera data supports this result. The statistics of the cloud camera study however, are poorer than that of the LIDAR due to the lower frequency of very clear and very overcast cloud fractions being reported. The tendency towards deeper showers during overcast conditions is due to the shallower showers being obscured by cloud, and thus not observed by Auger's fluorescence detectors.

Using cloud camera data to directly determine whether a measured extensive air shower has been affected by cloud is inadvisable, as the cloud camera only provides cloud directional data. Combining cloud camera data with cloud height information allows the position of cloud relative to a particular extensive air shower to be much more accurately determined.

Approximately 23% of events are reported as being obscured by cloud, but without independent measurements it cannot be known what the fraction actually is. Analysis of those timebins thought to be obscured by cloud, and those not, indicates no significant difference in the 'goodness-of-fit' of timebins with the fitted Gaisser-Hillas function for either distribution. This indicates that any cuts made using this information will be very conservative.

The conservative nature of the cloud obscuration information is due to the lack of availability of a direct measurement for the height of each bank of cloud (in the cloud camera's field of view). Cloud heights must be extrapolated from measurements of the cloud directly above each FD facility by LIDARs or the CLF. In order to be conservative, only the lowest measured cloud height is used. This sometimes results in the cloud being thought closer to the FD than it actually is, and thus the erroneously reported cloud obscuration of an extensive air shower measurement.

The cloudy pixel veto is a quality cut aimed at removing, or improving, those measurements that are influenced by cloud. This quality cut can both improve and degrade (due to perfectly good measurements being ignored) the fit of a Gaisser-Hillas profile to a particular longitudinal measurement. This indicates that it is unwise to blindly apply this quality cut to all extensive air showers, due to the significant reduction in the number of good measurements. As the cut greatly improves the quality of the Gaisser-Hillas fit to some events it is suggested that this quality cut only be applied to those events that show a great deal of improvement with its application. The increase, however, in available events for analysis is small (less than 1%).

Chapter 8

Appendices

A. Geometry quality cuts

SD/FD time offset	$\in$	(0,200) ns	Time fit: $\frac{\chi^2}{nDoF}$	$<$	7.0
Station axis distance	$\in$	(0,2000) m	χ_0	$\in$	(0,180) degrees
Rp	$>$	0 m	θ	$<$	60°
Min. viewing angle	$>$	20°			

Table 8.1: *Geometry cuts.*

B. Profile quality cuts

X_{max} error	$<$	50 $g.cm^{-2}$	X_{max} is bracketted		
Tracklength	$>$	320 $g.cm^{-2}$	Gaisser-Hillas: $\frac{\chi^2}{nDoF}$	$<$	2.5

Table 8.2: *Quality cuts.*

Bibliography

[1] **J.Abraham et al.**, 2001, *Optics and mechanics of the Auger Fluorescence Detector*, Proc. 27th Int. Cosmic Ray Conf. (Hamburg)

[2] **J.Abraham et al.**, 2003, J. Phys. G: Nucl. Part. Phys., 29, 867-879

[3] **J.Abraham et al.**, 2004, Nuclear Instruments and Methods, A523, 50

[4] **J.Abraham et al.**, 2005, *Absolute Calibration of the Auger Fluorescence Detectors*, Proceedings of the 29th ICRC

[5] **J.Abraham et al.**, 2007, Proc. 30th ICRC (Merida), 594

[6] **J.Abraham et al.**, 2007, Nucl. Instrum. Meth., A578, 180

[7] **J.Abraham et al.**, 2007, *Measurement of the UHECR energy spectrum using data from the Surface Detector of the Pierre Auger Observatory*, 30th ICRC (Mexico), arXiv:0706.2096v1 [astro-ph]

[8] **J.Abraham et al.**, 2007, *Reconstruction accuracy of the surface detector array of the Pierre Auger Observatory*, 30th ICRC (Mexico), arXiv:0709.2125v1 [astro-ph]

[9] **J.Abraham et al.** *The Lidar System of the Pierre Auger Observatory*, Nucl. Instrum. Meth., A574:P171-184, 2007

[10] **J.Abraham et al.**, 2008, Astropart. Phys., 29, 188-204

[11] **J.Abraham et al.**, 2008, Phys. Rev. Lett., 101, 061101

[12] **J.Abraham et al.**, 2008, Nucl. Instrum. Meth., A588, 433

[13] **J.Abraham et al.**, 2009, *Measurement of the cosmic ray energy spectrum above $10^{18} eV$ using the Pierre Auger Observatory*, Proceedings of the 31^{st} ICRC, Lodz

[14] **J.Abraham et al.**, 2009, Nuclear Physics B (Proc. Suppl.), 190, 12-19

[15] **M.Abramowitz and I.A.Stegan** *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*, p16, US government printing office, Washington, 1964

[16] **M.Aglietta et al.**, 1999, Proc. 26th Int. Cosmic Ray Conf. (Utah), HE 1.3.04

[17] , **B.Afanasiev et al.**, 1996, Proc. Int. Symp. on EHE Cosmic Rays, 32

[18] **L.Alvarez et al.**, 1933, Phys. Rev.,43, 835

[19] **M.Ambrosio et al.**, 2002, Nucl. Instrum. Meth., A478, 125-129

[20] **T.Antoni et al.**, 2005, Astropart. Phys., 24, 1

[21] **F.Arqueros et al.**, 2008, *Inferring Xmax from asymmetry in risetime*, Auger Technical Notes, GAP-2009-179

[22] **D.Arthur et al.**, 2007, Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms, 1027-1035

[23] **K.Asakimori et al.**, 1991, Proc. 22nd Int. Cosmic Ray Conf. (Dublin), 2, 57

[24] **K.Asakimori et al.**, 1991, Proc. 22nd Int. Cosmic Ray Conf. (Dublin), 2, 97

[25] **P.Auger et al.**, 1938, C. R. Acad. Sci.(Paris), 206, 151

[26] **P.Auger et al.**, 1939, Rev. Mod. Phys., 11, 288

[27] Auger Project Design Report, http://www.auger.org/technical_info/design_report.html

[28] http://www.auger.org

[29] **M.Ave et al.**, 2000, Astropart. Phys., 14, 91

[30] **M.Ave et al.**, 2007, 30th International Cosmic Ray Conference (Mexico), arXiv:0709.2125 [astro-ph]

[31] **W.Axford et al.**, 1977, Proc. 15th Int. Cosmic Ray Conf. (Plovdiv), 11, 132

[32] **W.Baade et al.**, 1934, Proceedings of the National Academy of Science, 20, 259

[33] **R.Baltrusaitis et al.**, Phys. Rev. Lett., 52, 1380

[34] **H.Barbosa et al.**, 2004, Astropart. Phys., 22, 159-166

[35] **P.Bhattacharjee et al.**, 2000, Phys. Rep., 327, 109

[36] **A.Bell**, 1978, Mon. Not. R. Astron. Soc., 182, 443

[37] **A.Bell**, 1978, Mon. Not. R. Astron. Soc., 182, 147

[38] **J.Bellido et al.**, 2001, Astropart. Phys., 15, 167-175

[39] **P.Biermann**, 1997, J. Phys., G23, 1

[40] **D.Bird et al.**, 1993, Proc. 23rd Int. Cosmic Ray Conf. (Calgary), 2, 38

[41] **R.Blandford et al.**, 1978, Ap. J. Lett., 221, L29

[42] **M.Block et al.**, 2000, Phys. Rev. D, 62

[43] **C.Bonifazi et al.**, 2005, 29th International Cosmic Ray Conference (India), 7, 17

[44] **J.Brack et al.**, 2004, Astropart. Phys., 20, 653-659

[45] **L.Brown et al.**, 1992, Phys. Rev. D, 45

[46] **W.Brutsaert**, 1975, Water Resource Research, 11, 742

[47] **J.Canny**, 1983, *Finding edges and lines in images*, MIT Artif. Intell. Lab., Cambridge, MA, TR-720

[48] **J.Carlson et al.**, 1937, Phys. Rev., 51, 220

[49] **G.Cassiday**. 1985, Ann. Rev. Nucl. Part. Sci, 35, 321

[50] **E.Châtelet et al.**, 1991, Proc. 22nd Int. Cosmic Ray Conf. (Dublin), 2, 45

[51] **X.Chi et al.**, 1990, J. Phys. G., 16, 1409

[52] **N.Chiba et al.**, 1992, Nucl. Inst. Meth., A311, 338

[53] **T.Chiyu et al.** *The Moving Velocity of Cumulus Humilis Clouds*, J. Meteor. Soc. Japan, 51, p43-53, 1973

[54] **A.Cillis et al.**, 2001, Nucl. Phys. Proc. Suppl., 97, 239-242

[55] **G.Clark et al.**, 1957, Nature, 180, 353

[56] **G.Clark et al.**, 1968, Ap. J. Lett., 153, 203

[57] **G.Clark et al.**, 1978, Proceedings of the second national passive solar conference, Philadelphia, Part 2, 676

[58] **J.Clay**, 1927, Proc. Roy. Acad. Amsterdam, 36, 1115

[59] **R.Clay et al.**, 1982, Aust. J. Phys., 35, 59

[60] **R.Clay et al.**, 1998, PASA 15, http://www.atnf.csiro.au/pasa/15_2

[61] **G.Cocconi et al.**, 1946, Phys. Rev., 70, 852

[62] **G.Cocconi**, 1961, Handbuch der Physik, Springer-Verlag, 215

[63] **A.Compton et al.**, 1935, Phys. Rev., 47, 817-821

[64] **J.Cronin**, 2005, Nucl. Phys. B (Proc. Suppl.), 138, 465

[65] **P.Davidson** *Studies of a Pyroelectric-Based Infrared Camera*, Honours Thesis, Adelaide University, p37, 2002

[66] **J.Daudin**, 1945, Ann. de Phys., 20, 563

[67] **B.Dawson**, 1985, PhD thesis, Adelaide University

[68] **E.Deacon**, 1970, Q. J. R. Met., 96, 313

[69] **M.Dova et al.**, 2009, Astroparticle Physics, 31, 312-319

[70] **N.Efimova**, 1961, Meteorol. Gidrol., 10, 28

[71] **D.Ellison et al.**, 1987, Ap. J., 318, 474

[72] **W.Elsasser**, 1942, *Heat Transfer by Infrared Radiation in the Atmosphere*, Harvard University

[73] **P.Facal et al.**, 2000, Auger Technical Note, GAP-2000-010

[74] **E.Fermi**, 1949, Phys. Rev., 73(8), 1169

[75] **B.Fick et al.**, *The first Central Laser Facility*, Auger Technical Note, GAP-2004-003, 2004

[76] **B.Fick et al.**, *The Central Laser Facility at the Pierre Auger Observatory*, JINST, 1:P1 1003, 2004

[77] **Y.Fomin et al.**, 1996, J. Phys. G, 22, 1839

[78] **P.Fowler et al.**, 1967, Proc. Roy. Soc., 301, 39

[79] **P.Fowler et al.**, 1987, Ap. J., 314, 739

[80] **H.Freudenreich et al.**, 1990, Phys. Rev. D, 41(9), 2732

[81] **G.Frichter et al.**, Phys. Rev. D, 50, 3135

[82] **T.Gaisser et al.**, 1977, Proc. 15th ICRC (Plovdiv), 8, 353

[83] **T.Gaisser**, 1990, *Cosmic Rays and Particle Physics*, Cambridge University Press

[84] **D.Garcia-Pinto et al.**, 2006, *Prospects for separating UHECR species using asymmetries in time distributions*, 20th European Cosmic Ray Symposium

[85] **R.Geiger**, 1965, *The climate near the ground*, Harvard University Press

[86] **R.Giacconi et al.**, 1912, Physik. Zeitschr., 13, 1804

[87] **L.Gleeson et al.**, 1968, Astrophysics and Space Science, vol. 2, 4, 431-437

[88] **K.Greisen**, 1956, *Progress in Cosmic Rays*, 3, 1

[89] **K.Greisen**, 1960, *Cosmic ray showers*, Ann. Rev. Nucl. Sci., 10, p63-108

[90] **K.Greisen**, 1966, Physical Review Letters, 16(17), 748-750

[91] **C.Giulio**, 2009, PhD thesis, Universita Degli Studi di Roma

[92] **C.Hanson et al.**, 1992, Proc. SPIE, 1735, 17-26

[93] **T.Hara et al.**, 1983, Proc. 18th Int. Cosmic Ray Conf. (Bangalore), 11, 281.

[94] **D.Harari et al.**, 2006, JCAP, 11, 12

[95] **D.Hartmann et al.**, 1992, J. Climate, 5, 1281

[96] **E.Haug et al.**, 2004, The elementary process of Bremsstrahlung, published by World Scientific

[97] **N.Hayashida et al.**, 1999, Proc. 26th ICRC, Salt Lake City, 3, 256

[98] **V.Hess**, 1912, Phys. Z., 13, 1084

[99] **A.Hillas**, 1970, Acta Physica Academiae Scientiarum Hungaricae Suppl., 3, 29, 355-360

[100] **A.Hillas et al.**, 1971, Proc. 12th Int. Cosmic Ray Conf. (Hobart), 3, 1001

[101] **A.Hillas**, 1984, Ann. Rev. Astron. Ap., 22, 45

[102] **M.Honda et al.**, 1993, Phys. Rev. Lett., 70, 525

[103] **M.Horvat et al.**, Auger Technical Note, GAP-2007-057

[104] **J.Houghton et al.**, 1984, *Remote Sounding of Atmospheres*, Cambridge Planetary Science

[105] **T.Johnson**, 1933, Phys. Rev., 43, 834

[106] **J.Jokipii et al.**, 1985, Ap. J. Lett., 290, L1

[107] **J.Jokipii**, 1987, Ap. J., 312, 170

[108] **J.Jokipii et al.**, 1987, Ap. J., 312, 170

[109] **K.Kamata et al.**, 1956, Prog. Theor. Physc., Suppl. 6, 93

[110] , **M.Kaplon et al.**, 1949, Phys. Rev., 76, 1735

[111] **B.Keilhauer et al.**, 2003, Auger Technical Note, GAP-2003-009

[112] **T.Kifune et al.**, 1986, J. Phys. G. 12, 143

[113] **B.Kimball et al.**, 1982, Water Resource Research, 18, 931

[114] **J.Knapp et al.**, 2003, Astroparticle Physics, 19, 77-99

[115] **T.Kohonen**, 1990, Proceedings of the IEEE, Vol 78, 9, 1464-1480

[116] **W.Kolhörster**, 1914, Deutsch. Phys. Gesell. Verh., 16:719

[117] **W.Kolhörster et al.**, 1938, Naturwuss., 26, 576

[118] **H.Kon** *A Determination of Entrainment Rate and Drag Coefficient of Cumulus Clouds*, Jour. Fac. Sci., Hokkaido Univ., S7(Geophysics), Vol.7, No. 1, p41-46, 1981

[119] **K.Kondratyev**, 1969, *Radiation in the Atmosphere*, Academic Press, NY

[120] **G.Krymskii**, 1977, Dokl. Akad. Nauk SSSR, 234, 1306

[121] **P.Lagage et al.**, 1983, Astron. Ap., 125, 249

[122] **L.Landau et al.**, 1953, Akad. Nauk. SSSR, 92, 535 & 735

[123] **M.Lawrence et al**, 1991, J. Phys., G17, 733

[124] **J.Lindsley et al.**, 1962, J. Phys. Soc. Japan, 17 (supp A-III), 91

[125] **J.Lindsley**, 1963, Phys. Rev. Lett., 10, 146

[126] **J.Linsley**, 1975, Phys. Rev. Lett., 34, 1530

[127] **J.Linsley**, 1983, Proc. 18th ICRC Bangalore, 12, 135

[128] **J.Lloyd-Evans et al.**, 1983, Nature, 305, 784

[129] **J.MacQueen**, 1967, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 281-297

[130] **A.Maghrabi**, 2007, PhD thesis, Adelaide University

[131] **J.Matthews**, 2005, Astropart. Phys., 22, 387

[132] **T.McComb et al.**, 1979, J. Phys. G, 5, 1613

[133] **H.Mielke et al.**, J. Phys. Rev. D, 24, 2536

[134] **A.Migdal**, 1956, Phys. Rev., 103, 1811

[135] **R.Mussa et al.**, 2007, *Detecting Clouds with Elastic Lidar Telescopes*, Auger Technical Note, GAP-2007-038

[136] **K.Mora et al.**, 2008, J. Phys. : Conf. Ser. 116 012007

[137] **D.Nagle et al.**, 1988, Ann. Rev. Part. Sci., 38, 609

[138] **D.Newton et al.**, 2007, Astroparticle Physics, 26, 6, 414-419

[139] **N.Otsu** *A threshold selection method from gray-level histograms*, IEEE Trans. Sys. Man. Cyber., 9, 62-66

[140] **R.Owen** *Producibility advances in hybrd uncooled infrared devices*, Proc. SPIE, 2225, p79-86, 1994

[141] **G.Paltridge**, 1970, Q. J. R. Met. Soc., 96, 645

[142] **Particle data group**, 2002, *Review of Particle Physics*, Phys. Rev., D66, 010001

[143] **B.Peters**, 1961, Nuovo Cimento C(1), 22(4), 800

[144] **C.Powell et al.**, 1959, The Study of Elementary Particles by the Photographic Method, Pergammon Press

[145] **R.Protheroe**, 1984, Nature, 310, 296

[146] **R.Protheroe et al.**, 1992, Phys. Rev. Lett., 69, 2885

[147] **J.Ramsey et al.**, 1982, J. Appl. Met., 21, 566

[148] **R.Rand et al.**, 1989, App. J., 343, 760

[149] **H.Reis et al.**, 1999, Auger Technical Note, GAP-1999-013

[150] **D.Riordan et al.**, 2005, J. Geophys. Res., 110, D03207

[151] **M.Roberts et al.**, 2003, Auger Technical Note, GAP-2003-010

[152] **G.Robinson**, 1947, Q. J. R. Met. Soc., 73, 127

[153] **G.Robinson**, 1950, Ibid., 79, 37

[154] **B.Rossi**, 1933, Z. Phys., 82, 151

[155] **B.Rossi et al.**, 1941, Rev. Modern Phys., 13, 240

[156] **B.Rossi**, 1952, *High Energy Particles*, Prentice Hall

[157] **M.Roth et al.**, “SD Reconstruction”, Offline Reference Manual

[158] **L.Rothman et al.**, 1992, J. Quant. Spectrosc. Radiat. Transfer, 48, (5/6) 469

[159] **M.Samorski et al.**, 1983, Proc. 18th Int. Cosmic Ray Conf. (Bangalore), 11, 244

[160] **R.Sato et al.**, 1999, Auger Technical Note, GAP-1999-014

[161] **H.Scharr**, 2000, *Optimal Operators in Digital Image Processing*, PhD thesis, University of Heidelberg

[162] **F.Schüssler et al.**, 2008, Auger Technical Note, GAP-2008-013

[163] **W.Sellers**, 1965, *Physical Climatology*, The University of Chicago Press, Chicago

[164] **J.Simpson**, 1983, Ann. Rev. Part. Sci., 33, 323

[165] **J.Simpson**, 1983, *Composition and Origin of Cosmic Rays*, Reidel Publishing Company

[166] **K.Simpson**, 2001, *Studies of cosmic ray composition using a hybrid fluorescence detector*, Thesis, Adelaide University

[167] **D.Skobelzyn**, 1929, Z. Phys., 54, 686

[168] **I.Sobel et al.**, 1968, *A 3x3 isotropic gradient operator for image processing*, presented as a talk at the Stanford Artificial Project

[169] **P.Sommers**, 2004, C.R. Acad. Sci. Paris, t. 4, IV

[170] **C.Song et al.**, 2000, Astropart. Phys., 14:7

[171] **V.Souza et al.**, 2006, Astroparticle Physics, 25, 1, 84-92

[172] **W.Swinbank**, 1963, Q. J. R. Met. Soc., 89, 339

[173] **A.Szabo**, 1992, PhD thesis, Adelaide University

[174] **B.Valeur**, 2001, Molecular Fluorescence: Principles and Applications, Wiley-VCH

[175] **M.Véron-Cetty et al.**, 2006, *A catalogue of quasars and active nuclei: 12th edition*, Astronomy and Astrophysics, Volume 455, Issue 2, 773

[176] **H.Völk et al.**, 1988, Ap. J. Lett., 333, L65

[177] **B.Wiebel-Sooth et al.**, 1998, Landolt-Boernstein Numerical Data and Functional Relationships, New Series Vol. VI 3C, Springer

[178] **M.Winn et al.**, 1986, J. Phys., G12, 653

[179] **B.Xavier et al.**, 2003, Proc. 28th Int. Cosmic Ray Conf. (Japan), 813

[180] **G.Yodh et al.**, 1983, Phys. Rev. D, 27, 1183

[181] **G.Yodh et al.**, 1984, Phys. Rev. D, 29(5), 892

[182] **G.Zatsepin et al.**, 1966, Journal of Experimental and Theoretical Physics Letters, 4, 78-80