ASTRONOMICAL PIPELINE PROCESSING USING FUZZY LOGIC

By
LIOR SHAMIR

A DISSERTATION
Submitted in partial fulfillment of the requirements
for the degree of
DOCTOR OF PHILOSOPHY
(Engineering – Computational Science & Engineering)

MICHIGAN TECHNOLOGICAL UNIVERSITY
2006

This dissertation, "Astronomical Pipeline Processing Using Fuzzy Logic," is hereby approved in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY in the field of Computational Science & Engineering.

DEPARTMENT or PROGRAM:

Computational Science & Engineering

Signatures:

Dissertation Advisor _____

Typewritten Name _____

Department Chair _____

Typewritten Name _____

Date _____

Abstract

In the past few years, pipelines providing astronomical data have been becoming increasingly important. The wide use of robotic telescopes has provided significant discoveries, and sky survey projects such as SDSS and the future LSST are now considered among the premier projects in the field astronomy. The huge amount of data produced by these pipelines raises the need for automatic processing. Astronomical pipelines introduce several well-defined problems such as astronomical image compression, cosmic-ray hit rejection, transient detection, meteor triangulation and association of point sources with their corresponding known stellar objects.

We developed and applied soft computing algorithms that provide new or improved solutions to these growing problems in the field of pipeline processing of astronomical data. One new approach that we use is fuzzy logic-based algorithms, which enables the automatic analysis of the astronomical pipelines and allows mining the data for not-yet-known astronomical discoveries such as optical transients and variable stars.

The developed algorithms have been tested with excellent results on the *NightSkyLive* sky survey, which provides a pipeline of 150 astronomical pictures per hour, and covers almost the entire global night sky.

Contents

Chapter 1

Introduction

In the past few years, pipelines providing astronomical data have been becoming increasingly important. The wide use of robotic telescopes has provided significant discoveries, and sky survey projects are now considered among the premier projects in the field astronomy. In this paper we will concentrate on the ground based missions, although future space based missions like Kepler (Borucki et al. 2003), SNAP (Smith et al. 2003), and JWST (Clampin et al. 2003) will also create significant pipelines of astronomical data.

Pan-STARRS (Kaiser 2004), a 60 million dollar venture, is being built today and is expected to be completed by 2006. Pan-STARRS will be composed of 4 telescopes pointing simultaneously at the same region of the sky. Each telescope will be equipped with a 1.8 meter primary mirror and a CCD focal plane mosaic with one billion pixels. With coverage of 6000 degrees^2 per night, Pan-STARRS is expected to observe the entire available sky several times each month, looking for transients that include supernovas, planetary eclipses, and asteroids that might pose a future threat to Earth. Similarly but on a larger scale, ground-based LSST (Tyson 2002) is planned to use a powerful 8.4 meter robotic telescope that will cover the entire sky every 10 days. LSST will cost $200M, be completed by 2012, and produce 13 terabytes per night. Current sky survey projects include SDSS (Loveday 2003; York 2000), which uses a 2.5 meter robotic telescope trying to map the entire visible universe. In addition, many smaller scale robotic telescopes are being deployed and their number is growing rapidly.

However, in the modern age of increasing bandwidth, human identifications are many times impracticably slow. Therefore, one can reasonably assume that many discoveries of significant scientific value are yet hidden inside the huge databases provided by the deployment of robotic telescopes.

Useful automatic pipeline processing of astronomical images depends on accurate algorithmic decision making. For previously identified objects, one of the first steps in computer-based analysis of astronomical pictures is an association of each object with a known catalog entry. This necessary step enables such science as automatically detected transients and automated photometry of stars. Since computing the topocentric coordinates of a given known star at a given time is a simple and common task, transforming the celestial topocentric coordinates to image (x, y) coordinates might provide the expected location of any star in the frame. However, in many cases slight shifts in the orientation, inaccuracy of the optics or imperfections in the CCD can make this seemingly simple task formidable.

Another essential step in astronomical pipelines is the removal of cosmic ray hits. Except from their annoying presence in astronomical images, cosmic ray hits might be mistakenly detected as true astronomical sources. Algorithms that analyze astronomical frames must ignore the peaks caused by cosmic ray hits, yet without rejecting the peaks of the true astronomical sources. This problem becomes even more significant in space-based telescopes located far from a planetary magnetic field (Offenberg et al. 1999).

A common problem in the field of astronomical pipelines is image compression. Astronomical images consume a relatively large amount of space, and broad astronomical pipelines introduces the problem of storage (Axelrod et al. 2004). Space-based missions, such as the million-miles-away JWST, face the problem of bandwidth availability (Nieto-Santisteban et al. 1999). Therefore, the efficiency of astronomical pipelines is, in some cases, dependent on image compression algorithms.

In this thesis we present several novel algorithms that provide solutions to some of the basic problems of astronomical pipelines. The algorithms have been implemented and regularly used by the *Night Sky Live* sky survey (Nemiroff & Rafert 1999), which incorporates all-sky cameras in some of the worlds premier observatories. In Chapter 2 a lossy image compression algorithm that protects a user defined levels of photometric integrity is described. This algorithm can be used for compressing astronomical images using a user-defined compression factor so that the size of the files can fit the available bandwidth or storage devices,

yet without practically loosing photometric data. In Chapter 3 a fuzzy logic-based star detection algorithm for wide-angle astronomical images is presented. This algorithm allows associating objects detected in an astronomical image with known catalogued objects. By applying this algorithm photometric data of known stars can be archived and analyzed. Chapter 4 discusses a fuzzy logic-based cosmic-ray hit rejection algorithm. Cosmic-ray hits have a disturbing effect on astronomical images, and the rejection of the traces they leave on the images is a well-known problem. Chapter 5 presents the generation of all-sky relative opacity maps, which allows detecting even minor changes in the atmospheric opacity by using a pipeline of all-sky astronomical images. In Chapter 6 a method of analysis of meteor trails using stereo fish-eye cameras is described. The described technique allows deducing the 3D trajectory by using two fish-eye CCD cameras located in two different places. Chapter 7 describes a mechanism that automatically detects transients in panoramic all-sky pipelines. Chapter 8 provides a comprehensive description of the software system that analyzes continuous astronomical data provided by all-sky fish-eye cameras.

*A part of this chapter was published in: Shamir, L., Nemiroff, R. J.; Using Fuzzy Logic for Automatic Analysis of Astronomical Pipelines, *Lecture Notes in Artificial Intelligence*, Vol. 3614, pp. 634–637, 2005.

Bibliography

Axelrod, T., Connolly, A., Ivezic, Z., Kantor, J., Lupton, R., Plante, R., Stubbs, C., Wittman D., 2004, AAS Meeting 205, 108.11

W. J. Borucki, D. Koch, G. Basri, T. Brown, D. Caldwell, E. Devore, E. Dunham, T. Gautier, J. Geary, R. Gilliland, A. Gould, S. Howell, J. Jenkins, 2003, Proc. of the Conf. on Towards Other Earths, 69

Clampin, M., Atkinson, C., Feinberg, L., Hayden, W., Lightsey, P., Mountain, M., Texter, S., 2003, AAS Meeting 203, 121.01

Kaiser N., 2004, Proc. SPIE 54, 11

Loveday, J., 2003, JBAA, 113, 364

Nemiroff, R. J., Rafert, J. B., 1999, PASP, 111, 886

Nieto-Santisteban, M. A., Fixsen, D. J., Offenberg, J. D., Hanisch, R. J., Stockman, H. S., 1999, ASP Conf., 172, 137

Offenberg, J. D., Sengupta, R., Fixsen, D. J., Stockman, H. S., Nieto-Santisteban, M. A., Stallcup, S., Hanisch, R. J., Mather, J. C., 1999, ASP Conf. 172, 141

Smith, J. A., Allam, S. S., Bohlin, R. C., Duestua, S. E., Ebbets D., Kent S., Lampton, M., Laubscher, B. E., Mostek, N., 2003, AAS Meeting 203 82.22

Tyson, J. A., 2002, Proc. SPIE, 4836, 10

York, D. G., 2000, AJ, 120, 1579

Chapter 2

PHOTZIP: A Lossy FITS Image Compression Algorithm that Protects User-Defined Levels of Photometric Integrity

Abstract: A lossy compression algorithm is presented for astronomical images that protects photometric integrity for detected point sources at a user-defined level of statistical tolerance. PHOTZIP works by modeling, smoothing, and then compressing the astronomical background behind self-detected point sources, while completely preserving values in and around those sources. The algorithm also guaranties a maximum absolute difference (in terms of σ) between each compressed and original background pixel, allowing users to control quality and lossiness. For present purposes, PHOTZIP has been tailored to FITS format and is freely available over the web. PHOTOZIP has been tested over a broad range of astronomical imagery and is in routine use by the Night Sky Live (NSL) project for compression of all-sky FITS images. Compression factors depend on source densities, but for the canonical NSL implementation, a PHOTZIP (and subsequently GZIP or BZIP2) compressed file is typically 20% of its uncompressed size.

*This chapter was published in: Shamir, L., Nemiroff, R. J., 2005, *The Astronomical Journal*, Vol. 129(1), pp. 539–546.

2.1 Introduction

Astronomy research images are now almost exclusively digital. There is an increasing need to store these images and to send them over the Internet. Average bandwidth and storage capacity, although increasing, continue to be a bottleneck that frequently limits scientific exploitation of these images. Lossless compression of astronomical image files therefore creates a clear advantage over non-compression since it increases effective storage and bandwidth and so bolsters scientific utility. Lossy compression is more controversial, however, as the scientific value of the data lost in the compressed images must be weighed against the scientific value of the data gained by the extra bandwidth and storage space.

One of the premier scientific uses for astronomical images is photometry. Whether detecting the presence of, for example, distant supernovae, Local Group microlensing, binary star variability, or planetary transits, the need for photometric accuracy in astronomy images remains a primary objective for many astronomical research projects.

Frequently, the photometric value of an astronomical image is concentrated in the point sources in the image. Conversely, the bulk of the image size is concentrated in the background behind these sources. When the number of pixels taken up by sources is small compared to the number of pixels that compose the background, it becomes possible to significantly compress the image size while preserving a certain level of photometric integrity.

While lossless compression algorithms preserve 100% of the signal, lossy algorithms can provide a better compression factor while losing some of the signal (Press 1992; Fixsen et al. 2000; Watson 2002). However, lossy compression algorithms tend to convolve science with art. Astronomy-specific lossy data compression algorithms are not new, and many have been proposed and widely used (Pence 1994; Pence et al. 2000, 2002; White 1992; Press 1992). HCOMPRESS (White 1992) is a commonly used FITS compression program based on a two-dimentional Haar wavelet transform. This compression program is fast and provides a relatively high compression factor. However,

while providing the user control of the lossiness\compression tradeoff, HCOMPRESS provides a limited control over the type of the signal that is lost in the compression \ decompression process.

The most similar compression approach to that discussed here is the insightful FITS compression program FITSPRESS (Press 1992). FITSPRESS is a wavelet based compression algorithm that has, among other things, sensitivity to preserving the brightest image pixels.

Lossy compression (e.g. JPG, HCOMPRESS) tends to be controlled by parameters that have no scientific or statistical meaning. In contrast, the PHOTZIP FITS compression algorithm provides the user control of the preserved\lost signal in scientific terms. Like (Veran & Wright 1994), PHOTZIP should be used as a preprocessor to multi-purpose lossless compression algorithms (e.g. LZW) and allows those algorithms a higher compression factor by losing some of the signal, but ensures that only *background* information of the image will lose some of its signal. The algorithm provides an interface which can be used in order to define, in terms of σ, the criteria for a pixel to be considered as background, and guarantees a user-defined maximum absolute difference (also in terms of σ) for any background pixel that loses signal. In Section 2 we describe the algorithm, in Section 3 we present ways to improve the compression factor, in Section 4 we discuss photometric integrity and in Section 5 we discuss the performance of the algorithm.

2.2 Lossy Compression That Preserves Bright Signals

Since a main purpose of our algorithm is to allow lossiness only for background pixels, the first stage of the algorithm is to determine for every pixel in the frame whether or not it is a background pixel. For PHOTZIP, we achieve this by using square window median filtering. The background value of each pixel is determined to be the median of the values of all pixels within its window. Assuming that the bias is zero, the gain is 1 electron and the read noise is negligible, we can compute σ for each pixel by $\sigma = \sqrt{B_{x,y}}$, where $B_{x,y}$ is the estimated background of the pixel at coordinates (x, y).
The background computation stage can be summarized by the following algorithm:

1. for y $\leftarrow$ 1 to height do
2. for x $\leftarrow$ 1 to width do
3. $B_{x,y} \leftarrow$ median of $C_{x-s,y-s}, C_{x-s+1,y-s},$
$\ldots, C_{x-s,y-s+1}, \ldots, C_{x+s,y+s}$
4. $\sigma_{x,y} = \sqrt{B_{x,y}}$
5. end for
6. end for

$C_{x,y}$ is the value of the pixel at the coordinates (x, y), and $B_{x,y}$ is the estimated background value of the pixel at the coordinates (x, y). s is the half width size of the window. The nested loops in lines 1 and 2 make sure that the background computation is done for every pixel. Therefore, every pixel in the frame is attached to the background value that is the median of the pixels in the $(2s+1) \times (2s+1)$ window centered on (x, y). After the estimated background is computed for a pixel, the σ of that pixel is also computed by $\sigma = \sqrt{B_{x,y}}$. In our current implementation, PHOTZIP assumes a default half-width size of 10, although the user can specify any half-width size.

When an image has a great many pixels, a pixel-by-pixel median computation might turn into a computationally expensive task. For an $N \times N$ frame we will need to compute the median value of N^2 square windows, where each window has $(2s+1) \times (2s+1)$ pixels. For instance, for a 1024 $\times$ 1024 frame, when computing the background values of the pixels using a windows with half width size of 10, The algorithm will need to compute 1,048,576 (1024 $\times$ 1024) times a median of 441 (21 $\times$ 21) numbers. In order to reduce needed processing power, we chose to compute the background value not for *every* pixel, but for windows of 5 $\times$ 5 in which the median value is computed only for the "leader" pixel, which is the pixel at the center of the window. All other pixels in that 5 $\times$ 5 window are associated with the same background value as their "leader". Since backgrounds do not tend to drastically change over small windows, we consider this technique as an acceptable approximation. In order to compute the median efficiently, we use the common algorithm for finding a median in linear time described by (Corman et al. 1990).

Once the background value (and hence σ) is determined for every pixel in the frame, the pixel values are quantized. The quantization stage is affected by two parameters. The first, d, is the minimum brightness, in terms of σ, of a pixel such that every pixel that is less bright is classified as background. The second parameter, b, is the maximum absolute difference, also in terms of σ, that is allowed between a background pixel in the original image and the same pixel in the compressed\decompressed image. The basic idea of the quantization is that a value of a certain pixel (x, y) is quantized only if it is lower than $B_{x,y} + d \cdot \sigma_{x,y}$, so the lossiness of the algorithm does not affect sufficiently bright pixels. For every pixel (x, y) in the frame, we first check if it is brighter than $B_{x,y} + d \cdot \sigma_{x,y}$. If the pixel meets this criterion then it is not quantized and does not loose any of its signal.

The criteria for a pixel to be quantized is

$$C_{x,y} < B_{x,y} + d \cdot \sigma_{x,y}. \tag{2.1}$$

The top-level algorithm for selecting the pixels that should be quantized is:

1. for y ← 1 to height do
2. for x ← 1 to width do
3. if $C_{x,y} - B_{x,y} < d \cdot \sigma_{x,y}$ then
4. $C_{x,y} \leftarrow quantize(C_{x,y}, \sigma_{x,y}, b)$
5. end for
6. end for

In line 4, the subroutine *quantize* is called in order to perform the quantization of any pixel that does not meet the criteria of line 3. The size of the quanta used is $2 \cdot 2^{\lfloor \log_2 b \cdot \sigma \rfloor}$, where b is a user-defined positive value ($b > 0$) such that the maximum absolute difference between the value of the pixel in the original frame, and the value of the same pixel in the compressed\decompressed frame cannot be greater than $2^{\lfloor \log_2 b \cdot \sigma \rfloor}$.
The quantization algorithm is simply:

quantize (c,σ,b)
1. $quantum\_size \leftarrow 2 \cdot 2^{\lfloor \log_2 b \cdot \sigma \rfloor}$
2. $quantized\_value \leftarrow quantum\_size \cdot Round(\frac{c}{quantum\_size})$
3. return($quantized\_value$)

c is the pixel's value and b is the maximum absolute difference (in terms of σ) that is allowed between a pixel in the original frame and the same pixel in the compressed\decompressed frame. *Round* is a function that rounds its argument to the nearest integer. This quantization symmetrically increases or decreases pixel values. The interval of each quantum is $[quantized\_value - 2^{\lfloor \log_2 b \cdot \sigma \rfloor}, quantized\_value + 2^{\lfloor \log_2 b \cdot \sigma \rfloor}]$. Since c is a value within this interval, the absolute difference $|c - quantized\_value|$ can never be greater than $2^{\lfloor \log_2 b \cdot \sigma \rfloor}$. Since $2^{\lfloor \log_2 b \cdot \sigma \rfloor} \leq b\sigma$, the absolute difference between a pixel value in the compressed\decompressed frame and the same pixel in the original frame can not be greater than $b\sigma$. It might seem that simple quanta at the size of $2b\sigma$ can improve the compression factor by providing larger quantum sizes, yet still comply with the maximum absolute difference criteria. However, since σ is different for every pixel, this might lead to a large variance in the quantum sizes and therefore severely reduce the compression factor. For instance, suppose that we have two pixels with values of 97 and 99, and σ of 9 and 10 respectively. Assuming $b = 1$, with quantum size of $2b\sigma$ the first value, 97, will be quantized using quantum sizes of 18 and the second value, 99, will be quantized using quantum sizes of 20. After the quantization process, the values will be, therefore, 90 and 100 respectively. However, if using quantum size of $2^{\lfloor \log_2 b \cdot \sigma \rfloor + 1}$, the quantization of both values is done using the same quantum size (16 in this case). After the quantization process, the value of both pixels will be 96, which increases the compression potential of pattern matching based compression algorithms. The low variance of quantum sizes leads to smaller variance of quantized values, which is an important factor in the performance of many multi-purpose compression algorithms. Examples of the differences in the compression factor (using *BZIP2* with PHOTZIP) when using quantum size of $2b\sigma$ and quantum size of $2 \cdot 2^{\lfloor \log_2 b \cdot \sigma \rfloor}$ are listed in the following table:

d	b	s	**quantum size** $2 \cdot 2^{\lfloor \log_2 b \cdot \sigma \rfloor}$	quantum size $2b\sigma$
1	1	8	73.6%	64.2%
2	1	8	76.5%	66.1%
1	2	10	77.3%	67.0%

The file that was used for the samples is "ci040325ut005115p.fits" which is an unsigned integer FITS image of size of 2102 KB. This file is discussed more thoroughly in Section 5.

The function *quantize* might fail when σ is equal to zero. However, when $\sigma = 0$, the condition stated in line 3 of the top level algorithm cannot be satisfied and the function *quantize* is not invoked.

Since the quantization is symmetric, the mean of the pixel values should be preserved in the compressed \ decompressed frame. Assuming a normal distribution for the pixel values in the original frame, the median of the original frame should be equal to the mean. As the mean is preserved, the median of the original frame can be taken from the mean of the compressed \ decompressed frame.

Since the number of integers within each $[quantized\_value - 2^{\lfloor \log_2 b \cdot \sigma \rfloor}, quantized\_value + 2^{\lfloor \log_2 b \cdot \sigma \rfloor}]$ quantum is odd, there is always one integer value that can be quantized either up or down. A systematic policy that always quantizes these values in the same fashion (either up or down) will cause a systematic bias to the mean. Thus, in order to avoid statistical biasing, the standard *Round* function always rounds half integers to the nearest even integer (Wolfram 1999). Due to that behavior of *Round*, if the quantized value is exactly in the center of the interval, it can either gain or lose value so that systematic bias is avoided.

2.3 Compressing Non-Astronomical Edges and Artifacts

In practice, image pixels might exist that are clearly non-astronomical and do not need to be preserved. One example is the edge of a frame that is not exposed to the sky but dominated by other sources of noise. In some cases, pixels such as these would not have a high pixel value, but since they are in a relatively dim area of the frame, their value is high compared to their background. Pixel values of non-astronomical edges can sometimes have a high variance, so some of the pixels can be significantly brighter than other pixels around them. Since σ is determined by the local background of each pixel, a low background value of a pixel leads to a low σ, and a low σ leads to a low $d\sigma$. Given their low $d\sigma$, the algorithm would normally preserve the signal of pixels which do not have a high value by meeting the $C_{x,y} > B_{x,y} + \sigma_{x,y}d$ criteria due to the extremely low value of their background. This might result in an unnecessarily low compression factor. In order to allow a user to avoid this trap, we set another optional parameter t that sets a threshold value for pixels that are allowed to lose signal. The threshold value is set by t in terms of the median value of the frame. For instance, if $t = 1.5$ then any pixel with value less than $1\frac{1}{2}$ times the value of the median pixel will not be required to preserve its value even if it meets the $C_{x,y} > B_{x,y} + \sigma_{x,y}d$ criteria. Therefore, to include this, line 3 of the top level algorithm presented in section 2 should be changed to:

3. if $C_{x,y} - B_{x,y} < d \cdot \sigma_{x,y}$ or $C_{x,y} < t \cdot$ (median of $C_{1,1}, C_{2,1}, \ldots, C_{1,2}, \ldots, C_{width,height}$) then

We found this parameter is very effective in compressing Night Sky Live (Nemiroff et al. 2005) FITS frames, in which a significant portion of the frame is not directly exposed to the sky, and therefore consists of scattered low values that have no scientific utility. If those areas were uniform, then no special action would need to be taken and the pixel values would be naturally quantized.

2.4 Photometric Integrity

Many astronomical images are sparsely populated. When the number of pixels taken up by point sources is small compared to the number of pixels that compose the background, it becomes possible to significantly compress the image while preserving a useful level of photometric integrity. In this Section we estimate this level of photometric integrity.

When symmetrically quantizing the background pixels, the mean of the pixel values is preserved in a level that depends on the number of background pixels averaged. A high number of averaged pixels will lead to a low standard error.

When a pixel value is quantized, the absolute difference between the pixel value in the original frame and the pixel value in the compressed\decompressed frame is bound by $2^{\lfloor \log_2 b \cdot \sigma \rfloor}$. Let Δ be the difference between the value of a certain pixel (x, y) in the original frame and the value of the same pixel in the compressed\decompressed frame such that:

$\Delta = C_{x,y} - \text{quantize}(C_{x,y}, \sigma, b)$

Let $\bar{C}$ be the mean of n pixel values in the compressed \ decompressed frame such that:

$\bar{C} = \frac{\sum_{i=1}^{n} (C_i - \Delta_i)}{n} = \frac{\sum_{i=1}^{n} C_i}{n} - \frac{\sum_{i=1}^{n} \Delta_i}{n}$

So $|\frac{\sum_{i=1}^{n} \Delta_i}{n}|$ is the absolute difference between the mean of the original pixel values and the mean of the quantized pixel values.

Let $q = 2^{\lfloor \log_2 b \cdot \sigma \rfloor}$. Since the function *quantize* described in section 2 guaranties that the absolute difference $|\Delta|$ is bound by q, Δ can be any value within the interval $[-q, q]$. Since Δ is uniformly distributed in the interval $[-q, q]$, the expected value of Δ is zero, and the standard deviation of Δ is $\frac{q}{\sqrt{3}}$.

$stddev(\Delta) = \frac{q}{\sqrt{3}}$ (*stddev(X)* is defined as the standard deviation of a random variable X).

The variance of Δ is:

$var(\Delta) = stddev^2(\Delta) = \frac{q^2}{3}$

Let $\bar{\Delta}$ be the absolute differences between the mean of the original values and mean of the quantized values of some n pixels.

$\bar{\Delta} = \frac{\sum_{i=1}^{n} \Delta_i}{n} = \sum_{i=1}^{n} \frac{\Delta_i}{n}$

$var(\bar{\Delta}) = var(\sum_{i=1}^{n} \frac{\Delta_i}{n}) = \sum_{i=1}^{n} var(\frac{\Delta_i}{n}) = \sum_{i=1}^{n} \frac{var(\Delta_i)}{n^2} = \frac{n \cdot var(\Delta)}{n^2} = \frac{var(\Delta)}{n}$

$stddev(\bar{\Delta}) = \sqrt{var(\bar{\Delta})} = \sqrt{\frac{var(\Delta)}{n}} = \sqrt{\frac{q^2}{3n}} = \frac{q}{\sqrt{3n}} = \frac{2^{\lfloor \log_2 b \cdot \sigma \rfloor}}{\sqrt{3n}}$

We can see that the most likely value of the difference between the mean of the original values and the mean of the quantized values is zero. We can also see that the standard deviation of that difference decreases in an asymptotically order, and approaches zero when $n \to \infty$. For instance, if we choose to use $b = 1$, the standard deviation of the mean of 1600 pixels is equal to $\frac{2^{\lfloor \log_2 1 \cdot \sigma \rfloor}}{\sqrt{3 \cdot 1600}}$, which is smaller than $\frac{1}{69}\sigma$. In typical integer FITS frames, the standard deviation in this case will usually be smaller than 1, which indicates that the mean of a group of many pixels in the original frame will practically be equal to the mean of those pixels in the compressed\decompressed frame. Since analysis of background pixels usually involves very many pixels, the mean will be only negligibly affected by the

presented quantization.

This analysis is based on two assumptions mentioned above: 1. The mean of the difference between the original values and the quantized values is zero, and 2. The distribution of the differences is approximately uniform. (Watson 2002) suggests that this is not necessarily true for quanta sizes greater than 2σ. Therefore, the algorithm can guarantee the preservation of the mean only for values of b such that $b \leq 1$.

The same approach also applies to the sum:
Let S_0 be sum of n pixels values in the original frame and S_q be the sum of the n quantized pixel values in the compressed \ decompressed frame.
$S_0 = \sum_{i=1}^{n} C_i$
$S_q = \sum_{i=1}^{n} (C_i - \Delta_i) = \sum_{i=1}^{n} C_i - \sum_{i=1}^{n} \Delta_i$
$\frac{S_q}{S_0} = \frac{\sum_{i=1}^{n} C_i - \sum_{i=1}^{n} \Delta_i}{\sum_{i=1}^{n} C_i} = 1 - \frac{\sum_{i=1}^{n} \Delta_i}{\sum_{i=1}^{n} C_i}$
Since the most likely value of Δ_i is 0, the most likely value of $\frac{\sum_{i=1}^{n} \Delta_i}{\sum_{i=1}^{n} C_i}$ is also 0. The standard deviation of the expression $\frac{\sum_{i=1}^{n} \Delta_i}{\sum_{i=1}^{n} C_i}$ is:
$stddev(\frac{\sum_{i=1}^{n} \Delta_i}{\sum_{i=1}^{n} C_i}) = \frac{n \cdot stddev(\bar{\Delta})}{n \cdot \bar{C}} = \frac{2^{\lfloor \log_2 b \cdot \sigma \rfloor}}{\sqrt{3n} \cdot \bar{C}}$
Where $\bar{C} = \frac{\sum_{i=1}^{n} C_i}{n}$
For instance, when computing the sum of 1600 quantized pixels when $\bar{C} = 1000$ and $b = 1$, will provide $\frac{S_q}{S_0} = 1 - \frac{2^{\lfloor \log_2 1 \cdot \sigma \rfloor}}{\sqrt{3 \cdot 1600} \cdot 1000} > 1 - \frac{\sigma}{69000}$

Why not quantize all of the pixels, including the bright pixels of obvious point sources? The practical risk here is that point sources so quantized might involve a small number of pixels and lead to a large error. For instance, if we have only 5 bright object pixels which can be relied on for photometry of a certain astronomical object, the standard deviation of the mean of the pixel values is $\frac{2^{\lfloor \log_2 b \cdot \sigma \rfloor}}{\sqrt{3 \cdot 5}}$, which might be $\sim 0.26 b\sigma$. Since the pixels of the astronomical objects are usually the most interesting to science, the presented algorithm allows a user to completely preserve their values through the quantization process. However, since signal loss due to noise might be greater than signal loss due to the quantization process, a user might choose to set d to ∞ in order to force the algorithm to quantize *all* the pixels in the frame. This will increase the compression factor (examples are given in Section 5), but will also result in additional signal loss. Even though the additional signal loss caused by the quantization can be smaller than the signal loss already caused by the background noise, since those pixels are the most valuable for science we chose to allow absolute preservation of their values. This also allows a user to use extreme values of b, while still preserving the point spread functions of the sources. Quantizing the brightest pixels can not only change the raw sum of these pixels, but also the point spread function. Note that the point spread function shapes can be useful for everything from photometry to discerning cosmic-rays.

Also, in some case, the photometric brightness of a source can be estimated from the brightest source pixels alone, after background subtraction. Such brightness measures are particularly useful when the background changes significantly and unpredictably over the wings of the PSF. One project that uses such photometric measures is the Night Sky Live project, which struggles against a ill-behaved sloping background and so records quantities like C1, C5, C9, etc., meaning the level of the brightest pixel, the average of the five brightest pixels, etc. Preserving the brightest pixel values then specifically enables such photometry schemes.

Lastly, in cases of sub-pixel point spread functions, when only the single brightest pixel is measured, not preserving the brightest pixel could lead to a loss of any signal of on the order of $b\sigma$.

Concentrating again on the background, using a large number of pixels reduces the affect of the quantization on the mean, so that the mean of the compressed \ decompressed frame should be practically equal to the mean of the original frame. Assuming a normal distribution of the pixel values, the *median* of the original frame should be equal to the *mean* of the original frame. Since the mean of the original frame is nearly preserved, the median of the original frame can be equated to the mean of the compressed\decompressed frame. However, since the value of the median pixel is changed, computing the median directly from the compressed\decompressed frame produces a different value than the median of the original frame. Therefore, the median preservation is strongly subject to the assumption of normal distribution of the pixel values. Practically, the number of pixel values involved in the median computation is finite, and can actually be smaller than the quantum size. As the number of pixels involved is smaller, the perturbation to the median is higher. In the worst case scenario, the perturbation can be $b\sigma$. In some cases, this perturbation may not easily avoided, but since b is user defined, a user can know and control the maximum perturbation allowed for the median. Due to that perturbation, users of the compressed\decompressed data might need to adjust their photometry algorithms accordingly. In order to avoid computing the background using the median or mode, users might choose, for instance, to use the mean with outlier rejection.

A higher d quantizes more and brighter pixels. Therefore a lower d makes fainter objects peak above the quantization and hence easier to detect in the compressed \ decompressed frame. However, a relatively high b might

also contribute to the lossiness of faint objects. For instance, when $b = 1$, faint objects with pixel values such that $C_{x,y} - B_{x,y} < \sigma$ might be completely gone from the compressed\decompressed frame. Therefore, one using the algorithm should consider her expectations of photometry integrity and set the b and d parameters accordingly. We believe that an advantage of our method is that it provides a guarantee on the lost signal in standard statistical terms.

The local background estimation is particularly efficient when the ratio of the number of point source pixels to background pixels is small. Additionally, the median filter window size (s) used for background computation should be significantly larger than the point spread function of the astronomical sources. Yet, the median (or mean) is sometimes not an optimal background estimation when the objects are large (Patat 2003). In astronomy, fortunately for our PHOTZIP algorithm, variability is prevalent only among small objects always below the angular size of the point-spread function.

A non-uniform unknown background is best determined locally. In pictures such as those taken by the Night Sky Live project, the background is highly non-uniform and usually unpredictable, making a local background estimation highly important. The local estimation also simplifies the usage of the algorithm, since one does not have to be aware of the type of the background of the frame when applying the compression. Nevertheless, in cases a uniform background is desired, the t parameter describe in Section 3 can be set for that purpose.

2.5 Performance of the PHOTZIP Algorithm

PHOTZIP should be used along with a multi-purpose lossless compression utility such that the lossless utility is applied after running PHOTZIP. We tested PHOTZIP with two common compression utilities: *gzip*, which is based on Ziv & Lempel's compression algorithm (Ziv & Lempel 1977), and *bzip2*, which is based on Burrows & Wheeler's algorithm (Burrows & Wheeler 1994). These algorithms, like others, are based on finding and compressing repetitive patterns in the data. Additionally, quantization of the signal increases redundancy and allows multi-purpose compression algorithm a yet higher compression factor (Yang & Kieffer 1998).

The compression factor and the amount of lost signal determine the utility of the algorithm. In order to test the algorithm, we used all-sky 1024×1024 FITS images used by Night Sky Live all-sky monitoring network that deploys CONtinuous CAMeras (CONCAMs). We also tested FITS images taken by other optical instruments. In all cases, the algorithm provided a significantly better compression factor then all popular multi-purpose compression utilities alone. Table 2.1 below presents the affect of the d, b, s, t parameters on the performance of the algorithm in terms of compression factor, and compares it to compression factors of common lossless compression programs. The four rightmost columns are the compression factors when using PHOTZIP before applying gzip and bzip2, and when using gzip or bzip2 without using PHOTZIP. The compression factor used here is the amount of data that was compressed as a percentage of the size of the original frame.

Figure 2.1 depicts the image file ci040325ut005115p.fits, a typical FITS image from the NSL (Nemiroff et al. 2005) project. The picture is available in FITS and JPG formats at http://www.NightSkyLive.net. The file n3166_lj.fits and n3184_lj.fits were taken from the galaxy catalog at http://www.astro.princeton.edu/~frei/catalog.htm. The later is shown as Figures 2.2 and 2.3.

As the maximum allowed absolute difference (b) increases, the compression factor is higher, but so is the signal loss. Since the tested frames contained mostly background, like many astronomical images, the value of b has a substantial affect on the compression factor. To test the utility of the b parameter, we set d to ∞, effectively letting the algorithm quantize *all* the pixels and therefore achieving a higher compression factor. Although a better compression factor can be achieved, not using the d parameter can increase the bright signal loss as discussed in Section 2.4.

A higher t parameter increases the compression factor, but this parameter should be used with extra caution since it can also potentially cause signal loss of bright pixels. The s parameter normally does not have a significant affect on the compression factor, but has some affect on the time required to compress an image.

2.6 Conclusions

When even a single pixel is quantized, photometry can be at least partially compromised. A background estimated from the mean of surrounding pixels, however, will usually end up closer to the original background. We have shown that using even as little as 1600 pixels for background estimation can reduce the background error in the mean of the quantized pixels to below a single count. Practically, this means that the background estimation and hence photometric measurements can be protected in the PHOTZIP compression\decompression process.

In sum, we present here a simple lossy compression algorithm for astronomical images. The main advantage of this algorithm is that it can preserve the signal of bright pixels, while symmetrically losing signal only from the background. The criteria for preserving a pixel's value is user-defined (in terms of σ), so the user can accurately control the compression factor/signal loss trade-off. In addition, the algorithm guarantees a user-defined maximum absolute difference (also in terms of σ), so the user can control the amount of lost signal for those areas in the frame that are not preserved. The algorithm was implemented and tested on unsigned 16-bit integer FITS images, but we

File Name	File Size	d	b	s	t	photzip + gzip	gzip	photzip + bzip2	bzip2
n3166_lr.fits	204 KB	1	1	20	0	76.6%	50.3%	81.9%	60.1%
n3166_lr.fits	204 KB	2	1	20	0	79.7%	50.3%	86.4%	60.1%
n3166_lr.fits [1]	204 KB	1	2	20	0	80.3%	50.3%	84.4%	60.1%
n3166_lr.fits [1]	204 KB	3	6	10	0	98.6%	50.3%	98.6%	60.1%
n3166_lr.fits [1]	204 KB	3	3	20	0	91.7%	50.3%	92.6%	60.1%
n3166_lr.fits	204 KB	3	1	20	0	82.8%	50.3%	86.4%	60.1%
n3166_lr.fits	204 KB	∞	1	20	0	84.0%	50.3%	87.9%	60.1%
n3166_lr.fits [1]	204 KB	∞	2	20	0	91.8%	50.3%	92.7%	60.1%
n3184_lj.fits	204 KB	1	1	20	0	75.0%	50.5%	81.3%	55.5%
n3184_lj.fits [1]	204 KB	3	1	20	0	82.8%	50.5%	87.1%	55.5%
ci040325ut005115p.fits	2102 KB	1	1	8	0	66.7%	38.1%	73.6%	54.1%
ci040325ut005115p.fits	2102 KB	∞	1	8	0	76.2%	38.1%	84.0%	54.1%
ci040325ut005115p.fits [1]	2102 KB	1	2	10	0	74.0%	38.1%	77.3%	54.1%
ci040325ut005115p.fits [1]	2102 KB	2	3	10	0	82.6%	38.1%	86.9%	54.1%
ci040325ut005115p.fits [1]	2102 KB	2	3	10	0.5	88.7%	38.1%	91.1%	54.1%
ci040325ut005115p.fits	2102 KB	3	1	10	0.5	78.5%	38.1%	84.8%	54.1%
ci040325ut005115p.fits	2102 KB	3	1	10	0	73.4%	38.1%	81.8%	54.1%
ci040325ut005115p.fits [1]	2102 KB	1	2	10	0.5	81.5%	38.1%	85.2%	54.1%
ci040325ut005115p.fits [1]	2102 KB	1	3	10	0.5	87.4%	38.1%	89.2%	54.1%
ci040325ut005115p.fits [1]	2102 KB	1	3	10	0	79.1%	38.1%	86.9%	54.1%
ci040325ut005115p.fits [1]	2102 KB	1	6	10	1.15	92.1%	38.1%	93.2%	54.1%
ci040325ut005115p.fits [1]	2102 KB	3	6	10	1.15	94.9%	38.1%	95.1%	54.1%

Table 2.1: Compression Factor

believe that the same approach can be applied also to signed integer and floating point images. The algorithm was tested on some typical astronomical 16-bit integer FITS images and appeared to be effective. PHOTZIP is now in routine use on FITS data taken by the Night Sky Live project.

Bibliography

Burrows M. and Wheeler D. 1994, Digital Systems Research Center, Research Report 124

Corman., T. H., Leiserson, C. E. and Rivest, R. L. 1990, "Introduction to Algorithms", MIT Press

Fixsen, D. J., Hanisch, R. J., Mather, J. C., Nieto-Santisteban, M. A., Offenberg, J. D., Sengupta, R., Stockman, H. S., 2000, Astronomical Data Analysis Software and Systems IX, ASP Conference Proceedings, 216, eds. N. Manset, C. Veillet, and D. Crabtree. Astron. Soc. Pac., 539

Nemiroff, R. J., Shamir, L., Pereira, W. E., Rafert, J. B., Ftaclas, C. and Cook, D. 2005, in preparation

Patat. F, 2003, Astronomy & Astrophysics, 401, 797

Pence, W , 1994, A.S.P. Conference Series, 61, D.R. Crabtree, R.J. Hanisch, and J. Barnes, eds., 523

Pence, W. D., 2002, Eds: S. Jean-Luc, D. Murtagh, Fionn, Proc. SPIE, 4847

Pence, W., White, R. L., Greenfield, P., Tody, D, 2000, ASP Conference Proceedings, 216, eds: N. Manset, C. Veillet, and D. Crabtree, Astron Soc. Pac., 551

Press, W. H. 1992, Astronomical Data Analysis Software and Systems I, A.S.P. Conf. Ser., 25, eds. D.M. Worrall, C. Biemesderfer & J. Barnes

Stephen Wolfram , 1999, "The Mathematica Book", section 3.2, Cambridge University Press & Wolfram Media

Veillet, C., 1998, CFHT Information Bulletin, 39

Veran, J. P., Wright, J. R., 1994, ASP Conference Series, 61, Astronomical Data

Watson A. M., 2002, Revista Mexicana de Astronomia y Astrofisica, 38, 233

Wells, D. C., Greisen, E. W., and Harten, R. H. 1981, Astron. Astrophys. Suppl., 44, 363

White, R., Postman, M. and Lattanzi, M. in H. T. MacGillivray and E.B. Thompson (Eds.), 1992, Digitized Optical Sky Surveys, Kluwer, Dordrecht, 167

Yang E., Kieffer J. C. 1998, IEEE Trans. Inf. Theory, 44, 47

Ziv, J. and Lempel, A. 1977, IEEE Trans. Information Theory, 23, 337

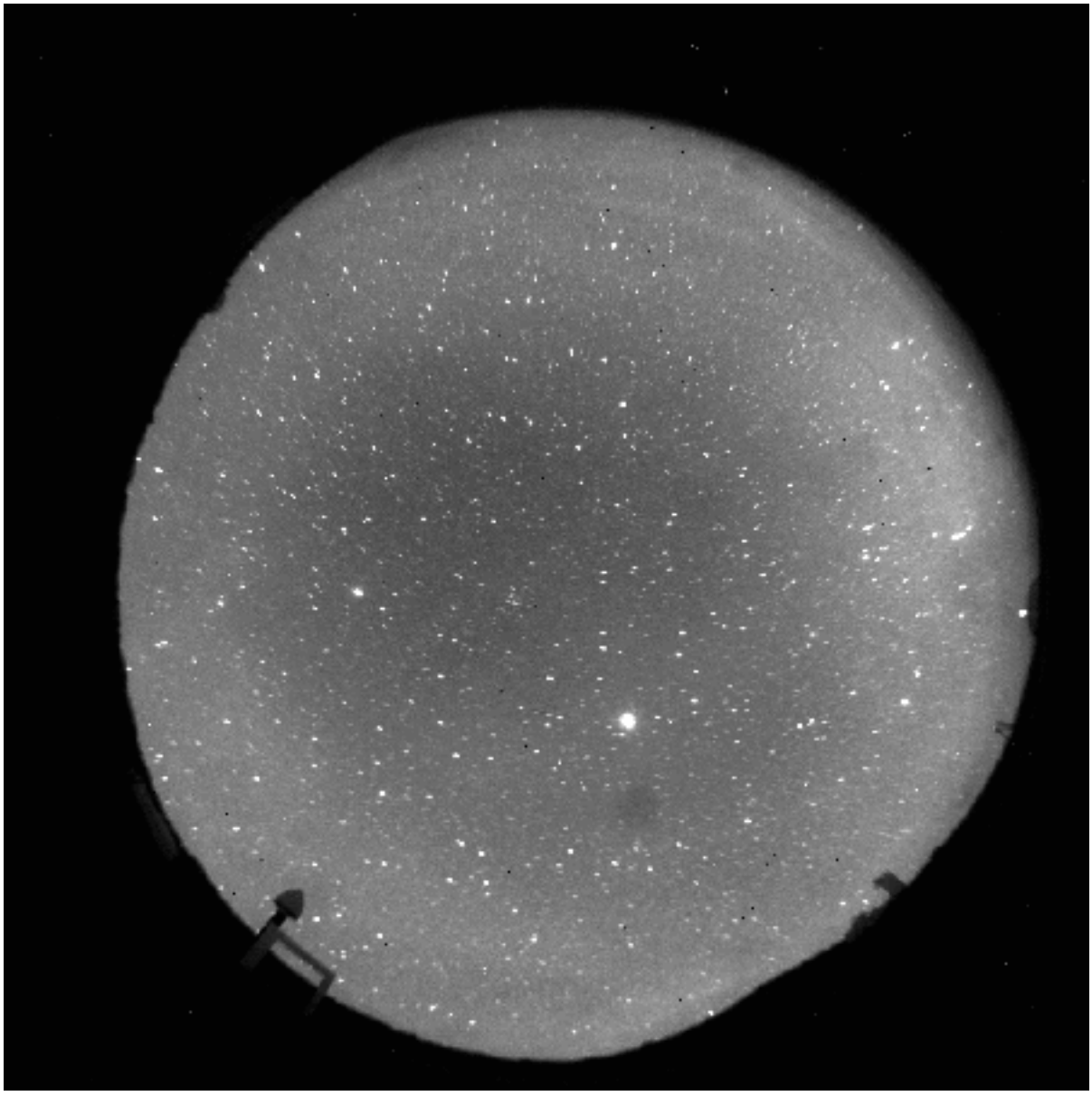

Figure 2.1: ci040325ut005115p.fits: A Night Sky Live all-sky picture.http://www.NightSkyLive.net

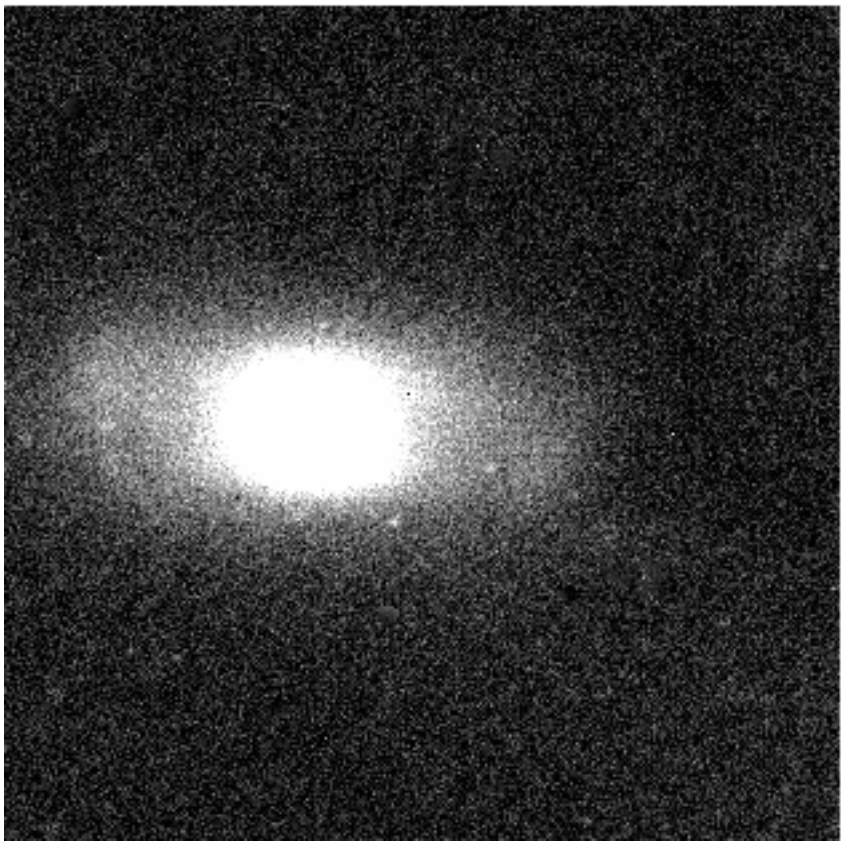

Figure 2.2: n3166_lr.fits: A picture taken from the *galaxy catalog.* http://www.astro.princeton.edu/~frei/catalog.htm.

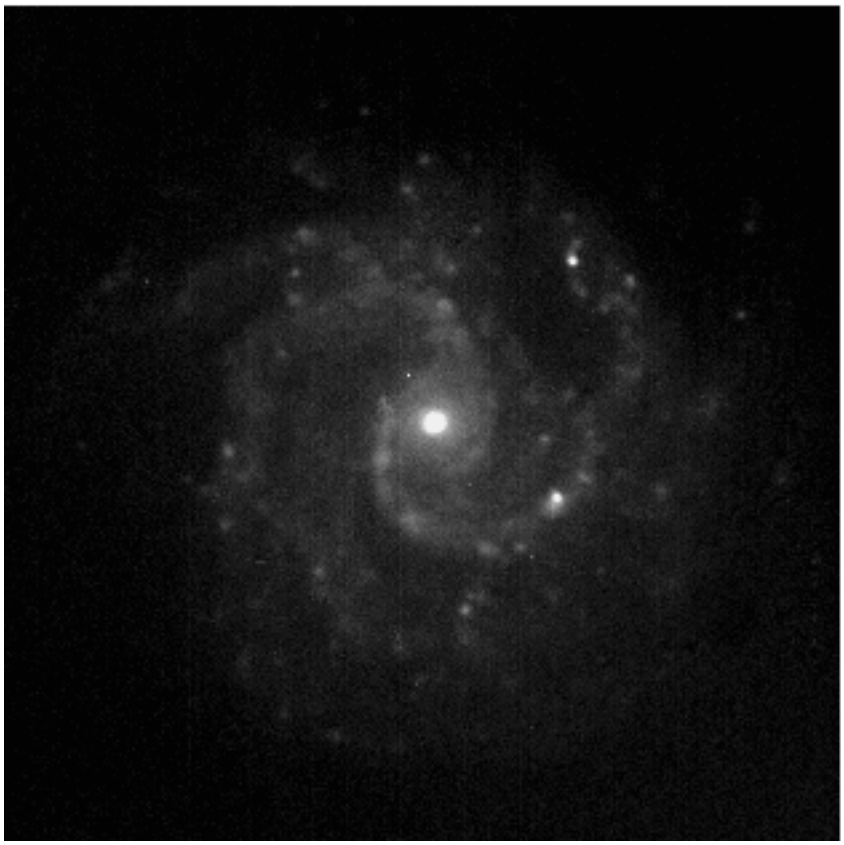

Figure 2.3: n3184_lj.fits: A picture taken from the *galaxy catalog*.
http://www.astro.princeton.edu/~frei/catalog.htm.

Chapter 3

A Fuzzy Logic Based Algorithm For Finding Astronomical Objects in Wide-Angle Frames

Abstract: Accurate automatic identification of astronomical objects in an imperfect world of non-linear wide-angle optics, imperfect optics, inaccurately pointed telescopes, and defect-ridden cameras is not always a trivial first step. In the past few years, this problem has been exacerbated by the rise of digital imaging, providing vast digital streams of astronomical images and data. In the modern age of increasing bandwidth, human identifications are many times impracticably slow. In order to perform an automatic computer-based analysis of astronomical frames, a quick and accurate identification of astronomical objects is required. Such identification must follow a rigorous transformation from topocentric celestial coordinates into image coordinates on a CCD frame. This paper presents a fuzzy logic based algorithm that estimates needed coordinate transformations in a practical setting. Using a training set of reference stars, the algorithm statically builds a fuzzy logic model. At runtime, the algorithm uses this model to associate stellar objects visible in the frames to known-catalogued objects, and generates files that contain photometry information of objects visible in the frame. Use of this algorithm facilitates real-time monitoring of stars and bright transients, allowing identifications and alerts to be issued more reliably. The algorithm is being implemented by the Night Sky Live all-sky monitoring global network and has shown itself significantly more reliable than the previously used non-fuzzy logic algorithm.

*This chapter was published in: Shamir, L., Nemiroff, R. J., 2005, *Publication of the Astronomical Society of Australia*, Vol. 22(2), pp. 111–117.

3.1 Introduction

Useful pipeline processing of astronomical images depends on accurate algorithmic decision making. For previously identified objects, one of the first steps in computer-based analysis of astronomical pictures is an association of each object with a known catalog entry. This necessary step enables such science as automatically detected transients and automated photometry of stars. Since computing the topocentric coordinates of a given known star at a given time is a simple and common task, transforming the celestial topocentric coordinates to image (x, y) coordinates might provide the expected location of any star in the frame. However, in many cases slight shifts in the orientation, inaccuracy of the optics or imperfections in the CCD can make this seemingly simple task formidable.

Fuzzy logic (Zadeg 1965; Zadeh 1978, 1983, 1994) is an extension of Boolean logic that is useful for making complex decisions by computers. While in Boolean logic an item has only two levels of memberships to a set (false or true; 0 or 1), fuzzy logic supports any value within the interval [0,1] as a level of membership of an item to a fuzzy set. A fuzzy logic model consists of three different types of entities: fuzzy sets, fuzzy variables and fuzzy rules. The membership of a fuzzy variable in a fuzzy set is determined by a function that produces values within the interval [0,1]. These functions are called *membership functions*. Fuzzy variables are divided into two groups: *antecedent variables* that contain the input data of the fuzzy logic model, and *consequent variables* that contain the results calculated by the fuzzy logic model.

In this paper, we present a fuzzy logic based algorithm that transforms celestial coordinates into image coordinates for even complex combinations of wide-angle non-linear optical distortions, slight optical imperfections, and small unrecorded orientational perturbations. In Section 2 we describe the fuzzy logic algorithm, and in Section 3 we present

and discuss a practical implementation of the algorithm for automatic analysis of an operational astronomical project called "Night Sky Live" (Nemiroff et al. 2005).

3.2 Fuzzy Logic Based Coordinate Transformations

The transformation of celestial topocentric spherical sky coordinates to local Cartesian image coordinates can be defined by a set of two functions. Mathematical conversion from right ascension, declination, latitude, longitude, and local time to altitude and azimuth is straightforward. Further transformation of azimuth and altitude sky coordinates to (x, y) image (CCD) coordinates is frequently harder in practice when faced with practical inaccuracies of knowledge. For practical application of fuzzy logic, this later transformation is broken up into two parts:

$$(altitude, azimuth) \longmapsto x \tag{3.1}$$

$$(altitude, azimuth) \longmapsto y \tag{3.2}$$

On a CCD image, pixel locations can be specified in either Cartesian or polar coordinates. Let x_{zen} be the x coordinate (in pixels) of the zenith in the image, and y_{zen} be the y coordinate of the zenith. x and y coordinates of any given star visible in an astronomical image can be computed as follows:

$$x = x_{zen} + \sin(angle) \cdot distance \tag{3.3}$$

$$y = y_{zen} + \cos(angle) \cdot distance \tag{3.4}$$

Where *angle* is a polar azimuthal angle and *distance* is a polar radial distance.

In order to use these equations it is necessary to compute a polar distance and angle for given objects. Given the observer's latitude and longitude, this can be done by converting the celestial coordinates (azimuth and altitude) of a given stellar object at a given time to the corresponding angle and distance in the image. Since the azimuth and altitude of any given bright star or planet at any given time can be easily computed, the only missing link here is the transformation of the altitude and azimuth to image coordinates, so the object can be found in the image.

3.2.1 Reference Stars

Each of the two models is built based on manually identified reference stars. A reference star can be any star within an image that was correctly associated with the corresponding stellar object. Being familiar with the night sky, we can inspect the frame by eye and identify the stellar objects that appear in it. The image (x, y) coordinates of the object can be taken from the peak of its point spread function (PSF). Each reference star provides a record with the following fields: azimuth, altitude, angle, and distance.

Each identified star contributes an azimuth and altitude (by basic astronomy) and also an angle and distance (by measurement from the image). These provide the raw data for constructing a mapping between the two, using the fuzzy logic model that will be described later in the paper. In order to obtain an accurate fuzzy logic model that calculates the angle, it is necessary to select reference stars that uniformly cover the entire image. This assures that the calculation of one value will depend on reference points that are relatively close to it. This is also true for the model that calculates the distance.

3.2.2 Building the Fuzzy Logic Model

In order to transform celestial coordinates into image coordinates, two different fuzzy logic models are being built based upon the two transformations:

$$f_1 : azimuth \longmapsto angle \tag{3.5}$$

$$f_2 : altitude, azimuth \longmapsto distance \tag{3.6}$$

Here, *altitude*, *azimuth* and *angle* are angular measures, while *distance* is measured in pixels. Each transformation (f_1 and f_2) is computed by a different fuzzy logic model, thus one model calculates the angle and the other calculates the distance. The fuzzy logic model f_1, which calculates the angle, is an approximation based on the assumption that the optical axis is perfectly aligned, so the angle is not depended on the altitude.

The fuzzy logic model f_1 has one antecedent (input) fuzzy variable and one consequent (output) fuzzy variable, while the fuzzy logic model f_2 has two antecedent variables (*altitude*,*azimuth*) and one consequent fuzzy variable.

3.2.3 Converting Azimuth to Polar Angle on the CCD (f_1)

The first model (f_1) is built according to the reference stars such that each reference star adds to the model one fuzzy set and one fuzzy rule. Each fuzzy set is associated with a membership functions that is built in the form of a triangle (Zadeg 1965). Each of these membership functions reaches its maximum at the reference value, and intersects with the x-axis at the reference values of its neighboring reference stars. For instance, suppose we would like to build the

fuzzy logic model with a data set that contains the following four reference stars:

azimuth	**altitude**	**angle**	**distance**
0	ϵ_0	θ_0	R_0
α_1	ϵ_1	θ_1	R_1
α_2	ϵ_2	θ_2	R_2
α_3	ϵ_3	θ_3	R_3

The first reference star maps azimuth 0^o. Assuming $\alpha_1 < \alpha_2 < \alpha_3$, the membership functions that will be added to the model are described in Figure 3.1.

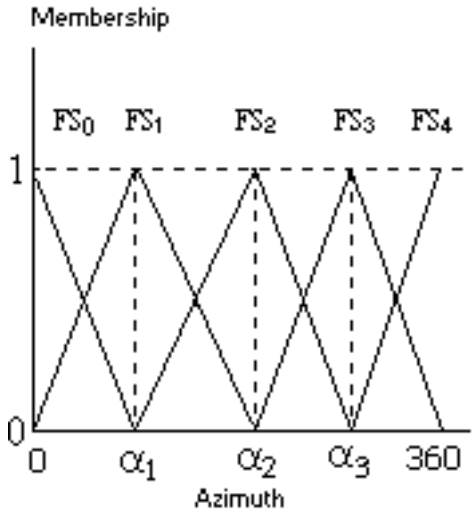

Figure 3.1: The membership functions of the fuzzy sets (FS_0 to FS_4) created by the four reference values $(0,\theta_0),(\alpha_1,\theta_1),(\alpha_2,\theta_2),(\alpha_3,\theta_3)$ for $f_1(azimuth \longmapsto angle)$.

The membership of the fuzzy sets is determined by the following membership functions:

$$F_0(x) = \begin{cases} 1 - \frac{1}{\alpha_1}\cdot \mathrm{x} & 0 \le x \le \alpha_1 \\ 0 & x < 0 \text{ or } x > \alpha_1 \end{cases}$$

$$F_1(x) = \begin{cases} \frac{1}{\alpha_1}\cdot \mathrm{x} & 0 \le x < \alpha_1 \\ 1 - \frac{1}{\alpha_2 - \alpha_1} \cdot (x - \alpha_1) & \alpha_1 \le x \le \alpha_2 \\ 0 & x < 0 \text{ or } x > \alpha_2 \end{cases}$$

$$F_2(x) = \begin{cases} \frac{1}{\alpha_2 - \alpha_1} \cdot (x - \alpha_1) & \alpha_1 \le x < \alpha_2 \\ 1 - \frac{1}{\alpha_3 - \alpha_2} \cdot (x - \alpha_2) & \alpha_2 \le x \le \alpha_3 \\ 0 & x < \alpha_1 \text{ or } x > \alpha_3 \end{cases}$$

$$F_3(x) = \begin{cases} \frac{1}{\alpha_3 - \alpha_2} \cdot (x - \alpha_2) & \alpha_2 \le x < \alpha_3 \\ 1 - \frac{1}{360 - \alpha_3} \cdot (x - \alpha_3) & \alpha_3 \le x \le 360 \\ 0 & x < \alpha_2 \text{ or } x > 360 \end{cases}$$

$$F_4(x) = \begin{cases} \frac{1}{360 - \alpha_3} \cdot (x - \alpha_3) & \alpha_3 \le x \le 360 \\ 0 & x < \alpha_3 \text{ or } x > 360 \end{cases}$$

F_0 to F_4 are the membership functions of the fuzzy sets FS_0 to FS_4 that were created according to the reference stars (the function F_m is the membership function of the fuzzy set FS_m). The membership functions are built such that almost all azimuth values belong (with non-zero membership) to two fuzzy sets. Only the points of maximum (0,α_1,α_2,α_3,360) have a non-zero membership to just one set.

The defuzzification method used in this model is *weighted average*, which is an efficient defuzzification method when the fuzzy logic model is built according to a set of signleton values ([Takagi & Sugeno 1983; Takagi & Sugeno

1985). Since *weighted average* is used, the consequent part of each rule is a crisp value and not a fuzzy set. Therefore, the fuzzy rules that will be added to the model are:
$FS_0 \longmapsto \theta_0$
$FS_1 \longmapsto \theta_1$
$FS_2 \longmapsto \theta_2$
$FS_3 \longmapsto \theta_3$
$FS_4 \longmapsto 360$

For instance, suppose that the first three reference stars have azimuths of 0^o, 10^o, and 20^o, and their polar angles are 3^o, 12^o, and 22^o respectively, such that $\alpha_0 = 0^o$, $\theta_0 = 3^o$, $\alpha_1 = 10^o$, $\theta_1 = 12^o$, $\alpha_2 = 20^o$, and $\theta_2 = 22^o$. Each reference star adds one fuzzy set to the model such that the membership functions of the first two fuzzy sets FS_0, FS_1 are:

$$F_0(x) = \begin{cases} 1 - \frac{1}{10} \cdot x & 0 \le x \le 10 \\ 0 & x < 0 \text{ or } x > 10 \end{cases}$$

$$F_1(x) = \begin{cases} \frac{1}{10} \cdot x & 0 \le x < 10 \\ 1 - \frac{1}{10} \cdot (x - 10) & 10 \le x \le 20 \\ 0 & x < 0 \text{ or } x > 20 \end{cases}$$

Each reference star also adds one fuzzy rule such that the first three fuzzy rules are:
$FS_0 \longmapsto 3$
$FS_1 \longmapsto 12$
$FS_2 \longmapsto 22$

Now suppose that we want to use this model in order to compute the polar angle (in the image) of a stellar object with an azimuth of 6^o. The value 6 has a level of membership of $1 - \frac{1}{10} \cdot 6 = 0.4$ to the fuzzy set FS_0 and $\frac{1}{10} \cdot 6 = 0.6$ to the fuzzy set FS_1. Since the level of membership of 6 to all other fuzzy sets is 0, the only rules that will have any affect on the output of the computation are rules 0 and 1 above. Since the *weighted average* defuzzification method is performed, the output of the computation is $\frac{0.4 \cdot 3 + 0.6 \cdot 12}{0.4 + 0.6} = 8.4$. This computation practically provides the same results as a simple linear interpolation of the value 6 with the two closest neighboring reference stars.

3.2.4 Converting Altitude and Azimuth to Radial Distance on the CCD (f_2)

Unlike the simpler f_1 model used for transforming the azimuth to angle, the computation of the distance (in pixels) from (x_{zen}, y_{zen}) should be computed based on *two* parameters, which are the altitude and the azimuth. Using both the altitude and azimuth allows the model to deal with asymmetric behavior of the optics as well as inaccurate orientational information. In other words, using the assumption that the optics orientation is directly at the zenith and the distortion of the optics and hardware is completely symmetric, a reference point at a certain azimuth would allow calculating the distance of a point at the same altitude but at a different azimuth. However, this is not always the case. For instance, a stellar object with the azimuth of 0^o (north) and altitude of 30^o can be at distance of 150 pixels from (x_{zen}, y_{zen}), while another stellar object at the same altitude (30^o) but at azimuth of 60^o will be at distance of 155 pixels from (x_{zen}, y_{zen}). Moreover, (x_{zen}, y_{zen}) does not necessarily appear in the center of the frame, and the frame is not necessarily centralized. Therefore, when computing the distance of a stellar object from (x_{zen}, y_{zen}), it is required to take not only the altitude of the stellar object into considerations, but also the azimuth. In order to do that, the fuzzy logic model that calculates the distance is built according to reference stars that not only have different altitudes, but also different azimuths.

We build the f_2 model that computes the distance using four different sets of reference stars such that each set contains reference stars that share approximately the same azimuth. For the sake of simplicity, the first set contains reference stars that are near azimuth 0^o, the second set contains reference stars near azimuth 90^o, and the other two sets contain stars near azimuth of 180^o and 270^o respectively. I.e., all reference stars used for this model should be fairly close to the azimuth of 0^o, 90^o, 180^o or 270^o. In order to use the four sets of reference stars, four new fuzzy sets are added to the model. Those fuzzy sets are "North","East", "South", and "West".

Fuzzy logic can be viewed as a complex interpolation method. In two dimensions, our fuzzy logic realization allows f_2 to perform the combined interpolation of linear and Gaussian functions in a straightforward fashion, and to alter the model easily when needed. Note that this is quite different than four separate linear and/or Gaussian interpolations for the four directions. The membership functions of the fuzzy sets are described in Figure 3.2.

The membership in the fuzzy sets is determined by the following Gaussian functions:

North:

$$F_{\text{n}}(x) = \begin{cases} e^{-\frac{x^2}{2\sigma^2}} & 0 \le x \le 180 \\ e^{-\frac{(x-360)^2}{2\sigma^2}} & 180 < x \le 360 \end{cases}$$

East:

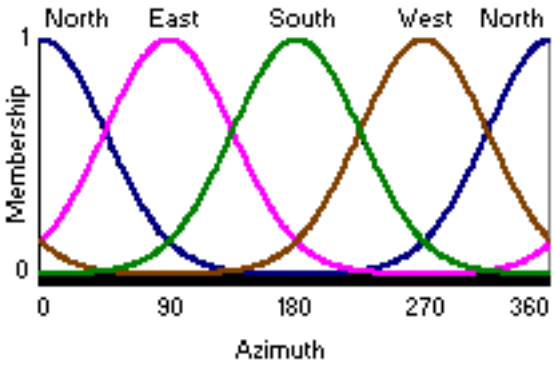

Figure 3.2: The fuzzy sets of the four directions (North, South, East, West).

$$F_{\mathrm{e}}(x) = \begin{cases} e^{-\frac{(x-90)^2}{2\sigma^2}} & 0 \le x \le 270 \\ e^{-\frac{(x-450)^2}{2\sigma^2}} & 270 < x \le 360 \end{cases}$$

South:

$$F_{\mathrm{s}}(x) = e^{-\frac{(x-180)^2}{2\sigma^2}}$$

West:

$$F_{\mathrm{w}}(x) = \begin{cases} e^{-\frac{(x-270)^2}{2\sigma^2}} & 90 < x \le 360 \\ e^{-\frac{(x+90)^2}{2\sigma^2}} & 0 \le x \le 90 \end{cases}$$

Where σ is set to 45.

In this case, the level of membership for each of the four membership functions is always greater than zero, as opposed to the f_1 angle model where only two membership functions have a positive level of membership.

Building f_2 model can be demonstrated by an example: Suppose the model is built based on the following 6 reference stars listed in Table 3.1.

star	azimuth	altitude	angle	distance
S_1	0^o	68^o	2.4^o	215 pixels
S_2	359^o	62^o	1.2^o	224 pixels
S_3	1.5^o	72^o	3.4^o	206 pixels
S_4	90^o	66^o	94.2	180 pixels
S_5	91^o	62^o	95.6	188 pixels
S_6	91.6^o	70^o	95.9	172 pixels

Table 3.1: Reference stars

As with f_1, each reference star adds to the model one fuzzy set that has a triangle membership function. For instance, the membership function of the fuzzy set added by S_1 reaches its maximum of unity at 68, and intersects with the x-axis at the points of maximum of its neighboring reference stars. The neighboring stars are the two stars that their altitudes are closest to the altitude of S_1 (such that one is greater than 68^o and one is smaller than 68^o), *and* have approximately the same azimuth as S_1. In this example, one neighboring star would be S_2 and the other would be S_3. All three stars share approximately the same azimuth (which is North). Therefore, the fuzzy set *Alt68N* that will be added by S_1 will have a triangle membership function that reaches its maximum at 68, and intersects with the x-axis at 62 and 72. This membership function is described in Figure 3.3.

The membership function added by S_1 is:

$$F_{Alt68N}(x) = \begin{cases} \frac{x-62}{68-62} & 62 < x \le 68 \\ 1 - \frac{x-68}{72-68} & 68 < x \le 72 \\ 0 & x < 62 \text{ or } x > 72 \end{cases}$$

Since the azimuth of S_1 is approximately north, the fuzzy rule added to the model by S_1 is:

Alt68N $\wedge$ North $\longmapsto$ 215

This rule is more significant for stars that appear in the northern part of the sky (and in this case, also at an altitude of around 68^o). The practical affect of this rule will be stronger as the coordinates are closer to the 68^o parallel. The fuzzy rules that will be added by the other reference stars are:

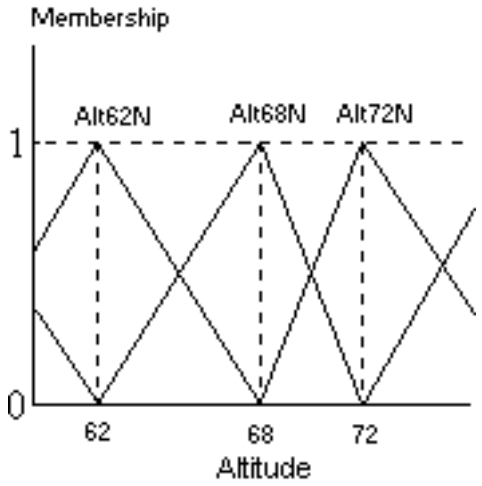

Figure 3.3: The membership function of the fuzzy set *Alt68N*. The figure also include parts of the membership functions of *Alt62N* and *Alt72N* that were added by the two neighboring reference stars S_2 and S_3.

Alt62N $\wedge$ North $\longmapsto$ 224
Alt72N $\wedge$ North $\longmapsto$ 206
Alt62E $\wedge$ North $\longmapsto$ 188
Alt66E $\wedge$ North $\longmapsto$ 180
Alt70E $\wedge$ North $\longmapsto$ 172

Now suppose that we want to use this model in order to compute the distance (in pixels) of a stellar object with an azimuth of 30^o and altitude of 64^o. The value 64 have a membership value of $\frac{64-62}{68-62} = \frac{1}{3}$ to the fuzzy set *Alt68N* (that was added to the model by S_1), and $1 - \frac{64-62}{68-62} = \frac{2}{3}$ to the fuzzy set *Alt62N* (that was added to the model by S_2). Also, this value would have a membership value of $\frac{64-62}{66-62} = 0.5$ to the fuzzy set added by S_4, and a membership value of $1 - \frac{64-62}{66-62} = 0.5$ to the fuzzy set added by S_5. The membership value of the azimuth 30^o to the fuzzy set *North* is $e^{-\frac{30^2}{2 \cdot 45^2}} \simeq 0.8$, and to the fuzzy set *East* it is $e^{-\frac{(30-90)^2}{2 \cdot 45^2}} \simeq 0.412$. The membership values to the fuzzy sets *South* and *West* are very close to 0 in this case, and therefore have very little effect on the output value. For the sake of the simplicity of the example, these membership values are assumed to be exactly 0.

The computation process is based on *product* inferencing and *weighted average* defuzzification. Therefore, the output value of the computation would be:
$\frac{215 \cdot 0.8 \cdot 0.333 + 224 \cdot 0.8 \cdot 0.667 + 180 \cdot 0.412 \cdot 0.5 + 188 \cdot 0.412 \cdot 0.5}{0.8 \cdot 0.333 + 0.8 \cdot 0.667 + 0.412 \cdot 0.5 + 0.412 \cdot 0.5} \simeq 208.47$

3.3 Example Application to Night Sky Live Data: Accuracy and Complexity

The fuzzy logic based transformation formula has been tested and is in practical use with the Night Sky Live project (NSL) (Nemiroff et al. 2005). Purposes of the global Night Sky Live (NSL) network of fisheye CONtinuous CAMeras (CONCAMs) include the ability to monitor and archive the entire bright night sky, track stellar variability, and search for transients. Fully 2π steradians – half the sky – are monitored passively by each CONCAM, without tracking. Currently, the NSL project deploys nine CONtinuous CAMeras (CONCAMs) at many of the world's premier observatories. When the Moon is down, CONCAMs take 180-second exposures every 236 seconds, and can detect stars down to visual magnitude 6.8 near the image center.

The fuzzy logic based transformation formula is used by NSL for converting the celestial coordinates to image coordinates, so known catalogued stellar objects can be associated with PSFs that appear in the Night Sky Live frames. One simple task that is enabled by this transformation formula is the annotation of the all-sky images with the names of bright stars, constellations and planets. This task is mostly used for educational or "cosmetic" purposes. A more important task is the automatic detection of non-catalogued objects. This task is required for automatic detection of meteors, comets, novae and supernovae, as well as other astronomical phenomena visible in the night sky. The following algorithm uses the transformation formula in order to associate PSFs in the image to stellar objects.

```
1. function check_stars(image, date_time)
2. image_PSFs ← GetPSFs(image)
3. for each image_PSF_cords in image_PSFs do
4. begin
5.      min_distance ← ∞
6.      for each star in catalog do
7.          star_celestial_cords ← CelestialCoordinates(star, date_time)
8.          if InView(star_celestial_cords) then
9.              image_cords ← F(star_celestial_cords)
10.              if distance(image_cords,image_PSF_cords) < min_distance then
11.                  min_distance ← distance(image_cords, image_PSF_cords)
12.         end if
13.     end for
14.     if min_distance < TOLERANCE then
15.         associate(image_PSF_cords,star)
16. end for
```

The function F transforms celestial coordinates to image coordinates based on the fuzzy logic models described in this paper. The function *GetPSFs* returns a list of coordinates of the PSF peaks that appear in the picture. This function can be implemented by using some available algorithms for detection of sources from astronomical images such as *SExtractor* (Bertin & Arnouts 1995). The function *InView* returns *true* if its argument coordinates are inside the relevant view of the optical device. In the inner loop the algorithm searches the catalog for a star that should appear closest to the center of the PSF. Since the hardware used for the Night Sky Live project currently cannot get deeper than magnitude 6.8, the catalog being used is a subset of Hipparcos catalog (ESA 1997) that is sure to include objects this bright. In line 14, the minimum distance found in the inner loop is compared with a constant value *TOLERANCE* that is a tolerance value. Only if the distance is smaller than *TOLERANCE* then *image_cords* and *image_PSF_cords* are considered as referring to the same star.

In Figure 3.4 taken by the Mauna Kea CONCAM, the names of bright stars and constellations were annotated automatically using the above transformation. The coordinates of one bright point (appears at the lower right of the frame) could not be associated with any catalogued object so it was automatically marked with a yellow square. This object is believed to be a meteor.

The present fuzzy-logic algorithms allow practically 100 percent chance of accurate identification for NSL stars down to a magnitude of 5.6. We are currently unaware of any exceptions. A previously used NSL identification algorithm that employed a straightforward analytic transformation was only accurate to about magnitude 3.5, although that was somewhat dependent on the NSL station. This dramatic improvement was the driving impetus for the creation of this paper.

A useful by-product of the newly accurate identifications is the automatic generation of photometry files. The ability to associate each PSF with a cataloged star allows the system to provide continuous monitoring of many bright stars. This information is provided in the form of text (XML tagged) files. Each frame produces one text file that lists all PSFs that were detected in the frame and the name and catalogue number of the stellar object associated with it. It also lists some additional data about each detected object such as the previously cataloged visual magnitude, spectral type, and celestial coordinates. Identification allows other algorithms to process the frame and match each star with real time photometric data such as estimated counts of the background and the counts of the PSF. For each bright star, the average of the brightest 1, 5, 9, 16 and 25 pixels are listed. For dimmer stars, however, the realized PSF is much smaller so only the average of the top five pixels is listed. The on-line database allows browsing the records for past events. Since the photometry data itself is out of the scope of this paper, more information about it can be found at (Nemiroff et al. 2005).

The inverse computation of the presented transformation formula (converting angle and distance to altitude and azimuth) can be built in the same method described in this paper, with the exception of using the angle and distance for defining the membership functions, while the azimuth and altitude are used as the crisp output values of the fuzzy rules. This transformation formula is currently being used by NSL for computing 3D trajectories of meteors detected by the twin CONCAMs located at Mauna Kea and Haleakala.

3.3.1 Computation Accuracy

In order to check accuracy, the combined f_1 and f_2 models were used to compute the image coordinates of some 150 stars recorded by the NSL project (Nemiroff et al. 2005). The NSL project uses fisheye images with an extreme viewable angle of 180^o. For each pair of image (x, y) coordinates, we calculated the Euclidean distance from the location calculated by the model and the location where the star appeared in the image. We took into considerations the average of the 150 Euclidean distances (the average error) and the worst case error, which is the longest distance among all computed 150 Euclidean distances. Table 3.2 below shows the accuracy levels according to the number of reference stars that were used in order to build both fuzzy logic models (the altitude f_2 model and the azimuth f_1 model).

Stars in f_2	Stars in f_1	Average Error (pixels)	Maximum Error (pixels)
50	40	3.2	4.8
42	40	3.2	4.8
42	30	3.5	5.5
36	30	3.6	5.7
32	20	4.4	6.2
30	20	5.2	8.0
20	20	6.5	10.2

Table 3.2: Accuracy Level of the Transformation as a Factor of the Number of Reference Stars

3.3.2 Computation Complexity

The complexity of the computation is a function of the number of fuzzy rules and fuzzy sets in the model, which, in turn, is depended on the number of reference stars. The accuracy needed to identify stars unambiguously is depended on the density of stars per pixel in the frame. Clearly, the higher the average star density, the greater the accuracy needed to avoid confusion.

Experiments suggest that accuracy within five pixels will allow automatic analysis of *Night Sky Live* frames. As shown in Table 3.2, this can be achieved using a fuzzy logic model that is built with constant number of around 80 well spread out reference stars. Therefore, the final fuzzy logic model contains a constant number of around 80 rules and around 160 fuzzy sets, so the theoretical complexity of the computation is O(1). Practically, the CPU time that is required for the computation is negligible, and around 1000 coordinates are transformed in less than one second by a system equipped with an Intel Pentium III processor at 850 MHZ and 128MB of RAM.

3.4 Conclusions

Inaccuracies of optics and hardware lead to non-trivial transformation formulae that are sometimes required for the automatic analysis of the digital frames. A fuzzy logic based method does not achieve an exact solution to the problem, but rather provides simple and maintainable transformation formulae with an accuracy that makes them more useful in practice than previously attempted straightforward analytic transformations (based on the linear transformation $R = k\epsilon$, where R is the radial distance from x_{zen},y_{zen}, ϵ is the altitude and k is some constant). When applied to the NSL project, the presented method is accurate enough to perform automatic processing of all-sky images in order to associate PSFs in the frames with their corresponding stellar objects nearly 100 percent of the time down to magnitude 5.6, and automatically detect non-catalogued bright objects. This technique demonstrates that it is possible to use fuzzy logic based algorithms to reduce the affect of minor defects and inaccuracies of the optical hardware and/or slightly inaccurate telescope orientation information. For NSL frames, only 80 reference stars are required to build the fuzzy logic model so that automatic identifications do not form a computational bottleneck.

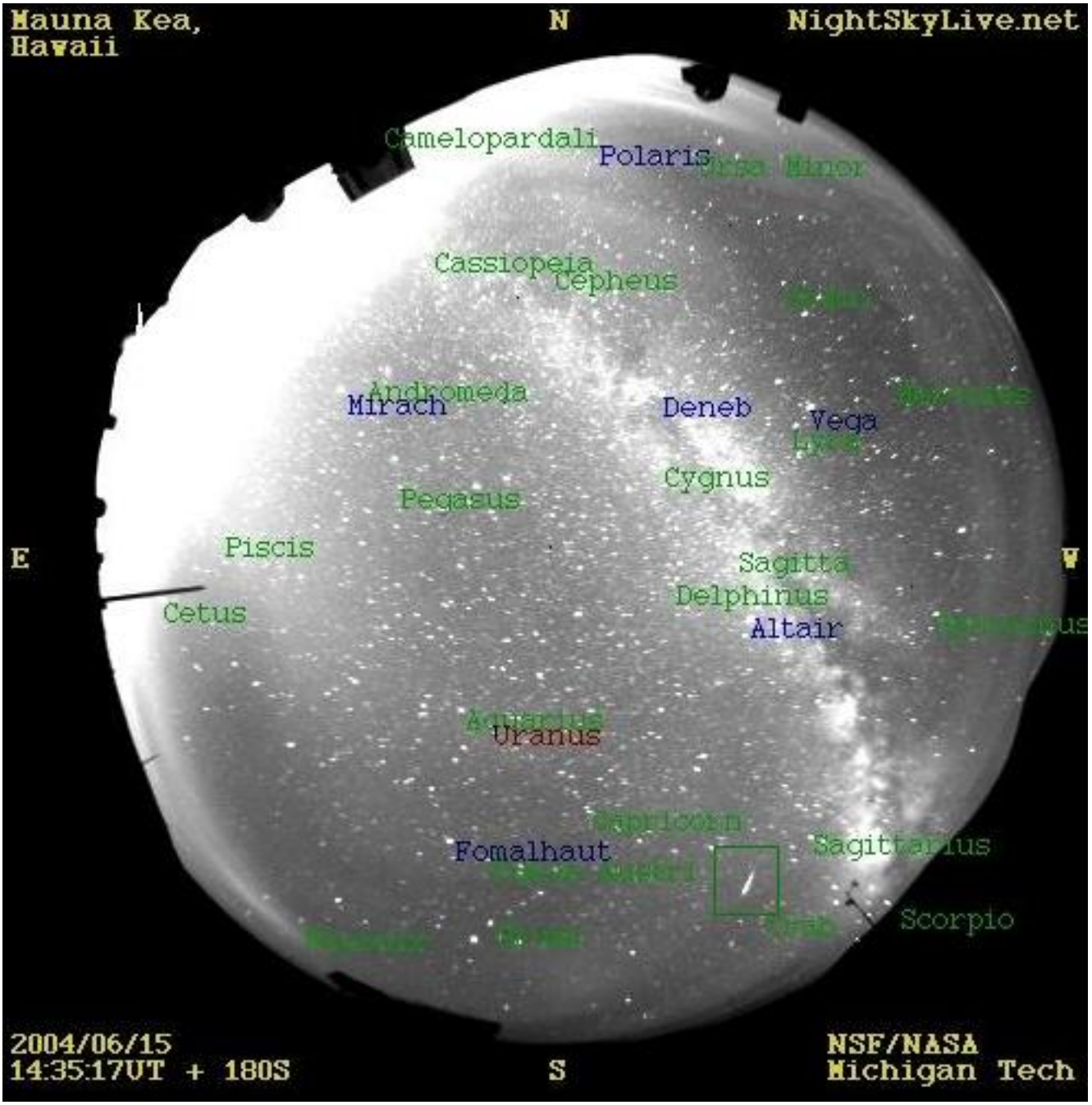

Figure 3.4: A Night Sky Live picture with labeled stars, constellations, a planet and a meteor (boxed) processed using the fuzzy logic based transformation formula.

Bibliography

Bertin, E. and Arnouts, S., 1996, A&AS, 117, 393

ESA, 1997, ESA SP-1200

Nemiroff, R. J., & Rafert, J. B., 1999, PASP, 111, 886

Takagi T., Sugeno M., 1983, Proc. of the IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision analysis, 55

Takagi T., Sugeno M., 1985, IEEE Trans. Syst. Man & Cybern., 20, 116

Zadeh, L.A., 1965, Information and Control, 8, 338

Zadeh, L.A., 1978, International Journal of Man-Machine Studies, 10, 395

Zadeh, L.A., 1983, Memorandum No. UCB/ERL M83/41, University of California, Berkeley

Zadeh, L.A., 1994, IEEE Software, 11, 48

Chapter 4

A Fuzzy Logic-based Algorithm for Cosmic-ray Hit Rejection From Single Images

Abstract: An algorithm for cosmic-ray rejection from single images is presented. The algorithm is based on modeling human perception using fuzzy logic. The proposed algorithm is specifically designed to reject multiple-pixel cosmic ray hits that are larger than some of the point spread functions of the true astronomical sources. Experiments show that the algorithm can accurately reject ~97.5% of the cosmic rays hits, while mistakenly rejecting 0.02% of the true astronomical sources. The major advantage of the presented algorithm is its computational efficiency.

*This chapter was published in: Shamir, L., 2005, A Fuzzy Logic-Based Algorithm for Cosmic Ray Hit Rejection from Single Images, *Astronomical Notes* (Astronomische Nachrichten), Vol. 326(6), pp. 428–431.

4.1 Introduction

The presence of cosmic ray hits in astronomical CCD frames is frequently considered as a disturbing effect. Cosmic rays add an undesirable signal to astronomical images, and can weaken algorithms for astronomical image processing. For instance, it can decrease the compression factor of astronomical image compression algorithms (Offenberg et al. 1999), and can disturb the operation of autonomous astronomical pipelines (Axelrod et al. 2004; Becker et al. 2004; Otuairisg et al. 2004). Exposures taken at high altitude observatories get more cosmic ray hits than sea level observatories. This becomes even more significant in space-based telescopes located far from a planetary magnetic field (Offenberg et al. 1999).

Several methods for cosmic ray hit rejection have been proposed. One common technique is by comparing several exposures of the same field (Fixsen et al. 2000; Windhorst, Franklin & Neuschaefer 1994; Shaw & Horne 1992). However, exposures of the same fields are not always available. Other approaches, such as (Salzberg et al. 1995; Rhoads 2000; Van Dokkum 2001; Pych 2004), have been proposed in order to perform cosmic ray hit rejection in a single CCD exposure. Rhoads (2000) performs filtering by adapted point spread functions, Van Dokkum (2001) uses Laplacian edge detection, Pych (2004) performs an analysis of the histogram of the image data and Salzberg et al. (1995) use an artificial neural network. The more difficult cases are when some of the multiple-pixel cosmic ray hits are larger than some of the PSFs of true astronomical sources (Van Dokkum 2001). Thus, reasonably trained humans can usually perform this task with a considerable percentage of accuracy. In this paper we present an algorithm that aims to reject cosmic ray hits based on human perception, and implemented using Fuzzy Logic modeling (Zadeh 1965, 1988). In section 2 we discuss the human intuition of detecting cosmic ray hits in an astronomical exposure, in section 3 we describe the fuzzy logic model, in section 4 we describe the computation process, in section 5 we present how the fuzzy logic model is used, and in section 6 we discuss the performance of the algorithm.

4.2 Manual Detection of Cosmic-ray Hits

Cosmic ray hits in astronomical exposures are usually noticeably different then PSFs of true astronomical sources, and a reasonably trained human can usually tell between the two. One examining an astronomical frame can notice that cosmic ray hits are usually smaller than PSFs of astronomical sources, and their edges are usually sharper.

Although many cosmic ray hits are not larger than just one pixel, in some cases they can be larger than some of the point spread functions of astronomical sources (Van Dokkum 2001). An observer trying to manually detect cosmic ray hits in an astronomical frame would probably examine the edges and the surface size of the peaks. For instance, if the surface size of the peak is very small and it has sharp edges, it would be classified as a cosmic ray hit. If the surface size of the peak is larger and its edges are not very sharp, it would be probably classified as a PSF of an astronomical source. Since some of the cosmic ray hits have only one or two sharp edges, it is also necessary to examine the sharpest edge of the PSF.

This intuition can be summarized by a set of intuitive natural language rules such as:

1. If the peak is small and the edges are sharp then the peak is a cosmic ray hit.
2. If the peak is large and the edges are not very sharp then the peak is not a cosmic ray hit.
3. If the peak is medium and most of the edges are not very sharp except from one extremely sharp edge then the peak is a cosmic ray hit.
4. If the peak is small and the edges are moderately sharp then the peak is a cosmic ray hit.
5. If the peak is large and the edges are not sharp except from one edge that is moderately sharp then the peak is not a cosmic ray hit.

4.3 A Human Perception-based Fuzzy Logic Model

One of the advantages of fuzzy logic is its ability to mathematically describe human intuition (Zadeh 1965, 1988, 1994). The first step in compiling the rules of intuition described above into a fuzzy logic model is to define the antecedent (input) and consequent (output) fuzzy variables. The antecedent variables in this model are the surface size of the peak (in pixels), the sharpness of the sharpest edge (in σ, where σ is the estimated noise) and the average sharpness of the edges (also in σ). The consequent variable is the classification of the peak. The domain of this variable is {Yes/1,No/0}, such that YES/1 means that the peak is classified as a cosmic ray hit and NO/0 means that the peak is classified as a PSF of a true astronomical source. Therefore, the model can be defined by the following function *f*:

f: *surface_size*, *sharpest_edge*, *average_edge* $\longmapsto$ {0,1}

The surface size is determined by counting the pixels around the peak that are at least 3σ brighter than the local background. When a pixel less than 3σ above the local background is reached, the pixel is not counted and the edge sharpness is determined as the difference between the value of that pixel and the value of its neighboring pixel (in the direction of the peak).

The fuzzy sets defined for the surface size of the peak are described in Figure 4.1, and the membership functions of the fuzzy sets are defined by the following formulae:

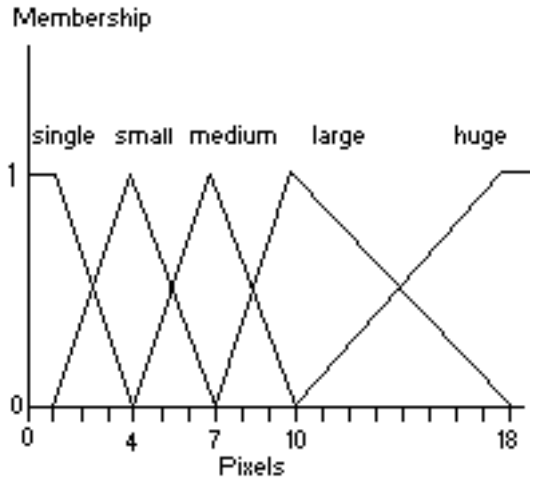

Figure 4.1: The membership functions of the fuzzy sets defined for the surface size: single, small, medium, large, huge.

$$F_{single}(x) = \begin{cases} 1 - \frac{x-1}{3} & 1 \leq x \leq 4 \\ 1 & 0 < x < 1 \\ 0 & x < 0 \text{ or } x > 4 \end{cases}$$

$$F_{small}(x) = \begin{cases} \frac{x-1}{3} & 1 \leq x < 4 \\ 1 - \frac{x-4}{3} & 4 \leq x \leq 7 \\ 0 & x < 1 \text{ or } x > 7 \end{cases}$$

$$F_{medium}(x) = \begin{cases} \frac{x-4}{3} & 4 \leq x < 7 \\ 1 - \frac{x-7}{3} & 7 \leq x \leq 10 \\ 0 & x < 4 \text{ or } x > 10 \end{cases}$$

$$F_{large}(x) = \begin{cases} \frac{x-7}{3} & 7 \leq x < 10 \\ 1 - \frac{x-10}{8} & 10 \leq x \leq 18 \\ 0 & x < 7 \text{ or } x > 18 \end{cases}$$

$$F_{huge}(x) = \begin{cases} \frac{x-10}{8} & 10 \leq x < 18 \\ 1 & 18 \leq x \\ 0 & x < 10 \end{cases}$$

The antecedent variables *sharpest_edge* and *average_edge* use the same fuzzy sets. The fuzzy sets defined for these variables are *low*, *moderate*, *sharp* and *extreme*, as described in Figure 4.2.

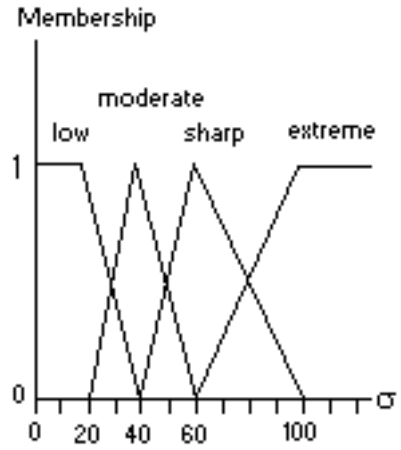

Figure 4.2: The membership functions of the fuzzy sets defined for the edge sharpness

The membership functions of the fuzzy sets described in Figure 4.2 are defined by the following formulae:

$$F_{low}(x) = \begin{cases} 1 - \frac{x-20}{20} & 20 \leq x \leq 40 \\ 1 & 0 \leq x < 20 \\ 0 & x < 0 \text{ or } x > 40 \end{cases}$$

$$F_{moderate}(x) = \begin{cases} \frac{x-20}{20} & 20 \leq x < 40 \\ 1 - \frac{x-40}{20} & 40 \leq x \leq 60 \\ 0 & x < 20 \text{ or } x > 60 \end{cases}$$

$$F_{sharp}(x) = \begin{cases} \frac{x-40}{20} & 40 \leq x < 60 \\ 1 - \frac{x-60}{40} & 60 \leq x \leq 100 \\ 0 & x < 40 \text{ or } x > 100 \end{cases}$$

$$F_{extreme}(x) = \begin{cases} \frac{x-60}{40} & 60 \leq x < 100 \\ 1 & 100 \leq x \\ 0 & x < 60 \end{cases}$$

The fuzzy rules are defined using the membership functions of the antecedent variables and the domain of the consequent variable {0,1}, and are based on the natural language rules of intuition described in section 4.2.

single, low, low $\mapsto$ 0
single, low, moderate $\mapsto$ 1
single, low, sharp $\mapsto$ 1
single, low, extreme $\mapsto$ 1
single, moderate, low $\mapsto$ 1
single, moderate, moderate $\mapsto$ 1
single, moderate, sharp $\mapsto$ 1
single, moderate, extreme $\mapsto$ 1
single, sharp, low $\mapsto$ 1
single, sharp, moderate $\mapsto$ 1
single, sharp, sharp $\mapsto$ 1
single, sharp, extreme $\mapsto$ 1
single, extreme, low $\mapsto$ 1
single, extreme, moderate $\mapsto$ 1
single, extreme, sharp $\mapsto$ 1
single, extreme, extreme $\mapsto$ 1
small, low, low $\mapsto$ 0
small, low, moderate $\mapsto$ 0
small, low, sharp $\mapsto$ 1
small, low, extreme $\mapsto$ 1
small, moderate, low $\mapsto$ 1
small, moderate, moderate $\mapsto$ 1
small, moderate, sharp $\mapsto$ 1
small, moderate, extreme $\mapsto$ 1
small, sharp, low $\mapsto$ 1
small, sharp, moderate $\mapsto$ 1
small, sharp, sharp $\mapsto$ 1
small, sharp, extreme $\mapsto$ 1
small, extreme, low $\mapsto$ 1
small, extreme, moderate $\mapsto$ 1
small, extreme, sharp $\mapsto$ 1
small, extreme, extreme $\mapsto$ 1
medium, low, low $\mapsto$ 0
medium, low, moderate $\mapsto$ 0
medium, low, sharp $\mapsto$ 0
medium, low, extreme $\mapsto$ 0
medium, moderate, low $\mapsto$ 0
medium, moderate, moderate $\mapsto$ 0
medium, moderate, sharp $\mapsto$ 0
medium, moderate, extreme $\mapsto$ 1
medium, sharp, low $\mapsto$ 1
medium, sharp, moderate $\mapsto$ 1
medium, sharp, sharp $\mapsto$ 1
medium, sharp, extreme $\mapsto$ 1
medium, extreme, low $\mapsto$ 1
medium, extreme, moderate $\mapsto$ 1
medium, extreme, sharp $\mapsto$ 1
medium, extreme, extreme $\mapsto$ 1
large, low, low $\mapsto$ 0
large, low, moderate $\mapsto$ 0
large, low, sharp $\mapsto$ 0
large, low, extreme $\mapsto$ 0
large, moderate, low $\mapsto$ 0
large, moderate, moderate $\mapsto$ 0
large, moderate, sharp $\mapsto$ 0
large, moderate, extreme $\mapsto$ 0
large, sharp, low $\mapsto$ 0
large, sharp, moderate $\mapsto$ 0
large, sharp, sharp $\mapsto$ 1
large, sharp, extreme $\mapsto$ 1
large, extreme, low $\mapsto$ 0
large, extreme, moderate $\mapsto$ 1
large, extreme, sharp $\mapsto$ 1
large, extreme, extreme $\mapsto$ 1
huge, low, low $\mapsto$ 0
huge, low, moderate $\mapsto$ 0
huge, low, sharp $\mapsto$ 0
huge, low, extreme $\mapsto$ 0
huge, moderate, low $\mapsto$ 0
huge, moderate, moderate $\mapsto$ 0
huge, moderate, sharp $\mapsto$ 0
huge, moderate, extreme $\mapsto$ 0
huge, sharp, low $\mapsto$ 0
huge, sharp, moderate $\mapsto$ 0
huge, sharp, sharp $\mapsto$ 0
huge, sharp, extreme $\mapsto$ 0
huge, extreme, low $\mapsto$ 0
huge, extreme, moderate $\mapsto$ 0
huge, extreme, sharp $\mapsto$ 0
huge, extreme, extreme $\mapsto$ 0

4.4 The Computation Process

The computation process is based on *product* inferencing and *weighted average* defuzzification (Takagi & Sugeno 1983, 1985), and can be demonstrated by the following example:
Suppose that *surface_size*=5, *average_edge*=35 and *sharpest_edge*=60. The membership of the value of the fuzzy variable *surface_size* in the fuzzy set *small* is $1 - \frac{5-4}{3} = \frac{2}{3}$ and the membership in the fuzzy set *medium* is $\frac{5-4}{3} = \frac{1}{3}$. The membership in the fuzzy sets *single*, *large* and *huge* is 0.

Similarly, the membership of the value of the fuzzy variable *average_edge* in the fuzzy set *moderate* is $\frac{35-20}{20} = \frac{3}{4}$ and in the fuzzy set *low* it is $1 - \frac{35-20}{20} = \frac{1}{4}$. 60 is the point where the membership function of the fuzzy set *sharp* reaches its maximum of unity, so the membership of 60 in *sharp* is $\frac{60-40}{20} = 1$, while its membership in all other fuzzy sets is 0.

In the inference computation stage, the first rule $single, low, low \mapsto 0$ is dependent on the fuzzy sets *single*, *low* and *low*, so the strength of this rule is $0 \cdot \frac{1}{4} \cdot 0 = 0$. When using *product* inferencing, if one of the fuzzy variables has a membership level of 0 (no membership) in one of the fuzzy sets referred by the rule, the rule does not have any effect on the final result of the computation. In this example, the only fuzzy rules that have non-zero membership values in all of 3 fuzzy sets are:

1. small, low, sharp $\mapsto$ 1
2. small, moderate, sharp $\mapsto$ 1
3. medium, low, sharp $\mapsto$ 0
4. medium, moderate, sharp $\mapsto$ 0

The membership of the values of the antecedent variables in the fuzzy sets of rule 1 (*small*, *low* and *sharp*) are $\frac{2}{3}$, $\frac{1}{4}$, 1 respectively. Similarly, the membership of the antecedent variables in the fuzzy sets of rule 2 are $\frac{2}{3}, \frac{3}{4}$,1, the membership in the fuzzy sets of rule *3* are $\frac{1}{3}$, $\frac{1}{4}$, 1, and the membership in the fuzzy sets of rule *4* are $\frac{1}{3}$, $\frac{3}{4}$, 1. Since the defuzzification is performed using the *weighted average* defuzzification method (Takagi & Sugeno 1983, 1985), the computed value of the consequent variable is:

$$\frac{\frac{2}{3}\cdot\frac{1}{4}\cdot 1\cdot 1+\frac{2}{3}\cdot\frac{3}{4}\cdot 1\cdot 1+\frac{1}{3}\cdot\frac{1}{4}\cdot 1\cdot 0+\frac{1}{3}\cdot\frac{3}{4}\cdot 1\cdot 0}{\frac{2}{3}\cdot\frac{1}{4}\cdot 1+\frac{2}{3}\cdot\frac{3}{4}\cdot 1+\frac{1}{3}\cdot\frac{1}{4}\cdot 1+\frac{1}{3}\cdot\frac{3}{4}\cdot 1} = \frac{2}{3}$$

The value of the consequent variable is handled such that values greater than 0.5 are classified as cosmic ray hits. Otherwise, the values are classified as non-cosmic ray hits. Since in the above computation the output value is $\frac{2}{3}$, the input values in this example are classified as a cosmic ray hit.

4.5 Using the fuzzy logic model

The fuzzy logic model is used in order to classify peaks in the frame as cosmic ray hits or non-cosmic ray hits. Each cosmic ray hit or PSF has one (or several) brightest pixels that can be considered as the center of the peak. In the presented algorithm, searching for peaks in a FITS frame is performed by comparing the value of each pixel with the values of its 8 neighboring pixels. If the pixel is equal or brighter than its 8 neighboring pixels, it is considered as a center of a peak. After finding the peaks in the frame, the fuzzy logic model is applied on the peaks in order to classify them as cosmic ray hits or non-cosmic ray hits.

If a background pixel happens to be brighter than its 8 neighboring pixels, it will be mistakenly considered as a center of a peak. The probability of that event is $0.5^8 = 0.00390625$. For instance, in an astronomical frame of 1024×1024, 4096 background pixels are expected to be mistakenly considered as peaks. However, since the computation process is relatively fast, these additional peaks do not significantly slow down the algorithm.

4.6 Performance of the Algorithm

Measurements of the performance of the algorithm were taken using 24 *Night Sky Live* (Perez-Ramirez, Nemiroff & Rafert 2004) all-sky exposures. Each NSL frame contains an average of 6 noticeable cosmic ray hits brighter than 20σ, and around 1400 astronomical sources brighter than 20σ over their local background. Out of 158 cosmic ray hits that were tested, the algorithm did not reject 4, and mistakenly rejected 6 true astronomical sources out of a total of 31,251 PSFs. These numbers do not present better accuracy than other reported cosmic ray rejection algorithms such as (Rhoads 2000; Van Dokkum 2001; Pych 2004), but the proposed algorithm has a clear advantage in terms of computational efficiency. While some of the above algorithms are relatively slow, the presented algorithm can process a 1024×1024 integer FITS frame in less than 12 seconds, using a system equipped with a SUN ULTRA-1 at 200 MHZ. This advantage can be significant in systems such as the *Night Sky Live* all-sky survey, in which each node of the network takes an exposure every 3 minutes and 56 seconds.

4.7 Conclusions

In this paper we presented an algorithm for cosmic ray hits rejection that is based on fuzzy logic modeling of human intuition. Experiments show that the algorithm rejects ∼97.5% of the cosmic ray hits, but also rejects ∼0.02% of the PSFs of true astronomical sources. The algorithm is implemented and used regularly by the Night Sky Live all-sky survey, in which the system automatically searches for transients and monitors known bright astronomical sources.

Bibliography

Axelrod, T., Connolly, A., Ivezic, Z., Kantor, J., Lupton, R., Plante, R., Stubbs, C., Wittman, D., 2004, AAS Meeting 205, 108.11

Becker, A.C., Rest, A., Miknaitis, G., Smith, R.C., Stubbs, C., 2004, AAS Meeting 205, 108.12

Fixsen, D.J., Offenberg, J.D., Hanisch, R.J., Mather, J.C., Nieto-Santisteban, M.A., Sengupta, R., Stockman, H.S., 2000, PASP, 112, 1350.

Offenberg, J.D., Sengupta, R., Fixsen, D.J., Stockman, H.S., Nieto-Santisteban, M.A., Stallcup, S., Hanisch, R.J., Mather, J.C., 1999, in ASP Conf., 172, 141

Otuairisg, S., Golden, A., Butler, R.F., Shearer, A., Voisin, B., 2004, Ground-based Telescopes, 5493, 467

Perez-Ramirez, D., Nemiroff, R.J., Rafert, J.B., 2004, AN, 325, 568

Pych, W., 2004, PASP, 116, 148

Rhoads, J.E., 2000, PASP, 112, 703

Salzberg, S., Chandar, R., Ford, H., Murphy, S. K., White, R., 1995, PASP, 107, 279

Shaw, R. A., Horne, K., 1992, ASP Conf., 25, Astronomical Data Analysis, Software and Systems

Takagi, T., Sugeno, M.: 1983, Proc. of the IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision analysis, 55

Takagi, T., Sugeno, M., 1985, IEEE Trans. Syst. Man & Cybern., 20, 116

Van Dokkum, P.G., 2001, PASP, 113, 1420

Windhorst, R.A., Franklin, B.E., Neuschaefer, L.W., 1994, PASP, 106, 798

Zadeh, L.A., 1965, Information and Control, 8, 338

Zadeh, L.A., 1988, Fuzzy Logic, Computer, 21, 83

Zadeh, L.A., 1994, IEEE Software, 11, 48

Appendix A

Computational Complexity

The computational complexity of the proposed algorithm is determined by the number of fuzzy rules. Since the number of rules in the model is constant, the complexity of the computation of one pixel is also constant, and the complexity of the computation of an image is linear to the number of pixels.

Table 4.1 shows the time required by each algorithm to process a 1024×1024 FITS image using a SUN ULTRA-1 machine at 200 MHz.

Algorithm	**Time (seconds)**
Van Dokkum	106
Pych	34
Rhoads	>75
The Proposed Algorithm	12

Table 4.1: Time required to process a 1024×1024 FITS image

As reported in (Van Dokkum 2001), the time required by the Van Dokkum cosmic ray hit rejection algorithm to process a 800×800 FITS image is 65 seconds using a Sun Ultra-1 machine at 200 MHz. Since it is also stated that the complexity is linear to the number of pixels, the information provided in the paper can be interpolated so it can be deduced that the time required to process a 1024×1024 image is 1 minute and 46 seconds.

The Pych algorithm (Pych 2004) was tested by compiling and running the C-code (available at http://www.camk.edu.pl/~pych/) using a SUN Ultra-1 machine.

As reported in (Rhoads 2000), Rhoads algorithm requires 10 minutes for processing a 2048×4096 FITS image using a SUN Ultra-30 machine at 295 MHz, which is favorably comparable to the ULTRA-1. Interpolating this information shows that it should take 1 minute and 15 seconds for Rhoads algorithm to process 1024×1024 FITS image using a SUN Ultra-30 machine. Since the Ultra-30 machine is faster than the Ultra-1, it can be concluded that by using an Ultra-1, the same task should take no less than 75 seconds. However, since the performance of the two systems lies within the same order of magnitude, the processing time under Ultra-1 is not expected to be dramatically greater than 75 seconds.

Chapter 5

All-sky Relative Opacity Mapping Using Night Time Panoramic Images

Abstract: An all-sky cloud monitoring system that generates relative opacity maps over many of the world's premier astronomical observatories is described. Photometric measurements of numerous background stars are combined with simultaneous sky brightness measurements to differentiate thin clouds from sky glow sources such as air glow and zodiacal light. The system takes a continuous pipeline of all-sky images, and compares them to canonical images taken on other nights at the same sidereal time. Data interpolation then yields transmission maps covering almost the entire sky. An implementation of this system is currently operating through the Night Sky Live network of CONCAM3s located at Cerro Pachon (Chile), Mauna Kea (Hawaii), Haleakala (Hawaii), SALT (South Africa) and the Canary Islands (Northwestern Africa).

*This chapter was published in: Shamir, L., Nemiroff, R. J., 2005, All-sky Relative Opacity Mapping Using Night Time Panoramic Images, *Publications of the Astronomical Society of the Pacific.*

5.1 Introduction

The effectiveness of ground-based telescopes is modulated by atmospheric opacity. Previously implemented systems created to monitor clouds and sky opacity have been limited to monitoring a single point, usually the zenith (Ananthasubramanian et al. 2004; Chamberlin 2003). Single-point opacity measurements, however, inherently ignore even dramatic non-uniform opacity across the night sky. Information regarding the opacity of the entire sky can be used in a variety of situations. For example, a robotic telescope surveying the entire sky would be used inefficiently if it observed a cloudy part of the sky, while other parts of the sky remained relatively clear. Human observers also benefit from using these data by manually selecting from equally desirable targets based on local sky transparency (SpaceWatch, private communication).

A cloud monitoring system utilizing the thermal infrared and covering 135 degrees^2 operates at Apache Point (Hogg et al. 2001; Hull Limmongkol & Siegmund). The Apache Point system gives a simple boolean indication of the existence of a cloud at a certain point in the sky. The system is used to accept or reject collected data.

Providing a real-time fisheye view of the optical night sky, the *Night Sky Live* (NSL) network of CONtinuous CAMeras (CONCAMs) automatically creates a pipeline of all-sky images that is routinely used by astronomers as a contemporary cloud monitor (Rutten 2003; Pereira et al. 2000, 2005). NSL CONCAM3 images are slightly more sensitive than the human eye, particularly in the CCD-sensitive red part of the spectrum. This allows sub-visual cirrus clouds, airglow, and other sky glows such as zodiacal light to be easily discernable on the images, along with over a thousand stars. This NSL network and its opacity mapping software is only one realization of a more general visual night sky monitoring system that can generate opacity maps. In this paper, we therefore first delineate the workings of a general night sky opacity mapping system, and later describe our specific NSL CONCAM implementation. In Section 5.2 we describe the construction of the database of canonical images, in Section 8.2.3 we describe the process in which opacity maps are generated, in Section 6.4 we discuss the accuracy of the system and in Section 6.2 we discuss our implementation of the sky opacity monitoring system using the Night Sky Live network.

5.2 Canonical Image Database

A first step to building an opacity map is to create a comparison image from which any image can be compared. This image can be composed of images taken at the same sidereal time as the given frame, and will be called the "canonical image". One can think of the "canonical image" as a hypothetical image taken when the sky is completely clear. In reality, the canonical image is better created from several relatively clear images taken at the same sidereal time. To proceed, the system first needs to acquire a sufficient database of relatively clear images.

One approach of obtaining this database is to manually select images taken at clear nights. However, observing images by eye is a time-consuming task, and the required human resources are not always available. Moreover, since thin cirrus clouds are hard to detect, a human observer might consider some images as *clear* while in fact parts of the sky are covered with light clouds.

Another approach, which we have found more appropriate, is to use an algorithm that automatically searches for images taken at clear nights. The algorithm is based on applying a star recognition algorithm that associates the point spread functions in the image with star catalog entries (Shamir & Nemiroff 2005a). After the point spread functions are associated with known stars, the algorithm checks if the stars that are brighter than a pre-set visual magnitude are detected at a pre-set statistical confidence level. If the PSFs of a sufficient percentage of expected stars are found in the image, the image is classified as *clear*, and added to the database.

The percentage of stars visible at a certain frame is dependent not only on the sky clarity, but also on the darkness of the sky. For instance, faint stars near the galactic plane will not be detected as easily as stars far from the Milky Way. Therefore, the pre-set threshold of visual magnitude should be chosen such that stars near the center of the galactic plane brighter than the threshold can be detected. Since this problem becomes more substantial when the moon is up, canonical images are added to the database only when the moon is down.

When an image is added to the database, it is averaged with all other images taken at the same sidereal time that are already present. Averaging the images gives a better signal to noise ratio than using a single exposure. The large pixel size and the absence of moving parts in the all-sky monitoring system simplify the process of image co-adding.

The algorithm described above is used in order to build the database while the system is operating by adding all images that are classified by the algorithm as *clear*. This policy allows the system to provide useful data even in the first days of its operation, while autonomously improving the quality of the data with time.

An alternative approach would be to iterate toward a standard canonical image that does not change after a given number of images are averaged. Once the standard canonical image is established, it is understood that all opacity measurements are given relative to this standard.

5.3 All-Sky Relative Opacity Maps

The building of relative opacity maps requires several logical steps. The first is the building of a canonical image taken from previous images taken at the same sidereal time, as discussed last section. Another step is the rejection of pixels dominated by cosmic-ray generated counts (Shamir & Nemiroff 2005b). Such rejections might make use of the non-point source nature of cosmic ray splashes, or the fact that the apparent cosmic ray "source" cannot be found on the canonical frame. Next, bright planets and variable stars with relatively large amplitude such as Algol (Muzzin Shamir & Nemiroff) are also rejected using a star recognition algorithm such as (Shamir & Nemiroff 2005a). Any PSF that is a given σ brighter than its background is assigned with two values: a stellar intensity, and an estimated intensity of the background. The values assigned to the PSFs are then compared with the values of the PSFs of a canonical image. The comparison is based on Equations 8.1 and 8.2.

$$M_{star} = T \cdot I_{star} + T \cdot I_{space} + I_{cloud} \tag{5.1}$$

$$M_{\mathrm{background}} = T \cdot I_{space} + I_{cloud} \tag{5.2}$$

Where M_{star} is the intensity of the light coming from pixels at the location of a star in the given image, $M_{\mathrm{background}}$ is the intensity of the light coming from pixels *not* at the location of a star, I_{star} is the intensity of the star, I_{space} is the intensity of light from background space, I_{cloud} is the intensity of the cloud covering the star and T is the transmission around the star. Using Equations 8.1 and 8.2, the transmission is determined by Equation 8.3.

$$T = \frac{(M_{star} - M_{\mathrm{background}})}{I_{star}} \tag{5.3}$$

Where I_{star} is based on measurements taken from the canonical image such that $I_{star} = Mo_{star} - Mo_{\mathrm{background}}$, where Mo_{star} and $Mo_{background}$ are the measured intensity of the star and the background in the canonical image. This calculation of I_{star} is based on the assumption that no light is lost as starlight travels through air on a clear night. This assumption, however, is not true, and provides a simplification of the problem that allows estimating the normalized transmission comparing to a clear night. This simplification gives Equation 8.4.

$$T = \frac{(M_{star} - M_{\mathrm{background}})}{(Mo_{star} - Mo_{\mathrm{background}})} \tag{5.4}$$

Since the desired product is a broad relative transmission map, the transmission of each pixel is calculated by interpolating the transmission measured directly along the lines to the stars in the image. This is performed by choosing, for example, the four nearest stars $S_{left}, S_{top}, S_{right}, S_{bottom}$ such that $|Y_{Sl} - Y_0| < X_0 - X_{Sl}$, $|Y_{Sr} - Y_0| < X_{Sr} - X_0$, $|X_{St} - X_0| < Y_0 - Y_{St}$, $|X_{Sb} - X_0| < Y_{Sb} - Y_0$, where X_{Sn} is the X image coordinate of the star S_n, Y_{Sn} is the Y image coordinate of the star S_n, and (X_0, Y_0) are the image coordinates of the given pixel that its sky transmission is being estimated. After finding the four nearest stars, a two-dimensional linear interpolation of the transmission measured by the four stars is performed, and the calculated value is determined as the relative transmission of the pixel. The computed relative transmission values of all pixels in the frame are then used for generating the all-sky opacity maps such that high relative transmission is an indication of high relative sky opacity, while low relative transmission indicates low relative opacity.

The computed normalized transmission T provided by this method is not intended to be used for scientific purposes such as photometric reduction. However, providing a numeric indication of the estimated atmospheric transmission is believed to have an advantage over a simple boolean clear/cloudy bit of information. The numeric indication can be used by robotic telescopes for the purpose of decision making (for instance, weighing the importance of the observation against the atmospheric transmission) and can also provide human observers with more informative indications regarding the visibility conditions of the area of the sky they observe.

5.4 System Accuracy

The maps generated by the presented system provide an approximation of the relative transmission of the sky. The accuracy of the data is dependent on the noise of the CCD measurements. When the gain is 1, the estimated error of the measured intensity of the star is defined by Equations 5.5 and 5.6.

$$\sigma_{star} = \sqrt{M_{star}} \tag{5.5}$$

$$\sigma_{star0} = \sqrt{Mo_{star}} \tag{5.6}$$

Assuming a perfect detector, when the intensity of the background $M_{background}$ is calculated by averaging the 1600 pixels around the peak of the PSF, the estimated error of the background is defined by Equations 5.7 and 5.8

$$\sigma_{background} = \frac{M_{background}}{\sqrt{1600}} \tag{5.7}$$

$$\sigma_{background0} = \frac{Mo_{background}}{\sqrt{1600}} \tag{5.8}$$

Let σ_Δ and $\sigma_{\Delta 0}$ be the estimated error of $(M_{star} - M_{background})$ and $(Mo_{star} - Mo_{background})$ respectively. σ_Δ and $\sigma_{\Delta 0}$ are defined by Equations 5.9 and 5.10.

$$\sigma_\Delta = \sqrt{\sigma_{star}^2 + \sigma_{background}^2} \tag{5.9}$$

$$\sigma_{\Delta 0} = \sqrt{\sigma_{star0}^2 + \sigma_{background0}^2} \tag{5.10}$$

The fractional standard deviation of the transmission T (calculated by Equation 8.4) is defined by Equation 5.11.

$$\sigma_T = \sqrt{(\frac{\sigma_\Delta}{M_{star} - M_{background}})^2 + (\frac{\sigma_{\Delta 0}}{Mo_{star} - Mo_{background}})^2} \tag{5.11}$$

For example, suppose that $M_{star} = 6000$, $Mo_{star} = 7000$ and $M_{background} = Mo_{background} = 1000$. Using Equation 8.4, T is $\sim$0.83, giving $\Delta M = -2.5 \log 0.83 \simeq 0.2$. Using Equations 5.5 - 5.8, $\sigma_{star} \simeq 77.46$, $\sigma_{star0} \simeq 83.67$ and $\sigma_{background} = \sigma_{background0} = 2.5$. Equations 5.9 and 5.10 give $\sigma_\Delta \simeq 77.5$ and $\sigma_{\Delta 0} \simeq 83.71$, and Equation 5.11 gives $\sigma_T \simeq 0.021$. Therefore, $\sigma_{\Delta M} = -2.5 \log(\frac{0.83 - 0.021 \cdot 0.83}{0.83}) \simeq 0.023$.

Figure 5.1 shows transmission measurements at RA=00h 40m, DEC=56.5^o taken using the *Night Sky Live* implementation of the proposed method described in Section 6.2. The data were recorded at Mauna Kea throughout the night of 10/15/2004. Figure 5.2 shows the calculated values of σ_T.

One problem that may affect the reliability of the system is stellar variability. All-sky monitoring devices such as the *CONCAM*s (Nemiroff & Rafert 1999) are sensitive to variable stars (Muzzin Shamir & Nemiroff), and in some cases can sense the variability of low-amplitude variable stars such as Polaris (Nemiroff et al. 2005). An eclipsed star, for example, might create a false region of high opacity. Figure 5.3 shows the measurements of a variable star (Algol) and a non-variable star (Alpha Cassiopeia) over the period between 12/8/2004 and 12/15/2004. The data were recorded at KPNO using an implementation of the proposed system described in Section 6.2 by taking measurements on different days at the same sidereal time. The graph shows the average of the 9 brightest pixels of the PSFs of the two stars subtracted by the estimated local background. The decreased luminosity of Algol during the eclipse might mistakenly lead the system to conclude that Algol is covered by a cloud. However, since known variable stars

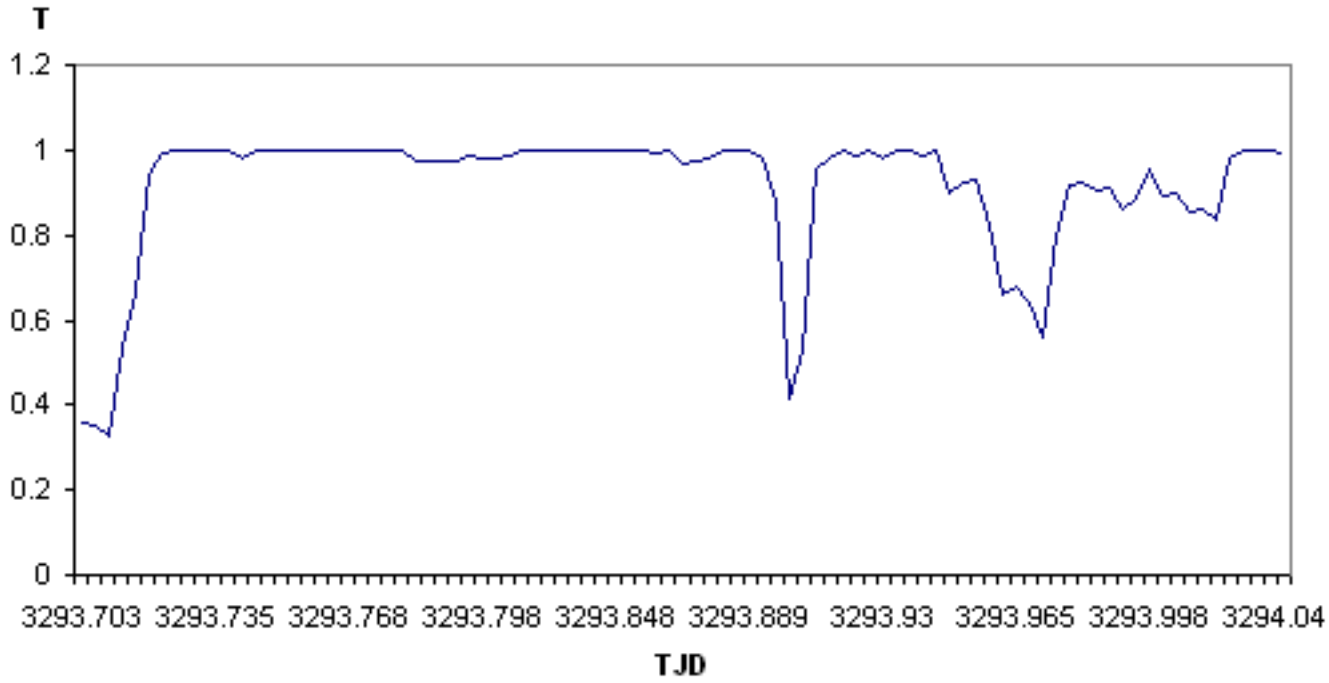

Figure 5.1: Transmission measured at RA=00h 40m, DEC=56.5^o recorded at Mauna Kea on 10/15/2004

with amplitudes larger than a given amount are ignored, the variability of Algol is not expected to have any effect on the results. Additionally, each pixel is assigned with a sky transmission value based on more than one star, further minimizing variability error.

Another problem affecting the reliability of the system is transmission local to the optics that has nothing to do with the real transmission of the sky. The most common example is water droplets on the lens, although even spiders have been noted in this regard. The result, of course, is an artificially lower I_{star}, and therefore an artificially low transmission estimate for part of the sky. Effects like these are sometimes unavoidable, however experience shows that a heated glass dome protecting the lens can be a practical deterrent.

One method of detecting the presence of moisture is by comparing several images. If the measured opacity is always lower at the same region of the sky (in terms of topocentric coordinates), it is likely that the lens is covered with moisture in that region. A simpler approximation of the problem is to count the number of stars brighter than a certain threshold as described in Section 5.2. If the percentage of stars detected at a certain region of the sky is less than a pre-set low value while other regions are relatively clear, the region is assumed by the system to be covered with moisture.

5.5 Implementation Using the *Night Sky Live* Network

The technique described in sections 5.2 - 8.2.3 has been implemented using the infrastructure of the Night Sky Live network (Nemiroff & Rafert 1999; Nemiroff et al. 2006), which consists of 10 nodes called *CONCAM* located at some of the world's premier observatories. Currently CONCAMs operate at the following observatories: Haleakala (Hawaii), Mauna Kea (Hawaii), Mt. Wilson (California), Kitt Peak (Arizona), Rosemary Hill (Florida), Cerro Pachon (Chile), Canary Islands (Spain), SALT (South Africa), Wise (Israel), and Siding Spring (Australia). Each node incorporates an SBIG ST-8 (CONCAM2s) or ST-1001E (CONCAM3s) CCD camera, a Nikon FC-E8 or SIGMA F4-EX 8mm fish-eye lens and an industrial PC. Each *CONCAM* takes one 1024×1024 180-second exposure all-sky FITS (Wells Greisen & Harten) image every 236 seconds. The FITS files are then transmitted to the main server where they are copied to the public domain and can be accessed at *http://nightskylive.net*. FITS frames are stored in the main server for two months, after which they are archived on DVDs and removed from the server, but are still available upon specific request. Each *CONCAM* node of the *Night Sky Live* network provides one all-sky image every 3 minutes and 56 seconds.

The pipeline of all-sky images provides the data required to generate all-sky relative opacity maps using the technique described above. To date, however, relative opacity maps are only generated for the CONCAM3 stations. Figure 5.4 is an example of an all-sky opacity map generated by the Cerro Pachon CONCAM3.

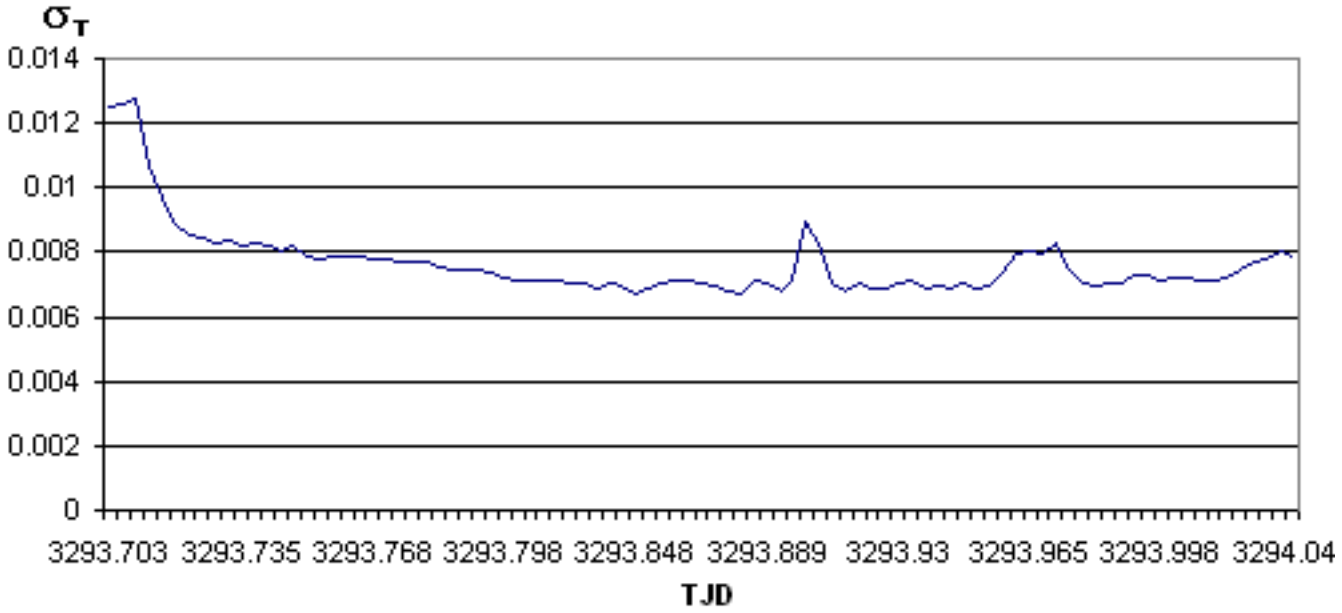

Figure 5.2: σ_T of RA=00h 40m, DEC=56.5^o recorded at Mauna Kea on 10/15/2004

In an all-sky image taken in a clear night, CONCAM3s record approximately 1200 stars 20σ brighter than their background and at altitude greater than 12^o above the horizon. The side of each pixel in a *CONCAM* all-sky image is approximately $10'$, giving that each pixel covers ~100 minutes^2. Sub-pixel opacity estimation is currently not provided by the system.

In order to visualize the sky opacity map, the pixel is colored in blue such that a stronger blue color represents higher opacity. The maps are generated by changing the B component of each RGB triple such that $B = B_0T + 255(1-T)$, where B_0 is the original B value of the pixel and T is the sky transmission calculated using Equation 8.3. A scale added to the top left part of the transmission map presents the colors of several levels of opacity such that $\Delta M = -2.5 \log_{10} T$.

In order to test the effectiveness of the data provided by the Night Sky Live implementation of the proposed method, we compared them to data provided by the Mercator 1.2M telescope and its P7-2000 photometer (Raskin et al. 2004) installed at the Roque De Los Muchachos observatory on La Palma, where a CONCAM system is also installed. The Mercator telescope provides between 10 to 30 single-point atmospheric extinction measurements per night, and the data (including archived data from previous nights) are available on-line.

Figure 5.5 shows a comparison between the K(V) atmospheric extinction measurements provided by Mercator and the transmission computed by the CONCAM implementation of the proposed method in the period between 3/8/2004 and 3/14/2004. Although the data provided by the systems do not provide an exact match, the comparison shows that the two systems are correlated.

5.6 Conclusions

In this paper, a simple cloud monitoring system that provides all-sky relative opacity maps was presented. The maps are generated by comparing all-sky images taken in real-time with canonical images taken on clear nights at the same sidereal time. Although the maps are approximation of the true relative transmission, they have been found useful by observers, and provide data that have never been provided by other available cloud monitoring systems. The system is deployed at five premier observatories, which are Cerro Pachon (Chile), Mauna Kea, Haleakala, SALT (South Africa) and Canary Islands.

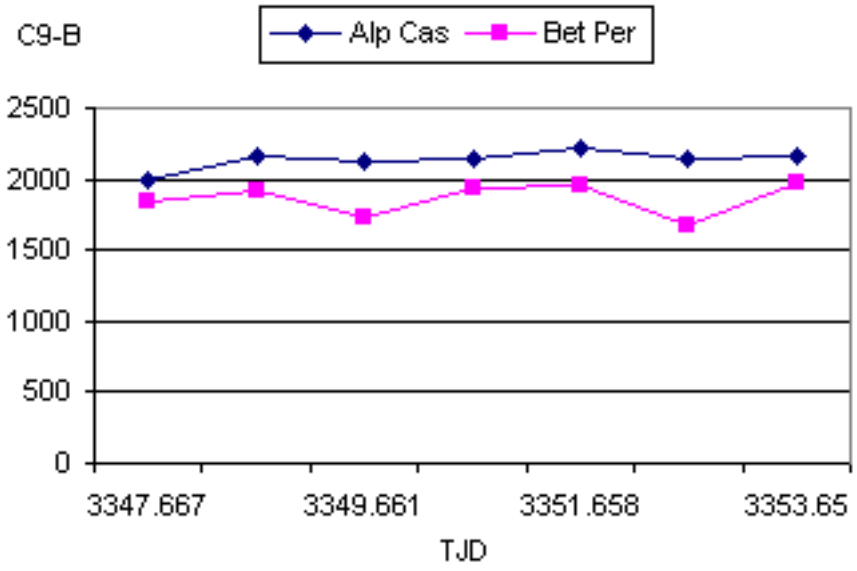

Figure 5.3: The average of the 9 brightest pixels of Alp Cas and Bet Per subtracted by the background, as measured by *CONCAM* at KPNO between 12/8/2004 and 12/15/2004

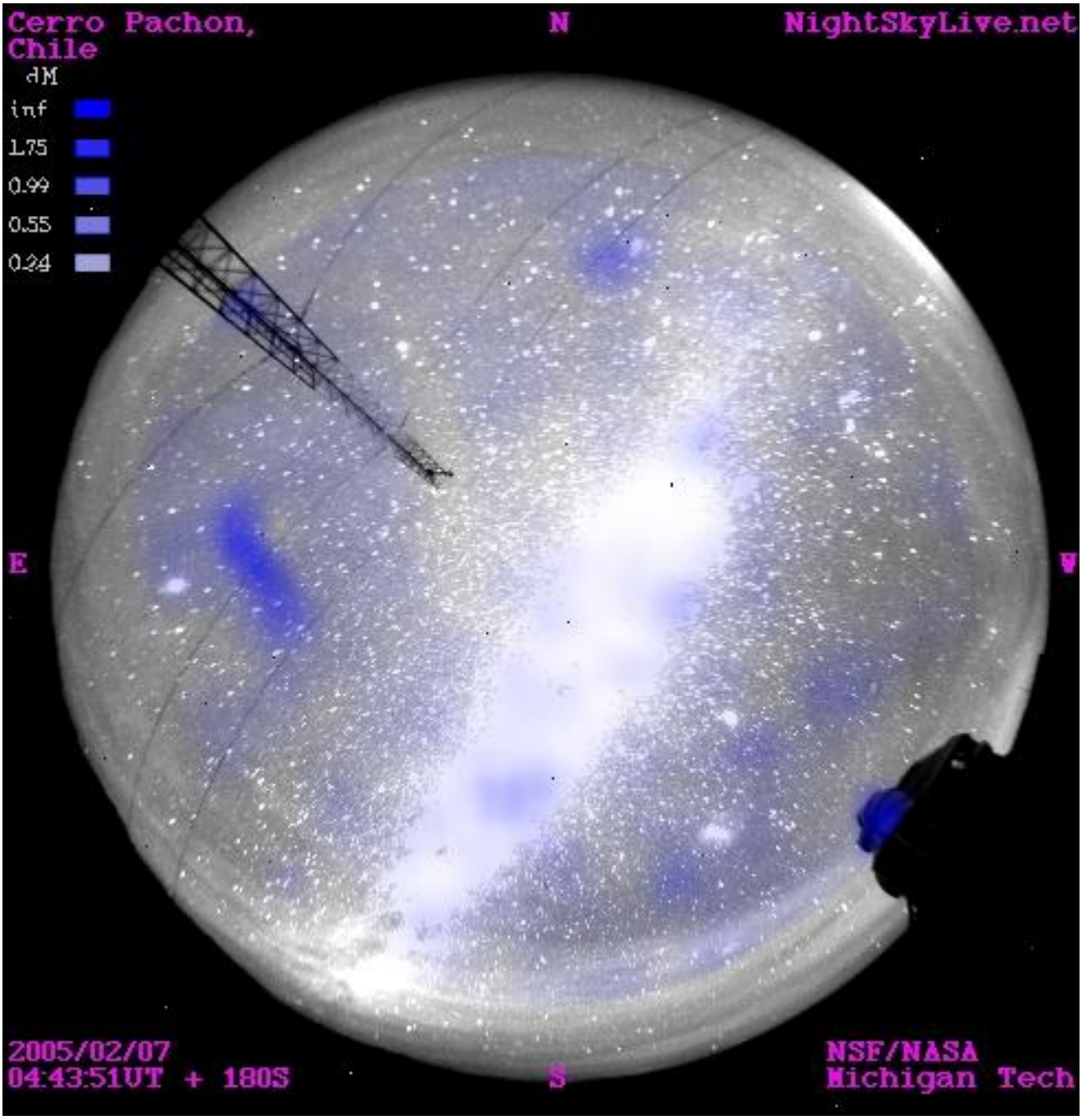

Figure 5.4: All-sky opacity map generated in Cerro Pachon, Chile

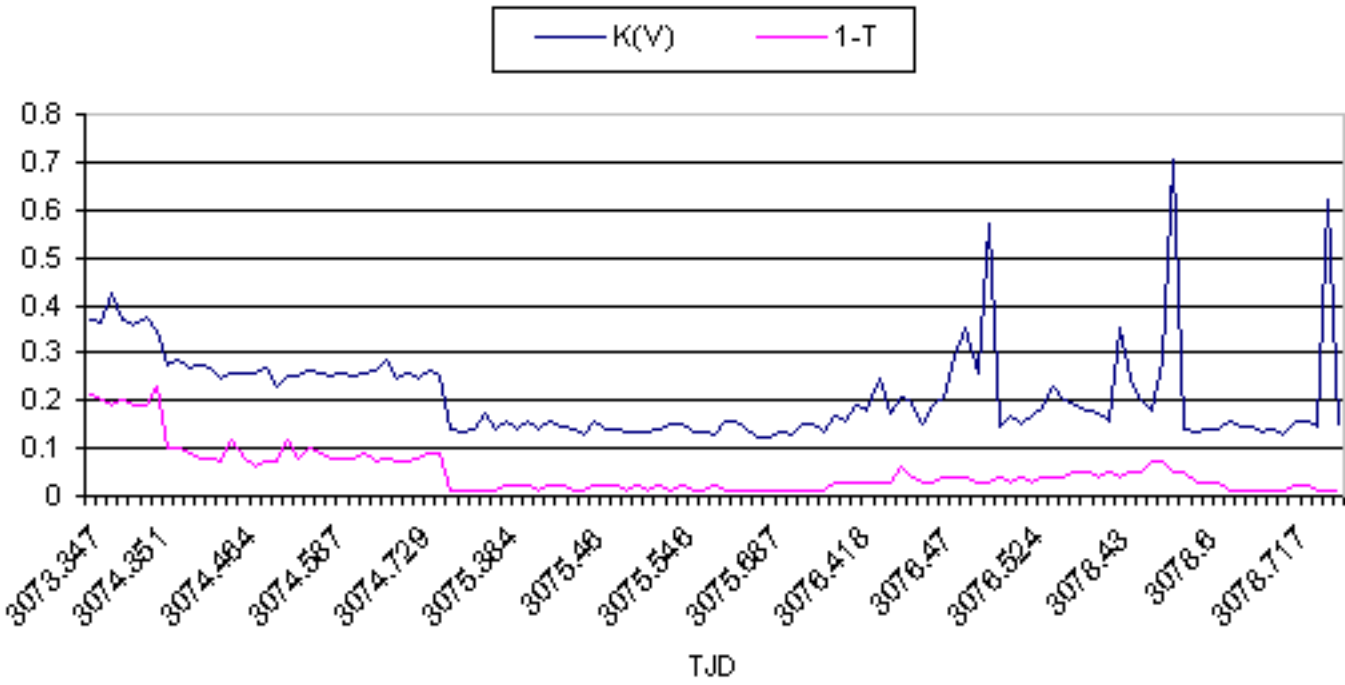

Figure 5.5: K(v) provided by Mercator and 1-T provided by CONCAM measured in La Palma in the period between 3/8/2004 and 3/14/2004

Bibliography

Ananthasubramanian, P. G., Yamamoto, S., Prabhu, T. P., & Angchuk, D., 2004, Bulletin of the Astronomical Society of India, 32, 99

Chamberlin, R. A., 2003, IAU Meeting 25.

Hogg, D. W., Finkbeiner, D. P., Schlegel, D. J., & Gunn, J. E., 2001, AJ, 122, 2129

Hull, C. L., Limmongkol, S., & Siegmund, W. A., 1994, Proc. SPIE, 2199, 852

Muzzin, V., Shamir, L., & Nemiroff, R. J., 2005, AAS Meeting 205, #152.01

Nemiroff, R. J., & Rafert, J. B., 1999, PASP, 111, 886

Nemiroff, R. J., Shamir, L., Muzzin, V., Merlo, M., & Tilvi, S. V., 2005, AAS Meeting 205, 54.12

Nemiroff, R. J., Shamir, L., Pereira, W. E., Rafert, J. B., Ftaclas, C., & Cook, D., 2006, in preparation

Pereira, W. E., Muzzin, V., Merlo, M., Shamir, L., & Nemiroff, R. J., 2005, AAS Meeting 205 #48.13

Pereira, W. E., Nemiroff, R.J., Rafert, J. B., Ftaclas, C. & Perez-Ramirez, D., 2000, AAS Meeting 197, 115.10

Raskin, G., Burki, G., Burnet, M., Davignon, G., Dubosson, R., Ischi, E., George, M., Grenon, M., Maire, C., Van Winckel, H., Waelkens, C., & Weber, L., 2004, UV and Gamma-Ray Space Telescope Systems, Proc. of the SPIE, 5492, 830

Rutten, R., 2003, ING Newsl., 7, 24

Shamir, L., & Nemiroff, R. J., 2005, PASA, 22, 111

Shamir, L., 2005, AN, 326, 428

Wells, D. C., Greisen, E. W., & Harten, R. H., 1981, A&AS, 44, 363

Chapter 6

Analysis of Meteor Trails Using the *Night Sky Live* Network of Panoramic CCD Cameras

Abstract: The analysis of meteor trails using a publicly accessible array of two panoramic all-sky CCD cameras is presented. Located at Mauna Kea and Haleakala, Hawaii, the array captures meteor showers as well as sporadic meteors, and provides information regarding meteor atmospheric trajectories and light curves. The system also allows light curve analysis using the FITS data and the absolute distances between specific points of interest in the trail. Data collected by the system are available to the public in real-time, and can be accessed using a simple internet browser.

*This chapter was published in: Shamir, L., 2005, *WGN Journal of the International Meteor Organization*, Vol. 33(3), pp. 75–81.

6.1 Introduction

Observing meteors is commonly done using arrays of cameras (Ceplecha 1986; Ceplecha et al. 1999). These arrays can consist of narrow-field optics allowing imaging of faint meteors, or wide-field optics that capture only the brighter meteors, but covers a larger portion of the sky.

By using two (or more) cameras located far enough from each other, a 3D analysis of the meteor trail is obtained by using parallax (Molau 1995; Koten et al. 2004; Spurny et al. 2004). However, while meteor research is a field of scientific interest, deploying and operating on-going arrays of cameras is a logistically demanding task. In this paper, a technique of using the *Night Sky Live* (Nemiroff & Rafert 1999) network for the purpose of meteor science is presented. The twin all-sky CCD cameras constantly operated at Mauna Kea and Haleakala observatories are used for obtaining altitude, absolute length and light curves of meteors. Archived and real-time data collected by the Night Sky Live network are available to the public.

In Section 6.2 we briefly present the *Night Sky Live* network, in Section 6.3 the analysis of meteor trails is presented, in Section 6.4 the estimated error is discusses and Section 6.4 presents the analysis of meteor light curves.

6.2 The Night Sky Live Network

The Night Sky Live (Nemiroff & Rafert 1999) consists of 10 nodes called *CONCAM* located at some of the world's premier observatories. Each node incorporates an SBIG ST-8 or ST-1001E CCD camera, a Nikon FC-E8 or SIGMA F4-EX 8mm fish-eye lens and an industrial PC. Each *CONCAM* takes one 1024×1024 180-second exposure all-sky image every 3 minutes and 56 seconds. The FITS files are then transmitted to the main server where they are copied to the public domain and can be accessed at http://nightskylive.net. The Night Sky Live network provides features such as bright star monitoring (Shamir & Nemiroff 2004) and all-sky opacity maps (Nemiroff & Shamir 2003). FITS frames are stored in the main server for two months, after which they are archived on DVDs and removed from the server, but are still available upon specific request.

Currently there are 10 *CONCAM* nodes located in Mauna Kea and Haleakala - Hawaii, Cerro Pachon - Chile, Kitt Peak - Arizona, Mt. Wilson - California, Rosemary Hill - Florida, Siding Spring - Australia, Wise Obs. - Israel,

Canary Islands and South Africa. Among the operating *CONCAM*s, the ones discussed in this paper are the twin *CONCAM*s located at Mauna Kea and Haleakala, which are close enough to capture the same meteors.

6.3 Analysis of Meteor Trails

Mauna Kea CONCAM is located at Lon=−155 deg 28 min 8.7 sec, Lat=19 deg 49 min 21.1 sec, and Haleakala CONCAM is located at Lon=−156 deg 15 min 21.2 sec, Lat=20 deg 42 min 25.9 sec. The distance between the two stations is 128.14 km. In order to analyze the trajectory of a meteor trail, the celestial coordinates of the start and end of the meteor trail are required. These coordinates are obtained by manually finding the image (X, Y) coordinates of the start and end of the light curve in both images, and then converting the image coordinates into (Alt, Az) topocentric celestial coordinates using a fuzzy logic-based transformation formula (Shamir & Nemiroff 2005a). The source code of the computer program that implements the transformation formula can be downloaded at http://nightskylive.net/wolf/source/, and the algorithm is thoroughly discussed in (Shamir & Nemiroff 2005a).

Figures 6.1 and 6.2 show the same meteor recorded by Mauna Kea and Haleakala CONCAMs at 2004 October 18, $13^h36^m07^s$ UT.

The image coordinates of the meteor are given in Table 6.1.

	Mauna Kea	Haleakala
Start	(339,670)	(219,483)
End	(329,695)	(203,483)

Table 6.1: Image coordinates (in pixels) of the start and end of the meteor trail in the images taken at Mauna Kea and Haleakala

The topocentric celestial coordinates calculated by applying the fuzzy logic-based transformation formula are given in Table 6.2.

	Mauna Kea	Haleakala
Start	Az=44.30deg	Az=99.84deg
	Alt=44.24deg	Alt=32.70deg
End	Az=41.84deg	Az=99.66deg
	Alt=39.72deg	Alt=28.47deg

Table 6.2: Topocentric coordinates of the start and end of the meteor trail

The azimuth of Haleakala from Mauna Kea is 320.35deg. Given the distance from Mauna Kea to Haleakala and the azimuths of the meteor from both stations, the horizontal distance of the end of the meteor trail from Mauna Kea can be easily obtained by calculating the side y in Figure 6.3.

The horizontal distance of the end of the meteor trail from Mauna Kea can be calculated using Equation 6.1.

$$y = \frac{x \cdot \sin(180 - \delta - \alpha)}{\sin(\alpha - \beta)} \tag{6.1}$$

Given that $x = 128.14$ km, $\alpha = 99.66$ deg, $\beta = 41.84$ deg, and $\delta = 39.65$ deg, y can be calculated to be 98.7 km. The horizontal distance z from Haleakala to the end of the meteor trail can be calculated using Equation 6.2

$$z = y \cdot \frac{\sin(\delta + \beta)}{\sin(180 - \delta - \alpha)} = 149.72km \tag{6.2}$$

Using the angular altitude of the end of the meteor trail measured in Mauna Kea (which is 39.72deg), the altitude of the end of the meteor trail is $H_{end} = \tan(39.72)y = 81.9$ km above the Mauna Kea.

Using the altitude and the horizontal distance, the absolute distance of the end of the meteor trail from Mauna Kea is $D_{mk} = \sqrt{H_{end}^2 + y^2} = 128.25$ km

Similarly, the horizontal distance of the start of the meteor trail from Mauna Kea can be obtained using Equation 6.3.

$$v = \frac{x \cdot \sin(180 - \delta - \gamma)}{\sin(\gamma - \phi)} \tag{6.3}$$

Where γ is the azimuth of the start of the meteor trail from Haleakala and ϕ is the azimuth of the start of the meteor trail from Mauna Kea. Using the values from Table 6.2, $\gamma = 99.84$ deg and $\phi = 44.30$ deg, gives $v = 100.9$ km.

The horizontal distance from Haleakala to the start of the meteor trail can be calculated using Equation 6.4.

$$v \cdot \frac{\sin(\delta + \phi)}{\sin(180 - \delta - \gamma)} = 154.5km \tag{6.4}$$

Using the angular altitude of the start of the meteor trail measured in Mauna Kea (44.24deg), the altitude of the start of the meteor trail is $H_{start} = \tan(44.24)v = 98.2$ km above the Mauna Kea.

The horizontal length of the meteor trail can be obtained by calculating the side w using Equation 6.5.

$$w = \sqrt{(\sin(\phi - \beta)y)^2 + (v - \cos(\phi - \beta)y)^2} \tag{6.5}$$

where ϕ is the azimuth of the start of the meteor trail measured in Mauna Kea, y is the horizontal distance of the end of the meteor trail from Mauna Kea and v is the horizontal distance of the *start* of the meteor trail from Mauna Kea. Given that $\phi = 44.30\,\text{deg}$, $\beta = 41.84\,\text{deg}$, $y = 98.7$ km and $v = 100.9$ km, the horizontal length of the meteor trail is 4.7 km.

Given the altitudes of the start and end of the trail and the horizontal length, the absolute length of the trail (assuming linear trajectory) can be calculated simply by $L = \sqrt{(98.2 - 81.9)^2 + 4.7^2} = 16.96$ km.

6.4 Estimated Error

The fuzzy logic-based transformation formula is accurate to a level of 3.2 pixels (Shamir & Nemiroff 2005a). Since each pixel in a NSL frame is approximately 10min, the maximum difference between the computed celestial coordinates and the true celestial coordinates would be $3.2 \cdot 10\,\text{min} = 32\,\text{min}$. Due to sub pixel positioning, an extra error of $0.5 \cdot 10\,\text{min}$ should be added so the total error is $32\,\text{min} + 5\,\text{min} = 37\,\text{min} \simeq 0.617\,\text{deg}$.

The coordinates of the meteor trail with the estimated error are given in Tables 6.3 (in degrees) and 6.4 (in radians).

	Mauna Kea	Haleakala
Start	Az=44.30$\pm\frac{0.617}{\cos(44.24)}$	Az=99.84$\pm\frac{0.617}{\cos(32.70)}$
	Alt=44.24$\pm$0.617	Alt=32.70$\pm$0.617
End	Az=41.84$\pm\frac{0.617}{\cos(39.72)}$	Az=99.66$\pm\frac{0.617}{\cos(28.47)}$
	Alt=39.72$\pm$0.617	Alt=28.47$\pm$0.617

Table 6.3: Topocentric coordinates (in degrees) of the start and end of the meteor trail and their estimated error

	Mauna Kea	Haleakala
Start	Az=0.7732$\pm\frac{0.0108}{\cos(0.7721)}$	Az=1.7425$\pm\frac{0.0108}{\cos(0.5707)}$
	Alt=0.7721$\pm$0.0108	Alt=0.5707$\pm$0.0108
End	Az=0.7302$\pm\frac{0.0108}{\cos(0.6932)}$	Az=1.7394$\pm\frac{0.0108}{\cos(0.4969)}$
	Alt=0.6932$\pm$0.0108	Alt=0.4969$\pm$0.0108

Table 6.4: Topocentric coordinates (in radians) of the start and end of the meteor trail and their estimated error

6.4.1 Estimated Error of the End of the Meteor Trail

The horizontal distance y of the end of the meteor trail from Mauna Kea is calculated based on α and β (the azimuths of the end of the meteor trail measured in Mauna Kea and Haleakala) using Equation 6.1.

Let $y_n = x \cdot \sin(\pi - \delta - \alpha)$ and σ_α be the estimated error of α specified in Table 6.4. The estimated error of y_n is defined by Equation 6.6.

$$\sigma_{yn} = x \cdot \cos(\pi - \delta - \alpha) \cdot \sigma_\alpha = 128.14 \cdot \cos(\pi - 0.692 - 1.7394) \cdot \frac{0.0108}{\cos(0.4969)} \simeq 1.193 \tag{6.6}$$

Let $y_d = \sin(\alpha - \beta)$. The estimated error σ_{yd} of y_d is defined by Equation 6.7.

$$\sigma_{\mathrm{y}d} = \cos(\alpha - \beta) \cdot \sqrt{\sigma_\alpha^2 + \sigma_\beta^2} =$$
$$\cos(1.7394 - 0.7302) \cdot \sqrt{(\frac{0.0108}{\cos(0.4969)})^2 + (\frac{0.0108}{\cos(0.6932)})^2} \simeq 0.0099 \tag{6.7}$$

Using the estimated error provided by Equations 6.6 and 6.7, the estimated error σ_{y} of y is defined by Equation 6.8.

$$\sigma_{\mathrm{y}} = \frac{x \cdot \sin(\pi - \delta - \alpha)}{\sin(\alpha - \beta)} \cdot \sqrt{(\frac{\sigma_{yn}}{y_n})^2 + (\frac{\sigma_{yd}}{y_d})^2} \simeq 1.82 \tag{6.8}$$

The estimated error σ_{y} introduces an error of $\sim$1.84% in y.

Let $t = \tan(\psi)$, where ψ is the altitude of the end of the meteor trail measured in Mauna Kea. The estimated error of t is defined by Equation 6.9.

$$\sigma_{\mathrm{t}} = \sigma_\psi \cdot \cos^{-2}(\psi) = 0.0108 \cdot \cos^{-2}(0.6932) \simeq 0.0183 \tag{6.9}$$

Let $\sigma_{\mathrm{H}e}$ be the estimated error of H_{end}, defined by Equation 6.10.

$$\sigma_{\mathrm{H}e} = \tan(\psi)y \cdot \sqrt{(\frac{\sigma_{\mathrm{y}}}{y})^2 + (\frac{\sigma_{\mathrm{t}}}{\tan(\psi)})^2} \simeq 2.35 \tag{6.10}$$

$\sigma_{\mathrm{H}e}$ introduces an estimated error of $\frac{2.35}{81.9} \simeq 2.87\%$ in H_{end}.

6.4.2 Estimated Error of the Start of the Meteor Trail

Let $v_n = x \cdot \sin(\pi - \delta - \gamma)$. The estimated error $\sigma_{\mathrm{v}n}$ of v_n is defined by Equation 6.11.

$$\sigma_{\mathrm{v}n} = x \cdot \cos(\pi - \delta - \gamma) \cdot \sigma_\gamma =$$
$$128.14 \cdot \cos(\pi - 0.692 - 1.7425) \cdot \frac{0.0108}{\cos(0.5707)} \simeq 1.25 \tag{6.11}$$

Let $v_d = \sin(\gamma - \phi)$. The estimated error of v_d is defined by Equation 6.12.

$$\sigma_{\mathrm{v}d} = \cos(\gamma - \phi) \cdot \sqrt{\sigma_\gamma^2 + \sigma_\phi^2} =$$
$$\cos(1.7425 - 0.7732) \cdot \sqrt{(\frac{0.0108}{\cos(0.5707)})^2 + (\frac{0.0108}{\cos(0.7721)})^2} \simeq 0.0112 \tag{6.12}$$

Using the estimated error provided by Equations 6.11 and 6.12, the estimated error of v is defined by Equation 6.13.

$$\sigma_{\mathrm{v}} = \frac{x \cdot \sin(\pi - \delta - \gamma)}{\sin(\gamma - \phi)} \cdot \sqrt{(\frac{\sigma_{\mathrm{v}n}}{v_n})^2 + (\frac{\sigma_{\mathrm{v}d}}{v_d})^2} \simeq 2.04 \tag{6.13}$$

σ_{v} introduces an error of $\frac{2.04}{100.9} \simeq 2.02\%$ in v.

Let $q = \tan(\xi)$, Where ξ is the altitude of the start of the meteor trail measured in Mauna Kea. The estimated error of q is defined by Equation 6.14.

$$\sigma_{\mathrm{q}} = \sigma_\xi \cdot \cos^{-2}(\xi) = 0.0108 \cdot \cos^{-2}(0.7721) \simeq 0.0210 \tag{6.14}$$

Let $\sigma_{\mathrm{H}s}$ be the estimated error H_{start}, defined by Equation 6.15.

$$\sigma_{\mathrm{H}s} = \tan(\xi)v \cdot \sqrt{(\frac{\sigma_{\mathrm{v}}}{v})^2 + (\frac{\sigma_{\mathrm{q}}}{\tan(\xi)})^2} \simeq 2.90 \tag{6.15}$$

This introduces an estimated error of $\frac{2.90}{98.2} \simeq 2.95\%$ in H_{start}.

The procedure described in Section 6.3 was tested by measuring the altitude of known satellites such as the International Space Station and several Iridium satellites with accuracy of less than 1%.

6.5 Meteor Light Curves

FITS frames provide an effective infrastructure for pixel-by-pixel analysis of meteor light curves (Campbell et al. 2001; Cordell et al. 2004; Brosch & Manulis 2002; Brosch et al. 2004). A 3D plot of the pixel values of the meteor recorded in Figure 6.1 is illustrated in Figure 6.4.

The light curve presented in Figure 6.4 has two peaks of maximal brightness. The first peak is at image coordinates (334,681) and the second is at the end of the trail at (329,695). Using the analysis presented in Section 6.3, the first peak is at altitude of 91.2 km above Mauna Kea and occurred when the meteoroid was 7.29 km away from where it started its luminous trail. The second peak is at altitude of 81.9 km above Mauna Kea and 16.96 km away from where the luminous trail started.

The limiting stellar magnitude of CONCAM is approximately 6.8. Assuming a meteor duration is 0.4 seconds and the length of the trail is 20 pixels on the CCD chip (such as the meteor in the paper), the duration on each pixel would be $\frac{0.4}{20} = 0.02$ seconds, and the limiting magnitude of the meteor would be $6.8 + 2.5 \log \frac{0.02}{180} \simeq -3.08$.

6.6 Conclusions

The twin CONCAM stations located at Mauna Kea and Haleakala provide data that can be used for the purpose of meteor research without the need to set up and operate a dedicated array of cameras. The fish-eye images cover the whole 2π view of the night sky, and the stations are active 24/7 year round. Data recorded in real-time, as well as archived data are copied to the public domain and can be accessed and used easily. The analysis presented in this paper allows obtaining the altitude and total length of meteor trails, which can be used for light curve analysis.

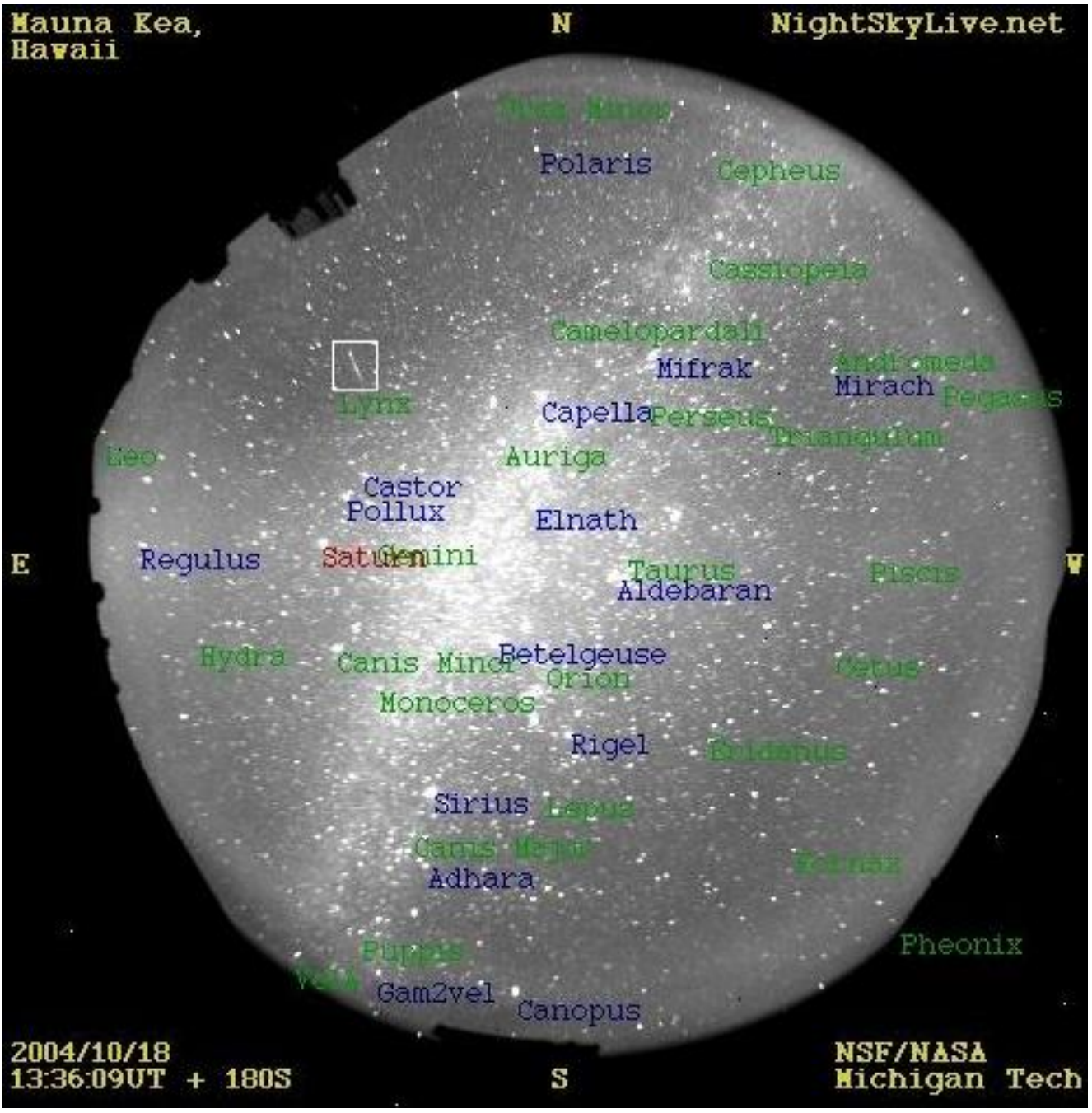

Figure 6.1: A meteor (in the white rectangular frame) recorded at Mauna Kea

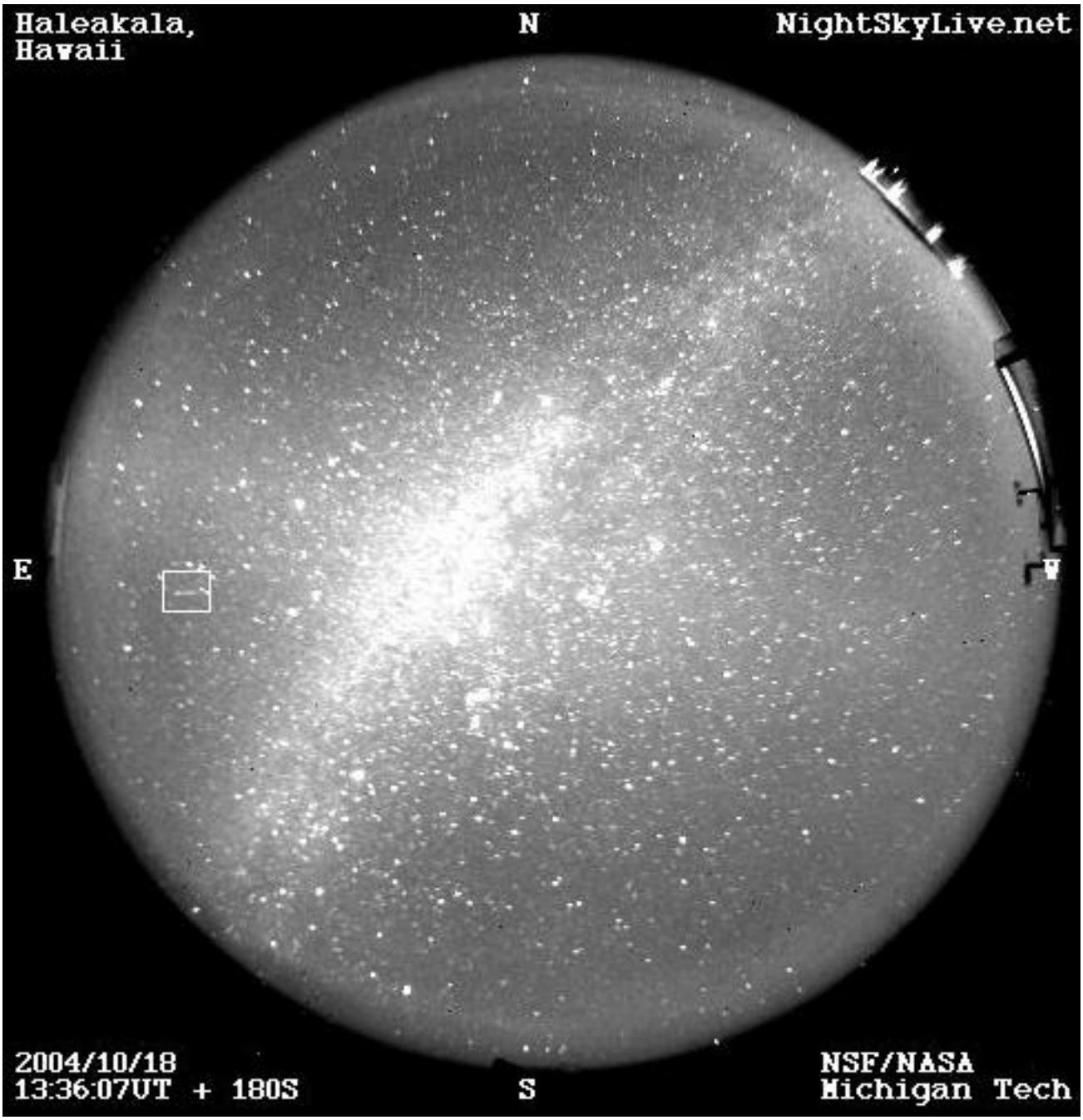

Figure 6.2: A meteor (in the white rectangular frame) recorded at Haleakala

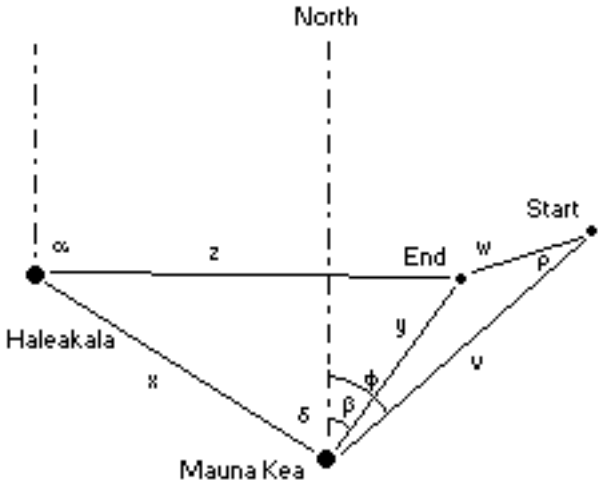

Figure 6.3: Illustration of the meteor trail as seen from above. α is the azimuth of the end of the meteor trail from Haleakala, β is the azimuth of the end of the meteor trail from Mauna Kea, x is the distance from Mauna Kea to Haleakala and δ is the azimuth of Haleakala from Mauna Kea.

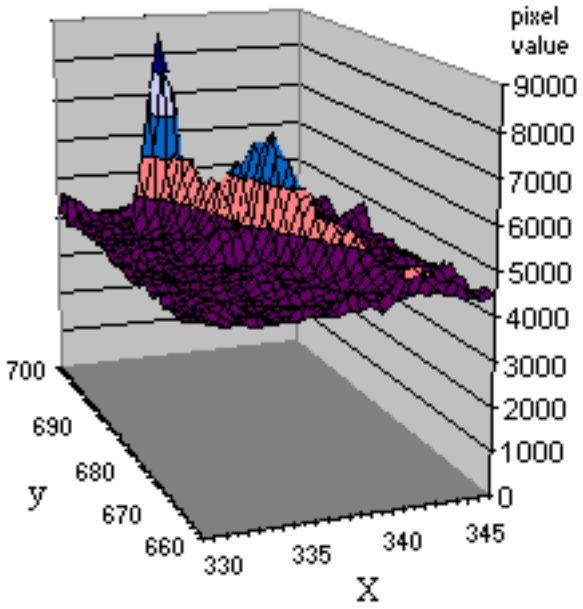

Figure 6.4: Light curve of the meteor recorded in Figure 6.1.

Bibliography

Brosch N. and Manulis I., 2002, Proc. of Asteroids, Comets, Meteors – ACM 2002. Berlin (Germany) 209

Brosch N., Helled R., Poolishook D., Almoznino E., David N., 2004, MNRAS, 355, 111

Campbell M., Theijsmeijer C., Jones J., Hawkes R. L., Brown P., 2001, Proc. of the Meteoroids 2001 Conf., Kiruna (Sweden), 197 – 201

Ceplecha Z., 1986, Asteroids, comets, meteors II, Proc. of the Intl. Meeting, Uppsala (Sweden)

Ceplecha Z., Spalding R. E., Jacobs C., ReVelle, D. O., Tagliaferri, E., Brown, P., 1999, Meteroids 1998, Proc. Intl. Conf., Tatranska Lomnica (Slovakia), 37

Cordell D., Tilvi S. V. and Nemiroff, R. J. (2004) AAS Meeting 205, 56.08

Koten P., Borovicka J., Spurn P., Betlem H. and Evans S., 2004, A&A, 428, 683

Molau S., 1995, Proc. Intel. Meteor Conf., Belogradchik (Bulgaria).

Nemiroff R. J. and Shamir L., 2003, AAS Meeting 203, 38.14

Nemiroff R. J. and Rafert, J. B., 1999, PASP, 111, 886

Nemiroff R. J., Shamir L., Muzzin V., Merlo M. and Tilvi S. V., 2004, AAS Meeting 205, 54.12

Shamir L. and Nemiroff R. J., 2004, AAS Meeting 76.07

Shamir L. and Nemiroff R. J., 2005, PASA, 22, 111

Spurny P., Olech A. and Kedzierski P., 2004, JIMO, 32, 48

Chapter 7

Transient Detection Using Panoramic All-sky Cameras

Abstract: A method of using on-going panoramic all-sky monitors for the purpose of bright optical transient detection is described. The detection is performed by comparing all-sky images to canonical images taken at the same sidereal time. The described approach has been implemented and tested using the infrastructure of the *Night Sky Live* network of panoramic all-sky cameras. Preliminary results of detected bright flashes are presented in the paper.

*This chapter was submitted to *Astrophysics & Space Science*

7.1 Introduction

Real-time detection of bright optical transients has an important role in modern astronomy. In order to search for astronomical transients, many robotic telescopes have been built and operated around the globe (Cern et al. 2003; Vestrand et al. 2003; Akerlof et al. 2003; Steele et al. 2004; Lipunov 2004). However, since traditional narrow-angle telescopes cover only a very small portion of the sky at a given time, one can reasonably assume that some bright short-timescale transients may not be recorded by the available narrow-angle sky surveys (Nemiroff & Shamir 2003).

One approach of searching for bright optical transients is by deploying and operating continuous panoramic all-sky cameras. This policy guarantees that any optical transient appearing on a clear night is recorded, given that the transient is brighter than the limiting magnitude of the all-sky camera.

The ability of all-sky optics to observe faint sources is not comparable to even simple narrow-angle telescopes. However, since brighter optical transients are sometimes considered more interesting to science (Lindley 1987; Menzies et al. 1987), the approach of using fish-eye lenses that can record bright transients by covering almost the entire sky can be weighed against narrow-field robotic telescope that are capable of recording much fainter sources, but cover only a narrow field of view.

In Section 7.2 we describe the construction of the database of canonical images, in Section 7.3 the transient detection using all-sky images is described and in Section 7.4 the implementation and preliminary results are presented.

7.2 Canonical Image Database

A first step all-sky transient detection is to create a comparison image from which any image can be compared. This image can be composed of images taken at the same sidereal time as the given frame, and will be called the "canonical image". One can think of the "canonical image" as a hypothetical image taken when the sky is completely clear. In reality, the canonical image is better created from several relatively clear images taken at the same sidereal time. To proceed, the system first needs to acquire a sufficient database of relatively clear images.

One approach of obtaining this database is to manually select images taken at clear nights. However, observing images by eye is a time-consuming task, and the required human resources are not always available. Moreover, since thin cirrus clouds are hard to detect, a human observer might consider some images as *clear* while in fact parts of the sky are covered with light clouds.

Another approach, which we have found more appropriate, is to use an algorithm that automatically searches for images taken at clear nights. The algorithm is based on applying a star recognition algorithm that associates the point spread functions in the image with star catalog entries (Shamir & Nemiroff 2005a). After the point spread functions are associated with known stars, the algorithm checks if the stars that are brighter than a pre-set visual

magnitude are detected at a pre-set statistical confidence level. If the PSFs of a sufficient percentage of expected stars are found in the image, the image is classified as *clear*, and added to the database.

The percentage of stars visible at a certain frame is dependent not only on the sky clarity, but also on the darkness of the sky. For instance, faint stars near the galactic plane will not be detected as easily as stars far from the Milky Way. Therefore, the pre-set threshold of visual magnitude should be chosen such that stars near the center of the galactic plane brighter than the threshold can be detected. Since this problem becomes more substantial when the moon is up, canonical images are added to the database only when the moon is down.

When an image is added to the database, it is averaged with all other images taken at the same sidereal time that are already present. Averaging the images gives a better signal to noise ratio than using a single exposure. The large pixel size and the absence of moving parts in the all-sky monitoring system simplify the process of image co-adding.

7.3 Transient Detection Using All-sky Images

The detection of transients in all-sky images requires several logical steps. The first is the building of a canonical image taken from previous images taken at the same sidereal time, as discussed last section. Another step is the rejection of pixels dominated by cosmic-ray generated counts. Rejecting cosmic rays by comparing two images of the same field (Shaw & Horn 1992; Fixsen et al. 2000) cannot be used for this purpose since this method inherently rejects flashes appearing in one image but not appearing in the other. Since the use of this approach is inappropriate for the described system, rejections should make use of the non-point source nature of cosmic ray splashes. We chose to use a fuzzy logic-based algorithm for cosmic ray hit rejection from single images (Shamir & Nemiroff 2005b). This algorithm is reasonably accurate and provides low computational complexity allowing it to process images in a relatively short time.

Next, bright planets are also rejected using a star recognition algorithm designed to find astronomical objects in wide angle frames (Shamir & Nemiroff 2005a). In the same fashion, high-amplitude variable stars can also be rejected from the image.

After cosmic ray hits and bright planets are rejected, the system searches for PSFs that are a given σ brighter than the local background. Each PSF is then compared with the canonical frame. If the PSF does not appear in the canonical image, it is assumed to be a transient.

Due to the relatively high density of artificial objects in orbit around the earth, one can expect that most detected flashes would result from the presence of artificial objects (Schaefer et al. 1987a; Varady & Hudec 1992). In fact, some flashes that were suspected to be true astronomical transients (Halliday et al. 1987) appeared later to be nothing but background flashes (Schaefer et al. 1987b). Another source of background flashes is bright meteors and fireballs that may also be recorded by all-sky cameras (Shamir 2005b).

The more interesting flashes may be those that seem to rotate with the sky, which may be an indication of a true astronomical source. One way to find such flashes is by simply observing the list flashes recorded on a specific night. A flash appearing in more than one frame at the same geocentric coordinates can be a good candidate to be regarded as a true astronomical transient.

Some bright flares may be too short to appear in more than one image. One approach to detect transients that appear in just one frame is by comparing images taken by two all-sky cameras located far enough from each other, but record the same area of the sky where the transient was detected. If the flash was detected in two images taken at the same time by two cameras, the flash may be a true astronomical source.

All-sky cameras are usually passive and do not track the sky. Therefore, stars may "trail" in the image, given that the duration of the exposure is long enough to let the star move more than (at least) one pixel on the CCD chip. This seemingly disadvantage can be used here to find flashes that rotate with the sky. If the PSF of the flash seems to trail to the same direction of nearby stars, this may be an indication of a true astronomical transient. Flashes that their PSF does not trail can be either very short flashes or flashes resulting from geosynchronous satellites (Shamir & Nemiroff 2005d).

7.4 Implementation and Preliminary Results

The described method has been implemented using the infrastructure of the global *Night Sky Live* network (Nemiroff & Rafert 1999). The Night Sky Live network consists of 11 nodes called *CONCAM* located at some of the world's premier observatories. Each node incorporates an SBIG ST-8 or ST-1001E CCD camera, a Nikon FC-E8 or SIGMA F4-EX 8mm fish-eye lens and an industrial PC. Each *CONCAM* takes one 1024×1024 180-second exposure all-sky image every 236 seconds. The FITS files are then transmitted to the main server where they are copied to the public domain and can be accessed at http://nightskylive.net. The Night Sky Live network provides features such as bright star monitoring (Shamir & Nemiroff 2004) and all-sky opacity maps (Shamir & Nemiroff 2005c). FITS frames are stored in the main server for two months, after which they are archived on DVDs and removed from the server, but are still available upon specific request. The limiting stellar magnitude is dependent on the hardware being used and can get to 6.8 near the image center.

The described system is currently being tested using the CONCAM camera located in Roque de los Muchachos observatory in La Palma. Between June 24th and July 24th 2005, the system recorded an average of ∼8 bright optical

flashes per night. Most flashes were recorded only in one image. Some flashes appeared in more than one image, but did not rotate with the sky and therefore assumed to be glints of geosynchrnous satellites (Shamir & Nemiroff 2005d).

The limiting stellar magnitude of the CONCAM camera located in La Palma is ∼5.9. The limiting magnitude of a flash is dependent on its duration. For instance, if the duration of the flash is 2 seconds, its limiting magnitude would be $5.9 + 2.5 \log \frac{2}{180} \simeq 1.01$. The limiting magnitude as a function of the duration is described in Figure 7.1.

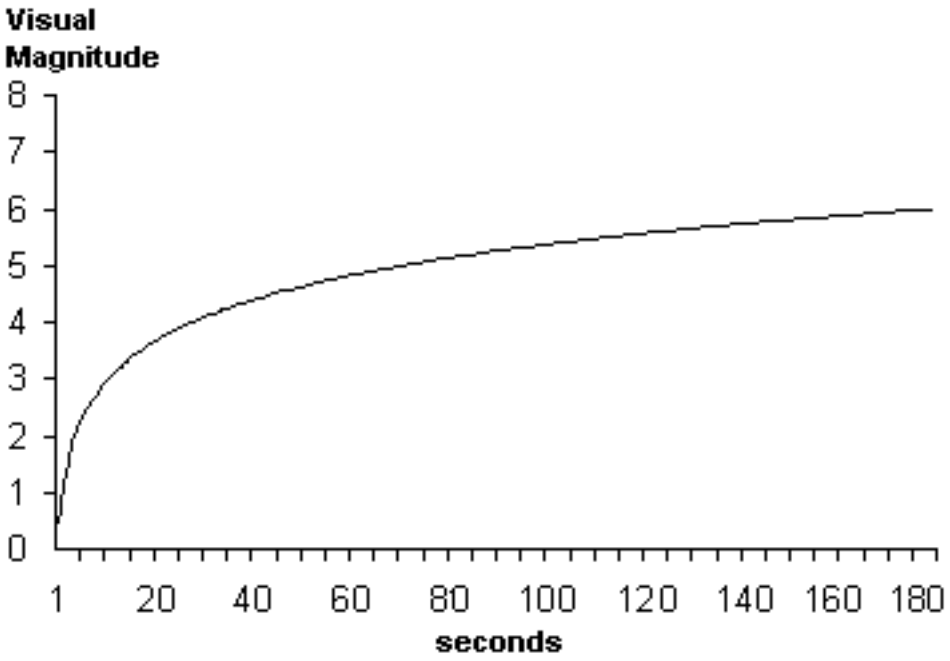

Figure 7.1: Limiting magnitude as a function of the exposure duration

Figure 7.2 shows a flash recorded at RA=∼18.36h, DEC=19.65^o by the CONCAM camera operating in La Palma on the night of July 11th 2005 at 3:25:03 UT. The flash was recorded in just one image. From the non-trailing nature of the point spread function, one can deduce that the flash was either of a very short timescale, or did not rotate with the sky (meaning it was probably a glint from a geosynchronous satellite).

Figures 7.3 and 7.4 show a flash that may be rotating with the sky, and therefore may be a true astronomical source. The flash was recorded in two consecutive images taken on the night of July 18th 2005 at 04:32:30 UT and 04:44:06 UT. The flash appeared in the constellation Aries around RA=∼2.39h, DEC=∼ 25.35^o, and the point spread function seems to trail in a fashion similar to the other neighboring stars recorded in the same frame. The bright star at the bottom of Figure 7.3 is Alpha Arietis and the star at the top left corner is 41 Arietis. The intensity of the flash is comparable to the intensity of Kap Arietis (∼5th magnitude). Unfortunately, other CONCAM cameras covering the same portion of the sky were not active due to technical problems so the flash could not be cross-checked with other cameras covering the same area of the sky.

7.5 Conclusion

In this paper we showed that all-sky cameras can be used for the purpose of bright optical transient detection. The full coverage of the sky provided by all-sky cameras can be weighed against the low limiting magnitude, and can be effective for the purpose of automatic detection of very bright optical transient. Due to artificial luminous objects in orbit around the earth, short-timescale transient detection requires an array of all-sky cameras sharing the same sky so that possible discoveries can be cross-checked.

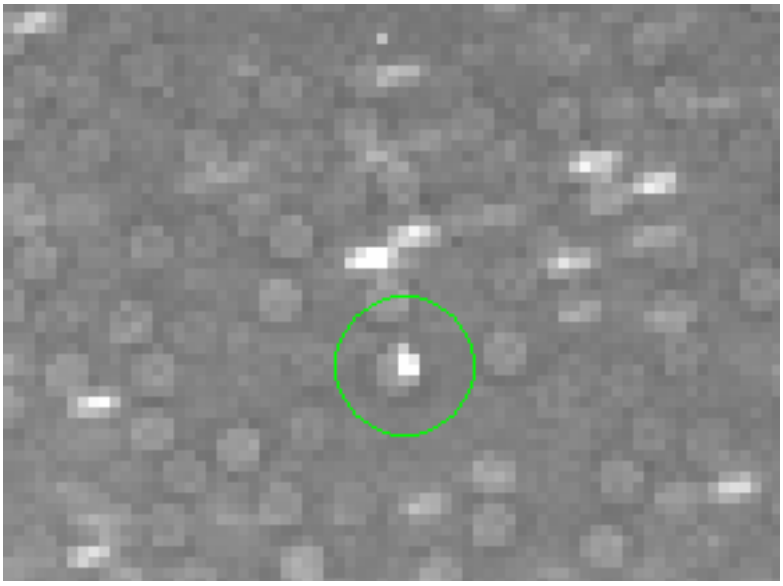

Figure 7.2: A flash recorded by CONCAM all-sky camera located in La Palma on July 11th 2005 at 3:25:03 UT

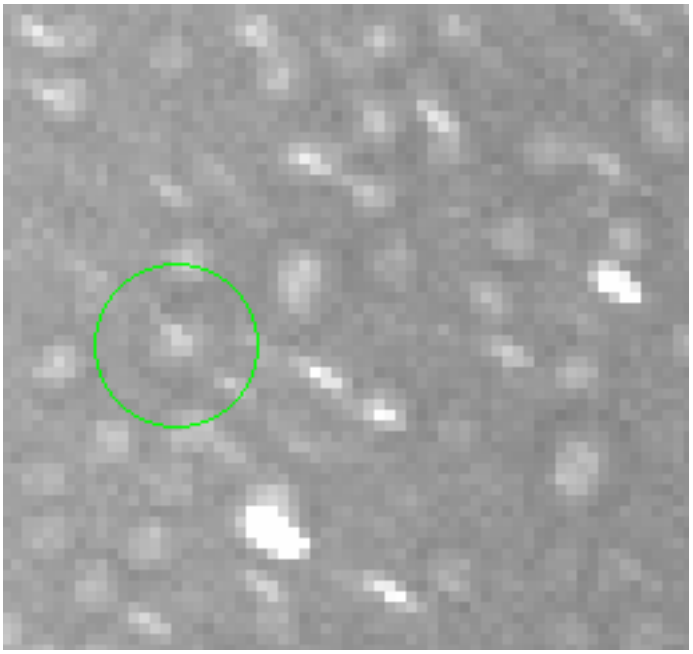

Figure 7.3: A flash recorded by CONCAM all-sky camera located in La Palma on the night of July 18th 2005 at 4:32:30 UT

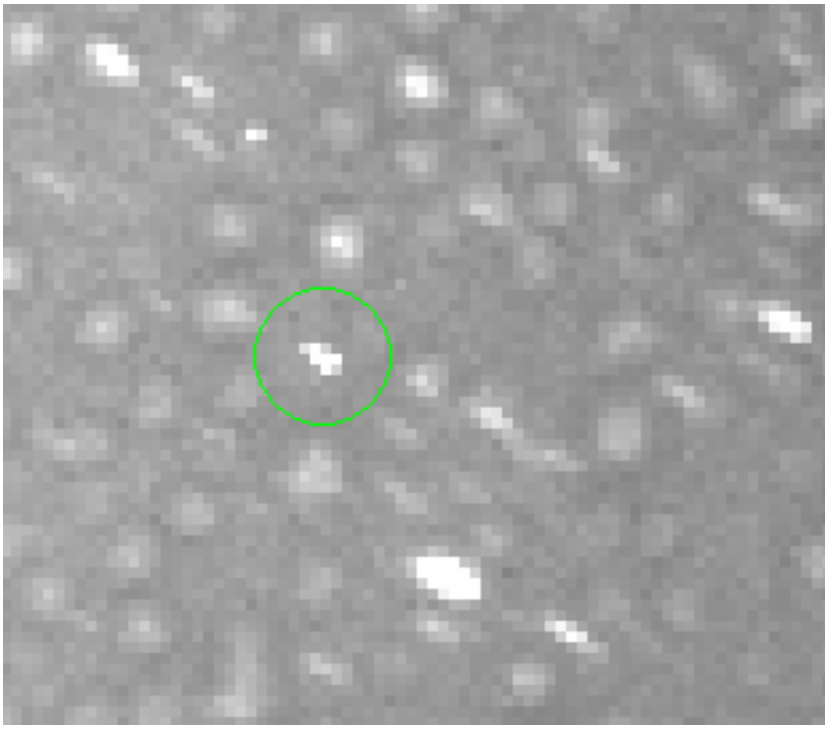

Figure 7.4: A flash recorded by CONCAM all-sky camera located in La Palma on the night of July 18th 2005 at 4:44:06 UT

Bibliography

Akerlof, C. et al., 2000, ApJ, 532, L25

Akerlof, C. et al., 2003, PASP, 115, 132

C. Cern et.al, 2003, Rev. Mex. Astro. y Astrof., Ser. Conf. 16, 77

Fixsen, D. J., Offenberg, J. D., Hanisch, R. J., Mather, J. C., Nieto-Santisteban, M. A., Sengupta, R., Stockman, H. S., 2000, PASP 112, 1350

Halliday,I., Feldman,P.A., Blackwell,A.T., 1987, ApJ, 320, 153L

Lindley D., 1987, Nat, 327, 90

Lipunov V. M., 2004, AN, 325, 580

Menzies J. W. et al., 1987, MNRAS, 227, 39

Nemiroff, R. J. & Rafert, J. B., 1999, PASP, 111, 886

Nemiroff, 2003, AJ, 125, 2740

Schaefer,B.E., Pedersen,H., Gouiffes,C., Poulsen,J.M., Pizzichini,G., 1987a, A&AS, 174, 338

Schaefer et al., 1987b, ApJ, 320, 398

Shamir L. & Nemiroff R. J., 2004, AAS Meeting 204, 76.07

Shamir, L., & Nemiroff, R. J., 2005a, PASA, 22, 111

Shamir, L., & Nemiroff, R. J., 2005b, PASP, accepted

Shamir, L., & Nemiroff, R. J., 2005c, AAS Meeting 207, 15.08

Shamir, L., 2005a, AN, 326, 428

Shamir L., 2005b, JIMO, 33, 75

Shaw, R. A., Horne, K., 1992, ASP Conf. 25, Astronomical Data Analysis, Software and Systems

Steele, I.A et al., 2004, Proc. SPIE, 5489, 679

Varady, M., Hudec, R., 1992, ApJ, 261, 365

Vestrand, W. T., et al., 2004, Proc. AIP, 662, 547

Chapter 8

Summary and Conclusions

Abstract: We describe the software requirement and design specifications for all-sky panoramic astronomical pipelines. The described software aims to meet the specific needs of super-wide angle optics, and includes cosmic-ray hit rejection, image compression, star recognition, sky opacity analysis, transient detection and a web server allowing access to real-time and archived data. The presented software is being regularly used for the pipeline processing of 11 all-sky cameras located in some of the world's premier observatories. We encourage all-sky camera operators to use our software and/or our hosting services and become part of the global *Night Sky Live* network.

*This chapter was published in: Shamir, L., Nemiroff, R. J., Torrey D. O., Pereira, W. E., 2006, Software Design for Panoramic Astronomical Pipeline Processing, *Monthly Notices of the Royal Astronomical Society*, Vol. 366(2), p. 353–357.

8.1 Introduction

In the past several years, all-sky cameras have become increasingly popular in the field of astronomy, and many systems incorporating super-wide fish-eye optics have been developed, installed and operated. Applications of all-sky cameras include cloud detection (Hull Limmongkol & Siegmund; Hogg et al. 2001; Takato et al. 2003), bright star monitoring (Shamir & Nemiroff 2004; Gelderman 2004), transient detection (Kroll & Fleischmann 2001; Shamir & Nemiroff 2005d) and meteor science (Soper 1992; Oberst et al. 1998; Brosch & Manulis 2002; Spurny Oberst & Heinlein). However, while a substantial portion of the effort is directed toward the design of the hardware, less has yet been reported on the development of the software required for the autonomous operation of all-sky cameras. This software development can sometimes turn into an unexpectedly demanding task (Brooks 1987) and may delay the beginning of the effective operation of the camera.

In this paper we describe a software system that enables the operation and the analysis of astronomical pipelines. The algorithms, software and hosting services described in the paper can be used by all-sky camera operators who seek to increase the effectiveness of the data produced by their cameras, yet prefer to avoid the costs of software development. In Section 8.2 we describe the design of the software running on the remote station, including cosmic-ray hit rejection, star recognition, opacity maps and transient detection, and in Section 8.3 we describe the software design of the main server, which collects and archives the data transmitted from the remote stations.

8.2 Remote Station Software Design

The software that runs on the remote station is based on Linux Red-Hat, which is considered robust comparing to other existing PC operating systems. The remote station starts taking exposures in time intervals of 236 seconds. For instance, if the duration of the exposure is set to 180 seconds, an image will be taken every 236 seconds. If the duration is set to 360 seconds, an image will be taken every 472 seconds. Since 236 second intervals are used, a given image can be compared with another image taken on a different day, but at the exact same sidereal time so that the positions of the stars appearing in the two images are identical. For instance, the positions of stars recorded by an image taken at 12:00:00 UT are the same as the positions of stars recorded on the previous day at 12:03:56 UT or at 12:07:52 two days before. This feature is used for all-sky opacity maps and transient detection, which will be described later in the paper.

Images are taken only after sunset (based on USNO data) and before dawn. No images are taken during the day. The system also changes the duration of the exposure according to the position and phase of the moon. When the moon is up, the duration of the exposure is shorter.

In order to support as many stations as possible, yet without overloading the main server, all CPU-intensive image processing is done on the computer hosting the camera (the remote station). This approach takes advantage of the free CPU resources available while the camera is taking an exposure.

8.2.1 Cosmic Ray-Hit Rejection

After each new exposure is taken, cosmic ray-hits are rejected from the frame so the image can be processed. One technique of cosmic-ray hit rejection is by comparing several exposures of the same field (Windhorst Franklin & Neuschaefer; Fixsen et al. 2000). However, since one of the main purposes of the described system is to automatically detect optical transients that appear in single images (Shamir & Nemiroff 2005d) as well as meteors (Shamir 2005b), such techniques are not suitable for this system. In order to reject cosmic-ray hits without rejecting true transients, a fuzzy-logic based algorithm for cosmic-ray hit rejection from single images (Shamir 2005a) is applied. Like some other cosmic-ray hit rejection algorithms from single images, the rejections make use of the non-point source nature of cosmic ray splashes. Even though this algorithm does not introduce any better accuracy than other reported cosmic-ray hit rejection algorithms, it has a clear advantage in terms of computational complexity, which is a major concern in high image-rate pipelines. While some of the previously proposed algorithms may be too slow for systems at this scale (Rhoads 1994; Van Dokkum 2001; Pych 2004), the fuzzy-logic based algorithm processes a 1024×1024 frame in just 12 seconds (Shamir 2005a).

8.2.2 Star Recognition

In order to monitor bright stars, the software associates stellar objects visible in the frames with known-catalogued objects. This is done by applying a fuzzy-logic based star recognition algorithm designed specifically for super-wide angle astronomical images (Shamir & Nemiroff 2005a).

For each recognized star, the intensity of the local background is estimated and photometric data are collected. The intensity of the local background is estimated by the median of the 1600 pixels surrounding the point spread function (PSF). The intensity of the star is estimated by averaging the brightest pixels of the stars PSF and then subtracting the estimated local background. For each bright star (brighter than a certain preset magnitude), the averages of the brightest 1, 5, 9, 16 and 25 pixels are collected. For dimmer stars, however, the realized PSF is smaller so only the average of the top five pixels is currently listed. The average of the brightest pixels corresponds to their sum, but provides smaller numbers that can be handled using 16-bit integers. This reduces the bandwidth and storage space required to store these data.

Photometry files also list for each detected object some additional data taken from the Henry Draper catalogue such as the visual magnitude, the spectral type and celestial coordinates. The data are then stored in text-based HTML files that can be viewed by internet browsers and electronic spread sheets. The HTML files are xml-tagged, so that customized external software tools can easily browse and collect the data. The photometric precision of a single measurement is around 0.2 magnitudes, which allows star variability monitoring of high-amplitude variable stars such as Algol (Muzzin Shamir & Nemiroff). However, using measurements obtained over several nights can lead to photometric precision of 0.02 magnitudes, which allows monitoring low-amplitude variable stars such as Polaris (Nemiroff et al. 2005).

Another product of this processing is the generation of annotated JPG images, which labels the names of the constellations and bright stars (Shamir & Nemiroff 2004). These images can be used for "cosmetic" purposes, but can also be applied to educational purposes such as studying the night sky (Nemiroff 2005).

8.2.3 Opacity Maps

All-sky cameras can be used as simple cloud monitoring devices by simply inspecting the images by eye (Rutten 2003; Pereira et al. 2005, 2006). However, while thick clouds can be easily noticed, thin cirrus clouds at night are often invisible to the unarmed human eye. In order to provide a more informative view of the opacity of the night sky, photometric measurements of numerous background stars are combined with simultaneous sky brightness measurements to differentiate thin clouds from sky glow sources such as air glow and zodiacal light (Shamir & Nemiroff 2005b; Nemiroff & Shamir 2003). This is done by comparing all-sky images to canonical images taken on other nights at the same sidereal time. The canonical images are collected automatically by counting the number of stars brighter than a certain magnitude detected by the star detection algorithm (Shamir & Nemiroff 2005a). If the PSFs of a sufficient percentage of expected stars are found in the image, the image is classified as *clear*, and added to the database of canonical images where it is co-added to all other clear images taken at the same sidereal time.

When an image is taken, any PSF brighter than its background by a given σ is assigned with two values: a stellar intensity, and an estimated intensity of the background, as explained in Section 8.2.2. The values assigned to the PSFs are then compared to the values of the PSFs of a canonical image. The comparison is based on Equations 8.1 and 8.2.

$$M_{star} = T \cdot I_{star} + T \cdot I_{space} + I_{cloud} \tag{8.1}$$

$$M_{background} = T \cdot I_{space} + I_{cloud} \tag{8.2}$$

where M_{star} is the intensity of the light coming from pixels at the location of a star in the given image, $M_{background}$ is the intensity of the light coming from pixels *not* at the location of a star, I_{star} is the intensity of the star, I_{space} is the intensity of light from background space, I_{cloud} is the intensity of the cloud covering the star and T is the transmission around the star. Using Equations 8.1 and 8.2, the transmission is determined by Equation 8.3.

$$T = \frac{(M_{star} - M_{background})}{I_{star}} \tag{8.3}$$

where I_{star} is based on measurements taken from the canonical image such that $I_{star} = Mo_{star} - Mo_{background}$, where Mo_{star} and $Mo_{background}$ are the measured intensity of the star and the background, respectively, in the canonical image. This calculation of I_{star} is based on the assumption that no light is lost as starlight travels through air on a clear night. This assumption is not true, however, and provides a simplification of the problem that allows estimating the normalized transmission comparing to a clear night. This simplification gives

$$T = \frac{(M_{star} - M_{background})}{(Mo_{star} - Mo_{background})} \tag{8.4}$$

Since the desired product is a broad relative transmission map, the transmission of each pixel is calculated by interpolating the transmission measured directly along the lines to the stars in the image. This is performed by choosing, for example, the four nearest stars $S_{left}, S_{top}, S_{right}, S_{bottom}$ such that $|Y_{Sl} - Y_0| < X_0 - X_{Sl}$, $|Y_{Sr} - Y_0| < X_{Sr} - X_0$, $|X_{St} - X_0| < Y_0 - Y_{St}$, $|X_{Sb} - X_0| < Y_{Sb} - Y_0$, where X_{Sn} is the X image coordinate of the star S_n, Y_{Sn} is the Y image coordinate of the star S_n, and (X_0, Y_0) are the image coordinates of the given pixel whose sky transmission is being estimated. After finding the four nearest stars, a two-dimensional linear interpolation of the transmission measured by the four stars is performed, and the calculated value is determined as the relative transmission of the pixel. The computed relative transmission values of all pixels in the frame are then used for generating the all-sky opacity maps such that a high relative transmission implies high relative sky opacity and low relative transmission implies low relative opacity.

The computed normalized transmission T provided by this method is not intended to be used for scientific purposes such as photometric reduction. However, providing a numeric indication of the estimated atmospheric transmission is believed to have an advantage over a simple boolean clear/cloudy bit of information. The numeric indication can be used by robotic telescopes for the purpose of decision making (for instance, weighing the importance of the observation against the atmospheric transmission) and can also provide human observers with more informative indications regarding the visibility conditions of the area of the sky they observe.

In order to visualize the sky opacity map, the pixel is coloured in blue such that a stronger blue colour represents higher opacity. The maps are generated by changing the B component of each RGB triple such that $B = B_0 T + 255(1 - T)$, where B_0 is the original B value of the pixel and T is the sky transmission calculated using Equation 8.3. A scale added to the top left part of the transmission map presents the colours of several levels of opacity such that $\Delta M = -2.5 \log_{10} T$. Currently, opacity maps are evaluated visually by human observers, but the data may also be used to improve the effectiveness of robotic telescopes and autonomous sky surveys.

8.2.4 Image Compression

Some of the best sites for astronomical observations are located far from urban centers or other sources of light pollution. Therefore, a fast internet connection is not always available at some of these sites. Since FITS files consume a relatively large space, a data compression algorithm is sometimes required in order to maintain a high image-rate pipeline.

The FITS image compression algorithm used in the design described in this paper is *PHOTZIP* (Shamir & Nemiroff 2005c). PHOTZIP works by modeling, smoothing, and then compressing the astronomical background behind self-detected point sources while completely preserving values in and around those sources. Since the compression of the many background pixels is done in a symmetric fashion, the statistical properties such as mean and median (assuming normal distribution) are practically preserved. Since the values of the bright pixels are also preserved, photometry measurements of $I_{star} - I_{background}$ in the compressed/decompressed FITS images are practically identical to the original image.

Another advantage of PHOTZIP is that it allows users to control the compression/lossiness trade-off in terms of σ, so the compression factor can be adjusted to the available bandwidth at the site while users have an accurate sense of the lossiness due to the algorithm.

As an example of an existing application, the above algorithm is used regularly in CONCAM (Nemiroff & Rafert 1999) all-sky camera operating at SALT observatory (South Africa) with a compression factor of 85%. Experiments show that the photometric measurements in the compressed/decompressed images are practically the same as in the original images.

A more detailed description of the *PHOTZIP* algorithm is available at (Shamir & Nemiroff 2005c).

8.2.5 Transient Detection

All-sky wide angle astronomical pipelines can be used for the purpose of optical transient detection (Kroll & Fleischmann 2001). Since the ability of all-sky optics to observe faint sources is not comparable to even simple narrow-angle telescopes, only bright transients can be detected by all-sky super-wide angle devices. However, the low limiting magnitude of all-sky optics can be weighed against the large portion of the sky covered by the wide lens.

Brighter astronomical transients are sometimes considered more interesting to science (Lindley 1987; Menzies et al. 1987). Naturally, a wide-angle system is much more likely to capture such transients than narrow-angle telescopes covering only a very small portion of the sky at any given moment. In fact, one can reasonably assume that some bright short timescale transients visible to the unarmed eye are not noticed at all by the available narrow-angle sky surveys (Schaefer 2005).

In the proposed system, all-sky transient detection (Shamir 2005c; Shamir & Nemiroff 2005d) requires several logical steps. The first is the rejection of pixels dominated by cosmic-ray generated counts, as described in Section 8.2.1. Next, bright planets and high-amplitude variable stars are also rejected using a star recognition algorithm designed to find astronomical objects in wide angle frames (Shamir & Nemiroff 2005a). After cosmic-ray hits and bright planets are rejected, the system searches for PSFs that are a given σ brighter than the local background. Each PSF is then compared with the corresponding PSF in the canonical frame taken from the database of canonical images described in Section 8.2.3. The absence of moving parts in all-sky instruments simplifies the comparison of all-sky images so that finding corresponding PSFs in two images taken at the same sidereal time is relatively easy. The detected transients are listed in an xml-tagged HTML file and transmitted to the main server where they are available in real time. The current version of the transient detection mechanism does not provide information regarding the shape of the transient PSF so that users who wish to analyze the nature of the transient are required to observe the image by eye.

This system is currently operating in two all-sky cameras located in La Palma, Canary Islands and Cerro Pachon, Chile, and searches for optical transients brighter than 20σ over their local background.

One down-side of this transient detection mechanism is noise produced by satellite glints (Schaefer et al. 1987). This problem can be solved by comparing the detected transients with a database of predicted known satellites glints such as the one provided by *Heavens-Above.com*. This feature is not yet implemented in the current system. Experiments based on a device with a limiting stellar magnitude of 6.8 showed that approximately 5 bright transients are reported each night, most of which are satellite flashes and some others are bright meteors. Procedures performed in the server-side of the system reject satellite glints by comparing data from different cameras and by looking for sources that seem to rotate with the sky, as will be described in Section 8.3.2.

8.3 Server Side Software Design

The purpose of the server is to receive and archive the data sent from the nodes, generate GIF animations presenting the whole night as a movie, and provide web access to the data using a simple internet browser.

8.3.1 Web Server

In order to increase the availability of the data, all files are stored on a web server and are made accessible via the internet. The web site is organized such that the front page is a world-map labeled with the locations of the active all-sky cameras. Clicking on a location leads to a web page showing the last image taken at that site. Each location has an archive that provides access to previous data. The archive provides the FITS images, opacity maps and annotated JPG images described in Section 8.2.2. In addition, photometry files (also described in Section 8.2.2) are also provided. The files are xml tagged, so that external software tools can easily collect data from the server. The URL of each file is also of a standard format which is */xx/xxyymmdd/xxyymmddUThhnnss.html*, where *xx* is the symbol of the observatory, and *yy, mm, dd, hh, nn, ss* are the year, month, day, hour, minute and second of the beginning of the exposure, respectively. This file naming policy makes it simpler to find specific web pages, and also makes it easier for external automatic tools to navigate through the data.

The data are also copied to a database which allows performing simple queries. The database is based on the open source *My-SQL*. All queries are performed on-line using a web-based interface that supports queries according to stars (Henry Draper catalogue number), date and time.

Once the last image for the night is transmitted to the server, a GIF movie, which is a sequence of all images recorded that night, is generated on the server and placed in the archive. We found this feature useful for quick viewing of entire nights.

In addition, the web server also provides a discussion board where users can share their thoughts and experiences. The discussion board also serves as a log by documenting interesting events in a public fashion.

8.3.2 Transient Detection (server side)

In Section 8.2.5 we described a mechanism that detects transients in single images. The result of the single-image transient detection generated in the remote sites is transmitted to the server in the form of xml-tagged files. Each file

contains a list of transients detected so far that night, and each entry in the list includes the date, time, approximate celestial coordinates (RA,DEC) and the filename of the image in which the transient was detected. The RA, DEC coordinates are obtained by using the fuzzy-logic based coordinate transformation algorithm described in (Shamir & Nemiroff 2005a).

A computer program running on the server reads the files transmitted by the remote stations and searches for transients detected in the same celestial coordinates and at the same time. A transient recorded by two separate all-sky cameras in the same region of the sky at the same time may indicate that the source is astronomical, and not a satellite or a bright meteor.

Another criterion taken into account is persistency. The system searches for transients appearing at the same geocentric celestial coordinates for at least two consecutive exposures taken by the same camera. The purpose of this mechanism is to find optical transients that rotate with the sky.

8.3.3 Meteor Science

Networks of all-sky cameras are often used for meteor science (Soper 1992; Oberst et al. 1998; Brosch & Manulis 2002; Spurny Oberst & Heinlein) and can provide a comprehensive coverage of meteor showers (Rafert et al. 2001; Fleming et al. 2002; Brosch & Manulis 2002). It also proved to be efficient in the search for sporadic fireballs (Oberst et al. 1998; Spurny Oberst & Heinlein). The described software system allows users to access images recorded during meteor showers and automatically detects sporadic bright fireballs as discussed in Section 8.2.5.

Another feature related to meteor science using all-sky cameras is the 3D analysis of meteor trails by using data from two (or more) neighbouring all-sky cameras (Shamir 2005b). Since exposures at different stations start at the same time, it is likely that a meteor recorded by one camera is also recorded by another nearby camera. After transforming the image coordinates of the start and end of the meteor trail to celestial coordinates (Shamir & Nemiroff 2005a), simple triangulation can provide the altitude of the start of the meteor trail, the altitude of the end of the meteor trail, and the absolute trails length with accuracy of $\sim$3% (Shamir 2005b). When a meteor has more than two peaks of maximum brightness, the absolute distance between each peak can also be obtained (Shamir 2005b).

8.4 Conclusions

We described a software design that aims to provide a comprehensive software solution to all-sky panoramic astronomical pipelines. The software controls both the remote station, at which images are taken and processed, and the main server that archives and provides access to the data. The described software is currently used for the processing of all-sky panoramic astronomical pipelines generated by 11 all-sky cameras located in Mauna Kea and Haleakala (Hawaii), Cerro Pachon (Chile), Kitt Peak (Arizona), Mt. Wilson (California), Rosemary Hill (Florida), Siding Spring (Australia), Wise Obs. (Israel), Canary Islands (Spain), SALT (South Africa) and Hanle (India). The data collected in these observatories are available at http://nightskylive.net.

Bibliography

Brooks F. P., 1987, IEEE Computer, 20, 10

Brosch N., Manulis I., 2002, Proc. ACM, 209

Fixsen D., J., Offenberg J., D., Hanisch R., J., Mather J., C., Nieto-Santisteban M., A., Sengupta R., Stockman H., S., 2000, PASP, 112, 1350

Fleming, A. J., Nemiroff R. J., Perez-Ramirez D., Pereira W. E., Rafert J. B., 2002, BAAS, 34, 1173

Gelderman R., 2004, AN, 325, 559

Hogg, D. W., Finkbeiner D. P., Schlegel D. J., Gunn J. E., 2001, AJ, 122, 2129

Hull C. L., Limmongkol S., & Siegmund W. A., 1994, Proc. SPIE, 2199, 852

Kroll P., Fleischmann F., 2001, AN, 322, 315

Lindley D., 1987, Nat, 327, 90

Menzies J. W. et al., 1987, MNRAS, 227, 39

Muzzin V., Shamir L., Nemiroff R. J., 2005, AAS Meeting 205, 152.01

Nemiroff R. J., Rafert J. B. 1999, PASP, 111, 886

Nemiroff R. J., Shamir L., Muzzin V., Merlo M., Tilvi S. V., 2005, AAS Meeting 205, 54.12

Nemiroff R. J., 2005, AAS Meeting 205, 36.02

Nemiroff R. J., Shamir L., 2004, AAS Meeting 203, 38.14

Oberst J., Molau S., Heinlein D., Gritzner C., Schindler M., Spurny P., Ceplecha Z., Rendtel J., Betlem H., 1998, MP&S, 33, 49

Pereira W. E., Muzzin V., Merlo M., Shamir L., Nemiroff R. J., 2005, AAS Meeting 205, 48.13

Pereira W. E., Muzzin V., Shamir L., Nemiroff R. J., Merlo M., 2006, PASP, submitted

Pych W., 2004, PASP, 116, 148

Rafert, J. B., Nemiroff R. J., Perez-Ramirez D., Pereira W., Ftaclas C., Ashe S., 2001, BAAS, 34, 560

Rhoads J. E., 2000, PASP, 112, 703

Rutten R., 2003, ING Newsl., 7, 24

Shamir L., Nemiroff R. J., 2004, AAS Meeting 204, 76.07

Shamir L., Nemiroff R. J., 2005a, PASA, 22, 111

Shamir L., Nemiroff R. J., 2005b, PASP, 117, 972

Shamir L., Nemiroff R. J., 2005c, AJ, 129, 549

Shamir L., Nemiroff R. J., 2005d, AAS Meeting 206, 15.08

Shamir L., 2005a, AN, 326, 428

Shamir L., 2005b, JIMO, 33, 75

Shamir L., 2005c, Ap&SS, submitted

Schaefer B. E., 2005, ApJ, 621, 53

SchaeferB.E. et al., 1987, ApJ, 320, 398

Soper R. H., 1992, JBAA, 104, 27

Spurny P., Oberst J., Heinlein D., 2003, Nat, 423, 151

Takato N., Okada N., Kosugi G., Suganuma M., Miyashita A., Uraguchi F., 2003, Large Ground-based Telescopes, 4837, 87

Van Dokkum P. G., 2001, PASP, 113, 1420

Wells D. C., Greisen E. W., Harten R. H., 1981, A&AS, 44, 363

Windhorst R. A., Franklin B. E., Neuschaefer L. W., 1994, PASP, 106, 798