CSIRO PUBLISHING

www.publish.csiro.au/journals/pasa *Publications of the Astronomical Society of Australia*, 2005, **22**, 111–117

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

Lior Shamir[A,B] *and Robert J. Nemiroff*[A]

[A] Michigan Technological University, Department of Physics, Houghton, MI 49931, USA
[B] Corresponding author. Email: lshamir@mtu.edu

Received 2004 October 5, accepted 2005 February 26

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 cataloged 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.

Keywords: methods: data analysis — methods: statistical — techniques: image processing

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 (Zadeh 1965, 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 membership 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 (NSL; Nemiroff et al., in preparation).

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

10.1071/AS04072 1323-3580/05/02111

https://doi.org/10.1071/AS04072 Published online by Cambridge University Press

latter transformation is broken up into two parts:

$$(altitude, azimuth) \longmapsto x \tag{1}$$

$$(altitude, azimuth) \longmapsto y \tag{2}$$

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. The x and y coordinates of any given star visible in an astronomical image can be computed as follows:

$$x = x_{\mathrm{zen}} + \sin(angle) \cdot distance \tag{3}$$

$$y = y_{\mathrm{zen}} + \cos(angle) \cdot distance \tag{4}$$

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.

2.1 Reference Stars

Each of the two models is 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.

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:

$$f_1\colon azimuth \longmapsto angle \tag{5}$$

$$f_2\colon altitude, azimuth \longmapsto distance \tag{6}$$

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 dependent 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.

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 function that is built in the form of a triangle (Zadeh 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°. Assuming $\alpha_1 < \alpha_2 < \alpha_3$, the membership functions that will be added to the model are described in Figure 1.

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

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

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

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

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

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

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

https://doi.org/10.1071/AS04072 Published online by Cambridge University Press

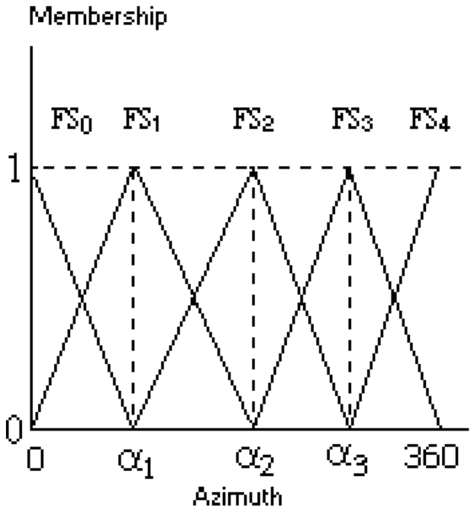

Figure 1 The membership functions of the fuzzy sets (FS_0 to FS_4) created by the four reference values $(0, \theta_0), (\alpha_1, \theta_1), (\alpha_2, \theta_2), (\alpha_3, \theta_3)$ for $f_1(azimuth \longmapsto angle)$.

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 the *weighted average*, which is an efficient defuzzification method when the fuzzy logic model is built according to a set of singleton values (Takagi & Sugeno 1983, 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:

$$\begin{aligned} FS_0 &\longmapsto \theta_0 \\ FS_1 &\longmapsto \theta_1 \\ FS_2 &\longmapsto \theta_2 \\ FS_3 &\longmapsto \theta_3 \\ FS_4 &\longmapsto 360 \end{aligned}$$

For instance, suppose that the first three reference stars have azimuths of 0°, 10°, and 20°, and their polar angles are 3°, 12°, and 22° respectively, such that $\alpha_0 = 0°$, $\theta_0 = 3°$, $\alpha_1 = 10°$, $\theta_1 = 12°$, $\alpha_2 = 20°$, and $\theta_2 = 22°$. 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:

$$\begin{aligned} FS_0 &\longmapsto 3 \\ FS_1 &\longmapsto 12 \\ FS_2 &\longmapsto 22 \end{aligned}$$

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°. The value 6 has a level of membership of $1 - (1/10) \cdot 6 = 0.4$ to the fuzzy set FS_0 and $(1/10) \cdot 6 = 0.6$ 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 effect 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 $(0.4 \cdot 3 + 0.6 \cdot 12)/(0.4 + 0.6) = 8.4$. This computation practically provides the same results as a simple linear interpolation of the value 6 with the two closest neighboring reference stars.

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 $(x_{\text{zen}}, y_{\text{zen}})$ 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° (north) and altitude of 30° can be at a distance of 150 pixels from $(x_{\text{zen}}, y_{\text{zen}})$, while another stellar object at the same altitude (30°) but at azimuth of 60° will be at a distance of 155 pixels from $(x_{\text{zen}}, y_{\text{zen}})$. Moreover, $(x_{\text{zen}}, y_{\text{zen}})$ 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 $(x_{\text{zen}}, y_{\text{zen}})$, it is required to take not only the altitude of the stellar object into consideration, 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°, the second set contains reference stars near azimuth 90°, and the other two sets contain stars near azimuth of 180° and 270° respectively, i.e. all reference stars used for this model should be fairly close to the azimuth of 0°, 90°, 180°, or 270°. 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

https://doi.org/10.1071/AS04072 Published online by Cambridge University Press

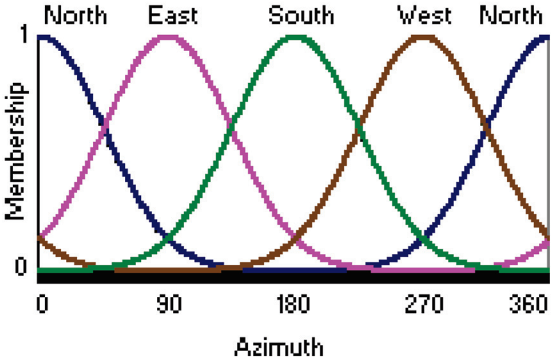

Figure 2 The fuzzy sets of the four directions (*North*, *South*, *East*, *West*).

different than four separate linear and/or Gaussian interpolations for the four directions. The membership functions of the fuzzy sets are described in Figure 2.

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

North:

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

East:

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

South:

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

West:

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

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 the f_2 model can be demonstrated by an example: suppose the model is built based on the following six reference stars listed in Table 1.

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 such that their altitudes are closest to the altitude of S_1 (such that one is greater than 68° and one is smaller than 68°), *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.

Table 1. Reference stars

Star	Azimuth	Altitude	Angle	Distance (pixels)
S_1	0°	68°	2.4°	215
S_2	359°	62°	1.2°	224
S_3	1.5°	72°	3.4°	206
S_4	90°	66°	94.2°	180
S_5	91°	62°	95.6°	188
S_6	91.6°	70°	95.9°	172

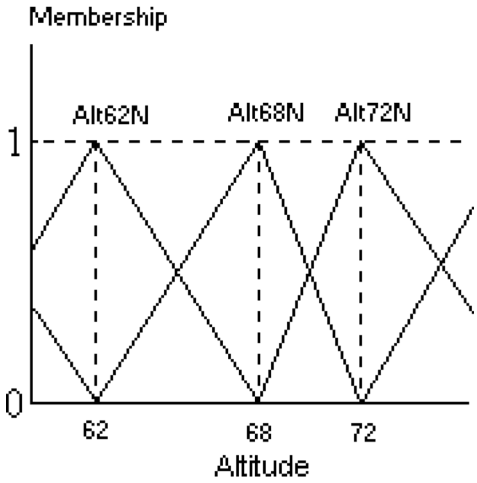

Figure 3 The membership function of the fuzzy set *Alt68N*. The figure also include parts of the membership functions of *Alt62N* and *Alt72N* that were added by the two neighboring reference stars S_2 and S_3.

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°). The practical effect of this rule will be stronger as the coordinates are closer to the 68° parallel. The fuzzy rules that will be added by the other reference stars are:

$$\begin{gathered} Alt62N \wedge North \longmapsto 224 \\ Alt72N \wedge North \longmapsto 206 \\ Alt62E \wedge North \longmapsto 188 \\ Alt66E \wedge North \longmapsto 180 \\ Alt70E \wedge North \longmapsto 172 \end{gathered}$$

Now suppose that we want to use this model in order to compute the distance (in pixels) of a

https://doi.org/10.1071/AS04072 Published online by Cambridge University Press

stellar object with an azimuth of 30° and altitude of 64°. The value 64 has a membership value of $(64-62)/(68-62)=1/3$ to the fuzzy set *Alt68N* (which was added to the model by S_1), and a membership value of $1-(64-62)/(68-62)=2/3$ to the fuzzy set *Alt62N* (which was added to the model by S_2). Also, this value would have a membership value of $(64-62)/(66-62)=0.5$ to the fuzzy set added by S_4, and a membership value of $1-(64-62)/(66-62)=0.5$ to the fuzzy set added by S_5. The membership value of the azimuth 30° to the fuzzy set *North* is $\mathrm{e}^{-30^2/2\cdot 45^2}\simeq 0.8$, and to the fuzzy set *East* it is $\mathrm{e}^{-(30-90)^2/2\cdot 45^2}\simeq 0.412$. The membership values to the fuzzy sets *South* and *West* are very close to zero 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 zero.

The computation process is based on *product* inferencing and *weighted average* defuzzification. Therefore, the output value of the computation would be:

$$\frac{\begin{array}{c}215\cdot 0.8\cdot 0.333+224\cdot 0.8\cdot 0.667\\ +180\cdot 0.412\cdot 0.5+188\cdot 0.412\cdot 0.5\end{array}}{\begin{array}{c}0.8\cdot 0.333+0.8\cdot 0.667\\ +0.412\cdot 0.5+0.412\cdot 0.5\end{array}}\simeq 208.47.$$

3 Example Application to NSL Data: Accuracy and Complexity

The fuzzy logic based transformation formula has been tested and is in practical use with the NSL project (Nemiroff et al., in preparation). Purposes of the global NSL network of fish-eye 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 CONCAMs at many of the world's premier observatories. When the Moon is down, CONCAMs take 180-s exposures every 236 s, 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 cataloged stellar objects can be associated with PSFs that appear in the NSL 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-cataloged 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 ← $\infty$
 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 1996). 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 NSL project currently cannot get deeper than magnitude 6.8, the catalog being used is a subset of the 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 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 (appearing at the lower right of the frame) could not be associated with any cataloged 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% 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

https://doi.org/10.1071/AS04072 Published online by Cambridge University Press

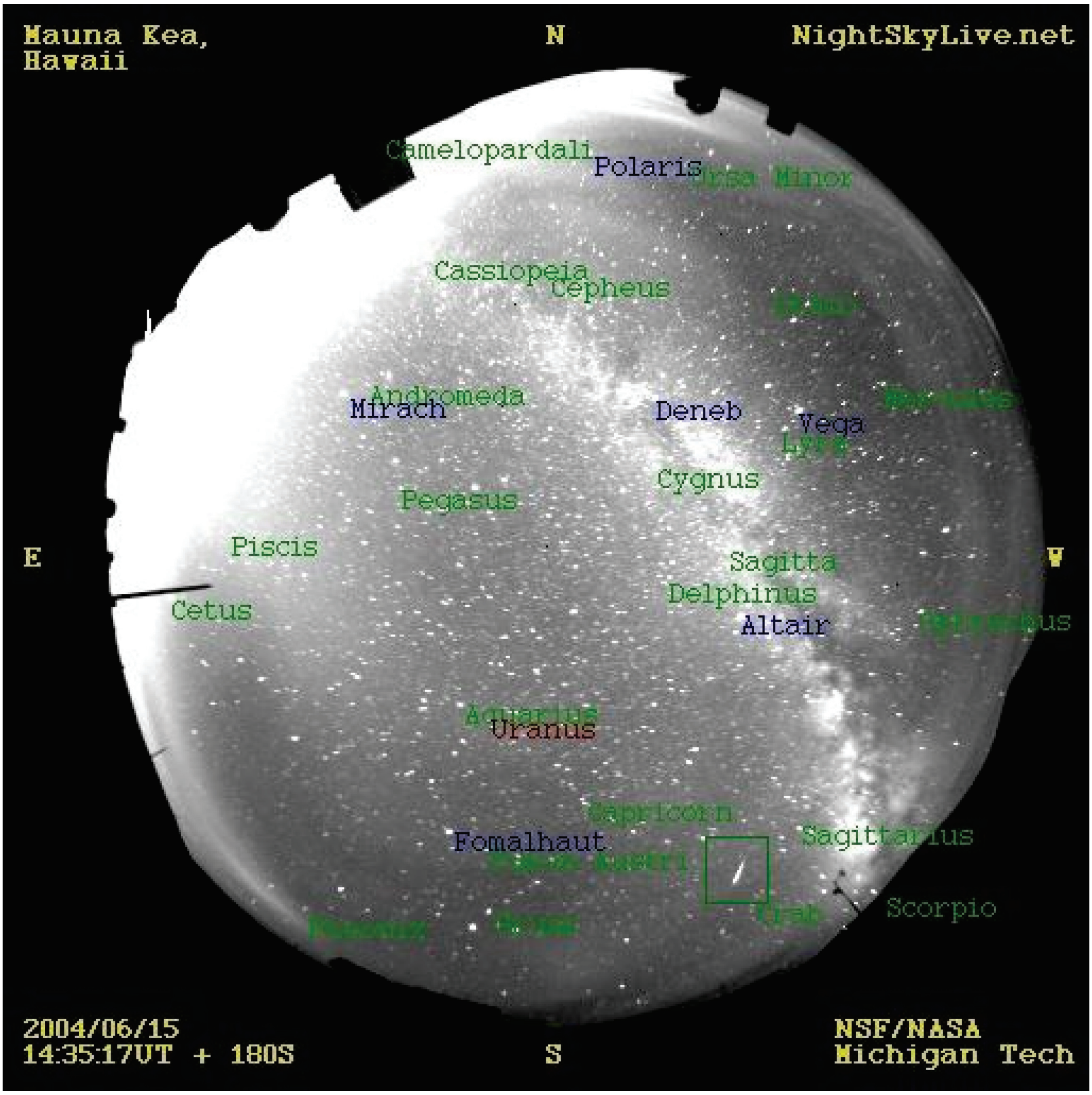

Figure 4 An NSL picture with labeled stars, constellations, a planet, and a meteor (boxed) processed using the fuzzy logic based transformation formula.

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 catalog 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 is 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 are beyond the scope of this paper, more information can be found elsewhere (Nemiroff et al., in preparation).

The inverse computation of the presented transformation formulae (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

https://doi.org/10.1071/AS04072 Published online by Cambridge University Press

Table 2. Accuracy level of the transformation as a factor of the number of reference stars

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

of meteors detected by the twin CONCAMs located at Mauna Kea and Haleakala.

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., in preparation). The NSL project uses fish-eye images with an extreme viewable angle of 180°. 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 consideration 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 2 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).

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 dependent upon the number of reference stars. The accuracy needed to identify stars unambiguously is dependent upon 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 NSL frames. As shown in Table 2, this can be achieved using a fuzzy logic model that is built with a constant number of ~80 well spread out reference stars. Therefore, the final fuzzy logic model contains a constant number of ~80 rules and ~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 ~1000 coordinates are transformed in less than one second by a system equipped with an Intel Pentium III processor at 850 MHz and 128 MB of RAM.

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 a constant). When applied to the NSL project, the method presented 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% of the time down to magnitude 5.6, and automatically detect non-cataloged bright objects. This technique demonstrates that it is possible to use fuzzy logic based algorithms to reduce the effect 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.

References

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

ESA 1997, The Hipparcos and Tycho Catalogues, ESA SP-1200

Takagi, T., & Sugeno, M. 1983, in Proc. IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision Analysis, 55

Takagi, T., & Sugeno, M. 1985, ITSMC, 20, 116

Zadeh, L. A. 1965, InfCo, 8, 338

Zadeh, L. A. 1978, IJMMS, 10, 395

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

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

https://doi.org/10.1071/AS04072 Published online by Cambridge University Press