Simulation of point-like optical flashes in the sky

Lech Wiktor Piotrowski[a] and Marcin Sokolowski[b]

[a]Institute of Experimental Physics, Warsaw University;
[b]Soltan Institute for Nuclear Studies, Warsaw

ABSTRACT

We present optimized performance of algorithms developed for detection of point like, visible light bursts in the sky. The algorithms based on analysis of series of consequtive images and detecting local differences between them were tested on simulated data obtained by inserting images of stars of different magnitudes at random positions. We discuss the choice of parameters which results in the maximal detection efficiency and simultaneously keeps the number of false detections as low as possible.

1. INTRODUCTION

The "π of the sky" project[1] is conducted by Soltan Institute for Nuclear Studies in co-operation with Institute of Experimental Physics (Warsaw University), Center for Theoretical Physics (Polish Academy of Sciences) and Institute of Electronics Systems (Warsaw University of Technology). It's ultimate goal is real-time analysis of sky images coming from two sets of 16 CCD cameras covering over π spherical angle of the sky in a search for optical flashes of cosmological origin.[2] Due to large primary data flow only tiny fraction of all acquired images can be stored on hard disks. The selection of interesting events is performed in several steps called trigger levels. The goal of the level 1 algorithm described in this paper is to detect at least 80% of visible burst events and reduce the amount of data about 100 times.

In this paper we describe methods of detection and vetoing based on analysis of local differences between subsequent images. Understanding of virtues and defficiencies of such algorithms have lead us to a concept of a family of more efficient, kernel-based algorithms. Description and analysis of kernel-based algorithms will be a subject of a subsequent paper.

2. FLASH RECOGNITION METHODS

In an ideal situation, a digital sky image would consist of isolated pixels of different brightness representing stars on a uniform, dark background. Subtraction of two frames would then show only these objects which were either present only on one of the frames or changed their brightness. The real image of the sky as observed through the Earth atmosphere and instrument optics makes flash detection more complex.

2.1. Object imaging

Unlike an ideal, the real digital star image appears as a distribution of light intensity among several pixels, usually well approximated by a gaussian profile. There are three main reasons that cause such a blur of the light:

- Scattering in the atmosphere
- Diffraction of light in the aparatus
- Non-ideal optics of the aparatus

The number of photons reaching a given pixel in a given time is described by the Poisson distribution, which for a large number of photons can be approximated by a gaussian. On top of that there are effects related to the CCD sensor and readout electronics, like dark current, readout noise, different sensitivities of different pixels, etc. All this makes efficient flash detection algorithms a rather non-trivial concept.

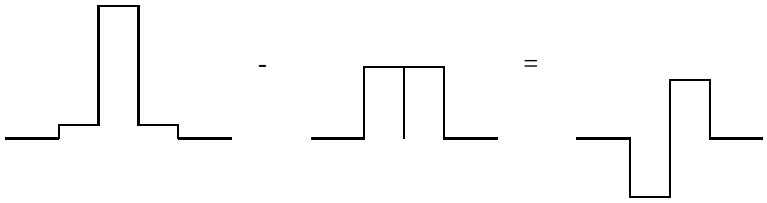

Figure 1. Sample star distribution subtraction

2.2. Pixel based algorithms

Subtraction of two subsequent frames should give an empty image containing only uncorrelated electronics noise and sky background fluctuations. No stars should be visible. However, as an average star profile FWHM in analysed data is about 1.5 pixel, the intensity distribution is concentrated on a single bright pixel only when the star centre is located near the pixel centre. In a case where the star centre is located close to the pixel edge, the brightness from a single pixel is divided between two pixels, which still are much brighter than the background.

Both kinds of distribution are possible for a single star due to rotation of the Earth between images. Thus the subtraction of two frames will often leave a dipol-like objects as stars remnants, with one bright pixel and one very dark (Fig. 1).

A single-pixel based algorithm treats separately (ignoring the neighbours) every bright pixel on a frame which results from subtraction. The positive part of the dipol is often higher than selection threshold and causes a false detection. Thus this kind of algorithm is inefficient as far as this project is concerned.

The effect of different light distributions among pixels can be reduced if instead of a single pixel *a group* of neighbouring pixels is taken into account. In both cases the sum of intensities should be similar. The only difference should be due to fluctuations. We have tried several multi-pixel based algorithms, in which all pixels contained in a region of a specific shape are analysed.

The main advantage of using a multi-pixel based algorithm is reduction of problems caused by a variable star image. An influence of the background fluctuations is also reduced, for the sum of randomly intensive pixels comes to a constant value along with increasing of number of pixels in a shape. However, it also increases an overall background and finally makes the signal undetectable. Also, big shapes are likely to include not only a single star, but also other sky objects, which brightness is added to that of the star of interest. To minimalize effects of this kind one has to choose an optimal size and shape.

The main difficulty of optimizing an algorithm based on summation of pixels stems from balancing of a threshold for the background intensity, below which an object is treated as "sky background", i.e. it remains undetected and a threshold for event intensity, above which a region is treated as a containing a "detected flash". A local change of brightness in a frame, caused by Moon light, thin clouds etc., is often undetectable, but may cause mistakes comparing succeeding frames.

Pixels in an object can be analysed in a more complex way, introduced in the kernel-based algorithms, where the image is convoluted with a specially designed kernel. Those algorithms are, however, more time consuming. A family of such algorithms will be discussed in a subsequent paper.

2.3. Object based algorithms

Many sky analysis algorithms are based on detection (and identification) of all interesting objects on a frame, like stars, satellites, etc. A list of detected objects for each frame is compared with lists arising from similar analysis of other frames and results of such a comparision are analysed. Algorithms of this kind are rather accurate, but identifying all stars on each frame is very time-consuming and does not fit well into a real-time, big data stream analysis.

In the current stage of the "π of the sky" development some object based algorithms are forseen after the first, pixel based data reduction, but they are not discussed here.

2.4. Possible backgrounds

False events rate reduction requires that the algorithm deals properly with some special cases - mostly artificial objects that are visible in the sky, like satellites or planes and cosmic particles.

Fast objects, like planes, appear on a frame as a long line, not visible in the preceding frame. In general, simple algorithm recognizes them as a number of flashing objects. A possible filter introduced in later stage of testing is based on vetoing of a group of events that appear in a close neighbourhood.

More difficult to filter are satellites and rubbish on Earth's orbit. They appear as pointlike, bright flashes, visible in different places on several frames. Currently an object based algorithm tries to fit all detected events to a straight line or a parabola - if the fit is successful the events are classified as human artifacts and rejected.

The third problem are cosmic radiation particles that excite parts of the CCD chip and appear as bright pixels in a small area on a single frame. A method to filter them is to add a condition, that a "detected flash" in the given region appears on at least two succeeding frames.

3. FLASH DETECTION ALGORITHMS

This section describes specific methods used for detection of light bursts in the sky. Basic parameters are defined using algebraic notation; parameter names used in the program are given in parenthesis.

3.1. Data reduction levels

As it was mentioned before, the amount of data gathered (1 frame per camera per 10 seconds) requires online analysis and reduction before the final analysis of selected events can be performed. For tests described in this paper the analysis was split into two levels:

- Object recognition and constant stars veto (level 1)
- Object surrounding (confirmation) recognition and constant star veto (level 1.5)

and the corresponding naming convention is used thorough this paper. By convention, fractional numbers indicate optional levels which, as it turns out, can be omitted in the analysis. Subsequent selection levels are under development and will be described elswhere.

3.2. Object recognition and constant star veto (level 1)

For every frame a brightness of every pixel or sum of brightness of pixels contained in specific objects - denoted here as "shapes" (with shape and size strictly defined by specific parameters) are registered. We have tried several methods of processing this information. In the first method, the shape is qualified for further analysis, if the sum of brightness on current frame Σ_n exceeds a specified threshold defined for a pixel T_n (CCD_SIG_TRESH_PER_PIXEL_LEVEL_1) multiplied by the number of pixels in a shape k:

$$\Sigma_n > T_n \cdot k$$

If the object is qualified, a shape's overall brightness in its position on the previous frame Σ_v is compared to a specified threshold defined for a pixel: T_v (CCD_MAX_PREV_PER_PIXEL_LEVEL_1) multiplied by the number of pixels in the shape, and is not vetoed if the following condition is fulfilled:

$$\Sigma_v < T_v \cdot k$$

Additionaly, another method was implemented, but abandoned after initial tests. In that method, recognition and vetoing are performed simultaneously. The following condition must be fulfilled for the object recognition and acceptance:

$$\Sigma_n - \Sigma_v > T_n \cdot k$$

Note that T_n plays now a role of a threshold for a difference of the sums. In order to improve the signal to the background ratio, for Σ_v one can use an average of several frames.

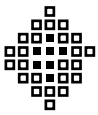

Figure 2. Example of a cluster (fi lled rectangles) and an extended cluster (empty rectangles)

3.3. Object surrounding (confirmation) recognition and constant star veto (level 1.5)

An optional level 1.5 can be used for reduction of data coming from level 1 analysis. The conditions are similar, although more complicated shapes and different thresholds can be used: a threshold for current frame - T_{cn} (CCD_CONFIRM_TRESHOLD_BY_PIXEL), a threshold for previous frames - T_{cv} (CCD_CONFIRM_MAX_NOISE_PER_PIXEL). We follow the convention that the index c stands for the confirmation step, index n for the current (new) frame and index v for previous (vetoing) frames. In level 1.5 conditions Σ_{cn} means sum of pixels is a shape on the current frame and Σ_{cv} on previous frames. Thus the following double condition results:

$$\Sigma_{cn} > T_{cn} \cdot k_c$$

$$\Sigma_{cv} < T_{cv} \cdot k_c$$

where n_c is a number of pixels in the level 1.5 shape. A single condition is written below:

$$\Sigma_{cn} - \Sigma_{cv} > T_{cn} \cdot k_c$$

3.4. Pixels, Clusters and Shapes

As mentioned before, a burst can be recognized by comparing brightness - pixel by pixel - on different frames, or by comparing sums of pixels contained in a shape or a cluster (in level 1.5). In this paper we are dealing with clusters and shapes, which are defined below.

3.4.1. Clusters

For every pixel or shape passing level 1, we search for neighbouring pixels with brightness higher than T_k (CCD_MAX_NOISE_LEVEL). All those pixels form a cluster, for which the sum is computed. It can be further extended to a bigger field including CCD_PIXELS_AROUND_TO_CONFIRM more pixels (Fig.2). This was expected to be helpful to correct for nonideal aligment between frames.

3.4.2. Shapes

Three types of shapes were introduced in this algorithm, which were set by s (CCD_NEIGHB_SHAPE) for level one and s_c (CCD_CONFIRM_SHAPE) for level 1.5. Examples of these shapes, with radius from 0 to 4 are shown in Fig.3.

The shape with s=0 stands for a cluster. In principle, one could use a cluster in level 1 analysis, but it was not implemented because of poor performance.

Radius is defined by parameters r (CCD_NEIGHB_REDIAL) and r_c (CCD_CONFIRM_REDIAL). The radius r or $r_c = \frac{W-W_b}{2}$, where W stands for the width of a shape of this radius in pixels and W_b is a width of a shape with $r = 0$ (one pixel for $s = 1, 2$ and three pixels for $s = 3$).

3.5. Σ_v and Σ_{cv} definition

Next, we consider the following question: which frame, or set of frames, should be used to calculate Σ_v and Σ_{cv}. Two answers to this question have been tested.

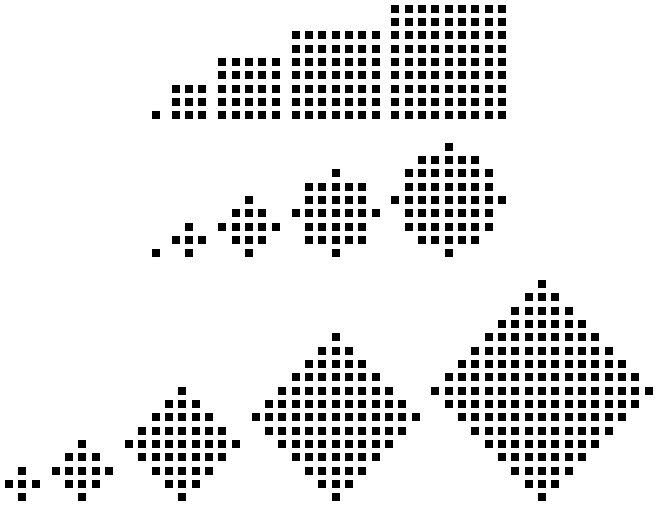

Figure 3. Examples of shapes with radius 0 to 4: squares ($s = 1$), "circles" ($s = 2$) and diamonds ($s = 3$)

3.5.1. Maximum from previous frames

In this method, pixel values were remembered for N frames preceding the current frame. Initially, $N = 5$ was chosen. The maximum value of Σ_v and Σ_{cv} from these frames were used. We denote this method as max_from_prev.

3.5.2. Homeopatic summation

This method was used only for level 1 algorithm with the parameter definition:

$$\Sigma_v = a \cdot \Sigma_{n-1} + (1 - a) \cdot \Sigma_{v-1}$$

where Σ_{n-1} and Σ_{v-1} are Σ_n and Σ_v for previous frame. Parameter a defines the level of sensivity to current and previous frames and in the tests we took $a = \frac{1}{2}$. In general, Σ_v is convergent if $0 < a < 1$.

4. TEST DATA

The test frames were taken with 768×512 CCD camera connected with $f = 58$ mm, $f/2$ Zenit lenses. For the compatibility with future cameras with larger pixels, 2×2 binning was used. Exposure time was set to 5 seconds, and the read-out time was equal to 5 seconds.

5. SIMULATION

Calculation of algorithm's detection rate was made by simulating bursts: on every frame (except for a few frames at the beginning) an image of a star of a selected magnitude was inserted at a certain, randomly chosen position. Two samples of cosequtive frames without and with an inserted star are shown in Fig.4 and 5.

The algorithm registers positions of inserted and detected bursts. If the inserted and detected burst positions are compatible with each other the event is treated as a correctly recognized flash. Otherwise, it is treated as a false detection. The number of correct detections divided by the number of insertions forms "efficiency", while number of false events forms "background".

The influence of various parameters on the efficiency and the background is easily visible on plots showing parameter value versus the background or the efficiency obtained with a number of frames. This way a general idea on parameter role in detection can be determined. Plots of the efficiency versus the background have allowed us to determine the best values of parameters.

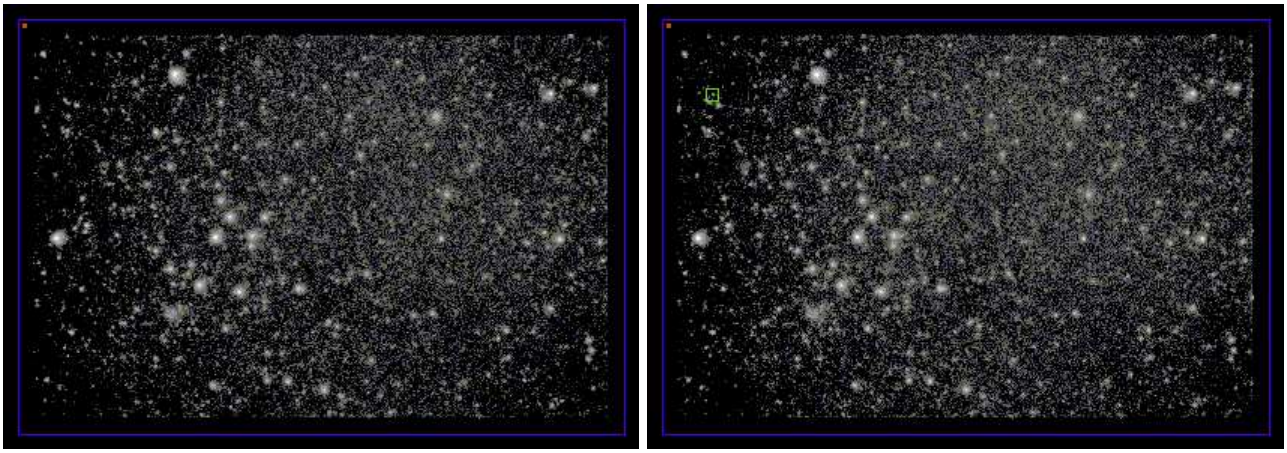

Figure 4. Sample frame with a pasted star indicated by a small square (the upper left corner of the right image) and without a pasted star (the left image)

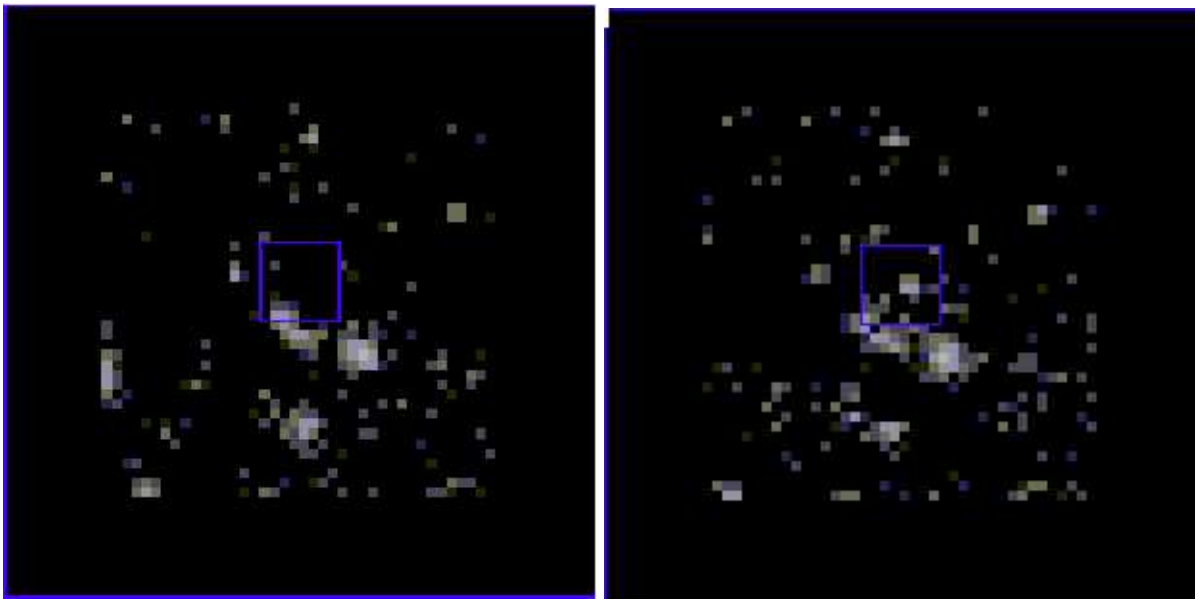

Figure 5. Magnifi ed areas of sample frame with (the right image) and without (the left image) pasted sample

6. RESULTS

In this section we describe results of our search for optimal values of parameters. We study separately the influence of thresholds and shape and sizes, respectively, on the background and the efficiency of flash recognition algorithms.

6.1. The difference between $homeopathic$ and max_from_prev methods note

No important differences in results between $homeopathic$ and max_from_prev methods were observed. Plots, not included in this paper, showed similar efficiency and background curves for similar values of parameters.

6.2. Thresholds influence

6.2.1. Thresholds for the current frame T_n and T_{cn}

These two parameters define thresholds for detection on the current frame, accordingly for level 1 and 1.5 and thus set restrictions on minimal brightness for a shape on a current frame to be qualified as an "event". It is expected that the efficiency and the background will decrease with an increase of these thresholds.

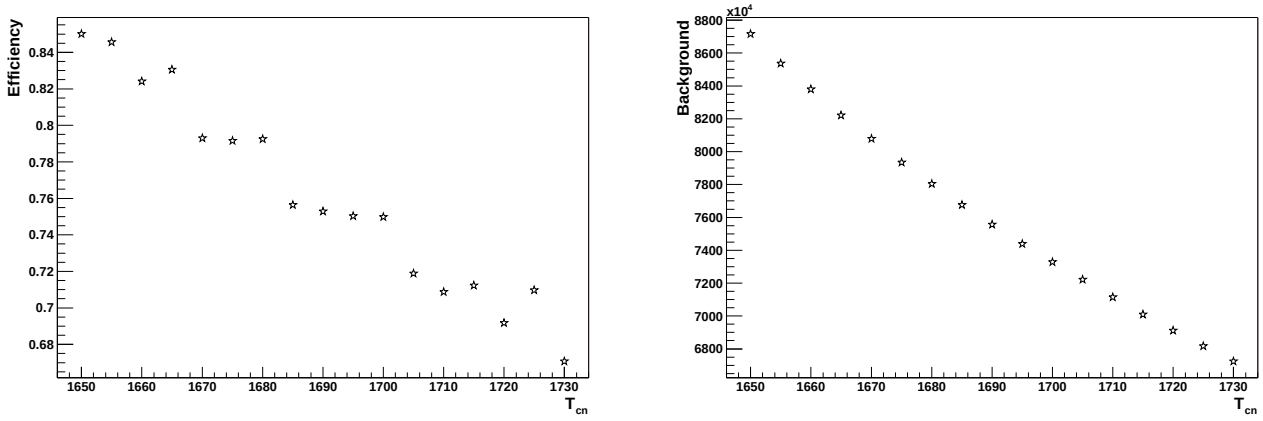

Figure 6. Effi ciency and background vs T_{cn} for single-pixel

Fig.6 shows curves for single-pixel shape, which fall along with increase of T_{cn}. Plots made for larger shapes, not included in this paper, contain lower range of the efficiency and the background and show that corresponding curves asymptotically fall to zero.

In order to see that such behaviour is in fact consistent with expectations one should consider that

- the number of stars visible on the sky decreases with increasing expected brightness
- an object qualified as “event” has not only to exceed T_{cn} on current frame but also to be lower than T_{cv} on the previous one

The consequence of the first fact is obvious. The second means that the qualified star needs to fluctuate between two frames by a factor of $T_{cn} - T_{cv}$. With constant T_{cv} such a fluctuation - which would lead to qualification of the star as a flash - becomes less and less probable along with increasing T_{cn}.

"Steps" visible on the the efficiency graph of Fig.6 are due to the small choice of sample stars pasted on frames in simulation. Each step probably appears as another pixel in the sample star falls below T_{cn}. More detailed study have to use larger set of sample stars to avoid this effect.

6.2.2. Thresholds for the previous frames T_v and T_{cv}

T_{cv} and T_v define thresholds for "sky background" on the previous frame for level 1 and 1.5, respectively. An average brightness inside a qualified shape cannot exceed these thresholds. In contradistinction to section 6.2.1, it is expected that the efficiency and the background will increase with increase of T_v and T_{cv}.

The expected efficiency curve should follow an error function, because of the background pixels brightness distribution, which can be approximated by a gaussian. The area under the gaussian curve represents an overall number of pixels with brightness in specified range, which begins at 0 and ends at T_v or T_{cv}. Thus the efficiency, dependent on the number of pixels contained in a given shape, is a gaussian integer.

Probably most of false events are due to fluctuations of stars brightness. Contrary to section 6.2.1, here T_n and T_{cn} are constant. Thus assuming $T_{cv}\ (T_v) < T_{cn}\ (T_n)$, the background curve is expected to increase with T_v and T_{cv}.

Fig.7 and 8 show T_{cv} curves for single-pixel and circle and diamond shapes with $r = 1$. It may seem that figures show different dependences, but in fact Fig.7 includes only the higher part of erf(), while Fig.8 the whole of it. The width of the

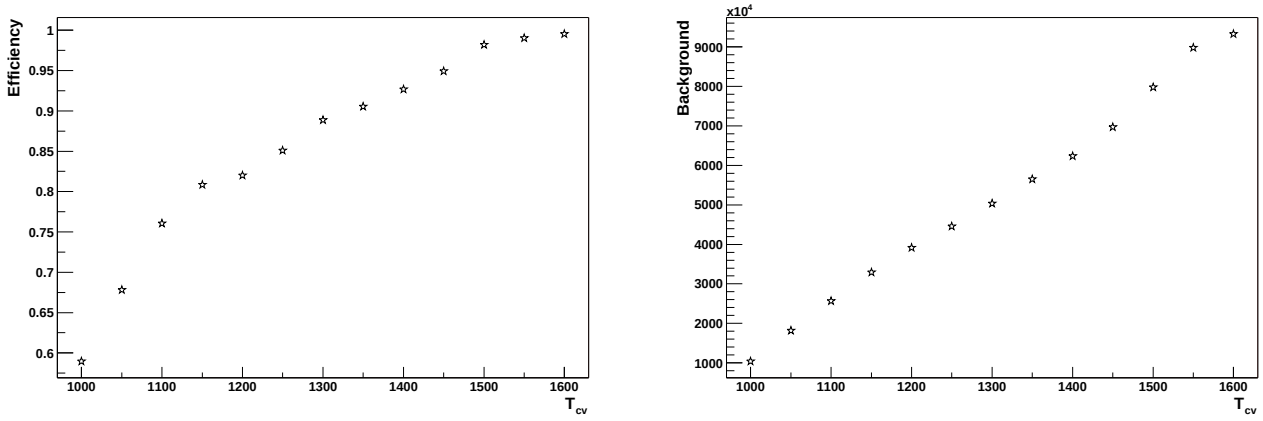

Figure 7. Effi ciency and background vs T_{cv} for single-pixel

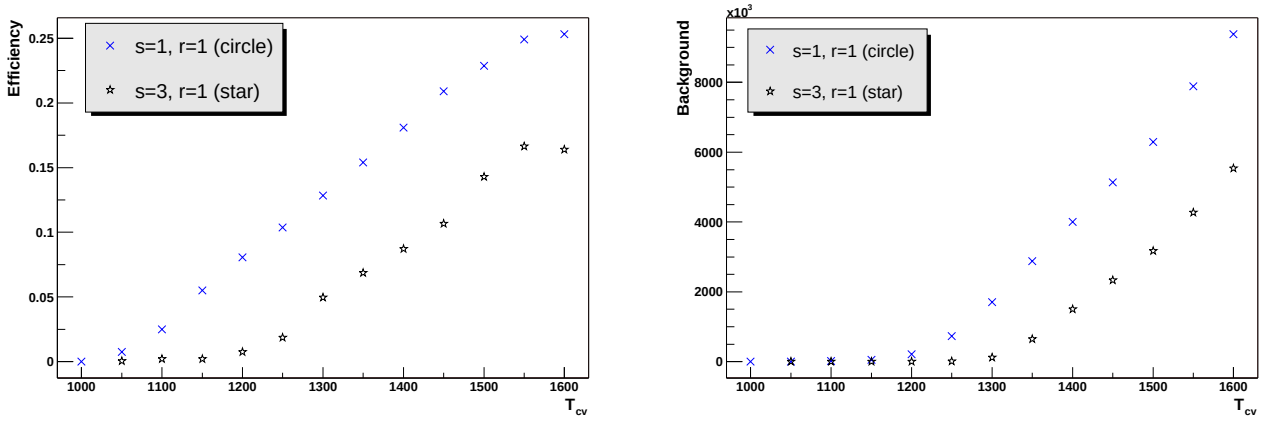

Figure 8. Effi ciency and background vs T_{cv} for $s = 1, 3$ with $r = 1$

flat area visible at the beginning of the background curve increases with shape size. For a single-pixel shape the area is not visible in the tested range.

Apparently the gaussian integral is dependent on the number of pixels included in a shape. The FWHM is proportional to $\frac{1}{\sqrt{n}}$, where n stands for the number of pixels in the shape. Thus the same $(0, T_{cv})$ range covers different areas of the gaussian curve for shapes with different n. This causes erf() scaling along the x-axis. In addition, one observes also a shift of the erf() along the x-axis. The gaussian maximum location is dependent on the T_{cn} cut. The reason was mentioned in section 6.2.1 - an average brightness of pixel in a shape shifts towards average brightness of pixel in a frame with increase of n. One another thing observed is the scaling of the y-axis - erf()s for bigger shapes are much lower than for smaller shapes. This is simply due to the smaller efficiency for bigger shapes, described later.

For background curves, flatness of the area extending with shape size results from use of cluster as a vetoing shape. The cluster is formed from pixels neighbouring to the level 1 shape, exceeding a specified brightness. Increasing the level 1 shape size increases the number of neighbouring pixels and ipso facto a chance to accept a larger number of them. For such level 1.5 shape, normal vetoing calculations are performed. Increasing the amount of pixels reduces the chance of overall cluster fluctuation and thus the flat, near-zero area of the curve appears.

6.3. Field rotation effects

During the tests an improved frame alignment algorithm has been implemented. In previous tests the frame contents were shifted by a constant vector and currently they are also rotated by a small angle. Similar tests as in section 6.2.2 were

performed and a significant background decrease was expected without any reduction of the efficiency.

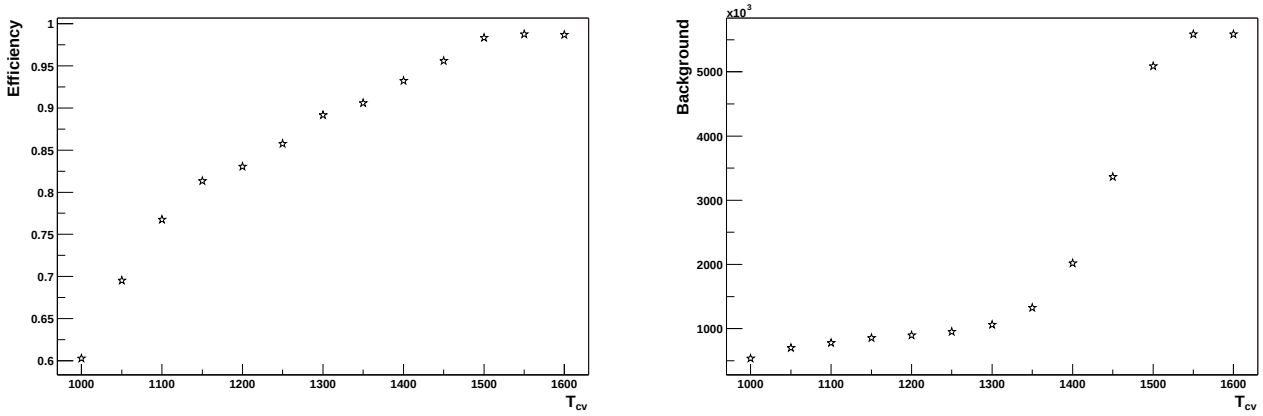

Figure 9. Effi ciency and background vs T_{cv} for single-pixel (rotation)

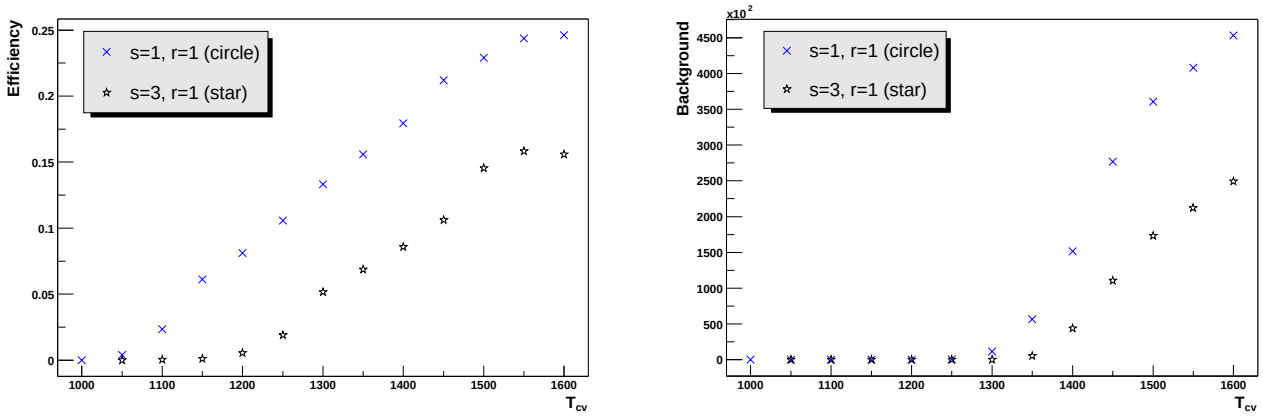

Figure 10. Effi ciency and background vs T_{cv} for $s = 1, 3$ with $r = 1$ (rotation)

There is nearly no difference between efficiency curves in Fig.9 and 10, likewise in Fig.7 and 8. However, a false events rate was reducted by one order of magnitude. Background curves shapes have also changed, see Fig.9.

6.4. Shapes and sizes

Tests showed that the parameter of a shape that dramatically changes the efficiency behaviour is the number of pixels in a shape rather than a shape type. The main conclusions coming from testing of different shapes are described below.

6.4.1. Rejection of big shapes

Shapes were introduced for reduction of false events ratio caused by single-pixels fluctuations. However, reducing the background by increasing the shape size cannot be done endlessly. The efficiency should become low for r bigger than a specific value, where influence of bright pixels outside a pasted star is greater than those of the star.

Tests of the efficiency and the background vs shape's r were performed, with constant T_{cn} and T_{cv}.

A smooth background decrease is observed on Fig.11, with a slightly lower background for a circle. The efficiency curve approaches a constant value below a specific r, then suddenly drops to very low values and then asymptoticaly comes to zero.

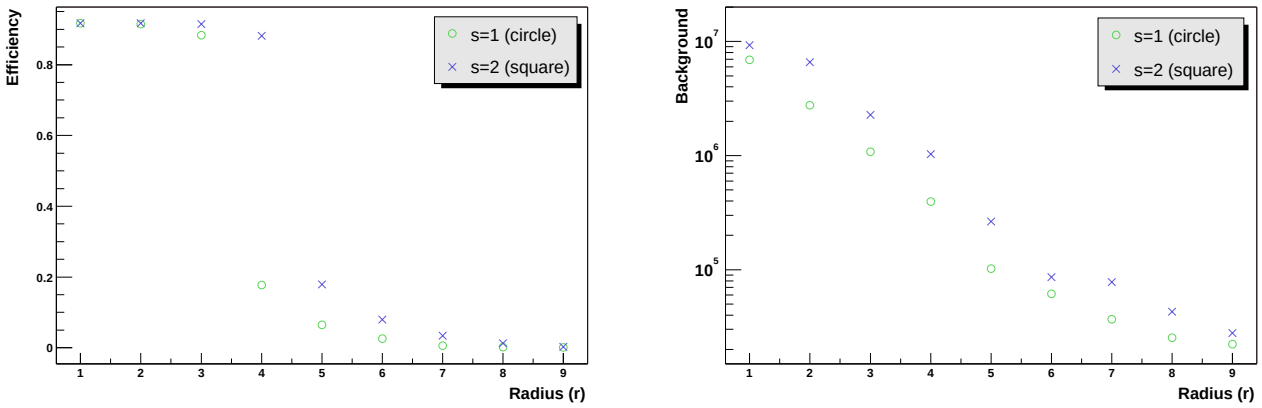

Figure 11. Effi ciency and background vs r_c for $s = 1, 2$ with $r = 1$

Note that despite that a square with $r = 1$ has more pixels than a circle with $r = 1$, the background is higher for the square. If only the pixel number mattered, the background should be higher for the circle. Apparently, the shape type is also important - probably the circle shape is closer to the shape of a real star.

The efficiency behaves as expected. Increasing the shape size moves an average brightness of the shape towards an average brightness of the frame. Probably, the sudden drop of the curve appears when the shape brightness shifts below T_{cn}. However, this phenomenon should be visible only when T_{cn} is higher than the average brightness of the frame.

If T_{cn} is lower than the avearge brightness of the frame, the average brightness of the shape is always higher than T_{cn}. Thus shape size should not affect the efficiency determined by T_{cn}. However, tests not included in this paper show that in this case an efficiency drop is also visible, although not so narrow. Apparently T_{cv} is responsible for this phenomenon. Increasing the shape size means decreasing background gaussian FWHM. It probably escapes $(0, T_{cv})$ range.

7. CONCLUSIONS

Several algorithms of point like flash recognition in CCD images have been tested. Flashes were simulated by inserting star images into real sky images. The star images where cut out from other frames. The efficiency of flash detection and the false detection rate were calculated in each case. This method was found to be very useful for studying various algorithms.

It was found that algorithms based on simple shapes are better than those with single pixels. The optimal shape size is $r = 1$ to 3. The exact type of shape is less important.

The study has shown that the method of frame brightness calculation - homeopatic or maximum from previous frames - is not critical for simulation results. However, very important is the alignment precision taking into account appropriate rotation.

ACKNOWLEDGMENTS

The authors acknowledge a support by the Polish Committee for Scientific Research under grant 2 P03B 038 25.

REFERENCES

1. M. Cwiok et al., *Search for optical flashes accompanying gamma ray bursts - "π of the Sky" Collaboration*, these proceedings.
2. B. Paczynski, *Optical Flashes Preceding GRBs*, **astro-ph/0108522**, 2001.