Investigation of astrophysical phenomena in short time scales with „Pi of the Sky" apparatus

Marcin Sokolowski
Department of High-Energy Physics
Soltan Institute for Nuclear Studies

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Physics

Supervised by dr hab. Grzegorz Wrochna

Warsaw, 2007

Contents

List of Figures

Introduction

For thousands of years people believed that sky is a constant being. The Universe was believed to be eternal and unchanged. This idea has been questioned by Galileo Galilei who used supernova discovered by Johannes Kepler in 1604 as an argument. Since that time big progress in astronomy was achieved. The Universe is not believed to be eternal and unchanged anymore. Several observations prove that the Universe began in the Big Bang, about 14 billion year ago and this is a well established theory now. The evolution of the Universe is studied as well as evolution of its components. The evolution of stars is quite well understood and it is now well known fact that these objects evolve. The timescale of this evolution in comparison the human life is huge. However, there are many processes in the Universe which occur in much smaller timescales. The first example can be already mentioned process, supernova explosion. These events occur at the end of star life and timescale of the explosion itself stars on the level of seconds. Recent observations suggest that the most violent and spectacular processes in the Universe occur in timescales of seconds. Among these processes the most energetic are Gamma Ray Bursts (hereafter GRB), which were main motivation of this thesis. The GRBs are processes occurring in timescales ranging from 0.01 s to hundreds of seconds. In order to study such rapid processes in optical domain a new approach was required. Such a need triggered development of the "Pi of the Sky" project. Data analysis presented in this thesis used the data collected with the "Pi of the Sky" detector, which was optimized for investigating short timescale astrophysical processes with main focus on GRBs.
The thesis is divided in five chapters. In the first chapter short timescale astrophysical processes are reviewed with main focus on GRB. In the second chapter the "Pi of the Sky" experiment is presented, with several technical solutions

described in details. The third chapter contains description of data analysis, including description of flash recognition algorithms created by author. In the fourth chapter results are presented. The fifth chapter contains the summary and future perspectives.

Chapter 1

Short timescale astrophysical phenomena

For centuries astronomical observations were performed in long timescales. It was mainly due to instrumental limitations. Evolution of stars and the Universe is a very slow process, so for long time it was enough for understanding many astrophysical processes in the Universe. The main limitation was due to detectors, since the middle of XIX century photographic films were used. They didn't allow for short exposures due to very poor quantum efficiency. The next generation of detectors, photomultipliers allowed much better time resolution, however they were limited to observations of single object. Big progress in electronic technologies allowed to introduce new type of detectors Charged Coupled Devices (CCD). This type of detector together with computers gives a powerful tool to the researchers. Using the CCD detectors it is possible to perform astrophysical observations with time resolution of seconds. Except the optical domain big progress was also achieved in the area of X-ray, gamma and particle detectors, which allowed to begin studies of the Universe in other bands. Progress in detectors allowed to simultaneously monitor many objects in short time scales. We now know that there is a number of such fast astrophysical events occurring on very short timescales down to milliseconds. List of interesting astrophysical processes acting in short timescales is long. The most rapid, but still possible to observe are listed below :

- Gamma Ray Bursts (GRB)

- Supernovae stars
- Active Galactic Nuclei (AGN) in particular blasars
- Nova stars
- Flare and other variable stars

It seems that the most violent astrophysical processes occur on short timescales. In most cases optical observations of these fast processes are performed time after the main explosion, when new object is observed on the sky or signal from other bands is detected and distributed to the community. Due to sudden character of these events it is very difficult to catch such event when it is going on. This is a strong motivation for wide field observations. Such observations give a big chance to discover many events with flash-like signature, when suddenly new object appears on the sky. Such signal may be used as trigger signal for other experiments to observe. The "Pi of the Sky" experiment was designed to be a good tool for performing this kind of observations.

1.0.1 Gamma Ray Bursts

The GRBs are the most violent and energetic events known in the Universe. They were first observed in late 60s by american satellites Vela [2]. For many years they were the most mysterious astrophysical processes. The main problems of early GRB researchers were the following :

- Origin - are the GRBs galactic or extra-galactic ? The extra-galactic origin of GRBs would require huge energetics of these processes which was very hard to accept by the researchers.
- If extragalactic, how such amounts of energy can be produced ?
- What is the central engine and progenitor responsible for such kind of explosion ?

A large progress was achieved in 90s due to BATSE instrument on CGRO satellite which detected 2700 GRBs. This allowed to determine the spatial distribution of the GRBs (Fig. 1.1) giving a strong argument for cosmological origin of the GRBs. Another success of the BATSE mission was discovery of 2 classes of short and long GRBs (Fig. 1.2). It seems that these two kinds of GRBs are caused by different processes.

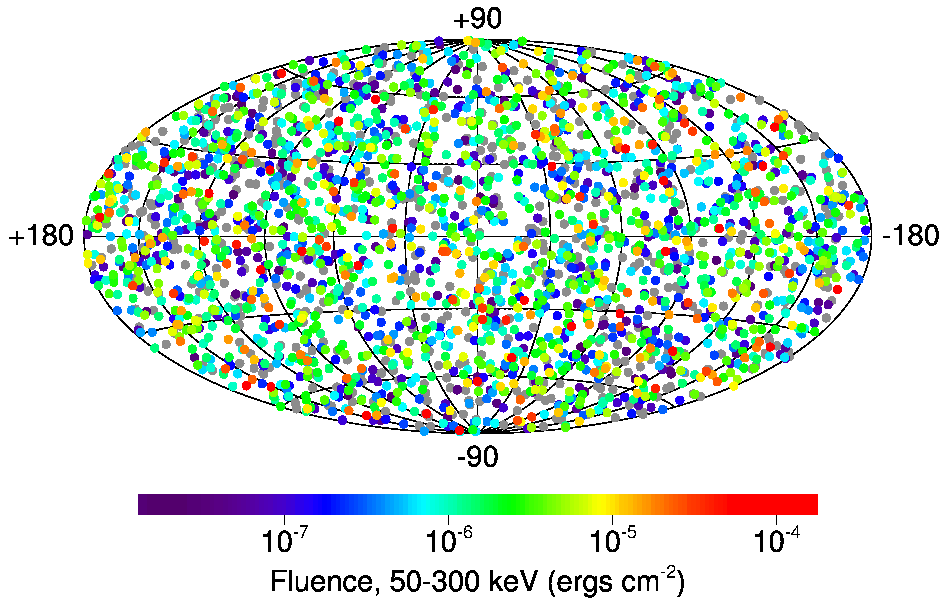

Figure 1.1: Spatial distribution of 2704 GRBs detected by BATSE detector on board the CGRO satellite

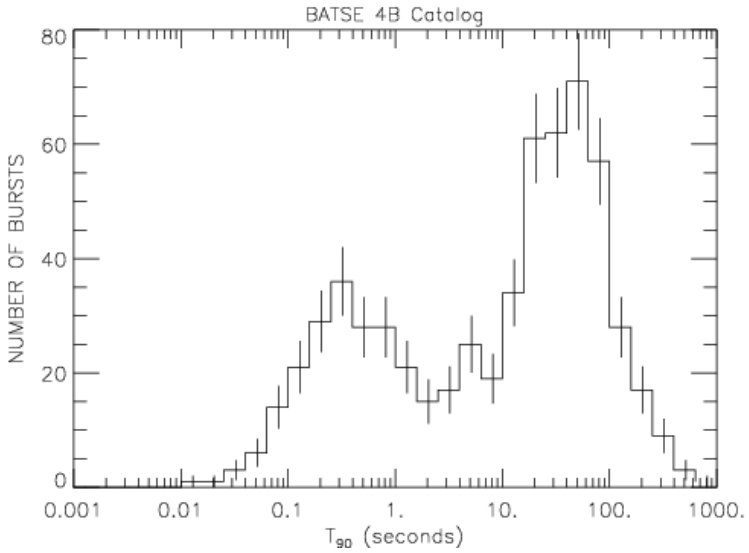

Figure 1.2: Duration of GRBs detected by BATSE detector on board the CGRO satellite

Figures 1.3 and 1.4 show examples of long and short GRBs light curves in the gamma band observed by the SWIFT satellite. Final prove of extragalactic origin of the GRBs was provided by the Beppo-SAX satellite which on 1997-02-28 observed first X-ray afterglow of the GRB970228. The Beppo-SAX observation of GRB971214 allowed determination of its redshift z=3.1418 , which confirmed extragalactic origin of GRBs. The cosmological origin of GRBs was well established after redshifts were measured for many more bursts. This fact implied huge energies produced in the explosion of order of 10^{53} ergs when isotropic emission is assumed.

Orbital missions (Tab. 1.1) and ground base telescopes allowed to formulate the fireball model (Fig. 1.5) which can describe most of the properties of GRBs [3]. The energy is produced by the central engine which in case of long bursts is believed to be collapse of the massive star in the so called "collapsar" scenario and it is a version of supernova explosion. In case of the short bursts the central engine is believed to be a merger scenario which is collision of two compact objects in the binary system. Such compact objects can be two neutron stars or neutron star and a black hole.

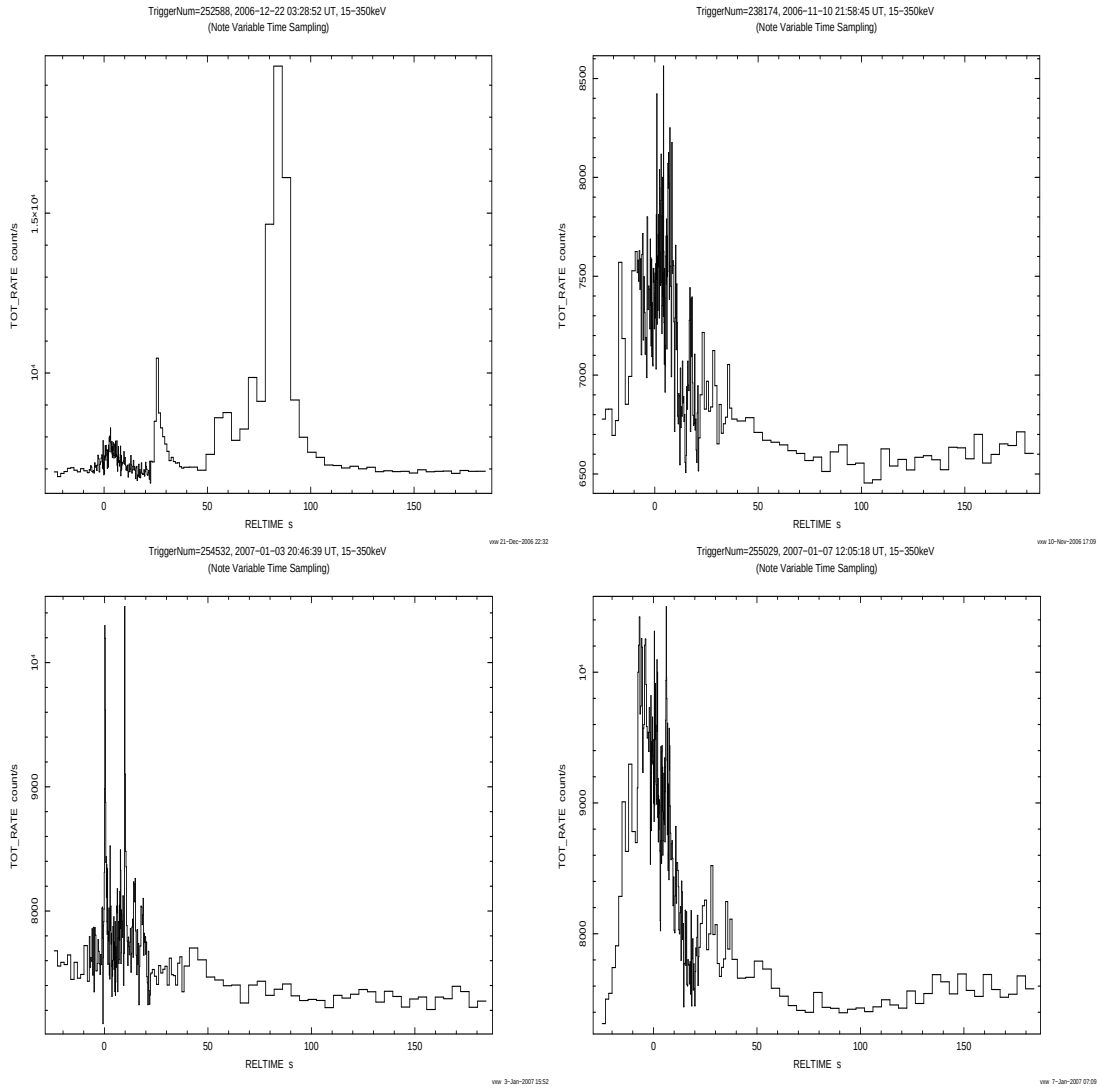

Figure 1.3: Examples of long GRBs observed by SWIFT satellite in years 2006 and 2007

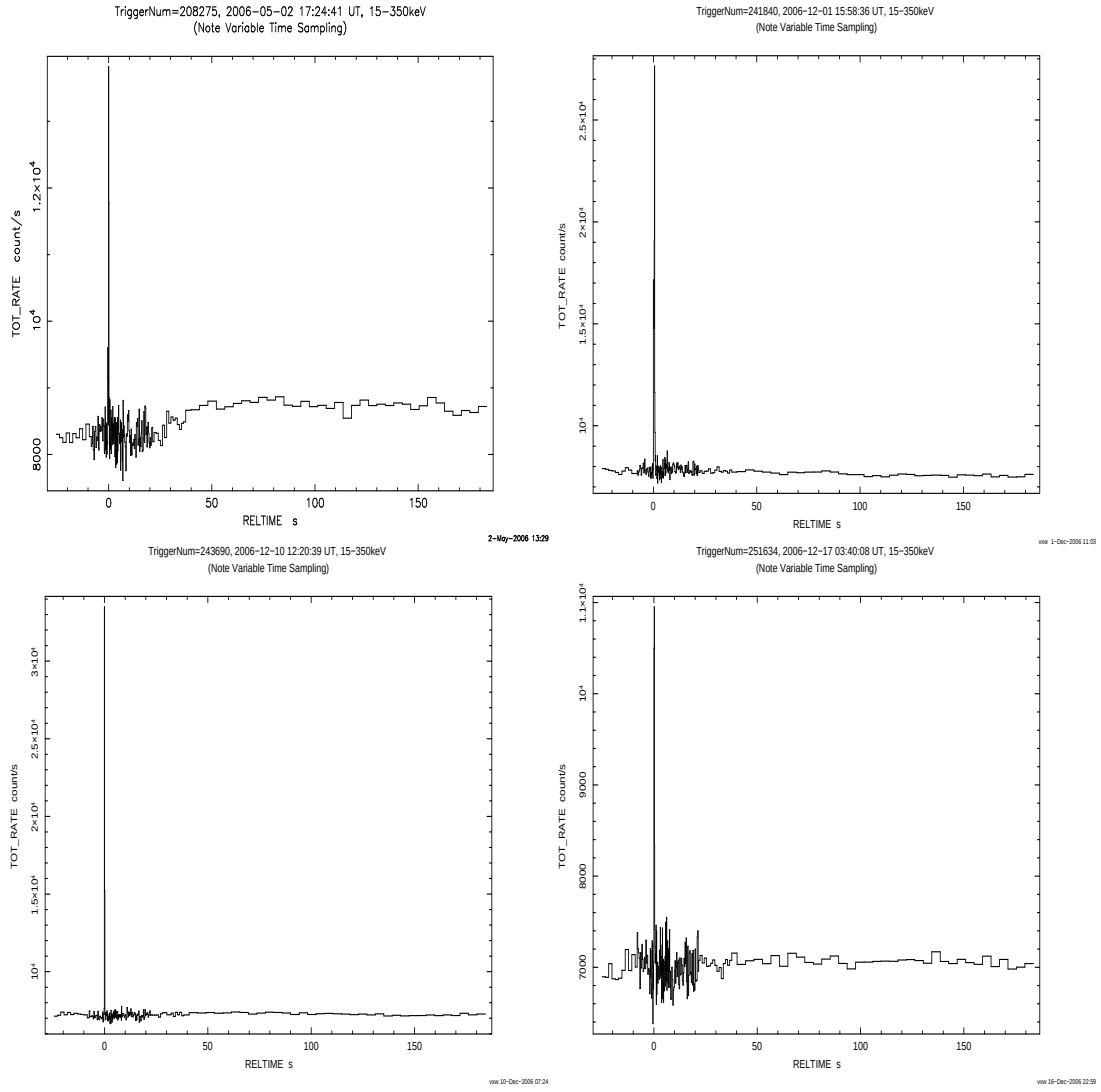

Figure 1.4: Examples of short GRBs observed by SWIFT satellite in years 2006 and 2007

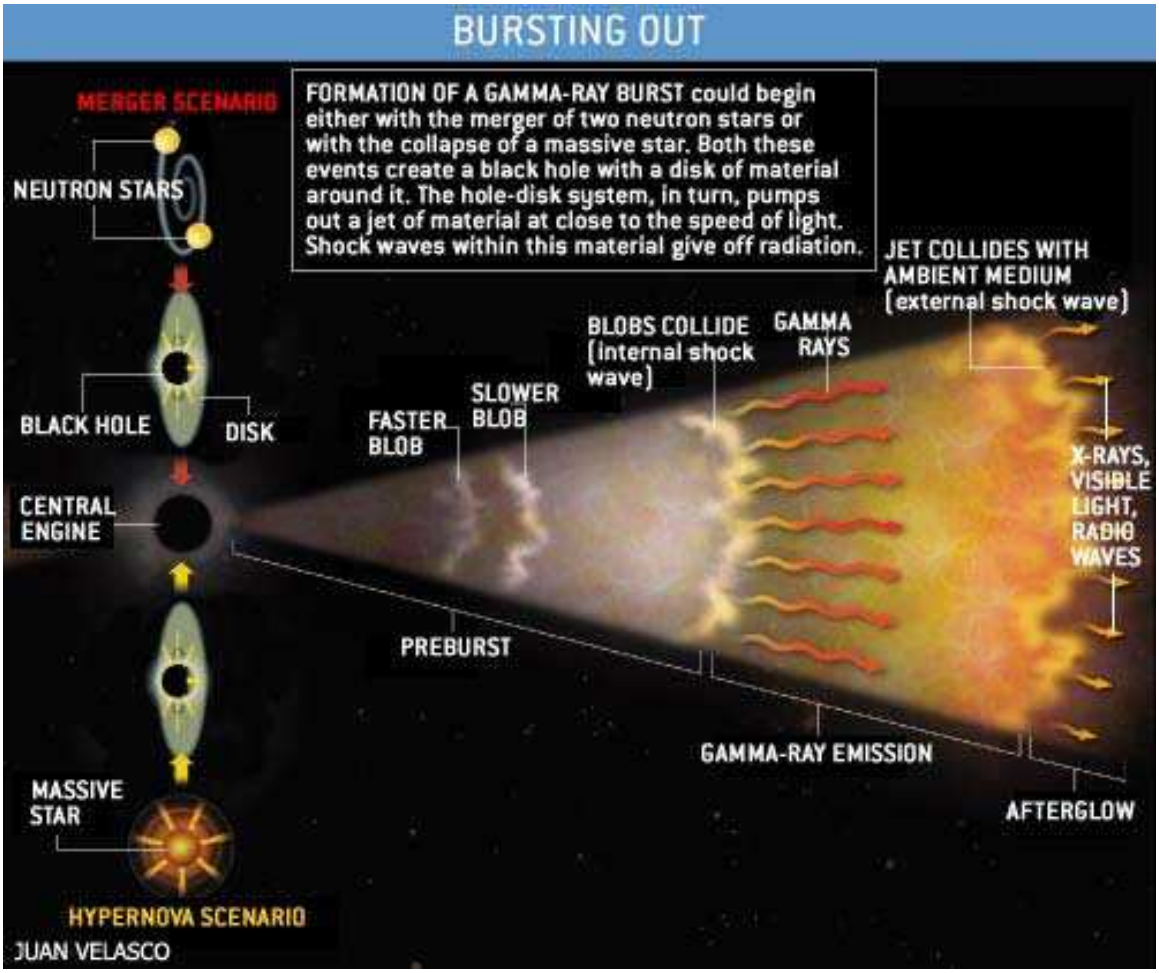

Figure 1.5: Fireball model of the GRB

Mission	Period	Main outcome
VELA	1969-1979	Discovery of GRBs
BATSE on CGRO	1991-1999	Long and short, spatial distribution, detection of 2700 GRBs
Beppo-SAX	1997-2003	X and Optical afterglows
Hete-2	2000-2006	OT from short GRB, 100 bursts detected and localized
Swift	2004-	OT from short GRB, early optical and X-ray observations, X-ray flares
Integral	2002-2010	detection of many GRBs
Agile	April 2007-	Explore 30MeV - 50 GeV regime
GLAST	2007 (?)	Explore 10MeV - >100GeV regime

Table 1.1: Past and future missions most important for understanding the Gamma Ray Bursts

The central engine explosion causes ejection of matter in the collimated jet. The matter is ejected in packets called "shells" with different Lorentz factors $\Gamma >> 100$ causing internal shocks when slower shells are crashed by faster shells ejected later. This internal shocks mechanism is believed to produce the gamma rays observed as the GRB event. The ejected matter finally reaches the interstellar medium (ISM) and the collision with ISM causes the external shock. This processes is believed to be responsible for the optical counterparts of the GRBs so called "afterglows" which were observed for a number of bursts. The jet opening angle was determined from the jet break time to be few degrees, which reduces the total GRB energy by factor of 100. Since the discovery of the GRBs a large progress in their understanding has been achieved, however, the main question how the central engine works still remains unanswered. There are also many other uncertainties and doubts concerning mechanisms of jet production etc. More multiwavelength data is required to understand better these processes. Especially optical data is very important for understanding the puzzle of the GRB. Due to technical limitations for many years optical counterparts of GRBs

were observed many hours or even days after the gamma emission. There was only one event [4] when the optical signal was observed when the GRB was still going on (Fig. 1.6). For most of the bursts position, if known, was determined long time after the GRB, causing optical observations to be delayed.

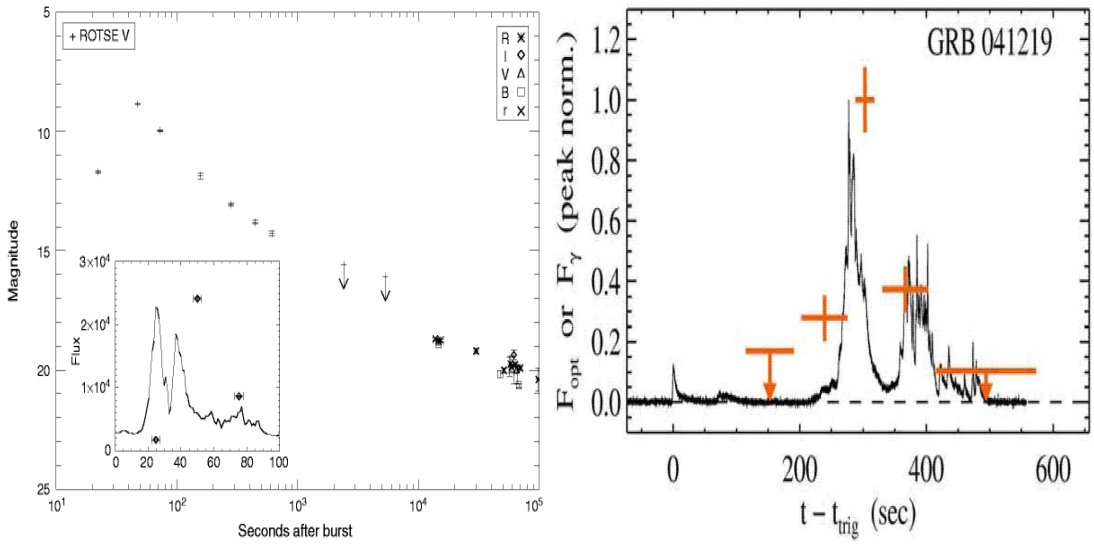

Figure 1.6: Observation of prompt optical signal from GRB990123 by ROTSE (left image) and GRB041219 by RAPTOR. In case of GRB041219 the optical signal variability is correlated with γ emission while in the case of GRB990123 it is not

Improvement of detection techniques allowed to measure the GRB position just after the gamma detection. In order to rapidly distribute alerts with burst position and time , the Gamma ray bursts Coordinates Network (GCN) was developed (Fig. 1.7). The system works in such a way that in case GRB is detected by any of the satellites and its position is determined, it is distributed among the registered observatories. They point their telescopes to the burst position and look for signal in other bands. Big immovable detectors of other messengers like neutrinos or gravitational waves try to find signal correlated in time and space with GRB, so far such correlation was not found.

New era of optical and X-ray GRB research was opened by the launch of the SWIFT satellite on November 20, 2004.

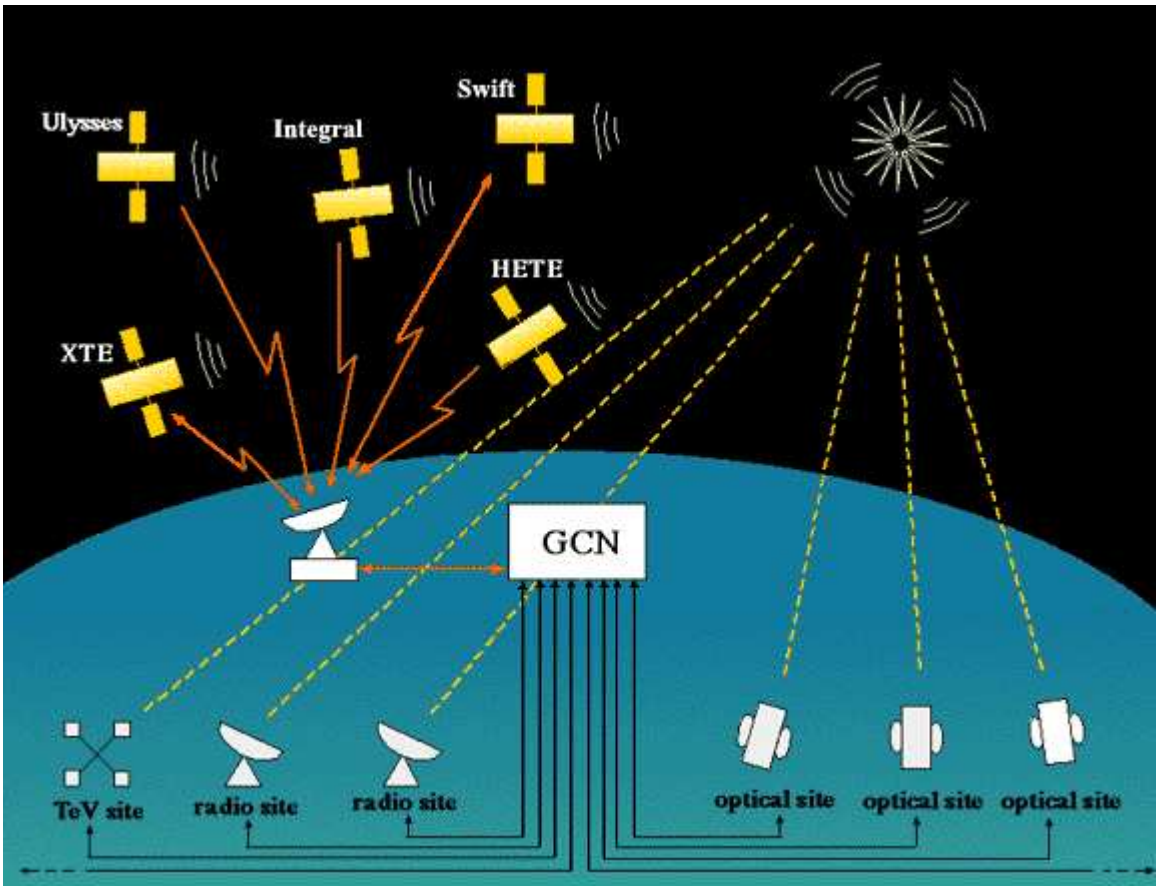

Figure 1.7: The Gamma ray bursts Coordinates Network

1.0.2 Optical counterparts in SWIFT era

The SWIFT satellite is dedicated to GRB research [5]. It has three scientific instruments on board (Fig. 1.8), which are :

- Burst Alert Telescope (BAT) - gamma detector working in energy band 15-150 keV. This instrument detects GRBs and uses coded mask technique to determine their positions with with the accuracy of 2-3 arc min. It has large Field Of View (FOV) ≈2 steradians.
- X-Ray Telescope (XRT) - X-ray detector working in energy band 0.2-10 keV. After the BAT detects the GRB, XRT is able to determine the position of the X-ray counterpart with the precision of 3 arc sec.
- UV/Optical Telescope (UVOT)- this is the optical telescope. After the position is determined by BAT and XRT the satellite slews to the burst location and optical observations are performed. The limiting magnitude of the UVOT is 20^m which allows to detect even very faint objects. The reaction time of UVOT is limited by the slewing time and is on average 40-100 sec.

The SWIFT satellite allowed to access the "dark area" of GRB early optical and X-ray data. It became possible to observe X-ray and optical counterparts just after the burst. Using XRT is was possible to discover previously not observed early X-ray flares. The fast position determination allowed UVOT, but also ground base telescopes to observe optical signal seconds after the gamma emission. It became clear that not all GRBs have bright optical counterpart. Since the launch till this moment (Feb 2007) SWIFT detected over 240 GRBs and for less then half of them optical counterpart was observed (Tab. 4.10). Early bright optical detections are listed in Table 4.11, it is clear that only small fraction of GRBs have very bright optical counterpart. Thanks to SWIFT (and also HETE) it was also possible to observe first optical counterparts of the short GRBs which were previously undetected. These counterparts were localized in the old elliptical galaxies which strongly supports merger scenario expected to occur in

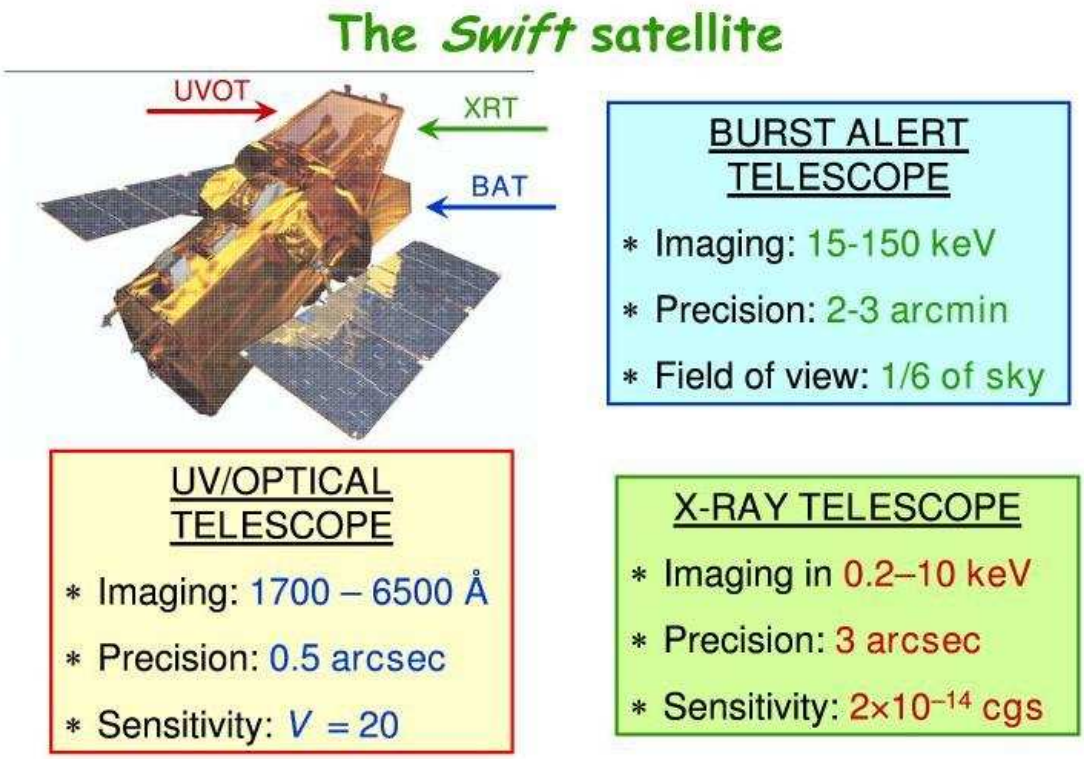

Figure 1.8: Main scientific instruments on board the SWIFT satellite

old binary systems. There are currently handful of long GRBs for which optical counterpart was observed during the GRB itself, the earliest and most bright are listed in Table 1.2.

GRB	**Telescope**	**Reaction Time [sec]**	**Max Magnitude**	**Description**
990123	ROTSE [4]	25	9	Optical lightcurve uncorrelated with γ lightcurve
041219A	RAPTOR [6]	<0	18.6	Optical lightcurve correlated with γ lightcurve, triggered by precursor
060111B	TAROT [7]	30	13.75	-
061007A	ROTSE	26.4	13.6	-
060904B	TAROT	23.1	15.8	-
051111A	ROTSE	26.9	13	-

Table 1.2: Observations of optical lightcurves of GRBs during the γ emission

However, the lightcurves of these events have different properties. For example the optical lightcurve of GRB990123 has no correlation with the gamma emission and lightcurve of the GRB041219 is apparently correlated with the gamma emission (Fig. 1.6).

This suggests that the optical emission mechanism can be different in these bursts, in the first case it is believed to be caused by external shock on ISM while in the second case it is believed to be caused by internal shocks. Moreover there are models suggesting that in some cases optical emission may precede the γ-ray signal or can even be stronger [8]. More early data is needed to resolve these doubts. Current experiments have very limited chances to observe GRB in the optical band when it is going on, because time delay due to trigger propagation and pointing of the telescope in best cases limits the reaction time to 30-40 seconds. Another limitation of current experiments is the time resolution, only few ground base telescopes perform observations with a time resolution of the order of seconds. An experiment observing full FoV of the satellite with good temporal resolution would greatly contribute to resolving the GRB puzzle [8].

The "Pi of the Sky" experiment is dedicated to observe GRBs when they are going on.

1.0.2.1 Orphan afterglows

The jet structure of the GRB explosion implies that only small fraction of the total number of the GRBs in the Universe can be observed from the Earth. Another consequence is the so called "jet break" effect observed in the optical lightcurves of many GRB (Fig. 1.10). This effect is related to relativistic beaming of the emitted radiation, the radiation is emitted by relativistic matter causing collimation. Thus only radiation emitted by the core of the jet is visible on Earth. When matter reaches the Interstellar Medium (ISM) is is slowed down and collimation becomes smaller, allowing observation of radiation from outer parts of the jet. Finally matter is slowed down so much that collimation is of order of magnitude of the jet opening angle Θ_{JET}, since this moment observer sees radiation emitted in all parts of the cone. This causes faster decay of the signal, because amount of light is not increased anymore due to visibility of larger part of the cone (Fig. 1.9). It appears as "jet break" on the optical lightcurve. Since radiation emitted in this late times is observed in optical band it is expected that more optical afterglows should be observed then the GRBs. Such kind of afterglows without the GRB event are called "orphan afterglows", they would appear as optical flashes without corresponding GRB event. One of the results presented in this thesis is the upper limit for rate of the orphan afterglows.

1.0.3 Other fast astrophysical processes

Besides described in detail GRBs there is a variety of other astrophysical processes manifesting themselves in time scales of order of seconds up to days. It is impossible to describe all of them in detail, however a short description of those which are possible to investigate with the "Pi of the Sky " apparatus will be given :

- Nova explosions - these processes occur in binary systems with white dwarf accreting matter from a companion star. The matter accumulates on the

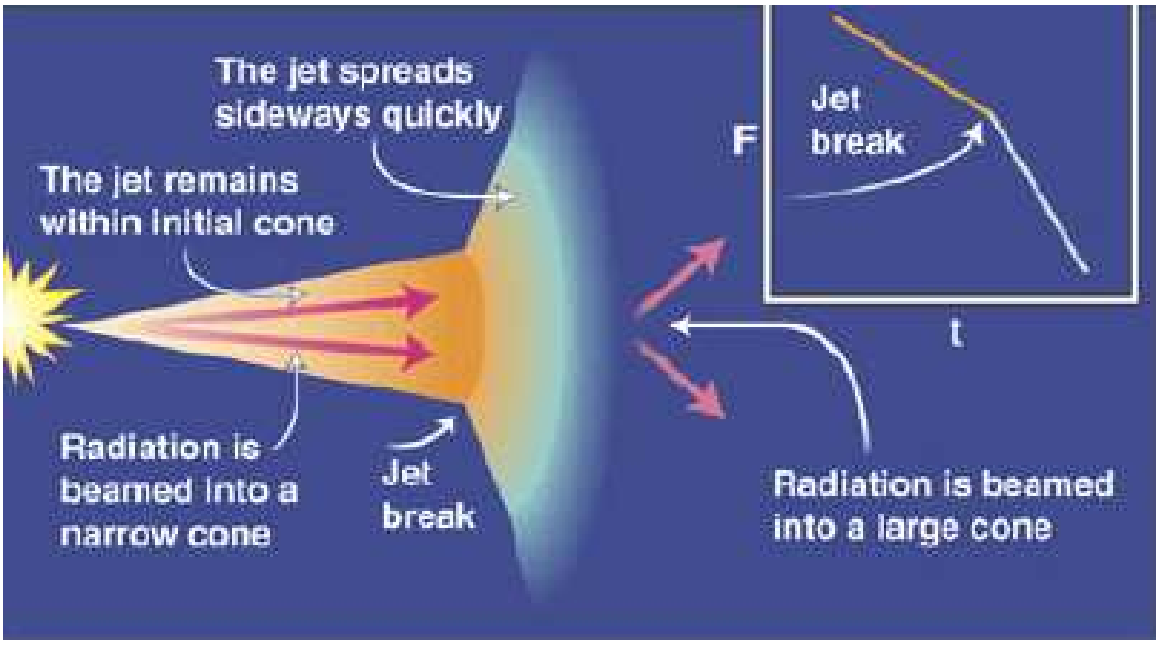

Figure 1.9: The mechanism of the jet break caused by slow down of relativistic matter on the ISM

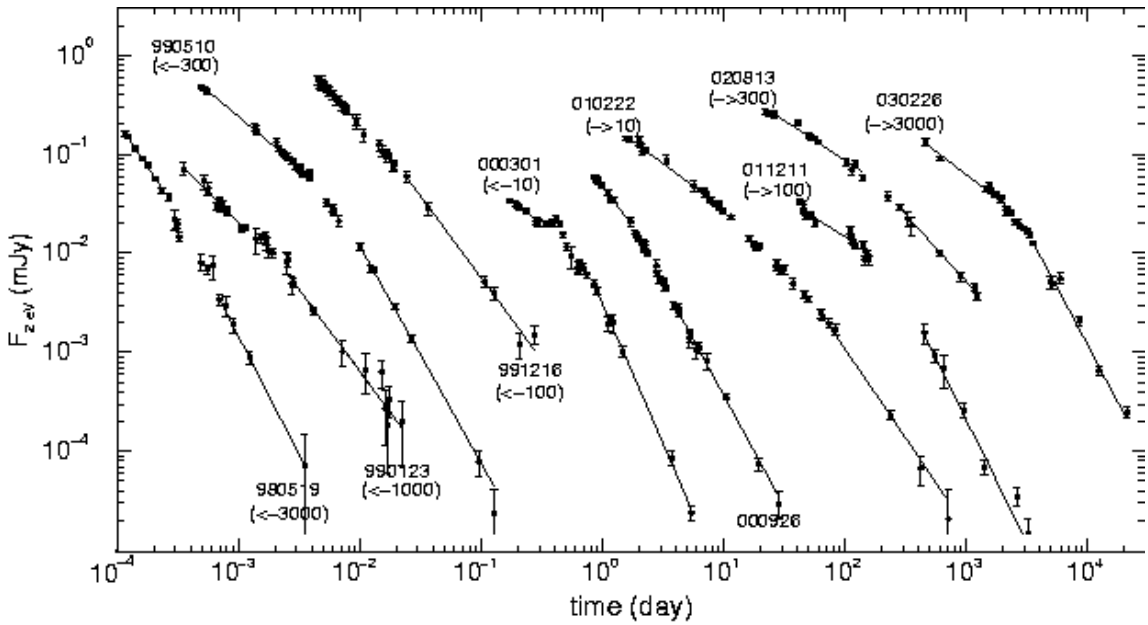

Figure 1.10: Optical light-curves for 10 GRB afterglows with breaks and power-law fits to the pre- and post-break emission (image from [1])

surface of the white dwarf and when certain critical mass is reached a thermonuclear explosion occurs. This process may repeat when enough matter is accreted again, these kind of novae are called *recurrent novae*. A very comprehensive introduction to subject of novae explosions is given in [9]. Depending on the range of the telescope novae explosions signature is sudden increase of brightness of the system or appearance of new object (in case normally it is below limiting magnitude of the telescope), so this is very much a flash-like signature.

- Variable stars - about 1% of all stars are variable stars. The time scale of brightness variations ranges from years and days down to milliseconds. The variability mechanisms can be geometrical (e.g. eclipsing binaries) or due to internal properties of the star (e.g. pulsations). A particular type of variable stars are flare stars (also known as UV Ceti variables), which undergo sudden and unpredictable explosions related to release of magnetic energy. The mechanism is the same as in the case of Solar flares, however, flar stars explosions can be even thousand times brighter. The brightness increase can be as large as 100-1000 times. The variability in the flare stars is characterized as a rapid, irregular large-amplitude increase of brightness, followed by a much slower decay (from minutes to hours).

- AGNs and particularly blasars - these objects are active galactic nuclei which are as far as we known powered by accretion of matter on central super massive black holes. In many cases jet of relativistic matter is observed, it is ejected in the direction of spin of the black hole and in the cases it is pointing towards the Earth, such AGN is called blasar. AGNs manifest rapid (time scales of days or even hours) brightness variations in all wavelengths. Monitoring of such objects and alerting on the begin of activity is very important for observations by big telescopes and can be realized by small wide field telescopes.

- Supernovae - these processes are related to death of massive stars. When the thermonuclear reactions in the core are not able to balance gravitational pressure the star collapses and huge explosion occurs. There are over 400

supernovae discovered per year, but they are extragalactic supernovae. This means they are very dim and mostly below the range of the telescopes like "Pi of the Sky", however some of them could also be discovered by a system like this.

The above list is just an indication of what kind of processes are aim of analysis described in this thesis. In most cases the main goal of the analysis is a discovery of an object and sending alert to larger telescopes as the "Pi of the sky" system is not able of performing spectroscopic measurements. However, in many cases good time resolution of the system gives a possibility of investigating brightness variations in time.

Chapter 2

The Pi of the Sky Experiment

2.1 General Idea

In order to study rapidly varying astrophysical objects a telescope with time scale resolution at least of the order of seconds is needed. Exposures and dead time between images must be short. This will allow to investigate light curve structure of rapidly varying objects. However this is not enough to study short and unpredictable processes like optical flashes related to GRB or any other kind of optical flashes. In such cases it is impossible to predict where and when an event will occur. Thus it is not possible to point the telescope at certain star and wait for an event to occur. In order to be able to observe such class of processes when they are going on a large field of view (FOV) is required. If the telescope system will observe the whole sky continuously, then any outburst will already be in its FOV. There will be no delay, due to telescope movement and trigger information distribution. The area of interest will be observed continuously. The price for this is the range, the more sky telescope observes the less sensitive it is to faint objects.

$$m_{MAX} = m_0 + 5 \cdot log\{\frac{p \cdot N}{170.45 \cdot S \cdot FOV}\} \tag{2.1}$$

Where m_0 is limiting magnitude of telescope with given detector and aperture of 1cm, p is pixel size in μm, N is chip size in pixels, S is light power and FOV is field of view in degrees. For human eye detector $m_0 = 7.5^m$ and using FOV=21° $m_{MAX} \approx 11.8^m$ which is close to limiting magnitude of camera with Cannon

f=85mm f/D=1.2. However for precise result m_0 should be determined from the specific CCD camera properties. It can be clearly seen that limiting magnitude decreases with field of view. One needs to find a compromise between FOV of the single telescope and the maximum limiting magnitude it will be able to observe. It is of course impossible to build a single camera which would cover the whole celestial sphere at once with satisfying limiting magnitude. The solution for this is to build a system of many telescopes, each covering a fraction of sky and pointing in different direction. The range of such a system will be limited by the range of single telescope, in principle it would be possible to build a farm of large telescopes with high limiting magnitude, but the cost would be huge. One has to find another compromise between limiting magnitude (number and size of telescopes) and the price. The "Pi of the Sky" system was designed to cover significant fraction of the sky with the satisfying limiting magnitude on the level of 14-15^m. Data analysis presented in this thesis was done on data collected with the prototype of the full system. In the next section this prototype will be described in detail. The full detector design will be presented in Section 2.3.

2.2 The Prototype

The prototype was build to test components of the final system including hardware and software solutions. One of the goals was to probe efficiency of optical flashes detection and background rejection. The prototype was installed in June 2004 in the Las Campanas Observatory (LCO) in Chile and it is working until now with a few months break for upgrade. The prototype after upgrade is shown in Figure 2.1. The system in LCO operates automatically and is fully controllable via the Internet. There is no person in LCO who takes care of the system. It is very important requirement for both the prototype and the full system that it must be remotely controlled and failure proof. The current setup of the prototype allows even some of the hardware failures to be handled remotely and continue operation of the detector. The maintenance trips to LCO are very expensive so the need for them must be minimized. Current experience says that one maintenance trip a year should be enough.

Figure 2.1: Prototype in LCO, two cameras on single mount in ASAS dome

2.2.1 Hardware

The detector consists of two cameras on a single parallactic mount (Fig. 2.1). The mount was adopted from the ASAS experiment. It is driven in two axis by the step motors. A TMCM 300 microcontroller driver by Trinamic was used to control mount from the PC [10]. The microcontroller box is connected to PC with the RS-232 interface. On the PC side mount driver program controls settings and movements of the mount. The position of the mount can be calculated from number of steps executed by the step motors. The resolution of step motors is $\Delta\lambda \approx 3.5$" and $\Delta\delta \approx 112.5$", where ($\lambda$,$\delta$) are equatorial coordinates right ascension and declination. As a cross-check potentiometers were added, they give rough estimate of the position with the accuracy of $\Delta\alpha \approx 0.2^{\circ}$, this allows to detect step motors position errors (for example due to slip of belt drive). During observations mount performs tracking which is a rotation around earth axis compensating for the Earth daily rotation.
The cameras are custom designed. The development of own CCD cameras was motivated by several reasons. The first factor is the price of commercial products available on the market. However there are other reasons which limit possibility of using commercial products in such kind of remotely controlled experiment :

- reliability of internal mechanical shutter
- remote control of camera settings which include lens heating and lens focus adjustments
- temperature and humidity sensors
- fast readout (in most of the solutions limited at 1MHz)
- possibility of changing firmware of the camera remotely

The cameras are based on Fairchild CCD 442A chip [11] with resolution 2048x2048 pixels (15 μ x15 μ each). The CCD sensor is placed in a separate chamber filled with a noble gas. The 16bit analog digital converter (AD9826) has been used. Communication with the PC is realized by USB2.0 interface by the Cypress FX2 USB CYC68013 chip. Camera is controlled by the FPGA Altera

chip. CCD chips are cooled with the thermoelectric Peltier junction, down to 30° C below ambient temperature. The following features have been implemented in cameras :

- Remote control of almost all functions : readout frequency, gain, CCD temperature, mechanical shutter, lens focus regulation, lens heating (against water condensation)
- Remote monitoring of atmospheric conditions (temperature and humidity sensors)
- Possibility of remote firmware upgrade (Cypress microcontroller program and FPGA configuration)
- Watchdog Timer which resets camera in case communication with PC is lost

The table below summarizes the parameters of the cameras :

Parameter	**Value**
Readout Time	1s - 1min
Readout noise	$<16e^-$ at 2MHz and $<12e^-$ at 1MHz
Shutter Durability	$>10^7$ cycles
USB2.0 max transfer speed	52MB/s
Maximal cooling	30° below ambient temperature

Table 2.1: CCD cameras parameters

More details on design of the cameras can be found in [12]. The cameras are equipped with CANON EF f=85, f/d=1.2 lenses. Short time exposures (5-10 sec) imply many ($\approx 2000 - 4000$) exposures during a single night. This results in $\approx 10^6$ images per year. Huge number of collected images required special design of the shutter which can survive more then million cycles (commercial shutters usually work up to 10^5 cycles). In order to save shutter it is possible to make exposures with shutter permanently opened. The mount with cameras is installed in the ASAS [13] dome in the LCO (Fig. 2.2). The dome is controlled by the

OGLE telescope system ([14]), it opens when the OGLE telescope dome opens which depends on observer's decision motivated by the weather conditions. The mount and the cameras are controlled by the single PC computer which is placed in the lower part of the dome (Fig. 2.3). Custom driver was developed to control parameters of the cameras and collect images, it will be described in the next section. The system contains two other PC computers which are used for off-line data analysis.

Figure 2.2: Dome of the ASAS experiment in LCO

2.2.2 Software

2.2.2.1 Overview of the system components

The main "Pi of the Sky" software components are running under Linux (Fedora) operating system. Most of the software used in the experiment was custom developed. Some of the procedures and formats were adopted from the ASAS experiment [13]. Generally software can be divided into on-line part, which takes care of detector control during nightly data aquisition and off-line part which performs off-line data analysis.

Figure 2.3: Pi2 computer controlling nightly data aquistion is located in the lower part of the ASAS dome

General architecture of the night control system is presented in Figure 2.4. The system consists of several modules which communicate with each other by the CORBA interface [15]. They were designed in the client server architecture. The main components of the system are :

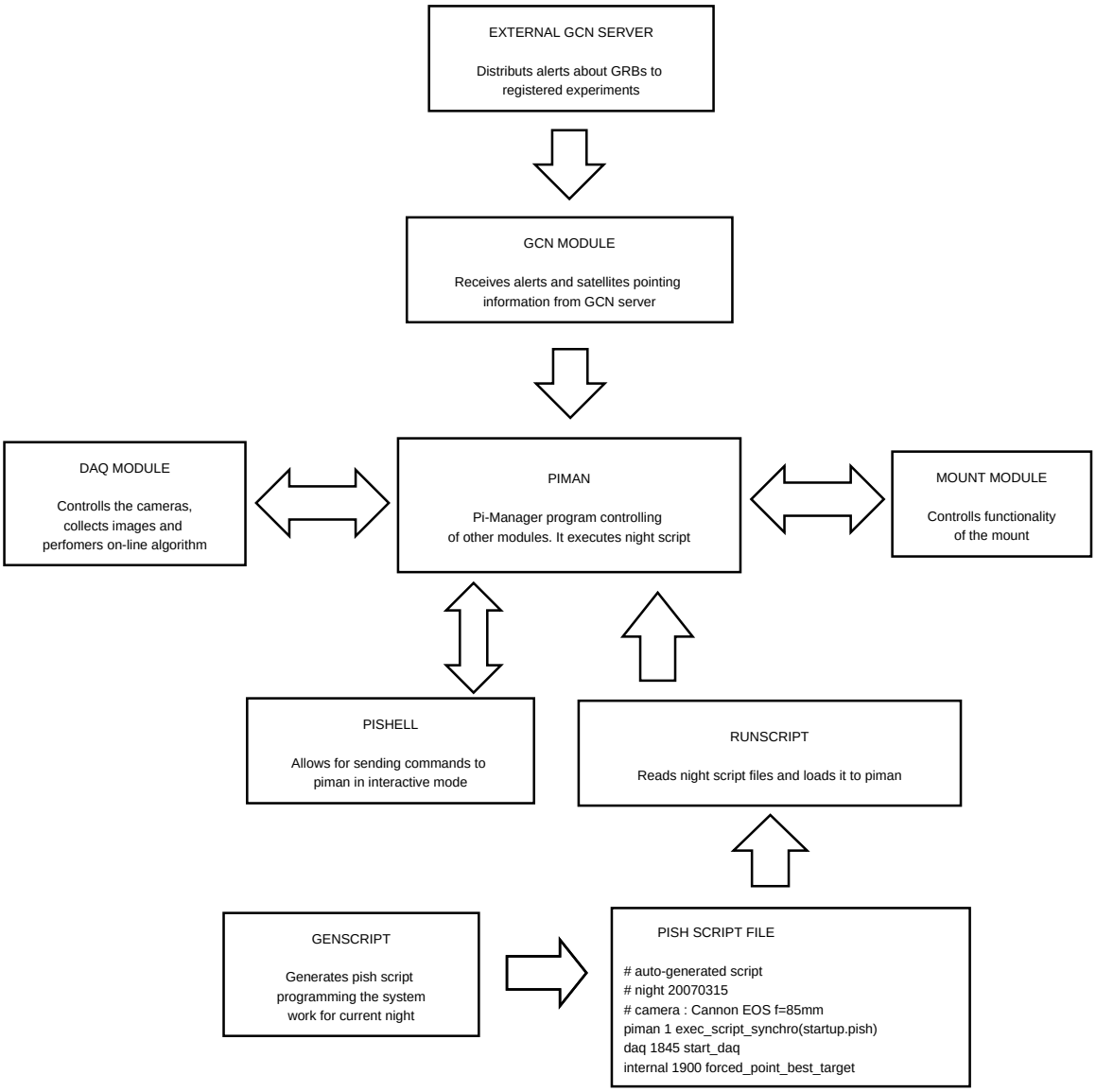

Figure 2.4: General architecture of system night control implemented in the prototype

- **PIMAN**
  It is the main manager program which controls the whole system. This program provides set of commands which can be executed by different modules. There are also complex commands, which use several different modules. In this case result of command executed by one module is used as an input to command for another module. Commands can be executed manually from `pishell` program or loaded in form of the `pish` script by `runscript` program. Example of the night script is given in Figure A.5. It is a set of commands with times at which they should be executed. Script is generated every evening by the `genscript` program. After system is started the `runscript` program is executed to read the night script and sent it to the `piman` program. The `piman` program executes commands at specified times. In case GCN alert is received by the GCN module the alert information is passed to `piman` and handled.

- **PISHELL**
  This `pishell` program is a client program which has interactive command line interface. User can launch this program to sent commands to `piman` in interactive mode. It allows modification of night scheduled program loaded from the `pish` script. All necessary actions like checking stack of commands, deleting commands and adding new commands can be executed by using this program.

- **RUNSCRIPT**
  This is a simple client program for reading given `pish` script and sending it to `piman` program.

- **DAQ**
  This program is responsible for controlling cameras settings, collecting images and on-line data analysis. On-line data analysis consists of two main actions : astrometry and flash recognition algorithm. In general astrometry is a transformation of chip (x,y) to celestial (λ,δ) coordinates. This transformation allows to calculate coordinates of image center. They can be compared with the expected coordinates and mount movement corrections

can be determined. On-line flash recognition algorithms looks for optical flashes on single 10 sec exposures. The more detailed description of the `DAQ` module will be given in next section.

- **MOUNT**
  This module is responsible for controlling the mount. All functionality of the mount hardware can be controlled by this program. This includes calibration, moves to the desired position, tracking enabling/disabling, changing the tracking speeds and calibration of the pointing using exact information from astrometry (`DAQ` module) or encoders (potentiometers in the prototype). The core of this module is compiled in form of library `libmount.so`. There are two programs using this library. Program `monit` allows to control the mount in the interactive mode, while `mount_server` is a server which provides a set of functions which can be executed by `piman` when performing commands from a night script.

- **GCN** This module is waiting for GCN alert messages sent through software socket[1] from the remote GCN server (Fig. 1.7). During the night every package obtained from the GCN server is passed to `piman` module which decides weather it is an interesting event which is possible to be observed by the "Pi of the Sky" system. In such case alert procedure is executed (Sec. 2.2.2.7). The `gcn` module is also responsible for writing satellite pointing information to log files (`swift_pointdir.log` and `integral_pointdir.log`). This information is used by pointing procedure to follow FOV of the GRB detecting satellites (Sec. 2.2.2.7).

- **GENSCRIPT** This program is responsible for preparing a plan of night observations. It is generated in form of `pish` script (Fig. A.5) which is loaded to the `piman` program memory and executed during the night. The syntax of the single command is the following :
  MODULE TIME COMMAND(parameters)
  The module name `internal` is used for complex commands (see above).

[1] socket - term for type of interprocess communication

More detailed description of the script generator module will be given in separate subsection 2.2.2.7.

Every module writes its current status to the specified status file. They also produce separate log files and most important information is saved to global "Pi of the Sky" system log file (Tab. A.4). The relations between pi modules and external systems are presented in Figure 2.4. Communication between modules is client-server architecture and is implemented in CORBA technology [15] which is efficient and reliable method of object oriented interprocess communication. Most of modules were written in C++ and C.

2.2.2.2 Data Acquisition System

The data acquisition system (`DAQ`) is a program which is responsible for collecting images from the cameras and for on-line data analysis. The program uses several custom developed libraries which provide different functionality. The most important libraries are listed in Table A.1 in Appendix A.1.

This libraries are linked by several programs, most important programs for data collection and analysis are listed in Table A.2. Programs which are used every night for data acquisition are `ccdsingle` and `ccddouble`. The first one is used when `DAQ` uses single camera to collect data. The program `ccddouble` is used to collect images from 2 cameras on the same mount working in coincidence. The main parts of the collection program are shown on diagram 2.5. The only difference in `ccdsingle` program is that images are collected from single camera and the algorithm for flash recognition is different.

The most important task of these programs is to collect images from the cameras and save them to a disk. There are also actions important for other modules in the system, most important one is astrometry (Sec. 3.3.1.1). Generally astrometry is a procedure for transforming (x,y) coordinates of objects on CCD chip to equatorial coordinates (λ,δ). More details on astrometry procedure can be found in Section 3.3.1.1. Astrometry is very important for finding real coordinates of image center which can be slightly different then values calculated from mount step motors. The position of the image center found by astrometry can

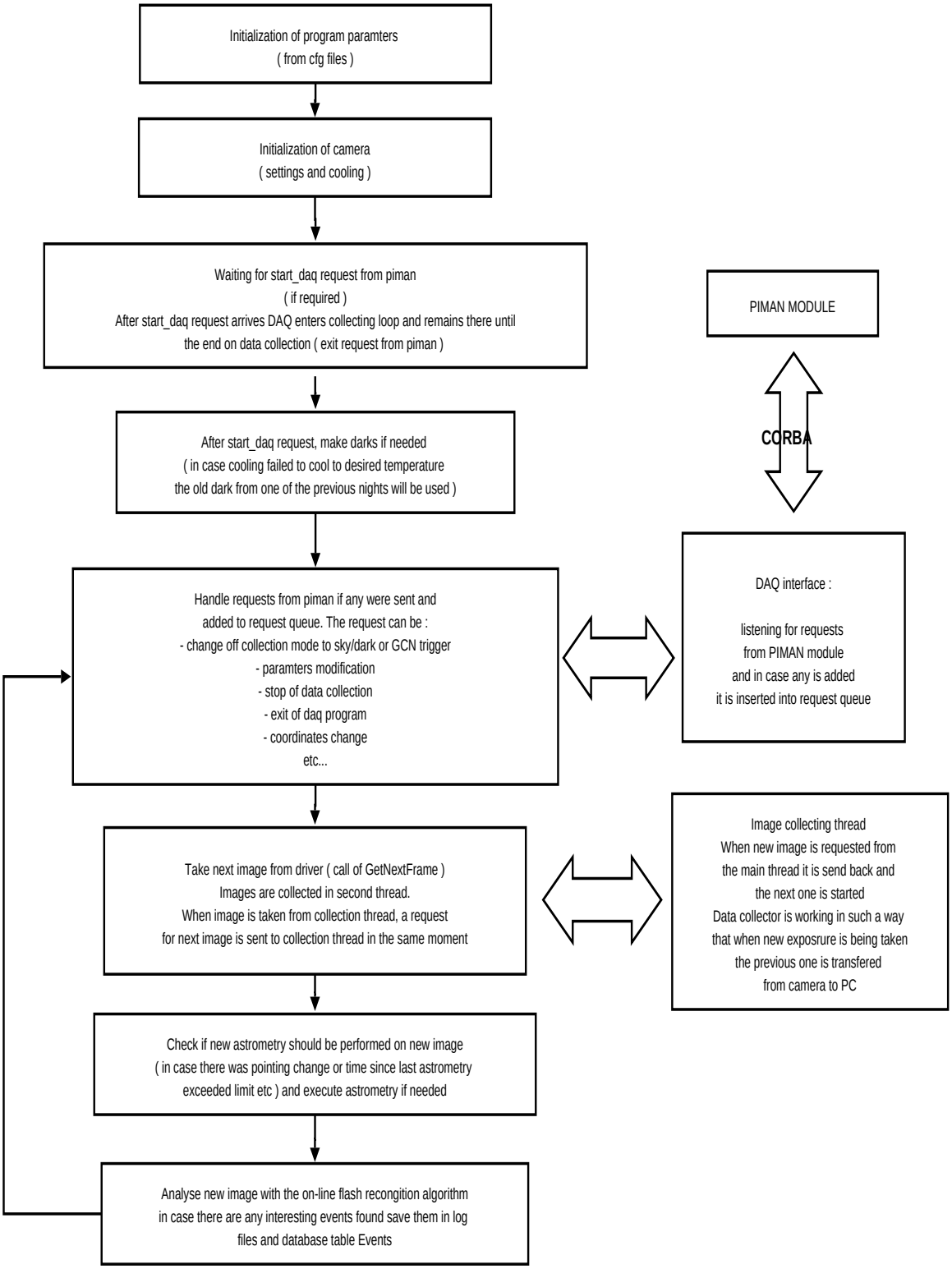

Figure 2.5: Block diagram of the Data AcQuisition program

be verified against position calculated by the `mount` module. In case they differ, position determined from astrometry can be used to correct mount position and also mount tracking speeds in both axis. The third main task of `DAQ` program is to analyze images in search for optical flashes (Sec. 3.2). Interesting flashes found by the algorithm are saved on disk and are almost immediately published on the WWW page in order to be reviewed by human. `DAQ` program exports several functions and acts as CORBA server (Tab. A.5). This functions are executed by `piman` in order to control the whole system. Information from `DAQ` program is stored in night log file (see Tab. A.4 in Appendix A.1) Most important information is stored in the database allowing for fast and easy access for reporting purposes (Sec. 2.2.2.5).

2.2.2.3 System nightly performance

System is started every evening from `crontab`[1] by starting script `run_pisys!`. This script waits until dome is opened and then it launches all system modules like `piman`, `DAQ`, `mount`, `gcn` etc. After all modules are started the night script is loaded to piman's memory and subsequent commands are executed at specified times. Since this moment the system can be controlled from the `pishell`. After `DAQ` is started communication with cameras is initialized and program waits until cameras are cooled to temperature specified in the configuration file. After cameras reach the desired temperature programs waits for command `start_daq`. This command is executed when dark images can be collected. Dark images are images collected in the same conditions as sky images but with closed shutter. This allows to obtain average values of noise and dark current in every pixel. Before analysis of a the sky image the dark image is subtracted from it. Typically 20 dark images are collected and median image is calculated to be used as a dark image in further analysis. After the dark image is ready, `DAQ` program is ready to collect sky images and waits for a command to start analysis. The `piman` module chooses the best object to be observed by launching `point_best_target` command and sends request to `mount` module to move to the position of this target. The `start_analysis` command passes sky position obtained from `mount` module

[1] crontab is a system tool for starting programs at specified dates and times

to DAQ, it is required to perform astrometry. Before any change of the position the DAQ stops collection of images, it is restarted after next start_analysis command is obtained from piman after the desired position if reached. In specified time intervals piman asks DAQ for current position (λ, δ) of image center resulting from astrometry and sends this information to mount in order to correct tracking speeds in both axis. Twice a night the whole sky scan is performed. During the scan DAQ takes images in single image mode without performing astrometry in order to complete the scan as fast as possible. Astrometry for scan images is performed off-line.

2.2.2.4 Camera Driver

The CCD cameras used in the project are custom designed. Due to this fact it was necessary to develop the software to control them. The camera driver was developed in C/C++. The original cameras were equipped with USB2.0 interface only. The driver for USB2.0 cameras consists of two parts. The first part is a kernel module pikam.ko which can be compiled under Linux kernel >= 2.6.5. The second part is the C++ class Device2K2K which opens the file (/dev/usb/pikamN) in order to communicate with the camera device. The final design of the camera was enriched in gigabyte Ethernet interface. The Ethernet camera exports the same set of functionality, so only the low level communication part of the driver has to be optionally replaced. This required development of communication protocol NUDP [16], it is implemented in camera firmware and in PC driver. Dedicated kernel module is not needed in this case. The C++ class CEthCamera using external library Sockets [17] implements communication with camera via Ethernet interface. The class DeviceEth2K2K derived from Device2K2K overwrites several low level communication functions using member class CEthCamera (see Fig. 2.6). New camera gives a possibility of using USB2.0 or Ethernet interface, depending on current user requirements. Change of communication can be easily done by editing configuration files or by command line option to the data acquisition program (see Table A.2 in Appendix A.1). The driver is compiled in form of library libpimandrv.a.

Images are saved to fits files ([18],[19]). Writing of fits files is realized by library libmyfitslib.a, but camera driver library is responsible for preparing

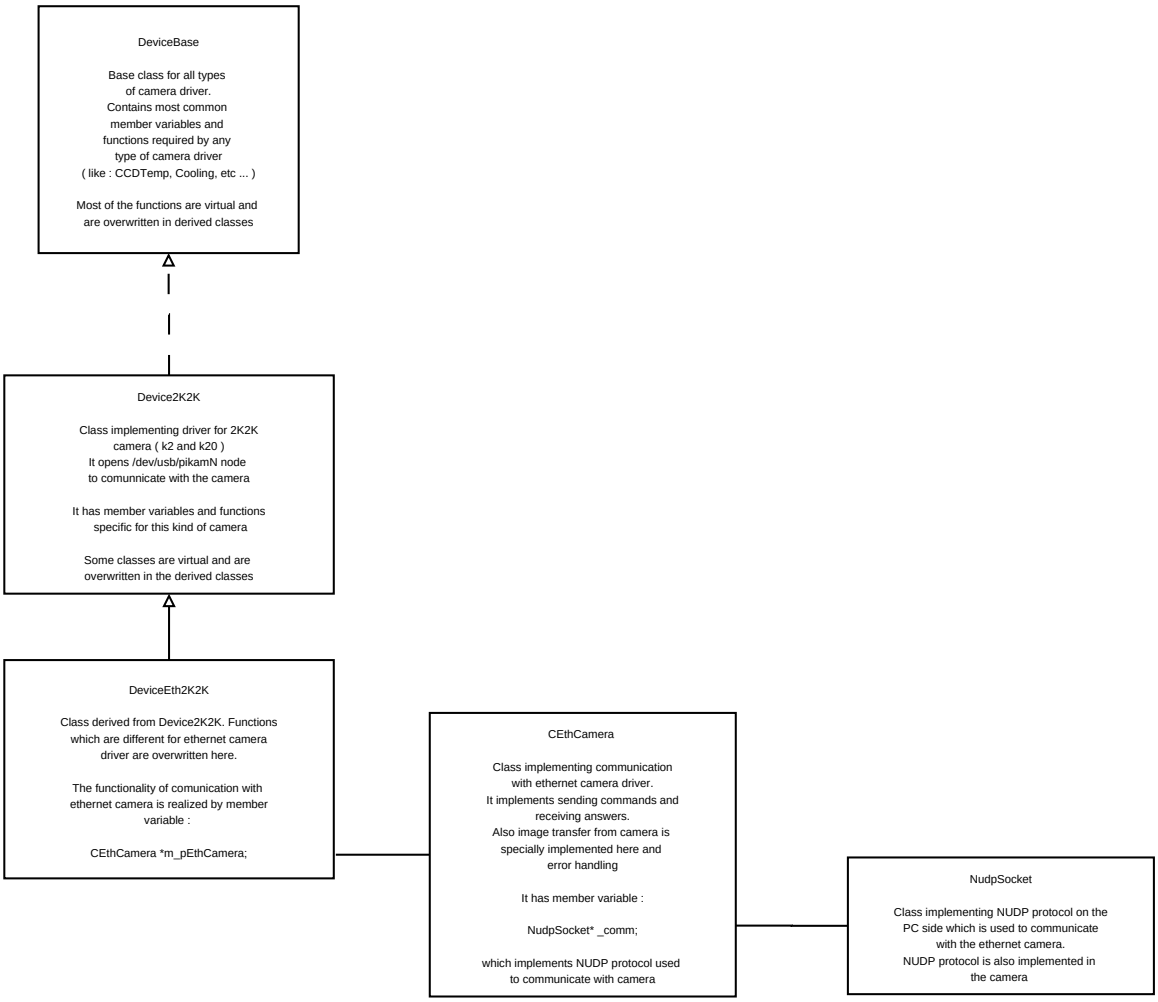

Figure 2.6: Dependencies of camera driver C++ classes

camera settings information to be saved in `fits` file header. The keywords related to camera settings are listed in Table A.3 in Appendix A.1. Most of this information is also written to the database. The driver library is linked by `DAQ` programs (`ccdsingle` and `ccddouble`). The best way to easily test cameras is to use program `test2K2K` (Tab. A.2). It is a simple text interface interactive program which allows to change settings of the camera, to take image and to realize any kind of camera functionality. It can be used to operate camera by the USB2.0 interface or by the Ethernet depending on options passed in the command line. It is also possible to use this program in a batch mode where actions to be performed are passed in the command line.

2.2.2.5 Database for on-line data

In order to store the most important information about system on-line performance a database structure was designed and developed. The PostgreSQL database system is used for this purpose [20]. The structure of the database used for storing on-line system information is shown in Figure 2.7. `DAQ` program stores on-line information in 3 tables :

- FRAME and FRAME_DET tables contain information about every image collected by the system. Generally all information from `fits` header is saved to the database, some additional information is also added.
- EVENT stores information on interesting events detected by the on-line algorithm

The other two tables are used for storing information from `piman` and `mount` modules :

- PIMANCOMMAND stores information about commands executed by the `piman` program
- MOUNTSPEED stores information about mount tracking speeds

Second part of "Pi of the Sky" database is star catalog and will be described in details in Section 3.3.1.2.

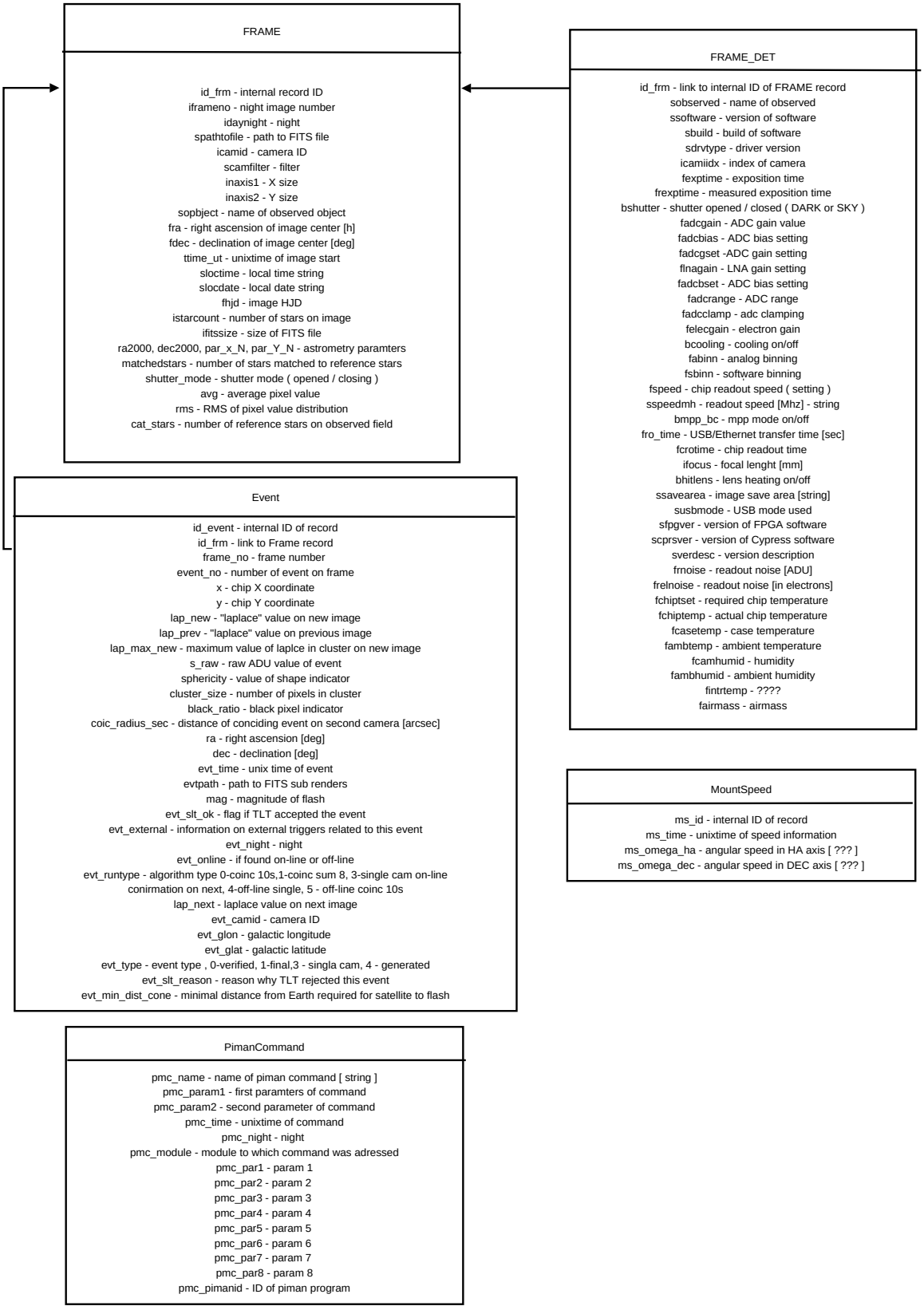

Figure 2.7: Structure of database for storing on-line information from the system

2.2.2.6 DAQ configuration

The `DAQ` can be configured by means of configuration files. Every parameter has a default value which is hard coded in the program code. In most cases this value is proper for typical system settings. However the default value can be overwritten by values read from configuration files. The priority of loading configuration files is the following (starting from the highest) :

- Some parameters may be overwritten by the options passed to the program from the command line
- Program looks for file `ccd.cfg` in the directory where it was started (current directory). In case it is found the settings from this configuration file are loaded.
- In case local `ccd.cfg` file is not found, program looks for global configuration file `$NDIR/cfg/ccd.cfg` and if found loads settings from this file

In case non of the `ccd.cfg` files is found the default values defined in program code are used. They may not be optimal for current system configuration so `ccd.cfg` file should be provided and good parameter values should be placed there. It is possible to change values of parameters during the data collection by executing command `change_param` from `pishell` program (Tab. A.5)

2.2.2.7 Observation Strategy

The experiment is mainly devoted to optical flashes related to GRB, thus observation planing depends on pointing direction of GRB detecting satellites. The final version of the detector covering 2 steradians will cover FOV of the SWIFT or GLAST satellite and pointing of cameras will depend on pointing direction of these satellites. In case of the prototype, FOV is much smaller, but observation strategy is very similar. The system tries to point the cameras to the center of FOV of one of the satellites which are capable of determining the burst position. Before June 2006 the primary targets were HETE and INTEGRAL, after this date highest priority target is SWIFT satellite. The observation plan is generated automatically in form of `pish` script (Fig. A.5). Currently the telescope

pointing is performed dynamically. Pointing information is obtained from GCN server through the software socket during the system operation and it is used to point the telescope. The `piman` program executes pointing command every half an hour. It calls procedure which finds the best target to be observed. The best target is chosen from list of targets sorted in order of priority, every target is checked and in case it satisfies several constraints :

- h $> 28°$
- distance from the Moon $> 30°$

It is chosen as the best target. The primary target is SWIFT, secondary target is INTEGRAL next targets are objects from list of interesting astrophysical objects (Tab. A.6) like blasars, AGNs etc. The list was compiled mainly from list of interesting objects used by Global Telescope Network (GTN) [21]. In case non of the targets on the list can be observed reserve target (az,h)=(90°,28°) is chosen. Always when new position of SWIFT is sent via software socket the `piman` program executes re-pointing procedure immediately. In order to optimize photometry, the telescope does not observe arbitrary positions. Instead it finds closest field from the predefined list and telescope points to the center of such field. In this way each star is measured always in the same position on the CCD chip. The above pointing procedure is performed during most of the night. Twice a night the whole sky scan is performed by the system to cover all sky, which takes about 2×1 hour.

2.2.2.8 Remote system control

The Pi of the Sky prototype is installed in LCO and full detector will also be installed in remote location. This imposes specific requirements for the system. The most important one is failure-free hardware. Another obvious requirement is that the system must be controllable via the Internet. Most of the system functionality can be controlled via the ssh protocol by logging to remote host and executing programs. However, nightly operation control does not require logging to remote host, most important information about the system performance is copied to local machines and is available by the WWW interface. This information is basically :

- Database records from on-line tables containing information about images and result from on-line algorithm (Sec. 2.2.2.5)
- Status files of all system components
- Log files of most crucial system components
- Parts of images containing events found by the on-line algorithm
- Some of collected sky images (converted to jpg format)

Under normal conditions the system does not require human attention, all jobs to be performed are started automatically from `crontab`. At given time night observation script is generated, configuration files and data folders are prepared, system is started and it runs until morning. The person on duty does not have to watch the system for the whole night. In order to warn about system problems special alert system was designed. In case status file from any module is not updated for long time or contains information about problems, e-mail or SMS is sent to person on duty. In such case probably human reaction will be necessary and will require logging to system controlling computer. The action would depend on the type of the problem, sometimes it is enough to execute `piman` command from `pishell` or restart one of the modules. In case of mount server problems calibration may be required, because mount can loss knowledge of its position.

2.3 Full Pi of the Sky detector

2.3.1 General Idea

The full version of the detector is currently under construction. The final system will consist of 2 sets of 16 cameras. Each camera will cover $20^\circ \times 20^\circ$ FOV, resulting in 2 steradians coverage of each set. The FOV of 2 steradians corresponds to FOV of the BAT detector on board of the SWIFT satellite ([22], [23]) , it also matches FOV of the LAT detector on the GLAST satellite ([24], [25]), which will be lunched in the fall of 2007. These two sets will be installed in different places separated by distance of ≈ 100 km. This separation distance will allow to reject optical flashes caused by near Earth objects (mostly satellites), using the

parallax effect. The stereo observations have already been tested in LCO (see Sec. 3.2.2). The parallax will allow to reject near Earth flashing objects up to orbit of the Moon.

2.3.2 Mounts

The mount design was based on the mount used in the prototype, but it was redesigned and improved (Fig. 2.8). In the new design 4 cameras are installed on a single mount. They can work in two modes so called *wide* and *deep*. In normal operation mode cameras are working in *wide* mode, looking at neighboring fields in the sky and covering FOV$\approx 40^\circ \times 40^\circ$. In order to obtain higher limiting magnitude and improve precision of the photometry all cameras on the mount can be pointed to the same position in the so called *deep* mode. This strategy can be used in case of GCN alert about GRB when it is important to increase the range of the system by averaging many images of same field from different cameras.
Hardware improvements in the mount design include also usage of harmonic drives to improve pointing precision. Better control of the position will be provided by 13 bit encoders which will be used instead of potentiometers. New design of the mounts includes changes in the control system New mounts will be controlled by the Controller Area Network (CAN) interface. However, to ensure flexibility CAN to Ethernet converters will be used and in fact all commands will be passed through the Ethernet interface. This allows much more flexible system, where every mount obtains IP address and can be controlled by any PC in the computer cluster. The system schematic is shown in Figure 2.9.

2.3.3 Cameras

Some improvements were also introduced in the final design of cameras. The major change is the Ethernet interface, which was added to decouple cameras from the dedicated computer. The major disadvantage of the USB2.0 interface in the prototype cameras was that they had to be connected directly to a PC computer. In case of the computer crash there was no way to use cameras until computer was repaired or replaced. This is not required in the case of the Ethernet

Figure 2.8: Design of new mount for full "Pi of the Sky" system (upper plot) and fully assembled mount in reality (lower plot)

interface, since the cameras can be connected to Ethernet switch and be part of local area network. In this configuration they can be controlled from any PC in the local network (Fig. 2.9). This ensures that crash of a single computer will not make any camera unusable. The second major modification was choice of the STA0820 CCD chip. There were also minor improvements. Stack of Peltiers was used to cool cameras down to 40° C below ambient temperature. Modifications in the analog electronics allowed to reduce the noise significantly. Lifetime of the shutter was also increased by special breaking algorithm implemented in electronic which reduces the shock caused by the shutter opening and closing.

2.3.4 Computer Cluster

As it was described above the final system will consist of two sites with 4 mounts, each carrying 4 cameras. It was established during tests of the prototype that data collection and analysis requires about 1 CPU per camera. This allows for data collection and on-line analysis during a night and off-line data reduction and analysis during a day. This implies that each set of cameras requires 16 CPUs to handle the system and data analysis. Instead of 16 computers, 8 machines with dual core architecture will be used. In any case they will form a cluster of computers, which must be efficiently managed. The system architecture is presented in Figure 2.9. The main idea of this cluster is that none computer is unique and in case any of the machines crashes the system must remain fully functional and other PC will take over the tasks of the crashed computer.

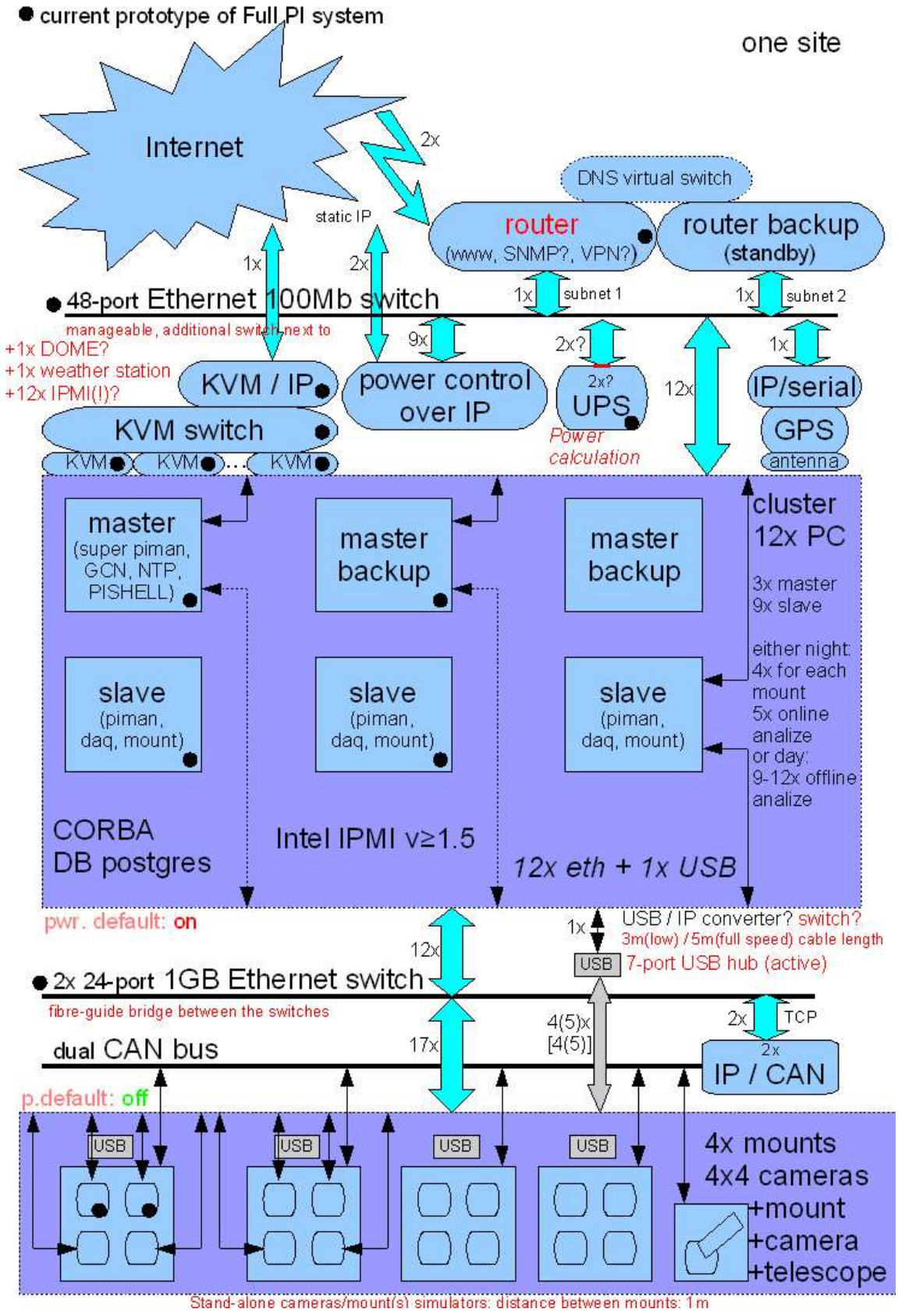

Figure 2.9: Design of the full "Pi of the Sky" system

Chapter 3

Data Analysis

Data analysis in the "Pi of the Sky" experiment consists of on-line and off-line parts. The on-line data analysis is required to control performance of the detector, but it is also responsible for finding optical flashes in very short timescale ($\approx$10 sec) in real time. Fast identification of optical flashes gives a possibility to distribute alerts in the community for follow-up observations. The off-line data analysis acts on the reduced data. The reduction pipeline consists of three main stages : photometry, astrometry and cataloging to the database. Final, reduced data consists of star brightness measurements stored in the database, which provides easy and effective access. Off-line analysis algorithms act on the database, there are several algorithms developed for different purposes. In this thesis two off-line algorithms implemented by author will be described in detail. These are flare recognition algorithm, looking for brightness increase of existing stars and algorithm for finding new objects on the sky. The database provides easy data access for broad spectrum of data analysis. Except algorithms presented in this thesis there are also other off-line algorithms implemented ([26], [27]).

3.1 On-line data reduction

On-line data analysis is required for detector control. The most important task is astrometry, it is a transformation of chip coordinates (x,y) to celestial coordinates (λ, δ) :

$$T : (x, y) \rightarrow (\lambda, \delta) \tag{3.1}$$

In order to obtain this transformation the following steps are performed :

- Fast image reduction - subtraction of dark image (this is step is also required by on-line analysis algorithm)
- Photometry - identification of stars on image and determination of instrumental magnitudes and chip coordinates (x,y)
- Astrometry - determination of non-linear transformation T:(x,y) $\rightarrow (\lambda, \delta)$ It is iterational minimisation procedure comparing stars identified in the image by photometry with reference stars from external star catalog. This procedure will be described in more detail in Section 3.3.1.1.

After finding astrometry transform it is possible to calculate celestial coordinates of any object on an image from its (x,y) coordinates. The (λ, δ) coordinates of image center are calculated and compared against the expected position and can be used to correct mount tracking speeds (so called `autoguiding` procedure). Before analyzing an image with on-line algorithm fast image processing is performed. The first step is dark image subtraction. In the next step, image is transformed by special transformation called `laplace`[1]. Value of each pixel is calculated as simple function of several surrounding pixels. Values in pixels just around transformed pixels are summed and values in other pixels far from it are subtracted. The idea of this transformation is to calculate simple aperture brightness for every pixel.

Several types of filters which were tested are shown in Figure 3.1. Images before and after applying the `g54 laplace` filter (aperture 4 in Fig. 3.1) in Figure 3.1) are shown in Figure 3.2 it can be clearly seen that stars are sharper on the filtered image. Finally, one filter was chosen according to lowest ratio of σ_g/g_{avg} value calculated for faint stars, where g_{avg} denotes average of maximum `laplace` value and σ_g stands for its dispersion. For old prototype version with

[1] because it resembles a discrete version of Laplace operator

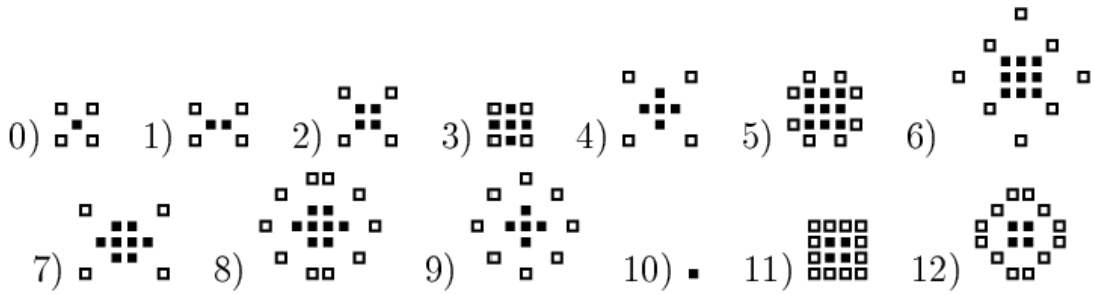

Figure 3.1: Different `laplace` types tested in on-line flash recognition algorithms

Carl Zeiss f=50mm lenses `laplace` 4 (`g54`) was used and for the Canon f=85mm `laplace` 12 is used. On-line algorithm is based on transformed images, distribution of pixel values after such transformation is centered around zero (Fig. 3.3). For every collected image a gaussian curve is fitted and signal threshold T_n is calculated as multiplicity of sigma value (typically $T_n = 5\sigma_B or 6\sigma_B$).

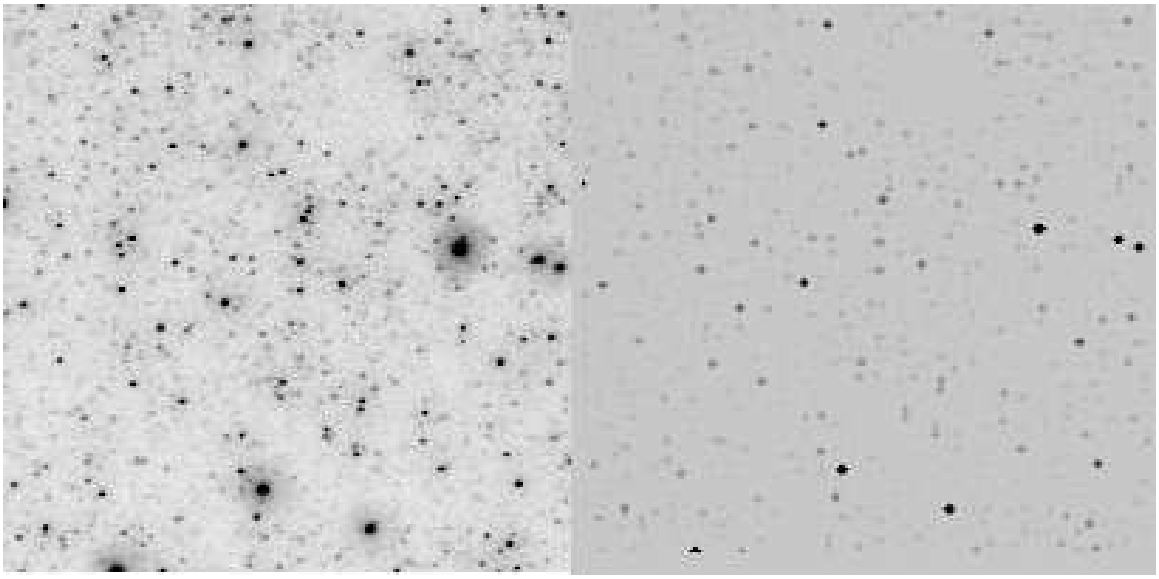

Figure 3.2: Sky image before and after applying the `laplace` filter

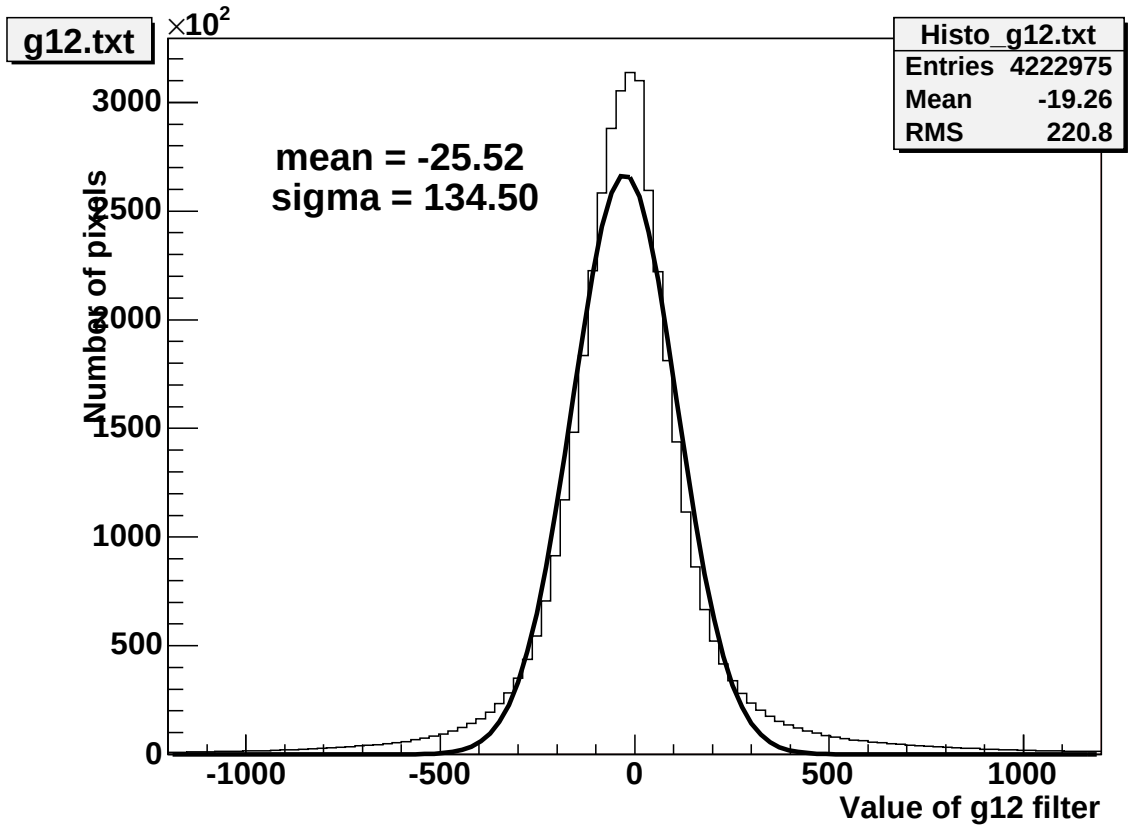

Figure 3.3: Distribution of `laplace` 12 values on single image

3.2 On-line flash recognition algorithms

The aim of this type of algorithms is to find optical flashes occurring in single image time scale (10 sec). The signature of such events is the following. Interesting event is an object which appears in a new image of the sky and was not present in the previous image of the same field which was taken a moment before. However, this simple idea is not so simple in practical realization. Most of the events found by such simple comparison of two images are due to background. The crucial task of the efficient algorithm is to reject most of the background without too much loss of the signal. It is realized by a multilevel triggering system based on ideas used in particle physics. Every image consists of $4 \cdot 10^6$ pixels suspected of being potential interesting event. Image should be analysed while next image is being collected which takes $\approx$ 12s in the current configuration. This means that the algorithm must be fast. First trigger levels are simple and fast, they reject big amount of event candidates with simple criteria. Higher trigger levels have more time and can use more sophisticated criteria to reject background events.

3.2.1 First Level Trigger

This level of algorithm must handle the highest data rate, so it must be very fast and simple. It should preserve most of the signal and reject big fraction of non-interesting pixels. At this stage flash-like events in single camera are identified and saved to log files and optionally to the database. The following list of cuts are applied to every pixel in the image :

- T_n - this cut selects stars on new image by requiring signal in the analysed pixel. The condition for signal presence is :

$$N(x, y) > T_n \tag{3.2}$$

  where N(x,y) is ADU value in pixel (x,y) of the new image and the threshold T_n is specified by DAQ configuration parameters in multiplicities of σ_B (Tab. 3.1). The goal of this cut is selection of all stars in new image.

- T_v - this cut rejects constant stars. It requires that there is no signal on previous frame. Previous frame in this case means not just one single image, but average of several previous images. The condition for value of pixel on previous image is :

$$P(x, y) < T_v \tag{3.3}$$

  where P(x,y) is the value in pixel (x,y) on the average of N_{aver} previous images. Pixels remaining after this cut should be new objects which appeared on new image and were not present on previous images. In fact some pixels belonging to edges of bright stars can also survive this cut.

- **MinLaplace** - rejects pixels which have value on previous image lower then minimal accepted value (T_{MinLap}). This cut allows to reject edges of bright stars where values of pixels after `laplace` filter often become negative, but can also vary to values exceeding T_n.

- **IfMoreAfterTv** - rejects the whole image if number of pixels accepted after T_v cut exceeds certain limit N_{MaxTv}. This cut allows to reject images with big number of events which are due to system error, Moon light or clouds. The image is bad, all events are certainly rubbish so they are rejected and no further analysis of this image is performed.

- **LocalMax** - requires that pixel value is a local maximum. This cut allows to choose only one pixel of star like object for further analysis.

- **SkipOverlaps** - checks number of accepted pixels in certain radius $R_{overlap}$ from current pixel and leaves only one event and skips the others. This cut narrows the number of pixels to be analysed which are related to the same object to a single one.

- **Shape** - object shape indicator S is calculated. It is defined as :

$$S = \frac{S_{cluster}}{S_{circle}} \tag{3.4}$$

where $S_{cluster}$ is area of cluster and S_{circle} is the area of the smallest circle circumscribed on this cluster. Cluster is defined as group of pixels around current pixel with values satisfying $N(x, y) >= T_{cluster}$. Shape is required not to be elongated by imposing :

$$S > T_{shape} \tag{3.5}$$

The idea of shape calculation is shown in Figure 3.4. The distribution of shape value for stars in single image is shown in Figure 3.4.

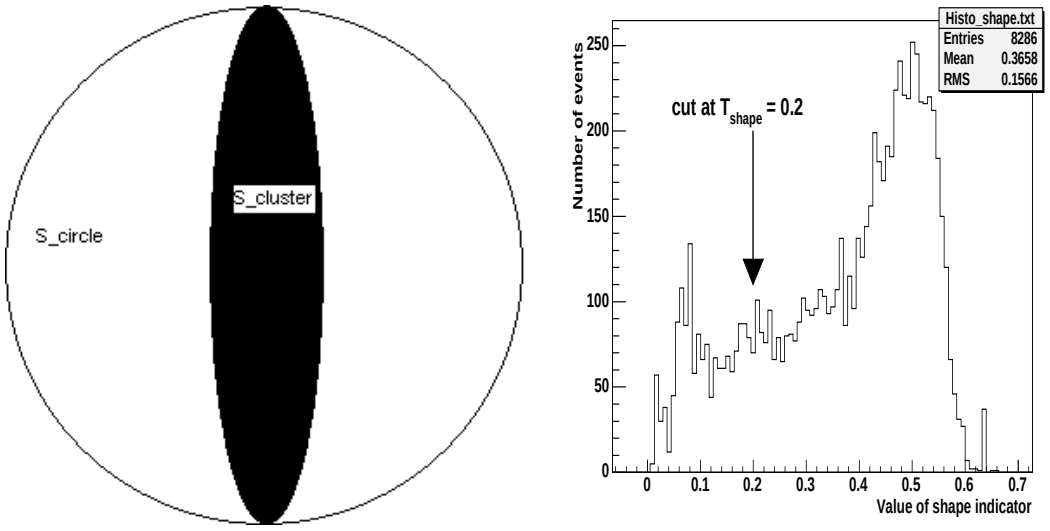

Figure 3.4: Idea of shape indicator calculation (left) and its distribution for events from single night and camera (right)

- **BlackPixels** - this cut rejects pixels which have much smaller signal then neighboring pixels causing `laplace` filter to be high due to underestimation of the background level. The following requirement is imposed on every pixel :

$$\frac{Min(P_{-})}{\sum P_{-}} >= T_{black} \tag{3.6}$$

where P_- is value of pixel entering the `laplace` function with minus sign. Black pixels on reduced image can be due to CCD defects or temperature fluctuations, but are rather very seldom (Tab. 3.2).

- **HotPixels** - due to CCD chip defects some pixels can shine much more then normal good pixels. Such effects should generally be subtracted by the dark image subtraction. However, sometimes it is not enough, because new hot pixels can appear temporary during a night and become quiet again later. The main reason for this are excitations from cosmic ray hits. Two ways of rejecting such events have been implemented. The first one is calculation of average value in pixel on previous images, the imposed criteria is :

$$\frac{\sum_{i=-1}^{N_{aver}} P_i(x,y)}{N_{aver}} < T_{hot} \tag{3.7}$$

where $P_i(x,y)$ is value in pixel on image i before current one. In the case of tracking mount this cut is neutral due to T_v cut which is stronger. Second anti-hotpixel cut is rejection of pixels by list of known hot-pixels. This list is updated regularly when new defects are found. This is done "manually" by running report which shows events described as hot pixel. In case certain pixel is regularly giving false alerts it can be added to list of known hot pixels which is stored in the database (Fig. 3.26).

- **IfMore** - checks nearby events in distance of R_{ifmore} pixels, in case number of events exceeds limit of N_{ifmore} all nearby events are rejected. This cut allows to reject events caused by planes or satellites.

Parameters for algorithms are passed as described in the previous Section 2.2.2.6. The parameters used in First Level Trigger (FLT) are listed in Table 3.1.

The output from FLT is basically a list of events from single camera. These events are saved to a log file allevents_N.log (N stands for index of camera). Optionally they can be saved to database to provide easy access for further analysis. Table 3.2 shows background rejection efficiency of subsequent FLT cuts.

Cut	Parameter Name	Value for confirmation on next	Value for coincidence	Notes
LaplaceType	LaplaceType	12	12	For Carl Zeiss f=50 `laplace`=4
T_n	T_n	6	5	For Carl Zeiss f=50 T_n=5
T_v	T_v	2	2	For Carl Zeiss f=50 T_v=2.5
	N_{aver}	7	7	Number of averaged previous images
MinLaplace	T_{MinLap}	0	0	-
IfMoreAfterTv	N_{MaxTv}	3000	3000	-
SkipOverlaps	SkipOverlaps	1	1	enable/disable
SkipOverlaps	$R_{overlap}$	4	4	radius in which overlaps are skipped
Shape	T_{shape}	0.2	0.2	-
BlackPixels	T_{black}	0.5	0.5	-
HotPixels	T_{hot}	3.6	3.6	Threshold in anti-hotpixel cut
IfMore	N_{IfMore}	20	20	
	R_{IfMore}	150	150	

Table 3.1: FLT parameters summary, real names of parameters used in configuration files can be found in Table B.1 in Appendix B.1

Cut	% of all events	% of events from previous level	Number of events
All	-	-	$3.212 \cdot 10^9$
Tn	$2.06 \cdot 10^{-2}$	2.06%	66434214
Tv	$2 \cdot 10^{-5}$	0.1%	68345
MinTv	$2 \cdot 10^{-5}$	96.99%	66293
Overlap	$5 \cdot 10^{-6}$	25.79%	17100
Black	$5 \cdot 10^{-6}$	100%	17100
Hot	$5 \cdot 10^{-6}$	100%	17100
IfMore	$5 \cdot 10^{-6}$	99.85%	17075
Coinc	$2.5 \cdot 10^{-6}$	46.28%	8099
Satellites Catalog	$2.49 \cdot 10^{-6}$	98.96%	8015
Star Catalog	$2.32 \cdot 10^{-6}$	93.11%	7463
Shape	$2.25 \cdot 10^{-6}$	97.10%	7247
Tracks	$1.55 \cdot 10^{-9}$	0.06%	5
Accepted	$1.55 \cdot 10^{-9}$	0.06%	5

Table 3.2: Number of events remaining after subsequent cuts of coincidence algorithm for night 20060527

3.2.2 Second Level Trigger

The action at this level depends on the type of the system setup. Generally three configurations are possible :

- two cameras on a single mount working in coincidence. In this configuration events found by the first camera are verified in the corresponding image from the second camera. Only events present in images from both cameras are accepted. This configuration is realized by the prototype system in LCO.
- single camera, coincidence is replaced by confirmation of signal on next image. This version of algorithm was also realized in LCO when one of the cameras was not working due to technical problems. This can be used as cross-check algorithm for two cameras working in coincidence.
- two cameras in separate locations working in coincidence. This will be realized in the full version of the system. Cameras will be paired, both cameras in the pair will observe the same field on the sky. Spatial and time coincidence of the flash in both cameras will be required.

In any case coincidence requirement is one of the most important cuts. The main goal is rejection of so called *cosmic events*. These are cosmic rays hitting the CCD chip and imitating astrophysical flashes (Fig. 3.5). In many cases *cosmic events* have Point Spread Function (PSF) completly different then PSF of stars and they could be rejected by a shape recognition procedure. However, in many cases they are very similar to PSF of the stars, even if this is a very small fraction of all *cosmic events* this would cause all flashes found be they algorithm to be uncertain. A way of definite rejection of all *cosmic events* is required for credible flash recognition algorithm. Probability that different cosmic ray particles will hit two chips in the same time and in the same positions (with respect to stars) is negligible. Coincidence is also very effective way of rejecting background events due to sky background fluctuations, edges of bright stars and clouds. In the prototype version cameras take images parallely so the only parameter is the angular distance of events in both cameras, currently used default value is

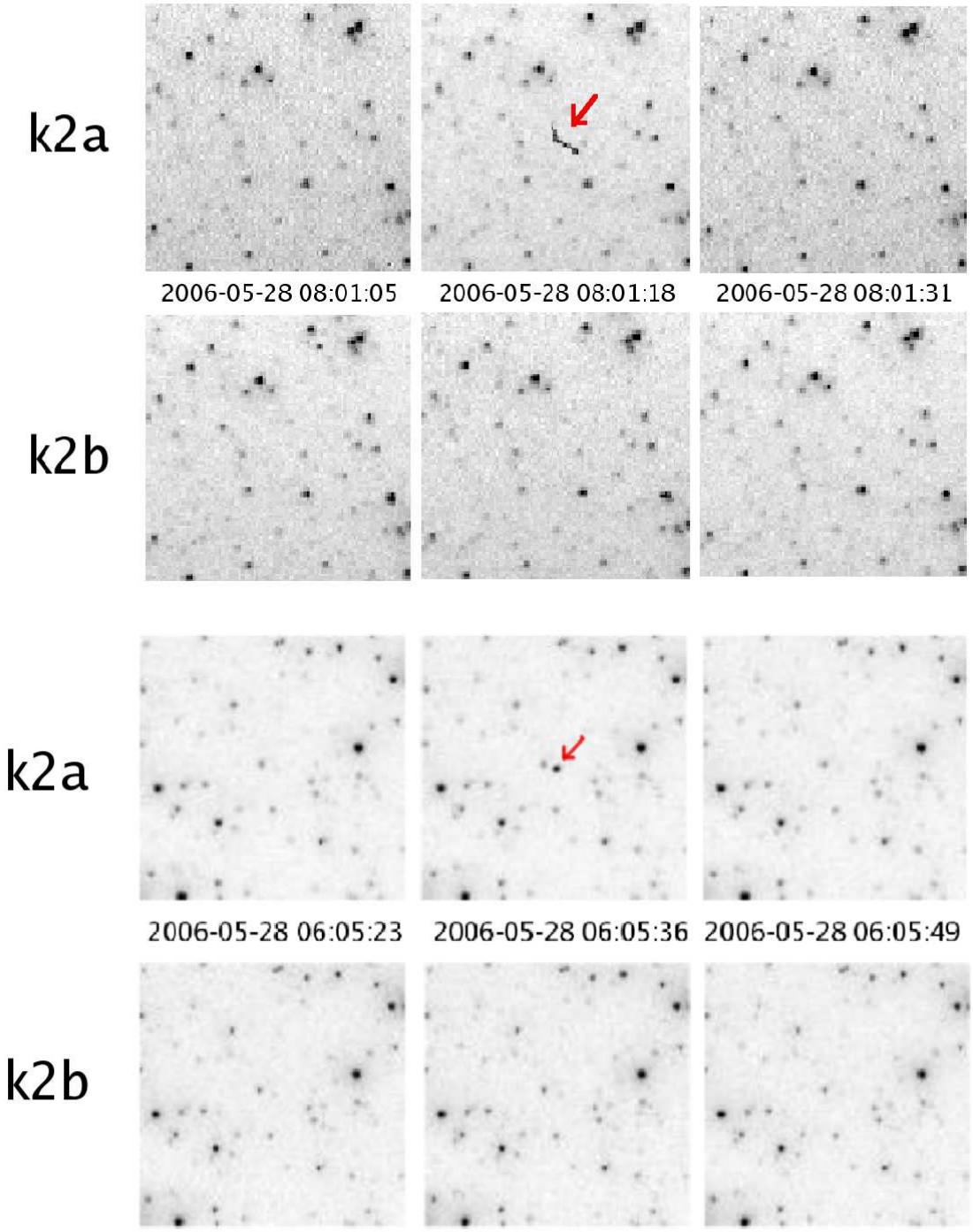

Figure 3.5: Examples of events caused by cosmic ray hitting the CCD chip, with PSF easy to distinguish from stars (upper image) and very similar to PSF of stars (lower image)

$R_{coinc} = 150$ arcsec (250 arcsec was used for Carl Zeiss f=50mm lenses). It was determined from distribution of angular distances of corresponding stars in both cameras (Fig. 3.6).

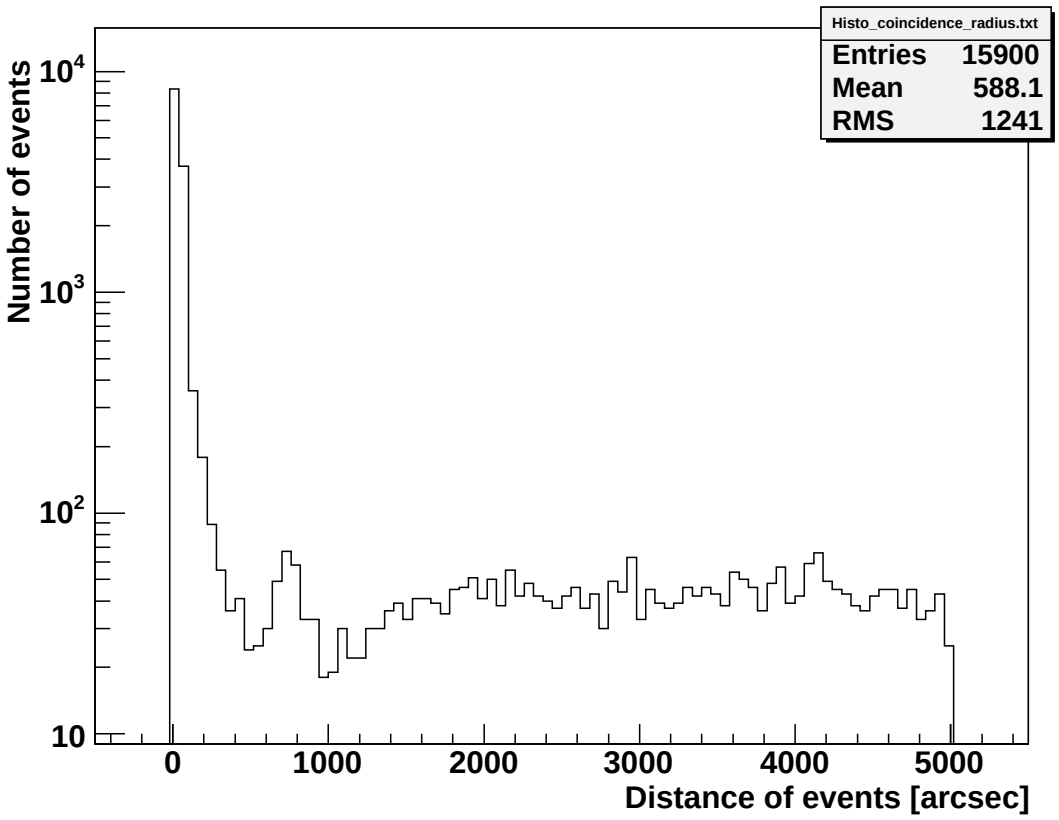

Figure 3.6: Distribution of angular distances between events from corresponding images collected by cameras k2a and k2b on 2006-05-28

In the case of coincidence between cameras in separated sites, the significance of coincidence is even bigger. In this configuration it is possible to reject close Earth flashing objects using the parallax (Fig. 3.7). Artificial satellites orbiting the Earth may rotate and sometimes reflex Sun light causing flash-like events. The best method to distinct such kind of flashes is to use parallax. In the prototype version of the experiment two cameras are installed on single mount. However, this method was tested by coincidence with the RDOT experiment [28] located on La Silla at the distance of ≈ 30 km.

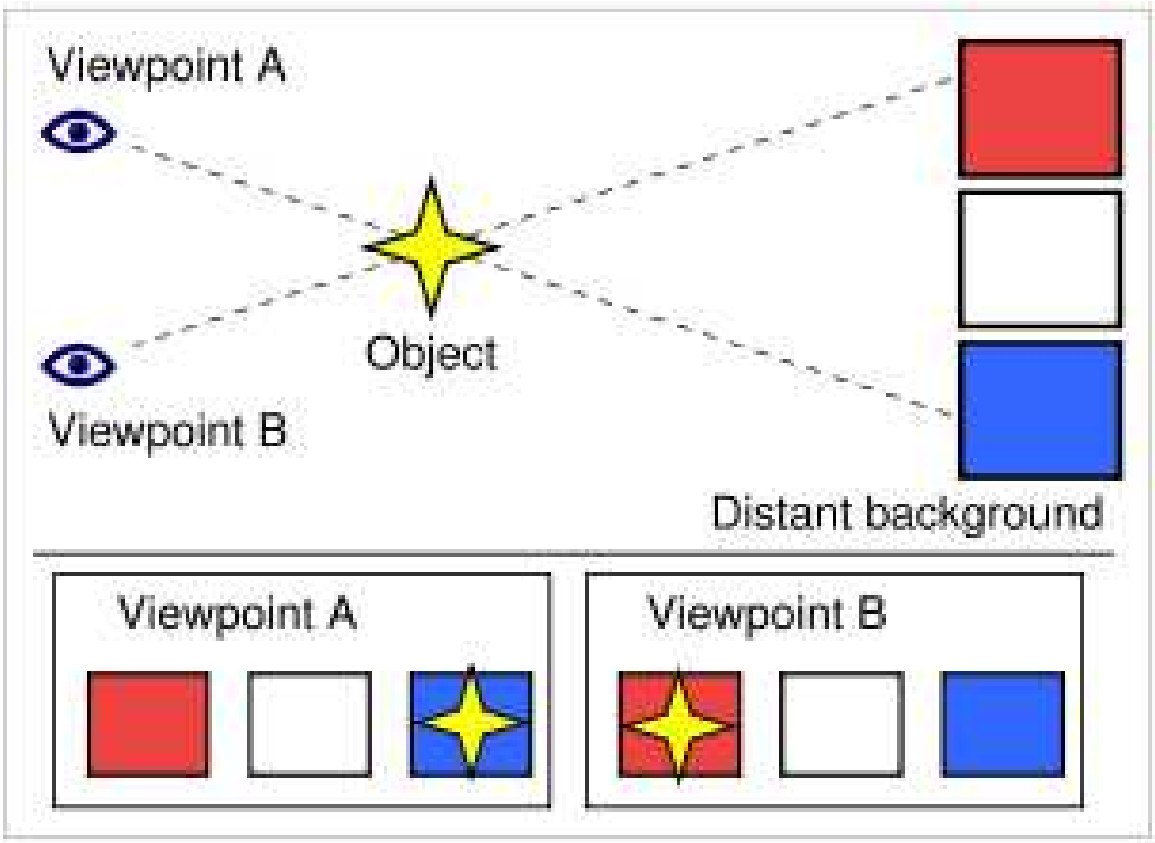

Figure 3.7: A simplified example of parallax

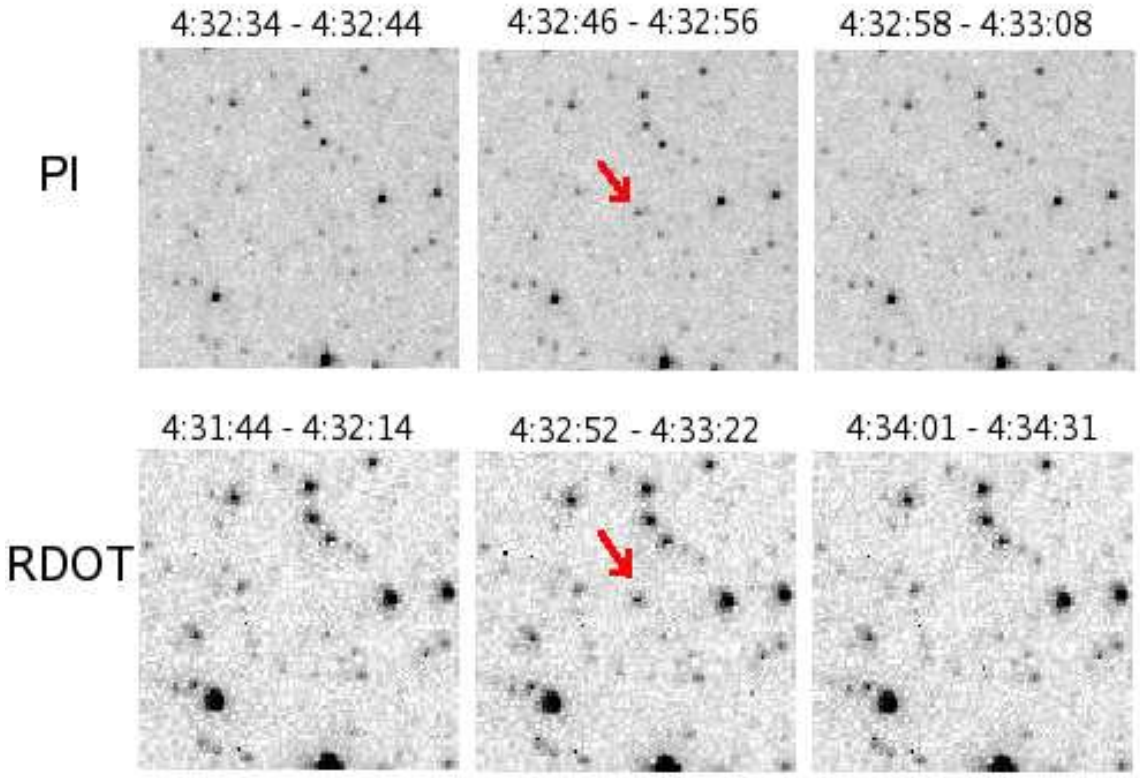

Figure 3.8: Stereo observations of near Earth satellite by experiments PI and RDOT

Image collected by the prototype containing an optical flash was compared with image of same area of the sky taken by RDOT telescope at the same time. It can be clearly seen in Figure 3.8 that optical flash is visible in different positions with respect to the stars. Requirement of spatial coincidence would reject such event. It is probably caused by a satellite in distance of < 25000 km from Earth. This method has been tested for a few nights only. During normal operation it is not possible for the prototype to use parallax to reject flashing satellites. In order to reject most of such events a database of known satellites is used. It is retrieved every night from the Internet and contains ≈ 10000 orbital elements. For every image, positions of all satellites in the database are calculated and every flash candidate is verified. In case it is closer then $R_{sat} = 0.5°$ from any of the satellites it is rejected. The rejection radius was determined from distribution of angular distances from flashes to closest satellite from the database which is clearly peaked around zero (see Fig. 3.9).
The red dots on this plot represent distribution of distances to the closest satellite from the catalog for randomly distributed flashes. A clear peak at $R_{sat} < 0.5°$ is visible, it corresponds to real satellites.

The orbital elements databases are not complete, there are many satellites which are not included (e.g. spy satellites). In order to reject such kind of objects event candidates from many consecutive images are examined against track conditions. In case it is possible to fit track to set of events from different images and velocity of object is constant all events on the track are rejected. This rejects big fraction of flashing satellites and planes (see Fig. 3.10), however it is possible that rarely flashing (rotating) satellites survive this cut.

At this level of the trigger each event candidate is checked against the catalog of constant stars. TYCHO-2 star catalog [29] is used for this purpose. The event candidate is rejected in case there is a star brighter then Mag_{max} in radius R_{star} (see Tab. 3.3). Stars can imitate flashes mainly due to clouds. When cloud moves and uncovers a star the FLT identifies such an event as flash. All events accepted by coincidence are saved to files `verifiedevents_N.log` and optionally to the database. Events accepted by the SLT are saved to `finalevents_N.log` and to the database. Parameters used in the SLT are listed in Table 3.3. Block diagram

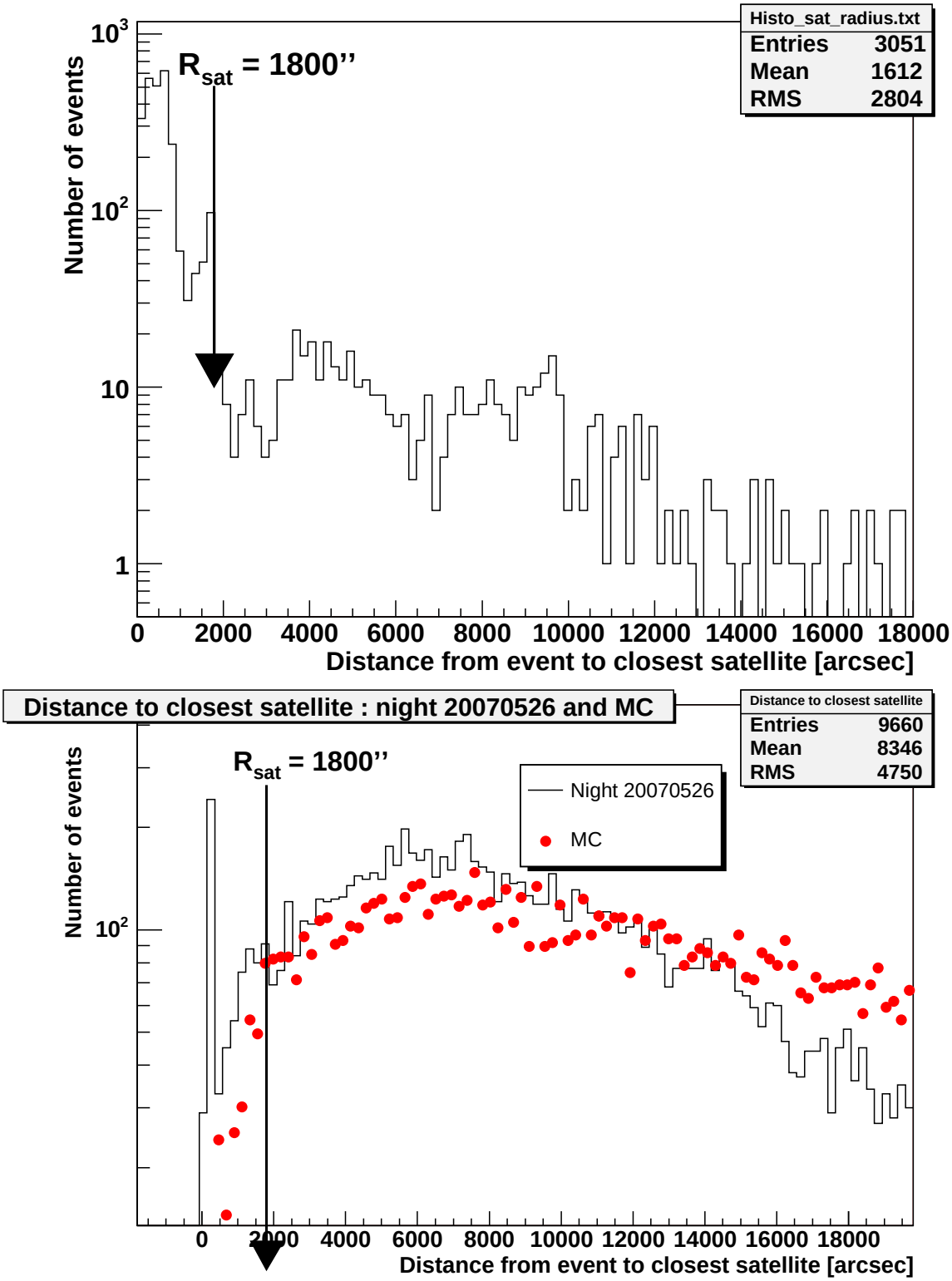

Figure 3.9: Distribution of distance from event to closest satellite from the catalog. For coincidence algorithm night 20041028 (upper plot) and for single camera events from night 20070526 (lower plot).

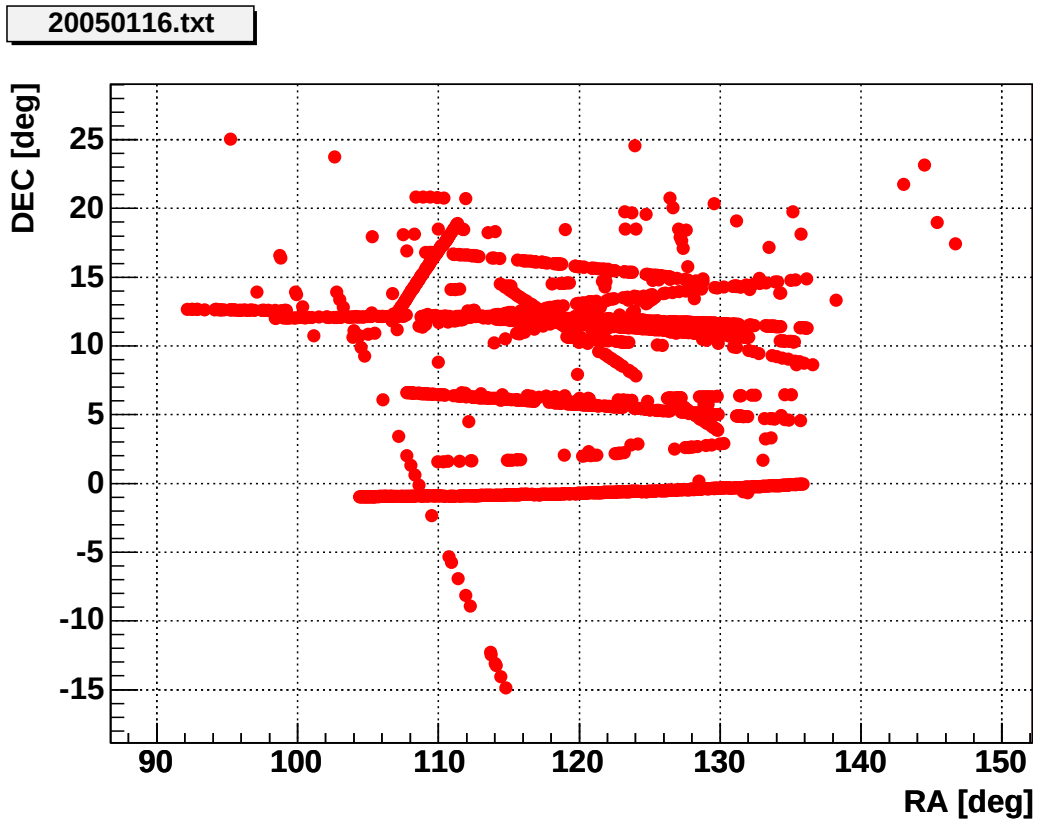

Figure 3.10: Events rejected by track cut during single night 2005-01-16/17

of on-line flash recognition algorithm is shown in Figure 3.12. Rejection efficiency of subsequent FLT and SLT cuts are show in Figure 3.11.

Cut	Parameter Name	Current Value	Notes
Coincidence	R_{coinc}	150"	It was 250" for objectives f=50mm
Confirmation on next	$N_{confirm}$	1	in case of 2 cameras working it is 0
Satellite Catalog	R_{sat}	1800"	
Star Catalog	R_{star}	120"	
	Mag_{max}	13	
Track	N_{track}	200	number of subsequent images used for track fit
	χ^2_{add}	700	Maximum allowed distance (in pixels squared) from existing track to match event
	χ_{seed}	100	Maximum value of χ^2 to initialize new track

Table 3.3: SLT parameters summary, real names of parameters used in configuration files can be found in Table B.2 in Appendix B.1

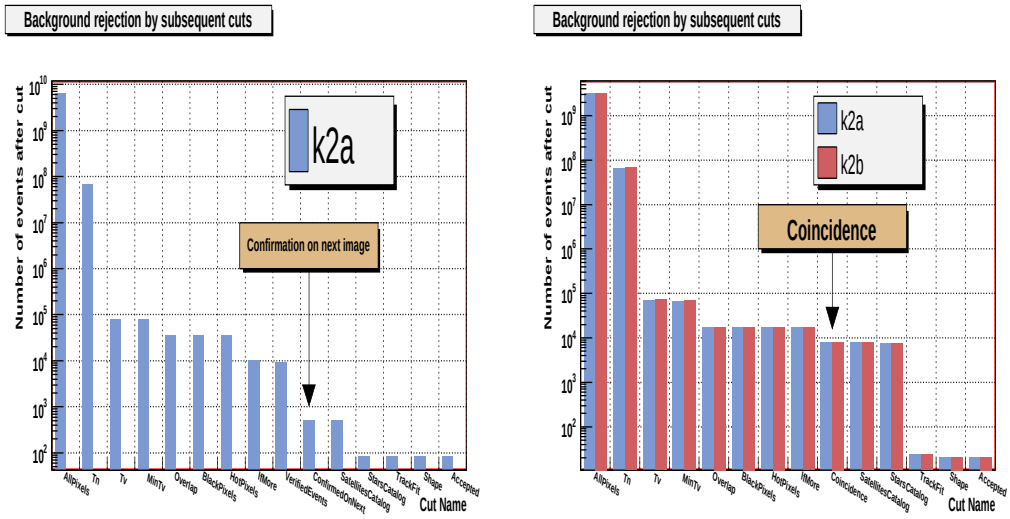

Figure 3.11: Background rejection efficiency of algorithm requiring confirmation of flash on next image (left) and coincidence of two cameras (right)

3.2.3 Third Level Trigger

The first two levels of the trigger retain very small number of events, on average it is not more then 10 per night. It depends strongly on weather conditions and in case of cloudy night this number can reach hundreds, but in this case fast inspection can reject most of them. However, in the full system the number of events will be 16 time larger reaching 100-200 per night and this will be much

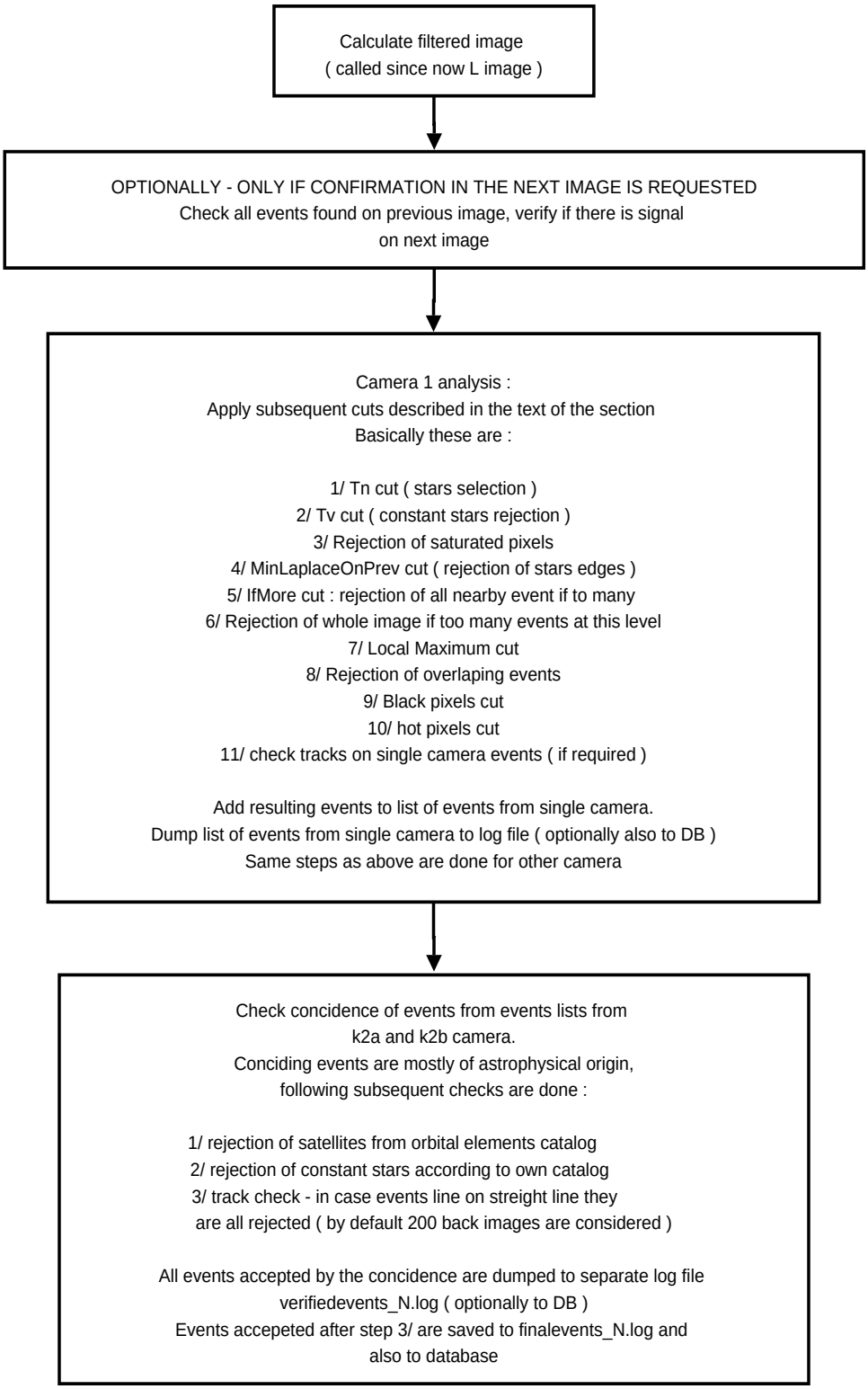

Figure 3.12: Block diagram of on-line flash recognition algorithm

more difficult to inspect. For this reason the third level of the trigger has been implemented. It checks final events accepted by previous levels which ensures that only small number of events will be examined. Thus it is possible to implement much more sophisticated and time consuming algorithms to check every event. Current implementation of the TLT consists of the following cuts :

- Comparison of event on both cameras and require signal to be similar on both cameras, by imposing condition : $L_{min}/L_{max} < L_{diff}$
- Checks sphericity again with optionally more strict criteria $Shape < T^{TLT}_{shape}$
- Simple Hough transform [1] - uses small image part surrounding event. It finds pixels with signal above certain level T_{hough} and creates distribution of ϕ coordinate ($\phi = atan((y - y0)/(x - y0))$). In case this distribution has significant peak this means it is probably due to straight line from a plane or a satellite (Fig. 3.13)

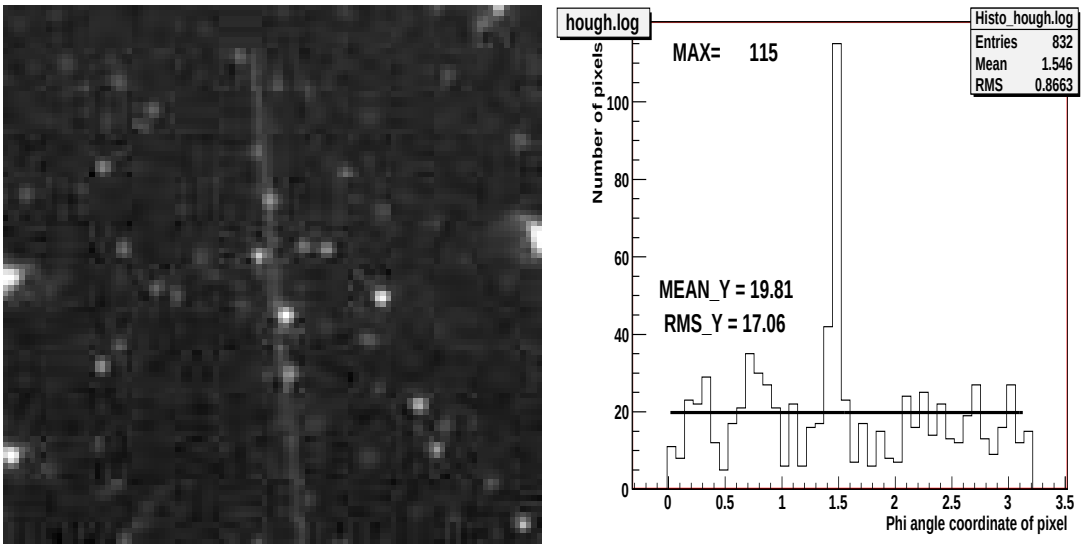

Figure 3.13: Original event image (left image) and distribution of ϕ angle coordinate for background event (right image)

[1] Hough transform is a technique of image transform from (x,y) to cylindrical (r,ϕ) cooridinates in order to find particular shapes in an image

Event will be considered as straight line in case maximum of distribution of angle ϕ satisfies the following condition :

$$\begin{aligned} N_{max}(\phi) > MEAN_Y + T_{hough\_distr} * RMS_y \\ AND \\ N_{max}(\phi) > T_{hough\_height} * MEAN_Y \end{aligned} \tag{3.8}$$

- Track check : checks event against fitted tracks in case event was not correctly merged to existing track by the on-line algorithm
- Algorithm on parts : algorithm is re-run on small parts around the event, with less strict threshold $T_n^{TLT} = 4 \cdot \sigma_B$. The event is rejected if tracks are found and it belongs to one of these tracks.
- Cloud check : checks number of stars in the full image and rejects event if $N_{stars} < 8000$.
- Frame line : in case line can be fitted to events from a single frame rejected by Hough transform, all events from the same frame and not yet rejected, are matched to this line. In case event matches this line it is rejected
- All line check : checks if straight line can be fitted to events from final list. It rejects final events matching this line.

Cut	Parameter Name	Current Value	Notes
Hough Transform	T_{hough}	1.5	Threshold for choosing pixels
Hough Transform	T_{hough_distr}	4.5	Threshold for peak in ϕ distribution (in σ)
Hough Transform	T_{hough_height}	2.0	Threshold for minimum peak height (in multiplicity of mean value)
clouds cut	N_{tstars}	40000	typical number of stars on whole image
clouds cut	R_{clouds}	0.2	reject if when number of stars on image $< R_{clouds} \cdot N_{tstars}$
Event Comparison	L_{diff}	0.25	requires similar signal on both cameras
Algorithm on parts	T_n^{TLT}	4.00	more loose then normal T_n

Table 3.4: TLT parameters summary, real names of parameters used in configuration files can be found in Table B.3 in Appendix B.1

Events rejected by this level of algorithm are not excluded from final list of alerts claimed by the system. They are only flagged, this flag can be useful

indication for a person inspecting all night events. Results of TLT are saved to log file and database. Parameters used in TLT are listed in Table 3.4.

3.2.4 Optimization of algorithm parameters

Optimization of algorithm parameters was a very important step of algorithm development. Algorithm parameters can be changed by settings in the ccd.cfg file as described in section 2.2.2.6. Tables 3.1 and 3.3 list most important parameters of algorithms on first and second levels of the trigger.

Some parameters were optimized by specific studies and the others were optimized by running algorithms on sky data with simulated optical flashes. Testing of algorithms was performed on regular sky images. Exactly the same software was used, but instead of reading images from the camera like during regular night observations, the images were read from the `fits` files stored on a disk. Images were analysed and found events were considered as background events. Optical flashes were simulated in such a way that samples of star of given brightness were extracted from images and pasted on image in random positions. Single test of given parameters set was performed on all images from single night, program was analyzing subsequent images and one generated flash was added in every image. After all images were analysed the number of background events N_{bkg} was counted, as those which were found by the algorithm, but were not generated. Generated events were checked on the list of identified events and number $N_{genident}$ of those generated and identified was determined. Efficiency of identifying optical flashes of given magnitude was determined as :

$$\epsilon(mag) = \frac{N_{genident}(mag)}{N_{generated}(mag)} \tag{3.9}$$

Every set of tested parameters was plotted as a single point on plot of ϵ vs N_{bkg}. Different points on this plot show values determined for different settings of the algorithm. For every star magnitudo separate plot was created. Figures 3.14 and 3.15 show results of efficiency and background rejection tests for algorithm requiring confirmation of flash on next image performed on 500 images from night 2007.04.25/26. These plots were created for simulated flashes of 9^m and

10^m respectively, best values of efficiency are also shown in Table 3.6. According to these results optimal parameter values were chosen (see Tables 3.1 and 3.3).

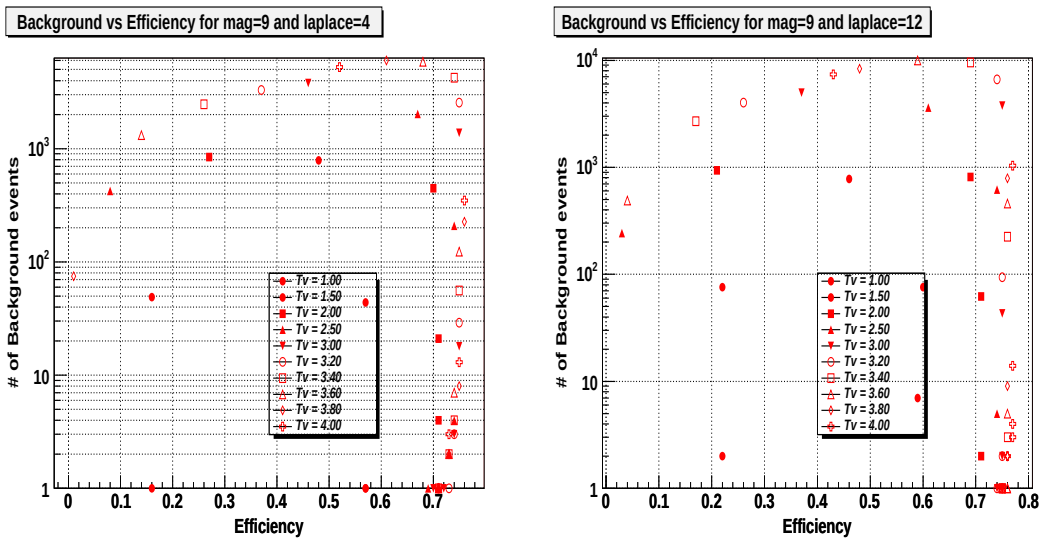

Figure 3.14: Efficiency of 9^m flash recognition and background rejection. Test performed on data from night 2007-04-25/26 for laplace=4 (left plot) and laplace=12 (right plot)

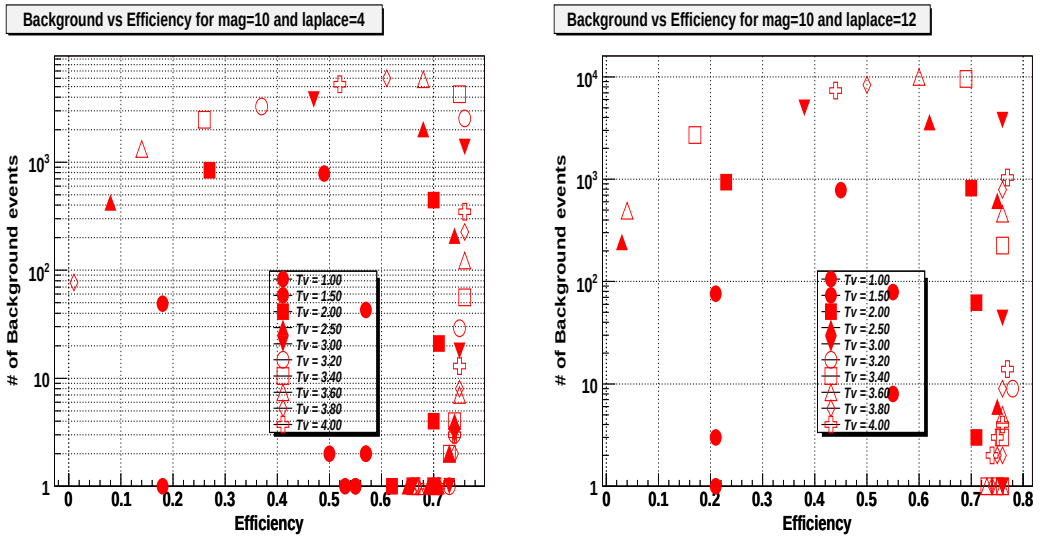

Figure 3.15: Efficiency of 10^m flash recognition and background rejection. Test performed on data from night 2007-04-25/26 for laplace=4 (left plot) and laplace=12 (right plot)

Magnitude	Laplace	Tv	Tn	Eff	Bkg
9	4	2.0	4.0	0.71	21
9	4	2.0	5.0	0.71	4
9	4	2.0	6.0	0.70	0
9	4	2.5	4.0	0.74	209
9	4	2.5	5.0	0.74	4
9	4	2.5	6.0	0.73	2
9	4	3.0	5.0	0.75	18
9	4	3.0	6.0	0.74	3
9	12	2.0	5.0	0.71	2
9	12	2.0	6.0	0.71	0
9	12	2.5	5.0	0.74	5
9	12	2.5	5.0	0.74	0
9	12	3.0	5.0	0.75	43
9	12	3.0	6.0	0.75	2

Table 3.5: Best values of efficiency of confirmation on next image algorithm obtained for simulated flashes of brightness 9^m, tested on 500 images from night 2007-04-25/26.

The efficiency and background tests were also performed for the coincidence algorithm. The Figures 3.16 show best points of T_n and T_v thresholds.

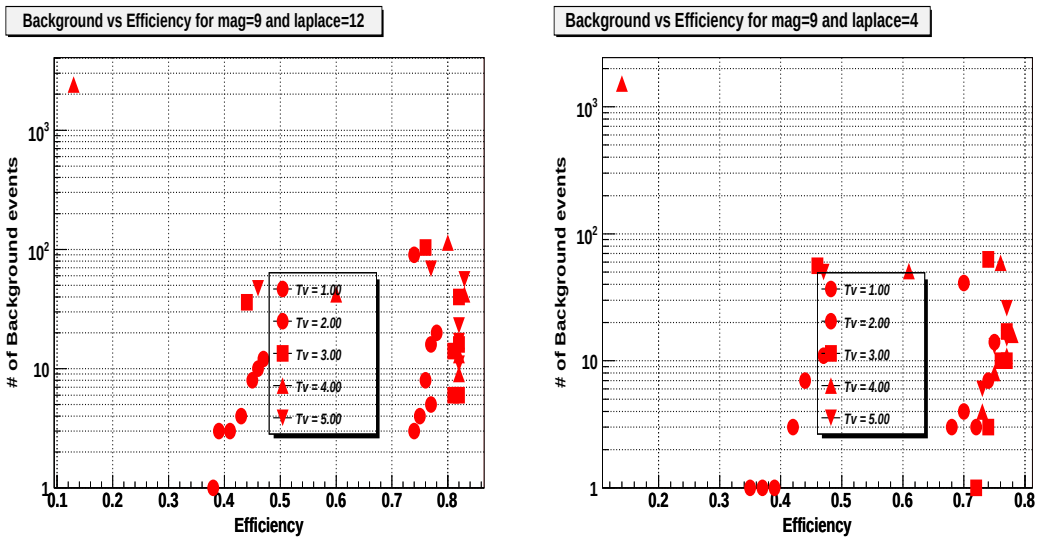

Figure 3.16: Results of efficiency of 9^m flash recognition and background rejection of coincidence algorithm. Tests were performed on data from night 2006-05-26/27 for laplace=4 (left plot) and laplace=12 (right plot)

The efficiency losses due to subsequent cuts of on-line algorithm was tested. It was done by counting how many of generated samples were rejected by on-line algorithm cuts. Figure 3.17 shows efficiency losses due to subsequent cuts for data collected during single night.

The efficiency of on-line algorithm cuts was determined for several different

Magnitude	Laplace	Tv	Tn	Eff	Bkg
9	4	2.0	5.0	0.75	14
9	4	2.0	6.0	0.74	7
9	4	3.0	5.0	0.74	63
9	4	3.0	6.0	0.77	17
9	12	2.0	5.0	0.78	20
9	12	2.0	6.0	0.77	5
9	12	3.0	6.0	0.82	40
9	12	3.0	7.0	0.82	16

Table 3.6: Best values of efficiency of coincidence algorithm obtained for simulated flashes of brightness 9^m, tested on 1952 images from night 2006-05-26/27.

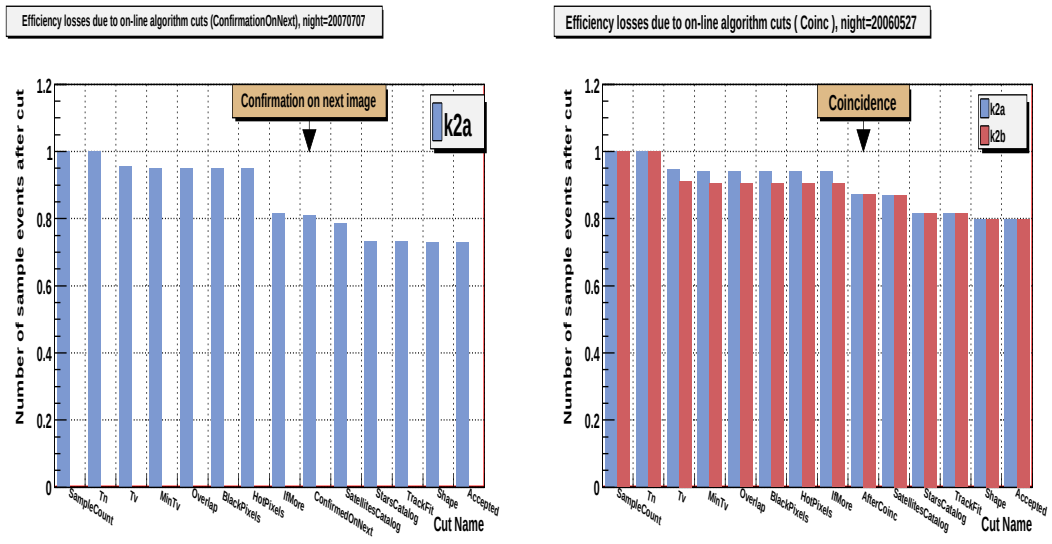

Figure 3.17: Efficiency losses due to subsequent cuts of on-line algorithm : confirmation on next image (left plot), coincidence (right plot)

nights (Fig. 3.18). The mean efficiency is $\approx 70\%$. The testing procedure which pasts samples of stars into image does not take clouds into account. In fact this procedure allows to estimate effieciency of all cuts after Tn cut. This is because sample is pasted on image also on top of clouds, so there is no possibility to have large loss of efficiency due to clouds in such kind of testing procedure. The efficiency of Tn cut was estimated with usage of the TYCHO-2 star catalog and cataloging procedure, it will be described in the Section 3.3.1.4.

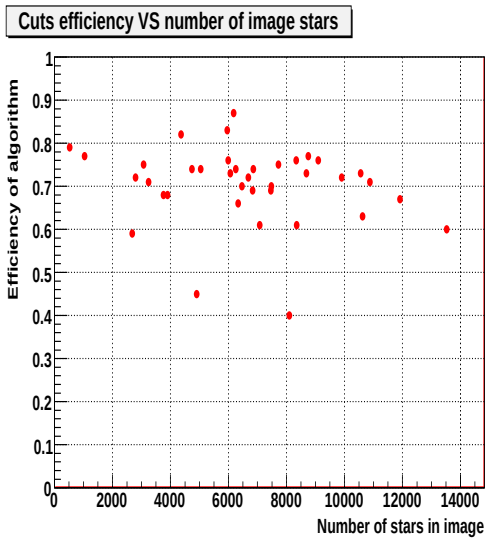

Figure 3.18: Efficiency of on-line algorithm cuts determined for data from several different nights. The X axis represents the average number of stars in images collected during a single night.

3.2.5 Sources of background

Final list of events from one night of the prototype work did not exceed 100 events , but typically remained on the level of 10 events per night. These events were mainly due to background events. The main sources of the background were flashing satellites, planes, meteors. There was also background due to clouds, in this case usually number of events on cloudy images was big so it was easy to simply reject whole image. Summary of background events statistics is given in Table 3.7. This table is divided into periods and types of algorithms in the

cases where more then one algorithm was running. Example of background event images are given in Figures 3.19, 3.20, 3.21 and 3.22.

Period	Algo	Flash	Satellite	Flash or satellite	Plane	Satellite or plane	Other[1]	Meteor	Clouds	Hot Pixel	Saturated Star	Opened Shutter	System Error
2004.06.25 - 2005.01.20	coinc	70	572	29	392	134	435	-	-	-	-	-	-
2004.06.25 - 2005.08.09	coinc	125	961	53	819	455	1810	-	-	-	-	-	-
2005.04.21 - 2005.08.09	conf. next	23	31	8	-	2	3676	-	-	-	-	-	-
2006.05.20 - 2006.08.08	coinc	42	352	10	262	324	1968	-	-	-	-	-	-
2006.05.20 - 2006.12.31	conf. next	3	118	2	6	12	4817	0	113995	168	12	60	64
2006.10.01 - 2006.12.31	conf. next	1	2	1	-	0	51	0	113995	168	12	60	64
2007.01.01 - 2007.05.29	conf. next	2	363	7	-	7	46725	0	34729	2151	298	329	64445

Table 3.7: Statistics of classification of events from on-line algorithm in period 2006.06 - 2007.06

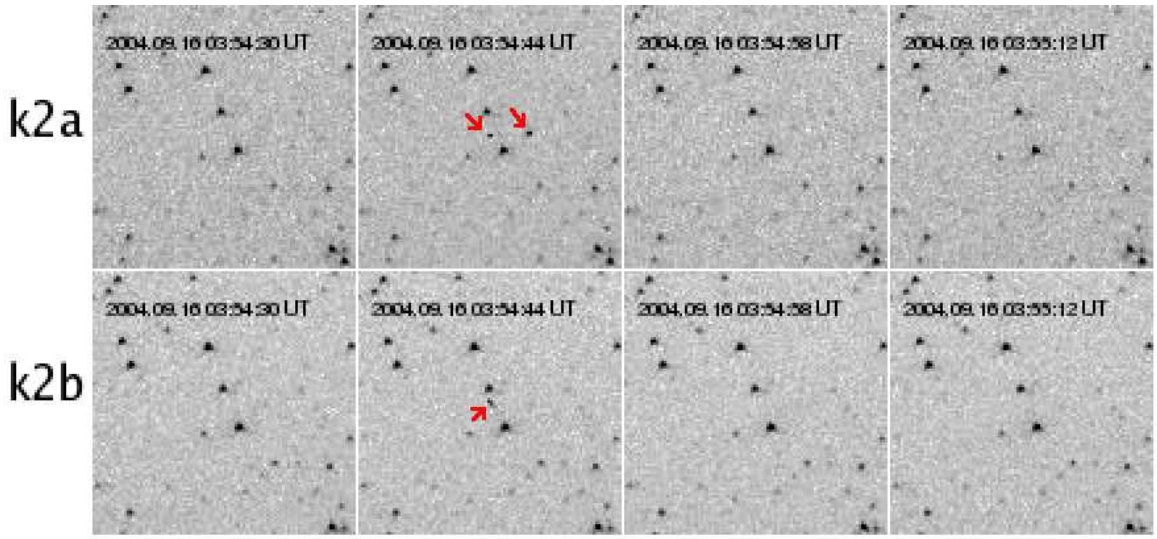

Figure 3.19: Rare example of coincidence of two cosmic ray hits

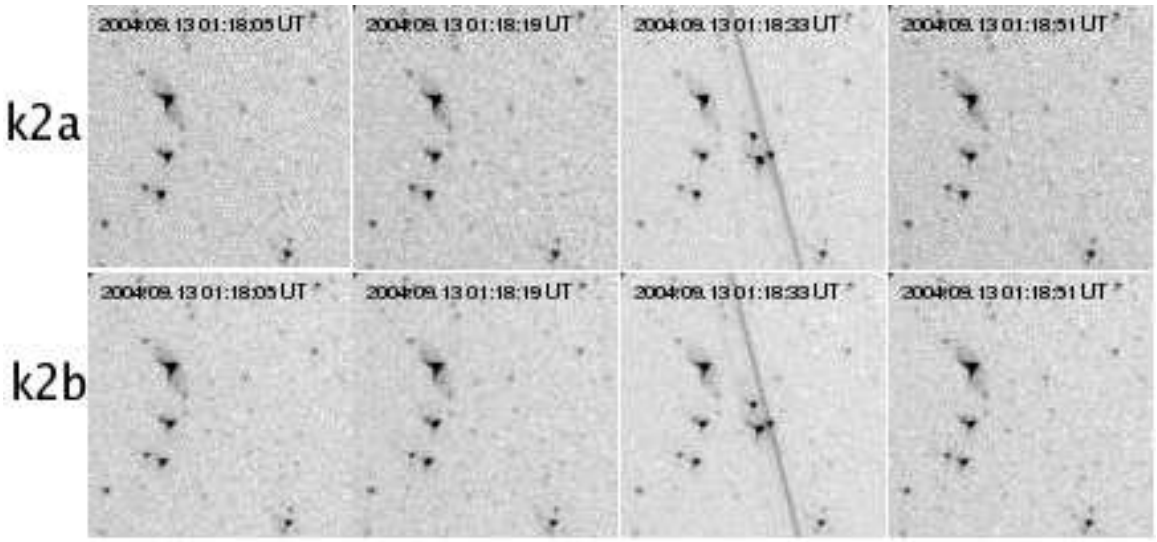

Figure 3.20: Plane-like background event

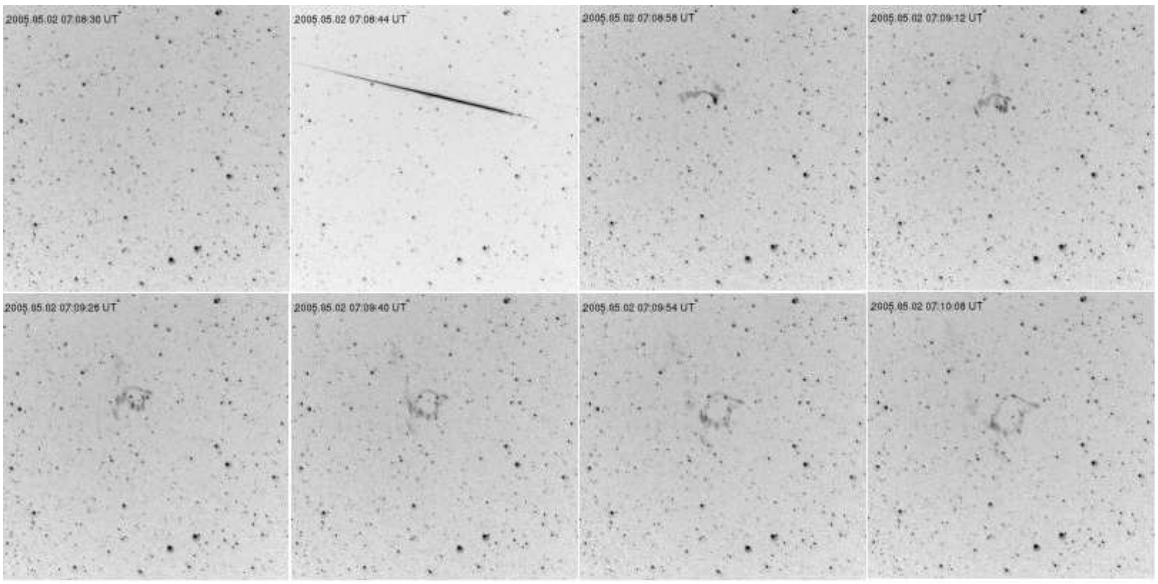

Figure 3.21: Meteor trace blown by the wind

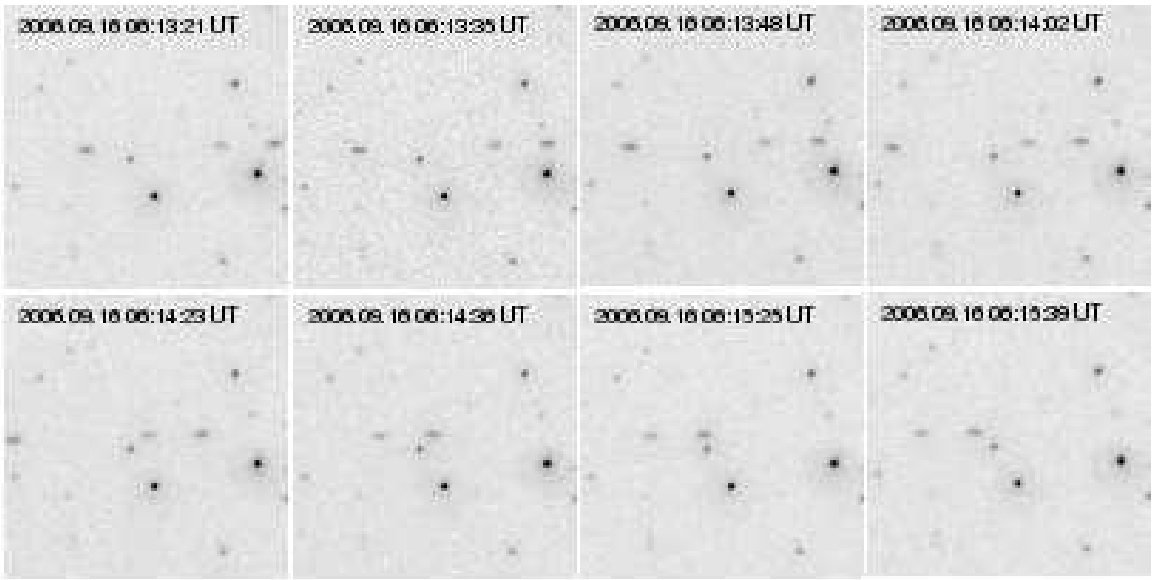

Figure 3.22: Flotilla of artificial satellites

3.2.6 Final verification of events

The final list of accepted events is very small. However, the prototype does not have a possibility to definitely reject all background events. The most difficult part of the background are flashing satellites. They are mostly rejected by catalog and track cuts described in the previous sections, but some objects are not cataloged and flash too rarely to be rejected by these criteria. Final events can be evaluated by several checks. In the case when object suddenly appears and remains visible in the next several images in the same position it is very probable that it is not a satellite. Assuming the object is Earth's satellite the following formula can be derived :

$$R > (\frac{\sqrt{G \cdot M} \cdot \delta t}{\alpha})^{(2/3)} \approx 16700 \cdot (\frac{\delta t_{sec}}{\alpha_{arcmin}})^{(2/3)}[km] \tag{3.10}$$

where α is the angular distance of the object in consecutive 2 images and δt is the time separation of images. They can be substituted in arcmin and seconds respectively if formula on the right is used. In the case of the prototype $\alpha \approx 0.6'$ which corresponds to a single pixel and $\delta t \approx 12s$ which corresponds to time separation of 2 subsequent images ($T_{exposure} + T_{dead}$= 10s + 2s). The minimal distance of the object visible on 2 consecutive images in the same position derived from these values is D $\approx$ 123 000 km. For comparison, the geostationary orbit is $R_{geostat}$ $\approx$42160 km (from the Earth center).

The distribution of the distance from the Earth to satellites in the catalog is shown in Figure 3.23. Peak from geostationary satellites is clearly visible. There are not many satellites more distant then 50000km, which supports "double image" events. However it is possible that these flashes are caused by spacecrafts on the long missions which are very far objects and can also reflect sun light towards the Earth, however the probability of such events is very small. The above check can not be applied to events visible only in a single image. For this class of events another check was implemented. Flashing satellites can only reflect sun light when they are outside the Earth shadow cone and not on the illuminated side of the Earth (Fig. 3.24).

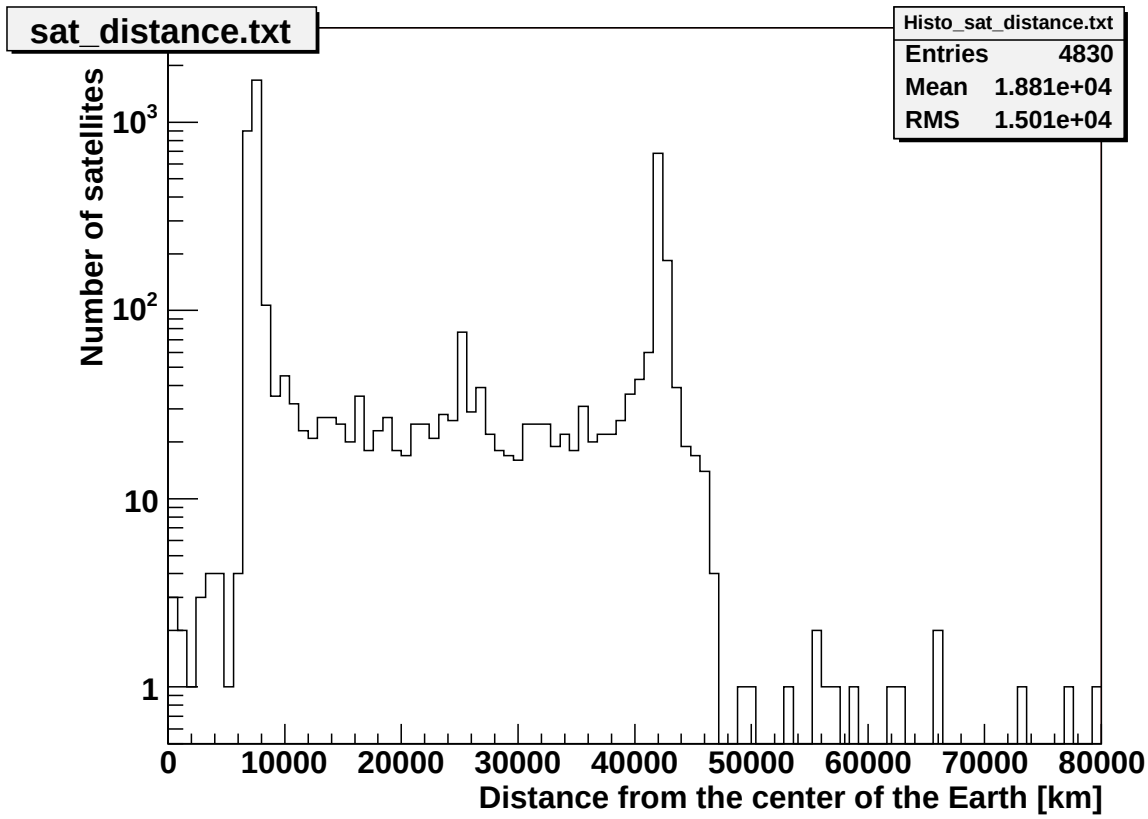

Figure 3.23: Distribution of distances of artificial satellites from the center of the Earth

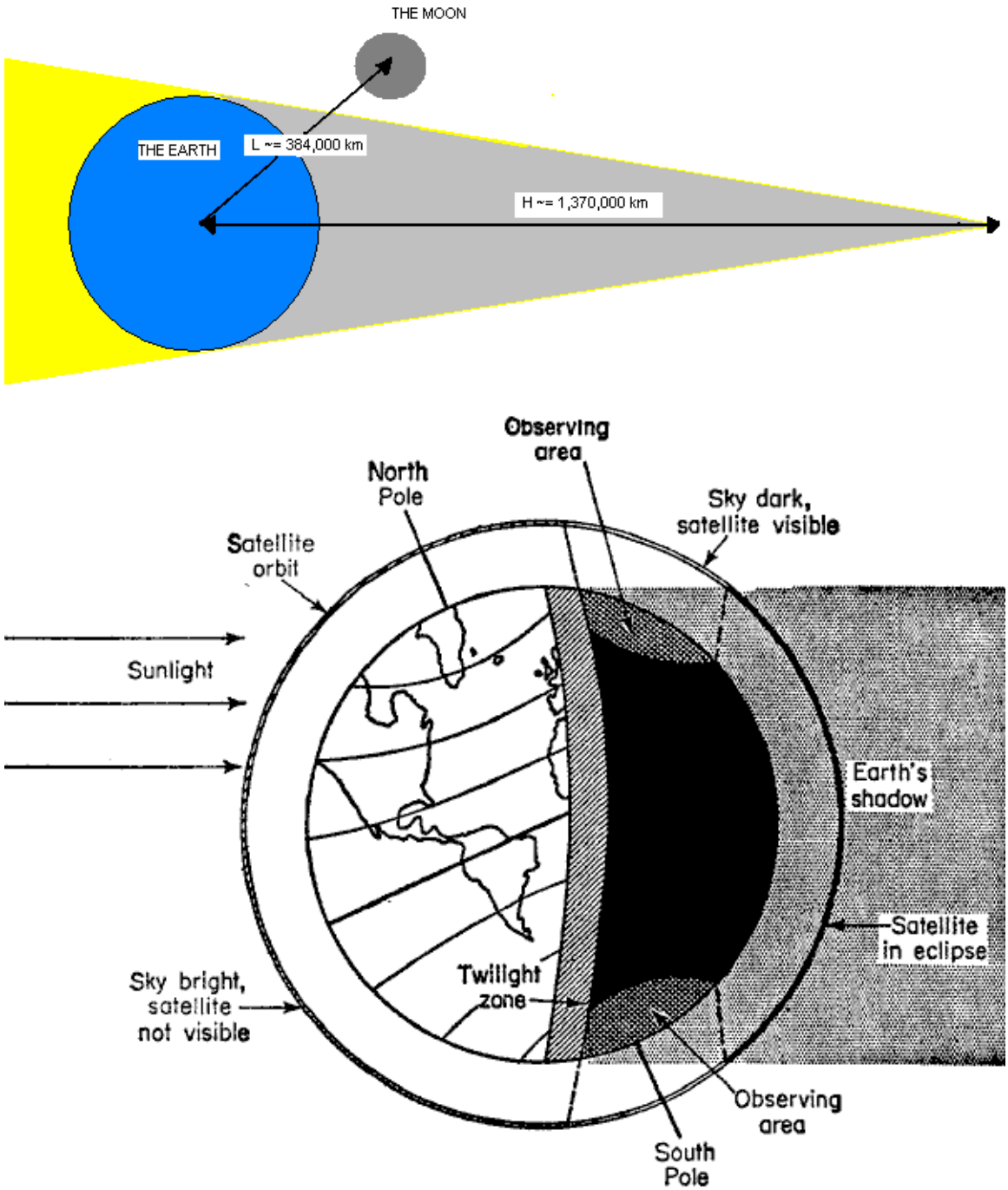

Figure 3.24: Positions where satellites can reflect sun light

Knowing the position and time of the flash it is possible to calculate the satellite's minimal distance from the Earth to be outside of the shadow cone which is required for the satellite to cause a flash-like event. The Earth shadow cone size is $H_{cone} \approx$ 1,37 mln km (Fig. 3.24), which is more then Moon's orbit size ($R_{moon} \approx$ 400 000km). In case any of the flash candidates would have $D_{cone} > R_{moon}$ it would be most probably an event of astrophysical origin.

3.3 Off-line data analysis

Off-line data analysis acts on data reduced and cataloged to the database. The reduction chain consists of several steps which will be described in detail in the next subsection. After this chain, star brightness measurements are stored in the database which is optimized for fast access. The structure of the database will be described in subsection 3.3.1.2. In the last subsection algorithms for detecting brightness increases and new object appearance will be described.

3.3.1 Reduction pipeline

The aim of the reduction procedure is to reduce raw data stored as images in `fits` files into essential data describing stars coordinates and brightness. This allows to reduce the amount of data by factor of ≈ 10 in the case of single images and ≈ 100 in the case of 20 averaged images.

3.3.1.1 Image reduction

Every image collected during a night is processed in the same way. Off-line data analysis described in this thesis was performed on data obtained by averaging 20 subsequent images. There are also other reduction pipelines for reducing single image and scan images (averaging 3 scan images), but almost all of the steps are the same. The main difference is that in pipelines acting on averaged images there is an additional step calculating average of specified number of images. Image reduction consists of the following steps :

- image averaging - it is present in reduction pipelines acting on averaged images. In the case of `aver20` pipeline 20 subsequent images are averaged

and in the case of pipeline `scan3` 3 subsequent images are averaged. Image coordinates are controlled and in case they change, average chain is stopped not to allow for averaging images from different positions. In the case of single image reduction the image averaging step is skipped.

- Dark frame subtraction - the reason for dark frame subtraction was already described in Section 3.1. As described in Section 2.2.2.3 in order to reduce fluctuations the dark image is calculated as a median of several dark images. This step allows to subtract signal offset produced by dark current and electronics. It also reduces the effect of hot pixels.

- Division by flat image - this step allows to correct for non-uniformity of the optics and differences between pixels amplifications. Standard way of finding this correction is taking images of uniformly illuminated field. It is usually the sky just after dusk or just before dawn, when the sky is bright and stars are not visible. An alternative way is to use uniformly illuminated screen. In case of wide field observations it is very difficult to obtain proper flat image. It is due to difficulty of obtaining uniformly illuminated field of size of FOV $\geq 20°$. The evening sky just after sunset is uniformly illuminated in scale of arc minutes, but in scale of degrees non uniformities due to sky gradient are significant. Due to this problem flat image is obtained by taking images of evening sky with the mount tracking switched off. After taking many images and calculating median image stars are eliminated, finally the image is normalized to one. This procedure requires collection of many (>=200) images so it is performed rarely and for most of the time the same flat image is used in analysis.

After the above operations the image is ready for photometry. The photometry is a procedure which finds stars in the image and determines their chip coordinates (x,y) and brightness. In the "Pi of the Sky" data analysis two photometry procedures are used depending on the type of reduction pipeline :

- ASAS photometry - aperture photometry adopted from ASAS experiment [13]. It is rather slow so cannot be used for reduction of all single images

from a night. It is used for photometry of 20 averaged images ($\approx 20 \cdot 12 =$ 240 sec timescale) and in reduction of scan images (3 images averaged).

- Fast photometry - it is fast aperture photometry algorithm. Simple aperture is used to calculate star brightness. This photometry is used in photometry performed on-line every 80 sec and in reduction of all night images.

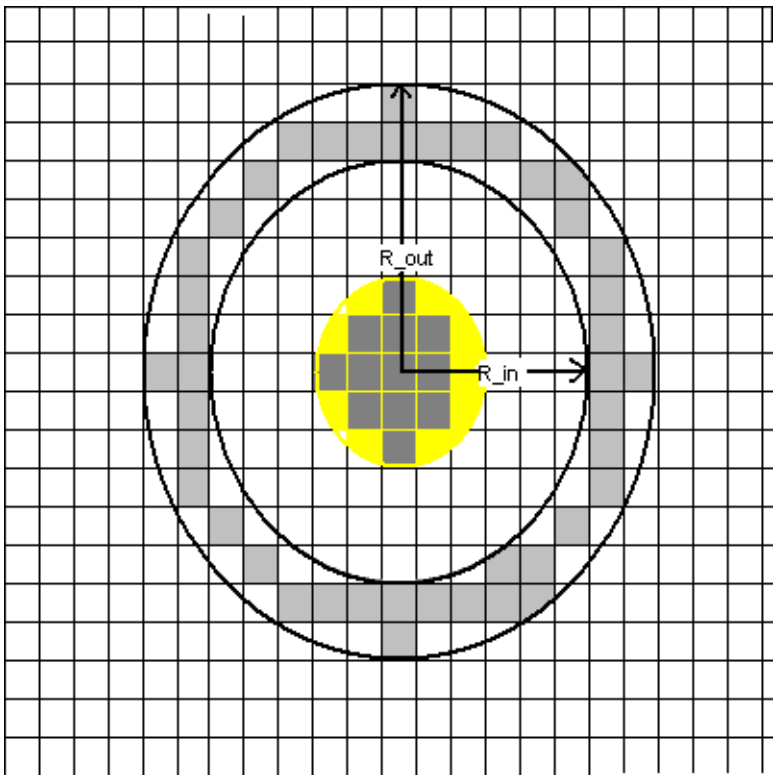

Figure 3.25: Aperture used in fast photometry algorithm

Aperture used for brightness calculation is shown in Figure 3.25. Final brightness of star is determined as :

$$I = \sum_{i}^{N_+} P_+ - N_+ \cdot B_{sky} \tag{3.11}$$

where B_{sky} stands for average value of sky background near analysed star. This value is obtained as median value of pixels in gray ring around the star (Fig. 3.25). P_+ denotes signal pixels, shown as dark gray pixels inside

the star, they form 3x3 square around the maximum brightness pixel with 3 most bright pixels contiguous to the square sides. This procedure is not good for dense fields where sky background can be calculated incorrectly due to stars entering the "background ring" and also $\sum_i^{N_+} P_+$ is affected by overlapping stars. Instrumental magnitude is calculated from the formula :

$$m_\pi = -2,5log(I) \tag{3.12}$$

Star coordinates (x,y) are determined as centroid of cluster of pixels according to the formula :

$$X_{star} = \frac{\sum_{cluster} x_i \cdot V_i}{\sum_{cluster} V_i} \ , \ Y_{star} = \frac{\sum_{cluster} y_i \cdot V_i}{\sum_{cluster} V_i} \tag{3.13}$$

Cluster of pixels is determined as pixels around the star which satisfy P(x,y) $> T_{cluster} = 3.5 \cdot \sigma$, but it depends on the photometry parameters.

Both photometry procedures write resulting list of stars with (x,y) coordinates and magnitudes to output `mag` files. The format of this files is similar to `fits` format. They consist of header which is taken from `fits` header with some additional fields added, after the header section, list of stars in binary format is written. The `mag` files are input files for astrometry procedure.
This procedure finds transformation T:(x,y) -> (λ, δ), the transformation T is described by the following formula :

$$\lambda = \sum_{i,j<=O} P_{ij} \cdot x^i \cdot y^i \ , \ \delta = \sum_{i,j<=O} Q_{ij} \cdot x^i \cdot y^i \tag{3.14}$$

Where O is the order of the transformation, in current configuration O=4 and due to this fact coefficients $P_{14}, P_{23}, P_{24}, P_{32}, P_{33}, P_{34}, P_{41}, P_{42}, P_{43}, P_{44} = 0$ and also corresponding Q coefficients are zero. It allows to calculate equatorial coordinates for any chip coordinate (x,y). Astrometry requires input information about image center position, pixscale [1] and rotation angle of the image with respect to the celestial coordinates, these settings are read from header of `mag` file.

[1] pixscale is an angular size of the CCD pixel

The astrometry procedure was adopted from ASAS experiment, it is an iteration procedure where stars in `mag` file are matched against catalog stars in given position in the sky. Star catalog currently used in the procedure is based on TYCHO catalog, however any star catalog can be used instead. The procedure consists of the following main steps :

- loading of `mag` file
- read stars from the reference star catalog
- estimation of shift from the expected position $(\lambda, \delta)_{mount}$ and real position (λ, δ) using the correlation function between image stars and stars in the catalog

After the above initialization steps the iterational procedure begins, every iteration consists of the following main steps :

- recalculation of reference and image stars coordinates to standard coordinates (with respect to image center)
- matching of image stars from `mag` file against reference stars
- determination of transformation parameters by using the method of singular value decomposition (SVD).
- check the convergence condition requiring astrometry error $\delta\alpha < \delta\alpha_{MAX}$
- recalculation of image center coordinates $(\lambda, \delta)_{center}$

The iteration steps are repeated until the convergence conditions are satisfied. In the case astrometry procedure converges, for all stars in the `mag` file coordinates (λ, δ) are calculated from the formula 3.14. The results are saved to `ast` file which consists of same information as `mag` file with additional fields for (λ, δ). All night images are processed in the same way and every sky image (`fits` file) has a corresponding `ast` file.

3.3.1.2 Star Catalog

The star catalog is developed as relational PostgreSQL database [20]. The database structure is shown in Figure 3.26. It consists of tables described earlier in Section 2.2.2.5 with additional tables :

- **Star** - this table contains all objects observed by given camera. Same physical star can be observed by many cameras so same physical star has N>=1 records in the table **Star**, where N stands for number of cameras by which this star was observed.
- **Measurements** - this table stores information on every observation of the star. It is linked to table Star by reference field star, it is also linked by field id_frm to image on which the star was observed.
- **SuperStar** - it is a table containing real physical stars. In case star is observed by different cameras it has multiple records in table **Star**, but only one record in the table **SuperStar**. Every record in the table **Star** is linked to corresponding **SuperStar** record by field `sstar_id`. The relation between **SuperStar** and **Star** table is one to many.
- **ObsFieldStat** - statistical table containing information on number of images collected for a specific field
- **Field_Def** - definitions of fields observed by the system

Star catalog database can be huge, after year of data collection it can reach 50-200 GB, so in order to be efficiently used it must be optimized. An important element of the database structure are indexes. They allow searching indexed fields by binary search algorithm. The most important database queries were optimized by creating indexes on fields used in conditional statements. Another optimization performed on the database is placing measurements records for a given star in the same physical location on the disk. This is very important for fast reading of star light curves. This optimization is executed by PostgreSQL command CLUSTER which must be called every time new data is added to the database. Also table **Star** is optimized by CLUSTER command according

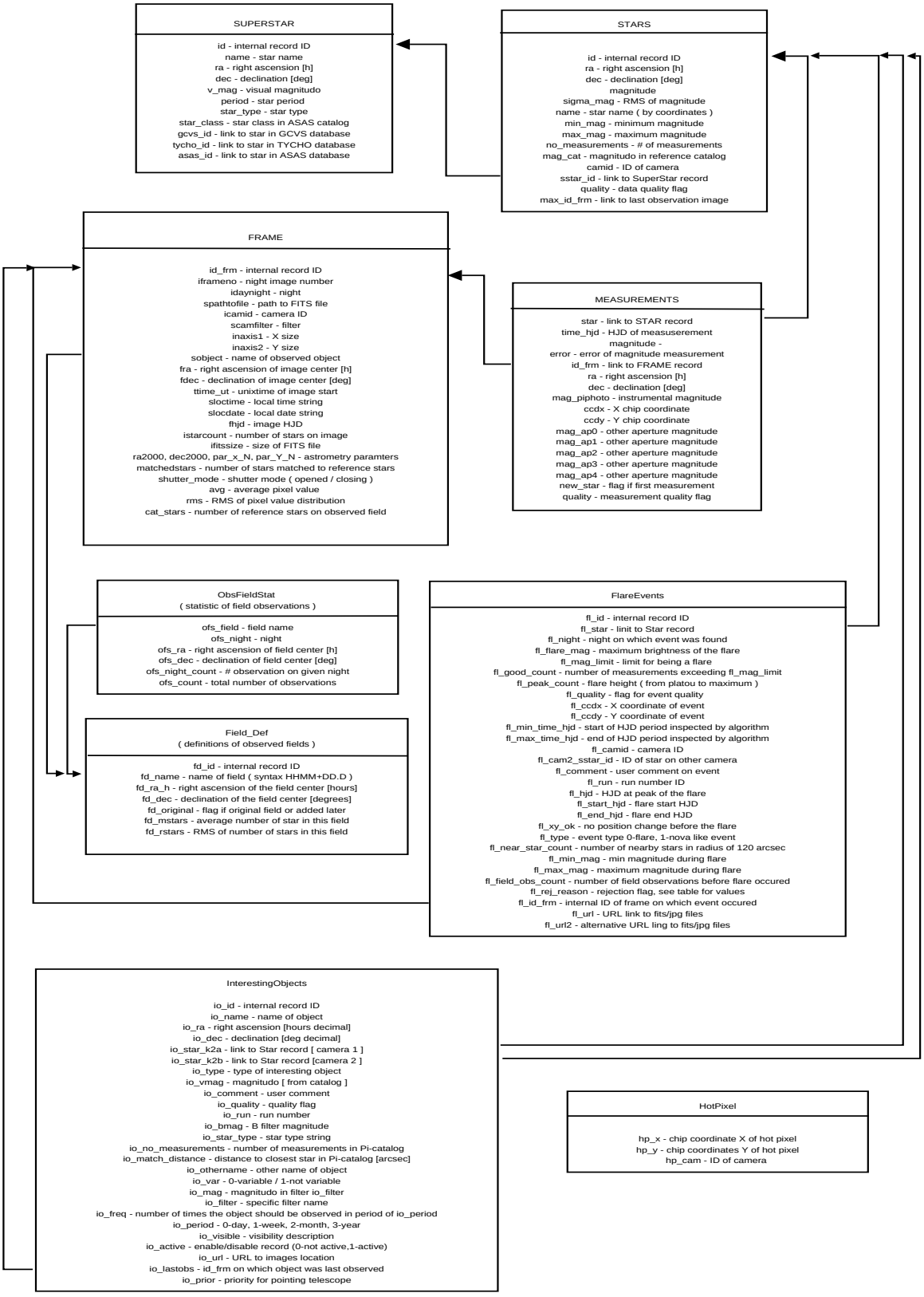

Figure 3.26: Structure of star catalog database

to the celestial coordinates in order to optimize reading of stars from database during cataloging. The most important indexes are listed in Table 3.8. There are also database optimizations on PostgreSQL server configuration level which set parameter value proper for huge database.

Index Name	Table	Indexed Fields	Clustered
stars_id_index	STARS	ID	no
stars_ra_dec_index	STARS	ra,dec	yes
stars_dec_index	STARS	dec	no
superstar_gcvs_id_index	SUPERSTAR	gcvs_id	no
measurements_star_index	MEASUREMENTS	star	yes
measurements_id_frm_index	MEASUREMENTS	id_frm	no

Table 3.8: Indexes most important for optimizing star catalog

3.3.1.3 Cataloging procedure

Reduced data is a set of `ast` files, these files have to be loaded to the database structure described in the previous section. This task is called cataloging and is performed by `piaddast2` program. Except of loading data to the database this program normalizes star magnitudes according to V filter magnitudes in catalog of reference stars. The block diagram of the `piaddast2` program is shown in Figure 3.27. Generally, cataloging procedure reads stars already observed before from the database and matches new observations to these stars. Subsequent `ast` files are organized in the memory until observation field changes which triggers dump of data from memory to the database and selection of stars for new position. Every `ast` file is processed in the following way :

- Read next `ast` file, determine range of celestial coordinates in `ast` file $(\lambda^{ast}_{min}, \delta^{ast}_{min}) - (\lambda^{ast}_{max}, \delta^{ast}_{max})$
- Save image header information to database table FRAME.
- Check if all stars in `ast` file have coordinates in the range $(\lambda^{prev}_{min}, \delta^{prev}_{min}) - (\lambda^{prev}_{max}, \delta^{prev}_{max})$. In case there are stars from outside this range program dumps stars and measurements stored in memory to `sql` files and loads `sql` files to

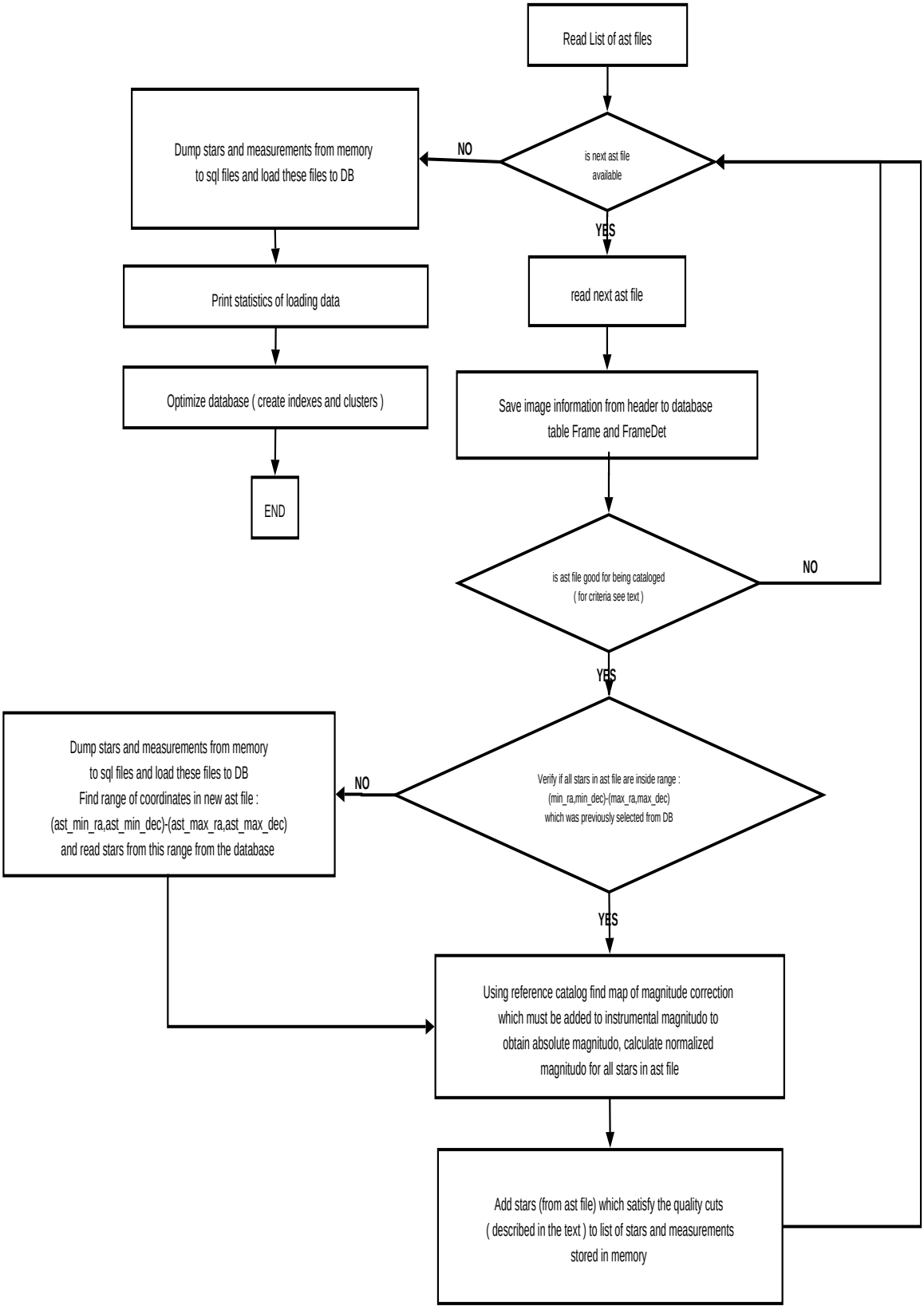

Figure 3.27: Block diagram of the cataloging program

database, then selects stars from database with coordinates matching range of coordinates in new `ast` file (as determined in first step)

- Check if the image represented by the `ast` file is good enough to be cataloged, the following criteria are checked :

  - number of images used for averaging (in the case of cataloging averaged images) for 20 averaged images the lower limit is 10 images and for `scan3` pipeline lower limit is 3.

  - check if $N_{star}^{image} >= N_{star}^{MIN}(= 5000)$, in case number of stars is lower it means that is is probably cloudy data (Fig. 3.28) .
  - check if number of stars is not too high : $N_{star}^{image} <= N_{star}^{MAX}(= 70000)$ (Fig. 3.28)
  - check if average astrometry error satisfies $A_{err} <= A_{err}^{max}(= 0.3)$ (Fig. 3.29)

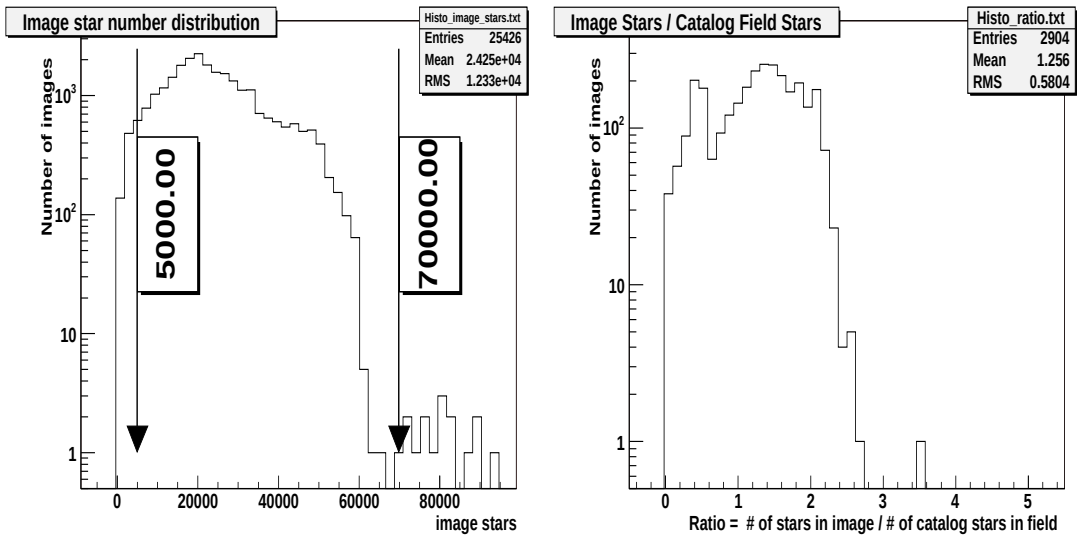

Figure 3.28: Distribution of number of stars on image (left plot) and ratio of number of stars on image to number of catalog stars on observed field (right plot)

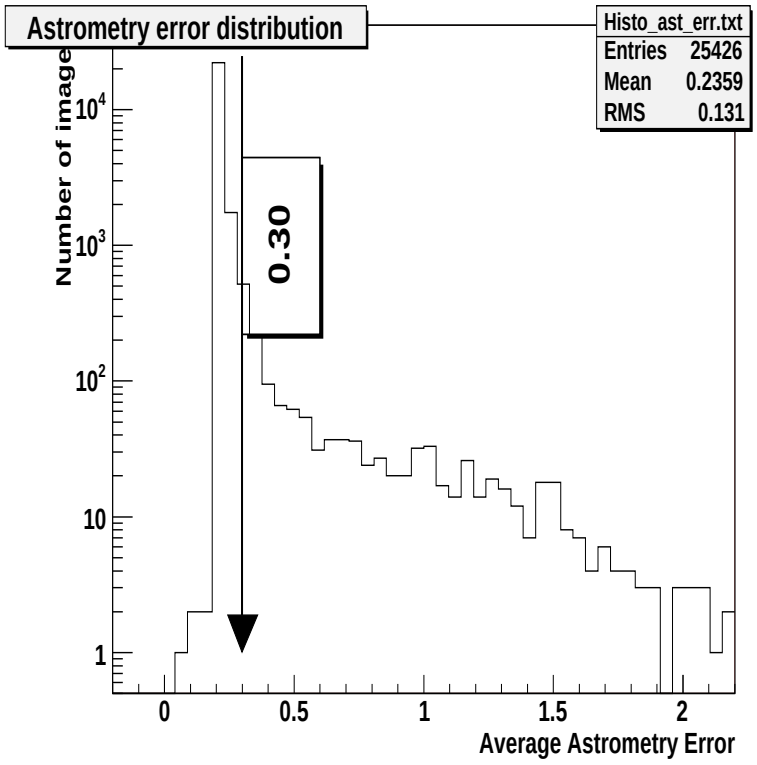

Figure 3.29: Distribution of average astrometry error on image

In case all criteria are satisfied `ast` file is accepted and stars are cataloged, otherwise it is skipped.

- Each star in ast file is examined against certain quality cuts :

  1. Star altitude is required to by h>h_{min}=15°, in order to reject measurements close to horizon which are of very poor quality

- Star magnitudes are normalized by comparing with the catalog of reference stars. Matching allows to create correction image (Fig. 3.30) which can be used to normalize magnitudes of all stars in the image. The normalization is performed in the following way :

  – catalog of reference stars is read to memory

  – for each star in `ast` file program finds a corresponding star in reference catalog. In case it is found field mag_{cat} is filled with star magnitude from the reference catalog. Correction for this star is calculated as :

  $$\delta_{mag} = mag_{cat} - mag_{\pi} \tag{3.15}$$

  this is value which must be added to instrumental magnitude to obtain normalized one. Typically about 50% of `ast` stars have corresponding star in the currently used reference catalog (TYCHO).

  – after matching procedure, corrections values are known for pixels where stars matched to catalog stars were present. In order to calculate correction for each star in the image, correction values are calculated for all pixels of the image by extrpolating values determined for reference stars. Correction value C(x,y) for each pixel in the image is calculated as average of corrections for nearby reference stars weighted by distance to pixel where this star was observed :

  $$C(x,y) = \sum_{ref.stars\ :R<R_{max}} w(\sqrt{(x-x_i)^2+(y-y_i)^2}) \cdot C_(x_i, y_i) \tag{3.16}$$

An example of resulting correction image is shown in Figure 3.30.

– normalized magnitude is calculated for every star on the image according to formula :

$$mag_{norm} = mag_{cat} + C(x, y) \tag{3.17}$$

where (x,y) are chip coordinates of the normalized star.

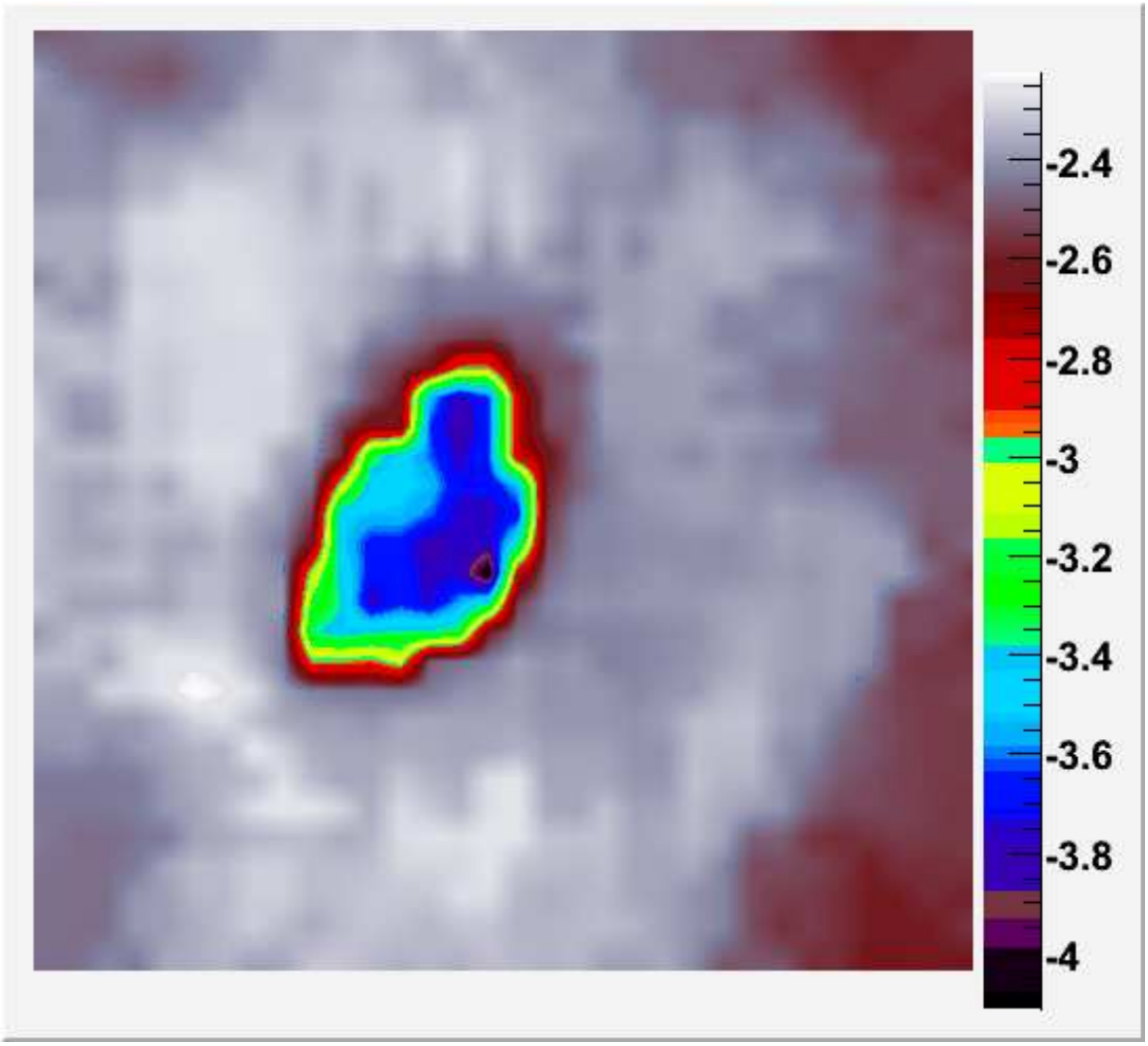

Figure 3.30: Correction image obtained in cataloging to normalize instrumental magnitudes to reference catalog

- Stars from the `ast` file are matched to stars read from the database (see step 3) and kept in the memory. In case given star is found in the list of stars in the memory the star measurement from current `ast` file is linked to the list of measurements of this star. In case this star was not yet observed, it is added to the list of stars in the memory and flagged as a new object in the catalog. After all stars in `ast` file are processed next `ast` file is read and process starts from the beginning (step 1). The procedure continues until all `ast` files are processed.

After the above steps all good quality data in `ast` files is saved to the database. However, there are several technical details which should be mentioned here. First of all, as can be seen in Figure 3.26 the **Star** table has statistical fields like magnitude, dispersion of magnitudo (`sigma_mag`), `no_measurements` which are not updated during data loading, but they are very useful in further data analysis. These fields are recalculated by special `pg/sql` procedure after the data is loaded to the database. The second step is optimization of the database, before loading of night `ast` files most of the indexes on tables **Star** and **Measurements** must be dropped in order to load data efficiently. After loading is finished these indexes are re-created. After loading new data its location on the disk must be re-organized in order to provide fast access. It is realized by PostgreSQL command CLUSTER. After all optimizations and recalculations are finished the database is unlocked and off-line algorithms can be executed on new night data.

3.3.1.4 Efficiency, purity and precision of observations

Efficiency and purity of reduction and cataloging was tested by the following procedure :

1. Initialization of database - star catalog database was initialized with stars from TYCHO-2 star catalog
2. Initial stars were flaged as TYCHO-2 stars in the database
3. Cataloging data to be tested was loaded to catalog initialized with TYCHO-2 stars

After this steps database was filled with data and there was an easy way to distinguish stars which are present in TYCHO-2 star catalog from those which are not and were observed only on "Pi of the Sky" images. The test was performed on two samples of images one from night 20070425 taken with shutter in normal mode and second sample were images from night 20070512 taken with shutter permanently opened. Total star idendification efficiency which is defined as :

$$\epsilon_{\pi-red} = \frac{N_{TYCHO2/PI}}{N_{TYCHO2}} \tag{3.18}$$

where $N_{TYCHO2/PI}$ stands for number of TYCHO-2 stars identified by "Pi of the Sky" photometry and N_{TYCHO2} is number of TYCHO-2 stars in the observed field. The Table 3.9 shows total efficiencies for single image and set of images.

Night	Number of images	Magnitude range	Efficiency
2007.04.25/26	1	0 - 12	0.74
2007.04.25/26	41	0 - 12	0.84
2007.04.25/26	41	0 - 15	0.77
2007.05.12/13	1	0 - 12	0.78
2007.05.12/13	41	0 - 12	0.87
2007.05.12/13	41	0 - 15	0.82

Table 3.9: Total efficiencies for night 2007.04.25/26 (data collected with shutter in normal mode) and 2007.05.12/13 (data collected with permanently opened shutter)

The Figure 3.31 shows the efficiency of star identification in function of star brightness for data collected with permanetnly opened shutter (2007.05.12) and shutter in normal open/close mode (2007.04.25).

The Figure 3.32 shows efficiency in function of number of field observations.

The Figure 3.33 and 3.34 show efficiency and background in function of chip coordinates. It is clear that efficiency and background drop in the corners of the CCD chip and reach the highest values in the center of the chip which is due to the fact that less light reaches corners of CCD because of properties of optical system.

The situtation in sligthly more difficult in the case of purity determination. The objects in the TYCHO-2 star catalog are only stars. Objects observed by

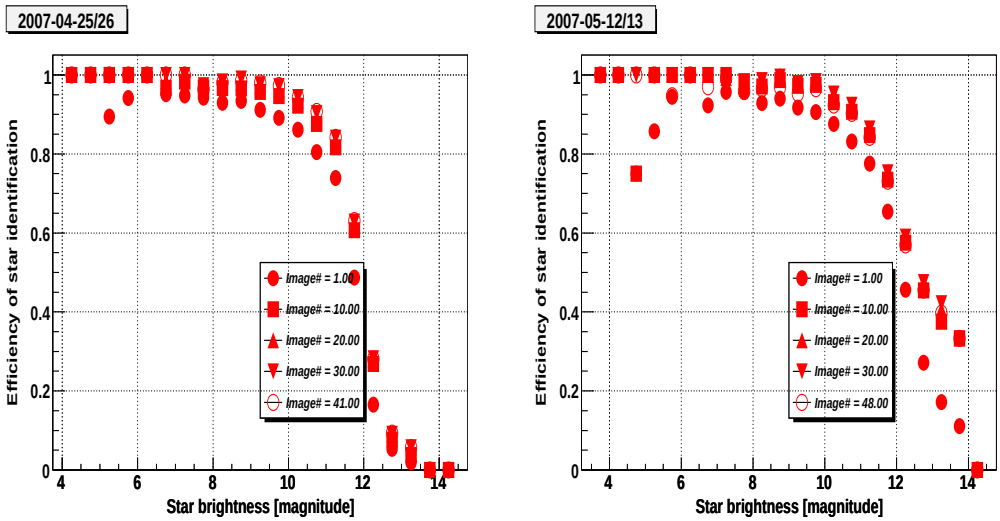

Figure 3.31: Efficiency of TYCHO-2 stars identification in function of magnitude for data collected during night 2007.04.25/26 with shutter in normal (open/close) mode (left plot) and data collected with shutter permanently opened during night 2007.05.12/13 (right plot)

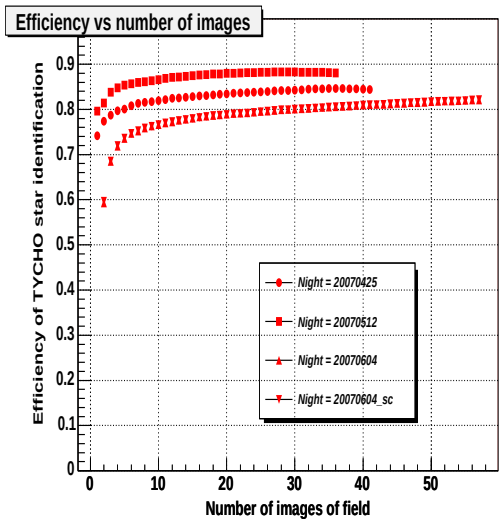

Figure 3.32: Efficiency of TYCHO-2 stars identification in function of number of field observations for 3 different nights and after applying correction of shutter opened effect.

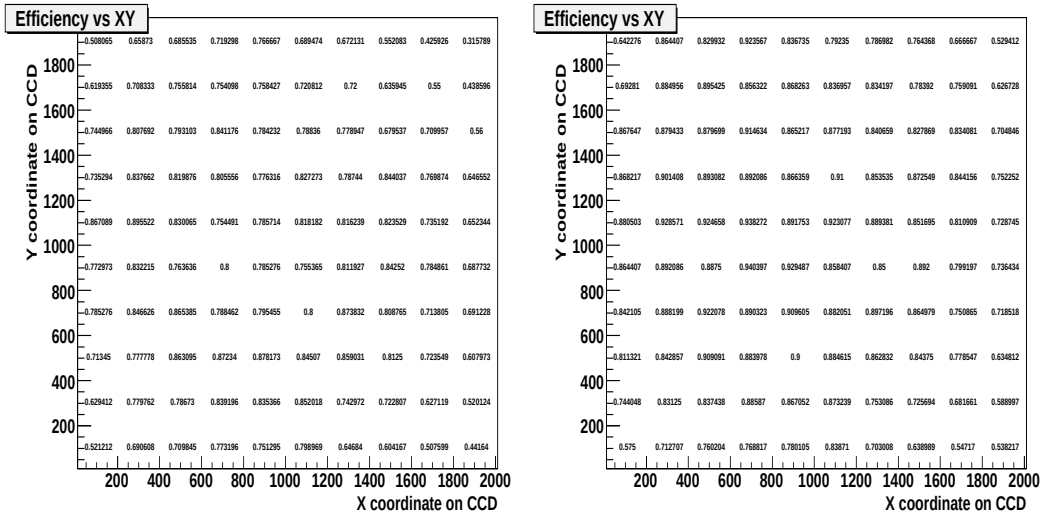

Figure 3.33: Efficiency of TYCHO-2 stars identification in function of star position (x,y) on the CCD chip. Left plot shows efficiency of single average of 20 images from night 2007.04.25/26 collected with shutter in normal mode and right plot shows efficiency on single image from night 2007.05.12/13 collected with permanently opened shutter

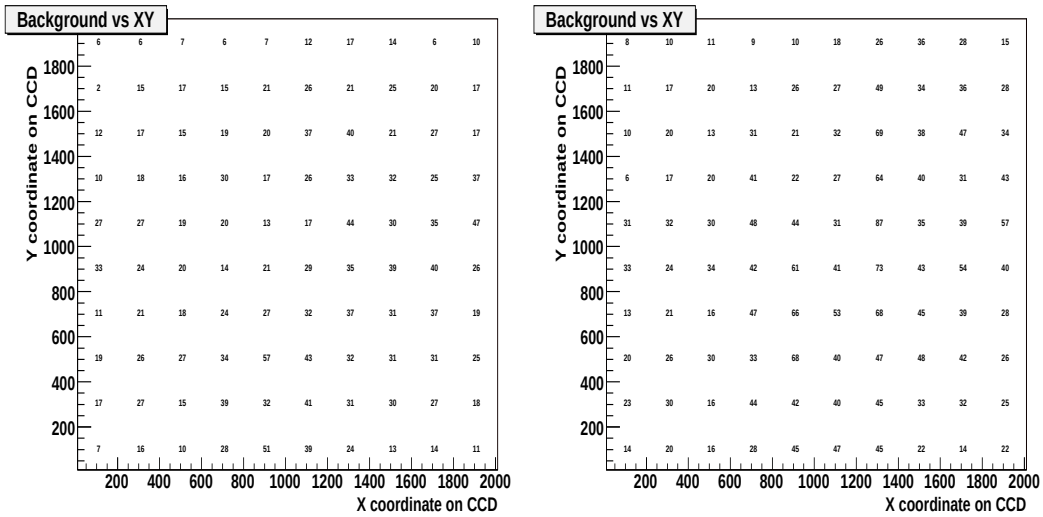

Figure 3.34: Number of observed objects not exisiting in TYCHO-2 catalog in function of star position (x,y) on the CCD chip. Left plot shows objects found on single average of 20 images from night 2007.04.25/26 collected with shutter in normal mode and right plot shows objects found on single image from night 2007.05.12/13 collected with permanently opened shutter

"Pi of the Sky" which are not present in the TYCHO-2 catalog are not only background events. They can be non-star objects like galaxies etc. However, after many observations of given field all objects in the range of the telescope should already be observed and new objects appearing in the catalog after many field observations can be considered as background. The Figure 3.35 shows number of new objects added to star catalog on subsequent images of the same field and also total number of new objects in function of number of field observations is shown. The data presented on this plot was collected during night 2007.04.25/26 with shutter in normal open/close mode. The steps at frame id=40 and id=34 are due to trace of satellite or plane which is shown on Figure 3.36. It is clear that after many (>40) field observations number of new objects added to star catalog on every image is very small and equals few events per image, unless background event (satellite, plane etc) is observed. For comparison the same plots for data collected on the same field with shutter permanently opened (night 2007.05.12/13) is shown in Figure 3.38.

Figure 3.37 shows cumulative number of events added to catalog in function of image number for couple of fields observed on many nights. These plots indicate that the minimum number of field observations to consider new objects as potential nova candidate is 20-30.

On Figure 3.39 results for field (0851-70) are shown, the number of background events added to catalog on every image is much larger. Most of these objects are artefacts coming from photometry of strip of charge which appears when images are collected with permanently opened shutter (see Fig. 3.40). On field 0851-70 number of stars (33000) is much higher then in field 0800+20 (18000) which causes much higher number of stars causing significant "open shutter" strips.

Open shutter causes the charge to be collected also in pixels above the readout pixel during chip readout (Fig. 3.40). It is possible to reduce this effect by subtracting from every pixel fraction of values in pixels below. The image before and after the correction is shown in Figure 3.40. The data from night 2007.06.04/05 was corrected and cataloged, number of new objects in function of image number is shown in Figure 3.41. The number of new objects in every image is reduced approximately by a factor of 2.

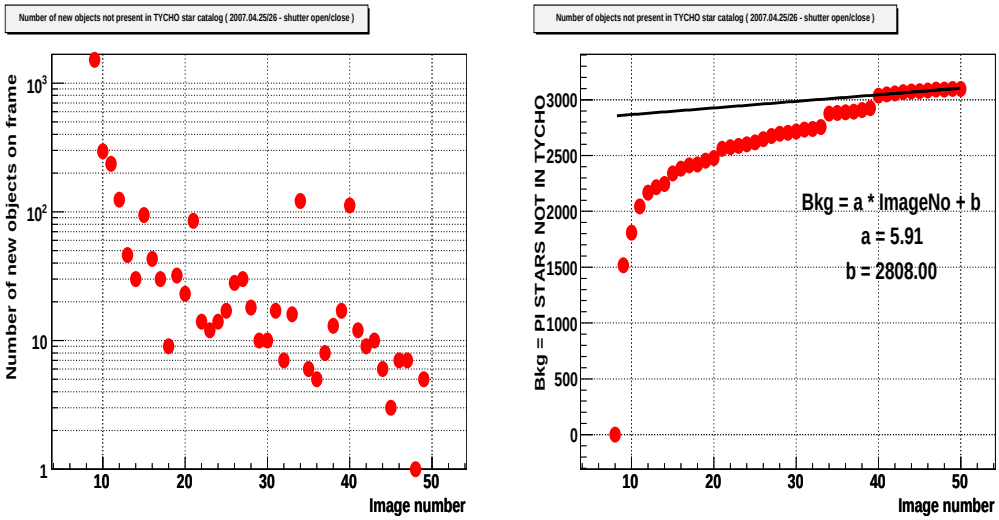

Figure 3.35: Number of new objects added to catalog on subsequent images (left plot) and total number of new objects added to catalog in function of image number right plot. Data was collected during night 2007.04.25/26 with shutter in normal open/close mode, on field 0800+20 (average number of stars 18000)

The efficiency of star identification can be paramterized in function of number of image stars. Figures 3.42 and 3.43 show star identification efficiency in function of number of stars in image for star catalog created from image averaged over 20 and star catalog created from single images (respectively) . These plots were obtained for single sky field 0800+20.

However as can be observed from Figures 3.42 and 3.43 this is not the best paramterization. Much better parametrization is efficiency in function of ratio R_{cat}, defained as :

$$R_{cat} = \frac{N_{image\_stars}}{N_{cat\_stars}} \tag{3.19}$$

where N_{image_stars} is number of stars in image and N_{cat_stars} is number of stars found in reference catalog (used in cataloging) in the observed field. Star identification efficiency in function of ratio R_{cat} is show in Figure 3.44, this efficiency was calculated for different sky fields and different nights. As expected this dependency is linear. The average efficiency of star identification on single 10s exosure was determined by averaging efficiency from many single images col-

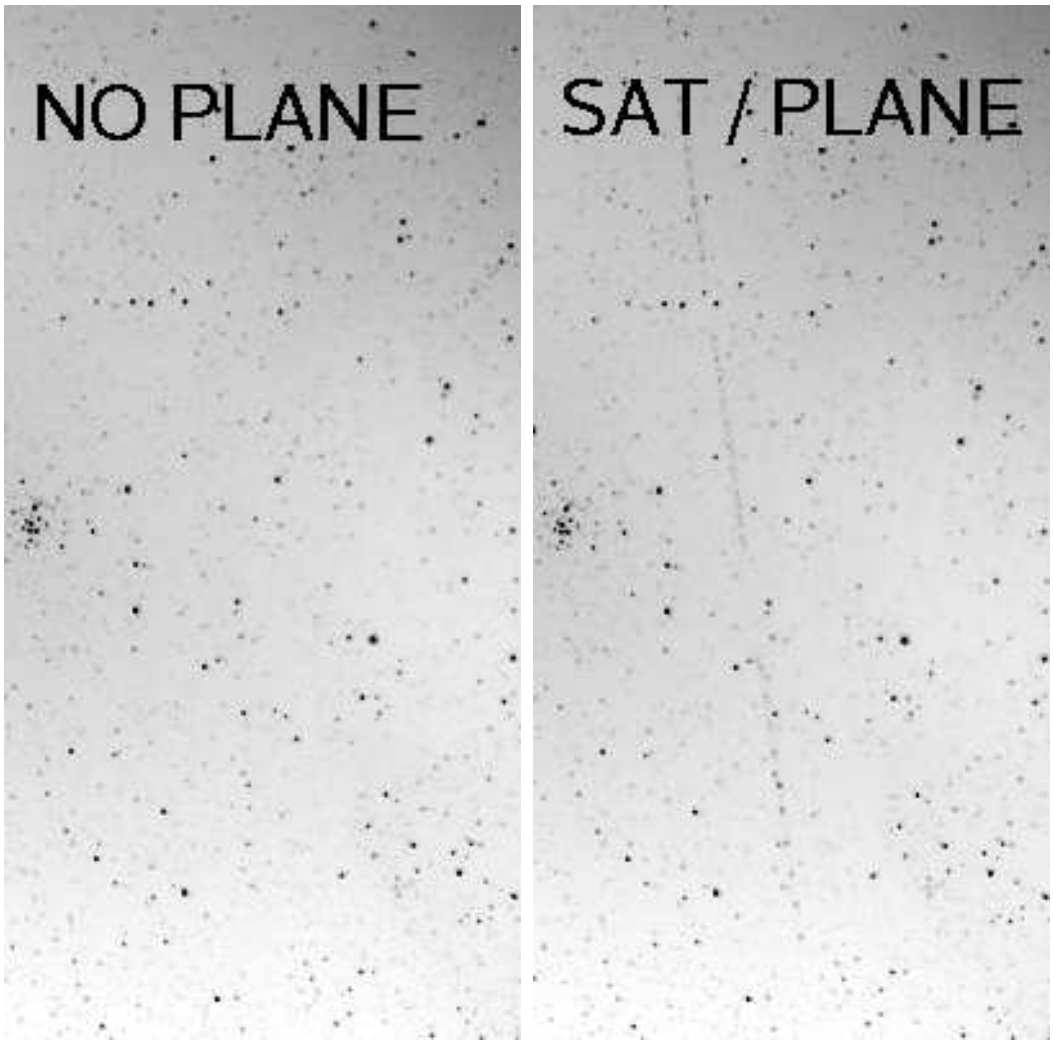

Figure 3.36: The reason for step visible on previous image at frame id=34. Part of image without satellite (or plane) trace is visible on left plot and with the trace on right plot. This trace causes addition of new objects to star catalog

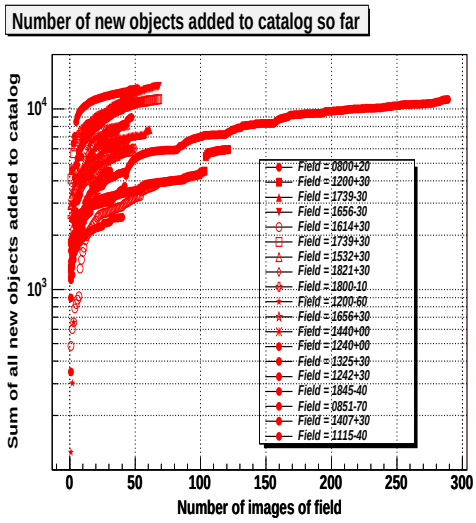

Figure 3.37: Number of new objects added to catalog from the begining to given frame number. Data for different field collected on many nights is shown

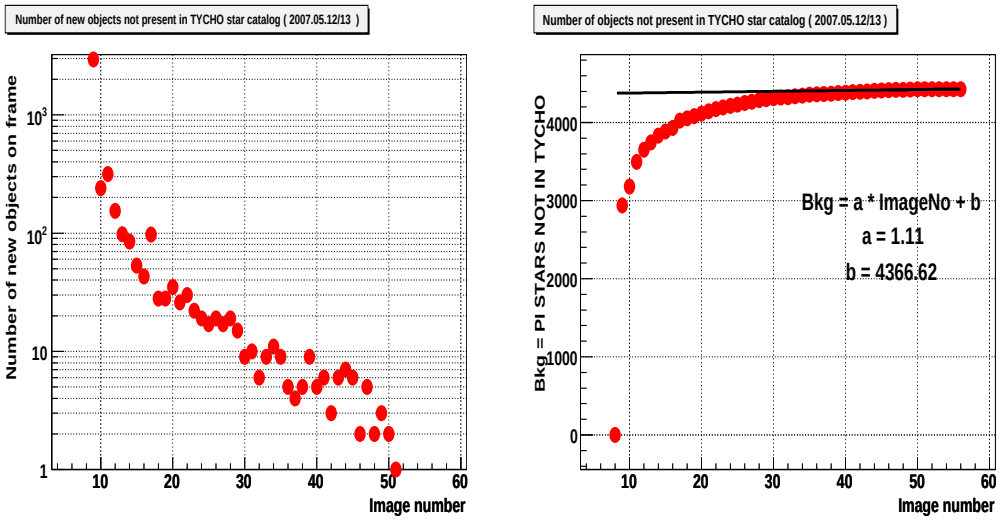

Figure 3.38: Number of new objects added to catalog on subsequent images (left plot) and total number of new objects added to catalog in function of image number right plot. Data was collected during night 2007.05.12/13 with shutter in normal open/close mode, on field 0800+20 (average number of stars 18000)

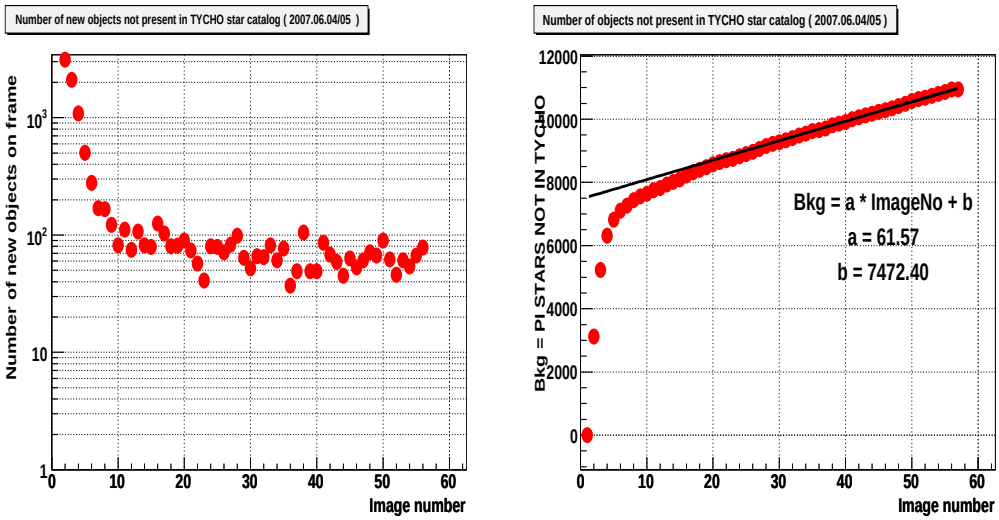

Figure 3.39: Number of new objects added to catalog on subsequent images (left plot) and total number of new objects added to catalog in function of image number right plot. Data was collected during night 2007.06.04/05 with permanently opened shutter, on field 0851-70 (average number of stars 33000)

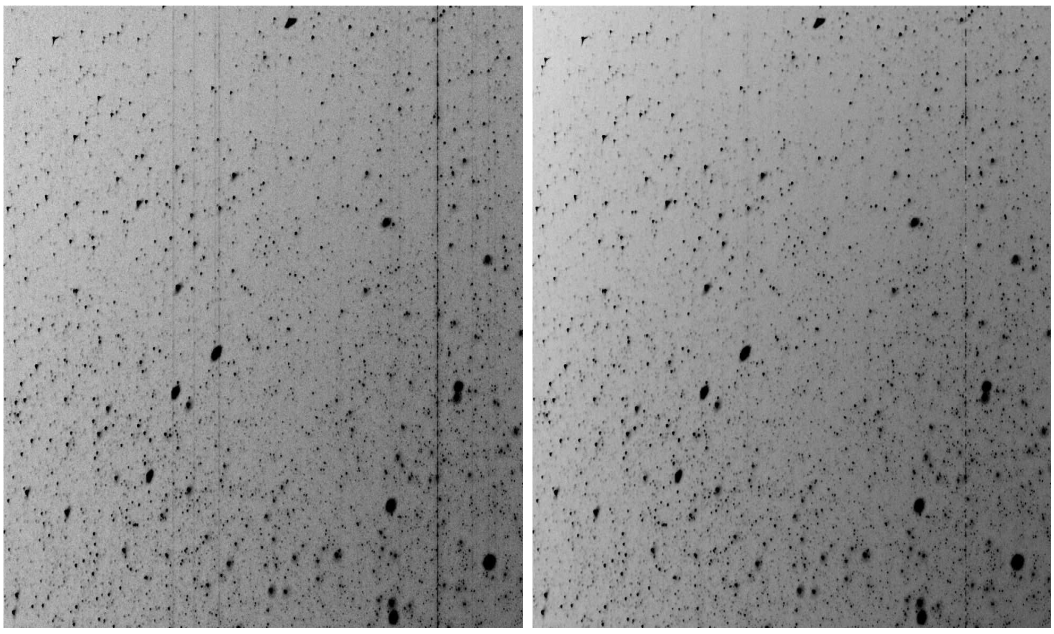

Figure 3.40: Image taken with permanently opened shutter before correction (left image) and after correction (right image)

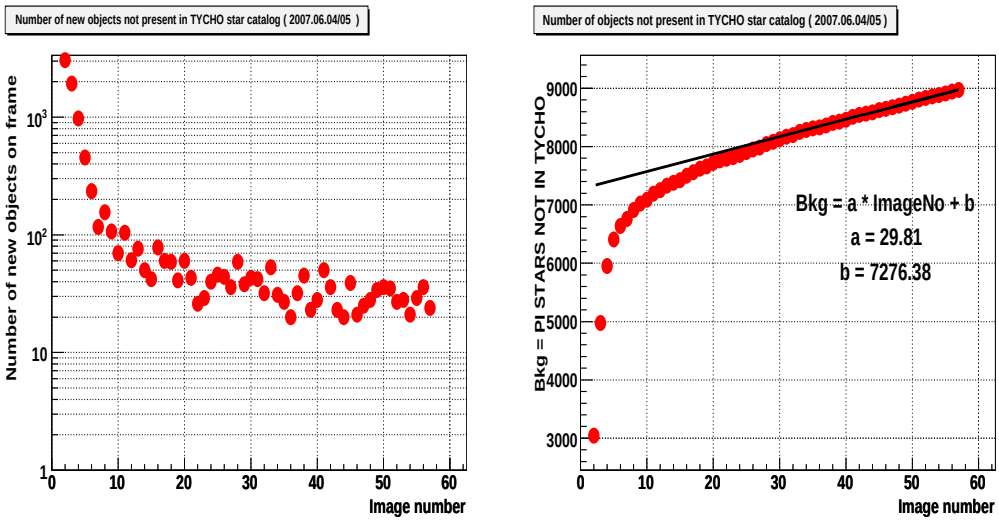

Figure 3.41: Number of new objects added to catalog on subsequent images (left plot) and total number of new objects added to catalog in function of image number right plot. Data was collected during night 2007.06.04/05 with shutter in normal open/close mode, on field 0851-70 and correction of opened shutter effect was applied

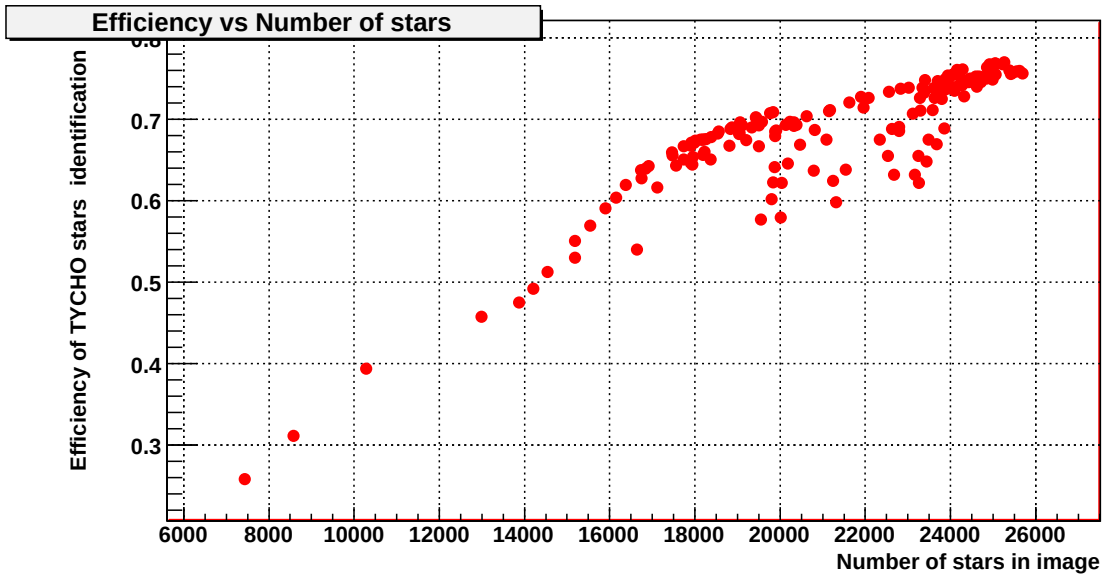

Figure 3.42: Efficiency of star identification in function of number of stars in image for sky field 0800+20. Star catalog obtained from images averaged over 20

lected on different nights (every 25 image from single night was cataloged). The resulting average effieciency is :

$$\epsilon_{star} \approx 0.49 \tag{3.20}$$

which multiplied by average efficiency of on-line algorithm gives average efficiency of on-line algorithm on the level of $\epsilon_{on-line} \approx 35\%$.

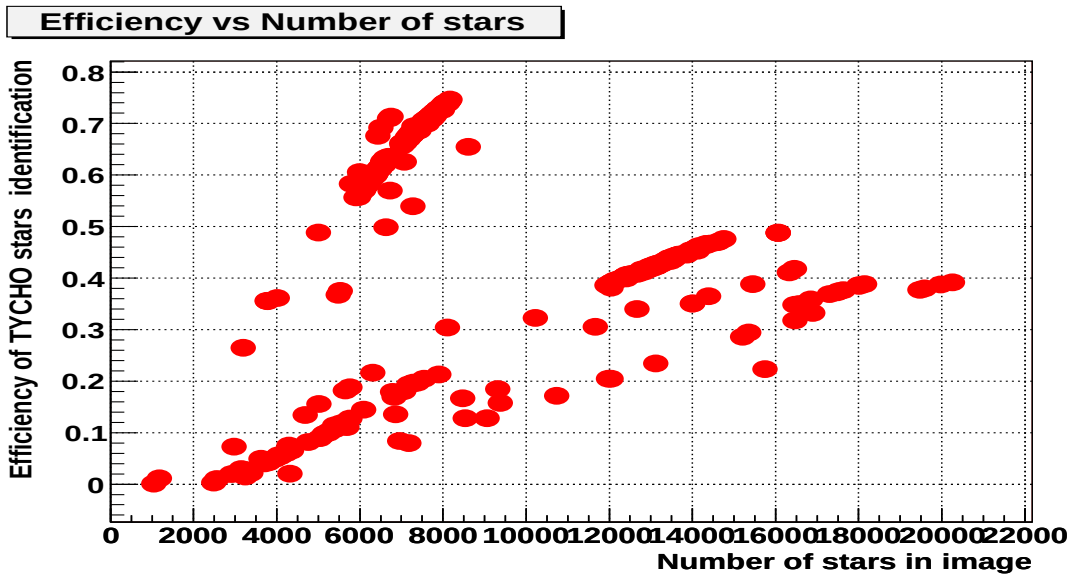

Figure 3.43: Efficiency of star identification in function of number of stars on single 10s image

The precision of star brightness measurements can be determined by finding dispersion of magnitude measurements for individual stars and ploting dispersion in function of star brightness. The Figure 3.45 shows distribution of dispersion in function of star brightness for stars observed during night 2007.06.06/07.

3.3.2 Off-line algorithms

Off-line algorithms act on data stored in the database. The data is cataloged in the way described in previous section and it is stored in tables **frame**, **star** and **measurements**. It is optimized for certain types of queries which are executed by analyzing programs. The algorithms described here have been implemented and tested on a star catalog created from images averaged over twenty - so called

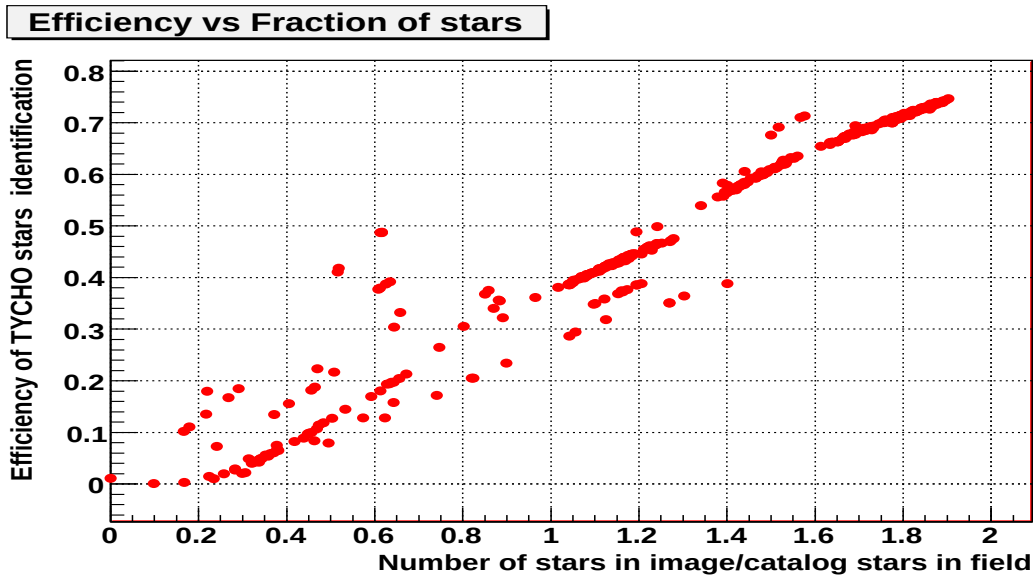

Figure 3.44: Efficiency of star identification in function of ratio R_{cat}

`aver20` database. However, they can also be used to analyse the catalog of single images data. Two algorithms developed by the author will be described. The first one looks for new objects appearing in the star catalog, which can be interesting events. Such kind of processes may be due to nova stars explosions or other kinds of processes when object below detection limit suddenly increases its brightness and appears as a new object. The second algorithm looks for sudden increases of stars brightness, such events can occur in flare stars, but this can also happen in other objects like blazars or AGNs. In both cases rejection of false event was the most important task. Both algorithms were implemented in `perl` scripting language.

3.3.2.1 Nova identification algorithm

This algorithm was developed to find new objects which were not present in the star catalog before. The algorithm is invoked for data collected during specified night. It performs the analysis of all new objects added to the catalog on analysed night. The objects which are expected to be found by this kind of algorithm are those which are normally below detection threshold of "Pi of the

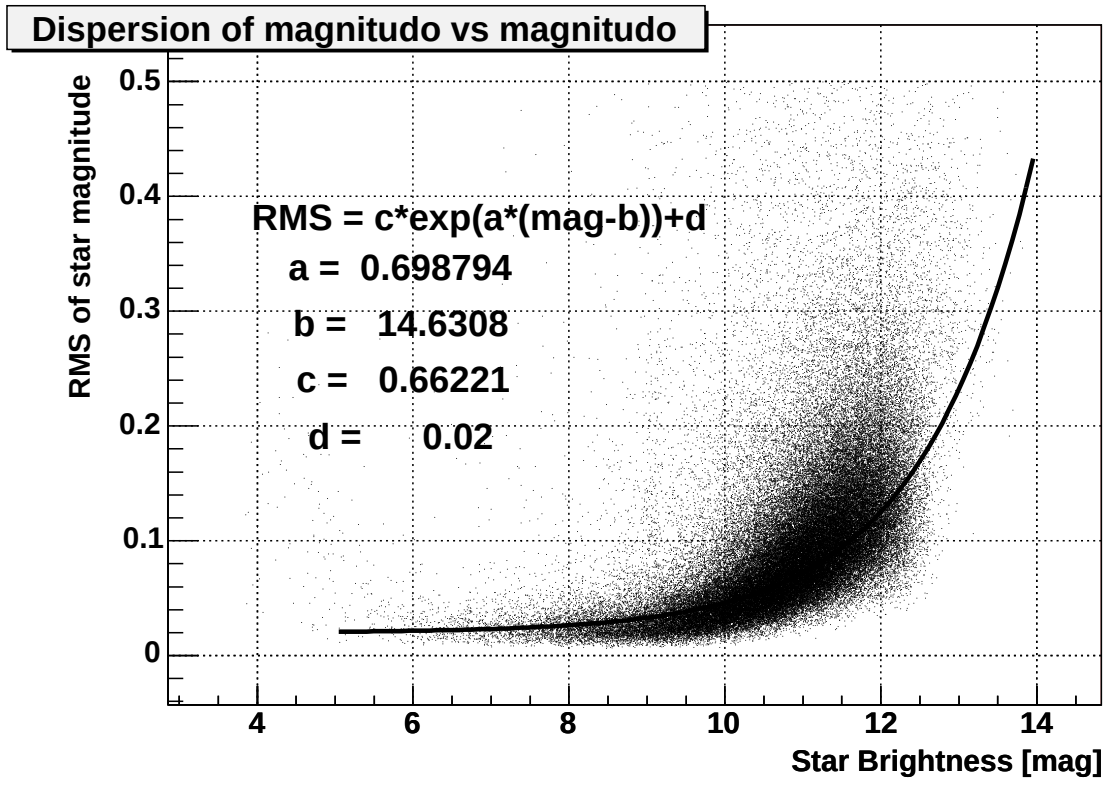

Figure 3.45: Precision of star brightness measurements in the photometry on 20 averaged images.

Sky" detector, but due to intrinsic reasons increase their brightness and become possible to be detected. The astrophysical processes can be nova star explosion, supernova or GRB, but it can also be a flare star or even variable star of big amplitude of brightness variations. There is also large amount of background processes which must be efficiently rejected. The algorithm relies on flag `Measurements.new_star` which is filled during cataloging. This field is true for the record which is the first measurement of the object. The algorithm is realized by `perl` script `do_flareevents.pl`. The main steps of the algorithm are :

1. connection to the database star catalog
2. selection of fields which were observed on the given night and have total number of observations $N^{total}_{obsfield} > N^{min}_{obsfield}$. Only images from selected fields will be considered in next steps.
3. every field selected in the previous step is analysed now. The first step is selection of all images from given night for a specific field, it is stored in list `id_frm_table`
4. images are verified against clouds and moonlight, average pixel value P_{avg} on the image is compared with the limit :

$$T_{avg} =< P_{avg} > +N_{avgcut} \cdot \sigma_{avg} \tag{3.21}$$

values $< P_{avg} >$ and σ_{avg} are obtained from distribution of number of stars on given field for all nights before the analysed night. In case the image does not satisfy condition $P_{avg} < T_{avg}$ it is rejected and skipped from further analysis. Cloudy images are also rejected in cataloging phase (see Sec. 3.3.1.3)

Every image from the analysed field kept in the array `id_frm_table` is now analysed, and the following subsequent criteria are applied to every image :

5. all measurements from an image, flagged as new object (`Measurements.new_star = true`) are selected with additional conditions for brightness of the new object mag $<= mag_{min}$ and for number of measurements $N_{obs} >= N_{obs}^{min}$. Every object accepted by the above criteria is analysed as potential nova-like event and subsequent criteria are applied.

6. verification of object in the second camera (if available)

7. determination of the number of object observations during the given night (in the case the second camera is not available it is the best indication of the event quality)

8. The event is checked if there is no nearby (R$<R_{bigstar}^{min}$) very bright star ($mag_{bigstar} < mag_{bigstar}^{max}$) in "Pi of the Sky" or TYCHO-2 [29] catalog.

9. Despite the fact that field was already observed at least $N_{obsfield}^{min}$ times, it is possible (most probably due to bad weather conditions or faintness of the object) that it is just a normal star which remained undetected before. Thus every event is checked against the TYCHO catalog and in case a star of brightness $mag < mag_{tycho}^{max}$ is found in distance R $< R_{star}^{tycho}$ the event candidate is rejected.

10. In the case the image was collected with permanently opened shutter, additional check for bright stars below and above (in CCD y coordinate) is performed. In case bright star ($mag < mag_{above}$) is found in the same image in the column close to the event candidate column ($|X_{event} - X_{bigstar}| < \delta X_{bigstar}^{max}$), then the event candidate is rejected.

11. event is verified against the list of known hot pixels, which are stored in database table HotPixels (Fig. 3.26)

All events which survived criteria 6) are saved to database table FlareEvents (Fig. 3.26). Later cuts do not reject events definitely, only rejection flag field `fl_rej_tycho` is set in the database. Its value depends on the cut which rejected the event. Table 3.10 lists possible values of rejection flag.

Rejection Flag Value	Description
0	accepted event
1	star found in TYCHO
2	nearby bright star found in TYCHO
3	saturated star found above event ($Y_{star} > Y_{event}$)
4	saturated star found below event ($Y_{star} < Y_{event}$)
5	bright star found in "Pi of the Sky" star catalog
6	hot pixel
7	sky background level $> T_{avg}$

Table 3.10: Rejection flags values

Events presentation is implemented in form of `php` script selecting events from the database. The additional difficulty is the fact that algorithm is executed on the remote server and results are originally saved to the remote database. The events data is copied to the local server (Sec. 3.3.2.3) for convenience and also in order not to overuse the Internet link to LCO by direct accessing of the remote database multiple times. The synchronization is executed once and all later analysis and inspections of events are using events data from local server not from LCO which is much faster.

Parameter	Default Value	Script option
$N^{min}_{obsfield}$	50	-min_obs_field
mag_{min}	11	-mag
$mag^{max}_{bigstar}$	8	-near_bigstar_max_mag
$R^{min}_{bigstar}$	15*36 [arcsec]	-
R^{tycho}_{star}	150 [arcsec]	-near_star_radius_arcsec
mag^{max}_{tycho}	13	-max_tycho_star
$mag^{above}_{bigstar}$	5	-max_bigstar_above
$\delta X^{max}_{bigstar}$	10	-max_bigstar_x_dist
N_{avgcut}	3	-

Table 3.11: Most important parameters of the nova identification algorithm

The cataloging program stors measurements of 10^4 - few $\cdot 10^5$ stars per night and flags new objects with special flag. The nova identification algorithm analysis stars from given night and flagged as new objects in catalog. It analysis $10^2 - 10^3$

stars, so large part of the job is done by the cataloging program. The nova identification algorithm itself takes 1-2 hours to analyse few hundred of events flagged as new objects. Figure 3.46 shows number of events after subsequent cuts for several nights.

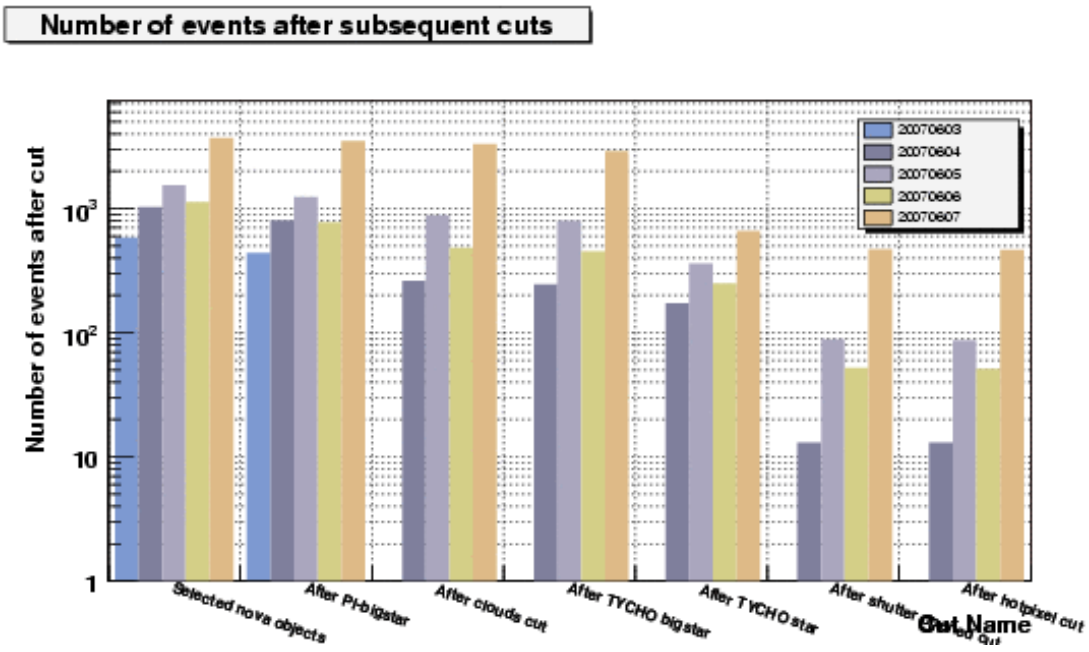

Figure 3.46: Number of events after subsequent cuts on nova identification algorithm, for five nights

Efficiency of nova identification algorithm was tested. The following procedure was used, given number of stars (typically 100 / image) of given brightness were added to star catalog database. Algorithm was executed on night for which stars with measurements were added. The number N_{gen}^{ident} of generated and accepted events was determined and using this number efficiency was determined as :

$$\epsilon_{nova} = \frac{N_{gen}^{ident}}{N_{gen}} \tag{3.22}$$

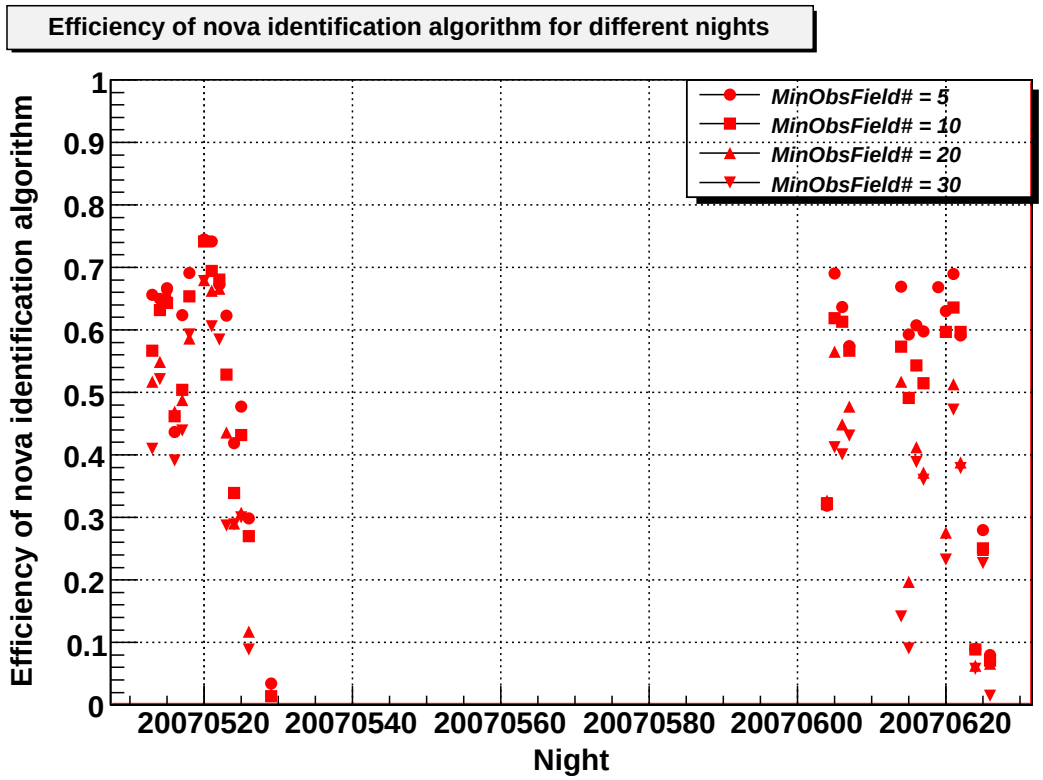

Figure 3.47: Efficiency of nova identification algorithm for several nights. Different values of minimum number of field observations requirement were tested

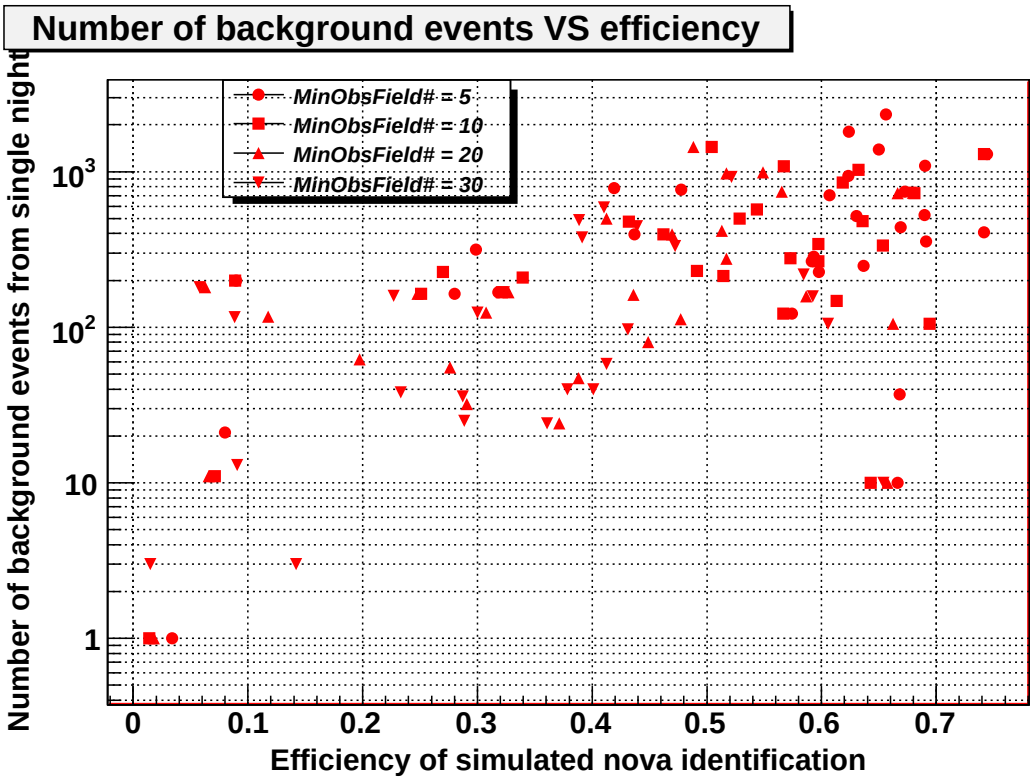

Figure 3.48: Efficiency of nova identification algorithm vs number of background events from single night

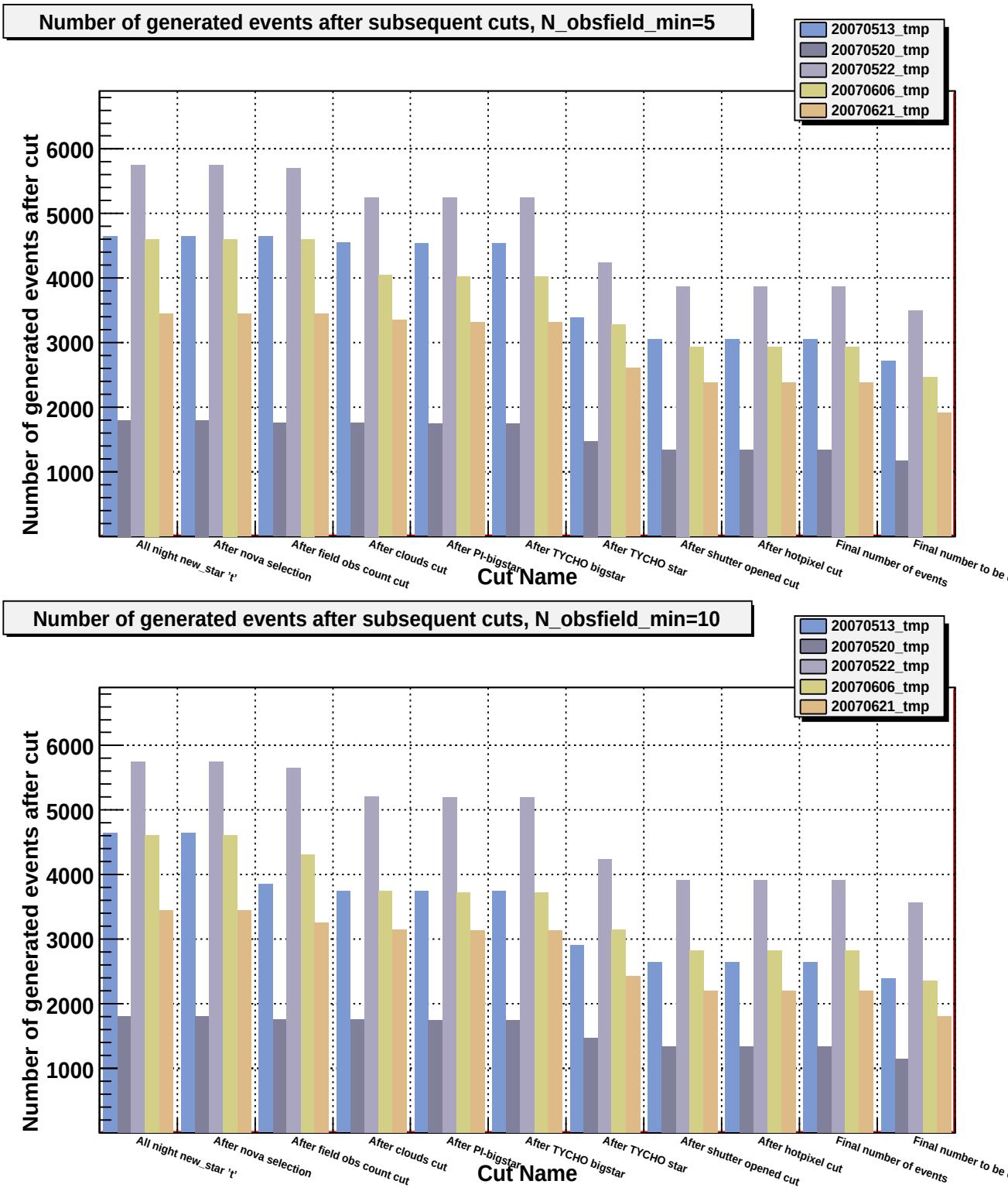

Figure 3.49: Efficiency losses on subsequent cuts of algorithm for different values of paramter $N^{min}_{obsfield}$=5,10

The Figure 3.47 shows efficiency for several subsequent nights. The best efficiencies were obtained when only 5 ealier observations of sky field were required. However, this is very small number of field observations and many weak objects still remain undetected. This implies large amount of background events, which can be seen in Figure 3.48. The Figures 3.49 and 3.50 show efficiency losses on subsequent nights for different values of $N^{min}_{obsfield}$ paramter.

The Figure 3.51 shows efficiency in function of brightness of stars added to catalog. It may be astonishing that it is independent of star magnitude, however it is clear that it should not. The procedure described above tested only efficiency of subsequent cuts of the algorithm and this does not depend on star brightness. The efficiency is different for different nights mainly due to differences in Moon light and observed fields. Due to cut on number of field observations $>N^{min}_{obsfield}$ causing rejection of events generated in images of rarly observed fields. The best way to eliminate dependency of the efficiency on number of field observations would be to initialize star catalog with the complete catalog of objects matching the range of "Pi of the Sky" telescope (12-13 mag) Such a catalog should contain all types of objects (star, galaxies etc) in the range of the telescope. The overall efficiency of nova determination does depend on the star magnitude, but this dependedncy is in the photometry procedure which was determined and described in Section 3.3.1.4. The result of multiplication of photometry efficiency on single image (result of average over 20) in function of star brightness by nova identification algorithm efficiency is shown in Figure **??**. Comparison of efficiency for several different nights is shown in Figure 3.52.

Another important result of the tests is the amount of background events which must be checked. The Figure 3.48 shows number of background events vs efficiency of the algorithm for several different values of $N^{min}_{obsfield}$ parameter.

3.3.2.2 Flare identification algorithm

This algorithm was developed to find outbursts of stars which are already in the database star catalog, but manifest sudden increase of brightness. The algorithm is executed on data collected during specified night, it is started after cataloging of new night is finished. Currently cataloging is performed on-line during data collection so analyzing script `find_flares_all_nights.pl` can be launched in

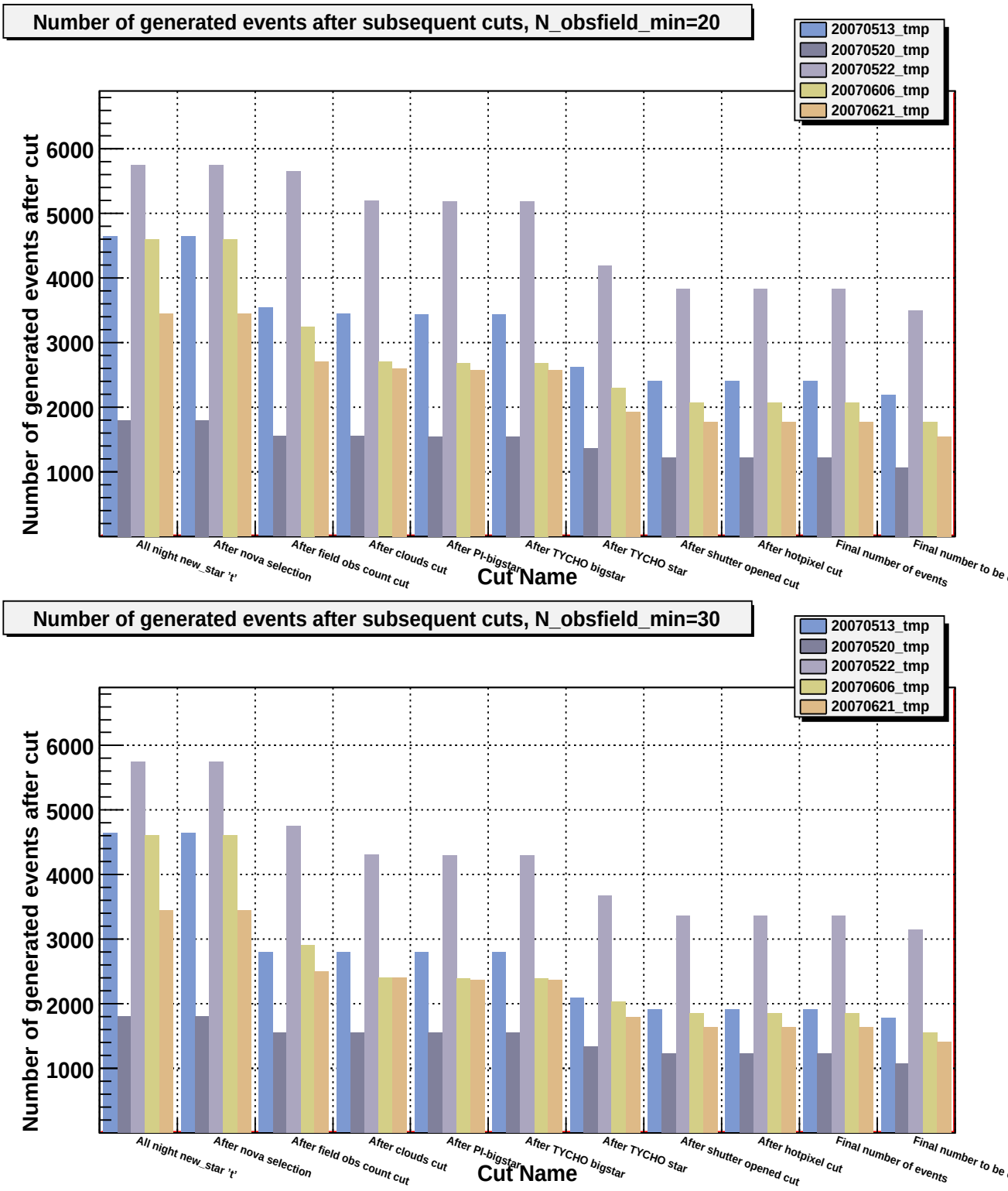

Figure 3.50: Efficiency losses on subsequent cuts of algorithm for different values of paramter $N^{min}_{obsfield}$=20,30

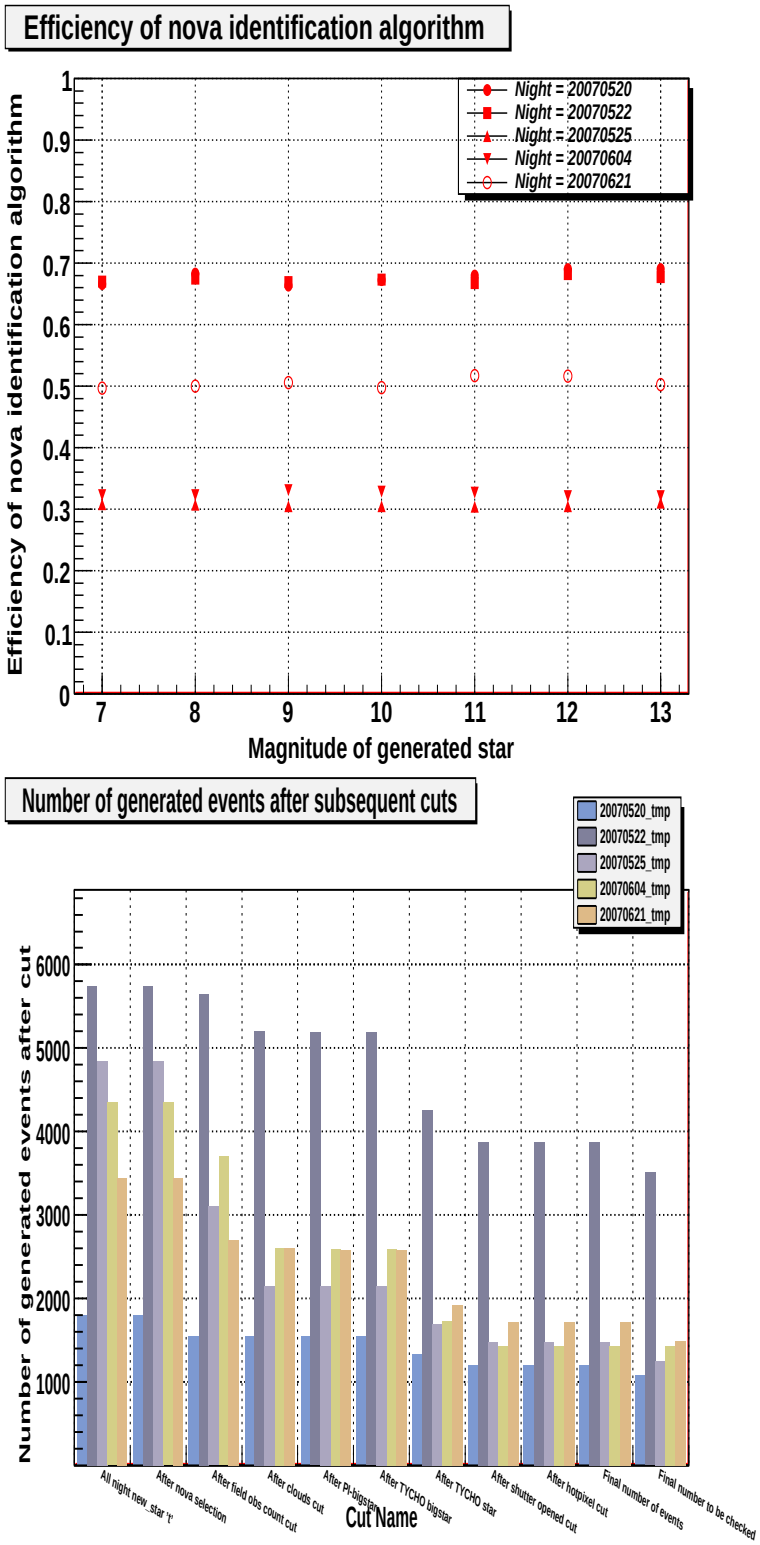

Figure 3.51: Efficiency of nova identification algortihm for different brightness of object added to star catalog, tested for different nights. The lower plot shows the number of generated events rejected by subsequent cuts

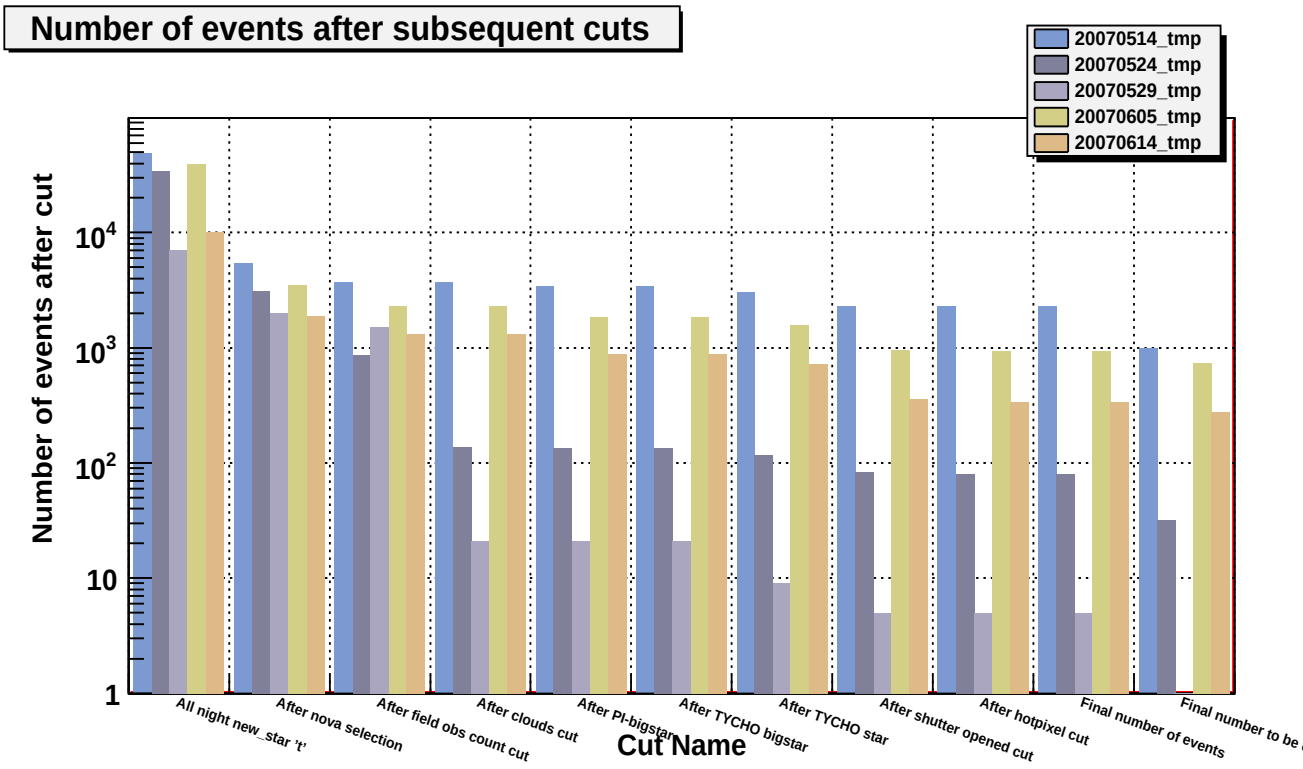

Figure 3.52: Comparison of efficiency on 5 nights, night 20070529 was collected at almost full moon and cut requiring low level of sky background rejected most of the generated events

the morning. The main steps of the algorithm are the following :

1. Selection of stars brighter then mag_{min}, observed during analysed night and having total number of measurements $N_{obs} >= N_{obs}^{min}$
2. Variability cut - star brightness range must satisfy condition : $mag_{MAX} - mag_{MIN} >= T_{magdiff}$. The number of stars selected at this stage is of order of $10^4 - 10^5$, it depends on quality of the night data and on $T_{magdiff}$ parameter. This dependency is shown in Fig. 3.53.

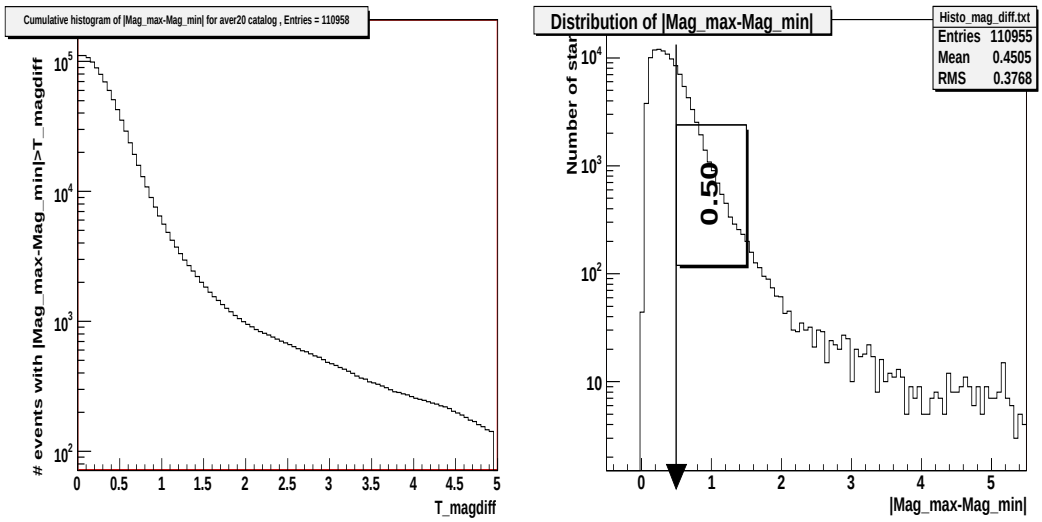

Figure 3.53: Number of stars with $mag_{MAX} - mag_{MIN} >= T_{magdiff}$ (left plot) and distribution of $mag_{MAX} - mag_{MIN}$ (right plot) only stars satisfying condition for number of measurements are shown (data from night 2007.05.26/27).

For every selected star the following steps and criteria are performed :

3. Select measurements for given star
4. Check if number of measurements for the given night $N_{obs}^{night} >= N_{minobs}^{night}$
5. Find upper limit of the magnitude range mag_{max}, which is defined as maximum value of magnitudo in the "last" non empty bin [1] of magnitudo measurements distribution (see Fig. 3.54). In most cases the value mag_{max}

[1] in the direction of increasing magniutde

is the same as maximum magnitude measurement for given star. However, in some cases (e.g. due to clouds) real "last" non empty bin can contain few bad measurements and can be separated from most of the measurements (see Fig. 3.55). In order to exclude those bad measurements, "last" non empty bin [M,M+δM] is choosen as the one before an empty bin and satisfying condition :

$$N(mag <= M + \delta M) >= 50\% \cdot N_{obs} \tag{3.23}$$

The maximum magnitude is found in the "last" non empty bin and used as upper limit mag_{max}.

6. Find lower limit of magnitude mag_{min} so that at 85% of all magnitude measurements for given star belong to range (mag_{min},mag_{max}) (see Fig. 3.54). Value of flare threshold is determined as $T_{flare} = mag_{min}$, all points brighter then this value are considered as belonging to a flare.

7. Find longest series of measurements with mag $< T_{flare}$ and require this set to have at least N_{min}^{flare} points

8. Check maximum brightness measurement M_{max}^{flare} in the flare series and determine its time T_{max}^{flare}

9. Verify if measurements within the series and before and after have the same chip coordinates (x,y), because due to calibration problems star brightness measurements can vary with the star position on the chip, causing background flare like signatures.

10. If the data was collected with permanently opened shutter verify if there is no star brighter then mag_{above} in the strip of $\delta X_{bigstar}^{max}$ pixels from the analysed star. Opened shutter causes that column obtains additional signal from stars with $Y <= Y_0$ and can cause background events. The rejection condition is the same as in described earlier nova identification algorithm.

11. Check if there is no bright star nearby which could affect measurements of flare suspected star. In case a star brighter then $mag_{bigstar}^{near}$ is closer then $R_{bigstar}$ then event is rejected. This condition is checked in Pi star catalog and also in TYCHO-2 [29] star catalog

12. Verify if brightness increase is not due to hotpixel. Event is required not to belong to list of known hot pixels stored in database table HotPixel (Fig. 3.26)

13. Peak height over the normal brightness level is determined and flare event obtains quality flag according to this value, events with $M_{max}^{flare} > 0.4^m$ obtain quality=1 and those with $M_{max}^{flare} > 1.0^m$ obtain quality=2

14. Finally event is accepted and saved to the database table **FlareEvent** with all information describing this event and calculated by the algorithm

15. Sky background level on the image is checked in the same way as in nova identification algorithm (Sec. 3.3.2.1).

Algorithm parameters are listed in Table 3.12.

Parameter	Default Value	Script option
N_{obs}^{min}	30	-min_points
mag_{min}	12 [mag]	-min_mag_star
$T_{magdiff}$	0.5 [mag]	-min_max_mag_limit
N_{minobs}^{night}	20	-min_night_points
N_{min}^{flare}	3	-min_points_above
$mag_{bigstar}^{strip}$	5 [mag]	-max_bigstar_above
W_{strip}	10 [pixels]	-max_bigstar_x_dist
$mag_{bigstar}^{near}$	8 [mag]	-near_bigstar_max_mag
$R_{bigstar}$	15 [pixels]	-near_bigstar_distance

Table 3.12: Most important parameters of the flare identification algorithm

Depending on the quality of the night, algorithm analysis $N_{star} < 10^5$ stars which takes about 1-2 hours. Event numbers after subsequent cuts are shown in Figure **??**.

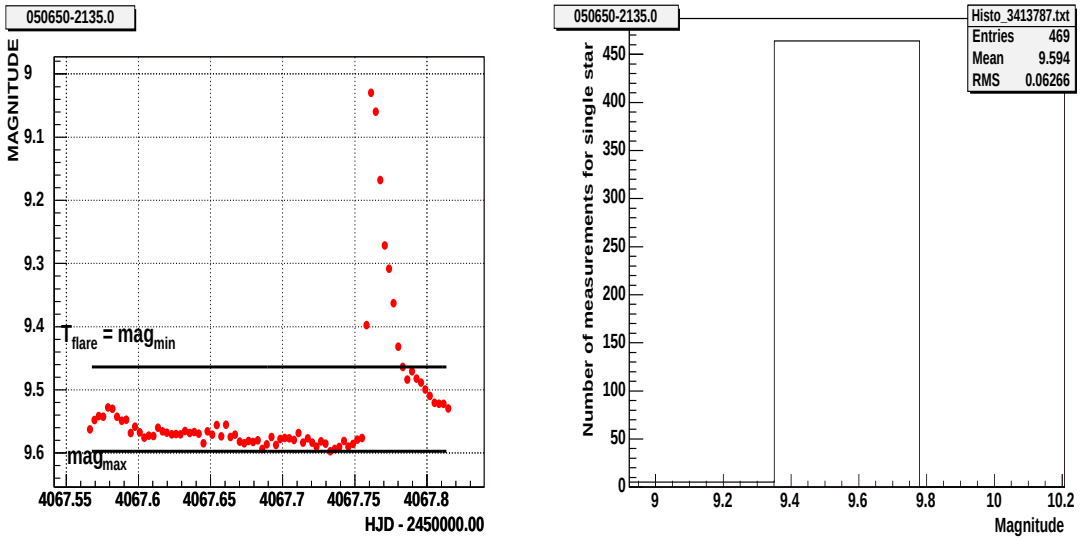

Figure 3.54: Example of determined range of 85% of measurements points - area between horizontal lines in the left plot and distribution of all measurements for the same star (right plot)

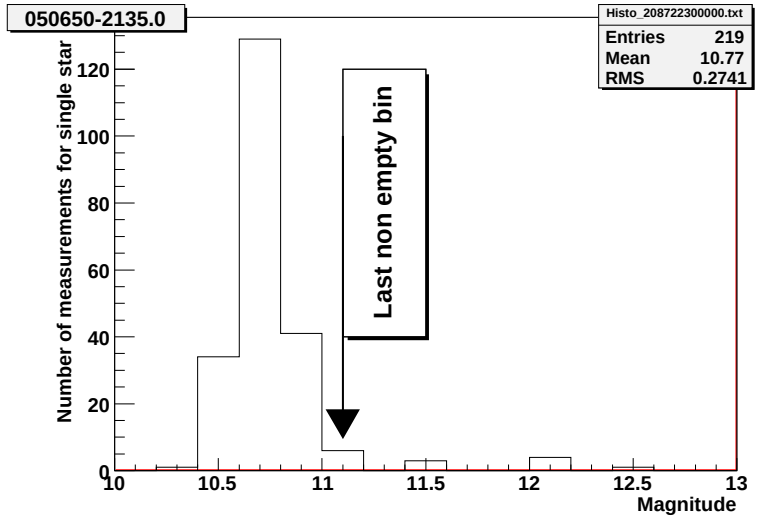

Figure 3.55: The distribution of magnitude measurements for single star an argument why not maximum magnitude is used as upper limit mag_{max} in the algorithm

In order to determine efficiency of this algorithm flare events were generated. Events were generated for specified night and analysied by the same script for flare identification used for analysis of the real data. The simulation consists of the following main steps :

- Certain number N^{gen}_{flare} of stars in the star catalog was selected (specified by simulation paramter)
- Time of flare start was randomly choosen from the series of measurements for given night
- Magnitude measurements after the starting time were replaced by values obtained from flare paramterization
- Star with generated points added was analysed and accepted or rejected by the algorithm

In order to simulate flare like outburst simple linear/exponential paramterization was fitted to real flare detected by the algorithm (see Section 4.3) :

$$mag = \begin{cases} m0 - F_{max} * \frac{t}{\delta t} & \text{if } t \leq \delta t \\ m0 - F_{max} * exp\left[-\frac{t-dt}{\tau}\right] & \text{if } t \geq \delta t \end{cases} \tag{3.24}$$

These parameterization is presented in Figure 3.56.

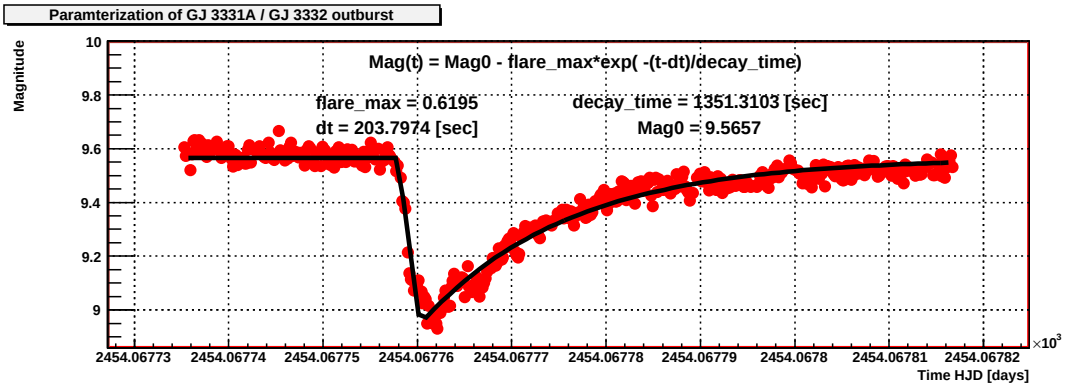

Figure 3.56: Linear/Exponential paramterization of the flare-like outburst. Fit was performed to real flare star GJ 3331A / GJ 3332 outburst which occured on 2006.11.28 06:03 UT

Flare identification efficiency was tested in function of 4 paramters used to describe the flare in fitted parametrization. Examples of generated flare lightcurves are shown in Figure 3.57.

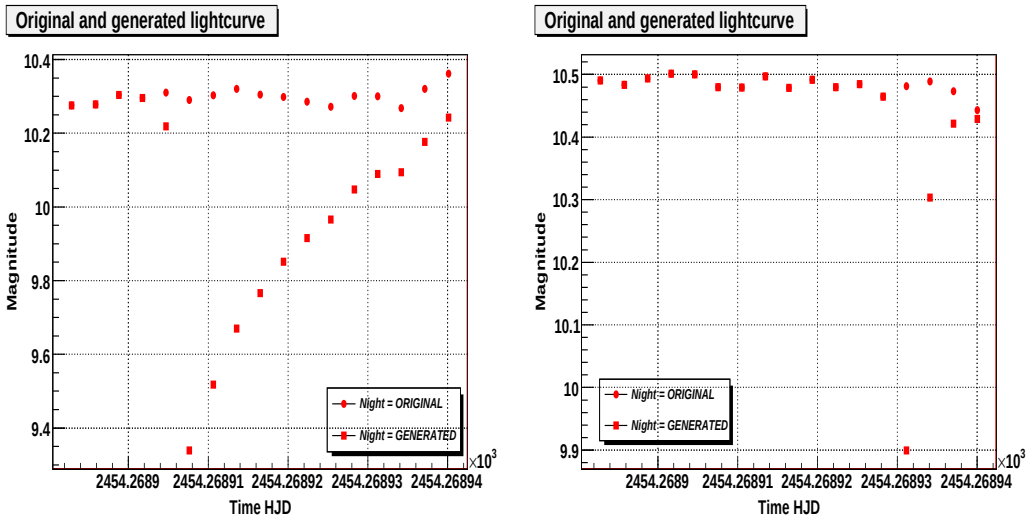

Figure 3.57: Example of lightcurves with generated flare with outburst peak of 1^m above average brightness level (left plot) and decay time of 200 sec (right plot), other paramters of the lightcurver are the same as those fitted to GJ3331A/GJ3332 outburst

The results of these tests are shown in Figure 3.58.

3.3.2.3 Data synchronization and presentation

The cataloging and data analysis are performed on the remote server. In the prototype version it is performed on pi2 and pi3 computers at LCO (before also pi1). In order not to overuse the bandwidth only the final results are copied to local server in Warsaw. It is impossible to synchronize the entire database because it is too large. However, results of algorithms can be synchronized with light curves of corresponding stars. There is also possibility to specify interesting objects which light curves should also be synchronized. After synchronization data is stored in the local database which is accessible from the command line and script level, but also through the WWW php interface. Results of on-line algorithms are synchronized on-line during a night. Besides database information also sub renders of sky images containing events are copied to local server allowing

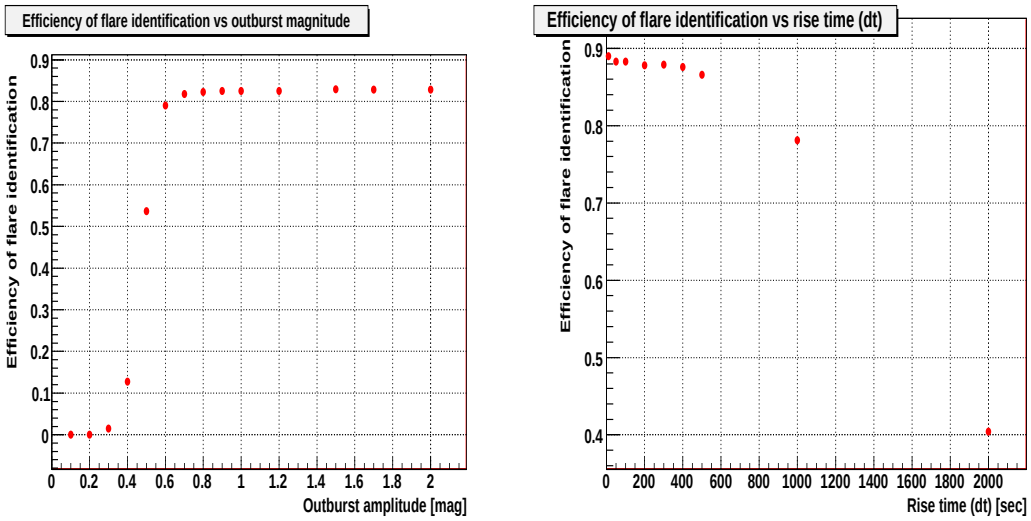

Figure 3.58: Efficiency of flare identification in function of outburst magnitude (left upper plot), rise time (right upper), decay time (left bottom), star average magnitude (right bottom)

for quick verification of algorithm results. They are also available through the WWW interface.

3.4 Periods when algorithms were working

Before showing the results of the algorithms it is important to mention that not all algorithms were used during the full time of the prototype work. Table 4.1 shows periods in which the prototype was working, but different configurations are also shown. Because of technical problems shown in this table coincidence algorithm could not work for all the time. Also not all the algorithms were ready to be used since the very beginning. Table 3.13 shows which algorithms were used in what periods. Some algorithms could also be run off-line, but large amount of events could not be verified. Currently all off-line algorithms are working and events are systematically checked after each night.

Algorithm Type	Working Periods	Notes
On-line coincidence	2004.06.25 - 2005.03.01 , 2006.05.20 - 2006.08.08	two cameras
On-line confirmation on next image	2005.03.12 - 2005.08.09 , 2006.05.20 -	single camera
Off-line nova on aver20	2006.07.21 -	
Off-line flare on aver20	2006.07.21 -	

Table 3.13: Periods in which different algorithms were used

Chapter 4

Results

4.1 Data from the prototype

The data from prototype system was collected in time period from June 2004 until now (31 May 2007) with 9 months break due to maintenance reasons. During this period the system collected data from 598 nights which is 81% of nights in period when system was operational. The remaining 19% was lost mainly due to clouds or some minor system failures which could be remotely solved after few hours or days. Table 4.1 lists data collection periods with description of major problems in the system.

The observation time was calculated using the image information saved in the database. The Table 4.2 shows observation times for two versions of the prototype. This times were normalized to 4π coverage. In the first period all sky images from nights with at least 100 images were taken into account, in the second period additional cut on number of stars on image $N_{stars} >= 1000$ was imposed in order to ignore cloudy data. It was not possible for the first period because this information was added to database during upgrade.

4.2 Optical flashes in 10s timescale

Every night when the system was running, the flash identification algorithm was looking for optical flashes on-line. The final results of the algorithm were verified by a person on duty and events which could not be rejected by any criteria were flagged as "flashes". The statistics of all optical flashes detected by the prototype

Start Date	End Date	Period Description
20040625	20050120	both cameras k2a and k2b working correctly
20050120	-	problems with k2b power supply began
20050223	-	cooling of k2b broken, due to Peltier junctions power supply failure. Also power supply for digital electronics was broken causing images to be black or having only few most bright stars. Since this moment k2b was working poorly, only with temporal improvements (20050421-20050509)
20050419	-	k2b shutter mechanism broken, data collected with opened shutter
20050702	20050809	k2a cooling failed
20050810	20060520	system down due to maintenance reasons
20060520	20060808	system working after cameras repair and upgrade, both cameras working
20060809	-	k2b camera broken due to electricity problems in LCO, since this moment system is working only with k2a camera

Table 4.1: Data collection periods of the prototype, with crucial moments of major failures listed

Time Period	Normalized observation time [days]	Objectives
20040623 - 20050809	2.04	Carl Zeiss f=50mm
20060520 - 20070308	0.51	Canon f=85mm

Table 4.2: Data collection periods of the prototype, with crucial moments of major failures listed

is shown in Table 4.3. Most of the flashes were visible on single 10s exposure in both cameras. All these events are listed in Tables 4.6, 4.5 and 4.4 they contain also the D_{cone} value (Sec. 3.2.6) , however none of the events had this value sufficiently big to confirm astrophysical origin of the flash. There are few events which were observed on two consecutive images, but only by a single camera (when the second was not operational). They are listed in Table 4.7. In this case also limit of distance derived from constant position of the flash is shown (Sec. 3.2.6).

Period	Algorithm	Number of flashes
20060520 - till now	Confirmation on next image	1
20060520 - 20060822	Coincidence	35
20040625 - 20050809	Coincidence	97
20040625 - 20050809	Confirmation on next image	6

Table 4.3: Statistics of optical flashes detected by the prototype

In one single case the flash was unambiguously identified as an astrophysical event. This is the event on 2005.04.02 1:13:40 UT , presented in Figure 4.2. It was identified as an outburst of the star CN Leo (RA = $10^h56^m29^s$, Dec = +7°01'), which was already known as a flare star. The light curve of this flash is shown in Figure 4.3. The star has normally brightness of m=13.5^m and increased it ≈ 100 times reaching $m_{max} = 9^m$. It was below the limiting magnitude of the telescope before the explosion and suddenly appeared as a new object. The signature of the outburst was clearly the flash-like signature which allowed on-line algorithm to identify it. The example of outburst visible on more then one image is shown in Figure 4.1, this outburst has not been assigned to any known source.

All other flash events has been analyzed, for some of them space coincidence with known astrophysical objects was found as shown in the tables above. No evident coincidence with known GRB event was found.

4.3 Outbursts in 240s timescale

The off-line algorithms acting on the 20 averaged images were not implemented from the very beginning of the prototype running. Part of the archival data was

Date	UT Time	RA	Dec	D_{cone}	M_{max}	coinciding alerts / sources
2006.07.25	07:21:27	18h39m03.14s	-10°37'43"	26877.59	10.49	
2006.07.20	07:24:02	18h41m08.83s	-10°43'39"	31842.61	9.78	
2006.07.17	05:17:15	18h22m42.45s	-12°00'06"	25099.11	-	
2006.07.17	04:48:47	23h56m13.09s	-23°52'06"	11975.36	10.1	
2006.07.17	03:46:37	19h18m36.07s	-10°50'05"	34274.14	8.8	
2006.07.17	00:55:35	12h12m16.84s	-65°13'37"	7597.41	10.7	
2006.07.15	05:57:24	19h42m01.54s	-09°26'54"	37745.07	10.5	
2006.07.15	03:01:12	18h24m02.21s	-11°56'53"	18369.61	10.1	
2006.07.14	07:29:28	18h46m13.81s	-10°48'56"	39593.87	10.2	
2006.07.10	04:51:00	20h05m44.80s	-10°08'05"	29063.88	9.04	
2006.07.10	00:18:02	12h42m52.55s	22°36'27"	6980.69	10.81	<1 arcmin from NGP9 F378-0444065 - Galaxy, 1 arcmin from FIRST J124239.4+223536 - Radio-source
2006.07.03	08:42:04	01h02m50.01s	04°29'27"	7380.32	9.76	
2006.07.02	09:48:58	00h40m30.20s	04°00'26"	6831.95	9.6	
2006.07.02	08:15:12	01h42m27.20s	09°34'42"	7413.05	10.45	
2006.06.29	22:58:34	11h40m20.20s	13°34'58"	6530.95	11.37	
2006.06.29	22:49:16	11h38m18.76s	11°50'08"	6492.79	8.05	
2006.06.27	03:19:54	16h10m39.49s	-11°14'39"	14128.27	11.17	
2006.06.27	02:24:34	18h04m16.33s	-08°04'15"	24909.05	-	
2006.06.27	02:19:55	17h59m55.77s	-07°56'07"	23450.17	-	
2006.06.26	03:51:28	16h39m03.75s	-11°40'55"	17908.50	11.11	
2006.06.24	02:47:18	16h15m34.48s	-37°06'22"	13787.51	10.81	
2006.06.23	09:13:04	23h41m41.95s	00°01'38"	7174.30	-	7 arcsec from J234135.2-000145 Seyfert 1 Galaxy
2006.06.18	04:19:38	19h20m32.60s	-31°07'50"	22918.17	10.1	
2006.06.14	09:48:41	22h17m47.85s	-06°26'54"	6873.38	9.7	
2006.06.13	07:38:58	22h24m32.22s	-07°17'11"	8471.05	9.26	
2006.06.12	07:53:49	22h01m06.08s	-10°47'12"	8405.26	10.5	
2006.06.11	00:03:36	09h59m22.93s	00°29'16"	6937.20	8.7	1 arcmin from [LPZ94] 279 – Radio-source
2006.06.10	04:57:53	18h32m04.61s	-10°04'08"	21981.23	8.93	
2006.06.08	07:44:44	17h31m54.42s	-11°02'14"	22001.87	8.5	2 arcmin from IRAS 17291-1057 - Infra-Red source
2006.06.08	06:08:36	17h44m25.43s	-17°24'02"	30564.76	-	
2006.06.06	07:20:08	17h41m41.34s	-62°07'57"	11407.20	-	
2006.06.05	00:57:09	12h15m14.43s	-20°19'19"	7959.37	-	
2006.06.01	03:51:14	12h49m26.68s	13°17'54"	10775.38	-	1 arcmin from FIRST J124926.0+131612 - Radio-source
2006.05.22	05:21:25	16h39m21.25s	-09°35'11"	29140.23	-	
2006.05.21	05:06:28	15h38m06.76s	12°49'19"	18604.53	-	

Table 4.4: Optical flashes identified by the on-line algorithm since June 2006 requiring coincidence of flash on 2 cameras. These flashes were obse ved on single 10s exposure

Date	UT Time	RA	Dec	D_{cone}	M_{max}	coinciding alerts / sources
2005.08.07	23:59:16	21h44m40s	-41°49'	8444.05	-	
2005.08.07	06:52:52	21h59m20s	-11°52'	21841.07	-	
2005.08.07	00:21:42	19h27m22s	-23°52'	7833.39	-	
2005.08.01	01:56:25	17h40m45s	+14°22'	9352.06	8.05	
2005.06.30	09:39:38	19h02m58s	-18°40'	11783.78	9.8	
2005.06.02	03:35:18	11h42m51s	+01°57'	9668.98	10.3	
2005.06.01	23:34:14	12h41m14s	+10°44'	6888.54	9.04	1arcmin from GRDG +11 54 Galaxy
2005.06.01	22:54:14	12h02m24s	+12°49'	6553.28	10.03	
2005.06.01	03:47:30	11h50m11s	+01°28'	9983.38	11.2	
2005.06.01	00:19:37	12h56m19s	+14°21'	7485.79	9.0	
2005.05.31	22:56:15	12h23m32s	+11°49'	6571.52	10.0	J122331.1+114752 radio
2005.05.31	04:12:52	11h58m51s	-02°02'	10570.86	11.0	Konus 4:27:26
2005.05.31	00:01:46	11h54m52s	+15°07'	7110.83	9.3	
2005.05.27	23:02:13	10h15m13s	+14°20'	6570.98	10.9	
2005.05.27	22:55:12	10h52m40s	+28°37'	6537.24	10.0	
2005.05.27	22:53:34	10h51m3s	+24°42'	6530.79	7.2	
2005.05.27	05:36:11	16h56m42s	-10°27'	29060.33	10.0	
2005.05.25	03:57:10	16h01m30s	-08°01'	36600.71	10.0	
2005.05.20	08:33:30	15h33m51s	-32°21'	18299.27	12.0(?)	
2005.05.15	23:04:50	13h39m42s	-39°34'	6659.66	10.1	
2005.05.14	03:28:22	13h28m41s	-15°57'	15960.10	9.3	
2005.05.14	03:19:27	13h18m45s	-21°30'	14734.27	10.3	
2005.05.14	02:35:02	13h33m57s	-27°26'	13338.49	10.5	
2005.05.12	04:59:08	12h47m39s	+10°24'	14673.83	8.9	
2005.05.11	03:05:50	11h13m34s	+12°15'	10060.39	8.5	
2005.05.06	05:53:02	14h02m13s	+00°35'	28414.20	8.5(?)	
2005.05.01	10:09:25	18h49m16s	+09°17'	6571.82	-	
2005.04.24	06:24:24	16h48m39s	-10°18'	11778.71	9.1	
2005.04.01	06:12:52	11h16m47s	+06°59'	26486.27	10.2	GSC 00269-00778
2005.03.30	04:45:08	11h00m03s	+14°38'	19964.19	8.9	
2005.02.16	00:32:49	10h00m19s	-13°16'	6745.62	8.3	
2005.02.15	08:01:54	10h08m17s	+13°10'	23523.08	8.4	
2005.02.15	04:54:47	11h00m10s	+06°16'	17082.21	9.0	
2005.02.12	03:24:46	09h45m31s	-14°43'	9815.57	9.3	
2005.02.04	07:07:33	09h51m46s	+12°19'	12323.42	9.6	
2005.01.31	00:35:47	07h00m11s	+11°37'	6567.26	9.2	
2005.01.28	07:09:05	06h48m39s	+13°23'	17545.88	9.6	
2005.01.17	04:01:01	08h49m00s	+17°12'	21936.75	9.0	
2005.01.10	00:54:15	05h10m31s	+13°40'	6624.79	9.7	
2005.01.09	02:23:36	06h58m53s	+17°11'	10701.16	9.25	
2005.01.07	02:52:50	06h02m36s	+09°06'	8639.52	11.0	
2005.01.03	06:34:48	07h59m55s	18°00'	10149.77	6.3	
2005.01.02	06:41:28	05h59m41s	+20°53'	30460.54	8.9	Konus+Integral 6:27:27

Table 4.5: Optical flashes identified by the on-line algorithm in 2005 requiring coincidence of flash on 2 cameras. These flashes were observed on single 10s exposure

Date	UT Time	RA	Dec	D_{cone}	M_{max}	coinciding alerts / sources
2004.12.27	07:22:21	05h48m30s	00°00'	8073.92	9.2	
2004.12.27	03:51:03	04h49m40s	+19°58'	11531.88	9.8	
2004.12.22	02:22:12	07h07m03s	+17°41'	20513.74	9.3	
2004.12.20	02:24:46	05h56m24s	+03°10'	8343.60	9.3	
2004.12.19	02:46:43	05h03m58s	+07°34'	8624.41	9.9	
2004.12.18	05:53:37	05h48m38s	+12°26'	11325.46	12.00	
2004.12.14	02:29:21	05h32m09s	+14°29'	11375.21	9.1	
2004.12.11	00:47:21	03h13m14s	+06°09'	6640.76	9.8	
2004.12.04	07:34:46	05h19m21s	+14°34'	7944.71	10.3	
2004.12.04	06:28:54	05h05m28s	+21°15'	18083.59	11.2	
2004.12.04	02:05:22	03h01m51s	+05°57'	7689.37	9.7	galaxy [ZHG90] 0259+0545
2004.12.02	05:13:41	04h07m24s	+20°27'	35733.74	9.6	
2004.12.01	01:28:47	01h34m19s	+14°51'	7123.42	9.8	
2004.11.30	04:09:42	03h21m04s	+19°01'	15993.36	11.5(?)	
2004.11.28	04:27:16	01h27m28s	+10°48'	10041.09	9.6	
2004.11.28	03:20:15	03h04m44s	+05°42'	9501.62	7.5	
2004.11.26	05:58:15	05h10m12s	+14°40'	11027.50	9.7	
2004.11.24	07:39:33	06h54m30s	+18°49'	7145.41	9.7	
2004.11.23	02:39:10	05h14m29s	-07°06'	9110.72	10.0(?)	
2004.11.19	00:53:08	03h14m50s	+03°20'	7165.51	9.2	
2004.11.18	07:53:55	04h22m13s	+12°31'	7505.47	10.5	
2004.11.18	01:36:09	03h10m22s	+03°12'	8120.09	9.0	
2004.11.18	00:25:36	01h30m17s	+11°44'	6655.27	8.9(?)	
2004.11.17	03:31:17	02h48m36s	+11°13'	13252.46	-	
2004.11.12	02:51:31	02h27m12s	+22°06'	23386.60	-	
2004.11.11	04:46:43	02h08m07s	+10°13'	16192.00	-	
2004.11.11	01:57:49	04h03m47s	+18°33'	39316.77	-	
2004.11.06	06:02:58	03h47m24s	+10°22'	12004.63	-	
2004.11.05	07:49:20	02h34m32s	+05°47'	9813.32	-	
2004.11.03	01:22:40	00h12m32s	+00°49'	7482.92	-	Konus 1:26:43, Integral 1:27:11
2004.11.02	07:14:15	01h23m53s	+01°13'	13855.20	-	Konus 7:27:56
2004.11.01	05:49:34	23h47m44s	-04°23'	10893.98	-	
2004.11.01	05:22:28	23h20m39s	-05°43'	10146.59	-	
2004.10.31	06:15:37	23h24m40s	-01°12'	11214.86	-	
2004.10.30	05:22:06	00h42m54s	-29°30'	9093.59	-	
2004.10.28	02:56:42	03h34m40s	-08°15'	11525.31	-	
2004.10.24	03:40:00	00h36m31s	-05°10'	11388.33	-	Radio-source Cul 0034-054
2004.10.21	02:36:00	01h02m16s	-04°17'	11101.45	-	
2004.10.11	07:37:09	03h12m08s	+10°39'	8485.28	-	
2004.10.11	06:15:13	02h54m57s	+18°16'	13331.99	-	
2004.10.05	04:18:40	01h24m04s	+14°17'	41060.68	-	
2004.10.02	02:20:50	23h05m26s	+12°36'	13430.52	-	
2004.10.01	04:09:12	21h46m26s	-08°18'	11642.37	-	
2004.09.30	00:49:38	23h19m50s	-02°36'	8307.80	-	
2004.09.27	04:20:55	03h29m33s	+14°08'	12965.21	-	
2004.09.17	03:50:25	23h42m55s	+31°44'	20126.16	-	
2004.09.13	03:33:52	00h14m45s	+02°50'	40744.17	-	

Table 4.6: Optical flashes identified by the on-line algorithm in 2004 requiring coincidence of flash on 2 cameras. These flashes were observed on single 10s exposure

Date	UT Time	RA	Dec	D_{cone}[km]	M_{max}	External Alerts or Sources
2006.10.10	02:44:43	00h8m57.80s	34°51'	18582.01	11.7	
2005.06.03	06:29:57	19h37m14s	-50°47'	11616.40	10.5	Konus 6:29:01, excl. by Swift coords
2005.05.31	04:33:17	17h43m48s	-19°55'	28323.30	6.1	Konus 4:27:26, excl. by IPN
2005.04.16	00:49:59	11h03m11s	11°06'	8118.52	10.5	SN2003L NGC3506
2005.04.04	05:37:59	11h49m37s	-05°31'	30852.52	10.3	GSC 04937-00794or LCRS B114710.1-051705or 1RXS J114951.0-053041
2005.04.02	01:13:42	10h56m29s	+07°02'	9081.94	9.0	V* CN Leo - flare star
2005.03.31	01:36:46	11h52m53s	-05°59'	12796.88	10.0	several galaxies or GSC 04938-00378
2004.02.21	00:27:00	11h18m44s	-06°06'	10298.58	9.0	LEDA 114805

Table 4.7: Optical flashes identified by the on-line algorithm requiring signal at least on 2 consecutive images

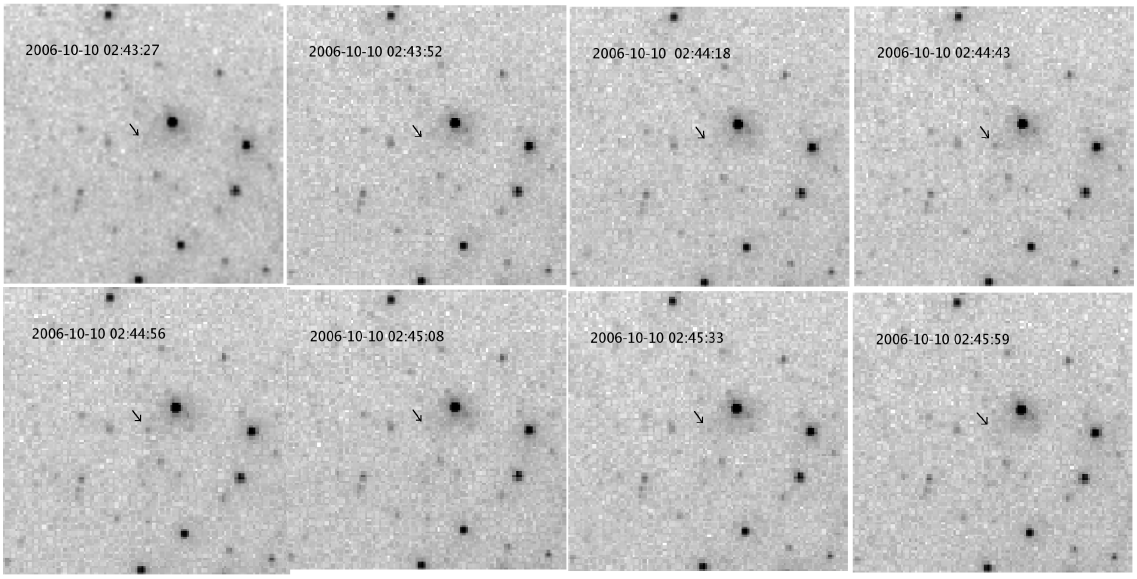

Figure 4.1: Optical flash visible on 2 consecutive images on 2006-10-10 02:44:43

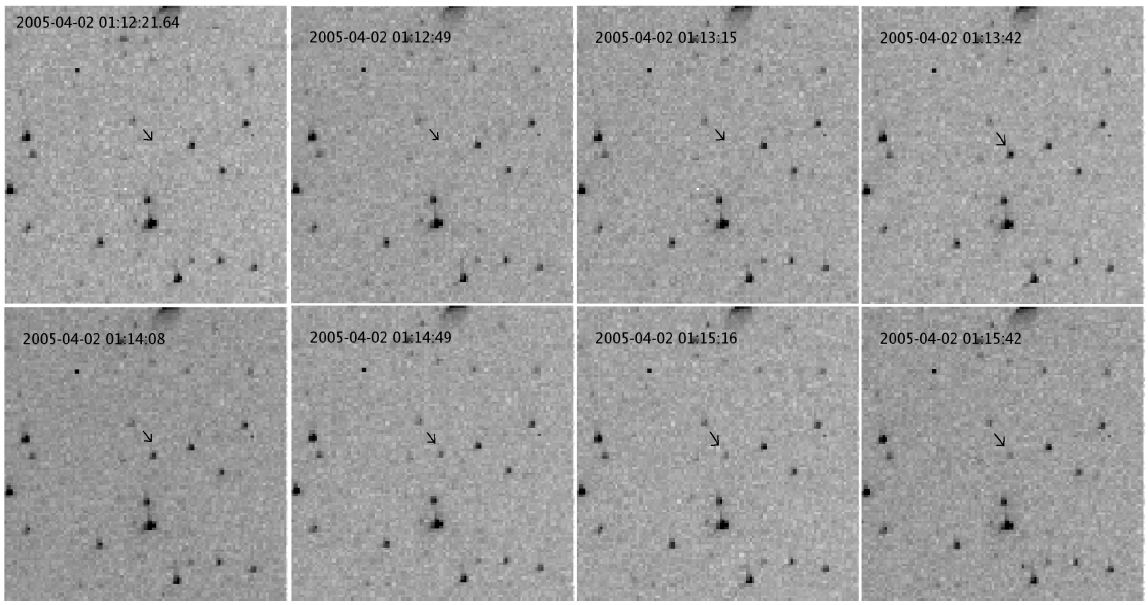

Figure 4.2: Images of CN Leo outburst

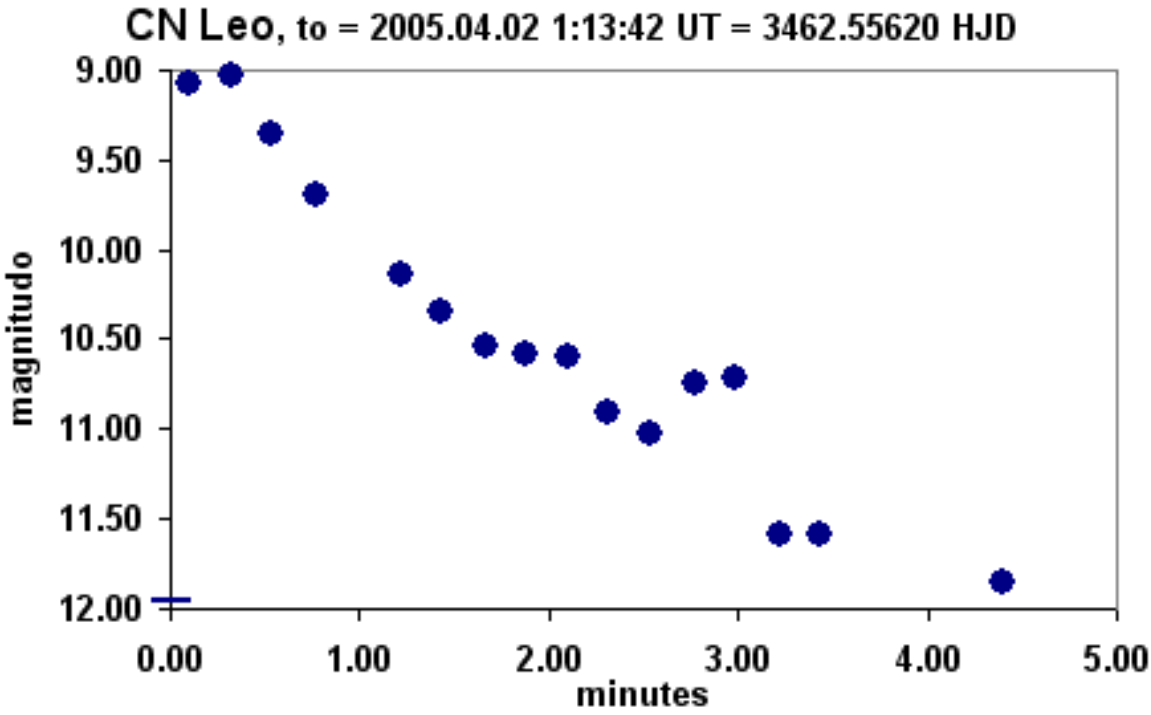

Figure 4.3: Light curve of outburst of flare star CN Leo identified by flash recognition algorithm on 2005.04.02 1:13:40 UT

analyzed and checked, but the amount was to high too inspect all events. Since June 2006 these algorithms are performed nightly and the results are inspected after every night. In this period flare identification algorithm identified a single flare event which occurred on 2006.11.28 06:03 UT and was identified as outburst of flare star GJ 3331A / GJ 3332 (RA = $05^h06^m50^s$, Dec = -21°35'). The light curves of this flare are shown in Figures 4.5 and 4.5. In this case the star was already present in the "Pi of the Sky" star catalog and the algorithm found sudden increase of brightness. The star has risen by Δm=0.55^m from m=9.58 to $m_{max} = 9.03$.

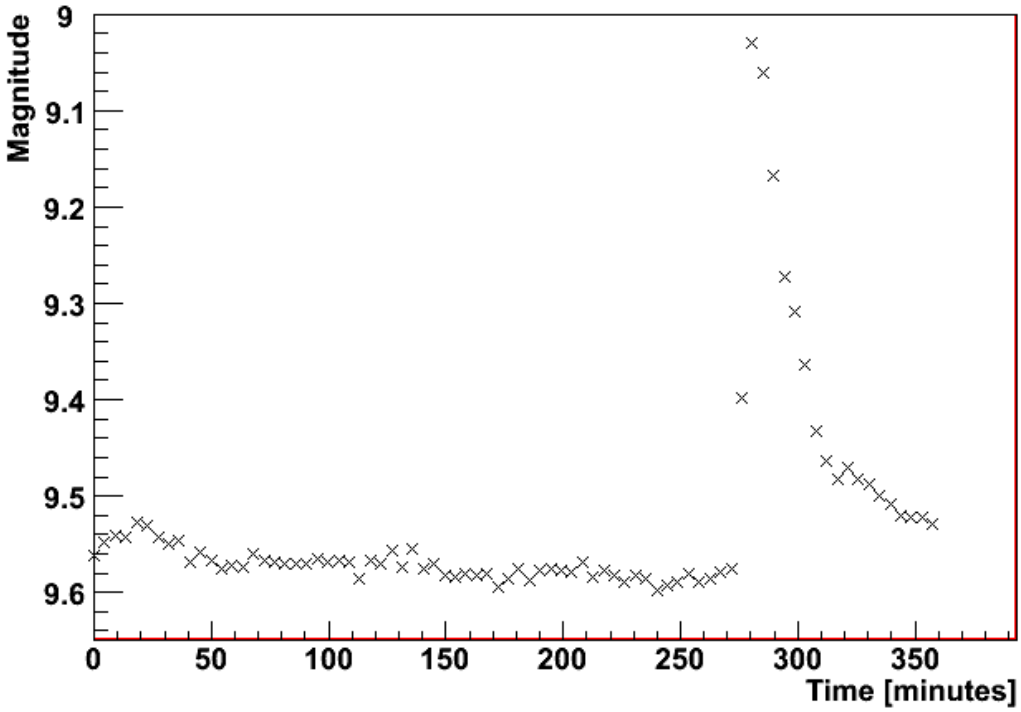

Figure 4.4: Light curve of outburst of flare star GJ 3331A / GJ 3332 in 240s time resolution

The off-line algorithm looking for objects of nova-like signature didn't find any interesting astrophysical event. The only events of astrophysical origin detected by this algorithm were background events caused by asteroids. They are identified as new objects because of changing their positions. They orbit the Sun and every

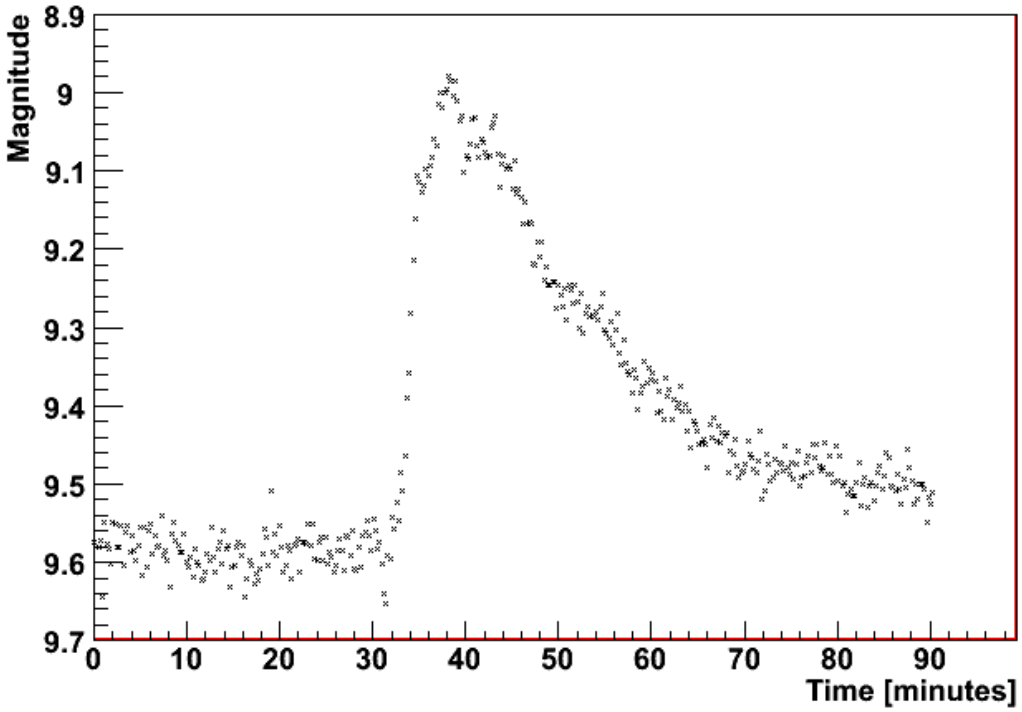

Figure 4.5: Light curve of outburst of flare star GJ 3331A / GJ 3332 in 10s time resolution

night can be observed in different position, thus they are cataloged as new objects in the star catalog. The algorithm identified 45 planetoids (till 2007.04.25). They turned out to be a good tool for testing the algorithm performance. Example of the asteroid detected by the prototype is shown in Figure 4.6. In the future version of the algorithm they will be rejected automatically. The algorithm was well checked and it is able to detected sudden appearance of nova-like objects in the sky.

4.4 GRB observation results

The main goal of the experiment are observations of the optical counterparts of GRBs. The prototype was not able to cover the whole FOV of any of the satellites. This strongly limits the chances to observe GRBs when they are going on. According to observing strategy (Sec. 2.2.2.7) in the first period the system followed the HETE's satellite FOV and since the June 2006 the SWIFT's FOV was observed. The Table 4.8 shows statistics of GRB observations for those two periods.

Period	Apparatus Off	North hemi-sphere	Daytime	Below horizon	Clouds	Outside FOV	Inside FOV
20040701-20050809	1	18	40	8	4	16	2
20060520-	4	2	36	15	2	6	0

Table 4.8: GRB observations statistics in period from 2004-07-01 to 2005-08-09 (number of GRB was 89) and since 2006-05-20 (actual number of GRB is 65)

Only 24 GRBs occurred during the night and in the range of the telescope. In two cases the GRB position was observed just before, during and after the GRB. All GRB events which were observed by the prototype are listed in table 4.9. The total time of the prototype observations normalized to 4π is $T_{total} \approx 2$ days, assuming 2-3 GRB occurring every night this means that on average 4-6 GRB events should occur in the prototype's FOV in the whole period, which is in agreement with the observed 2 events.

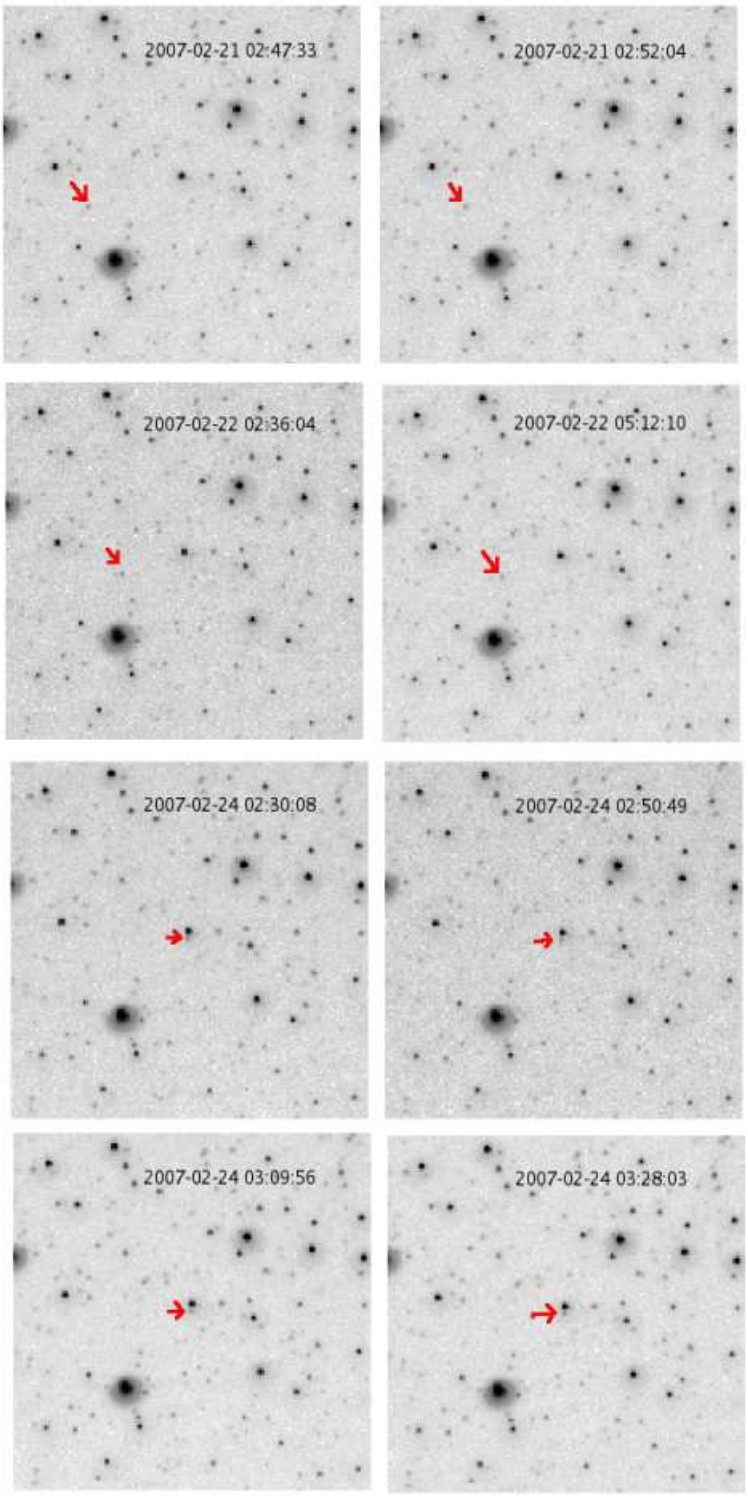

Figure 4.6: Observations of asteroid Papagena in time period from 2007-02-20 to 2007-02-24

GRB	Alert Source	Reaction Time [sec]	Magnitude/Upper Limit	GCN Cirrcular
GRB070126	SWIFT	180	$>13.0^m$	-
GRB061202	SWIFT	170	$>14.3^m$	GCN5891
GRB060719	SWIFT	65	$>12.8^m$	GCN5346
GRB060607	SWIFT	124	$>13.4^m$	GCN5241
GRB050607	SWIFT	60	$>12.5^m$	GCN3526
GRB050522	INTEGRAL	75	$>11.0^m$	-
GRB050412	SWIFT	<0	$>11.5^m$	GCN3240
GRB040916	HETE	1020	$>13.0^m$	GCN2725
GRB040825A	HETE	<0	$>10.0^m$	GCN2677

Table 4.9: Early observations of GRBs performed by the prototype in period 2004-2007

4.5 Interpretation of the results

4.5.1 Limits on orphan afterglows

As it was described in the Section 1.0.2.1 it is expected that big fraction of the GRB events cannot be observed from the Earth in γ-rays. These are so called orphan afterglows, only the optical counterpart of such events may be observed from the Earth. The number of such events on the whole celestial sphere depends on the model and is not well established. The "Pi of the Sky" experiment is a good tool to observe these kind of events due to its own flash recognition algorithm. The observed flash events can be considered as orphan afterglows and used to derive the limitations on the number of such events on the whole celestial sphere. This limit can be calculated from the following formula :

$$N_{oa} = \frac{N_{flashes}}{T_{obs} \cdot \epsilon} \tag{4.1}$$

where T_{obs} is the total observing time normalized to 4π, $N_{flashes}$ is the number of optical flashes observed in this period and ϵ is the efficiency of the flash recognition algorithm. The total time of the observations for data used in this thesis is $T_{obs} = 2$ days and the number of flashes depends on the type of the algorithms considered. The Table **??** shows the results for single image flashes and flashes visible on more then 1 image of unknown origin.

TABELKA Z WYLICZNIEM LIMTOW NA DZIEN

Current theoretical predictions coming from the models of the GRB events and population in the Universe predict $\approx$ ... orphan afterglow of brightness ...

per day. The limits derived from the prototype data are ... .

4.5.2 Full system observations predictions

Thanks to launch of the SWIFT satellite in the fall of 2004 it is possible to observe optical counterparts of GRBs at very early time. Table 4.10 shows the number GRBs observed by SWIFT in subsequent years. On average less then 1/2 of all GRBs have observed optical counterparts. Figure 4.7 shows the distribution of the minimal time of the optical counterparts observations and reaction time. The reaction time peaks at $\approx$ 110 seconds, and it is clear that early times (at T=0) are very poorly covered.

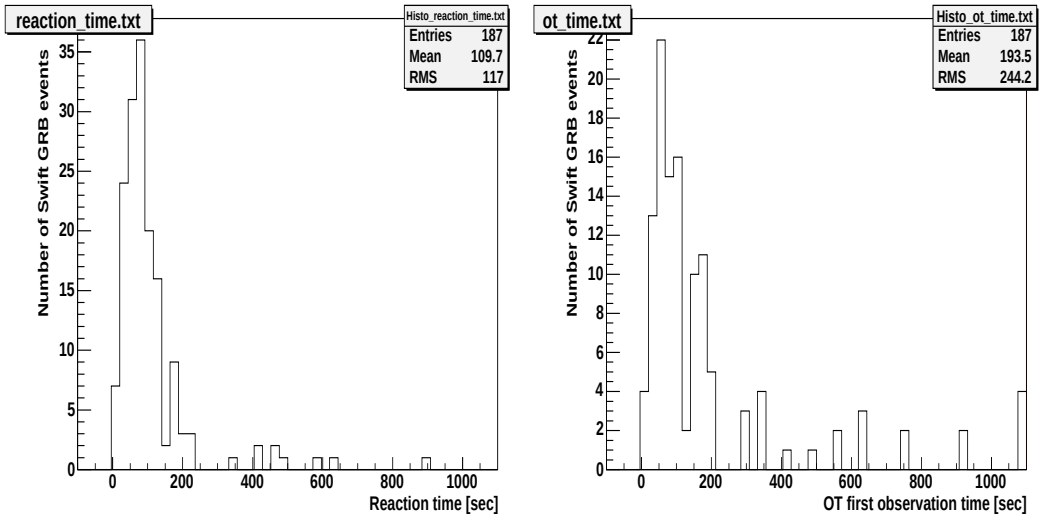

Figure 4.7: Distribution of reaction time (left plot) and minimal time of OT observation (right plot). Only Swift GRBs were taken into account

Year	Number of GRB	Number of GRB with observed OT
2004	3	0
2005	125	40
2006	119	60
2007	14	4

Table 4.10: Number of GRBs observed by SWIFT with number of those for which optical counterpart was observed

GRB	Maximum brightness	Filter	First observation time [s]
GRB 061126A	12.3	none	20.87
GRB 051111A	13.0	none	26.9
GRB 060117A	11.5	R	124

Table 4.11: Most bright optical counterparts of GRBs observed by SWIFT, which could be detected by "Pi of the Sky" if it appears within the range of the telescope

The full "Pi of the Sky" system will be a perfect tool for covering this early period of optical observations. According to the current SWIFT data it is possible to make conservative estimate on the number of events for which the system would see the signal from GRB optical counterpart. All GRBs detected by SWIFT were analyzed and events with observations in the first 15 minutes after the burst were selected. They were divided in to 4 classes :

- **Sure** : events which have observations brighter then 13^m and would certainly be detected by the "Pi of the Sky" experiment
- **Optimistic** : events which have observations brighter then 15^m and early observation of such events would have great scientific significance
- **Quasi-Optimistic** : events which don't have observations brighter then 15^m, but light curve of early points extrapolated to time T=0 sec is brighter then 15^m.
- **Pessimistic** : events with brightest measurements far below limiting magnitude of the "Pi of the Sky" experiment

Table 4.12 shows the classification results. The estimate of the number of events to be observed by the full system are derived from the following formula :

$$\begin{aligned} N_{\Pi-OT/year} &= \frac{N_{total}^{SWIFT}}{T_{total}^{SWIFT}} \times f_{Swift-FOV} \times f_{coll}, \\ f_{Swift-FOV} &:= f_{night} \times f_{satfov}, \\ f_{tot} &:= f_{night} \times f_{satfov} \times f_{coll} \end{aligned} \tag{4.2}$$

Observation prediction for "Pi of the Sky"	Number of GRB (2005-2006)
Sure	3
Optimistic	20
Quasi-Optimistic	15
Pessimistic	206

Table 4.12: Number of GRBs in different prediction groups

where N_{total}^{SWIFT} is the number of events of given type observed by SWIFT in period T_{total}^{SWIFT}, f_{night} is the fraction of time which can be used for observations at given site, f_{satfov} is the fraction of night when SWIFT satellite is above the horizon, f_{coll} is the fraction of nights when data could be collected (the weather was good and the system was operational). The value $f_{Swift-FOV}$ was calculated for period of one year under assumption that system collects data only when $h_{SUN} < 10°$ and using SWIFT pointing information from period 2005-2006. The value of FOV acceptance f_{satfov} was calculated in conservative way assuming that it is possible to observe SWIFT's FOV when it is at h>14° above horizon. However due to large FOV of SWIFT it is possible to observe part of its FOV also when it is h<14°. The error coming from this should not exceed few percent. The value of $f_{coll} \approx 83\%$ coefficient was estimated for 20060601-20070311 collecting period. The values of coefficients for different sites are shown in Table 4.13.

Site	Value of $f_{Swift-FOV}$	f_{coll}	f_{tot}
LCO	15%	83%	12%
La Palma	23%	<80%	18%

Table 4.13: Values of coefficients used to calculate π acceptance for SWIFT GRBs

The final estimate for event numbers for given class of events are listed in Table 4.14. It can be seen that number of sure positive observation is very small. However, it must be stressed that period at T=0 moment is almost unknown. The estimation is conservative, GLAST satellite will increase the value of f_{satfov} coefficient.

There is also advantage of observing short GRBs when they are going on. The numbers shown in Table 4.14 are safe estimates according to what is already

OT Class	Site	# of OT in 2006-2007	Expected # of observations / year
Sure	LCO	3	0.18
Optimistic	LCO	20	1.2
Quasi-Optimistic	LCO	15	0.9
Sure	La Palma	3	0.27
Optimistic	La Palma	20	1.8
Quasi-Optimistic	La Palma	15	1.35

Table 4.14: Expected number of events observed by full π system

observed. They show big chances of observing positive signal of great scientific significance in the first 2 years of the experiment and thus can be considered as optimistic. It is important that in fact only one single optical counterpart of Swift GRB was observed in T=0 by optical telescope [6], thus it is not well known what can be expected. The predictions in Table 4.14 are determined according to observations performed several dozens of seconds after the burst.

Appendix A

Technical Details on system controll

A.1 Libraries and programs for data aquisition and analysis

Table below lists libraries which were in big part developed by author and are components of Data AQuisition and Data Analysis programs. Paths to source location are relative and should be proceeded with /opt/pi/dev/pisys/daq/src/.

NAME	LIBRARY FILE	SOURCE LOCATON	DESCRIPTION
baselib	libbaselib.a	cmn/baselib/	low level common classes for file handling, date/time etc
mathlib	libmathlib.a	cmn/mathlib/	mathematical functions
log4pi	liblog4pi.a	cmn/log4pi/	interface for logging to sysylog
pidblib	libpidblib.a	ccd/pidblib/	postgres database interface
asaslib	libasaslib.a	ccd/fitslib/asaslib/	procedures adopted from ASAS experiment
myfitslib	libmyfitslib.a	ccd/fitslib/myfitslib/	classes for `fits` files I/O
cfglib	libcfglib.a	ccd/cfg/	definitions of default values of paramters
picamdrv	libpicamdrv.a	ccd/ccddriver/picamdrv/	camera driver library
ccdscript	libccdscript.a	ccd/ccdscripts/	script generator and pointing classes
ccdsat	libccdsat.a	ccd/ccdsat/	interfece to external library for calculating satellites positions
ccdlib	libccdlib.a	ccd/ccdlib/	main data aquistion library, implementing classes for image collection and on-line analysis
ccdinterface	libccdinterface.a	ccd/ccdinterface/	interface classes to communicate with other parts of the system and external world
ccdastro	libccdastro.a	ccd/ccdastro/	library for astronomical fomulae

Table A.1: Most important libraries developed for data collection and analysis

Table below lists most important programs for collecting data and photometry and astrometry.

PROGRAM NAME	SOURCE LOCATION	DESCRIPTION
ccdsingle	ccd/TESTY/ccdsingle/	program for collection and analysis of images from single camera
ccddouble	ccd/TESTY/ccddouble/	program for collection and analysis of images from two cameras
test2K2K	ccd/TESTY/test2K2K/	program for operating cameras in interactive mode. The usage of the program is the following : test2K2K CAMERA OPTIONS In order to specify camera, its name k2a, k2b, etc can used or if unknown the number 0,1,2, etc. In order to specify IP adress of the camera option -eth=100.100.100.1 must be provided. For more details see web page : http://hep.fuw.edu.pl/ msok/test2K2K.html
piastrometry	ccd/TESTY/piastrometry/	program for running astrometry
piphoto	ccd/TESTY/piphoto/	program for fast photometry algorithm

Table A.2: Most important programs for data aquisition and analysis. All programs show short help when launched with option -h

A.2 Description of FITS header keywords

The table below lists keywords saved to fits header files.

KEYWORD	DESCRIPTION
Standard FITS format keywords	
SIMPLE	Mandatory, T if conforms to standard, F otherwise
BITPIX	Mandatory, number of bits representing data value
NAXIS	Mandatory, number of axis in dat array
NAXIS1	Mandatory, number of elements along X axis
NAXIS2	Mandatory, number of elements along Y axis
EXTEND	T if FITS may contain extensions
BZERO	Used in case array value are not physcial, but : physical_value = BZERO + BSCALE * array_value
BSCALE	Used in case array value are not physcial, but : physical_value = BZERO + BSCALE * array_value
Observed object description	
OBJECT	Name of object or field observed, project conention is syntax : H0000+00, where letter is first letter of object name and digits stand for coordinates : HHMM±DD
ROTATE	if image is rotated (0 / 1)
DRVFLIP	if imaged was fliped by driver on PC side (0 / 1)
RA	Right Ascension of image center in hours decimal
DEC	Declination of image center in degrees decimal
HA	Hour angle of image center in hours decimal
AZIM	Azimuth of image center in degrees decimal
ALT	Altitude of image center in degrees decimal
ZENITH_D	Zenithal distance of image center in degrees decima
OBSMODE	Observation mode - 0 - mount not tracking, 1 - mount tracking
AVERAGE	Mean pixel value on image
RMS	RMS of pixel value on image
Observing Site	
ORIGIN	Name of project
SITE	Code of site, LCO - Las Campanas, BRW - Brwinow
TELLONG	Geographical longitude [deg]
TELLAT	Geographical latitude [deg]
TELALT	Altiutude above the see level [m]
Instrument	
INSTRUME	Name of instrument
CAMERA	Name of camera
CAMOPTIC	Camera optic description
FILTER	Used filters
PIXSCALE	Angular size of pixel [arcsec]
PIXSIZE	Physcial size of pixel [μm]

KEYWORD	DESCRIPTION
Exposure id	
OBSERVER	Observer name
DIMAGE	Image number (during single night)
NIMAGE	Obsolate
SOFTWARE	Software version
BUILD	Software build date and time
DRVTYPE	Camera driver type
CAMID	Camera ID (2-k2a, 3-k2b)
CAMIIDX	Camera internal index
FILENAME	FITS file name
Exposure settings	
EXPTIME	Require exposure time [sec]
REXPTIME	Measured exposure time [sec] (not very precise)
SHUTTER	values : OPEN / DARK
SHUTMODE	Shutter mode OPENED - permanently opened, NORMAL - open/close mode
ADCGAIN	Analog to Digital Converter (ADC) gain value
ADCBIAS	ADC offset value [mV]
ADCGSET	ADC gain setting value
LNAGAIN	Low Noise pre-Amplifier (LNA) gain value x8 or x20 (current value is x8)
ADCBSET	ADC offset setting value
ADCRANGE	2V or 4V
ADCCLAMP	ADC Clamping 2V or 4V
ELECGAIN	Gain of camera [e/ADU]
COOLING	Cooling enabled / disabled (YES / NO)
ABINN	Analog binning (disabled)
SBINN	Software binning (disabled)
SPEED	OBSOLATE
SPEEDMH	Speed description Vertical / Horizontal - to be corrected (horizontal is missing) !!!
MPP_BC	Multi-pinned phase or BC mode
RO-TIME	Measured data transfer time [sec] (from camera RAM to PC)
CROTIME	Measured chip readout time [sec]
FOCUS	value of current step motors position [steps]
HITLENS	Lens hitting ON/OFF
SAVEAREA	Area of chip saved in this FITS file [x_start y_start x_end y_end]
USBMODE	USB mode (1.1 or 2.0)
FPGAVER	FPGA firmware version date
CPRSVER	Cypress firmware version date
VERDESC	Camera Firmawre version description
RNOISE	readout noise [ADU] - to be verified !!!
RELNOISE	readout noise [e] - to be verified !!!
Exposure environment	
CHIPTSET	Required chip temperature [Celsius]
CHIPTEMP	Measured chip temperature [Celsius]
CHIP_TEM	Measured chip temperature [Celsius]
CASTEMP	Measured case temperature [Celsius]
AMBTEMP	Measured ambient temperature [Celsius]
CAMHUMID	Measured camera humidity
AMBHUMID	Measured ambient humidity
INTRTEMP	Not implemented
AIRMASS	Airmass - not implemented
DOME	Dome status : OPENED / CLOSED
Exposure date	
UT-START	UT time of image start
DATE-OBS	UT date of image start
DATE_OBS	Local date of image start
TIME_UT	unixtime of image start [sec]
UT-END	UT time of image end
TIME_OBS	UT time of image end
DATE-END	UT date of image end
LOCTIME	local time of image start
LOCDATE	local date of image start
EPOCH	current epoch
EQUINOX	current equinox
ST	siderial time [radians]
JD	Julian date
HJD	Heliocentric Julian date
Astrometry	
ASTROOK	Astro OK or FAILED (1 / 0)
POSANGLE	Rotation angle [deg]
AST_ORD	Order of transformation equation (4)
ASTUTIME	unixtime of astrometry (if not performed on every image)
AST_ERR	Error of astrometry [pixels]
PAR_X_0, PAR_X_0, ... , PAR_X_13	Coefficients of astrometric transformation
PAR_Y_0, PAR_Y_1, ..., PAR_Y_13	Coefficients of astrometric transformation
Photometry	
NSTARS	Number of stars detected on image
Mount	
MOUNTRA	Right Ascension as obtained from mount [hours decimal]
MOUNTDEC	Declination as obtained from mount [degrees decimal]
MOUNTAZ	Azimuth as obtained from mount [degrees decimal]
MOUNTTRK	Tracking mode as obtained from mount [0-non tracking, 1-tracking]
MOUNTTM	Timestamp of mount information
MOUNTHA	Hour angle as obtained from mount [hours decimal]
MOUNTALT	Altitude as obtained from mount [degrees decimal]
MOUNTDTM	Timestamp of mount information [unixtime - sec]
MOUNTMV	If image taken just after mount move (0 / 1)

Table A.3: Keywords in FITS files

A.3 System log files

The most important log files are listed in table below :

LOG FILE NAME	DESCRIPTION
pi.log	log file from all modules written by syslog daemon
run_daq_YYYYMMDD.out	standard output from DAQ program (ccdsingle or ccddouble)
mount.logfile_YYYYMMDD_HHMMSS	mount controll program log file
piman.logfile_YYYYMMDD_HHMMSS	piman server log file
gcn_server.log	log file of listener program receiving and redirecting GCN messages to gcn program
gcn.logfile	log file of gcn program sending alerts to piman server
gcn_imalive.logfile_YYYYMMDD	log file containing I'am alive packets from external GCN server
integral_pointdir.log	INTEGRAL satellite pointing information obtained from GCN server
swift_pointdir.log	SWIFT satellite pointing information obtained from GCN server

Table A.4: Most important log files from π-system, files are stored in directory /opt/pi/dev/pisys/log/, unless different path is specified, YYYYMMDD is date (for example 20070101) and HHMMSS is time (for example 200112)

A.4 DAQ controll from piman and pishell

PISHELL COMMAND	FUNCTION NAME	INPUT	DESCRIPTION
start_daq	StartDAQ	-	starts dark collection and sets daq status to started
-	IsDaqStarted	-	returns 1 if DAQ is already started, 0 otherwise
start_analysis	StartAnalysis	mount position (RA,DEC)	sets new position and starts image collection and analysis
stop_analysis	StopAnalysis	-	stops image collection (but waits for last image to be finished)
stop_analysis_nowait	StopAnalysisNoWait	-	stops image collection without waiting for last image
-	SendAlert	GCN alert time and position	informs DAQ about GCN trigger which is being observed
-	SetOnTriggerPosition	Alert position (RA,DEC)	starts image collection after trigger is received
fast_post_to_daq	CorrectMountPosition	mount position (RA,DEC)	corrects mount position
dodarks	DoDarks	N	collects N dark images
take_npictures	TakeNPictures	N	collects N images
take_npictures_synchro	TakeNPicturesSynchro	N	collects N images in synchronize mode
set_cooling	SetCoolingOnOff	temperature and camera	sets temperature for given camera
change_param	ChangeParam	param name, value and camera	changes parameter of daq
stat	GetStatus	-	returns status of daq program
set_fits_key	SetCustomKey	`fits` header key name and value	sets value of specified `fits` header key

Table A.5: Functions exported by DAQ

A.5 System controll commands

Example of night script for automatic system controll :

```
# auto-generated script
# night : 20060605
# camera : Cannon EOS f=85mm
# system start time is : 20060605_180643 local ( 20060605_220643 UT )
# PRIMARY SATELLITE = INTEGRAL
# SECONDARY SATELLITE = HETE
# SUN sets at 1840 LCO time, at (AZ,H)=(110.44,-9.90) [deg]
# SUN rises at 0645 LCO time
# SWIFT at 20060605_180643 local time is at (RA,DEC)=(144.94,15.00)
# HETE info file date : 20060605_072000
# HETE RA=249.05=16h36m12.00s DEC=-63.06
# HETE rises above horizont at 327, sets at 116
#  hete at h_hete >= 30.00 at 1901
# MOON RA=190.54=12h42m10.68s DEC=-5.28 illum = 72.06 %
# MOON will set at 20060606_025143, illum = 73.46 %
# INTEGRAL RA=125.47=08h21m52.32s DEC=-47.51
# INTEGRAL rises above horizont at 737, sets at 36
piman    0 cron_point_hete_off
piman    1 exec_script_synchro(startup.pish)
#
daq      1825 start_daq
piman    1825 auto_ag_mode_on
piman    1830 manual_mode_off
# Following INTEGRAL at (RA,DEC)=(125.47,-47.51) (az,h)=(46.77,55.89)
# Closest field (RA,DEC)=(128.50,-50.00), (az,h)=(41.23,56.76) => OBJECT=I0834-50
# At 1920 field (RA,DEC)=(128.50,-50.00) of object INTEGRAL is at (az,h)=(45.10,50.74)
# turning OFF cron
piman 1840 manual_mode_on
# internal 1149547243 goto_ra_dec(128.50,-50.00)
piman    1840 cron_point_hete_off
internal 1840 goto_ra_dec_auto_corr(128.50,-50.00,I0834-50)
mount 1840 raw_cmd(autoguide on)
internal 1840 send_pos_to_mount
mount 1840 stat
piman    1845 cron_send_pos_to_mount_on
internal 1850 send_pos_to_mount
mount 1850 stat
# turning ON cron
piman 1850 manual_mode_off
# End of tracking (ra,dec)=(128.50,-50.00) at 2155 , (az,h)=(45.26,26.05) deg
piman    1900 manual_mode_on
piman    1900 cron_point_hete_off
piman    1900 cron_send_pos_to_mount_off
piman    1900 exec_script_synchro(scan_evening.pish)
piman    1930 manual_mode_off
# Following INTEGRAL at (RA,DEC)=(125.47,-47.51) (az,h)=(49.99,47.46)
# Closest field (RA,DEC)=(128.50,-50.00), (az,h)=(45.79,48.99) => OBJECT=I0834-50
# At 2010 field (RA,DEC)=(128.50,-50.00) of object INTEGRAL is at (az,h)=(47.14,42.62)
# turning OFF cron
piman 1930 manual_mode_on
```

```
# internal 1149550254 goto_ra_dec(128.50,-50.00)
piman    1930 cron_point_hete_off
internal 1930 goto_ra_dec_auto_corr(128.50,-50.00,I0834-50)
mount 1930 raw_cmd(autoguide on)
internal 1930 send_pos_to_mount
mount 1930 stat
piman    1935 cron_send_pos_to_mount_on
internal 1940 send_pos_to_mount
mount 1940 stat
# turning ON cron
piman 1940 manual_mode_off

# End of tracking (ra,dec)=(128.50,-50.00) at 2145 , (az,h)=(45.64,27.43) deg
# Following HETE at (RA,DEC)=(249.05,-63.06) (az,h)=(334.65,47.49)
# Closest field (RA,DEC)=(255.00,-60.00), (az,h)=(328.70,46.52) => OBJECT=H1700-60
internal 2140 send_pos_to_mount
# At 2225 field (RA,DEC)=(255.00,-60.00) of object HETE is at (az,h)=(332.55,50.84)
# turning OFF cron
piman 2145 manual_mode_on
# internal 1149558354 goto_ra_dec(255.00,-60.00)
piman    2145 cron_point_hete_off
internal 2145 goto_ra_dec_auto_corr(255.00,-60.00,H1700-60)
mount 2145 raw_cmd(autoguide on)
internal 2145 send_pos_to_mount
mount 2145 stat
piman    2150 cron_send_pos_to_mount_on
internal 2155 send_pos_to_mount
mount 2155 stat
# turning ON cron
piman 2155 manual_mode_off
# End of tracking (ra,dec)=(255.00,-60.00) at 0630 , (az,h)=(33.97,26.78) deg
# morning
piman    0545 manual_mode_on
piman    0545 cron_point_hete_off
piman    0545 cron_send_pos_to_mount_off
piman    0550 exec_script_synchro(scan_morning.pish)
piman    0615 manual_mode_off
# HETE (RA,DEC)=(249.05,-63.06) at 0615 is at (AZ,H)=(30.32,26.21)
# Waiting for HETE at (RA,DEC)=(252.96,-63.41) (az,h)=(30.32,28.00)
# Closest field (RA,DEC)=(255.00,-60.00), (az,h)=(34.32,28.39) => OBJECT=H1700-60
# At 0655 field (RA,DEC)=(255.00,-60.00) of object HETE is at (az,h)=(33.00,23.52)
# turning OFF cron
piman 0615 manual_mode_on
# internal 1149588948 goto_ra_dec(255.00,-60.00)
piman    0615 cron_point_hete_off
internal 0615 goto_ra_dec_auto_corr(255.00,-60.00,H1700-60)
mount 0615 raw_cmd(autoguide on)
internal 0615 send_pos_to_mount
mount 0615 stat
piman    0620 cron_send_pos_to_mount_on
internal 0625 send_pos_to_mount
mount 0625 stat
# turning ON cron
piman 0625 manual_mode_off
# End of tracking (ra,dec)=(255.00,-60.00) at 0630 , (az,h)=(33.91,26.55) deg
# Do not worry about order, these two commads always go just before shutdown :
piman    0640 cron_point_hete_off
piman    0640 cron_send_pos_to_mount_off
piman    0650 exec_script(shutdown.pish)
```

Figure A.1: Example of night pish script

A.6 Observation targets

The table below lists observation targets in decreasing order of priority :

OBJECT NAME	POSITION	INFO
SWIFT Satellite	-	FOV≈2 staradians
INTEGRAL Satellite	-	FOV≈ 15°
PKS 2155-304	215852-301332.0	blazar
AO 0235+164	023838+163659.0	blazar
4C 29.45	115931+291444.0	quasar
OI 158	073807+174219.0	quasar
OJ 287	085448+200629.0	quasar
3C 273	122906+020307.0	quasar
OR -017	151250-090558.0	quasar
W Com	122131+281358.0	blazar
J0210-5055	021046-510100.0	blazar
OJ 049	083148+042939.0	blazar
GeV J1832-2128	183300-213600.0	blazar
OP 151	133335+164904.0	blazar
PKS 0537-441	053850-440508.0	blazar
Mrk 501	165352+394535.0	blazar
Mrk 421	110427+381230.0	blazar
QQ Vul	200541+223959.0	polar
RR Aqr (D)	211843-020812.0	variable star
RR Aqr	211501-025344.0	variable star
RR Aqr (C)	211917-014609.0	variable star
VV Pup	081506-190315.0	variable star
EF Eri	031413-223540.0	polar
V834 Cen	140907-451717.0	polar
V2214 Oph	171201-293732.0	polar
J0458-4635	045550-461557.0	blazar
BL Hyi	014100-675326.0	polar
MR Ser	155247+185626.0	polar
4C 11.69	223236+114350.0	blazar
OS 319	161341+341247.0	blazar
3C 279	125611-054721.0	blazar
V347 Pav	184448-741833.0	polar
GQ Mus	115202-671220.0	polar
RR Aqr (G)	211622-023914.0	variable star
PKS 1229-021	123159-022405.0	blazar

Table A.6: List of objects to be automatically followed in order of priority

Appendix B

Technical Details on Data Analysis

B.1 Parameters of on-line flash recognition algorithm

Parameter Name	**Parameter Name in configuration file**
LaplaceType	CCD_LAPLACE_TYPE
T_n	CCD_NEW_LAPLACE_TRESHOLD_IN_SIGMA
T_v	CCD_MAX_LAPLACE_ON_OTHER_IN_SIGMA
N_{aver}	CCD_AVERAGE_OF_PREV_N
T_{MinLap}	CCD_MIN_LAPLACE_ON_OTHER
N_{MaxTv}	CCD_MAX_NUMBER_OF_EVENTS_AFTER_TV
enable/disable	CCD_SKIP_OVERLAPS
$R_{overlap}$	CCD_OVERLAP_REDIAL
T_{shape}	CCD_CHECK_EVENT_SHAPE
T_{black}	CCD_BLACK_PIXELS_RATIO
T_{hot}	CCD_REJECT_HOT_PIXELS_BY_AVERAGE_TRESH
N_{IfMore}	CCD_SKIP_IF_MORE_THEN
R_{IfMore}	CCD_SKIP_IF_MORE_THEN_REDIAL

Table B.1: Table translating FLT paramter names into real names used in configuration file

Parameter Name	Parameter Name in configuration file
R_{coinc}	CCD_COIC_RADIUS_IN_SEC
$N_{confirm}$	CCD_CONFIRM_ON_N_NEXT_FRAMES
R_{sat}	CCD_SAT_REJ_RADIUS_IN_SEC
R_{star}	CCD_MATCH_STAR_TO_CAT_RADIUS_IN_ARCSEC
Mag_{max}	CCD_STARCAT_MAX_MAG
N_{track}	CCD_NUM_BACK_FRAMES_FOR_TRACK
χ^2	CCD_MAX_CHI2_FOR_POINT_TO_MATCH_LINE
χ^2	CCD_MAX_CHI2_IN_TRACK

Table B.2: Table translating SLT paramter names into real names used in configuration file

Cut	Parameter Name
T_{hough}	CCD_HOUGH_TRANSFORM_TRESH
T_{hough_distr}	CCD_HOUGH_DISTR_TRESH
T_{hough_eight}	CCD_HOUGH_DISTR_MAX_LIMIT
N_{tstars}	CCD_SLT_TYPICAL_STARS_COUNT
R_{clouds}	CCD_SLT_REJECT_FRAME_IF_LESS
L_{diff}	CCD_LAP_DIFF_MIN_RATIO
T_n^{TLT}	CCD_NEW_LAPLACE_TRESHOLD_IN_SIGMA

Table B.3: Table translating TLT paramter names into real names used in configuration file slt.cfg

References

[1] A. Panaitescu, *Models for achromatic light-curve breaks in gamma-ray burst afterglows: jets, structured outflows and energy injection*, Monthly Notices of the Royal Astronomical Society **362** (2009) 921–930. (document), 1.10

[2] Strong I.B. Klebesadel R.W. and Olson R.A., *Observations of Gamma-Ray Bursts of Cosmic Origin*, Ap.J. **182** (1973) 85–88. 1.0.1

[3] E. Waxman, *Gamma-Ray Bursts: The Underlying Model*, Lect.Notes Phys. **598** (2003) 393–419. 1.0.1

[4] K. Akerlof et al., *Observation of contemporaneous optical radiation from a γ-ray burst*, Nature **398** (1999) 400–402. 1.0.1, 1.0.2

[5] Goddard Space Flight Center, *The SWIFT Gamma-Ray Burst Mission*, http://heasarc.gsfc.nasa.gov/docs/swift/swiftsc.html. 1.0.2

[6] W. T Vestrand et al., *A link between prompt optical and prompt γ-ray emission in γ-ray bursts*, Nature **435** (2005) 178–180. 1.0.2, 4.5.2

[7] A. Klotz et al., *Continous optical monitoring during the prompt emission of GRB060111B*, A&A **451** (2006) L39. 1.0.2

[8] Paczynski B., *Optical Flashes Preceding GRBs*, astro-ph/0108522 (2001). 1.0.2

[9] Warner B., *Cataclysmic Variable Stars*, Cambridge University Press (1995). 1.0.3

[10] Trinamic Company, *Trinamic Motion Control Home Page*, http://www.trinamic.com/tmc/render.php. 2.2.1

[11] Fairchild Imaging, *Fairchild Imaging Product Page*, http://www.fairchildimaging.com/products/fpa/custom/ccd_442a.htm. 2.2.1

[12] A.Burd et al., *Low noise CCD cameras for wide field astronomy*, Proceedings of SPIE **6159** (2005) 160–166. 2.2.1

[13] G. Pojmanski, *The All Sky Automated Survey Home Page*, http://www.astrouw.edu.pl/ gp/asas/. 2.2.1, 2.2.2.1, 3.3.1.1

[14] OGLE collaboration, *The Optical Gravitational Lensing Experiment Home Page*, http://www.astrouw.edu.pl/ ogle/. 2.2.1

[15] Karel Gardas, *MICO IS CORBA web page*, http://www.mico.org/. 2.2.2.1

[16] L. Mankiewicz K. Nawrocki P. Sitek M. Sokolowski R. Sulej W. Tlaczala J. Uzycki, G. Kasprowicz, *Data transmission protocol for*. 2.2.2.4

[17] Anders Hedstrom, *C++ Sockets Library web page*, http://www.alhem.net/Sockets/. 2.2.2.4

[18] NASA/Science Office of Standards and Technology, *Definition of the Flexible Image Transport System (FITS)*, http://heasarc.gsfc.nasa.gov/docs/heasarc/fits.html. 2.2.2.4

[19] The FITS Support Officeat NASA/GSFC, *FITS, The Astronomical Image and Table Format*, http://fits.gsfc.nasa.gov/. 2.2.2.4

[20] The PostgreSQL Global Development Group, *PostgreSQL web page*, http://www.postgresql.org/. 2.2.2.5, 3.3.1.2

[21] Spear Gordon, *The Global Telescope Network web page*, http://gtn.sonoma.edu/public/. 2.2.2.7

[22] Goddard Space Flight Center, *Swift's Burst Alert Telescope (BAT)*, http://swift.gsfc.nasa.gov/docs/swift/about_swift/bat_desc.html. 2.3.1

[23] S. D. on behalf of the Swift Instrument Team Barthelmy, *The Burst Alert Telescope (BAT) on the Swift MIDEX Mission*, Proceedings of SPIE **4140** (2000) 50. 2.3.1

[24] Goddard Space Flight Center, *Glast Mission Home Page*, http://glast.gsfc.nasa.gov/. 2.3.1

[25] Goddard Space Flight Center, *Large Area Telescope (LAT)*, http://glast.gsfc.nasa.gov/science/instruments/. 2.3.1

[26] L. W. Piotrowski M. Sokolowski K. Malek, L. Mankiewicz, *All sky scan analysis algorithm for Pi of the Sky project*, Proceedings of SPIE **6347** (2006) 0Q. 3

[27] M. Sokolowski G. Wrochna M. Biskup, L. Mankiewicz, *Databases for the Pi of the Sky experiment*, Proceedings of SPIE **6347** (2006) 0T. 3

[28] H. Pedersen, *Rene Descartes Optical Telescope*, http://www.astro.ku.dk/ holger/rdot.html. 3.2.2

[29] Copenhagen University Observatory, *The Tycho-2 CatalogueInformation and Links*, http://www.astro.ku.dk/ erik/Tycho-2/. 3.2.2, 8, 11