Munipack

Filip Hroch

Munipack
by Filip Hroch

Copyright © 1998-2000 by

Table of Contents

List of Examples

Preface

já jsem šulín This document describes Munipack photometry package. The Munipack is set of independent routines for astronomers. Our package philosophy is clear: We need a few utilities for basic operations on CCD images. This utilities are sample programs or scripts and any user may run it from shell or from its own script. The utilities do not resolves one type of problem.

The "raw" use of this routines is simple. but it is not trivial or fast for everyday use, therefore a few "frontends" may help you for effective work with munipack. See ... for quick start.

The first part is introduction to astronomical photometry and user’s view to Munipack as package for make it.

The second part is detailed technical description Munipack routines for deep inside.

Chapter 1. Introduction, history and jokes

The CCD photometry is the most exciting part of the observational astronomy. But the relevant processing of the tusand of images is the most difficult work of the scientist. Since 1998, I working to developing of a system for automarize its and on fully matured algorithms to this. I started from the famous DAOPHOT package by P. B. Stetson. The name Munipack is inspired by DAOPHOT, P.B. Stetson wroted DAOPHOT at Dominion Astrophysical Observatory therefore I use Munipack originated from Masaryk University in Brno. Note, that T.G. Masaryk was a first Czech(oslovak) democratic president. I think, that name like Hrophot or Fiphot is a very stupid name for any quality code.

The CCD photomtery is a art of data processing. Any algorithm (Munipack's component or another) can diminantly help of this processing but it must be used with naximum warranty. Are you a astronomical photometry guru?

Chapter 2. Getting Munipack, other photometry sources

"Official" homepage of the Munipack drifts over this adresses during last years on *IAN.CZ* (http://www.ian.cz/munidos/), on *Altamira* (http://www.altamira.asu.cas.cz/munidos/) and probablly on many others.

All of the references contains obsolete versions. Please do not use it. The recent release available on my office computer *integral/sci.muni.cz/pub/munipack.new* (ftp://integral.sci.muni.cz/pub/munipack.new/) represents actual frozen development version. It is a fully functional but open for future changes.

My friend Rudolf Novak wrote DOS/WIN port of the prehistorical version of Munipack named MUNIDOS. This package contains main components for photometry and additional routines for user friendly interface. Many people reports MUNIDOS as new DOS clone (like MSDOS, DRDOS or FreeDOS). On base it, we (I) omited this name.

Chapter 3. Camera, telescope, monitor at night

An eccelent astronomer grabbs images every clear night.

Chapter 4. Brief description of package programs

Munipack is astronomical photometry package for a data processing of the time series of the CCD images. The munipack is collection of programs to the aperture photometry. It consists from some utilities briefly described here.

konve, picko

Munipack uses the FITS image format (provided by FITSIO library) as I/O format. Many of users do not works with FITS as the camera specific format for its CCD's. This utilities help him. Konve is useful for SBIG camera users (ST format) and picko for Ch. Buil's cameras format (PIC).

meandark

The meandark make the mean dark frame. A lot of (minimum 2) dark with the same temperature and exposure time uses to derive the arithmetical mean of the dark images. This meandark provide the best fotometry result (a few 0.001 mag).

autoflat

The autoflat tries to derive of the flat frame from the observed images of the stars fields (a regular scientific exposures, no twilight or cloud exposures). A lot of (practical minimum about 20) of the different images combines by the robust mean computing method to the flat frame. The autoflat eats many MB of the memory and CPU. The carefull use may make a good results.

darkbat, flatbat

Two utilities to correct of the flat frame and subtract of the dark frame.

muniphot

Muniphot make aperture photomery of CCD images. The core of this package. Munipack is modified DAOPHOT.

munimatch

Munimatch searches for cross references between coordinate files.

munilist

A simple utility to list useful informations from processed files.

Chapter 5. Munipack - Quick way to use of the Munipack

Direct usage of the Munipack utilities usually is the best way to use of the Munipack, but it is not easy. Many users require a simple command (or click) to run of photometry. Generally, it is not possible because astronomers develops a wide range of problems.

There are a few utilities (scripts) to easy manipulation with images. You may modify it with respect to your needs.

conve

Conve is not Konve! It converts ST images to FITS images. Conve is a SH script to more simple use of konve. Typical usage:

```
computer$ ls *.st8
image1.st8
image2.st8
computer$ conve *.st8
CONVE Version X.X Copyright (C) 2000 F.Hroch, Masaryk University, Brno, CZ
image1.st8 -> image1.fits
image2.st8 -> image2.fits
computer$ I am happy!
bash: I: command not found
computer$ :-)
```

This script is not extremally sophistic, but works ok. Optional parameter is regular expression (mask. expression with wildcards) which list of image names and converts its. If image name contains .st4 or .st5 or .st6 or .st7 or .st8, replace this suffix by the '.fits'. The images with other suffixes recognize according to first line (ST images contains keywords 'ST' and 'Image') and add suffix '.fits' to its name after conversion. I highly recommends use only st? (or ST?) suffix because many computer systems recognized file type on its base.

QMPhot

QMPhot - Quick Munipack Photometry. QMPhot is sofictical frontend to Munipack routines. It make a photometry of the CCD images serie. User only needs the set of images and (optionaly) dark images anf flat field images shapshots. This script make an another script named `qmphot.sh`. You can simply run

the command for complete photometry or edit this script (and parametrical files) before run it. We recommends run this script as background process.

Syntax of this routine is very simple: **qmplot [-h] [-d dark(s)] [-f flat(s)] [-i image(s)]** Option command `-h` give you a short help. The filenames after `-d` and `-f` specify dark frames and flat fields images and filenames after `-i` scientific images. All this options are optional, but the photometrical results after processing images without corrected dark and flat images are a very poor.

We give a typical session as example of use and for clarity.

Example 5-1.

Give help

```
computer:~/data$ qmphot -h
QMPhot - 0.0    Quick Photometry with Munipack
Usage: qmplot [-h] [-d dark(s)] [-f flat(s)] [-i image(s)]

  This script help you to create simple script to photometry.
  Create shell script with commands to:
  convert ST files to FITS, create mean dark and flat,
  correct scientific images and do photometry and matching.

computer:~/data$
```

Example 5-2.

Now, we have some files to photometry. We made dark frames `dark1.fits`, `dark2.fits`, `dark3.fits`, `dark4.fits` and `dark5.fits` during our observation. We have a some flats `flat1.fits`, `flat2.fits`, `flat3.fits` shoted at twilight. The scientific images of the interesting object are: `object1.fits`, `object2.fits`, `object3.fits` .. `object999.fits`. We invoke this script:

```
computer:~/data$ qmphot -d dark?.fits -f flat?.fits -i object*.fits
```

Chapter 6. Konve

Konve is utility for convert ST-4X, ST-5, ST-6, ST-7, ST-8, PixCel256 files (ST files) to the FITS format files. This utility is a very usefull for SBIG CCD camera owners. It converts a lot of ST images to FITS format non-interactively. All professional image handling packages works on FITS format.

All Munipack routines uses as I/O format the FITS format for compatibility with other astronomical packages. I highly recommends strict use of the FITS format for your work. Do not use ST files. The ST file is internal format of certain company and other companies/people do not uses it. I expect redefinition of this format in near future. Simply, use FITS for make you happy.

Syntax

Konve reads names of the ST files form standard input, file or command line and converts these files to FITS. The output files (default `kout????.fits` contains image array with all parameters in header. You can optionally correct exposure time or flip image. These features will be described below.

Konve is invoked with command:

konve [options] [@dirfile] [par=parfile] [tcor=?] [mask=?] [image1] ...

options

@dirfile

file with name(s) of image(s) to convert, no wildcards *,? You can use **ls** command to make it: **ls *.st8 > dirfile**

@

read image(s) name(s) from standart input

image1 .. imageN

name(s) of simple image(s) to convert, no wildcards *,?

par=parname

Parameter file. Any line of this file is directly copied to the FITS file header. Therefore, you must use FITS format. Ordinary record:

```
position     description

1 - 9        keyword (OBJECT,...)
```

```
 10           '='
 11 -
30      value (character: 'kjshfksjh', integer: 10000, real: 1.0000E6)
 30 - 79      '/' + comment ('/ UT..')
```

Comment and history record:

```
 "COMMENT" or "HISTORY" followed by any text.
```

Record length is 80 characters. Do not use ASCII characters above 127. For more detailed info about FITS view FITSIO (http://fitsio.nasa.gov). See also -g option.

tcor=seconds

Correction of time in seconds. Integer, signed or unsigned number up to your system specific long int type.

mask=string

Output mask (default = `kout????.fts`) The '*' has the same meaning as '?'.

-p parname

Parameter file. (equivalent with `par=`)

-o mask

output mask (equivalent with `mask=`)

-c number

counter initial value (default = 1)

-flip axis

flipping around axis x, y or both xy. For example, option **-flip xy** rotate image about 180 degrees.

-h

create only header of FITS file. Not help!

-help

give help

-L

display software license

-g

wrote an example of the parameter file to the file `konve.par`. You can use to generate correct file, edit it and use with `par=` option.

Examples

Example 6-1.

Give help

```
konve
```

Example 6-2.

Convert file `image.st8` to `kout0001.fts`

```
konve image.st8
```

Example 6-3.

Convert file `image.st8` to `image.fts`

```
konve image.st8 mask=image.fits
```

Example 6-4.

Convert all ST images at current directory to default mask

```
ls *.st8 | konve @
```

Example 6-5.

Convert R images of beta Lyr

```
ls betalyrR???.st8 | konve @ mask=betalyrR???.fits
```

Example 6-6.

Convert only headers of R images of beta Lyr

```
ls betalyrR???.st8 | konve -h @ mask=betalyrR???.fits
```

Warning

You must uses this utility a very *carefully.* If output mask (option -o or mask=?) is equivalent with input image, konve overwrite the input image and your *data* are *losted*. An example is the command **konve i.st8 -o i.st8** (your are overtyped suffix `.st8` and `fits`). Try allways set the permissions to read-only for input files before.

Don’t insert blanks between key (`par=`, `tcor=`,..) and argument.

Diagnostic

The error messages like "Can’t convert..","Can’t open.." usually means, that you typed wrong filename(s).

"No input files." - missed ’@’ at command line.

"Can’t create..", "Can’t write.." - disk full?

"Unknown image type." - wrong filename (suffix)?

Authors

F. Hroch, Masaryk University, Brno (mail:hroch@physics.muni.cz).

P. Pravec, M. Velen developed a part of the code for decoding ST compressed image type.

Miroslav Brož (mail::miroslav.broz@usa.net) added two file types for PixCel255 cameras, Nov 24th 1998 - Christmas format.

Chapter 7. Picko

Picko is utility for Ch. Buil's PIC image format files to the FITS format files. This utility is a very usefull for XXX(?) camera owners. It converts a lot of PIC images to FITS format non-interactively. All professional image handling packages works on FITS format.

WARNING!! CCD cameras by the Ch. Buil are a great secret for me. I haven't this one, so that I wrote this utility on base a few PIC images of an unknown origin. Please, check all output files carefully! This utility is not matured!

All Munipack routines uses as I/O format the FITS format for compatibility with other astronomical packages. I highly recommends strict use of the FITS format for your work. Do not use PIC files. The PIC file is internal format of certain company and other companies/people do not uses it. I expect redefinition of this format in near future. Simply, use FITS for make you happy.

Syntax

Picko reads names of the PIC files form standard input, file or command line and converts these files to FITS. The output files (default `kout????.fits` contains image array with all parameters in header. You can optionally correct exposure time. This feature will be described below.

Picko is invoked with command:

picko [options] [@dirfile] [par=parfile] [tcor=?] [mask=?] [image1] ...

options

@dirfile

file with name(s) of image(s) to convert, no wildcards *,? You can use **ls** command to make it: **ls *.st8 > dirfile**

@

read image(s) name(s) from standart input

image1 .. imageN

name(s) of simple image(s) to convert, no wildcards *,?

par=parname

Parameter file. Any line of this file is directly copied to the FITS file header. Therefore, you must use FITS format. Ordinary record:

```
 position     description

 1 - 9        keyword (OBJECT,...)
 10           '='
 11 -
30      value (character: 'kjshfksjh', integer: 10000, real: 1.0000E6)
 30 - 79      '/' + comment ('/ UT..')
```

Comment and history record:

```
 "COMMENT" or "HISTORY" followed by any text.
```

Record length is 80 characters. Do not use ASCII characters above 127. For more detailed info about FITS view FITSIO (http://fitsio.nasa.gov). See also -g option.

tcor=seconds

Correction of time in seconds. Integer, signed or unsigned number up to your system specific long int type.

mask=string

Output mask (default = `kout????.fts`) The '*' has the same meaning as '?'.

-p parname

Parameter file. (equivalent with `par=`)

-o mask

output mask (equivalent with `mask=`)

-c number

counter initial value (default = 1)

-h

create only header of FITS file. Not help!

-help

give help

-L

display software license

-g

wrote an example of the parameter file to the file `konve.par`. You can use to generate correct file, edit it and use with `par=` option.

Examples

Example 7-1.

Give help

```
picko
```

Example 7-2.

Convert file `image.pic` to `kout0001.fts`

```
picko image.pic
```

Example 7-3.

Convert file `image.pic` to `image.fts`

```
picko image.pic mask=image.fts
```

Example 7-4.

Convert all PIC images at current directory to default mask

```
ls *.pic | picko @
```

Example 7-5.

Convert R images of beta Lyr

```
ls betalyrR???.pic | picko @ mask=betalyrR???.fits
```

Example 7-6.

Convert only headers of R images of beta Lyr

```
ls betalyrR???.pic | picko -h @ mask=betalyrR???.fits
```

Warning

You must uses this utility a very *carefully.* If output mask (option -o or mask=?) is equivalent with input image, konve overwrite the input image and your *data* are *losted*. An example is the command **picko i.pic -o i.pic** (your are overtyped suffix `.st8` and `fits`). Try allways set the permissions to read-only for input files before.

Don’t insert blanks between key (`par=`, `tcor=`,..) and argument.

Diagnostic

The error messages like "Can’t convert..","Can’t open.." usually means, that you typed wrong filename(s).

"No input files." - missed ’@’ at command line.

"Can’t create..", "Can’t write.." - disk full?

"Unknown image type." - wrong filename (suffix)?

Authors

F. Hroch, Masaryk University, Brno (mail:hroch@physics.muni.cz).

Chapter 8. Autoflat

The autoflat utility makes the synthetic image of the uniform lighted plane. The flat-filed corrects the variable photometric sensitivity of the image due to the pixel to pixel sensitivity, the optics non-uniformity and others.

Many of greatest observatories uses the real uniformly brightet surface, but the image of the twilighted sky is more usuall way to getting its. The probably best way to determine of the flat-field is continuous grabbing of the zenit of the sky during the twilight. The intensities spread the interval from 20 000 to 50 000 ADU of the full 16-bit camera range, for the example. The telescope mount is swith off during this taking of the image and the anonymous stars draws the line on CCD chips. The autoflat utility remove this paths by the using robust procedure to determine of the arithmetical mean.

Another possibility is the use of the large serie of the regular images with different stars fields.

Syntax

The autoflat reads names of the FITS files form standard input represented by redirected file or the input from the terminal and create the single file autoflat.fits. This name can by optionaly changed from the parameter file. The input from the standard input is a little confusing. Please read the next exmaples carefully.

The autoflat is invoked with command:

aflat

Parameter file. The autoflat optionally reads the `autoflat.cfg` file for setting of the non-default values of the parameters. The commnets starts with ! character. The distributions parameter file contains default values of the parameters.

parameters

NAMEFLAT

the name of the output file

SATURACE

the saturation level in ADU

LEVEL

the output mean level in ADU

MINRANGE

the minimal input level in ADU

MAXRANGE

the maximal input level in ADU

NSTEP

the step over image to determine some levels, from 1 (the slowest computing but precise) to the number of pixels of the image (the fastest case but non-useful).

Examples

Example 8-1. without input

Without parameters is running a little confusing. Please use Control-D (unix) or Control-Z (Dos) to ending of standard input. You can also type names of the input files manually and end the input the described way.

```
debian:~/munipack-0.2.0/autoflat$ aflat
 Autoflat v. 0.0.4 - F90
STOP No input image. statement executed
```

Example 8-2. typical run

The typical run uses pipe or file on the input. Like this:

```
debian:~/obr$ ls rv?.fits | aflat
 Autoflat v. 0.0.4 - F90
 rv1.fits:
   No., Median, skysig, n: 1  559.880249  20.0045414 10880
 rv2.fits:
   No., Median, skysig, n: 2  566.333252  20.4462452 10880
  Number of the input images:  2
 mainsig,medmed,medsig=  361.028503  566.333252  20.4462452
 1 :   18.047327  17.8609619 (level, sig)
 2 :   17.6574478  17.6574478 (level, sig)
 An overflow has been occured during flat-fielding.
      Final median, skysig, n=  10001.1865  293.116913 10880
STOP Bye. statement executed
```

Chapter 9. Darkbat

The darkbat utility subtract dark image from a serie of the scientific exposures. If, the input image is D(i,j) and the scientific image is I(i,j) then the output image is difference

Syntax

Darkbat reads names of the image FITS files form standard input, file or command line and subtract the specified dark image.

Darkbat is invoked with command:

darkbat [options] [@dirfile] [dark=darkframe] [mask=?] [image1] ...

options

@dirfile

file with name(s) for dark substract, no wildcards *,? You can use **ls** command to make it: **ls images*.fits > dirfile**

@

read image(s) name(s) from standart input

dark=, **-d**

Dark frame image name. Both forms are equivalent.

image1 .. imageN

name(s) of simple image(s) to dark subtract, no wildcards *,?

mask=string, **-o**

Output mask (default = `dout????.fts`) The '*' has the same meaning as '?'. The number of '?' characters in the output mask not must exceed 9, eg. no more then 999 999 999 images is allowed.

-c

counter initial value

-h

give help

-L

display software license

Examples

Example 9-1.

Give help

```
darkbat
```

Example 9-2.

Subtract dark image `dark.fits` from image `image.fits`. Create output image `dark0001.fits`. Replace image, dangerous, but save your disk space.

```
darkbat dark=dark,fits image.fits
```

Example 9-3.

Subtract dark image `dark.fits` from images `image01.fits`... `image99.fits` . Replace input images, dangerous, but save your disk space.

```
ls image??.fits | darkbat @ dark=dark,fits mask=.
```

Chapter 10. Flatbat

The flatbat utility divide the scientific image by the image of the homologously lighted surface. If, the flat image is F(i,j) and the scientific image is I(i,j) then the output image is difference

where the function mean(F) is the robust estimation of the mean of the flat-field image.

Syntax

Flatbat reads names of the image FITS files form standard input, file or command line and divide its by the specified flat-field image.

Flatbat is invoked with command:

flatbat [options] [@dirfile] [flat=flatframe] [mask=?] [image1] ...

options

@dirfile

file with name(s) for flat correction, no wildcards *,? You can use **ls** command to make it: **ls images*.fits > dirfile**

@

read image(s) name(s) from standart input

flat=, **-f**

Flat-field frame image name. Both forms are equivalent.

image1 .. imageN

name(s) of simple image(s) to the flat-field correction, no wildcards *,?

mask=string, **-o**

Output mask (default = `fout????.fts`) The '*' has the same meaning as '?'. The number of '?' characters in the output mask not must exceed 9, eg. no more then 999 999 999 images is allowed.

-c

counter initial value

-h

give help

-L

display software license

Examples

Example 10-1.

Give help

```
flatbat
```

Example 10-2.

Divide by the flat-field image `flat.fits` the image `image.fits`. Create output image `fout0001.fits`. Replace image, dangerous, but save your disk space.

```
flatbat flat=flat.fits image.fits
```

Example 10-3.

Divide by the image `flat.fits` teh images `image01.fits`... `image99.fits` . Replace input images, dangerous, but save your disk space.

```
ls image??.fits | flatbat @ flat=flat,fits mask=.
```

Chapter 11. Meandark

The autoflat utility simply compute the arithmetical mean of the input images. The utility do not use any robust algorithm, the far values will be great affect to the mean value. The primary goal is computing of the mean of dark images. 3..5..7 is optimal number of images.

The meandark make the same work like the autoflat, but uses non-robust algorithm.

Syntax

The meandark reads names of the FITS files form standard input represented by redirected file or the input from the terminal and create the single file meandark.fits. The input from the standard input is a little confusing. Please read the next exmaples carefully.

The meandark is invoked with command:

mdark

Examples

Example 11-1. without input

Without parameters is running a little confusing. Please use Control-D (unix) or Control-Z (Dos) to ending of standard input. You can also type names of the input files manually and end the input the described way.

```
debian:~/munipack-0.2.0/autoflat$ aflat
 Autoflat v. 0.0.4 - F90
STOP No input image. statement executed
```

Example 11-2. typical run

The typical run uses pipe or file on the input. Like this:

```
debian:~/obr$ ls rv?.fits | aflat
 Autoflat v. 0.0.4 - F90
 rv1.fits:
   No., Median, skysig, n: 1  559.880249  20.0045414 10880
 rv2.fits:
   No., Median, skysig, n: 2  566.333252  20.4462452 10880
```

```
  Number of the input images:  2
 mainsig,medmed,medsig=  361.028503  566.333252  20.4462452
 1 :   18.047327  17.8609619 (level, sig)
 2 :   17.6574478  17.6574478 (level, sig)
 An overflow has been occured during flat-fielding.
      Final median, skysig, n=  10001.1865  293.116913 10880
STOP Bye. statement executed
```