

# Turbina-core(D) program

## Versions 2.3 – 2.25

V. Kornilov

March 23, 2008

## 1 Installation

### Requirements to the computer and OS

This program works under the Linux OS system with a kernel newer than 2.6.8. This is needed to provide a smooth work with IEEE1394-based digital cameras such as Prosilica EC650, for example. Although the program can be adapted to practically any IEEE1394 camera, the current version uses such IEEE1394 camera possibilities as the support of **Format 7** and of **ROI** (region of interest). For the right choice of the cameras, the information accessible at <http://damien.douxchamps.net/ieee1394/cameras/index.php> can be used.

However, our studies show that **Prosilica EC650** has very good performance to use as DIMM image detector. Camera has read-out noise about  $10e^-$  only and can provide frame rate about 200 fps with ROI  $\sim 100 \times 50$  and exposure 4 ms.

The computer must provide a sufficient performance to process video stream from the camera. We tested the 2GHz Celeron and 3GHz Pentium-IV machines and found that the average CPU loads were 20-25% and 10%, respectively. Note, that for our MASS/DIMM (installed near Kislovodsk), we use VIA EPIA computer with VIA C3 800MHz processor. The machine provides effective work DIMM and MASS software and few additional services with average CPU loads less 20%.

### Drivers and library installation

To work with cameras via IEEE1394 bus, the drivers included in the standard OS distribution are needed. Usually, the low-level drivers `ieee1394` and `ohci1394` are loaded by the OS at boot. The drivers `raw1394` and `video1394` can be loaded by the hot-plug service or manually before running the program, as well as at the system boot.

Two devices used by program are: `/dev/raw1394` for camera control and `/dev/video1394` for video stream input. If the devices are absent in system, issue the following commands manually under root:

```
mknod -m 666 /dev/video1394/0 c 171 16
mknod -m 666 /dev/raw1394 c 171 0
```

Considering that the program uses the latest but still unstable release of `raw1394` and `dc1394` libraries, these libraries are placed in a non-standard way to avoid any confusion with another software.

## Directory structure

For the program compilation, the source files can be placed in the any directory. Source files include the directory `iidc` containing camera interfaces and the directory `extra` containing the compiled libraries `libdc1394.a` and `libraw1394.a` and the sources for their compilation. Latest versions ( from 2.09 ) contain the directories `infras` and `server`, to where shared with other our projects modules were moved.

During the execution, the program uses absolute paths for its input and output files. We plan early that the `Turbina-core(D)` program will be located in the `/opt/turbina/bin` directory. At the present day, the DIMM software is placed in separate directory `/opt/dimm/`. It is recommended to include the path to this directory in the user's `.profile` or `.bashrc` files. The sub-directories `/opt/dimm/data/out` for the output data file, `/opt/dimm/data/log` for the output log file `yymmdd-dimm.log` and `/opt/dimm/etc` for the configuration file must be created. The output data are written in the file named `yymmdd-dimm.stm` and located in the sub-directory `data/out/`. The structure of the output file is described below.

Additionally, the directory `/opt/dimm/data/` contains the `data/`, `images/` and `spectra/` subdirectories. The `images/` serves as a place for output images, generated by the program. The `data/` and `spectra/` directories contain the files needed for `preat` routine.

## Running the program

Starting from the version 2.4 of the program, the executable file is named `dimm` (`pro_test` in earlier versions). The name `Turbina` is retained for the whole software project enabling the on-line work with MASS/DIMM instrument and off-line processing of its data.

Program allows few options in command line. They are listed below

```
-h          — short help
-c cfg      — another configuration file cfg than default turdimm.cfg,
-b          — put program into background after start (demonize it),
-a          — automatic initialization on start,
-d          — debug mode on,
-i name     — Internet machine name differing from default "0:0:0:0",
-p numb     — port number differing from default 16200
```

The debug mode permits to run the program even when the camera is disconnected. In this case, artificial frames will be generated with known parameters of the images. For example:

```
./dimm -c new.cfg -d
```

There is the possibility to redirect program standard output in the pipe for processing program such as `preat`:

```
./dimm -d | ./preat
```

To suppress the wide program output, the redirection is allowed:

```
./dimm -d > /dev/null
```

In the real usage as part of **ASM** (Automatic seeing monitor) system, the program must be run in the background mode:

```
./dimm -b
```

In this case standard and error outputs redirected automatically in `/dev/null` and program acts `daemon` role.

At the start of the program, the camera(s) is detected automatically and its identification number (UID) is displayed. The right number must be entered in the configuration file, otherwise an error message will be issued.

Remember that the program is only controlled via socket connection because it is designed for a remote control either by `supervisor` or another program. To control the program manually, the simplest way is to connect to it via the standard `telnet` utility. For example, connection from the same machine looks as:

```
telnet localhost 16200
```

where the 16200 is the program server port number. Then, the program control is performed by sending a Supervisor-format command, including 1) command ID number, 2) command itself, 3) commands keywords with or without value (see Sect. Control of the program through socket). To terminate the program execution the command `quit` must be used.

## 2 Initial conventions

### Coordinate system

It is supposed that the digital camera is installed on the MASS/DIMM device in correct orientation. The horizontal lines (rows) must be parallel to the line connecting the DIMM apertures, i.e. to the flange of the device. Then the longitudinal differential motion corresponds to the horizontal (along CCD rows) direction and the transversal differential motion corresponds to the vertical (columns) direction.

In the program the horizontal and vertical coordinates are designed as `x-` and `y-` coordinates. Pixel coordinates such as the optical center position or common images center are always written in the order first `x`, then `y`. To acquire more frames during `basetime`, the vertical size of the working subframe must be minimized. It is possible if two images are placed along horizontal line by correct alignment of the DIMM mirrors in the device.

In the program, an image with a smaller `x`-coordinate is always called “left”, the other image is “right”. Whether this corresponds or not to the image orientation on a computer screen, depends on the image viewing software.

In the output data, the coordinate with fractional part 0.5 corresponds to the center of a pixel. So, the coordinates 0,0 are the coordinates of the top left corner of the top left pixel on the CCD frame.

### Measurement timing

There are several time intervals in the program: `exposure` — one frame exposure duration in ms, `frame rate` — number of frames taken during 1 sec, `basetime` — a time interval for elementary calculation of the differential motion variance, `accumulation time` — the duration of one complete seeing measurement.

### 3 Measurement algorithms

The Turbina-core(D) program is constructed as a server-type program. If no activity, it waits for a request from client-type programs such a Supervisor. After such a request is received and checked, the program executes the requested function and waits for the next request. Any possible errors don't terminate the program's work, only the current function terminates.

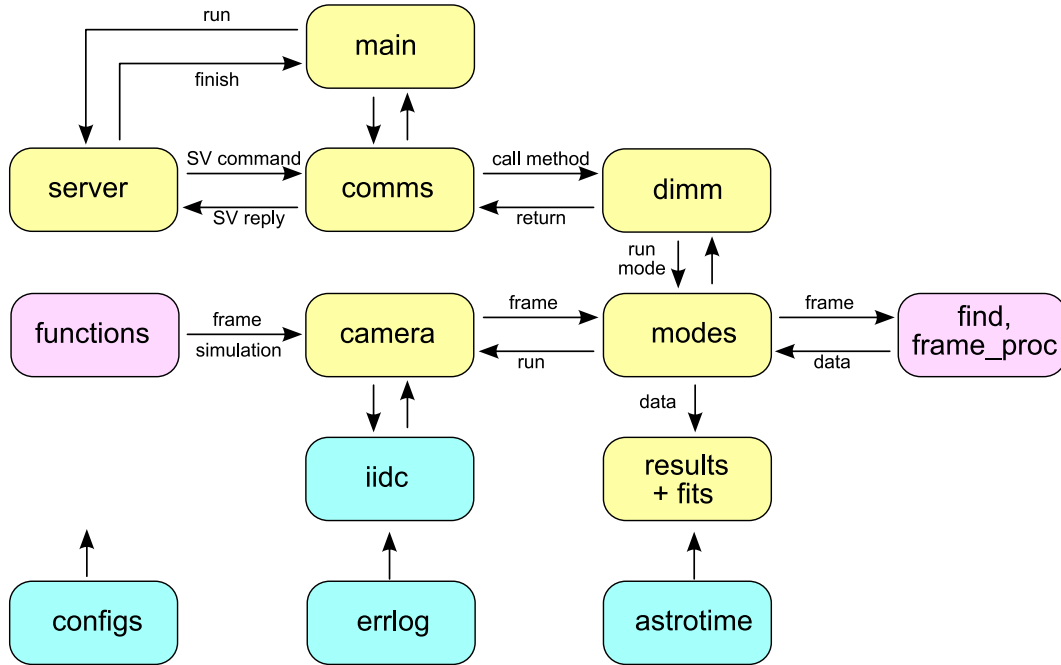


Figure 1: The structure of the `dimm` program. Yellow modules form logical chain, pink modules are different calculating procedures, cyan ones — basic infrastructure.

When the command **INIT** comes, all program's objects needed for the measurements are created, the connection with the camera is established and the camera is initiated. The input configuration file is opened and read. The output files are either opened or created, depending on the current time. This permits to run the Turbina-core(D) continuously for many days and to generate separate output files for each night.

The command **PARK** induces the following sequence of actions: all files are closed, the camera becomes free and the program's objects are destroyed. In the parked state, the program can only respond to the commands **INIT**, **GET STATUS**, **GET IDENT** and **GET ERROR**.

After the initialization, few measurement modes are supported: Test mode, Centering mode and Normal mode. Beginning from the 2.05 version, Raw mode was added. Additionally, latest version supports Pictures mode and Estimation mode.

#### Test mode

Test mode (Field Test) works with full-frame CCD images. Its functions are

1. to estimate the field aperture center position with respect to absolute CCD coordinates,

2. to estimate the field aperture radius  $R_A$  needed for sub-frame selection in the centering mode,
3. to calculate roughly the separation  $Sep$  between spots provided by the DIMM mirrors, using the double image of the field aperture,
4. to calculate main parameters of the CCD: the conversion factor (in electrons/ADU) and the readout noise (in electrons).

Before running the mode, the telescope must be pointed on a daytime sky or to a diffusely illuminated object. If the illumination is insufficient, the error “Not enough light” (code 621) will appear and the measurement will break. Otherwise, the program sets the exposure time optimal for the actual light level and does the measurement.

Some warnings may be issued to draw attention to the bad alignment of DIMM mirrors or to inappropriate parameters in the configuration file.

The configuration file must be edited to enter the determined field aperture parameters, so that no more tests will be needed at each program start.

The test mode writes the CCD image to the file “fieldtest.fits” (over-writes the same file) to document or examine the view of the field aperture (see Fig. 2).

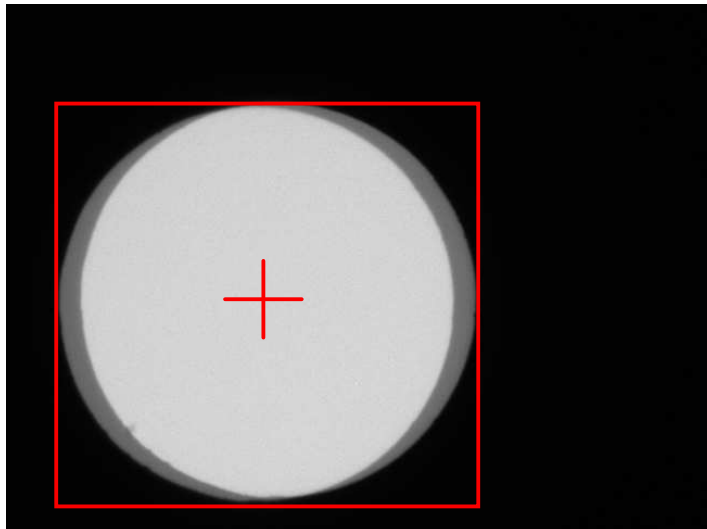


Figure 2: The view of the field aperture on the frame obtained during Field Test. The red box indicates the frame defined for Centering mode, the red cross marks the optical axis position

### Centering mode

The centering mode uses the position and size of the field aperture specified in the configuration file. In this mode, the subframe fully covers the double image of the aperture and has dimensions  $2R_A \times (2R_A + Sep)$ , i.e. approximately half of the full CCD frame. This mode is intended for the following tasks:

1. to check the telescope pointing (if the target star is in the aperture or not),
2. to determine the position of the common center of star images relative to the optical center of the aperture,

3. to determine the exact separation between the images.

This data can be used to correct telescope pointing and must be used by the main (normal) mode to define the ROI (stars box) position.

Usually, the centering mode uses the same camera settings (gain and exposure) as the normal mode, but different settings via configuration file are supported. The accumulation time in this mode is about few seconds only. Using a frame rate of about 60 — 70 fps, a good averaging of the star’s position is achieved.

The first stage of the algorithm is the calculation of the mean bias and its variance. Although these values can be obtained from one frame, the values averaged over several frames are used. The threshold for object detection is regulated by the parameter `Operations/Centering/ThresholdFactor`. The two brightest detected objects are considered as images of the target star if their total signal is greater than `Operations/Centering/MinObjectFlux`.

During the execution, the error “No two star images” (code 620) may appear. The error breaks the execution and returns `ERROR=ERRFAT`. Few warning messages describe the mismatches between the measured and desired images configuration.

The centering mode (over-)writes the CCD image to the file “centerframe.fits” (see Fig. 3).

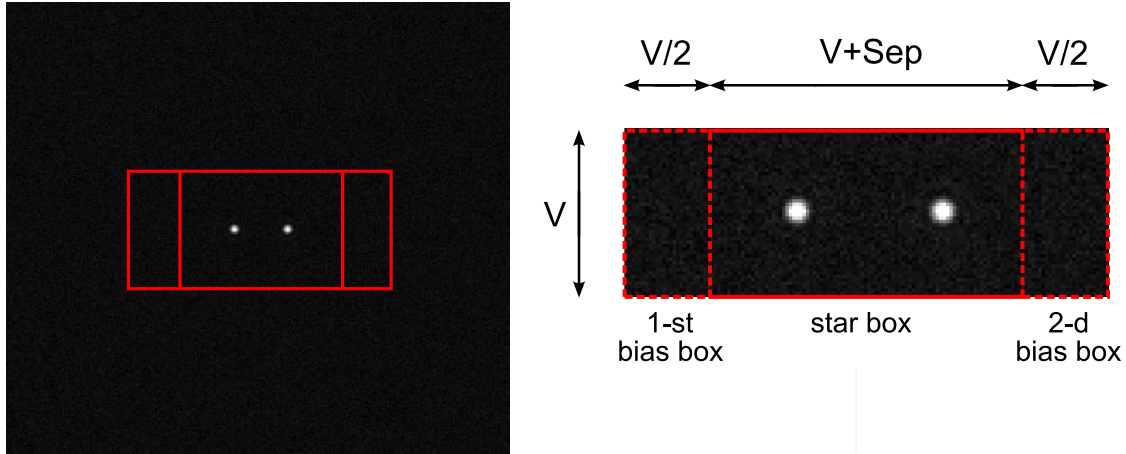


Figure 3: The view of the star images on the frame obtained during Centering Mode. The red box corresponds to the CCD frame used in the Normal mode, the central part of the box is used for the star images measurements, the left and right parts are intended for bias control

### Main mode (Normal mode)

The normal mode is the main mode of the measurements in the DIMM channel of the MASS/DIMM device. The result of this mode is a seeing value (after additional calculations). The mode can be launched only after successful completion of the centering mode, as far as it uses the results of the centering mode. It is possible to run this mode again and again, because the information about current images position is restored each `basetime` period.

A small fraction of the CCD frame is read-out in this mode. The parameter `Operations/Normal/MeasBoxSide` defines the vertical size  $V$  of this ROI (star box). The horizontal size  $H$  is equal to the vertical size plus the image separation:  $H = V + Sep$ . Two bias boxes ( $V/2 \times V$ ) are appended to the left and right sides of the star box.

The detection of images in star box is done in the same way as for the centering mode. The two brightest spots are considered as the star images. The x- and y-coordinates of each image are used for precise determination of the gravity center. One of the two centering methods can be used: thresholding or windowing. In addition to the coordinates, many characteristics of the images are calculated: total flux, maximal flux, image radius and ellipticity. Also, the contributions of the readout and photon noise to coordinate errors are estimated.

The normal mode checks the position of the star box. If it is located far from the optical center, the execution breaks with the error “Stars are far from center” (code 623). The algorithm assumes that a number of frames may be empty for various reasons, but if this number is greater than `Operations/Normal/MaxDropped` for one `basetime` cycle, the error “No two star images” (code 622) is generated. The error “Empty basetime set” informs that all images acquired during the `basetime` were rejected by the verification process.

The normal mode (over-)writes the CCD image to the file “boxframe.fits”, once per `basetime`.

## Raw data mode

To have a possibility of some expanded analysis, the mode **Raw data** was implemented in the program. This mode works in the same way as Normal mode, but the results of the processing of the frame (the position of the left and the right image spots and fluxes) are stored in the output file.

During the accumulation time, the star box position is not corrected and the calculations after each `basetime` are suppressed in order to provide temporal data sampling as regular as possible. Unfortunately, it is not possible to get the exact time when each frame was really exposed; instead, the time of the frame extraction from the kernel memory is recorded in the output file.

In the raw data mode, the configuration parameters for the normal mode were used before Version 2.10. Now, the configuration file contains the respective subsection. The mode is useful for testing purposes and for some research tasks.

## Estimation mode

To have a possibility of quick estimation of the current seeing data, the **Estimation mode** was added in the program. This mode works just the same way as Normal mode, but finishes after one `basetime`. The mode uses parameters from `Operations/Normal/` section of the configuration file.

This mode provides more correct measurements than Centering mode because develops more image frames within the same time. Also, this mode (just like Normal mode) calculates the image fluxes with respect to background level rather than to threshold level.

Since the Estimation mode uses image data within star box window, its position on the CCD frame must be known before.

## Pictures mode

The **Pictures mode** provides grabbing of the star box images and storing their sequence as one output file “boxrecord.fits” in `/opt/dimm/data/images/` subdirectory. The mode uses own subsection in configuration file to set the record length, frame exposure and frame rate. Output

file in FITS format has 2 dimensions only:  $x$  the same as the side of star box and  $y$  is equals of the side of star box multiplied by numbers of the frames.

## 4 Configuration file

The parameters needed for correct work are given in the file `turdimm.cfg`. It is organized into sections and sub-sections. The meaning of each section and parameter are obvious from the example of the configuration file provided below.

```

Section "General"
  SubSection "Site"
    DeviceNumber          MD09 ;device serial number
    SiteName              KGO ;observatory or site na
    Longitude             2 50 40 ;longitude of the site: hh mm ss
    Latitude              43 44 12 ;latitude of the site: dd mm ss
  EndSubSection
  SubSection "DIMM"
    ApertureBase          20 ;aperture separation, cm
    ApertureSize          9.3 ;aperture diameter, cm
  EndSubSection
EndSection
Section "Operations"
  SubSection "Normal"
    Exposure              4.0 ;frame exposure in ms (>=1)
    FrameRate             150 ;working framerate, Hz
    BaseTime              2.0 ;base time in sec.
    AccumTime             60 ;accumulation time in sec.
    MeasBoxSide           60 ;measuring subframe side, pxs
    StarRadius            7 ;image radius for CG calculation, px
    ThresholdFactor       3 ;number of sigma's
    CGMethod              threshold ;or window - CG computation method
    MaxDropped            10 ;maximal number of bad frames
  EndSubSection
  SubSection "Centering"
    Exposure              4.0 ;frame exposure in ms (>=1)
    FrameRate             30 ;working framerate
    AccumTime             2.0 ;full time of the centering phase, s
    ThresholdFactor       4 ;number of sigma's
    MinObjectFlux         300 ;min total flux for target images, ADU
  EndSubSection
  SubSection "FieldTest"
    Exposure              100.0 ;frame exposure in ms (>=1)
    FrameRate             0 ;working framerate, 0 - freerun mode
    AccumTime             25 ;full time of the centering phase, s
    ThresholdFactor       0.1 ;fraction of maximal signal
  EndSubSection

```



```

SubSection "RawData"
    Exposure          4.0 ;frame exposure in ms (>=1)
    FrameRate         0 ;working framerate, 0 - freerun mode
    AccumTime         10 ;full time of the mode
EndSubSection
SubSection "Pictures"
    Exposure          4.0 ;frame exposure in ms (>=1)
    FrameRate         0 ;working framerate, 0 - freerun mode
    AccumTime         10 ;full time of the record
EndSubSection
EndSection
Section "Camera"
    SubSection "Geometry"
        FieldAperture 184 ;radius of the field diaphragm, px
        Scale          0.634 ;image scale, arcsec/px
        FocalCoeff     200 ;connection to focal changeing
        Separation      40 ;preferable images separation
        OpticalCenter  320 240 ;optical center position on the frame
    EndSubSection
    SubSection "Type"
        Model          "Digital Camera (DCAM 1.31)" ;camera model
        Identification  0x00f31000000755c ;camera serial number (UID)
        Digitization    12 ;camera digitization, bit/px - not used
    EndSubSection
    SubSection "Parameters"
        Format          658 492 ;camera format, px
        Gain            0.0 ;working gain, dB
        ReadOutNoise    11.5 ;RON in electrons
        BiasControl      295 ;Relative number; not used in Prosilica
    EndSubSection
EndSection

```

Some of the configuration parameters are not used and probably will be removed in the future. Note, that unused in the current program version configuration parameters doesn't lead to any problems. In contrary, absence of needed parameter string may induce early termination of the program with corresponding error message.

## 5 Structure of the output file

The output file `yymmdd-dimm.stm` is an ASCII file. Each line begins with a prefix indicating the type of its contents. In the examples given below long lines are artificially split by the symbol `>>>`.

**Parameter lines** begin with the prefix **P** and contain the copy of the configuration parameters. They are written at the start of the program. Each time an INIT command is issued, the configuration file is re-read. However, only the parameters that are changed are written again in the output file. The parameter lines look like

P 2006-11-19 02:04:26 General/DIMM/ApertureBase = 20

### Centering mode output

Each of the program modes is reported in the output file with the M-line (prefix **M**), possibly accompanied by additional data strings. The mode **Center** provides only an M-line. The example of such line:

```
M 2006-11-18 01:42:47 Centering: X=9.7 Y=6.7 dX=42.2 dY=-4.5 FLUX_L=46761 >>>
FLUX_R=49528 BS=15.3 RMS=5.5
```

The fields of an M-line contain:

1. Prefix M
2. Date YYYY-MM-DD
3. UT time hh:mm:ss
4. Mode name "Centering:"
5. Offset X (in pixels) of the image pair common center with respect to the optical center of the device as defined in the configuration file field **Camera/Geometry/OpticalCenter**
6. Offset in Y
7. Horizontal separation dX, px
8. Vertical separation dY, px
9. FLUX\_L – the integral flux in the spot 1 (left), ADU
10. FLUX\_R – the integral flux in the spot, ADU
11. BS — bias (background) level, ADU
12. RMS — rms of the background noise, ADU

### Normal mode output

The start of this mode is marked by the M-line containing date/time stamp and the name of the mode. The data lines follow then. New data lines are output after each **basetime** and contain the raw results of the seeing measurement. They begin by the prefix **d** and look like

```
d 2006-11-19 02:33:59 294 18068 18568 0.232 0.240 1932 2001 42.89 -2.23 >>>
1.286 0.840 1.501 0.594 0.020 0.019 -17.4 16.3 0.62 0.58 3.33 >>>
-0.02 3.34 0.08 15.40 5.47
```

The data are mostly given for each spot separately. Spot 1 is the leftmost on each CCD line. The contents of each d-line is:

1. Prefix d
2. Date YYYY-MM-DD
3. UT time hh:mm:ss of the measurement
4. Number of frames averaged
5. Flux in the spot 1, ADU
6. Flux in the spot 2, ADU

7. RMS flux fluctuation in the spot 1 divided by the flux
8. RMS flux fluctuation in the spot 2 divided by the flux
9. Maximum intensity of the spot 1, ADU
10. Maximum intensity of the spot 2, ADU
11. Average spot separation in x, px
12. Average spot separation in y, px
13. RMS of the spot separation in x, px
14. RMS of the spot separation in y, px
15. Covariance in x with lag 1, px<sup>2</sup>
16. Covariance in y with lag 1, px<sup>2</sup>
17. Estimated rms noise in x, px
18. Estimated rms noise in y, px
19. Average position of the two-spot center in x, px
20. Average position of the two-spot center in y, px
21. RMS of the average x-position, px
22. RMS of the average y-position, px
23. Average FWHM of spot 1, px
24. Average ellipticity of spot 1
25. Average FWHM of spot 2, px
26. Average ellipticity of spot 2
27. Average background, ADU
28. RMS background fluctuation, ADU

At the end of each accumulation time, a D-line is written. It contains the same time stamp as the last d-line, the number of individual measurements (d-lines), average positions in x and y and the rms dispersions in x in y (all in pixels). These rms values take into account the low-frequency component of the differential image motion (time scales longer than *basetime*) and is better suited for the calculation of seeing. The D-line looks exactly like d-line:

```
D 2006-11-19 02:33:59 27 18068 18568 0.232 0.240 1932 2001 42.89 -2.23 >>>
1.286 0.840 1.501 0.594 0.020 0.019 -17.4 16.3 0.62 0.58 3.33 >>>
-0.02 3.34 0.08 15.40 5.47
```

### Test mode output

The mode **FieldTest** generates one M-line and several L-lines. The M-line contains the date/time stamp, the name of the mode and, in the following data fields, X0 and Y0 — the center of the field aperture in pixels with respect to full CCD detector plane; dX and dY – the estimates of the horizontal and vertical image separations; R1 and R2 – the estimates of the aperture radius in pixels obtained by two different methods; BS – the flux level outside the aperture; FLUX\_A – the mean flux inside the aperture. The example of this line is:

```
M 2006-11-18 01:42:47 FieldTest: X0=323 Y0=237 dX=36 dY=-1 >>>
R1=179 R2=181 BS=20 FLUX_A=1688
```

The L-lines represent the measurements of the variance of the difference  $D$  of two consecutive frames with the same exposure. After the date/time stamp, the exposure in ms follows, then the mean flux inside the aperture in ADU, the variance  $\sigma_D^2$  (not rms) of the difference in ADU<sup>2</sup>, and the flatness of the aperture illumination. The flatness characterizes the relative systematic variation of the brightness in the aperture, it is calculated as  $(\sigma_S^2 - \sigma_D^2)^{1/2}/S$ , where  $S$  — the sum of two these frames and  $\sigma_S^2$  is the variance of this sum.

Finally, the conversion factor ( $GAIN$  in  $e^-/ADU$ ) and readout noise ( $RON$  in  $e^-$ ) are calculated and written as a separate M-line for NoiseTest:

```
M 2007-01-12 22:31:21 NoiseTest: GAIN=2.01 RON=11.13
```

## Raw data mode output

As usual, the M-line declares the mode start:

```
M 2007-01-07 21:00:43 RawData
```

Then, the block of the data lines follows. Each data line begins with the prefix **r** and contains high-precision time mark, the frame sequential number, the x- and y-coordinates for the left and right star images and the total fluxes for both images.

```
r 75643.7503 0 80.521 20.485 40.114 20.500 9481 9498
```

The raw data output is finished by one d-string which summarizes the measurements.

## 6 Control of the program through socket

The program is controlled by commands issued through the socket connection (port 16200). The first group of symbols in each command is normally its number. However, it is not interpreted and can be any symbol(s) except space. The commands are not case-sensitive.

The accepted commands are:

1. **INIT** – initialize, read configuration
2. **RUN** or **RUN NORMAL** – start measuring seeing for one *accumtime*.
3. **RUN CENTER** – start measuring the spot centers, full frame.
4. **RUN TEST** – start measuring the illuminated aperture, full frame.
5. **RUN SCENARIO** [=“...”] – run scenario of measurements. The actual scenario can be either set directly in this command or defined earlier by the **SET SCENARIO** command.
6. **RUN RAW** – start measuring seeing for one *accumtime* with extended output.
7. **RUN ESTIMATION** – start measuring seeing for one *basetime* only.
8. **RUN PICTURES** – start recording of the frames during one *accumtime*.
9. **GET STATUS** can have the following answers: **OK STATUS=READY**, **OK STATUS=BUSY** or **OK STATUS=PARKED**
10. **GET IDENT** returns **IDENT=“Turbina-core(D) Vers.nnnn”**
11. **GER ERROR** returns the description of the last error
12. **GET OFFSET** returns offset of the two-spot center relative to the aperture center [px]

13. **GET SEPARATION** returns the spot separation in x,y [px]
14. **GET FLUX** returns the total and maximal fluxes in both spots [ADU]
15. **GET DATA** returns the full data string of current data (like d-type in the output file)
16. **GET MODE** *returns the current mode or last finished mode*
17. **SET SCENARIO="..."** sets the current scenario
18. **SET OBJECT="..."** – sets object ID and writes it to the output file
19. **STOP** – stop measurements after finishing the current mode
20. **STOP NOW** – stop immediately
21. **PARK** – disconnect the camera, stop all actions, wait for **INIT**
22. **QUIT** – park and exit

The program answers are similar to described early in Turbina 2.04 User Guide:

1. **OK WAIT=tt** – correct command is received, needed execution time is tt sec
2. **OK STATUS=READY** – command execution is successfully finished
3. **OK STATUS=BUSY** – command execution isn't possible due to current work
4. **OK STATUS=PARKED** – PARK command is successfully done
5. **ERROR STATUS=PARKED** – the command cannot be executed because DIMM is parked
6. **ERROR STATUS=ERSYN** – the command has wrong syntax
7. **ERROR STATUS=ERFAT** – an error has occurred during execution

The scenario syntax is the same as in the Turbina vers. 2.04. To denote Center mode, Normal mode, and Field test the symbols **c**, **n**, **t** are used. The scenario is case insensitive, as well as the commands. For example, to run Center mode once and Normal mode three times, the scenario "**c+3\*n**" can be used.

## 7 Changes in the version 2.4

1. All the coordinates output data are in pixels only. Now the configuration parameter **scale** is used for further processing. An external request **GET OFFSET** returns the star distance from the optical axis in arcsec.
2. The problem with the free-run camera mode (when the camera defines itself the maximal frame rate for a given exposure) is fixed. To initiate this mode, set the configuration parameter **framerate** equal to 0.
3. The relative dimensions of the Normal mode subframe are changed to provide more freedom for star images motion in the horizontal direction.
4. The program was renamed to **dimmm**.
5. Two options **-i** and **-p** are added for running the program in the command-line mode. The first key must be used with PC Internet name, the second — with the program's server port number. The default values are: "0.0.0.0" for the machine name and 16200 for the port number.
6. In the versions before 2.4, the star image diameter was output, instead of radius. This is fixed. Also, the definition of the image elongation is changed.

7. The calculation of the conversion factor (*Gain* in  $e^-/ADU$ ) and the readout noise (*RON* in  $e^-$ ) was added in Test mode.
8. The D-line format is extended to the d-line format.

## 8 Changes in the version 2.5

1. To provide more unified MASS and DIMM data structure the extended file names are introduced. The **dimmm** log file name is `yymmdd-dimmm.log` instead of an impersonal `yymmdd.log`, the output file name is `yymmdd-dimmm.stm`. The name `yymmdd.dimmm` is reserved for processed data file.
2. New command **RUN RAW** is added. In this mode, the individual data strings are output in the file during accumulation time. This mode is useful for testing purposes and for some scientific tasks.

## 9 Changes in the version 2.6

1. Two options `-b` and `-a` are added for command line usage. The first key forces program execution as background job. This option is useful when program must be started without terminal connection ( by `cron`, for example). The second key is used when automatic initialization is desired.

## 10 Changes in the version 2.9

1. New command **RUN PICTURES** is added.
2. New application-specific command **GET FOCUS** is added for automatic focus control in the case of telescope RCX400.

## 11 Changes in the version 2.20

1. The error in the output of measured radiuses is corrected. Correct radius is greater by factor  $\sqrt{2}$ . The values are not used in the program algorithm, so no recalculations are needed.
2. The maximal fluxes in left and right images are added in return of the command **GET FLUX**.

## 12 Changes in the version 2.25

1. The algorithm of the star images detection in the frame was modified to provide faster and more robust work. The algorithm have fixed image fragmentation which arisen during some seeing condition.
2. Integral and maximal fluxes calculation in the thresholding mode was changed to calculation above background not threshold. It provide Schtrel number undependent from target star brightness. Also, these fluxes may be used for photometric analysis without any correction.