

# *bm2img* : image preprocessing on BM2 beamline.

J.-F. Bézar, D2am-CRG/F, ESRF

The D2AM/CRG@ESRF beamline acquired numerous data using CCD cameras. This detectors allow an efficient data collection but the raw data need to be processed before being some physical meaning data. In order to avoid a lot of black box processing, the current data preprocessing is done on the beamline using a small program, called *bm2img*. The aim of this document is to describe its specification or characteristics.

## Contents

<b>1</b>	<b>Introduction.</b>	<b>1</b>
<b>2</b>	<b>Elementary task examples.</b>	<b>2</b>
2.1	Image Information : " <i>bm2img fname</i> ". . . . .	2
2.2	Format conversion: " <i>bm2img fname %X</i> ". . . . .	2
2.3	Dark current correction : " <i>bm2img D=dname fname</i> ". . . . .	3
2.3.1	Multiframe images. . . . .	3
2.4	Flatfield correction : " <i>bm2img F=dname fname</i> ". . . . .	3
<b>3</b>	<b>Input file specifications.</b>	<b>3</b>

## 1 Introduction.

*bm2img* was first intended to work with the 16 bits image produced by Winview (Princeton, Roper Scientific) but it can use other data files. Internally, all data were stored as 32 bits signed integers, this choice has been done in order to reach a good dynamic without losing the counting accuracy. At the time (since July 2004, release 1.11), internal data are either 32 bits signed integers or 64 bits float according their use.

The *bm2img* program performs few elementary tasks :

- cosmic removable
- dark current substraction
- flatfield correction
- geometrical correction
- image addition, rescaling
- format conversion

Note that these tasks are not commutative and the resulting image may depend on the operation order. *bm2img* program can also be used for :

- radial means
- ROI extraction
- statistical studies of images

*bm2img* can be used as an interactive program but in its most frequent usage, *bm2img* works with an initialisation file which facilitates serial treatment and use arguments from the command line which specify the file to be read and the treatment to be done.

The command *bm2img -h* or *bm2img -usage* produces a short inline help. In the following document, examples will use most often the basic (or default) syntax but some modifiers allow to precise images, ROI in file. They are described in adhoc section together with the "ini" parameters.

The command *bm2img* start the program in an interactive session.

The whole command line for starting a batch session is :

*bm2img* [*D=dnames*] [*F=fnames*] *name* [*names*] [*%commands*] [[*names*] [*%commands*]]

To recognize commands from file names, all commands begin with a "%".

This document will focus on the usage of arguments from the command line but the same functions can be accessed in the interactive mode.

## 2 Elementary task examples.

### 2.1 Image Information : "*bm2img fname*".

When run without any command, the program simply read the image file then output informations.

At the time, *bm2img* can read the following format, which are recognized by their extensions, If no extension is given *bm2img* look by default for *.spe* or *.SPE* files.

extension	name	sub format range
.SPE .spe	Roper Scientific CCD	16 bits unsigned
.edf	EsrF Data Format	32 bits signed 16 bits unsigned
.gel	TIFF	16 bits unsigned
.txt	ascii	by lines (small images only)

The output for *bm2img test\_dir/fluo6.SPE* will look like :

```
$ bm2img test_dir/fluo6.SPE
## bm2img : extracting info/preprocessing image
## bm2img_1.7 (27Jun03_1433)
## berar@esrf.fr (d2am),

# file 'test_dir/fluo6.SPE' : images size 670 * 650, frames 5
# data_type (INT16U) no swapping used
# edf header contents :
#00# HeaderID      = EH:000001:000000:000000 ;
#01# ByteOrder     = LowByteFirst ;
#02# DataType      = UnsignedShort ;
#03# Dim_1         = 670 ;
#04# Dim_2         = 650 ;
#05# Image         = 5 ;
#06# Detector_Xsize = 1340 ;
#07# Exposure_sec  = 200.000000 ;
#08# Detector_Ysize = 1300 ;
#09# Experiment_date = 01Feb2003 ;
#10# Experiment_time = 00:00 ;
#11# Exposure_msec = 1 ;
#12# ROI_info      = 1 X=(1-1340)/2 Y=(1-1300)/2 ;
#13# ROI_info      = 2 X=(500-901)/2 Y=(500-821)/2 ;
#14# Software      = ../xy32lib/wv_lib.c (Jun 27 2003) based on Winhead.txt (1.6) ;
# effective size : (670-0)/1 * (650-0)/1 => 670 * 650 = 435500
# test_dir/fluo6.SPE : 5 frames used
# 0 saturated pixels found
# test_dir/fluo6.SPE : 5 frame(s), mean 666.1 sigma 59.8 [165@39*0 - 13294@139*139] 435500!=0, 435500 used
# 0 pixel(s) corrected
# test_dir/fluo6.SPE : corrected (dark=,flat=,grid=none) and 0 pixel(s) corrections (s=0,t=0,s=0,l=0)
# test_dir/fluo6.SPE : 5 frame(s), mean 666.1 sigma 59.8 [165@39*0 - 13294@139*139] 435500!=0, 435500 used

# base image test_dir/fluo6.SPE -> test_dir/fluo6.SPE (test_dir/fluo6.SPE)
```

Note that in the above multiframe image, the statistics are associated with the whole image consisting in the sum of the frame and not by the dispersion within the frames.

### 2.2 Format conversion: "*bm2img fname %X*".

It can used simply to perform some format conversion, as the result of a multiframe file consists in a simple image. The added frames from the previous example can be saved to a 32 bit EsrfDataFormat (.edf) using the command *bm2img test\_dir/fluo6.SPE %X*. If you wish, you can specify the name and/or the extension of the new image : *bm2img test\_dir/fluo6.SPE %X=new.typ*

3 output commands are defined to specify the dynamic of the output image.

command	output dynamics	default output format
%X	32 bits	.edf
%Y	16 bits	.gel
%Z	8 bits	.tif

*bm2img* can output images in the following format :

- .edf Esrf Data Format : 32 signed integers

## 2.3 Dark current correction : "*bm2img D=dname fname*".

The first correction, for CCD image is to remove the dark current, in practice this correction is also used to remove the ADC offset. This is done in a simple way *bm2img D=dark\_fname img\_fname*. When the resulting file is saved a "d" is added to its name allowing to recognise easily dark corrected images.

The following command transforms the winview file *dani10.spe* into a 32bit file corrected for dark current... *dani10d.edf* file. *bm2img D=dark dani10 %X*

It is of the responsibility of the user to ensure that the dark and the image have been measured in similar conditions (exposure time, camera parameters...).

### 2.3.1 Multiframe images.

When the required dynamic is more than a few  $10^4$ , it is mandatory to add images, this can be done with the *winview* CCD program which is able to record multiframe images : in this case, the resulting file contains all the images. Such files can be corrected with the previous command as *bm2img* is able to recognise multiframe image and to rescale the dark file to the same number of frames than the image before removing it. However when the number of frame becomes large, it may be more efficient to build for the dark image a frame added file and to use it for further processing : this can avoid some hundred of megabytes to be read each time. In this case the number of added frames had to be specified as an input parameters, the following similar command applies to a preprocessed dark.edf file that is the sum of 15 frames. *bm2img D=dark.edf,U15 dani10 %X* When an image has been stored in edf format, it is always assumed that this image is made by only one frame, then it is mandatory to use the U parameter to correct this if the dark file contains more than one frame.

A reduced view of a SAXS image can be seen below. In the central part it shows a small angle scattering image using 2\*2 binning. To show the influence of the correction on 8 bit .gif images, intensities higher than 3 times the image average have been truncated; this occurs near the beamstop and/or on cosmic rays.

## 2.4 Flatfield correction : "*bm2img F=dname fname*".

## 3 Input file specifications.

When an argument is read, the file name is scanned to see if a modifier is specified. The modifiers are separated from the name by a comma, no spacing character is allowed. This values may override values specified in the initialisation file.

On read all file can be specified with the `[D=—F=]img_file[,Bb,Ee,X,Aa—Mm,Uu] [img...]` example `D=dark3,B2,A5` add frame starting at 2 in dark\_files dark3 to dark8, U force used frames to u `D=dark_file` to subtract, `F=flatfield_file` (must be defined before image Begin at frame b, End at frame e, X don't correct Add a files with increasing number (or Merge them, DMI specific)