Report for topical project on UVIT_simulator

Dipanjan Mukherjee
Under the supervision of
Shyam Tandon

August 25, 2010

1 Introduction

The UVIT_simulator simulation software simulates the image as expected to be made by the UVIT instrument on board the ASTROSAT telescope. It takes an input image supplied by the user and generates frames having photon footprints as will be obtained by the UVIT for a given exposure period. The final image is reconstructed from the position of the photon footprints which are determined by a centroiding algorithm. The software can be run on any Linux/UNIX based machine supporting bash shell and C compiler (gcc 4.1.1 or higher). Cfitsio routines are used and will be required to be installed by the user as directed in the installation section. The software needs 1GB free hard disk space where it is to be installed and used.

2 Installation

1. Unpack the UVIT_simulator.tar.gz into your local disk.

   > tar xvfz UVIT_simulator.tar.gz

2. A folder named UVIT_simulator will be created.

   > cd UVIT_simulator

   It will contain an empty folder C_lib. Cfitsio library would be needed to be installed in that folder. UVIT_simulator uses cfitsio library for operations with fits images.

   Cfits library can be freely obtained and downloaded from the website :

   http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html

3. Copy the downloaded tar file (cfitsio3030.tar.gz or whichever current version is available from the website) to the C_lib folder and untar it.

   > cd C_lib

> tar xvfz cfitsio3030.tar.gz

> cd cfitsio

4. Execute the following commands :

> ./configure

> make

Refer the Quick Start guide and C Reference guide in the above web page for more details if required.

Doing the above will create libcfitsio.a .

The UVIT_simulator can now be used for simulation runs.

3 Running UVIT_simulator

Exit from C_lib to the UVIT_simulator directory. The following files or directories will be seen :

1. **C_lib** : library files for cfitsio routines. The source code is linked to this directory and library files in it.

2. **input_image** : directory containing the input image to be simulated. Copy the input image into this directory for simulation. The image should be of the following type :

   (a) It should be a 2-D intensity distribution FITS-file (32 bit per pixel, float variable).

   (b) The intensity in a pixel should be in units of counts per second per sq cm.

   (c) It should be a square image of size not larger than 25.6 arc-minutes. It is however recommended that input image should not be larger than 15 arc-minutes to ensure uniform sampling during frame generation. This is due to the effect of satellite drift that has been incorporated as explained in [1]. Drift has been applied from data supplied by ISRO satellite center for total 4350s and is repeated for integration time more than this. The maximum drift of the center of the image is approximately 5 arc-minutes. For image size larger than 15' areas from the edge may drift out of the simulation region and drift in later depending on the nature of drift. Thus areas from the edge will not be sampled at the same rate as inner areas during the exposure. This may introduce errors in the photometry although the code will run even if image of larger size is provided.

3. **output_image** : directory where output image will be produced.

4. **Src** : directory containing the source codes. The user would generally not need to change anything in this directory.

5. **user_defined_parameters.dat** : This is a text file containing the user defined parameters which are needed to be defined at the start of the simulation. The file defines the following variables (the order should be maintained) :

   (a) **INPUT_IMAGE_FILENAME** : The name of the input image (eg. image.fits).

   (b) **INPUT_PIXEL_SCALE(ARC_SEC)** : Pixel scale of the input image in *arc-seconds* per pixel.

   (c) **SIGMA_OPTICS_PSF(ARC_SEC)** : The σ in *arc-seconds* on each of the x-y axes of point spread function of UVIT optics. (The spread is taken as a symmetric 2-D Gaussian. This spread includes the spread in optics and the detector. A typical value is 0.7.)

   (d) **UVIT_EFFECTIVE_AREA(cm_sq)** : Effective area of the UVIT-channel in cm^2. This depends on filter used. Typical value is 25. Refer UVIT specifications for more details.

   (e) **TOTAL_INTEGRATION_TIME(secs)** : Total exposure time in *seconds* in the range 5 s to 100000s.

   (f) **FRAMES_PER_SECOND** : Rate of frame sampling in units of *frames per second*. For full 25 arc-minute field the maximum rate possible is 30/s and it can be about 200/s for a field of 10 arc-minute by 10 arc-minute.

   (g) **Centroiding Algorithm** :
   Photon events in the frames are identified by taking a window of 3x3 square or 5x5 square of pixels (for 3-square and 5-square algorithms respectively), and moving it pixel by pixel on rows/columns in a raster scan. At each location of the window, several checks are made to decide if the window carries a candidate event. A candidate event is indicated if the following conditions are met:
   1. The central pixel has intensity higher than other pixels in the square.
   2. The central pixel has intensity higher than PIXEL_THRESHOLD (user defined variable).
   3. Total intensity of all pixels in the square is larger than ENERGY_THRESHOLD (user defined variable).
   In order to find intensity of the pixels, the background is determined by finding the minimum of the four corner pixels of the 5x5 square surrounding the central pixel. Since we need a 5x5 square for these operations, a buffer two pixels is left on all edges.
   Centroiding is done by center of gravity method.
   The choice of the centroiding algorithm is determined by the following two parameters :

      i. **FIVE_SQUARE** : Choice for five square centroiding algorithm [2]. In this algorithm the centroid is found for the pixels in a 5x5 square. To opt for this algorithm set this variable to 1. Set to 0 (or any other numeric value) to deactivate this algorithm.

ii. **THREE_SQUARE** : Choice for three square centroiding algorithm [2].In this algorithm the centroid is found for the pixels in a 3x3 square. To opt for this algorithm set this variable to 1. Set to 0 (or any other numeric value) to deactivate this algorithm.

Both three square and five square can be set to 1 to produce imaging by three square and five square algorithm simultaneously but time taken will be more in such a case.

(h) **REJECTION_THRESHOLD** : Threshold to identify and eliminate double events [1]. Double events occur when two photon footprints occur in nearly adjacent pixels and influence the photometry so that centroiding algorithms give inaccurate results. They are identified by finding the difference between the brightest and faintest of the corner pixels of a 5x5 array around the central pixel. If this difference is more than the REJECTION_THRESHOLD then such events are neglected. A typical value is 50.

(i) **PIXEL_ENERGY_THRESHOLD** : If intensity of the central pixel (background subtracted) is above PIXEL_THRESHOLD then those events are considered for subsequent centroiding analysis, rest are neglected as false events/noise. Typical value is 150.

(j) **TOTAL_ENERGY_THRESHOLD** : If total intensity (background subtracted) of the pixels in five square or three square (depending on centroiding algorithm) is larger than TOTAL_ENERGY_THRESHOLD then centroiding is done. Rest are neglected as false events. Typical value is 400.

6. **run_UVIT_simulator.sh** : This is the script to start the simulation run. When the input image has been placed in input_image folder and the parameters properly set in user_defined_parameters.dat then the simulation can be run by :

> ./run_UVIT_simulator.sh

4 Output

The output images are produced in the output_image directory. The output image has a resolution of 0.5 arc-seconds per pixel with size 3072x3072 pixel. The input parameters are displayed in the fits header. The output is produced as 3sqr_inputfilename.fits or 5sqr_inputfilename.fits for centroiding algorithms 3-square and 5-square respectively.

A logfile.dat will be produced during the course of a run to help track the progress of the simulation. The UVIT simulation frames are grouped together as a batch with 900 frames per batch.

The number of batches = int (TOTAL_INTEGRATION_TIME * FRAME_PER_SEC /900) + one extra batch to account for remaining frames when TOTAL_INTEGRATION_TIME * FRAME_PER_SEC is not a multiple of 900.

The logfile.dat tracks the current batch being generated.

Note : The output in the terminal often shows a warning message with the function lrint during compilation of source codes. This is compiler dependent warning message and does not affect the output.

5 Brief description of the code and algorithms

The algorithms used have already been discussed in [1] and [2]. A brief summary of the flow chart for a typical run is given here. The source codes are present in the folder Src.

1. **Initialisation** : First the initialise.c reads the user_defined_parameters.dat file and creates a subsidiary set of files in /Src/input which are later read by other source codes as needed.

2. **Frame Generation** : Individual frames of Star-250 detector are generated in the following fashion.

   The set of codes UVIT_frames_level2.c, UVIT_frames_level3.c, UVIT_frames_level4.c, UVIT_frames_level5.c, UVIT_frames_level6.c are compiled together and the executable frame_gen.out is produced. The above set of codes read the input image and stores them in a float array.

   Photon footprint generation :

   For each pixel of the input image, average photon flux is calculated for the total integration time. The actual number of photons for each pixel is obtained using Poisson statistics. These photons are then randomly distributed over the total number of frames. The actual position of each photon event is found by application of a uniform random distribution over the pixel and of the Gaussian PSF due to optics and the detector.

   Each photon event will give rise to shower of photons (generated by the intensifier) on Star-250 detector. This shower is assumed to be a symmetric 2-D Gaussian with sigma 0.7 pixels, on each of x-y axis, of Star-250 (each pixel is about 3x3 arc-second). The number of electrons generated by this shower is simulated in the code by taking a Gaussian distribution of average of 30,000 electrons and a σ of 6000 electrons. This number was divided by 20 to get the output in digital units (DU). Thus total energy of a photon footprint is expected to have 1500 counts in digital units with a σ of 300.

   These footprints are generated on a laboratory recorded raw dark frame. The raw dark frames were taken at $\sim 22^o$ C for 34 ms. These frames have a random fluctuation in pixel value with rms of 1 DU with a gradient (a changing bias level) of 200 DU along the columns.

   The UVIT_frames_level2.c has the following parameters fixed inside the code which the user cannot change :

(a) **UVIT_pixel_scale** = 3.0; (*arc-sec* per Star-250 pixel).

(b) **prob_rej** = 0.00001; (If probability of getting a photon within "1 sec" is less than this value for a pixel, then it is assumed that no photon would be collected within that pixel over the full exposure).

(c) **no_pelec_mean** = 30000; (Mean number of electrons on the Star-250 generated).

(d) **no_pelec_sigma** = 6000; (Sigma of mean number of electrons on the Star-250 generated).

(e) **sigma_pelec_spread** = 0.7; (Sigma, on each axis, of the distribution of electrons on Star-250 in units of STAR-250 pixels).

(f) **cutoff_pelec_spread** = 3; (In the units of 'sigma_pelec_spread'; The spread of the electrons on Star-250 would be confined within this limit).

(g) **gain** = 20; (Gain of the Electronics. The total number of electrons in Star-250 would be reduced by this factor to convert to digital units.).

Drift is applied to the photon events during frame generation from typical data on drift of satellite's aspect. Drift consists of shifts along rows and columns and rotation. The drift is given for every second on an interval of 4350s. Drift is applied to a frame by liner interpolation from this table to the current total time. For total exposure greater than 4350s the drift table is repeated starting from 0 drift.

The frames are created in the directory

/Src/data_directory/fits_images/batch .

3. **Centroiding** : After a batch of 900 Star-250 frames are created, centroid.c is executed to locate the centroid of photon events using one of the two (or both) centroiding algorithms : three square and five square. The algorithms have been discussed in [1] and [2].

Background is obtained by taking the minimum of the values of 4 corner cells in a 5x5 square surrounding the central pixel. A pixel is selected as photon event if :

1. the intensity of the central pixel is larger than surrounding pixels in the 3square/5square.

2. the intensity of central pixel is larger than PIXEL_ENERGY_THRESHOLD.

3. the total intensity of the pixels in 3square/5square is larger than TOTAL_ENERGY_THRESHOLD.

Centroiding is done based on center of gravity method. For the 3-square algorithm a correction needs to applied to the coordinates of the centroid, as discussed in section 3 of [1], to account for loss of wings of the intensity distribution of photon events when taking 3-square. Such corrections are not done in 5-square algorithm.

Two separate centroid list are created for five square and three square algorithms in

/Src/output/centroidlist3sqr.dat and

/Src/output/centroidlist5sqr.dat

respectively. The output file is a 8 column file (current integration time, event no, x centroid, y centroid, central pixel intensity, total intensity (background subtracted sum of intensities of 3square/5square), background, corner-max - bkgrnd).

4. **Reconstruction of output image** : The code produces separate output images for 3-square and 5-square algorithms in output_image directory.

   The parameters fixed in this code are :

   (a) UVIT_no_pxl = 512; (Size of the image in units of Star-250 pixels)
   (b) UVIT_pxl_scale = 3.0; (pixel scale of Star-250 in arc-sec)
   (c) output_image_pxl_scale = 0.5; (pixel scale of output image in arc-sec)

   The reconstruct.c is executed after frame generation to form output image by reading photon events from list of centroids. The user defined parameter REJECTION_THRESHOLD is used to detect double events.

   All events with (corner-max - background)> REJECTION_THRESHOLD are rejected as double events.

   The photon events are read from centroid list. Drift applied earlier is corrected and error in correcting the drift is also accounted for from data in drift_error_table.dat . The user defined parameters and relevant information are placed in the header of the output fits image.

6 Results of sample runs

6.1 Sample run with default parameters

Some sample runs were made using an input image adopted from the image of M-51 galaxy in GALEX database. The image in GALEX database is made on pixels of 1.5 arc-seconds. In order to generate an image with pixels of 0.5 arc-seconds, the galaxy was simulated to be placed at 3 times its actual distance.

Fig.1 shows the result of a sample run with the input image on the top panel for comparison. The run was carried out for the following input parameters :

INPUT_PIXEL_SCALE(ARC_SEC)   = 0.5
SIGMA_OPTICS_PSF(ARC_SEC)   = 0.7
UVIT_EFFECTIVE_AREA(cm_sq)   = 60
TOTAL_INTEGRATION_TIME(secs) = 9000.0
FRAMES_PER_SECOND   = 30.0
FIVE_SQUARE   = 0
THREE_SQUARE   = 1
REJECTION_THRESHOLD   = 50.0
PIXEL_ENERGY_THRESHOLD   = 150.0
TOTAL_ENERGY_THRESHOLD   = 400.0

Except for input pixel scale and total integration time, the values of other parameters used are the default parameters which will be supplied with the software on untarring the software.

For the above simulation the mean energy and σ per photon event (Total energy) for events with corner max - corner min < 30 was found to be :

Mean   = 1602.347
σ = 319.977
Total photon events considered   = 1775352

This is in approximate agreement to the initial assumption of a shower of 30,000 electrons with a σ of 6000 divided by gain of 20 to simulate a photon footprint. The deviation of the mean from expected 1500 is due to the inherent counts of the dark frames on which photon footprints are generated.

The sigma spread for each photon footprint was studied by selecting photon footprints from sample data frames and examining the FWHM using IRAF's imexamine quick gaussian/moffat fit tool. For a sample of 25 photon footprints, the average FWHM was found to be 1.67 pixels (in units of STAR 250 UVIT detector). If we assume a gaussian distribution this gives a $\sigma \sim 0.71$. This is again in agreement with the initial set up discussed previously.
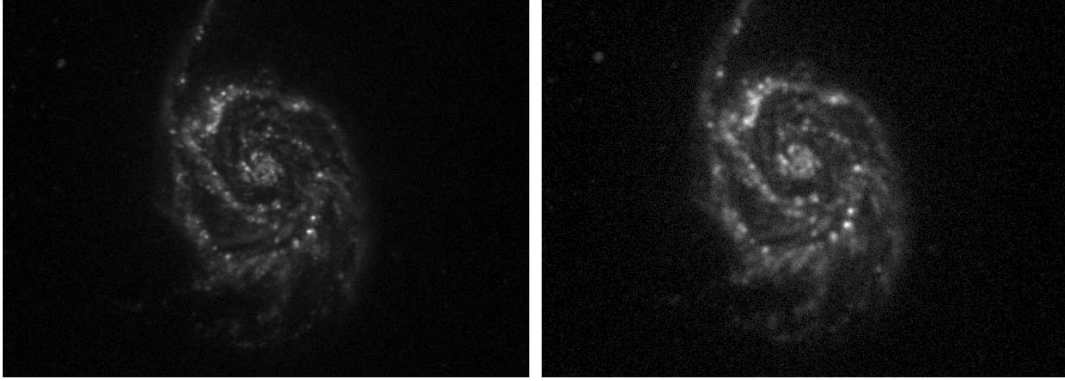
Figure 1: The left figure is the input image and the right figure is the output image of the simulation. The images are 3.7' x 5.1' with pixels of 0.5"x 0.5". The simulation was carried for an exposure time of 9000s. Three-square centroiding algorithm was used with a pixel energy threshold = 150 and total energy threshold = 400. For reconstruction rejection threshold was taken as 50.

6.2   Effect of SIGMA_OPTICS_PSF

Effect of PSF due to optics and the detector was studied using the software and on carrying out the same simulation with SIGMA_OPTICS_PSF(ARC_SEC) set to 2.1 it was seen that the image degrades drastically. The result is shown in Fig.2.
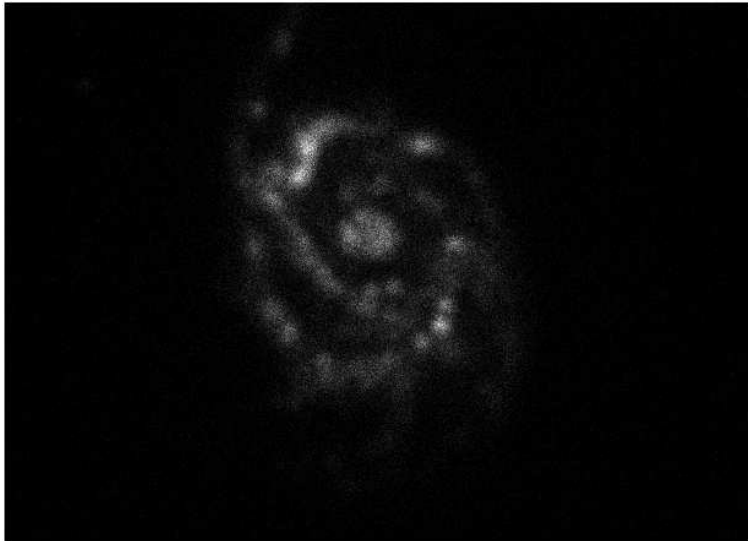

Figure 2: A sample simulation with σ_OPTIC_PSF set to 2.1. The rest of the parameters are same as in Fig.1 . The resolution degrades very much as compared to Fig.1
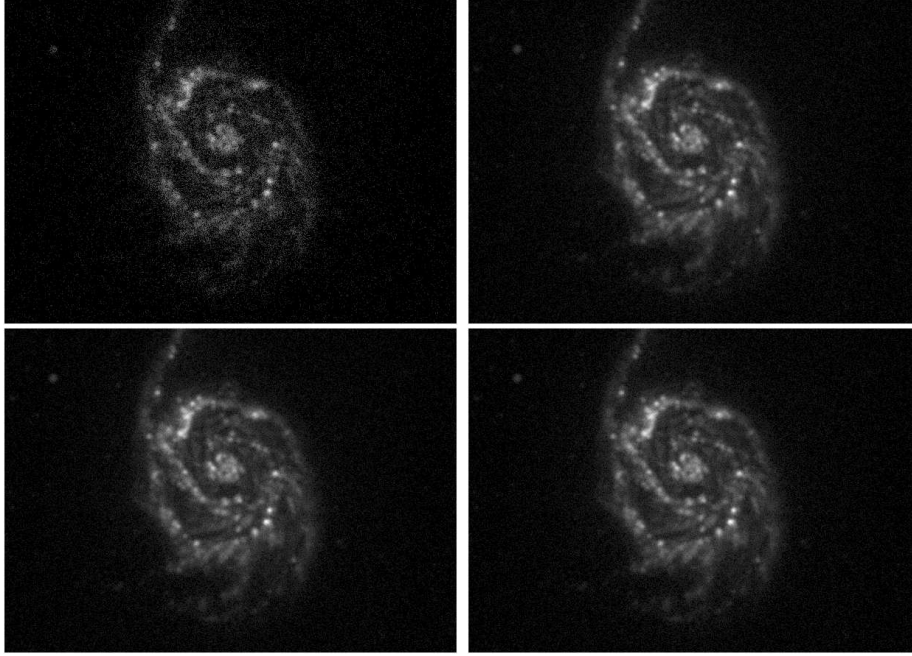
6.3   Effect of rejection threshold

Figure 3: A sample simulations with different values of rejection threshold. Rejection threshold chosen was 5 and 30 for upper left and right images respectively. Rejection threshold was 70 and 100 for lower left and right images respectively.

Multiple simulation runs were carried out with different values of rejection threshold. The results have been presented in Fig.3. For very low rejection threshold (for example rejection threshold = 5) the intensity is very low as most of the events are skipped. Setting rejection threshold to 0 gives a blank image as no events are selected. It is seen that rejection thresholds from 30 to 70 gives similar kind of images, with some increase in counts per pixel with higher rejection threshold.

The following table lists the mean, σ, minimum count and maximum count per pixel in an area of 12" by 12" in the centre of the image, where probability for double photon event is maximum. The figures are given as ratios corresponding to values for rejection threshold 50.

Rejection Threshold	Mean	σ	Min count	Max count
5	0.12	0.192	0	0.144
30	0.972	0.972	1	0.969
50	1	1	1	1
70	1.019	1.016	1	1.032
100	1.037	1.033	1	1.032

Table : 1
The values for Rejection Threshold = 50 are :

Rejection Threshold	Mean	σ	Min count	Max count
50	41.74	13.96	10	97

10

As is seen from above for rejection thresholds 30 to 70, the maximum count and mean varies marginally. For a rejection threshold of 100, there is not much change in pixel counts from the case with rejection threshold set as 70. Thus for the given image for rejection threshold above 70, less number of overlapping events are chosen as double events. The image thus contains events which are partially overlapping but not rejected as double events as rejection threshold is high. Thus photometry may be inaccurate for such a choice of parameter range.

In general, two or more photon events within 4 pixels of UVIT detector will cause overlapping photon footprints. The overlap is recognised from the value of corner max - corner min. For a high value of this quantity, there is photon footprint overlap (hence a double event) and photometry will be affected leading to incorrect centroid determination.
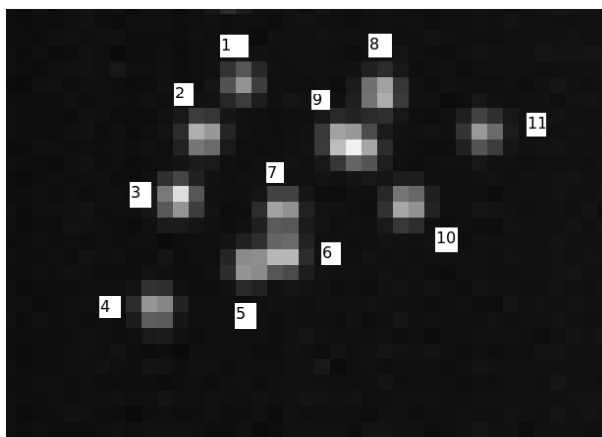

Figure 4: The above is the central part of a simulated image of a sample Star-250 data frame containing photo footprints and many potential double event candidates.

Fig.4 shows a zoomed in image of a sample UVIT data frame obtained from a simulation designed to increase the photon events per frame. This was done to study the effect of overlap and how the value of corner max - corner min varies for these events. Table 2 lists the events and the corner max - corner min values :

Event no	Corner Max - Corner Min
1	88
2	52
3	23
4	7
5	189
6	54
7	38
8	308
9	62
10	19
11	13

Table : 2

1. From table 2 and Fig.4 we can see that events 1 and 2 are overlapping and their corner max - corner min are high compared to single events. Events 3 and 4 are single events.

2. Events 5, 6 and 7 seem to be strongly overlapping with high value of corner max - min (value > 50).

3. Events 8 and 9 are again overlapping events with corner max - corner min > 50.

4. Events 10 and 11 are single events.

From the above trend it can be seen that when two photon events overlap (the central maxima are within ~ 12" or within 4 Star-250 pixel), the corner max - corner min is higher than 50. Setting a rejection threshold to 30 will effectively rule out all overlapping events and only consider single isolated photon events during image reconstruction. This will reduce the possibility of inaccuracies in photometry from incorrect centroid detection from double event.

The above trend was seen to hold in general for other data frames for this input image and also for another input image which was tested separately.

7 Reference

1. M.K. Srivasatava, S.M. Prabhudesai, S.N. Tandon; PASP, 121, 621; 2009.

2. J.B. Hutchings, J. Postma, D. Asquin, D. Leahy; PASP, 119, 1152; 2007. 3. 3. http://meghnad.iucaa.ernet.in/ astrosat/instruments.html