XMM Optical Monitor

MULLARD SPACE SCIENCE LABORATORY
UNIVERSITY COLLEGE LONDON

Author: D.A. Bone, UCL

Blue Detector Engineering Setup modes

DOCUMENT: XMM-OM/MSSL/TC/0032.01 20-Oct-94

1 Introduction

We are now entering the design phase of the processing electronics for the EOB-2 version of the blue detector, which will be based on space qualifiable components. As part of the design, two engineering setup facilities must be included. These are:

1. A test facility for characterising the performance of the image intensifier and CCD camera in the system. Over the 10 year lifetime of the XMM mission it is expected that the gain of the intensifier will decrease and so that the voltages to the intensifier can be adjusted to compensate, measurement of the changes will be required.
2. A procedure for calculating and loading the X and Y centroid lookup tables. The scheme as used by EOB-1 cannot be used due to the large amount of time used in the procedure and a different scheme, which has been adopted by the MIC ground based program, is detailed as the alternative.

A number of options for providing these facilities were detailed, from the point of view of the Blue Module Processing Electronics, in the document XMM-OM/MSSL/TC0030.1. From these the BDSG selected a baseline approach of the 'Special image format' (section 5). This document describes this approach in more detail with emphasis on how it operates in the spacecraft environment and the impact on the other systems involved — DPU, ICU and Ground.

2 Detector head characterisation

It is considered important that the characteristics of the intensifier, as seen by the CCD, of the blue detector, be measured at intervals throughout the lifetime of the detector. For these measurements the CCD video data must be examined to analyse details of individual events, giving peak height and total charge (energy) for each. The results are binned to give distributions as histograms.

The normal procedure is to illuminate the detector, preferably with an 'even' source of illumination ("flat field") of low to medium count rate to minimise the coincidences between events, which, because parameters of individual events are required, will corrupt the results. Then CCD data frames of digitised video are searched for event peaks. As shown in Fig. 1 these are recognised by pixels in the data exceeding a threshold level (to avoid noise) and being greater than the adjacent pixels 'right', 'left', 'top' and 'below'. Thus; $A < B > C$ and $B1 < B > B3$ and $B > theshold$. The central pixel, B (the event peak), points to the appropriate bin of a 256 bin histogram. The event energy or charge is found by summing the 9 pixels comprising the event (peak plus surrounds), then dividing by ten (or other suitable factor) to reduce the range of values to less than 256, then accessing the appropriate bin of a histogram.

The baseline calls for just event height and energy to be logged, thus avoiding event width measurement which would require more complex calculation. Frames are examined until sufficient data has been tallied

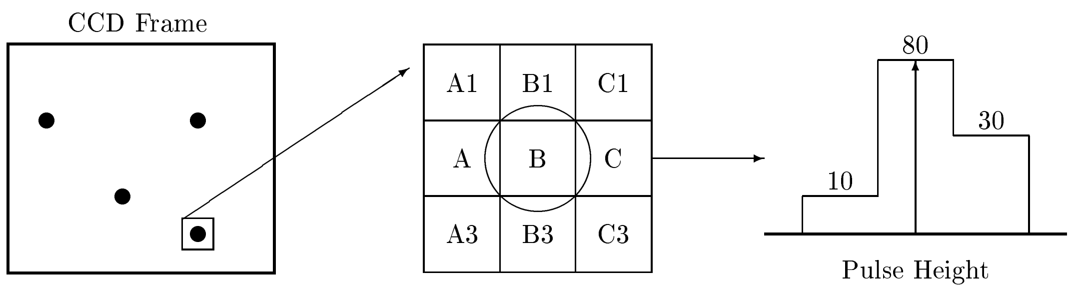

Figure 1: Photon event representation

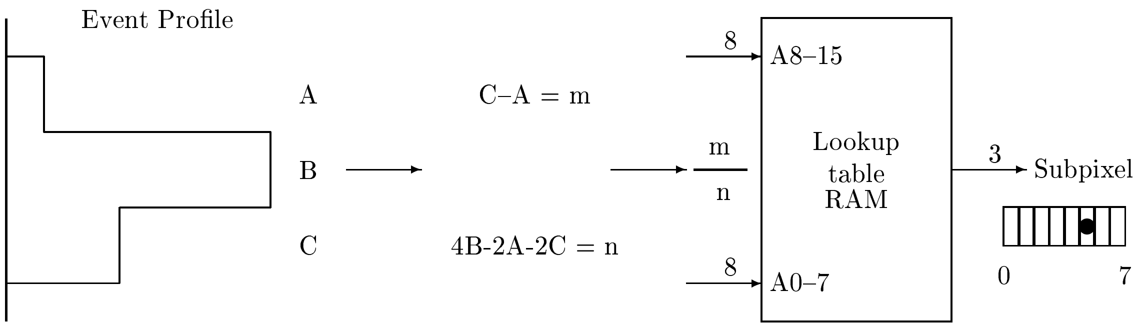

Figure 2: Event centroiding

— 3000 events is a usual figure to give a useful height distribution (a greater range of bins is covered than for the other distributions).

The histograms are made use of in the following manner. The intensifier gain is adjusted by changing the EHT voltages applied to it, until the peak of the pulse height distribution is around 60–75 ADUs. This ensures the optimum performance of the detector is maintained as far as possible (eg. avoiding the loss of photon events as the gain decreases). The event energy can be used as a additional diagnostic for gain decrease in addition to the determination of a double event threshold to be applied when gathering data. This feature, if enabled, checks the energy within an event against a threshold, which, if exceeded, is taken as being two events coinciding and will hence be counted twice.

3 Centroiding Lookup table calculation

After an event is located to a CCD pixel, it is centroided using an algorithm (as shown in Fig. 2), resulting in two numbers '**m**' and '**n**'. Their division yields a number between −0.5 and +0.5 which is the position within the CCD pixel of the centre of the event (this is done independently in X and Y). This range is divided up into 8 sections or bins which are known as sub-pixels. This division and sorting into bins is performed by a lookup table RAM whose input are the 8 bit numbers **m** and **n** and which gives a 3 bit output of the sub-pixel number for the event. A routine, given a list of bin sizes (or sub-pixel boundary positions), calculates the lookup table interpretation for each combination of **m** and **n**.

In an ideal case one would expect equally spaced boundaries in the answer range −0.5 to +0.5. This was used as a starting point for the original iterative method of obtaining the final lookup table(s) contents, and serves to introduce the problems involved in the centroiding.

If the lookup tables (X and Y) are loaded initially with centroid positions using this assumption then, for example, if the answer to **m/n** is in the range −0.375 – −0.25 the output is 1. Again, an 'even' source of illumination of the detector is desired, of low to medium count rate (the final table has some dependance on count rate). An integration is performed of the flat field for typically 50–300sec. The camera format is a single window, centrally placed, say 128 pixels square, and full resolution centroiding is used. After the integration, the image is examined — cross sections in X and Y will shown a fixed pattern repeating on a CCD pixel basis as shown in Fig. 3 (X and Y will be different). The signal to noise of the cross sections is improved by summing along the rows and columns of the image to give two 1D plots (ie. adding all 128 rows together for X). The data is reduced still further by folding the data sets, summing every eighth value, into two sets of 8 numbers (eg. Total 0 equals total counts in sub-pixels 0, Total 1 the total counts in sub-pixels 1, etc.), representing the total counts in the sub-pixels 0–7.

The fixed pattern arises because of factors like the event profile being non-gaussian, uneven response across a CCD pixel, asymmetries and so on, and has a severe effect on the positional accuracy of spot images on the detector. The pattern can be considered arising from some sub-pixels having too few events detected in them while others have too many. This may be overcome by altering the boundary positions of the sub-pixels so that all sub-pixels receive the mean count level (eg. if sub-pixel 2, starting

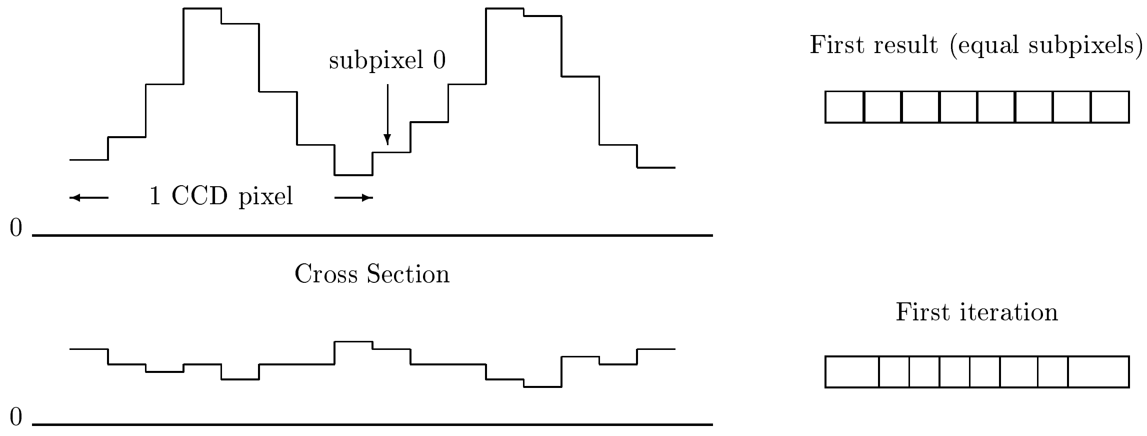

Figure 3: Pattern correction in flat fields

with boundaries -0.25 and -0.125 has too many counts, then according to its relative difference from the mean count level, the boundaries would alter to say -0.23 and -0.13). The lookup table is loaded again with the latest sub-pixel boundaries derived as above (note that X and Y can be calculated from the same data set) and another integration performed, as illustrated in Fig. 3. As the correction is linear in nature, it is quite crude, so though the next iteration reduces the fixed pattern further, residual pattern will be present and further iterations will therefore be required.

This was clearly a slow method but with access to the raw event positions before assignment to a sub-pixel, a more detailed look at the patterning across a CCD pixel is possible, enabling accurate sub-pixel boundaries to be determined from a single measurement. This is the principle of the baseline scheme described next.

The lookup tables are calculated from data obtained from individual events which arrive in frames of CCD data. Each frame is searched for valid events (In the MIC system it was possible to capture a frame of CCD data in a frame store) using the same system as is wired into the electronics. Then, *exactly as the hardware does it*, these are centroided to the stage of the division for $\mathbf{m}/\mathbf{n}$ giving a result in the range -0.5 to $+0.5$ a CCD pixel. Adding 0.5 and multiplying by 4096 allows the result to be stored in a 4096 element histogram for each event. This is done simultaneously in X and Y but into separate histograms. From experiments, in excess of 24000 events are needed to give good histograms for the next step, which could require some 500 (large) frames to be examined, depending on event rate. It may be noted from Fig. 2 that just 8 bits (lsb) are used to represent **m** and **n** and that it possible to overflow these values ($\mathbf{n}$ = 4B - 2A - 2C which can give a result greater than 255). With normal, typical events, it has been shown by experiment that **m** never exceeds the range -128 to $+127$ and that few events give **n** larger than 255. It is for unusual event profiles, such as those caused by coincident events, which are more frequent at high count rates, that the 8 bit range of **n** is exceeded significantly frequently. The range of **m** will be exceeded at high count rates also, but in this regime an accurate position of such an event cannot be expected. The ground based MIC uses an auto-ranging technique (which is to be applied to XMM systems from EOB-2 onwards, instead of throwing away the LSB) — normally the least significant 8 bits of **m** and **n** are passed, but if **m** exceeds 255, then the sign extended values of both **m** and **n** are divided by two before being passed, thus squeezing into 8 bits. The tiny loss of $\mathbf{m}/\mathbf{n}$ accuracy would only be noticeable at a sub-pixel boundary and in any case most often occurs with those events whose position is somewhat uncertain due to coincidence etc. Indeed, the maximum positioning error from this is approximately equivalent to the sub-pixel boundary positioning error imposed by the size of the lookup table (1/255). To prevent an effective sign-change in **m**, it is forced to 127 or -128 if its range is exceeded, so that the event doesn't get counted on the wrong side of the CCD pixel at least.

The histograms of $\mathbf{m}/\mathbf{n}$ values are examined — in an ideal world they would be a straight line indicating that data is spread evenly across the sub-pixels but there tends to be a general gathering of data towards the centre with a sharp central dip. There are also spikes present in the data, principly at $\mathbf{m}/\mathbf{n} = 0$, due to preferred values in the answer range (ultimately traceable to the digitisation of the CCD data). The total counts in the histogram are summed and divided by the number of sub-pixels — this is the desired total count for a single sub-pixel. Starting at one end of the histogram, sum counts from the histogram until the desired total is reached, whereupon the histogram bin number gives directly the **m/n** value or boundary position (in the range −0.5 to +0.5) of the first sub-pixel. The running total is cleared and further bins totaled to find the next sub-pixel boundary and so on. The correct sub-pixel sizes are found in one measurement and so the Lookup tables are loaded only once with no need of iteration. In certain cases of intensifier–CCD orientation, there could still a fixed pattern remaining from this due to spikes in the histogram. The effect cannot be removed without, perhaps, a more sophisticated allocation of sub-pixel boundaries.

4 Implementation by electronics

Within the Blue detector processing electronics there are circuit blocks designed to search for events in the frames of CCD data and to find values of event hieght, event energy, and the values of **m** and **n** in both X and Y. These are required in the process of normal integrations — we just need to provide a method of logging this information efficiently. The fastest data link is that to the **DPU** and this avoids heavy traffic on the **MACS**bus to the **ICU**.

In normal operation the link to the **DPU** uses a 23 bit wide FIFO in the processing electronics to buffer event address/window ID information, which is then serialised, input to the **DPU** and reorganised into an address within its memory space and that location incremented. This scheme is proposed to be extended by some engineering setup modes, and this is illustrated in Fig. 4.

The processing electronics already has a 4 way multiplexer to select the various resolution and windowing options — by adding further ways additional information can be sent. By carefully formatting the FIFO input data, it is possible to make use of the same **DPU** acquisition routine to increment memory locations in the form of 1 or 2D histograms/images (after all a histogram is an image with 1 row!). In Fig. 4 an example of how the FIFO input word could be formatted is shown — it is of course easy to re-sort the data to best suit the **DPU** software. This example will be described in more detail. It is not proposed to send raw CCD pixel data down this link as was an earlier option. The **ICU** selects the appropriate mode with the multiplexer — standard data acquisition modes, event characteristics mode, or a centroiding information mode. During data acquisition, the hardware event valid signal is used to store the formatted event address in the FIFO.

To collect event hieght and energy distributions mode 6(7) is selected. Here the eight bits of the event height are encoded into the least significant portion of the normal X address — the other address bits are zero, which produces a 256 by 1 image if handled by the standard **DPU** data acquisition routine. For each event detected, first the hieght is loaded into the FIFO, then some additional circuitry (pair logic in Fig. 4) switches the multiplexer and loads the event energy into the FIFO. Thus there are two transmissions to the **DPU** per event, with a coding in the window ID bits indicating which is which. As the processing electronics finds the event energy to an 11 bit total, the 8 most significant bits of the energy is encoded into the least significant bits of the X address field so giving a 256 by 1 image of event energy/8 in the **DPU**. It is, of course, possible to send up to 9 bits of event energy information by this means.

The proposed method of collecting the centroiding information is to make the event's value of **n**, as found by the processing hardware, the least significant portion of the address (8 bits from 9) and **m** in the next more significant portion. The ninth bit is not required as the lookup RAMs are incapable of dealing with it. With the window ID value fixed, this looks like the Low or High resolution windowed modes. Using the standard acquisition routine, the **DPU** will integrate into a 256 by 256 square pseudo-image (or 65536 bin histogram depending on one's point of view) containing the **m**–**n** data. X and Y centroiding data will be gathered in two such areas, a code in the Window ID bits indicating which. To avoid the need for two integrations a little additional logic can load first the X **m** and **n** into the FIFO followed by the Y. There will be two transmissions to the **DPU** per event in this mode (mode 4(5)).

After an integration, which might be as low as 30 seconds, the centroiding information pseudo-images must be processed in order to obtain the **m/n** histograms. This saves time over performing the calculation on each incomming event. The address of a location in the integrated pseudo-image directly gives a unique

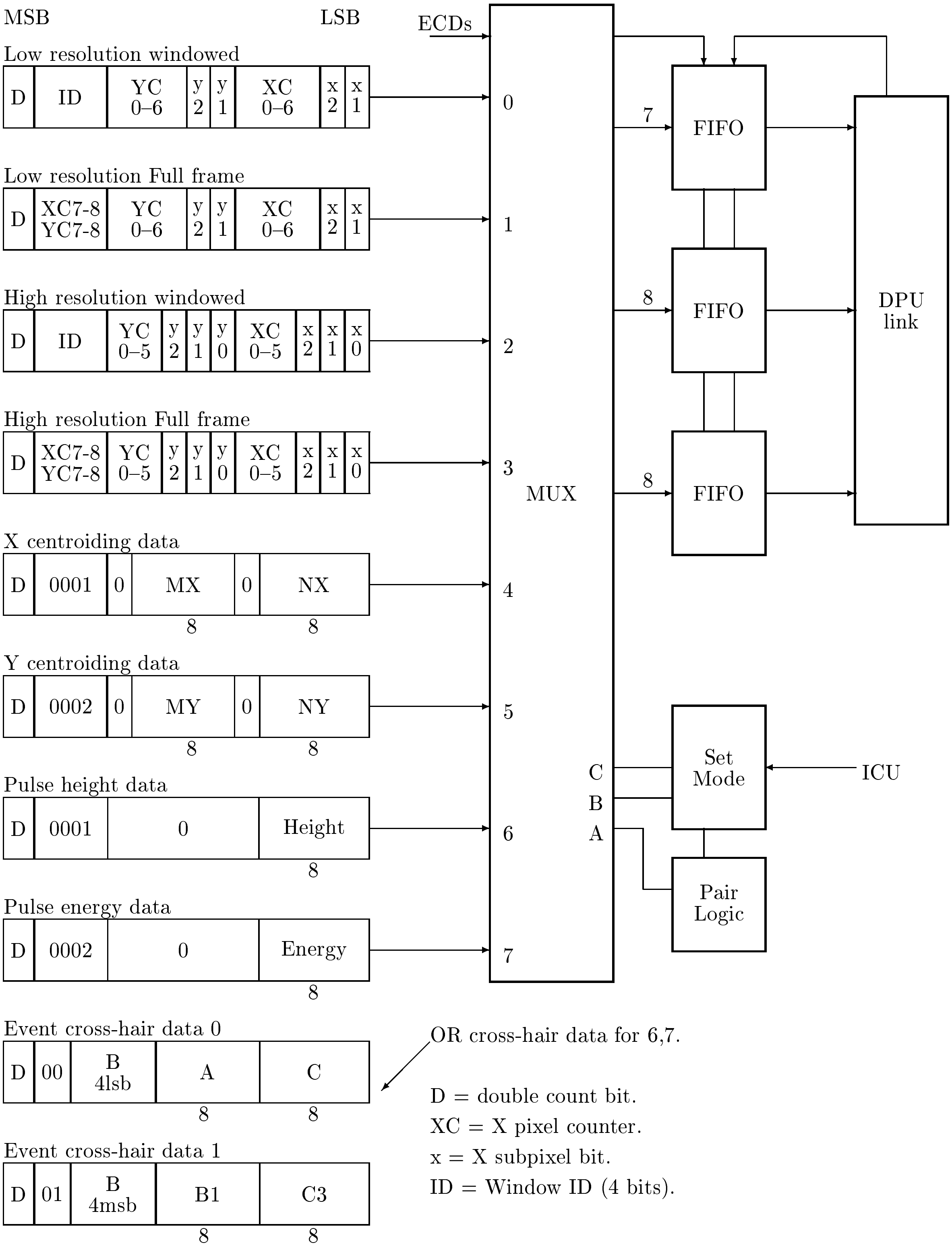

Figure 4: Special image formats

value of **m/n** (address = **m.n**) which after scaling by 4096, adding 2048 and binning by integer conversion points to a bin in a 4096 element histogram of event positions within a CCD pixel. Any (rare) values outside this range, due to asymmetric events, are added to either of the end bins. They should end up in the next CCD pixel but there is no provision in the processing electronics to cope with this. The value from this location in the pseudo-image is thus added to the histogram. An advantage of having the full array of **m–n** values is that more sophisticated algorithms for calculating sub-pixel boundaries are possible in future.

From these histograms, the sub-pixel boundaries can be calculated as previously described.

5 Setup mode procedure

Upon entering the Blue detector engineering setup phase the XMM-OM is set up to provide the roughly even illumination required, of the appropriate intensity. There are two sections to the detector setup operation requiring different interactions between the **DPU**, **ICU** and the Ground/user. Note that the exact formatting of command words to the Blue Module electronics has not yet been determined (there will be differences from EOB-1).

5.1 Intensifier characteristics measurement

The **ICU** configures the camera format to be a full frame format (ie. 256 CCD pixels square) so as to take an average over the full detector area. Smaller formats allow measurements to be taken from selected areas of the detector — allowance for this should be made in the **ICU** software, using input from the operator/user. The **ICU** disables the Double count facility (as in EOB-1) and sets acquisition mode 6 in the detector electronics, thus selecting Pulse height and energy for the **DPU** link. There is no need to load the centroiding lookup tables. The event detection threshold in the Blue module electronics is set to a much lower level than is used in normal integrations so as to include the lower end of the hieght distribution — typically 8.

Meanwhile the **DPU** performs an engineering mode one setup to suit the data format that will be arriving. It disregards the camera format information, and, since the data format is fixed, sets up two windows (IDs 1 and 2) of 256 by 1, 16 bit deep in the acquisition memory, then clears the areas. The normal windowed mode data acquisition routine is started to wait for events.

The **ICU** enables data acquisition in the blue module electronics for a few seconds, so that data on a few thousand events will be gathered, then disables it. During this time the words transmitted to the **DPU** will cause histograms of event height and energy to be built up in its acquisition memory.

At the end of the integration the **DPU** will transmit the acquisition memory contents to the Ground where they may be stored and displayed as histogram plots for viewing by the operator. If the result is unsatisfactory, then new EHT voltages for the blue detector image intensifier would be determined and the **ICU** would be instructed to access the EHT control section of the blue module electronics in order to change the voltages. Another measurement of the intensifier characteristics could follow, and so on until the detector performance is adequate.

5.2 Centroiding table calculation

In this process the **ICU** sets up the camera format to be a full frame format (ie. 256 CCD pixels square) so that an average centroiding table is found for the full detector area. There seems liitle need for smaller areas to be selected. The Blue module electronics setup is continued by disabling the Double count facility, then selecting acquisition mode 4, thus selecting event centroiding information for the **DPU** link. The **ICU** ensures that the event detection threshold in the Blue module electronics is set to the normal value for integrations — typically this is 30. There is no need to load the centroiding lookup tables for the measurement phase since they take no part in providing the centroiding information.

Meanwhile the **DPU** is setting up — the camera format information is disregarded as the data format is fixed in engineering setup mode 2. Two windows (IDs 1 and 2), each of size 256 by 256, of 16 bits deep, are allocated in its acquisition memory and cleared. These will hold the pseudo-images of **m-n** information in X and Y. The normal windowed mode data acquisition routine is started and waits for events.

The **ICU** then enables data acquisition in the Blue module electronics for some tens of seconds — at least 25000 events need to be collected. During this time pseudo-images of X and Y centroiding information build up in the DPU memory. Acquisition is then disabled.

On completion of the integration, the **DPU** transmits the acquisition memory contents to the Ground, where the two images are stored and processed for examination by the operator/user.

The 2D images need to be reduced to a histogram style distribution of the values of **m/n** in both X and Y. The address of an image location gives the values of **m** and **n**, and their division (floating point) needs to be binned — 4096 values has been used for the MIC ground based program. The following C code fragment illustrates the process:

```
short *image; /* location of an image */
/* n is the most rapidly varying index - LSB portion of address */
short m,n;
int tval;
double dval;
int histo[4096]; /* area for histogram */

for(m=0;m<256;m++)
   {
   dval = (double)(m>127 ? m-256 : m);
   for(n=1;n<256;n++)
      {
      /* find m/n value */
      tval = (int)(4096.0 * dval/(double) n) + 2048;
      /* Put any out of range values at the ends */
      if(tval > 4095) tval = 4095;
      if(tval < 0) tval = 0;

      /* add the value in the image to the histogram */
      histo[tval] += *(image + 256*m + n);
      }
   }
```

The histograms can be plotted/displayed so the user/operator can monitor the process but the next step can be automatic. The next stage is performed on the Ground — finding the new sub-pixel boundaries so that the fixed pattern can be flattened. The C code fragment following shows the method used to date:

```
int i,tot;
int *histo;  /*  to 4096 element histogram  */
double cmean; /* mean count that should be in sub-pixel */
int *tp;
int counts[8]; /* counts assigned to each sub-pixel bin */
double boundary[8]; /* storage for sub-pixel boundaries */

tot=0;
for(i=0,tp=histo;i<4096;i++) tot += *tp++;
cmean = (double)tot/8.0;
tot = 0.5 + cmean;  /* count to aim for in each bin */
```

```
for(i=0;i<8;i++) counts[i] = 0;

boundary[0] = -1.0;
for(i=0,tp=histo;i<7;i++)
   {
   while(counts[i]<tot) counts[i] += *tp++;

   if(counts[i] - tot > *(tp-1)/2)
      {
      /* used too much of bin - put it back */
      tp--;
      counts[i] -= *tp;
      }
   boundary[i+1] = (double)((int)tp - (int)histo)/4096.0 - 1.0;
   }
```

After this, the counts per sub-pixel bin and the boundaries found may be output to the user. Note: here the boundaries run from -1.0 to $+1.0$ so that the Blue module processing electronics needs not do a further division by two in the centroiding to give the -0.5 to $+0.5$ range from the centroiding algorithms. Also this method of finding the boundaries suffers from the spikes in the **m/n** distribution. If a boundary hits a spike, the spike bin is indivisible and hence in the sub-pixel counts on each side of the spike, one will perforce be too low and the other too high. This leaves residual fixed patterning in the data. In the future an algorithm may be derived if necessary, to process the full **m-n** image, but here simple boundary values (8 off) cannot be output.

The final step is to transmit the boundary values to the **ICU** and instruct it to load the lookup tables in the Blue module electronics based on this supplied information. This process is the same as used in EOB-1. It is suggested to leave provision for the **ICU** to receive a full centroiding lookup tables contents from the Ground and download it directly into the processing electronics. As this is a one shot process, this completes the setup phase.