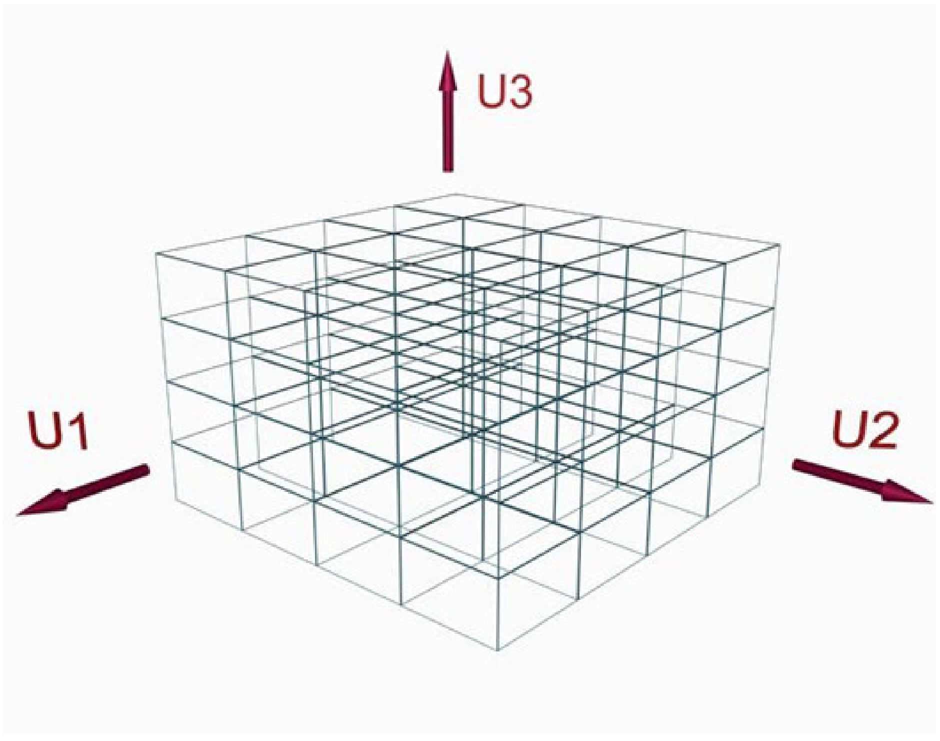
# Multiresolution Algorithm

Mslice Visualization Software add-on for Single Crystal with Position Sensitive Detectors.



Ibon bustinduy

14-11-2005

**Installation:**

Programme files are a collection of Matlab functions (source code ASCII *.m files). The .m source code should be portable between windows, UNIX/Linux platforms (such as different Linux OS, and Apple MacOs X).

```
ms_MA.m
multires2d.m
MAfromwindow.m
Micall_MA.m
Ms_updatelable_MA.m
msma.m
plot_MA.m
plotma.m
```

1. Copy all files to a directory on the hard disk, for example *'c:\mprogs\mslice\MA'*

2. Include this directory to the Matlab path: **`File > Set Path… > Add Folder…`** and select the folder where the .m files are stored ( for example: *'c:\mprogs\mslice\MA'* in windows, or *'/home/ibon/mprogs/mslice/MA'* in Unix OS.). If you do not want to change path you also can change matlab directory **`cd('c:\mprogs\mslice\MA')`** or **`cd('`*/home/ibon/mprogs/mslice/MA*`')`**and therefore routines will be visible from Matlab.

**How to use it:**

The author of this document, considers that the reader is a Mslice user, and therefore familiarized with mslice manual, therefore basic concepts will not be revised. The package usage is (by now) centred in single crystal using an instrument equipped with Position Sensitive Detectors. The process of visualization will require the user to

1. Load parameters from Mslice control window,
2. Load Data
3. Calculate Projections onto the viewing axis $U_1$, $U_2$ and $U_3$.

Once these steps had been taken, instead of plotting a equal-width binning slice, by means of clicking on "Plot Slice" button, the user can type in Matlab prompt the command:

```
>> ms_MA
```

Resulting in a new control window, where the data set constructed from Mslice has been loaded, as well as information concerning slice parameters; such as limits and binning steps. Where two new parameters: Number of Levels "NLEVEL" and noise to signal ratio "ERR/S" are added.

NLEVEL: Specifies the number of "levels" the algorithm will cover, that is, if we state, that the step in horizontal axis is given by dx, and the vertical axis step by dy, the algorithm will go collecting counts from a bin scheme of [dx, dy]. Up to a bin scheme given by [Dx, Dy]. Where Dx = dx.*(2.^(NLEVEL-1)) and Dy = dy.*(2.^( NLEVEL -1)), doubling the size of the bins as the Level grows.

ERR/S: will be used as a threshold, where a bin will be considered as a valid collection of counts as long as its noise to signal ratio stays below the given threshold; varying it from 0 to 1. For example, if we choose a ERR/S = 0.1 we are being very restrictive, forcing counts to be collected in larger bins in order to be bellow the threshold.

By default those two parameters will be 1. (number of levels = 1, means no multiresolution, and noise to signal ratio = 1, means all bins which ERR/S is bellow 100% will be accepted ).

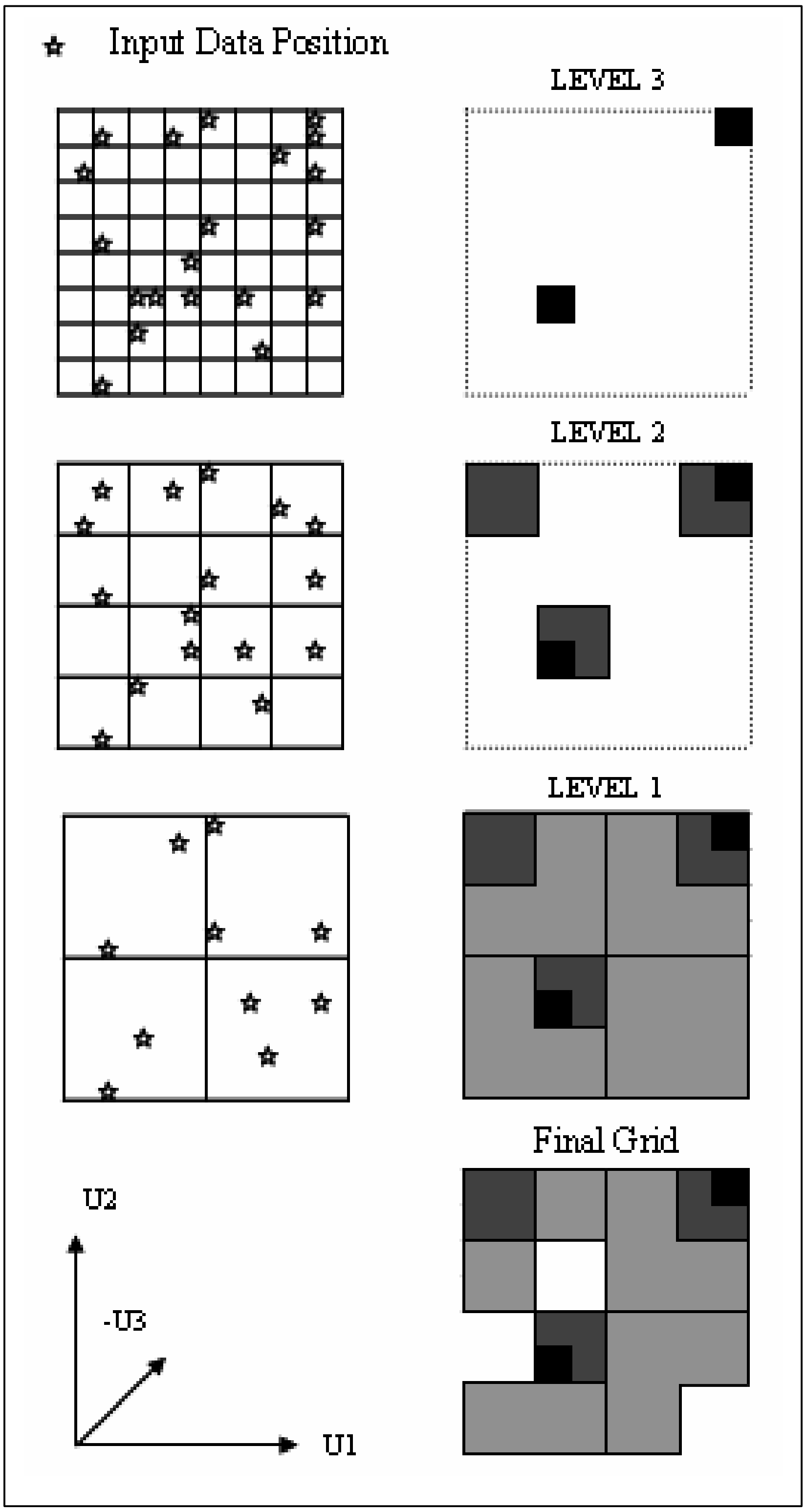
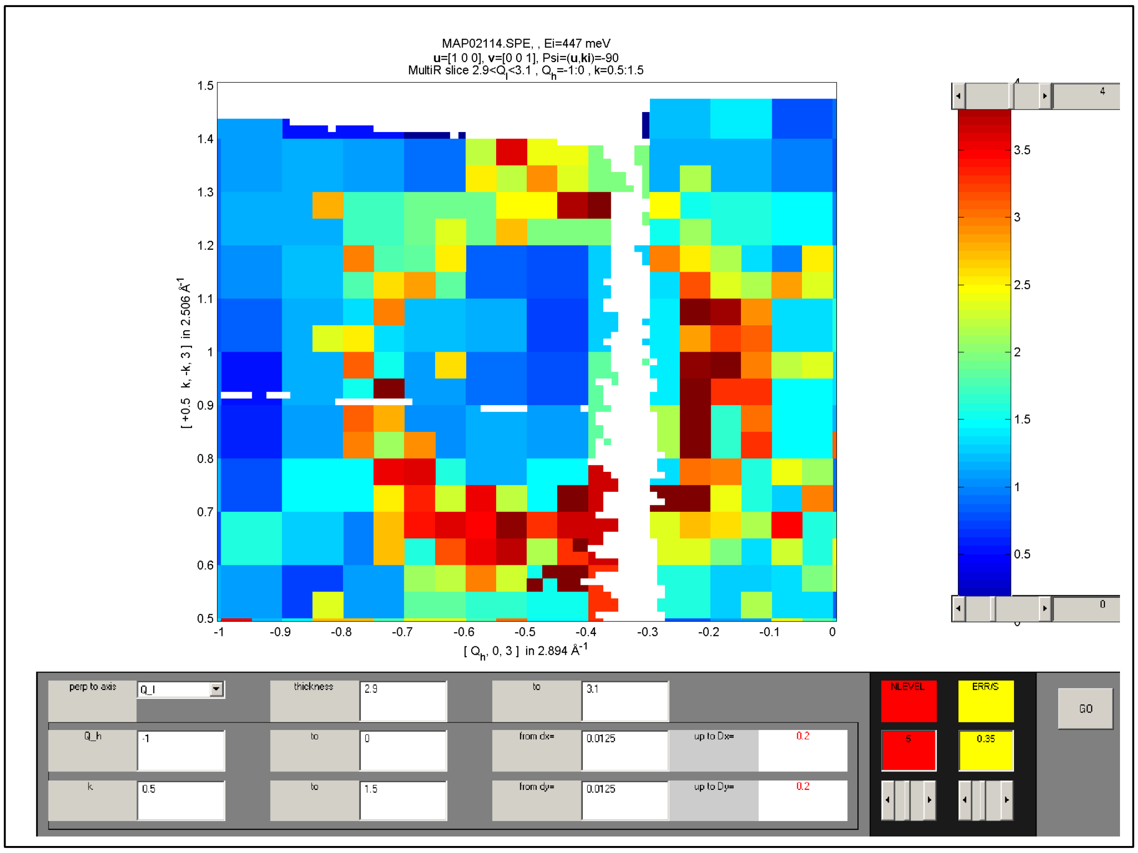


Fig. 1. To illustrate the multiresolution algorithm consider the 2D case, where the number of levels (NLEVEL) is equal to 3, and the resolution pattern has chosen to be determined by the Level 2 binning size. In the MA implementation for the sake of simplicity the resolution pattern related to instrument detectors is considered to be fixed to the highest level, correspondent to minimum bin sizes.

Those parameters can be modified from the control window, or if we prefer we also can type in from Matlab prompt the values we want to use.

```
>> ms_MA(NLEVEL, ERR/S)
```


MAP02114.SPE, , Ei=447 meV
**u**=[1 0 0], **v**=[0 0 1], Psi=(**u**,**ki**)=-90
MultiR slice 2.9<$Q_l$<3.1 , $Q_h$=-1:0 , k=0.5:1.5
[ +0.5 k, -k, 3 ] in 2.506 Å$^{-1}$
[ $Q_h$, 0, 3 ] in 2.894 Å$^{-1}$
4
0
GO

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| perp to axis | Q_l | thickness | 2.9 | to | 3.1 | | |
| Q_h | -1 | to | 0 | from dx= | 0.0125 | up to Dx= | 0.2 |
| k | 0.5 | to | 1.5 | from dy= | 0.0125 | up to Dy= | 0.2 |

| NLEVEL | ERR/S |
|---|---|
| 5 | 0.35 |

Fig. 2. Represents COBALT color map, integrated along the perpendicular axis U3= [0,0,*Ql*,0], from 2.9 to 3.1. Being U1 = [*Qh*,0,0,0], and U2 = [0.5*Qh*, -*Ql*, 0,0]. The color bar has chosen to go from 0 to 4 (abs. units). It has been produced using the Multiresolution algorithm, with a threshold of 0.35, going from the maximum binning size: [0.2, 0.2] to the minimum binning size: [0.0125, 0.0125].

## Limitations

- In order to make more efficient the algorithm has been built based in Matrix convolutions, the behaviour of the algorithm starts being inefficient when matrix size is approximately above 400 x 400.
- There is also an implementation issue related to the relation with mslice, in order to have access to the data structure, generated as a result of the main algorithm; `MA.m` is stored in the Mslice structure reserved for the equal-width binning slice structure. So if once we built our Multiresolution slice we want to compare it with a traditional slice, It will be necessary to *calculate projections* first and then *plot slice*.

- Main Question is what to do with those counts, that even in the Larger bins `LEVEL` they do not satisfy the imposed threshold by the user?, one solution is call them 'rejected counts', and plot them using the maximum bin size scheme. One way of distinguish them is by plotting them: like this:

  ```
  >> MA_d=MAfromwindow;
  >> figure
  >> p2=patch(MA_d.X2,MA_d.Y2,MA_d.S2); set(p2,'EdgeColor','none');
  ```

  The total number of counts that have not been collected below the threshold will be given by:

  ```
  >> MA_d.L_T(1)
  ```

**Other remarkable functions**

**`MAfromwindow.m`** extracts MA_d from mslice Control Window into the command line.

Shape of the structure we will obtain, where the multiresolution slice is stored:

```
MA_d =

              title: {1x3 cell}
        axis_label2: [3x36 char]
    axis_unitlength: [2x1 double]
                L_T: [4x1 double]
         axis_label: [3x3 char]
                  z: 3
              slice: [2.9000 3.1000 -1 0 0.0500 0.5000 1.5000 0.0500]

                 XX: [4x213 double]
                 YY: [4x213 double]
                 SS: [1x213 double]
                 EE: [1x213 double]

                 X2: [4x19 double]
                 Y2: [4x19 double]
                 S2: [1x19 double]
                 E2: [1x19 double]

                 XR: [4x81 double]
                 YR: [4x81 double]
                 SR: [1x81 double]
                 ER: [1x81 double]
```

We can distinguish four parts; the first one; where information related to labels, slice parameters are stored, as well as `L_T` array, this array stores the number of counts the algorithm stores at each level. `[Rejected counts; collected counts at LEVEL 1; collected counts at LEVEL 2; ….]`. `XX, YY, SS and EE` will store information about bins which `ERR/S` is below the imposed threshold, where `XX and YY` store positions of bin corners. `X2, Y2, S2 and E2` will store information about bins which `ERR/S` is NOT below the imposed threshold, and finally `XR, YR, SR and ER` will store information about bins with no counts, very useful in order to mask areas of the space where no detector exist.

**`plotma.m`** To plot a MA_d slice. Different input parameters are possible; if we specify the `edgecolor` parameter (for example *plotma(MA_d, 0, 4, 'black')*) edges of 'valid bins' (those which `ERR/S` is below the imposed threshold) will be coloured with this color, otherwise edges will not be painted. (It is useful to stick out valid bins, but at the same time if we have too many bins could be messy).

Syntax:

```
plotma(MA_d, i_min, i_max, edgecolor)
plotma(MA_d, i_min, i_max)
plotma(MA_d, i_min)
plotma(MA_d)
```

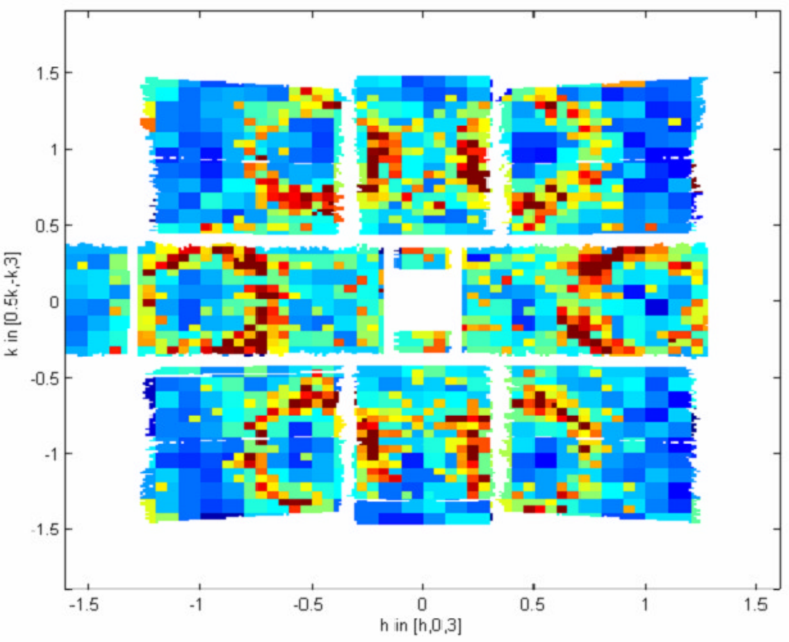


Fig. 3. Represents COBALT color map, integrated along the perpendicular axis U3= [0,0,*Ql*,0], from 2.9 to 3.1. Being U1 = [*Qh*,0,0,0], and U2 = [0.5*Qh*, -*Ql*, 0,0]. The color bar has chosen to go from 0 to 4 (abs. units). It has been produced using the Multiresolution algorithm, with a threshold of 0.35, going from the maximum binning size: [0.2, 0.2] to the minimum binning size: [0.0125, 0.0125]. Where limits have been extended approximately from -1.5 to 1.5 in both horizontal and vertical axis.