Pixel Mask Filtering of CXC Datamodel

Helen He, Mark Cresitello-Dittmar, and Kenny Glotfelty

Smithsonian Astrophysical Observatory

ABSTRACT We present pixel mask filtering, a new attribute to CIAO Datamodel (DM) library. DM facilitates a wide range of "on-the-fly" data manipulation capabilities such as copy, merge, binning and filtering. The pixel mask filtering of image/table is integrated into the CIAO region/shape filtering syntax and logic, and thus complements the conventional analytic shapes filtering of circle, box, polynomial. The integration allows for the standard operations on combining masks and shapes such as include, exclude, intersect, and union. The mask of 2D image is stored in a data block, referenced by the region filter string in the resulting file's data subspace. We will highlight the usage of mask filtering in the CIAO software and discuss some special features of the mask filtering.

1. Mask and Mask Filtering

DM provides flexible filtering syntax which can be applied to events files. The filter syntax includes analytic shapes, regions stacking shapes with arithmetic, and mask. Mask filtering is denoted by mask tag, mask(), and mask file in "mask(mask-file)".

- mask file is 2D image/table storing binary data in FITS or ASCII data columns.
- mask data type can be any numerical (real or integer) values in input.
- image can be read as mask file.
- any 2 numeric columns of binary table file can be binned as image, so as mask file
- mask filtering syntax are, but not limited to,

1. sky=mask(f)	-> f is 2D image or 2 columns numeric ascii file
2. exclude sky=mask(f)	-> read 'exclude' as to reverse mask bytes data
3. sky=mask(f1), (c1,c2)=mask(f2)	-> multi-masks to multi-desciptors: sky, (c1,c2)
4. sky=mask(f) && sky=circle(x,y,r)	-> combine (AND-operator) mask and circle
5. sky=mask(f1) || sky=region(f2)	-> combine (OR-operation) mask and region()

2. Masking Algorithm and Mask Block

- masking is to apply mask in Boolean AND/OR operation on events.
- mask data is stored as part of the data subspace in output.
- mask data is written as byte in 0 or 1 regardless of input data-type.
- mask data subspace in output is composed of 2 parts:

Part-1: 'sky' subspace Region String

sky	Real4	TABLE MASK
		MASK(MASK)||MASK(MASK2)
		Field area = <>, Region area = <>

MASK refer to Block 5
MASK2 refer to Block 6

Part-2: Data Blocks

Block 2:	EVENTS	Table	15 cols x 985	rows
Block 3:	GTI3	Table	12 cols x7	rows
Block 4:	GTI1	Table	2 cols x13	rows
Block 5:	MASK	Image	Byte(20x25)	
Block 6:	MASK2	Image	Byte(40x35)	

3. Masks Combining: AND/OR operations

- Masks Intersect is in AND-Operation

Run events masking twice in m1 and m2 each; combine m1&m2 into one mask for output.

```
dmcopy 'evt[sky=mask(m1)]' - | dmcopy '-[sky=mask(m2)]'  evt_m.out
```

- Masks Union is in OR-operation

Merge 3 events, evt1,evt2,evt3, having mask, m1, m2 and m3 each.

```
dmmerge infile=evt1,evt2,evt3  outfile=evt.merged
```

- masks are all unioned into one, MASK, if all the masks are overlap, or
- partially unioned if m1 and m3 overlap but m2 singled out, so 2 masks, MASK=m1Im3, MASK2=m2, are stored in output, or
- not unioned at all if existing GTIs per CCD regardless masks condition.

4. Mask Rebinning

The binning of events masking-and-binning is applied to the events itself and also to the mask. AND-operation is used in mask re-binning. The requirements of mask re-binning are that the run-time scale

- must match the existing masks' (spatial) coordinates, and
- must be the multiples of the mask's scale.

Create mask file in bin scale 2,

```
dmcopy 'evt1[bin sky=2]' mask.bin2
```

Bin events in scale 1 and 2 to two images, respectively

```
dmcopy 'evt[bin sky=1]'  img1
dmcopy 'evt[bin sky=2]'  img2
```

Matching Coordinates ?

NO: mask.bin2 not match img1 in scale

```
dmcopy img1'[sky=mask((mask.bin2)]' \
error.out
```

YES: mask.bin2 match img2 in scale

```
dmcopy 'img2[sky=mask(mask.bin2)]'  \
img2_msk.out
```

Bin Scale Multiple ?

look up the run-time bin scale vs masking's binning scale .

YES: bin scales (4,6) are 2's multiples.

```
'[sky=mask(mask.bin2)][bin sky=4]'
'[sky=mask(mask.bin2)][bin sky=6]'
```

NO: bin scales (3,5) are not 2's multiples.

```
'[sky=mask(mask.bin2)][bin sky=3]'
'[sky=mask(mask.bin2)][bin sky=5]'
```

Table Events Masking-and-Binning

Different outputs in masking-and-binning (1-command) and masking-then-binning (2-command) as the events is applied masking twice in 1-command run.

1-command run:

```
dmcopy 'evt[sky=mask(m)][bin sky=2]' evt1.img
```

2-command run:

```
dmcopy 'evt[sky=mask(m)]' - | dmcopy '-[bin sky=2]'  evt2.img
```

This work has been supported by NASA under contract NAS 8-03060 to the Smithsonian Astrophysical Observatory for operation of the Chandra X-ray Center.