URL: http://cxc.harvard.edu/ciao3.4/regarea.html
Last modified: December 2006

AHELP for CIAO 3.4

regArea

Context: region

Jump to: Description Examples CHANGES IN CIAO 3.3 CHANGES IN CIAO 3.2 Bugs See Also

Synopsis

Calculate the area enclosed by a region.

Syntax

```
Double_Type regArea( Region_Type Region )
Double_Type regArea( Region_Type Region, Double_Type Xmin, Double_Type
Ymin, Double_Type Xmax, Double_Type Ymax, Double_Type Pixelsize )
```

Description

The regArea routine returns the area inside a region. The simple form simply takes a CIAO region variable and returns the total area inside the region; the last 5 parameters are omitted. However, if the region is complex and needs to be 'pixelated' to compute the area you may want to use the second version.

How are the areas calculated?

For "simple" regions the area is calculated analytically. For more−complicated regions − such as when shapes overlap − the area is calculated numerically by counting how many pixels fall within the region. The default values are to loop over the bounding box of the region and consider each pixel individually. Greater accuracy can be achieved by calling the routine with the Pixelsize argument set to a value less than 1, although the routine will then take longer to run.

The xmin, ymin, xmax, and ymax arguments define the region over which the numerical calculation is performed. When the one−argument form of regArea() is called these limits default to the bounding box of the regions, if one exists, otherwise the range −DBL_MAX to DBL_MAX. An example of a region with no bounding box is

```
sector( 250, 400, 20, 45 )
```

The regExtent() routine can be used to find the bounding−box of a region.

Example 1

```
chips> require("region")
chips> r1 = regParse("circle(10,10,4)")
chips> vmessage( "The area of the region is %f", regArea(r1) )
```

Here we use the simple version of the regArea() call to calculate the area of the region

```
circle(10,10,4)
```

Example 2

```
chips> shape1 = "rotbox(4096.5,4096.5,100,150,45)"
chips> shape2 = "annulus(4096.5,4096.5,40,50)"
chips> r1 = regParse(shape1)
chips> r2 = regParse(shape2)
chips> r3 = regParse(shape1 + "−" + shape2)
chips> a1 = regArea(r1)
chips> a2 = regArea(r2)
chips> a3 = regArea(r3)
chips> vmessage( "Areas = %8.2f %8.2f %8.2f", a1, a2, a3 )
Areas = 15000.00 2827.43 12190.00
```

Here we create three shapes: a rotated rectangle, an annulus, and the rotated rectangle minus the annulus. The area of these three regions is then calculated using the simple (1 argument) version of regArea().

Example 3

```
chips> a1c = regArea(r1,4000,4000,4200,4200,0.1)
chips> a2c = regArea(r2,4000,4000,4200,4200,0.1)
chips> a3c = regArea(r3,4000,4000,4200,4200,0.1)
chips> vmessage( "Areas = %8.2f %8.2f %8.2f", a1c, a2c, a3c )
Areas = 15000.00 2827.43 12168.19
```

Here we use the extended version of the regArea() call to improve the accuracy of the calculation. It does not make a differrence for the first two regions, but the area of the third region has changed by

```
12190 − 12168.19 = 24.75
```

pixels, a 0.2% difference.

Note that the choice of bounding box is important for this calculation. The range 4000−4200 for both axes was picked since the rotated box should fit within it. The regExtent() routine can be used to automatically calculate the bounding box of a region, as follows:

```
chips> (x0,y0,x1,y1) = regExtent(r3)
chips> vmessage( "x= %.2f − %.2f y = %.2f − %.2f", x0, x1, y0, y1 )
x= 4008.11 − 4184.89 y = 4008.11 − 4184.89
```

If we had used a smaller region − i.e. one smalller than the region of interest − then regArea() would have returned the area of the region intersected with this region. For instance:

```
chips> regArea(r1,4000,4000,4200,4200,0.1)
15000
chips> regArea(r1,4050,4050,4150,4150,0.1)
9109.59
chips> regArea(r1,4070,4050,4100,4150,0.1)
3010.25
```

CHANGES IN CIAO 3.3

The calculation of composite areas − those consisting of multiple shapes − has been improved in this release.

CHANGES IN CIAO 3.2

The CIAO 3.2 release includes the regExtent() function, which can be used to automatically calculate the limits used in the six−argument form of regArea().

Bugs

See the bugs page for the region library on the CIAO website for an up−to−date listing of known bugs.

See Also

modules
    region
region
    regarea, regextent, reginsideregion, regparse, regprintregion, regregionstring

The Chandra X−Ray Center (CXC) is operated for NASA by the Smithsonian Astrophysical Observatory.
60 Garden Street, Cambridge, MA 02138 USA.

URL:
http://cxc.harvard.edu/ciao3.4/regarea.html
Last modified: December 2006