

---

Subject: Re: Map\_image questions

Posted by [bowman](#) on Mon, 30 Aug 1993 20:34:34 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

In article <doetzl.746473023@rintintin.Colorado.EDU>,  
doetzl@rintintin.Colorado.EDU (Joe Doetzl) wrote:

>  
>  
> I have a question concerning the Idl routine map\_image. I have an image  
> of a 2d field, however the data is valid only over continents, and not  
> the sea. The data values for points over the sea are there they are  
> just not valid and are not flagged with any convenient missing data  
> flag. When I use map\_image to warp this image onto a specific map  
> projection the bad sea values are interpolated with the good continental  
> values thus causing the values on the continental outlines to be  
> compromised. Is there a way around this? Does anyone have a map\_image  
> like routine that will only interpolate a selected range of valid data  
> values? Is something like this possible/easy to write?

I have the same problem, and have solved it this way.  
The map\_image routine is in the userlib. Make your own copy  
(e.g., my\_map\_image.pro) and replace the calls to interpolate with your own  
interpolator that doesn't interpolate with the missing values.

Since map\_image is used only for 2-D images, I wrote the following simple  
interpolator that checks for missing values at any of the corners of  
the box.

I do not guarantee that it is bug free, and I have not had time to  
test it very thoroughly, but it works for me. Because of the additional  
IF tests, it is necessarily slower than map\_image. If this one doesn't  
work  
for you, you can always write your own to call from map\_image.

Bug reports welcome.

Ken Bowman

```
*****  
,  
FUNCTION MY_INTERPOLATE2, a, x, y, MISSING = missing  
  
;+  
; NAME:  
;   MY_INTERPOLATE2  
; PURPOSE:  
;   This function interpolates the array a to the coordinates (x,y)  
;   using bilinear interpolation. For use in MY_MAP_IMAGE.  
; CATEGORY:
```

```

; Interpolation utility.
; CALLING SEQUENCE:
; MY_INTERPOLATE2, a, x, y
; INPUTS:
; a = two-dimensional array to be interpolated
; x = vector of x-coordinates to interpolate to
; y = vector of y-coordinates to interpolate to
; KEYWORDS:
; missing = if any one of the four 'corners' of the box used to
; interpolate are equal to missing, then the output
; is set to missing. Should be BYTE type.
; OUTPUT:
; Array a interpolated to (x,y)
; RESTRICTIONS:
; None.
; COMMON BLOCKS:
; None.
; SIDE EFFECTS:
; None.
; PROCEDURE:
; Missing uses bilinear interpolation to evaluate a(x(i),y(i)).
;-

```

```

;t = systime(1)

```

```

sa = SIZE(a)
IF (sa(0) NE 2) THEN BEGIN
  PRINT, 'Error in PRO MY_INTERPOLATE2:'
  PRINT, ' Input array must have two dimensions'
  RETURN, undefined
ENDIF
ni = sa(1)
nj = sa(2)
type = sa(3)

IF (N_ELEMENTS(x) NE N_ELEMENTS(y)) THEN BEGIN
  PRINT, 'Error in PRO MY_INTERPOLATE2:'
  PRINT, ' The number of elements in x and y must be equal'
  RETURN, undefined
ENDIF

sx = SIZE(x)
nx = sx(1)
IF(sx(0) EQ 1) THEN ny = 1 ELSE ny = sx(2)
b = REPLICATE(missing, nx, ny)

i1 = FLOOR(x)
i2 = i1 + 1

```

```

j1 = FLOOR(y)
j2 = j1 + 1

in = WHERE((i1 GE 0) AND (i2 LE ni-1) AND $
           (j1 GE 0) AND (j2 LE nj-1) AND $
           (a(i1,j1) NE missing) AND $
           (a(i2,j1) NE missing) AND $
           (a(i1,j2) NE missing) AND $
           (a(i2,j2) NE missing), nin)

```

```

IF (nin GT 0) THEN BEGIN
  a1 = x(in) - i1(in)
  a2 = 1.0 - a1
  b1 = y(in) - j1(in)
  b2 = 1.0 - b1
  b(in) = a2*b2*a(i1(in),j1(in)) + $
          a1*b2*a(i2(in),j1(in)) + $
          a2*b1*a(i1(in),j2(in)) + $
          a1*b1*a(i2(in),j2(in))
ENDIF

```

```

;PRINT, 'Time = ', SYSTIME(1) - t

```

```

RETURN, b
END

```

```

-----
Dr. Kenneth P. Bowman          409-862-4060
Climate System Research Program      409-862-4132 fax
Department of Meteorology          bowman@csr.p.tamu.edu
Texas A&M University              PP-Glider
College Station, TX  77843-3150

```

---