
Subject: Re: using VOXEL_PROJ to obtain "thick slice"
Posted by [Paul Sorenson](#) on Sun, 02 Nov 2003 21:29:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

The CUTTING_PLANE keyword just lops off parts of the volume. It doesn't extract a skinny slice unless you put two cutting planes really, really close to each other.

Also, just to mention it, I like to use CREATE_VIEW to manipulate !P.T. Here is a wrapper that makes it more friendly:

```
pro cntr_view,arr,xr=xr,yr=yr,zr=zr,undo=undo,$
  ax=ax, az=az, winx=winx, winy=winy, zoom=zoom, _extra=e
;
; Procedure cntr_view. Establish a 3d view. This is a wrapper
; for Create_View. Cntr_view works just like create view, except:
;
; 1. Takes range keywords XR, YR and ZR. These are optional
;    alternatives to the XMIN, XMAX, etc. keywords. Given an XR
;    array, for example, cntr_view will "do the work you",
;    finding the min and max in the XR array, and then feeding
;    those values to create_view via create_view's XMIN and
;    XMAX keywords.
;
;    IDL> x=[.7, -8, 6, 9]
;    IDL> y=[-.5, 2, -6,3]
;    IDL> z=[1, 1, 5, 5]
;    IDL> erase
;    IDL> cntr_view, xr=x, yr=y, zr=z
;    IDL> plots, x, y, z
;    IDL> surface, bytarr(2,2), /nodata, /noerase
;
; 2. Takes an optional 3d array argument. If cntr_view
;    is passed one 3d array, the sizes of the
;    array are used to determine xmax, ymax, and
;    zmax, and xmin, ymin and zmin will be set to
;    zero. If Keywords such as XMIN, XMAX, XR, etc. are
;    passed with this argument, they override the
;    ranges implied by this argument.
;
; 3. Provides an UNDO keyword to return all relevant
;    system variables to their defaults.
;
; 4. Uses size of current window (!d.x_size and !d.y_size)
;    as defaults for "winx" and "winy" keywords.
;    (I have not tested this last feature for use
;    with non-windowing (hardcopy) devices.)
;
```

```
; Paul C. Sorenson
; September 1995
;
on_error, 2
```

```
if keyword_set(undo) then begin
!P.T3D=0
!P.Position=0
!P.Clip=0
!P.Region=0
!X.S=0
!X.Style=0
!X.Range=0
!X.Margin=[10,3]
!Y.S=0
!Y.Style=0
!Y.Range=0
!Y.Margin=[4,2]
!Z.S=0
!Z.Style=0
!Z.Range=0
!Z.Margin=0
return
end
```

```
xmin=0
xmax=1
ymin=0
ymax=1
zmin=0
zmax=1
```

```
if n_params() gt 0 then begin
  s = size(arr)
  if s(0) ne 3 then begin
    message, 'argument must be 3D array.'
  end
  xmax=s(1)-1
  ymax=s(2)-1
  zmax=s(3)-1
end
```

```
if (n_elements(xr) gt 0) then begin
  if (n_elements(xr) lt 2) then begin
    message, 'keyword XR takes an array of at least 2 elements.'
  end
  xmin = min(xr)
  xmax = max(xr)
```

```

end

if (n_elements(yr) gt 0) then begin
    if (n_elements(yr) lt 2) then begin
        message, 'keyword YR takes an array of at least 2 elements.'
    end
    ymin = min(yr)
    ymax = max(yr)
end

if (n_elements(zr) gt 0) then begin
    if (n_elements(zr) lt 2) then begin
        message, 'keyword ZR takes an array of at least 2 elements.'
    end
    zmin = min(zr)
    zmax = max(zr)
end

if xmin eq xmax then begin
    message, 'specified x-range is infinitesimal.'
end

if ymin eq ymax then begin
    message, 'specified y-range is infinitesimal.'
end

if zmin eq zmax then begin
    message, 'specified z-range is infinitesimal.'
end

if (n_elements(winx) eq 0) then begin
    winx = !d.x_size
end

if (n_elements(winy) eq 0) then begin
    winy = !d.y_size
end

if (n_elements(ax) eq 0) then ax = -60
if (n_elements(az) eq 0) then az = 30
if (n_elements(zoom) eq 0) then zoom = 1/sqrt(3)

create_view, xmin=xmin, ymin=ymin, zmin=zmin, $
    xmax=xmax, ymax=ymax, zmax=zmax, $
    winx=winx, winy=winy, ax=ax, az=az, $
    zoom=zoom, _extra=e

end

```

-Paul Sorenson

"Edward Graves" <edwardg@OCF.Berkeley.EDU> wrote in message
news:bnpoam\$17ef\$1@agate.berkeley.edu...

> Hi all,

>

> I spent the last few hours futzing around with VOXEL_PROJ, and have

> finally figured out how to get it to return a maximum intensity projection

> of my data for an oblique view (specified in terms of the transformation

> matrix !P.T). Looking at the rather paltry documentation for VOXEL_PROJ,

> i noticed that the default is for the function to perform "average

> intensity projection" when both the MAXIMUM_INTENSITY and RGO keywords

> are not set. In conjunction with the CUTTING_PLANE keyword, I was

> thinking this may be useful for obtaining a thick slice from an image

> volume (one in which a single voxel in the slice may encompass several

> voxels in the source dataset). As opposed to the slice obtained by

> interpolating using the coordinates of the desired slice, in which you

> obtain a trilinear interpolate of the intensities of the voxels bounding

> the coordinate, rather than an average of all the voxels bounded by the

> slice voxel.

>

> I was curious if anyone has tried this, or if i'm even interpreting the

> operation of VOXEL_PROJ correctly. Thanks in advance for any advice you

> may have,

>

>

>

>

>

>

>

> Ted

> graves@reyes.stanford.edu

>

Subject: Re: using VOXEL_PROJ to obtain "thick slice"

Posted by [Paul Sorenson](#) on Mon, 03 Nov 2003 00:00:56 GMT

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

Another thought: The averaging in an "average intensity projection" happens
after CUTTING_PLANES do their thing. It sounds like you might have been
thinking the reverse. Perhaps you want to extract a plane (a la SLICER) from
a SMOOTHed volume.

-Paul Sorenson

> "Edward Graves" <edwardg@OCF.Berkeley.EDU> wrote in message
> news:bnpoam\$17ef\$1@agate.berkeley.edu...
>> Hi all,
>>
>> I spent the last few hours futzing around with VOXEL_PROJ, and have
>> finally figured out how to get it to return a maximum intensity
projection
>> of my data for an oblique view (specified in terms of the transformation
>> matrix !P.T). Looking at the rather paltry documentation for
VOXEL_PROJ,
>> i noticed that the default is for the function to perform "average
>> intensity projection" when both the MAXIMUM_INTENSITY and RGBO keywords
>> are not set. In conjunction with the CUTTING_PLANE keyword, I was
>> thinking this may be useful for obtaining a thick slice from an image
>> volume (one in which a single voxel in the slice may encompass several
>> voxels in the source dataset). As opposed to the slice obtained by
>> interpolating using the coordinates of the desired slice, in which you
>> obtain a trilinear interpolate of the intensities of the voxels bounding
>> the coordinate, rather than an average of all the voxels bounded by the
>> slice voxel.
>>
>> I was curious if anyone has tried this, or if i'm even interpreting the
>> operation of VOXEL_PROJ correctly. Thanks in advance for any advice you
>> may have,
>>
>>
>>
>>
>>
>>
>>
>> Ted
>> graves@reyes.stanford.edu
>>
>
>
