Subject: Re: fast search
Posted by greg michael on Wed, 18 Oct 2006 15:25:04 GMT
View Forum Message <> Reply to Message

All but one indestructible loop removed. Wish I had a use for this
program now... hope it's useful to you, Marie!

```
pro splitsearch3,p,dist
 ;recursively splits the search volume into n_split^3 subvolumes. When
 ;there are fewer than 'threshold' points
 ;in a subvolume, checks for matches the brute force way - every point
 ;against every other.

 n_split=4    ;1-D cutting factor (for 3, cube is cut into 3x3x3=27
subvolumes)
 threshold=75 ;no. of points to start pairwise comparison
 n=n_elements(p)

 if n gt threshold then begin
  mxx=max(p.x,min=mnx)
  mxy=max(p.y,min=mny)
  mxz=max(p.z,min=mnz)
  bx=fix((p.x-mnx)/(mxx-mnx)*n_split)<(n_split-1) ;< to ensure max
element not in new bin
  by=fix((p.y-mny)/(mxy-mny)*n_split)<(n_split-1)
  bz=fix((p.z-mnz)/(mxz-mnz)*n_split)<(n_split-1)
  b=bx+by*n_split+bz*n_split^2
  h=histogram(b,min=0,reverse_indices=ri)

  for i=0,n_elements(h)-1 do begin
   if ri[i] ne ri[i+1] then begin
    q=[ri[ri[i]:ri[i+1]-1]]
    if n_elements(q) ge 2 then splitsearch3,p[q],dist ;splitsearch
again, if enough to compare
   endif
  endfor

 endif else begin
  q1=rebin(indgen(n),n,n) ;set up indices for pairwise matching
  q2=transpose(q1)
   d=sqrt((p[q1].x-p[q2].x)^2+(p[q1].y-p[q2].y)^2+(p[q1].z-p[q2 ].z)^2)
;calculate pair distances
  i=where((d le dist) and (q1 gt q2)) ;select close neigbours (q1>q2 to
avoid reverse pairs)
  if i[0] ne -1 then print,p[q1[i]],p[q2[i]] ;print the neighbours, if
found
 endelse
end
```

```
IDL> p=ss_makegalaxies(5e6)
IDL> splitsearch3,p,.02
{     915.876      843.515      319.991}{     915.861      843.528
  319.994}
```