Subject: Re: fast search
Posted by greg michael on Tue, 17 Oct 2006 18:25:39 GMT
View Forum Message <> Reply to Message

And here's a proper IDL version that uses histograms and the pairwise comparisons that David likes... still got three loops, though - have to think some more to get rid of those.

```
function ss_MakeGalaxies,n
 p=replicate({x:0.,y:0.,z:0.},n)
 seed=0
 p.x=randomu(seed,n)*1000.
 p.y=randomu(seed,n)*1000.
 p.z=randomu(seed,n)*1000.
 return,p
end

pro splitsearch2,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=3    ;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
   hx=histogram(p.x,nbins=n_split+1,locations=x,reverse_indices =rx) ;1-D
binning
   hy=histogram(p.y,nbins=n_split+1,locations=y,reverse_indices =ry)
   hz=histogram(p.z,nbins=n_split+1,locations=z,reverse_indices =rz)

  for i=0,n_split-1 do begin
   for j=0,n_split-1 do begin
    for k=0,n_split-1 do begin
     qx=[rx[rx[i]:rx[i+1]-1]] ;indices from 1-D x bins
     qy=[ry[ry[j]:ry[j+1]-1]]
     qz=[rz[rz[k]:rz[k+1]-1]]
     q=where(histogram([qx,qy,qz],min=0) eq 3) ;indices from 3-D bins
(i.e. where all of x,y,z
                                    ;lie inside subvolume

     if n_elements(q) ge 2 then splitsearch2,p[q],dist ;splitsearch
again, if enough to compare
    endfor
   endfor
```

```
  endfor

 endif else begin
  q1=rebin(indgen(n),n,n) ;set up indices for pairwise comparison
  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 neighbours (q1>q2
to avoid reverse pairs and q1=q2)
  if i[0] ne -1 then print,p[q1[i]],p[q2[i]] ;print the neighbours, if
found
 endelse
end

IDL> p=ss_MakeGalaxies(1e6)
IDL> splitsearch2,p,.1
{     98.4014     461.355     673.278}{     98.4379     461.400
  673.197}
{     190.002     444.629     924.567}{     189.966     444.613
  924.605}
{     254.499     823.541     494.128}{     254.492     823.492
  494.062}
{     638.107     100.159     240.530}{     638.083     100.108
  240.606}
{     720.787     820.032     405.215}{     720.754     820.070
  405.199}

Greg
```