> The stupid solution is (sorry for the very very ugly code):
>
> ---
>   for i = 0l, n2 - 1 do begin
>   quad = (x2[i] - x1)^2 + (y2[i]- y1)^2
>   for j=0, 3 do begin
>   minquad = min(quad, p)
>   if N_ELEMENTS(p) gt 1 then p = p[0] ; it happens.....
>   out[j,i] = p
>   quad[p] = max(quad) * 2. ;dummy large distance
>   endfor
>   endfor
> ---
>
> But I just cannot find a cleverer solution with triangulation...
> Someone clever than me to help ?
>
> Thanks a lot!
>
> Fabz

While there is certainly a better solution, you can start by removing
the j loop, and use sort(quad) instead of min() ... you can then take
the first 4 index.
Jean