Plane Extraction from Point Cloud Data

Robert B. Fisher
School of Informatics
University of Edinburgh

Planar Segmentation Algorithm

Range image *versus* point clouds

Row×Column image representation

- Obvious neighbour relations
- Easier region growing algorithms

3D Point Clouds

- ? relations in R^3
- Good data structures can help with neighbour connections

Segmenting range image into planar regions:
Use region growing algorithm

? Detection Main Algorithm

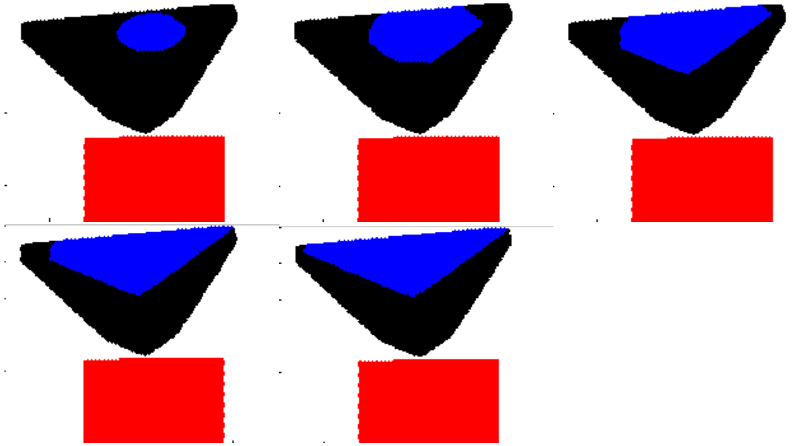
```
% find surface patches
[NPts,W] = size(R);
planelist = zeros(20,4);
foundcount=0;
while notdone

  % select small local surface patch from remaining points
  [oldlist,plane] = select_patch(remaining);

  % grow patch
  stillgrowing = 1;
  while stillgrowing

    % find neighbouring points that lie in plane
```



```
    stillgrowing = 0;
    [newlist,remaining] = getallpoints(plane,oldlist,
                                        remaining,NPts);
    [NewL,W] = size(newlist);
    [OldL,W] = size(oldlist);
    if NewL > OldL + 50
      % refit plane
      [newplane,fit] = fitplane(newlist);
      if fit > 0.04*NewL % fit going bad - stop growing
        break
      end
      stillgrowing = 1;
      foundcount = foundcount+1;
      planelist(foundcount,:) = newplane';
      oldlist = newlist;
      plane = newplane;
```

Region ? Principles

Given a planar region formed from points S with equation $\vec{n}'\vec{x} + d = 0$, and a new point $\vec{y}$, add $\vec{y}$ to S if:
1) $\mid \vec{n}'\vec{y}+d \mid < \tau_p$ and 2) there is a point $\vec{z}$ in S such that $\| \vec{y}-\vec{z} \| < \tau_n$.



Plane ?

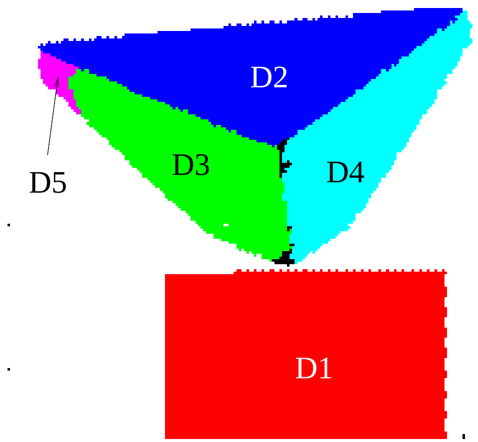
Given a set of datapoints $\{\vec{x}_i\}$, find the $\vec{n}$ and d that best fit $\vec{n}'\vec{x}_i + d = 0$ for all i.

Extend data: $\vec{y}_i = [\vec{x}_i, 1]$
Extend parameters: $\vec{p} = [\vec{n}, d]$
Plane equation is now: $\vec{y}_i'\vec{p} = 0$

Least squared error:
$\sum_i (\vec{y}_i'\vec{p})^2 = \sum_i \vec{p}'\vec{y}_i\vec{y}_i'\vec{p} = \vec{p}'(\sum_i \vec{y}_i\vec{y}_i')\vec{p} = \vec{p}'M\vec{p}$

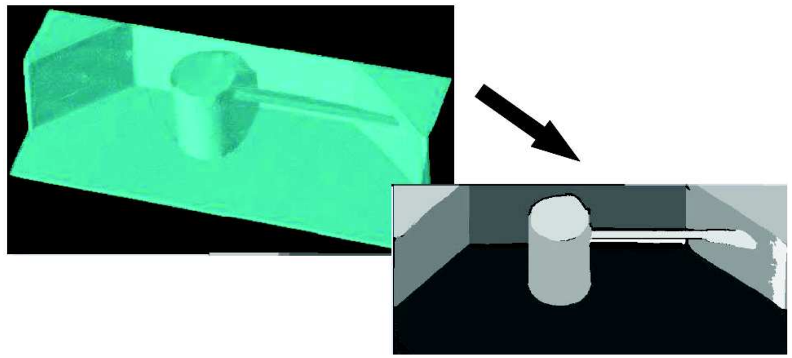
Eigenvector of smallest eigenvalue of M is desired parameter vector, provided eigenvalue is small.

Full ?



Could get 4 planes by parameter adjustment, but 5 means more data for matching stage

Extensions



Extend fitting to additional surface types:
? spheres, etc

Allows recognition of more complex objects

What We Have Learned

- A [?] algorithm
- A least squares algorithm for plane parameter estimation
- Some idea of how well it works on relatively clean data