CS664 Lecture #17: Robust statistics

Other M-estimators

- There are a bunch of other choices for ρ with relatively similar behaviors
  - Different models for how inlier noise behaves, and different shapes
- Important idea is the influence function: $\boxed{\frac{d\rho}{dr}}$
  - Controls how much a point pulls the line as a function of its distance from the line
  - "Redescending" M-estimators have influence functions that go to zero in the limit
    - In other words, ρ itself flattens out

Breakdown point

- Another important concept concerns how much “bad” data a method can ignore
  - For M-estimation, or LS, the answer is none
    - Since any bad point can move the line of best fit by at least a little bit
- Obviously you can’t in general tolerate more than 50% outliers, or the outliers become the inliers
  - Is there an optimal method?
    - Note: you can tolerate >50% outliers if you are willing to make some assumption (MINPRAN)

Least median squares

- Instead of changing ρ, let's compute the median of the set $\rho(r_i)$
  - Doesn't really matter what ρ we use
    - Any monotonic symmetric choice of ρ will obviously give the same answer
- We can now ignore up to 50% outliers
- Futhermore, there is no scale parameter
  - In effect, dynamically estimated from the data
- Nice geometric intuition also

Computing a robust fit

- It's possible to perform M-estimation fairly efficiently using a variant of least squares
- Think of Ax, where A is a matrix and x is a vector, as a linear combination of the columns of A, weighted by elements of x
  - If we consider all possible choices of x we span a subspace. The solution to $Ax = y$ is the "coordinates" of y in terms of the columns of A
  - What if y isn't in the subspace? We can ask for the point in the subspace that is as close as possible to y (the least squares fit)

Solving least squares

- The least squares solution to $Ax = y$ is
$$\hat{x} = \arg\min_x \|y - Ax\|$$
- An elegant result, due to Gauss, is that the solution to this is the pseudoinverse
$$(A^t A)^{-1} A^t y$$
  - Easy to re-derive: A^tA is square!
- If we weight each residual by w_i we get
$$\arg\min_x \|W(y - Ax)\| = (A^t W^2 A)^{-1} A^t W y$$
  - Here, W is a diagonal matrix of w_i's

Iterative reweighted least squares

- IRLS algorithm
  - Start with all weights being 1
  - Compute least squares fit and residuals
  - Adjust W to reduce the weighting of the large residuals
  - Re-fit and repeat

Non-parametric approaches

- Non-parametric model fitting usually relies on ranks (i.e., ordering information)
- Given our data points (x_i, y_i) we can replace the values by the ranks among all the x_i or y_i
  - Write this as $(\tilde{x}_i, \tilde{y}_i)$
  - Now based on the ranked data we can compute, for instance, the L_2 distance

$$r_{\text{spr}} = \sum_i \left(\tilde{x}_i - \tilde{y}_i\right)^2$$

Spearman and ranks

- r_{spr} is Spearman's correlation coefficient
  - Outliers lead to small change in rank
  - Distance is 0 if y_i is a monotonic function of x_i
  - Tolerates changes in image gain and bias
- Too slow to do locally
  - Each local window must be sorted!
  - Can't really re-use this computation
- Approximation via rank transform
  - What is the rank of the center pixel, with respect to the rest of the window?

Kendall and census

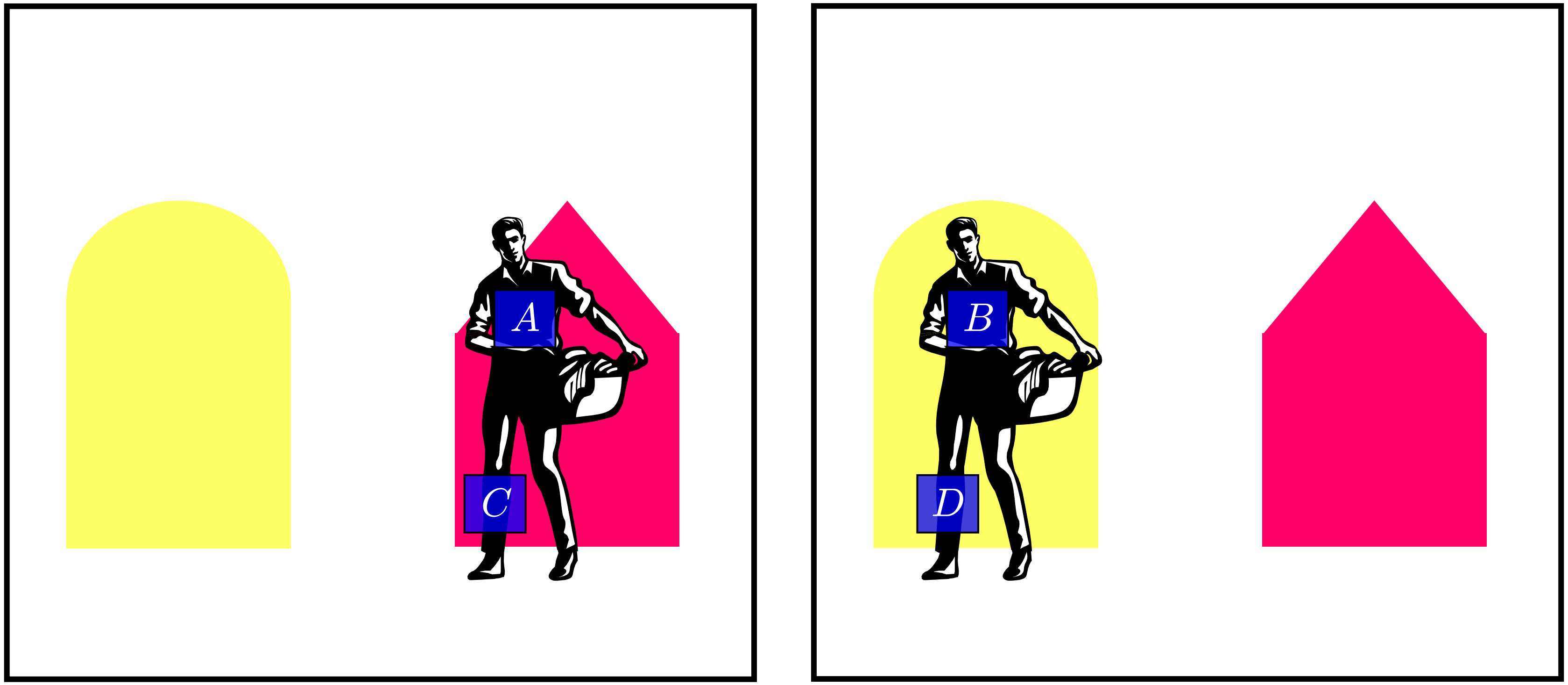
- Alternative: Kendall's τ
  - Consider a pair of points $(x_i,y_i),(x_j,y_j)$
  - We say this pair is concordant if

$$[(x_i > x_j) \wedge (y_i > y_j)]$$

    - Or if we replace ">" with "<"
  - τ counts the number of concordant pairs
- Local version: compare the center pixel with all other pixels (view as a bit string)
  - Can compare windows via Hamming distance
  - Ideal for hardware (Tyzx)

Robustness and discontinuities

- Why do we get outliers in vision?
  - Occasionally you will see them in stereo due to camera sampling artifacts
    - "Mixed pixels" with non-integer disparity
- They happen a lot in vision in general
  - Usually an image has a lot of very different phenomena going on in it
- The geometry of stereo causes them to occur at discontinuities
  - This is (partly) why discontinuities are so hard



Left image *Right image*

- Window A is very similar to B
  - Gaussian noise is the only difference
- Window C is not very similar to D
  - Red versus yellow pixels in background: outliers!

Local vs. global stereo

- Two basic classes of algorithm
  - Compute answer independently at each pixel
    - Local methods: compare windows
  - Search the space of possible disparity maps
    - Global methods: energy minimization
- Local methods are fast, and robust statistics can make them a bit better
- Global methods give much better answers
  - Now that we can minimize the energy fast