Patch data structure

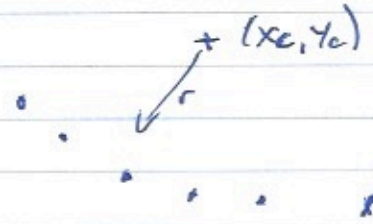
id: unique identifier (UUID?)
primitive: <~~circle~~, rectangle, line, irregular, ~~cone~~>
color: hue or hue and saturation value?
points: point cloud in X, Y, organized with nearest neighbors and CCW order
primitive des: ~~circle — center x, center y, radius, height~~ subset of cone
rect - SW X, SW Y, NE X, NE Y, height (5)
line - end1 X, end1 Y, end2 X, end2 Y, height (5)
cone - center x, center y, bottom rad, top rad, height (5)

Functions:

insert point into cloud → Knn? check against primitive?
infer primitive
compare point for membership
compare point for proximity
plot patch on given figure
compare patch to patch (by primitive metrics)

infer primitive
↳ circle → transform to polar coords, do a linear regression, determine quality of fit

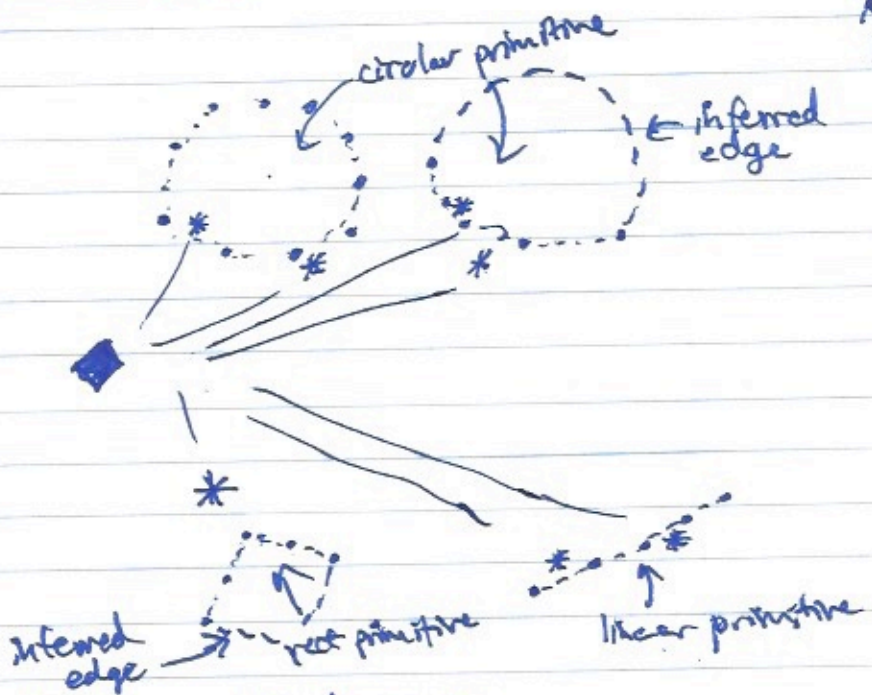
$$(x-x_c, y-y_c) = \left(r_{bottom} - \frac{(r_{bottom} - r_{top})}{height} z\right)(\cos\theta, \sin\theta)$$



$$r^2 = (x-x_c)^2 + (y-y_c)^2$$

Point to "patches" or areas

Noon
1:45



For solid objects, will only ever get surface points:
→ can use descriptive info for confirmation of association
→ e.g. orange barrel or grey concrete barrier
→ can also use z-coordinates (height) to avoid mismatches
↳ a layered approach that works by slices?
↳ some layers are "natural" such as surface level, mid-barrier or barrel, chest height?

Goals:
ID patches, confirm agreement w/ patch or quality
↳ easy statistics
what does a series of point hits mean?