Go over Program

1. Get command line input and initialize all variables
2. Convert some input to lowercase (easier to parse)
3. Load in the spelling dictionaries
4. Load the entire image / page / pages
5. Loop through each page(s)
6. Make the image greyscale
7. Scale the image
8. Rotate / Clean image
9. Find segments (Remove tables / text leave diagrams) **What are clusters and what does prune clusters do?**
10. Create a box (subimage) of each diagram
11. Infer font size from resolution
12. Copy diagram from entire image into a box
13. Thin the image
14. Rasterize the image POTRACE
15. Find atoms (Initially everything is an atom, but especially seem to exist on handles)
16. Find bonds (1 to 4 atoms have been pushed for every possible atom, if a corner, the direction changes between control points or if **the current atom is farther from the last atom on the curve than the next, what does this mean? Does it have to do with rings?**
17. Create bonds where atoms exist.
18. Detect characters
19. Make sure a curve does not overlap another +- potrace **Not to be confused with a ring? Maybe font size is determining factor.**
20. Look in area where an atom might exist
21. Clamp a bounding box around a character
22. Make sure the box is of reasonable size
23. Pass it off to character recognition libraries to do some magic!
24. Find average bond length (Sort all bonds and take 75th percentile length) **What constitutes a bond, only those with length above the 75th percentile?**
25. Continue to refine atoms / bonds / characters (find_plus_minus, find_small_bonds, find_old_aromatic_bonds, remove_disconnected_atoms, collapse_atoms, remove_zero_bonds, find_wavy_bonds, find_fused_characters, double/triple_bonds, dashed_bonds, remove_small_bonds, **find_numbers (how is this different than chars?)**, fix_one_sided_bonds, clean_unrecognized_characters, find_wedge_bonds, assemble_labels, remove_disconnected_atoms, collapse_atoms, remove_zero_bonds, **collapse**_double_bonds **(what does collapse mean?)**, **extend_terminal_bond_to_bonds (spaced by character?)**, find_up_down_bonds
26. Update array of confidence
27. Find / associate fragments
28. Construct structure

29. Create name string

--Why does the image, after being thinned, look anti-aliased?
I am not quite sure what do you mean by "look anti-aliased". The image may looked smudged after **anisotropic smoothing**, but thinning has nothing to do with it.

- Where / what does anisotropic smoothing and most importantly why?
- If image is thinned to a pixel thick why do some areas appear > 1 pixel wide. How does this affect wedge bonds?

--What does the thickness variable in osra_process_image and skeletize do?
It is an average thickness of a bond. Actually, I believe the value from "skeletize" is not used anymore, thickness is measured again in find_wedge_bonds. It is used later on as a threshold, similar to average bond length, to judge when things should be kept together or separate.

- Related to the anisotopic smoothing question

Questions for Next time:

--What does lbond_t mean? (Pairs of characters used for constructing atomic labels)

--How are labels associated to atoms? (See extend_atom label zero_atom)

--Inside of the label struct, what is the vector of character indices?

--Generally, are bond lengths relatively the same size?

--What is a fragment?

Begin Igor Shvartser's Notes

Importance of anisotropic smoothing:
    minor part
    allows a jagged line to be smoothed

Thinning algorithm:
    affects wedge bonds
superatom maps label to smile
spelling dictionary
superatom - a functional group

greyscale function - highest intensity of rgb taken
Clusters - handle multiple structures with text,
prune - gets rid of clusters that are irrelevant?
connected component - group of pixels that are touching one another (line, bond, etc)
Clamping when making boxes -- @ making boxes, we know where molecules are…

Parsing the graph that POTRACE gives is the biggest chunk of the program.

Generate control points after calling POTRACE

Going around each bond twice, since potrace will wrap around .. this avoids have extraneous points
Finding characters regardless of where they are
Characters found by finding POTRACE paths that are reasonable size.

Avg bond length used to generate a threshold to see how far bonds are
Measuring tool for later in the program

parentheses = bracket / round bracket
vs. square bracket
paren addition may have to be close to character recognition
(something exists for brackets, but is commented)

Useful to look at point density? YUP.
Data set on OSRA website to help test
Zero bonds -- bonds that have zero length

End Igor Shvartser Notes