Semi-Automatic Reconstruction of Cross-Cut Shredded Documents
Zachary Daniels (zad309@lehigh.edu), Mentor: Haroon Idrees (haroon@eecs.ucf.edu)

1 – Problem
- Cross-Cut Document Reconstruction:
  - Build bigram and trigram dictionaries
  - Grow reconstruction, pruning bad matches using oracle
- Approach
  - Build ‘oracle’: substitute for human input
  - Extract costs for chad matching and alignment
  - Combine chad costs and rank potential chad matches
  - Grow reconstruction, pruning bad matches using oracle
- System Architecture

2 – Preprocessing
- Segment chads from background
- For each chad:
  - Orient
  - Find bounding box
- Extract:
  - binary mask, derivative magnitude, derivative orientation, boundary, blurred boundary, Hough transform
- Segment foreground, background, and lines
- Cluster text and find top lines and baselines
- Annotate character information
- Build bigram and trigram dictionaries

3 – Oracle
- Binarize, smooth, and clean original document
- For each chad:
  - Binarize and scale
  - Correlate across solution image
  - Find approximately correct location
  - Correlate binary masks of remaining chads to fill gaps
  - Adjust each chad for pixel-level precision

4 – Optical Character Recognition
- OCRopus Engine
  - Segments text lines
  - Binarizes text
  - Performs character recognition
  - Accurate for chads with machine-printed text
  - Scanned pages from Google Books
  - Synthetically shredded with added noise and deformities
  - Each chad cleaned with Fred Weinhaus’ textcleaner script for ImageMagick

5 – Costs
- Shape Correlation ($C_s$)
- Line Alignment ($C_l$)
- Overlap ($C_o$)
- Gap ($C_g$)
- Text Alignment ($C_t$)
- Histogram of Overlap Sizes ($C_{o\text{ hist}}$)
- Histogram of Gap Sizes ($C_{g\text{ hist}}$)
- Character Combination ($C_c$)
- Text Line Overlap ($C_{l\text{ overlap}}$)
- Text Line Alignment ($C_{l\text{ align}}$)
- Bigram Lookup ($C_b$)
- Trigram Lookup ($C_t$)

6 – Reconstruction
- Pairwise Matching
  - Calculate pairwise costs:
    - $C_i = 0.01 C_s + 0.5 C_l + 0.0005 C_o + 0.0005 C_g + C_t$
  - Sort matches in ascending order
- Tree-Based Expansion
  - Find the top n pairwise matches where conditions hold:
    - One chad in each pair must appear in original solution
    - Overlap cost must remain below specified threshold
  - Match not discarded in past
  - Standardize costs between 0 and 1
  - Re-sort potential matches in ascending order:
    - $C_i = 3 C_s + 2.5 C_l + 1.5 C_o + 1.5 C_g + 0.5 C_t + 0.5 C_c + 0.5 C_b + 0.5 C_t$
  - Find first correct match and discard all others

7 – Results

8 – Future Work
- Color-Based Costs
- Adaptive Weighting of Costs
- Dictionary and Semantic Costs

9 – Acknowledgements
- This work is funded in part by NSF Grant # 1156990