DrawBuddy: virtual whiteboard to vectorize *black & white line drawings* that can be *modified by the user* and *sent to peers*

Lisa Mishra, Denise Yang, Ronald Gonzalez

Use Case Requirements

Smooth user experience:

- Latency to render simple diagram: < 2min
- Latency to modify diagram: < 100ms
- Accuracy: 9/10 average based on polling users

Accessibility:

- Writing utensil: 0.4 - 1.0 mm, black ink
- Capture distance: 1-3 feet
- Paper: 8.5” x 11” white printer paper
- 720p Camera

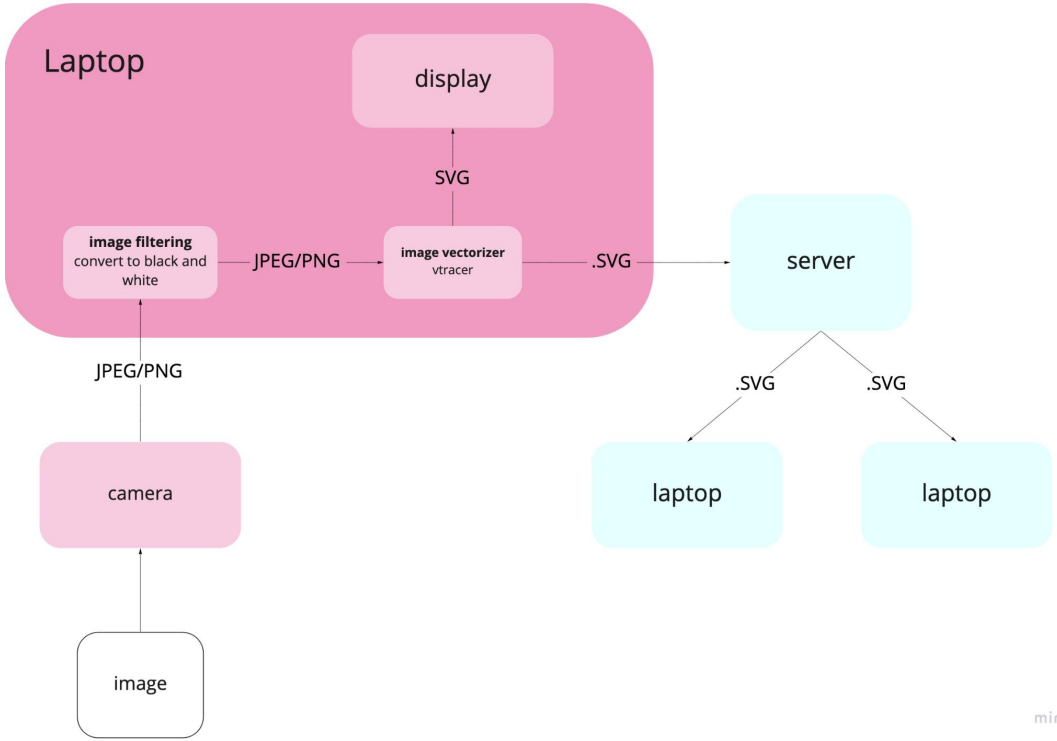
Communication:

- Connecting to communication server
- Sending message (< 500ms)
- Support at least 5 users in a shared session

Solution Approach

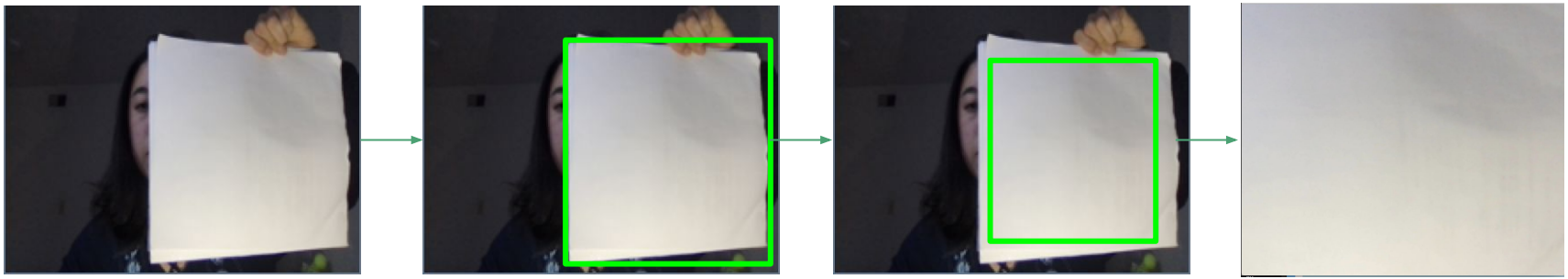
MVP: Vectorizes *black & white line drawings* that can be *modified by the user* and *displayed to peers*

- Capture image
- Apply mask to obtain paper
- Vectorize image
- Render
- Allow for translations and scaling
- Broadcast rendered image to connected users



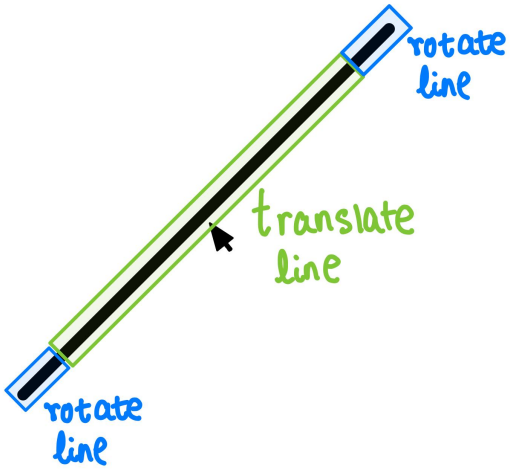
Final Solution - Vectorization

- Capture image
- Image Processing:
  - convert to black and white
  - Apply morphological erosion and dilation
- Find largest contour to crop using OpenCV
- Convert to SVG via Vtracer



Final Solution - GUI

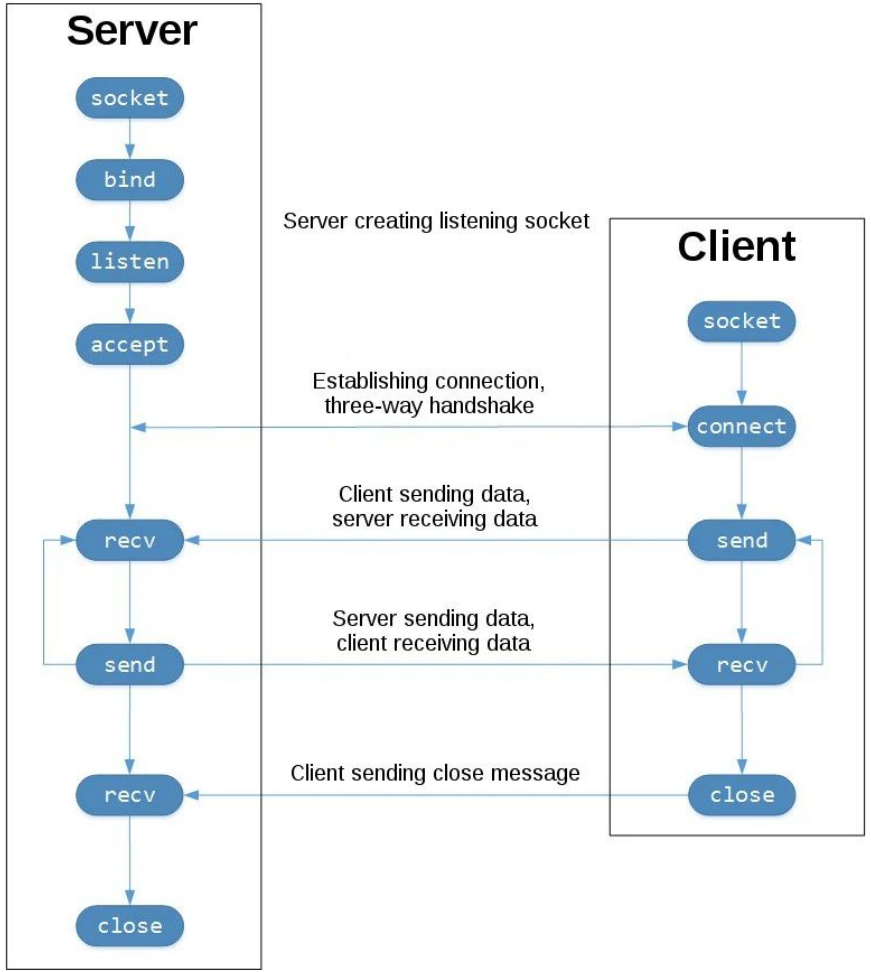
- GUI whiteboard can translate, rotate, and scale lines
- Click and drag line to translate it
- Mouse click x and y points determine whether the person is trying to rotate or scale the line
    - Any clicks within the green boundary will result in translation
    - Any clicks within the blue boundaries will result in rotation or scaling



Demo - Rotating and Scaling

Final Solution - Communication

- We used Python Sockets for our communications
- Allows us to have one person host the main server
- Everyone else has to connect to them via an access code
- The access code is the PORT number generated randomly when someone starts the room



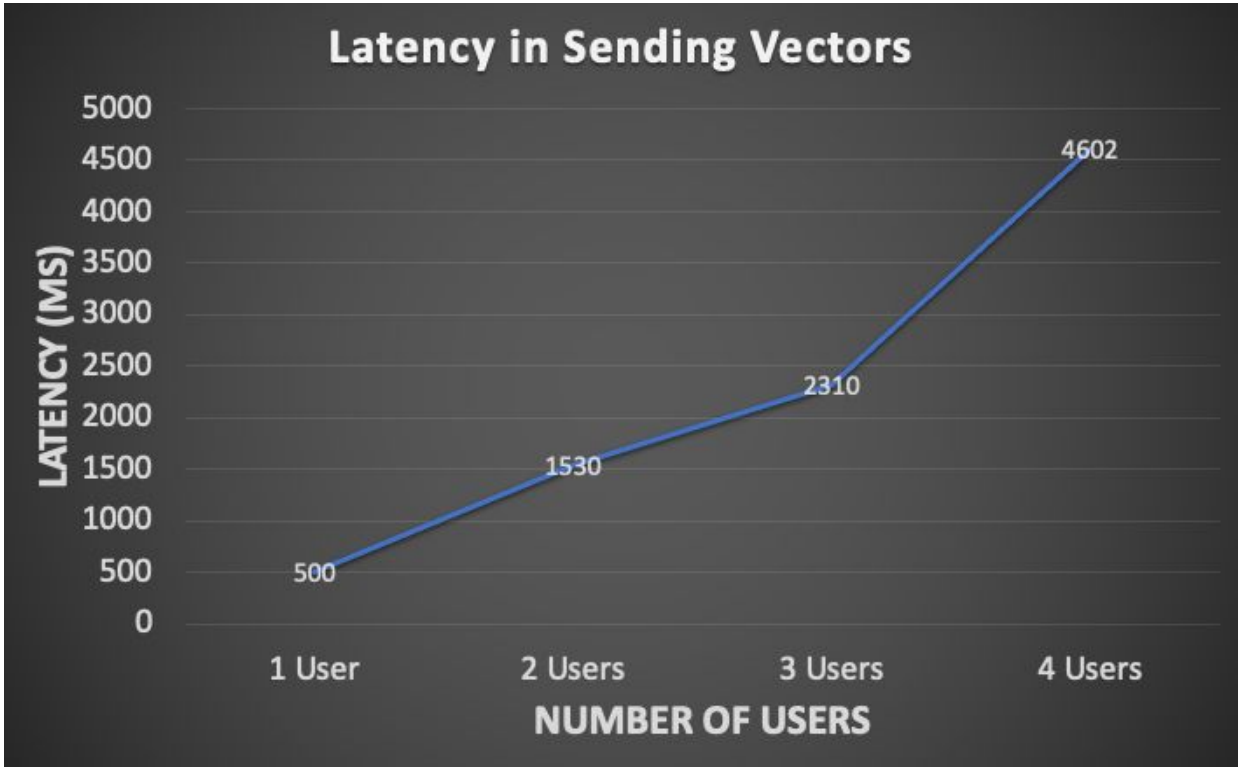
Testing & Validation Methods

Task	Requirement	Testing Method
Vectorization Code	<120s	Measure time from captured image to outputting onto the whiteboard
Translation and Rotation	<100ms	Measure how long it takes from selecting a line to moving it around
Communication	<500ms	Measure time it takes to send a compressed and encoded .svg file from one user to all other users
Qualitative Result	9/10 Score	Ask users how well the results reflect what the users originally intended to draw on a scale of 1-10

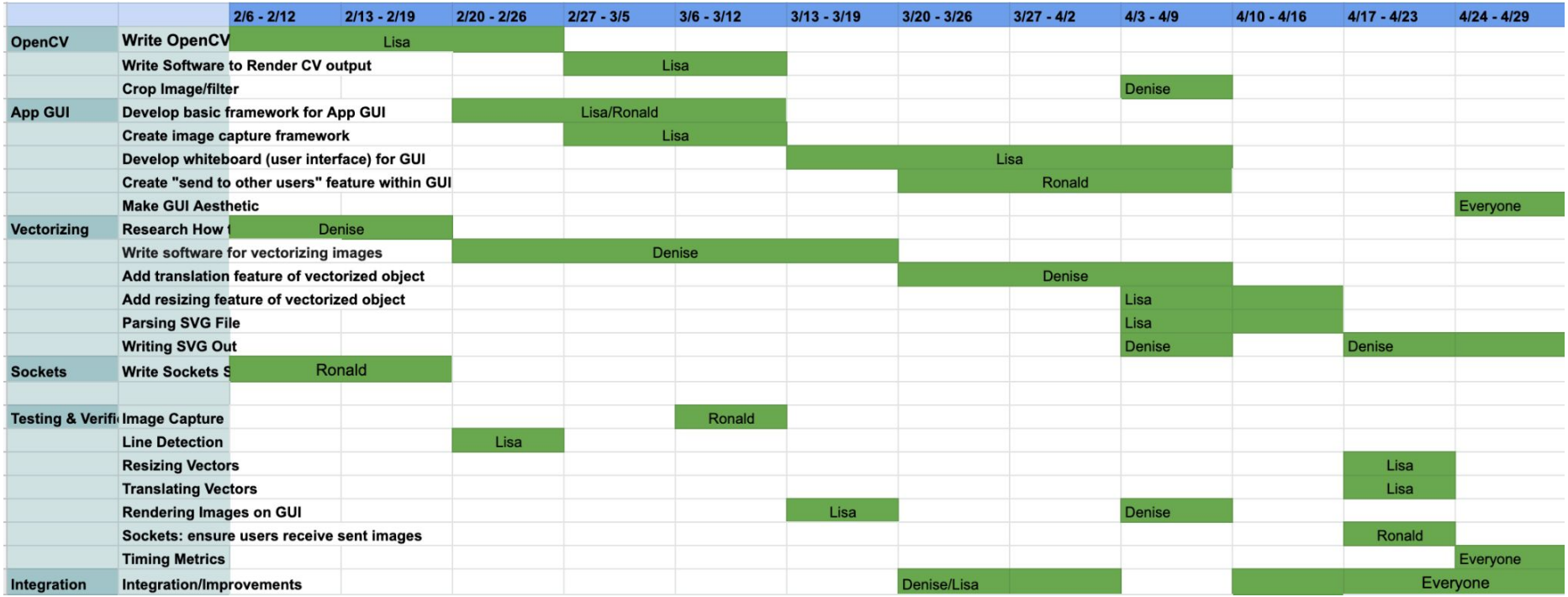
Results

Task	Result	Conclusion
Vectorization Time	100.5 ms	Success, 1200 times faster than expected
Line Modification Time	15 ms	Success, far faster than expected
Communication Time	Varied	Success/Failure, more testing should be done here
User Rating	5.3/10	Failure, we sped up vectorization time at the cost of accuracy

Results Communication



Schedule


2/6 - 2/12
2/13 - 2/19
2/20 - 2/26
2/27 - 3/5
3/6 - 3/12
3/13 - 3/19
3/20 - 3/26
3/27 - 4/2
4/3 - 4/9
4/10 - 4/16
4/17 - 4/23
4/24 - 4/29
OpenCV
Write OpenCV
Lisa
Write Software to Render CV output
Lisa
Crop Image/filter
Denise
App GUI
Develop basic framework for App GUI
Lisa/Ronald
Create image capture framework
Lisa
Develop whiteboard (user interface) for GUI
Lisa
Create "send to other users" feature within GUI
Ronald
Make GUI Aesthetic
Everyone
Vectorizing
Research How
Denise
Write software for vectorizing images
Denise
Add translation feature of vectorized object
Denise
Add resizing feature of vectorized object
Lisa
Parsing SVG File
Lisa
Writing SVG Out
Denise
Denise
Sockets
Write Sockets S
Ronald
Testing & Verifi
Image Capture
Ronald
Line Detection
Lisa
Resizing Vectors
Lisa
Translating Vectors
Lisa
Rendering Images on GUI
Lisa
Denise
Sockets: ensure users receive sent images
Ronald
Timing Metrics
Everyone
Integration
Integration/Improvements
Denise/Lisa
Everyone

What We Have Learned

- The faculty was correct, integrating takes time and dedication from all members of the team.
- The project taught me more about how to split up tasks in accordance with people’s strengths, and also to make time for integration!
- This project has helped me gain deeper understanding of how to ideate a product i.e. the process of analyzing what features are needed to satisfy the users and how that shapes the solution