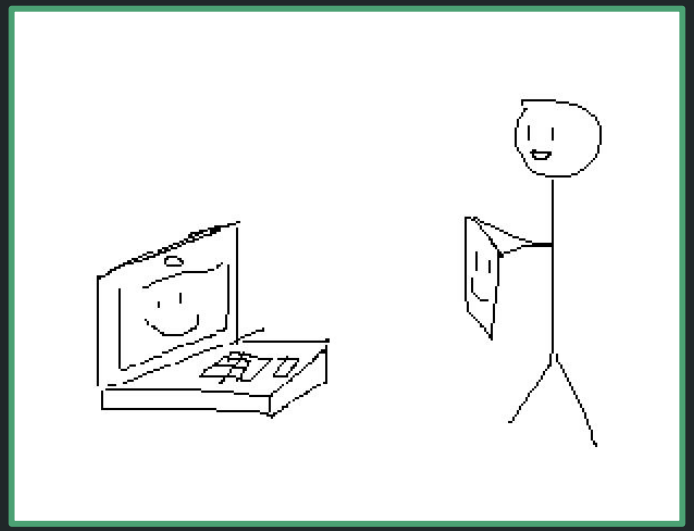
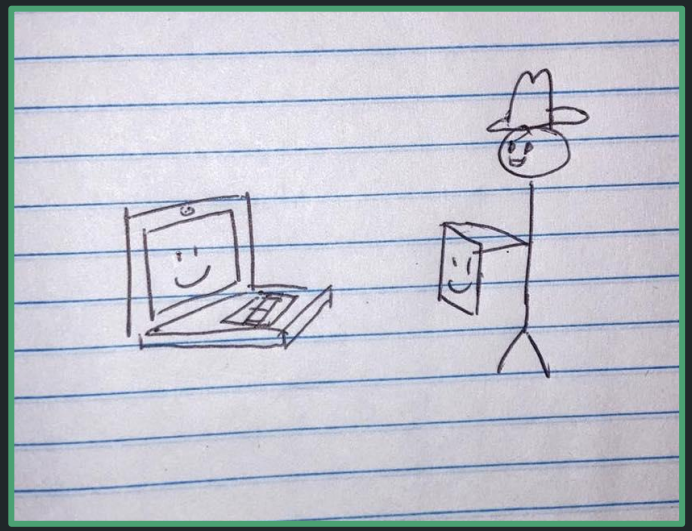
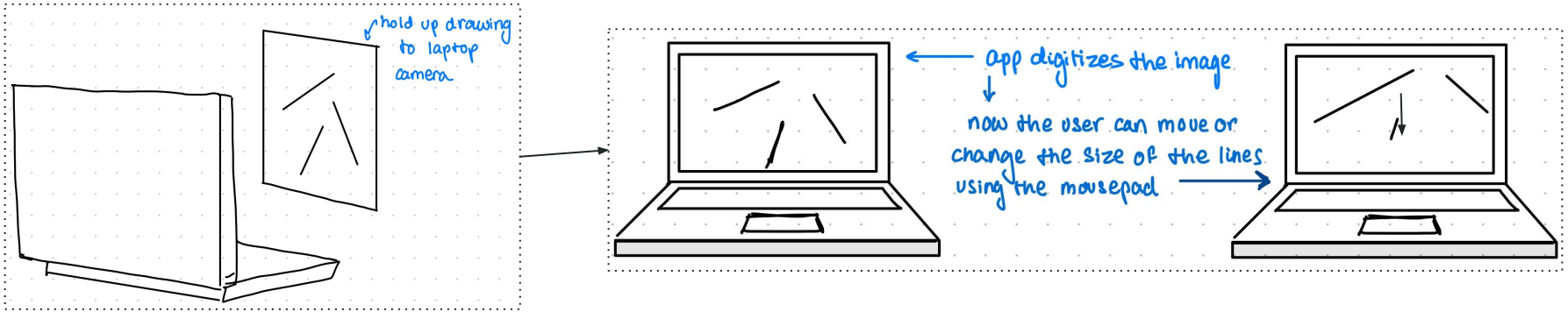




DrawBuddy: A more accessible, and natural platform for virtual collaboration

Use Case: Academia

- 84% of all undergraduates experienced some/all of their classes moved to to online only due to COVID (Cameron)
- Goal: Vectorize *black & white line drawings* that can be *modified by the user* and *displayed to peers*
- ECE domains: Software & Signals



Cameron, Margaux, et al. NCES, 2021, *2019–20 National Postsecondary Student Aid Study (NPSAS:20): First Look at the Impact of the Coronavirus (COVID-19) Pandemic on Undergraduate Student Enrollment, Housing, and Finances (Preliminary Data)*, https://nces.ed.gov/whatsnew/press_releases/06_16_2021.asp. Accessed 6 Feb. 2022.

Use Case Requirements

Smooth user experience:

- Latency to render simple diagram (< 50 primitives): < 2min
- Latency to modify diagram: < 100ms (Dabrowski)
  - Based on effects of latencies for Quality of Experience for gaming
- Accuracy: 9/10 average based on polling users

Dabrowski, Robert, et al. WORCESTER POLYTECHNIC INSTITUTE, 2014, *The Effects of Latency on Player Performance and Experience in a Cloud Gaming System*, https://web.wpi.edu/Pubs/E-project/Available/E-project-050514-142618/unrestricted/The_Effects_of_Latency_on_Player_Performane_and_Experience_in_a_Cloud_Gaming_System.pdf. Accessed 6 Feb. 2022.

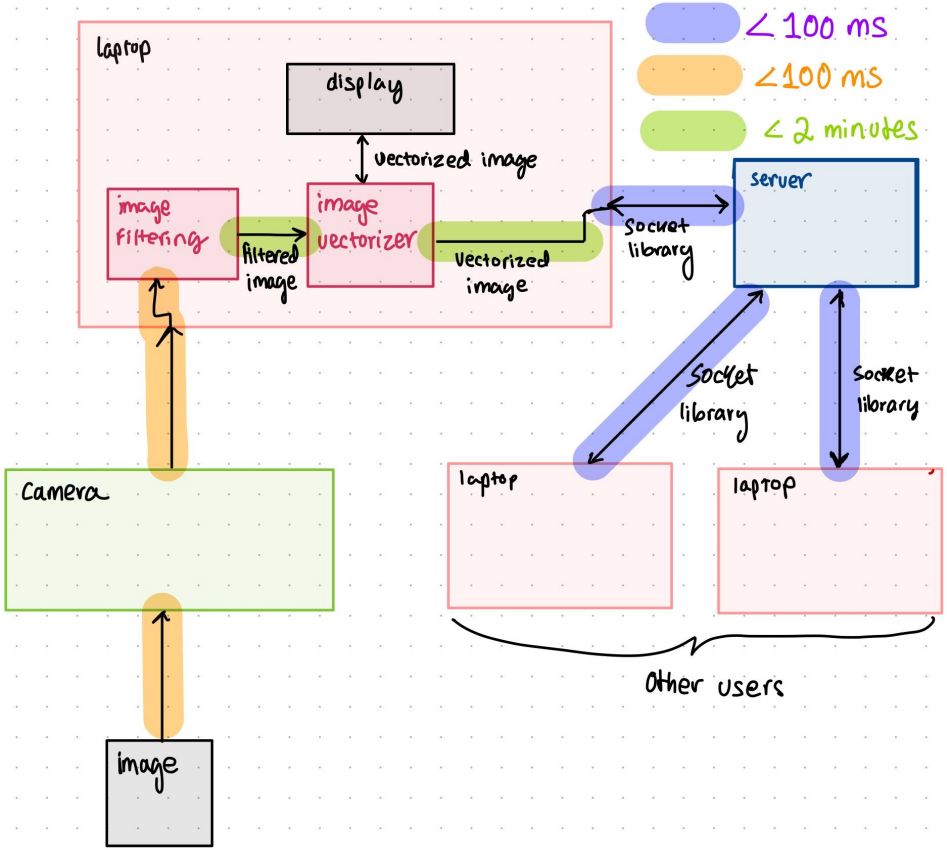
Use Case Requirements

Accessibility:

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

Use Case Requirements (Communications)

- Connecting to communication server
  - Almost negligible
  - As long as the delay to join isn't more than one or two seconds
- Sending message (< 100ms)
- Support at least 5 users in a shared session



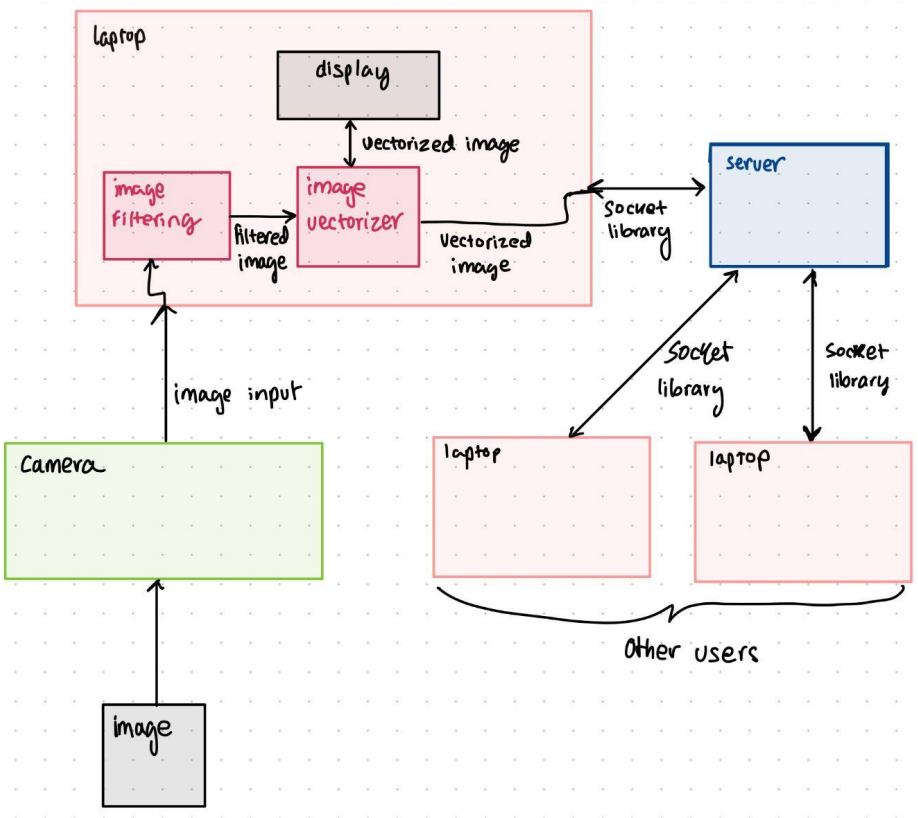
Technical Challenges

- Meeting latencies for smooth user experience:
  - Capturing the image with OpenCV
  - Filtering noise
  - Vectorizing the image
  - Rendering many objects
- Accurately recreating image
- Handling dropped connections

Solution Approach

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

- Capture image
- Vectorize image
- Render
- Allow for translations and scaling
- Broadcast rendered image to connected users



Solution Approach

- Camera: **Laptop camera**
- Computer Vision: **OpenCV**
- Vectorization algorithms: **Python**
- Server/clients: **pySocket**
- Frontend: **Python**

Testing

CV

- Test on different devices
- Use ballpoint pens, and markers
- Vary lighting and image distances

Vectorization

- Vary diagram complexity (10 primitives, 25 primitives, 50 primitives)
- Compose translations and scaling

Socket

- Connect up to 5 users

Verification, and Metrics

CV

- Time to capture image

Vectorization

- Time from captured image to output on web GUI
- Time for GUI response for translations and scaling
- Poll people on a 1-10 scale for image accuracy

Socket

- Time for diagram to appear on connected users

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 Software for line detection**	Lisa	Lisa										
	Write Software to Render CV output				Lisa	Lisa							
App GUI	**Develop basic framework for App GUI**			Lisa/Ronald	Lisa/Ronald	Lisa/Ronald							
	Create image capture framework				Lisa/Ronald	Lisa/Ronald							
	Develop whiteboard (user interface) for GUI						Lisa/Ronald	Lisa/Ronald	Lisa/Ronald	Lisa/Ronald			
	Create "send to other users" feature within GUI							Lisa/Ronald	Lisa/Ronald	Lisa/Ronald	Lisa/Ronald		
Vectorizing	**Research How to Vectorize Images**	Denise	Denise										
	Write software for vectorizing images			Denise/Ronald	Denise/Ronald	Denise/Ronald	Denise/Ronald	Denise/Ronald	Denise/Ronald	Denise/Ronald			
	Add translation feature of vectorized object							Denise/Ronald	Denise/Ronald	Denise/Ronald	Denise/Ronald		
	Add resizing feature of vectorized object								Denise/Ronald	Denise/Ronald	Denise/Ronald	Denise/Ronald	
	Parallelize Rendering Code								Denise/Lisa	Denise/Lisa	Denise/Lisa	Denise/Lisa	
Sockets	**Write Sockets Server Software**	Ronald	Ronald	Ronald									
Testing & Verification:	**Image Capture**					Ronald							
	Line Detection			Lisa									
	Resizing Vectors										Denise		
	Translating Vectors										Denise		
	Rendering Images on GUI						Lisa						
	Sockets: ensure users receive sent images										Ronald		
Integration	**Integration/Improvements**											Everyone	Everyone

Conclusion

Even though COVID is nearing its end virtual collaboration is still on the rise people need to work together from various parts of the world

Q

&A