AutoPuzzlr

Taking the fun out of puzzles

A0 - Andrew Conduff, Connor Maggio, Aneek Mukherjee

Application Area

Puzzles are fun, relaxing, and beneficial for mental acuity, but time consuming.

More people could reap those benefits if puzzles took less time.

We are creating a CV-based solution to help people get more enjoyment out of puzzles.

Those who may want to improve their fine motor skills will be able to work on puzzles without having to struggle through finding the pieces themselves.

Solution Approach

Big Changes: We removed the 4-legged PVC frame, the Leap Motion hand tracker, and the Epson 1776W Projector from our design due to losing access to the tools/spaces/stores necessary to build them.

AutoPuzzlr consists of two main components: 1) a laptop computer and 2) a Logitech C920S webcam.

1. Finger Detection is now entirely in OpenCV and detects a hand holding a point for 3 seconds instead of a tap
2. The laptop computer replaces the projector as the UI and displays the instructions, predicted solutions, and other visuals.

Solution Approach - Finger Detection

Big Changes: The Leap Motion tracker was removed and a new method of user input purely using OpenCV

AutoPuzzlr's finger detection now tracks the user's hand in the webcam view using OpenCV and identifies the fingertip in each frame.

When the user holds a position for 3 seconds (by resting their hand on the table), the detection will recognize it as a "point" and crop out the area pointed to and report that to the puzzle matching pipeline

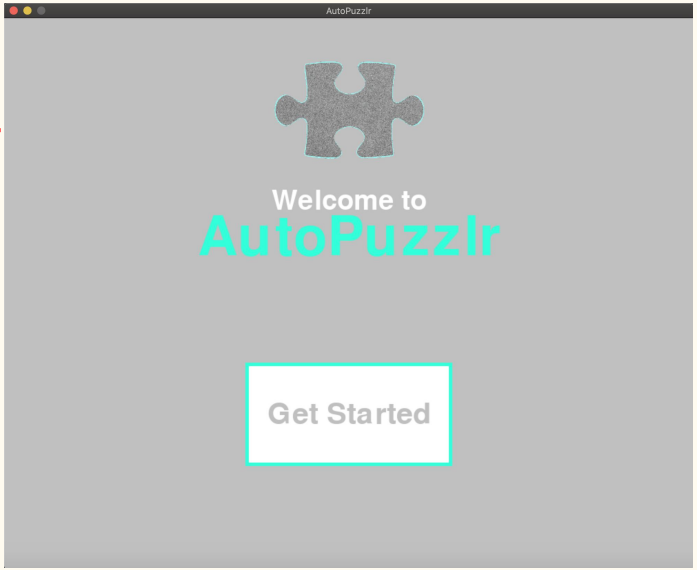
Solution Approach - Puzzle Matching Pipeline

Big Changes: Added extra safety to the end of the pipeline if the keypoint + RANSAC method was not strong enough.

The user inputs a picture of the final image of the puzzle (potentially the front of the box). Then as the user points at a piece that image data is captured and sent to a RANSAC based pipeline, wherein the features are matched and then displayed to the user. If there are not enough key points to match the piece, (i.e. a very featureless swath of sky) there is a fallback method which involves a more brute force method of template matching with intermittent rotations and it chooses the best approximation for the placement of the piece.

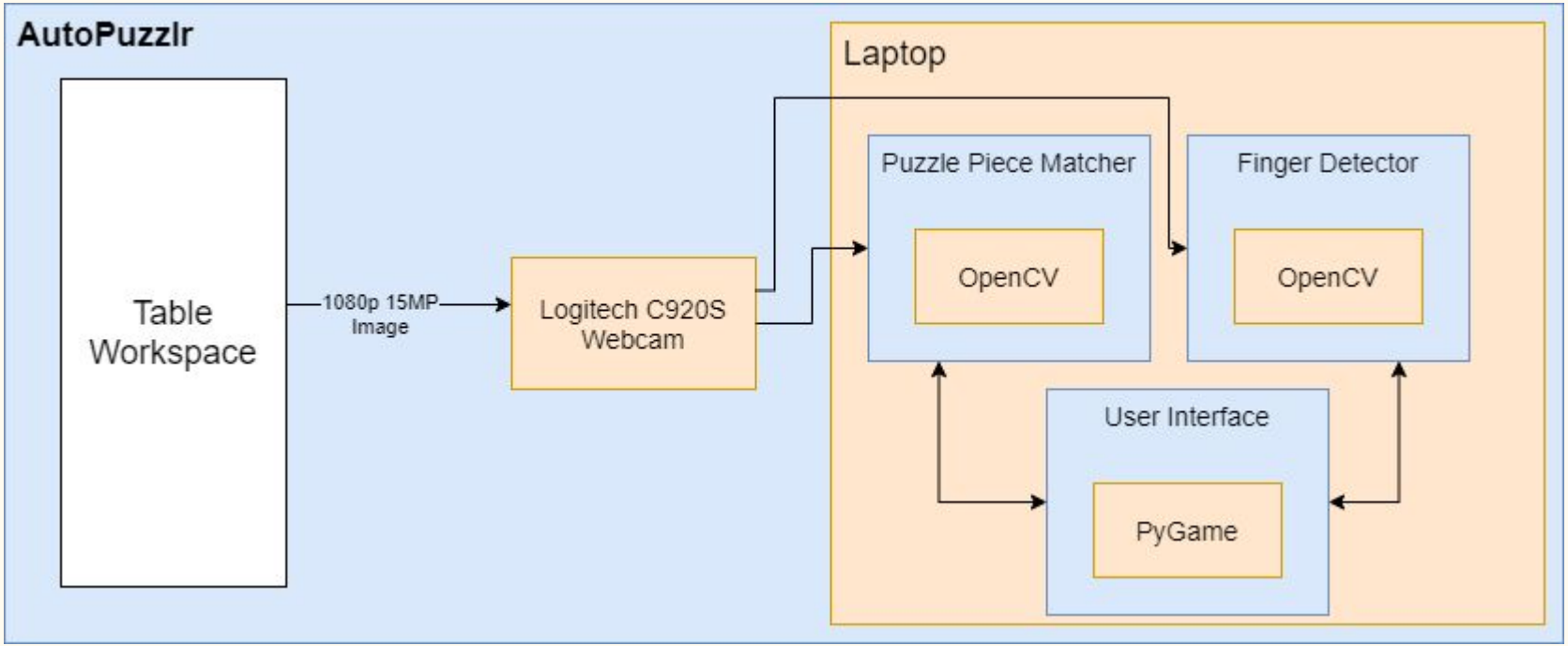
Solution Approach - User Interface

Big Changes: *Using PyGame to develop the UI*



- *Most interaction through webcamera*
- *Setup Phase:*
  - *Input final image*
  - *Upload or Take Photo*
- *Tap Phase:*
  - *Finger Detection*
- *Piece Place Phase:*
  - *Displays Puzzle with red square around placement*

Block Diagram



Complete Solution

Through the UI the user inputs a completed picture of the puzzle, then the camera is pointed at the assorted puzzle pieces, with the picture side facing up with space in between each piece. Then the user will point or tap a piece. This is detected using our finger detection software running OpenCV. Once a tap is detected, a picture of the tapped piece will be sent through the piece placement pipeline and display on the uploaded image of the final puzzle with a box surrounding where that piece should go. Then the user is able to place that piece where it should go on the surface where the puzzle is being completed.

High-Level User Requirements

Name	Requirement	Actual Measurement	Description
End-to-End Suggestion Latency	<4 seconds	1.2 seconds- 4.8 seconds Avg. 1.4	AutoPuzzlr must calculate the suggested location of a piece within 4 seconds of recognizing a user tap.
Suggestion Precision	<0.5 inch	0.0 - 1.5 inch (based on a 20 inch puzzle and pixel translations)	AutoPuzzlr must suggest a location within .5 inch of the piece's actual location.
Suggestion Accuracy	90%	82%	AutoPuzzlr must meet the precision requirement for at least 90% of the suggestions.

Technical Requirements - Camera & Projector

Name	Requirement	Actual	Description
Camera Field of View	100% of workspace	100% of workspace	AutoPuzzlr's camera must be able to see the entire workspace (defined by the 4 legs).
Camera Sensor	12+ MP resolution	12+ MP resolution	AutoPuzzlr's camera sensor must have high enough resolution and color detection to identify features in <1" puzzle pieces.
Projector Image	100% of workspace	N/A	AutoPuzzlr's projector should project an image that covers the entire workspace (defined by the 4 legs).
Projector Brightness	Sufficient to be visible against workspace	N/A	AutoPuzzlr's projector requires sufficient brightness and contrast to be visible against the workspace (puzzle pieces and background).

Metrics and Validation

Name	Requirement	Actual	Description
Tap Detection	<50 ms	250-300ms	AutoPuzzlr must detect a tap and its location in the camera frame within 50 ms.
Piece Identification	<500 ms	10ms	AutoPuzzlr must identify the piece being tapped within 500 ms.
Piece Matching	<3 seconds	.3s - 4.5s (worst case) (avg. .86s)	AutoPuzzlr must identify a probable location for the puzzle piece within 3 seconds.
Response Latency	<50 ms	10ms	AutoPuzzlr will display a response animation within 50 ms of a user action (tap) or suggestion.

Project Management & Schedule (COVID Revision)

Tasks	3/15-3/21	3/22-3/28	Mid-semester Demo 3/29-4/4	4/5 - 4/11	RIP Carnival 4/12-4/18	Final Demo 4/19 - 4/25	Final Presentations 4/26-5/2		
Ordering technology									Andrew
Design									Aneek
Make system architecture									Connor
Computer Vision + Projector									
Piece HSV Edge segmentation	Andrew								
Piece segmentation	Andrew								
Individual Piece Identification		Andrew							
Piece Feature detection (SIFT)			Andrew						
Puzzle feature detection (SIFT)				Andrew					
Piece matching in Puzzle					Andrew				
Piece Edge matching (Length)			Aneek						
Piece Orientation (RANSAC)	Aneek								
Piece Orientation matching		Aneek							
Research									
"Camera" research					Connor				
Alg. trade study				Aneek	Aneek				
Simulation									
Puzzle Input	Connor								
Puzzle slicing		Connor							
Click & Drag			Connor						
Prepare Mid-Semester demo		All				All			
Prepare Final demo									
Make Final presentation									
Testing & Integration									
Integrating Camera Stream									
Readings + Writeups							All		