Carnegie Mellon University

Robot Pixel Art

Team Copypasta

Bhaswanth Ayapilla, Daksh Adhar,
Parth Singh and Sreeharsha Paruchuri

04.22.25

Motivation and Problem Statement

Problem Statement:

Given an input image, use the system to generate an 8x8 grid representation of it with a fixed set of colours and use the arm to stamp the image while optimizing for the total time taken.

Motivation:

- Explore classical planning algorithms
- Understand the challenges associated with time optimization
- Utilize accurate planning and impedance control algorithms
- Cool robot art!

Approach

Input:

A square image of any resolution (up from a fixed resolution)

Major Systems:

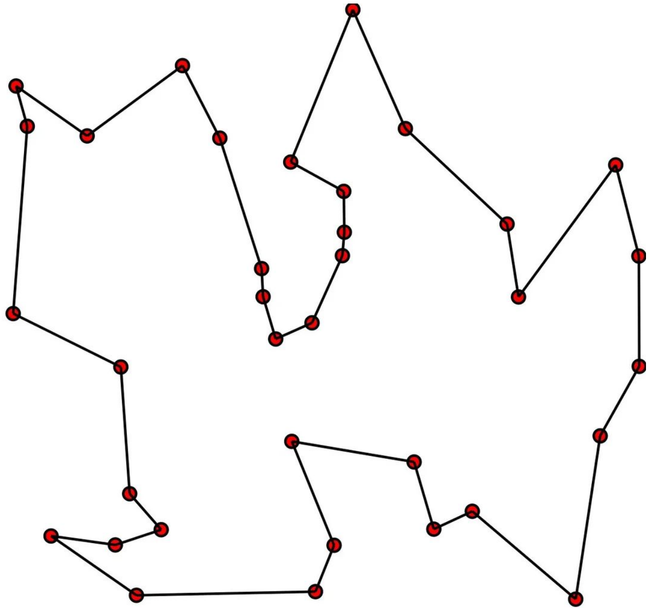
1. Input image processing
2. State machine
3. Motion planner - TSP
4. Low Level FrankaPy motion planner

Output:

An 8” x 8” pixel art image in the least possible time!

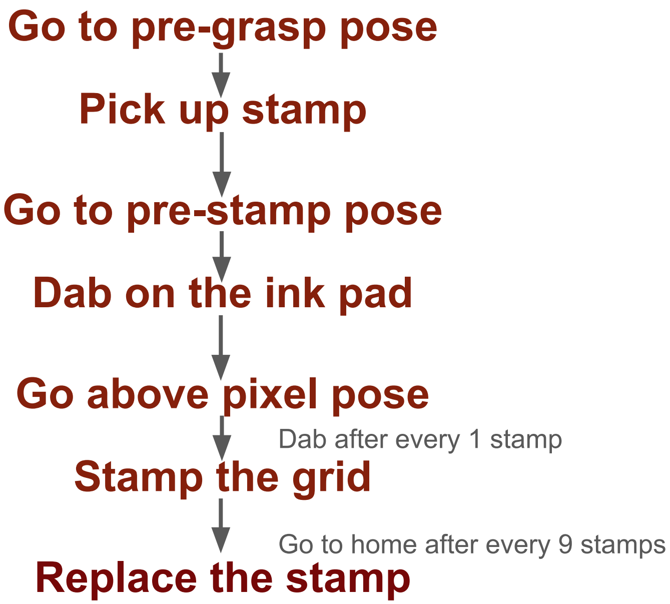
Travelling Salesman Problem

- Goal is to find the shortest possible route that visits a set of nodes and returns to the starting node
- NP Hard Problem
- In our case, each pixel corresponding to one color is the node… How do we TRAVEL these nodes in the shortest time?
- We solve this problem by using CHRISTOPHIDES ALGORITHM which is a approximation algorithm
  - Connect all nodes with cheapest edges (MST).
  - Find shortest edges to pair up odd nodes
  - Walk without repeats, using shortcuts if needed



Minimum Viable Product

Stamping procedure for a single pixel



Go to pre-grasp pose

Pick up stamp

Go to pre-stamp pose

Dab on the ink pad

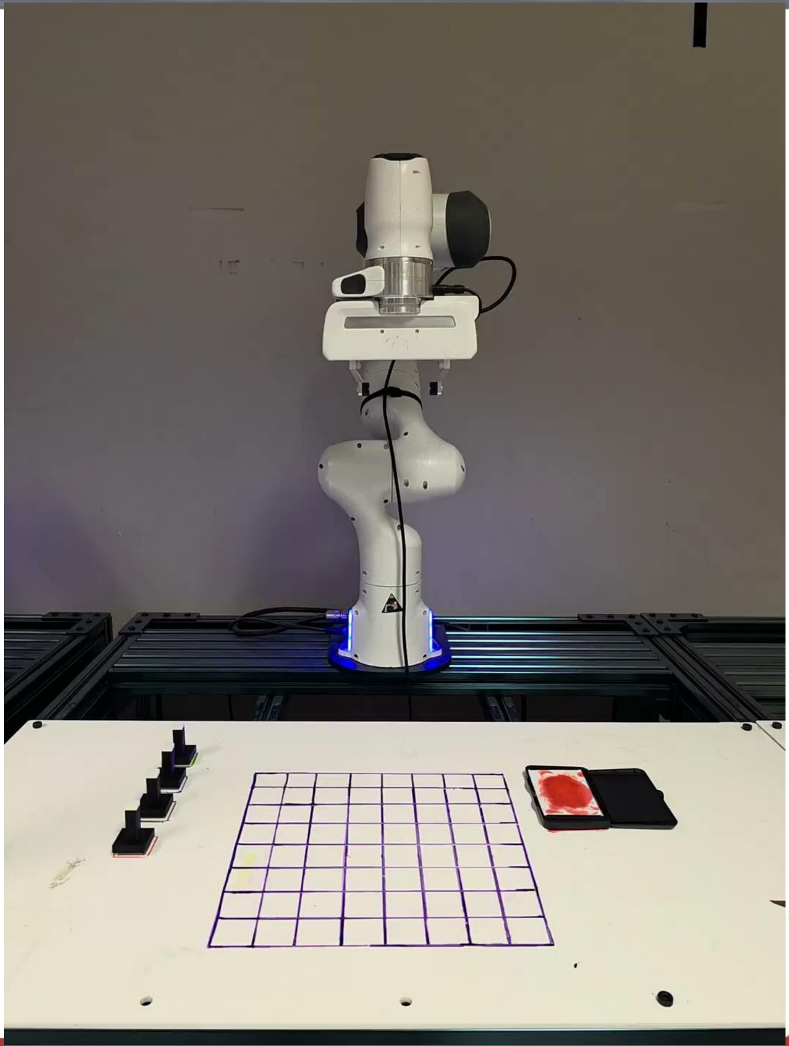
Go above pixel pose

Dab after every 1 stamp

Stamp the grid

Go to home after every 9 stamps

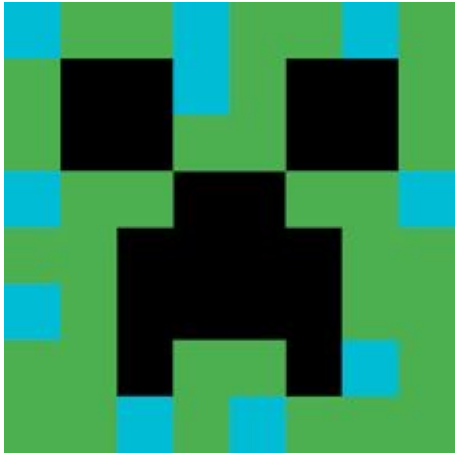
Replace the stamp



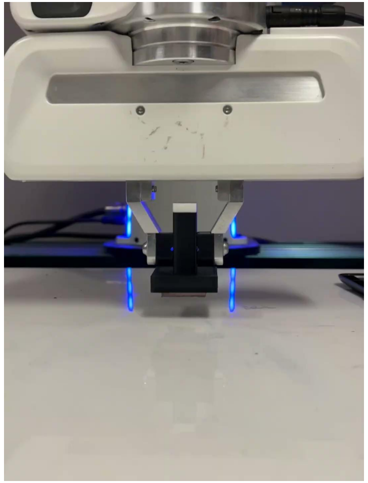
Input:

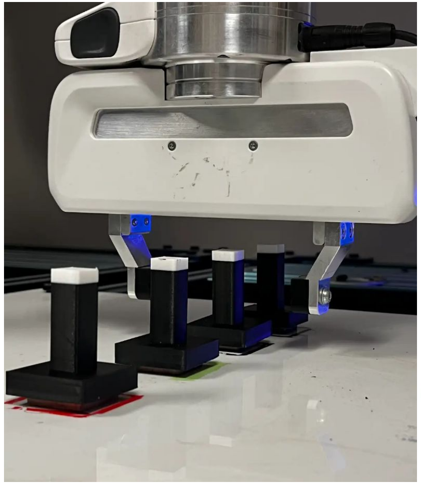
A 1200x1200 pixel minecraft creeper

This is a good test case given the various colours involved in the query image

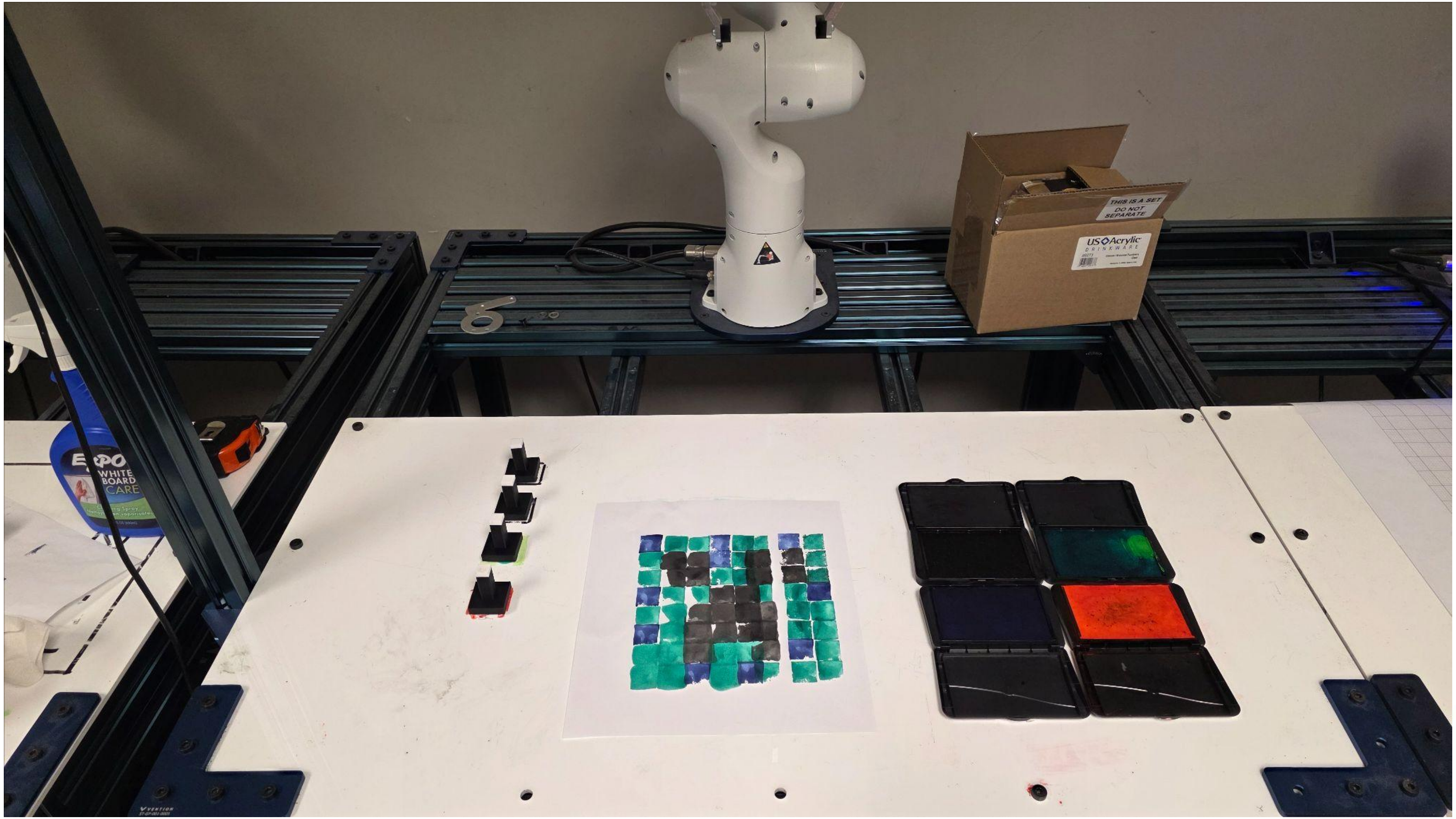


Stamping Mechanism:

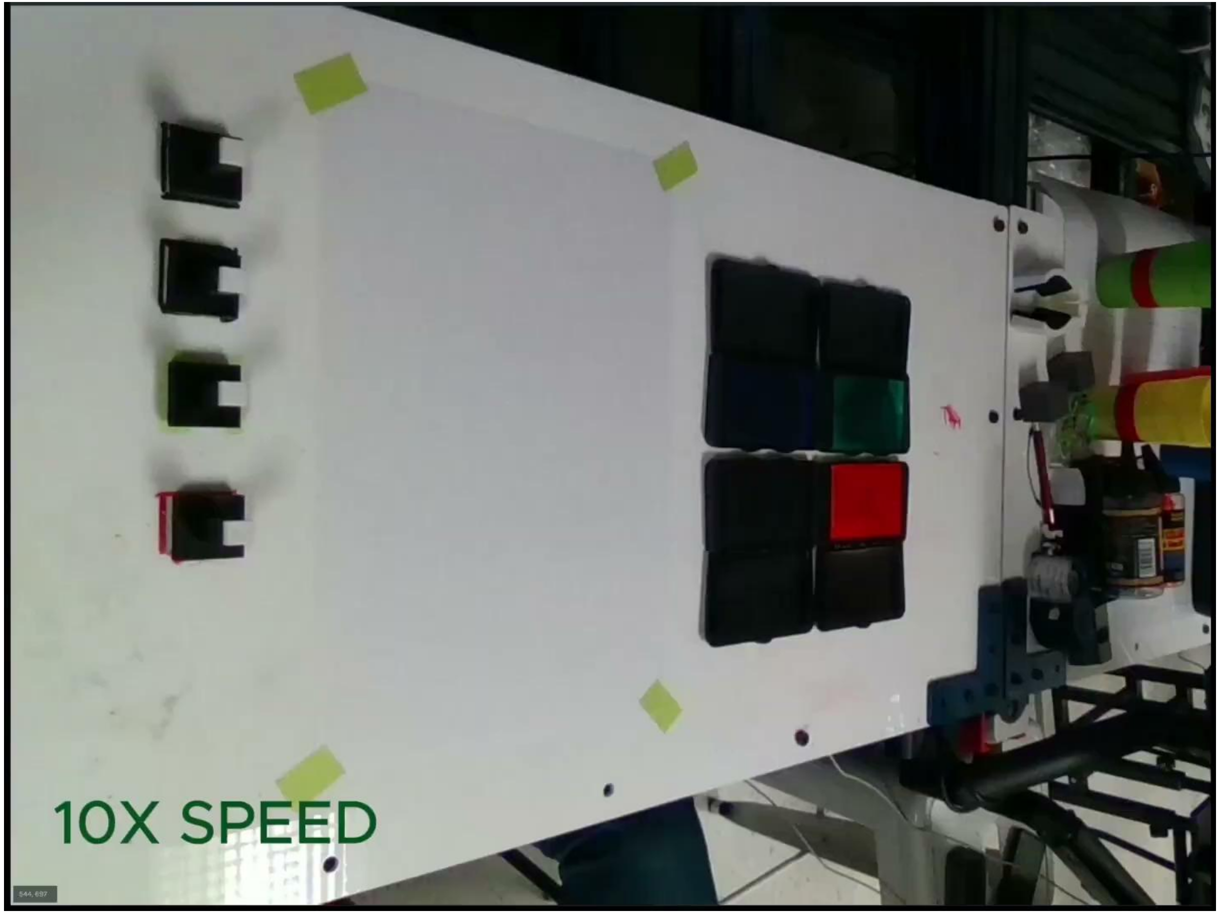


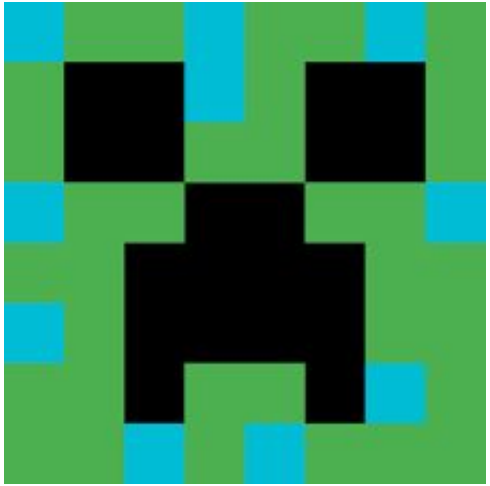


MVP System

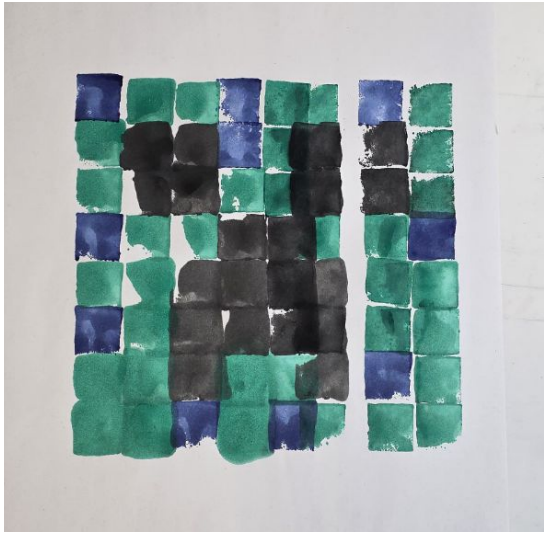

THIS IS A SET
DO NOT SEPARATE
US Acrylic
DRINKWARE
EXPO
WHITE BOARD CARE

Stamping Video





54 minutes



Blockers

- Can’t test frequently with all ink colours as they dry out quickly
- The stamp material doesn’t absorb the ink well enough for multiple stamp actions in a row, have to dab in the ink after every 3 stamps
- Long runtime for a single pixel art (~50 minutes for 8x8)
- Stamp does not lay flat on the board (Rubber material not sponge)
- Unfaithful goto_pose results in an offset between columns 5,6

Future Steps and Goals

- Track the location of ink pads in real time with aruco tags
- Visual feedback from the realsense to verify pixel stampage and retry stamping that pixel if needed
- Utilising force control to get consistent stamping quality on the pixel grid
- Dab on the stamp pad after every 3 stamps instead of 1 for faster pixel art completion

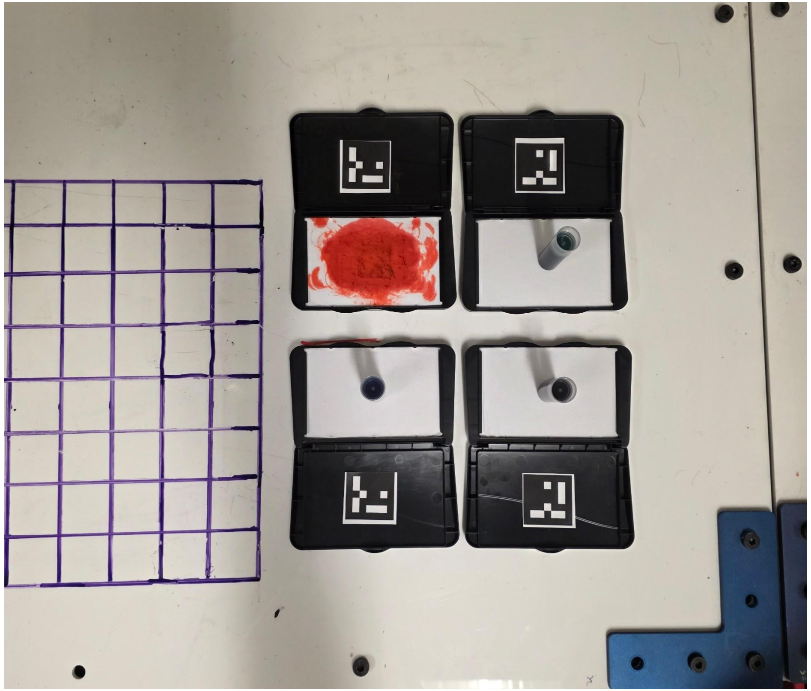
Intel RealSense Communication

Hardware:

- Intel RealSense depth camera
- ArUco tags strategically placed near pads

Workflow:

- RealSense camera detects and tracks ArUco tag positions and orientations
- Calculated poses inform the Panda arm where to precisely:
  - Pick up stamps
  - Dab stamps onto designated pads



Thank You

Questions?

THE ROBOTICS INSTITUTE

Stamping Video