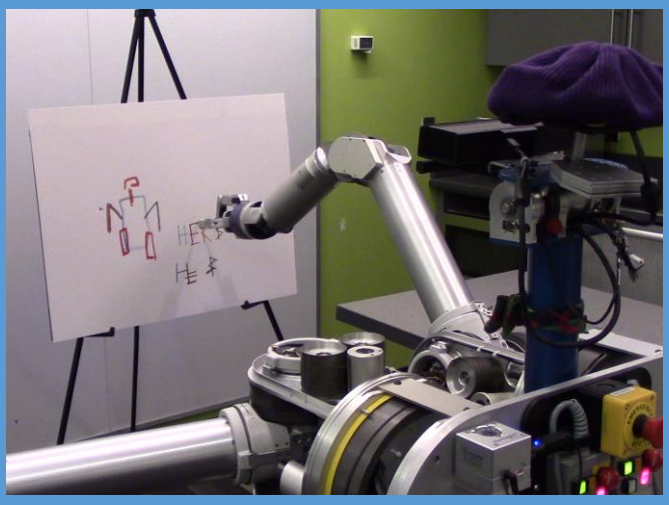
16-662 ROBOT AUTONOMY

AUTO PAINTING PROJECT REPORT



TEAM
BEN BALLARD
JIMIT GANDHI
MOHAK BHARDWAJ
PRATIK CHATRATH

DATE: MAY 6, 2016

1. Project Overview

1.1 Background

This project was undertaken over the course a 4-month semester for 16-662 Robot Autonomy class. During the project we kept in line with the 1st International Robot Art competition [1]. This competition is organized by Roboart.org [1] for schools and research teams across the world to build robots who can paint art. We decided to work on the complete autonomous category of the competition where the robotic system autonomously creates the artwork from scratch.

1.2 Problem Description:

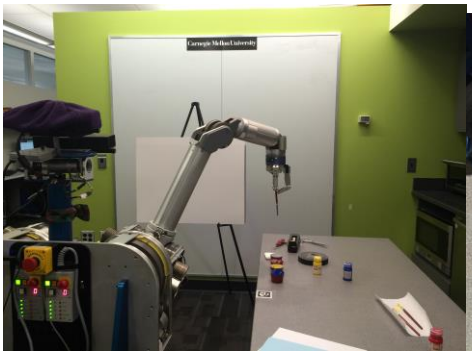
Our goal was to get HERB to paint autonomously. The following were the problems that were required to be solved.

- Environment Setup - Physical setup of easel, canvas, paint colors
- Grasping the paintbrush - Getting HERB to grasp the paintbrush
- Configuration - HERB localizing canvas, table, colors
- Gathering paint - HERB changing and refilling paint
- Planning on the canvas - Constraining movement of arm over the canvas plane

Section 2 describes each of the problem in detail. Once we solved the above mentioned problems next step was to draw different paintings. Section 3 describes the various paintings that HERB drew and the method we used for painting.

2. Problems & their Solution

2.1 Environment Setup



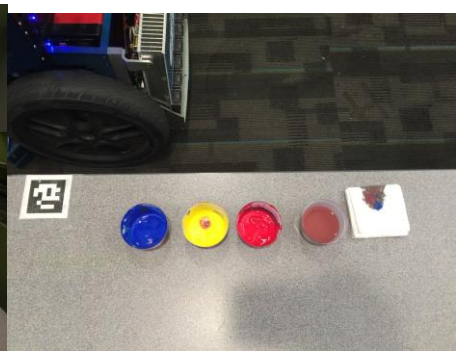


Figure 1 (left): Initial setup of the environment around HERB
Figure 2 (Right): 4 cups setup with paint and water on the table next to HERB

The environment was setup so that Herb does not have to move its base to paint on the canvas. Specifically, a canvas was set up in front of Herb and a table to his right. The canvas was on an easel; therefore, it was not exactly vertical but tilted at an angle.

On the table sits a set of 4 cups. Three of the cups have colors in them and one has water for cleaning the brush. There is also a folded paper towel setup beside the cups so that Herb can dab excess water and color off on the paper towel. The cups and paper towel were all taped down so that Herb would not knock them over.

2.2 Grasping the Paintbrush

The paintbrush is grasped by having Herb's three finger tips close around the midpoint of the brush. This holds the brush pretty well, however the brush pivots at the point where the finger tips touch. To fix this issue the back of the paintbrush was taped to the palm of Herbs hand.

Figure 3: HERB holding the paintbrush

We experienced a few problems with the brush moving over time. This was due to having arm plunge too far down into the paint cups and this would adjust the location of the brush in the hand. This issue was procedurally controlled by making sure that we had a sizable amount of paint so that plunge would not have to hit the bottom of the cup.

2.3 Configuration

Once the environment is setup and Herb has grasped the paintbrush, Herb needs to be configured for the environment. The configuration process allows the user to define position of three things with respect to HERB:

- Table and Items on the table
- The locations and plane of the canvas
- A position in front of the canvas. This position is required so that the joint angles always start from a similar position.

Figure 4: User configuring HERB to the canvas

We generated a python script to take the user through the configuration process. The first portion of the process was to configure the table locations. First robot arm is set to be made not stiff, meaning that a human can move the arm. Then the script would request the user to move the arm the following positions red, blue, yellow, preColor, paper towel (dab), water (clean), and canvas. The PreColor position is setup so that the arm will find a plan that goes straight from this position and into the cups without having any issues. The cups are not actually modeled in the simulation environment. Therefore, the planner does not know about cups as obstacles and could plan through them. This position fixes this issue.

After the table is configured, the canvas plane is configured. This is done by collecting 3 points that define the plane of the canvas. The three points are the bottom left point, top left point and the bottom right point. To find a point the user first moves the arm near the point. Then the robot takes over and makes sure that it's palm is facing away from Herb. This is done so that the length of the brush does not need to be taken into account when finding the plane. Then the user tele operates the robot using a set of key commands until the brush just touches the

surface. To tele operate the robot would move the end effector left,right,up,down,in and out by 1 cm based on what key was pressed.

When the color positions and the canvas positions are configured they are saved out to a file so that this only has to be done once per painting session.

2.4 Gathering Paint

Herb is able to gather paint so that Herb could change colors and refill when the brush had no more paint left in it. We did not have any perception feedback. Therefore, Herb gathered paint either after every so many strokes or when a predesigned painting was configured to. Here is a video showing the paint gathering process: https://www.youtube.com/watch?v=io82gjj8VUs

The process to gather paint is simple. Herb first moves his arm to the pre color position which is right above the colors. Then Herb moves his hand to the color position and then plunges 1 cm down into the paint. At this point Herb then rotates its wrist to the min limit, then to the max limit, and then to the initial joint value. This helps get paint on the brush. At this point Herb reverse his movements and goes back to the pre color position. This sequence of events is played to grab each color, clean, and dab. The only difference is that the dab procedure does have Herb spin his wrist. After grabbing a color Herb always performs a single dab so that any paint that might drip off goes onto the paper towel and does not cause a drip on the canvas.

2.5 Painting On the Canvas

Once HERB is configured and has the capability to change color the planning aspect of getting HERB to paint on the canvas can be broadly divided into two parts:

- **Planning the movement of the arm:** The goal of arm motion planning for painting was to draw a set of 2D points as a single connected line on the canvas. We used the Personal Robotics Lab's `herbpy [2]` and `prpy [3]` ROS package to plan along a unit vector which was along the plane of the canvas. The `PlanToEndEffectorOffset` function was used for this purpose. To find the unit vector along the canvas, the 2D points were transformed into 3D space then the difference between current point and the next point was used to find the vector in 3D space to plan along.
- **Constraining the movement of the arm to the canvas:** This was one of the most challenging parts of our project. Broadly, the procedure employed for controlled motion of the arm involved moving the arm to a preconfigured canvas position, rotating the wrist so that the paintbrush points directly away from the robot, move to an initial position near the origin of the canvas, move along the normal to the plane till you touch it followed by moving along the set of 2D points. The process was optimized using virtual robots for fluid movements so that the robot never has to move back to the 0,0 position on the canvas as it was done virtually and final dofs were sent to the robot.

3 Paintings

Many different ideas were explored when trying to actually make Herb paint autonomously. Below mentioned are the ideas that were implemented.

3.1 HERB's self-portrait and signature

HERB was able to draw its own self portrait as a stick figure and sign-off its name. These shapes are specified as waypoints to HERB who plans a path between them to generate trajectories. The path for every geometric shape was planned as one continuous trajectory and executed by HERB in one go. Routines to recover in the case that failed to plan for a certain part of the trajectory were also implemented.

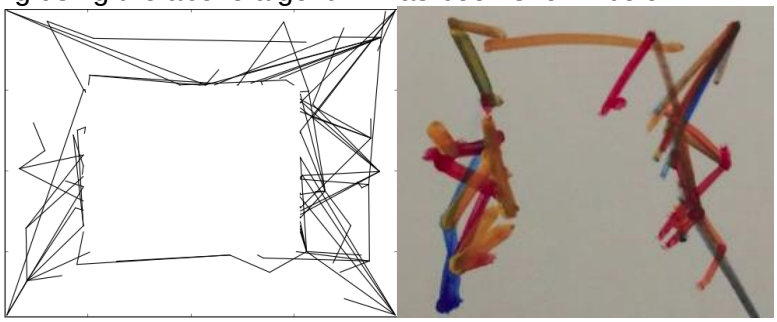
Figure 5: Herbs Self Portrait and Name

3.2 Random-Planned Art using RRT

Building up on the thought of determinism versus randomness in art, a parallel was drawn between painting on a canvas and random motion planning. The algorithm that was developed is as follows:

- "What HERB wants to show" was defined as obstacles in a costmap.
- The obstacles were softened, a gaussian filter was applied to the costmap and then costmap gradients were determined
- Coverage v/s detailing - This was a randomly drawn number which specified the trade-off between exploring the canvas and detailing particular obstacles.
- Based on the number drawn above, the underlying probability distribution from which the RRT samples was changed - uniform for coverage and weighted by gradients for detailing. This helped to control the voronoi properties of RRT.

An example painting using the above algorithm has been shown below:



6a 6b

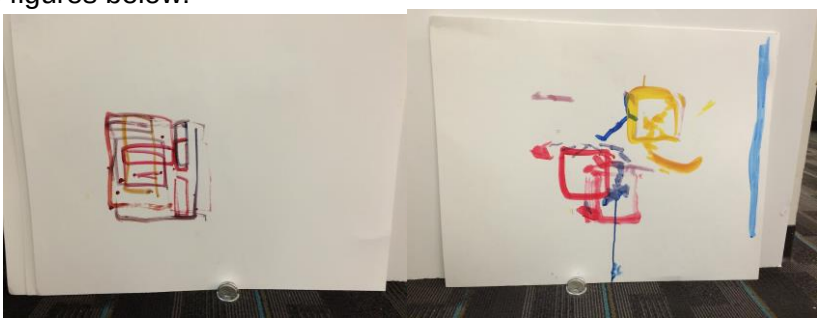
Figure:6a. A square using the random motion planning in simulation
6b. Part of the painting actually created by HERB

The key advantages of this algorithm include:

- Easy to extend to relatively abstract and complex ideas.
- Easy to apply on a real robot with poor accuracy and movement constraints
- Coverage v/s detailing allowing for faster computation

3.3 Geometric Shapes

Functionality had been developed for HERB to be able to draw different geometric shapes such as rectangles of different sizes and circles. These could be combined together to generate different complex objects. A few examples of the different attempts at drawing geometric shapes are shown in the figures below:



7a 7b

Figure:7a. Randomly sized rectangles
7b. Initial attempt at curves and rectangles

4. Complete project Video: https://www.youtube.com/watch?v=wi_v87_9ZQ4

5. Future Scope

This project can be extended to include many exciting features which will put up new interesting challenges and problems to solve. Some of these features are as follows:-

- Make Herb draw on a curved 3-dimensional surface (manifold) instead of a planar canvas.
- Make Herb mix paints and come up with new shades of different colors.
- Make Herb grasp the brush autonomously thus eliminating manual intervention for adjusting the brush.

Also the majority of issues we faced while testing the painting project was due to absence of perception system. There was no way to localize the plane after one stroke. It was all based on the accuracy at which Herb could attain the desired joint angles. The perception system will help eliminate errors that arise in cases when paint smudges as the brush is pressed hard against the canvas and cases when brush barely touches the canvas.

6. References

[1] Weblink - “ 2016 1st Annual International RoboArt Competition” - http://robotart.org/

[2] Weblink - “Personal Robotics Lab CMU, HerbPy python library” - https://github.com/personalrobotics/herbpy

[3] Weblink - “Personal Robotics Lab CMU, PrPy python library” - https://github.com/personalrobotics/prpy