EEL5666 Intelligent Machine Design Lab
Spring 2000
Prof. Dr. Arroyo
TA Ivan Zapata
TA Scott Jantz

SCAVBOTS

By DAVID GRINDLINGER
CISE, University of Florida

CONTENTS

ABSTRACT

EXECUTIVE SUMMARY

INTRODUCTION

Robert Brooks at MIT has suggested the future for robotics may lie in small, reactionary mobile constructs, deemed “mobots,” rather than in complex, centralized, contingency based systems. The Scavbots are my attempt to construct a pair of mobots, and show that they could work towards the same goal without vast contingency routines, and without inter-machine communication.

The Scavbots, so called be cause they mimic the food scavenging behaviors of social insects, were designed with certain philosophical limitations. Beyond being small, autonomous, and having limited computational powers, Scavbots should not require any type of transmitted information in order to work in the same space. I.e., they should not beam formatted signals to each other via IR or RHF or other electromagnetic wave, they should not be connected by a cable, etc. The Scavbots should be able to alter their environment in some way, as well as to perceive and react to it. All Scavbots should be identical in hardware and software.

The result were two identical robots designed to perform ant-like behaviors. The Scavbots are capable of following ink lines the way ants follow pheromone trails. Building on the concept of a pheromone trail, I designed the Scavbots to have color recognition—to be able to distinguish different trails and respond with different behaviors. To alter their environment, I equipped the Scavbots with line drawing equipment, simulating the ability of ants to lay new trails. In order to effect these mechanisms, my mobots must be operated in fixed environment, where the ground is a white dry-erase board.

In principal, the mobots' task is to search their environment, find a target representing food, create a trail from the food to a second target representing a hive, and then fall into a pattern of going back and forth between the two. This would simulate the scavenging and gathering behaviors of an ant-column bringing back food to their communal hive.

SYSTEMS OVERVIEW

The Scavbot systems can be broken down into three main categories: control, actuation, and sensors.

The control system is a Mekatronix TJ-Pro board, a setting for the Motorola 6811 8-bit microcontroller. This setting has eight analog inputs through which it receives sensor data. I felt that this was not enough, so I added an analog input multiplexor to expand the number of analog inputs to fifteen. This was done at a sacrifice of three digital input/output signal pins, leaving five. The MTJ-Pro board also has a memory bus with 32k of volatile RAM

Actuation is generated by three 40-0z standard servos. One servo powers the left wheel, another the right wheel, and the third operates the trail-marking dry-erase pens.

The Scavbots are equipped with three types of sensors, IR detectors, momentary contact collision detectors, and variable photo-resistor CdS cells. Some of the CdS cells are used simply to do line following behaviors, but others have been specially filtered to perform color recognition.

The basic operation of a Scavbot uses all of these elements. Scavbots moves around the environment, avoiding obstacles with short ranged IR emitters and detectors, and reacting to collisions when the contact switches are pressed. Using the bottom mounted CdS cells, Scavbots locate food, hive, or trails when they roll over a piece of floor that is significantly darker than the rest of the surface. When the target is under the center cells, the Scavbot software uses differential readings between the red and green filtered CdS cells to analyze the color of the surface. The color is the trigger for an appropriate ant-like behavior.

MOBILE PLATFORM

The Scavbot platform is a modified Mekatronix Talrik chassis. This platform is a wooden disc with an opening in the center. The opening allows wires to pass from the underside of the disc to the top. Contact switches mounted around the circumference of the disc are used for collision detection. Velcro squares and vertical tabs with holes are regularly placed around the disc's top surface perimeter for mounting IR sensors and emitters. I used four emitters and three sensors on each Scavbot. The switches are surround by a flexible plastic ring which provides a continuous contact sensor surface by pushing against the contact switches when his from any direction that is parallel to the ground. Wheels are attached to two standard servos, mounted in brackets on the underside of the disc. A third free-rotating wheel is mounted on the back to provide three-point stability for the platform. A bridge is top mounted on the Talrik, allowing a second tier of flat surface area, or a raised sensor array.

I adapted this platform to my mobots' need in several ways. I mounted a wood block across the center opening of the body disc, used to vertically suspend two plastic tubes. These are chambers for holding dry-erase markers in position to be raised from and lowered to the ground. A cut into the rear half of the body allows the surface mounting of an additional servo. This servo has a long arm centrally mounted on the rotator. From each end of the arm, fifty-pound test monofiliment is extended over a bridge support and dropped down to one of the pens. The pens are weighted with lead fishing symmetrically weights hot-glued to the sides of the pen. When the arm is in a neutral position, neither pen is touching the ground. When rotated forty-five degrees clockwise, one pen is touching the ground, and when rotated forty-five degrees the other way, the other pen touches the ground. The weights on the pens ensure a tight line between the servo and the pen, and also a good contact between the marker tip and the dry-erase board.

Another change I made was to move my control boards on top of the bridge. Although slightly more vulnerable this way, the move was necessary to open the center opening for the pen tubes and pen actuation.

The CdS cells, and their corresponding LED illumination sources, were mounted in black rubber tubes, and forced through a 1.5" Styrofoam crescent. The tubing, originally used for cool fluid transmission in cars, helped shield the cells from ambient light, and direct the light from the LED's at an area directly underneath a target CdS cell. I also found that colored plastic filters, cut from vinyl folder tabs, could be adhered to the rubber by carefully melting the surfaces together at their edges. Cheap and easy to cut, Styrofoam proved ideal for mounting the tubes. I later fine tuned the direction of the tubes by prodding them into shape, and then trapping them with non-drying, tacky mounting putty. This Styrofoam module was reinforced with black electrical tape and mounted as a unit to the bottom of the platform disc, in front of the wheels.

Finally, to isolate the CdS cells on the Styrofoam module from ambient light, I attached a black cloth skirt to the plastic bumper surrounding the disc. The skirt was attached with Velcro tabs. It provided consistent readings from the CdS cells even when the environmental lighting conditions drastically changed.

ACTUATION

Something tricky about line following and color detection is that the robot's speed must be reduced from the full potential of the standard servos. I learned the hard way that the MOTORTJP code and the SERVOTJP code in the standard hc11 software libraries can not be used at the same time, al low speeds, without causing signal interference to the wheels. I found it necessary to plug my wheel servos into auxiliary servo ports on the TJ-Pro board, and all three of my servos using the servo(#, setting) command, rather than the motorp(#,speed) command. This allowed me to reduce the speed of my wheels to about 20% and still have full control.

The only goals for my mobots' actuation, beyond basic mobility, was to manipulate pens and draw trails on the ground. This was achieved by throwing a monofiliment line over a Talrik bridge support and connecting it to a pen, much like a sandbag hanging from a catwalk in an old style theatre. The pen is weighted down with fishing weights and suspended a quarter of an inch off the ground. The other end of the line is connected to one end of a straight servo arm, mounted behind the bridge on the top of the Talrik disc. The connection is made by threading the monofilament line through a pre-fab hole in the servo arm, and using a reinforced alligator clip to keep the line from sliding back out. This method securely holds the weight of the pen suspended yet allows quick and minute adjustments with only two fingers from each hand.

The servo rotates one way to lift the pen up, the other to drop the pen down. Two to three ounces of lead weights on the pen take up any slack in the line and assure a good contact between the felt-tipped marker and the ground.

The best thing about this method: since one line merely goes slack if the servo over-rotated, and the other pen is simply lifted a little higher off the ground, it is not necessary to be overly precise with servo control of the pens.

SENSORS

A Scavbot has several types of sensors for several purposes.

Infra-red detectors: give analog readings based on the distance of objects directly in front of them. Rely on the IR light reflected off the objects, and which is being output by IR-LED's mounted facing the same direction as the IR sensor. They have a range of one to three feet, depending on the individual sensor unit and the associated IR emitters. Scavbot's IR was not central to its task, so I did not work to optimize this sensor. I was able to demonstrate good avoidance of obstacles up to two feet away under ideal circumstances., and six inches away under difficult circumstances. I found the whiter and flatter an object was, the easier the IR detectors could pick up its IR reflections.

Momentary Contact Switches—bump switches placed around the outside return a digital value when pressed. This represents a collision. A resistor net built into the TJ_Pro board allowed simple separation of three front bumper zones and a unified rear bumper zone. It was therefore possible to know generally from which direction the Scavbot had been struck.

Unfiltered CdS cells—variable Cadmium Sulfide photo-resistors give an analog value proportionate to light striking their surface. The ones used in the Scavbots were the cheapest available from Jameco Co., and had approximate resistance ranges of 10kOhms to 120kOhms. By isolating the cells from ambient light, and providing LED's as a low-power and steady illumination source, I was able to determine the relative "darkness" of the surfaces over which the Scavbots passed. By using a software routine to find a default threshold, based on whatever surface under the robot when activated, it was possible for the Scavbots to follow colored trails on many surfaces, from clean white to dirty and mottled industrial green. The two unfiltered cells, located on either side of the Styrofoam module, were illuminated with green LED's. This is because CdS cells are centered on green light, and give very accurate reaction to surface color changes with this illumination. All my CdS cells, filtered or unfiltered, used a simple voltage divider circuit with a resistor of 47k to generate output. Analog values from these cells ranged from 1 when in total darkness, up to 249 when pointed at the lights on the ceiling. The unfiltered CdS cells were primarily used for line following behaviors. The green filtered CdS cell, centrally located on the Styrofoam module, did double duty in the color recognition system and the line following system.

Filtered CdS-cells, or Color Detection—CdS cells covered with transparent plastic covers in red, green, and blue. Each of these cells only measures the quantity of one brand of light getting through the filter. By using these as negative values, I was able to roughly determine the color of the surface that a Scavbot was driving over. Problems occurred for several reasons. 1) My central light source, though sold to me as a "white" LED, was really a pale blue. This didn't seem to matter all of the time, but I believe it caused the inks of some pens to reflect differently under the Scavbot sensors than under my own eye. 2) Inks are optically complex reflectors, and may change under subtly different sources of illumination. I was able to garner much more consistent color assignment from colored plastics and paper than I could from inked surfaces. 3) The blue filter caused definite changes in the values of light hitting the CdS cells, but in very narrow ranges. I eventually opted to throw out the results of my blue filtered sensor altogether and stick to red and green. 4) Different CdS cells have different resistor ranges. I eventually learned to normalize my cells by multiplying each one by (120/default threshold for that CdS cell). This made the data much more consistent between different robots, and also set up a more absolute scale between the different CdS cells, making the data easier to analyze. This was probably the most important evolution to my software development.

The basic color recognition algorithm for red and green was ultimately. It assumes the robot starts on a white surface to calibrate itself.

1) get readings from the red and green filtered CdS cells
2) subtract these values from the threshold calibrated at startup. These two values represent a figurative distance from white.
3) Add the values together—if the sum is very high, the color under the sensors is black, because the distance is too far from white. If the distance is very small (or negative!), then the surface is still white.
4) If the distance is moderate, check which filtered CdS cell has the higher normalized value. If the red filtered light is further from white, then the color is green, because the red filter will block green light. If the green filtered light is farther from white, than the color is red.

The exact values for what is “very high” and “very low” must be determined experimentally.

I believe if I had learned to normalize my values sooner, and had darker blue filters, I could have used the blue filtered CdS cell to more advantage. I also learned the hard way that it may be necessary to make the filters several layers thick in order to consistently block out colors of light and achieve a truer negative value for each CdS cell.

BEHAVIORS

The main behaviors of the Scavbots are finding food, finding the hive, following lines, drawing lines, and color recognition. Since finding a specific target could be reduced to recognizing specific colors, I will discuss line following and how I integrated the color recognition and line following into that behavior.

Line Following – What made the Scavbot different from other robots I had seen do line following is how close together the CdS cells are on the Scavbots. There is only an inch between the CdS cells. By placing the CdS cells and LED's at sharp angles-- allowing the light to bounce directly into the CdS cells-- and by leaving them close to the ground-- I was able to optimize the sensitivity of these sensors. Also, tight motor control was made possible by slowing the robot down, and by the centrally located wheel of the Talrik chassis: I could stop one wheel and continue driving with the other, allowing much smoother changes in direction while correcting for drift. An important behavior associated with the line following is the Calibrate() routine in my software. The calibration function takes several readings from the sensors, averages them, and subtracts a token constant in deference to error, noise, or fluctuation. The result is a very accurate threshold representing the reflectivity of that surface, and an appropriate trigger point for the line following behavior. These factors combined to make the Scavbot very good holding lines, even on moderately dark or irregular surfaces.

When one CdS cell in the line following complex hits a line, the robot begins to locate(). When the sensors have passed over the line, the Scavbot knows which sensor was tripped last. It rotates in the direction towards that sensor so as to minimize the time to acquire the line. If the last sensor tripped is the center sensor, the Scavbot continues straight. Once the center cell is centered on the line, the Scavbot can analyze the line's color.

After acquiring the line, when the Scavbot's left sensor hits-center on the line. It does the same for the right.

The color of the line may require a different reaction than line following. A line drawn by another Scavbot indicates a trail exists between food and hive, and that this Scavbot should quire the trail. If the color represents the hive, the Scavbot can drop off food and go back along the trail. If empty handed, the Scavbot randomly turns and goes back out in search of food. If the color represents the food, the Scavbot must turn around and return (with food) to the hive. If the Scavbot has just stumbled upon the food, and there is yet no trail to the hive, the Scavbot must make one. Line drawing is simply a matter of putting down a pen and proceeding towards the target. These behaviors are controlled with flags such as GOTFOOD, GOTHIVE, FOLLOWING, and DRAWING.

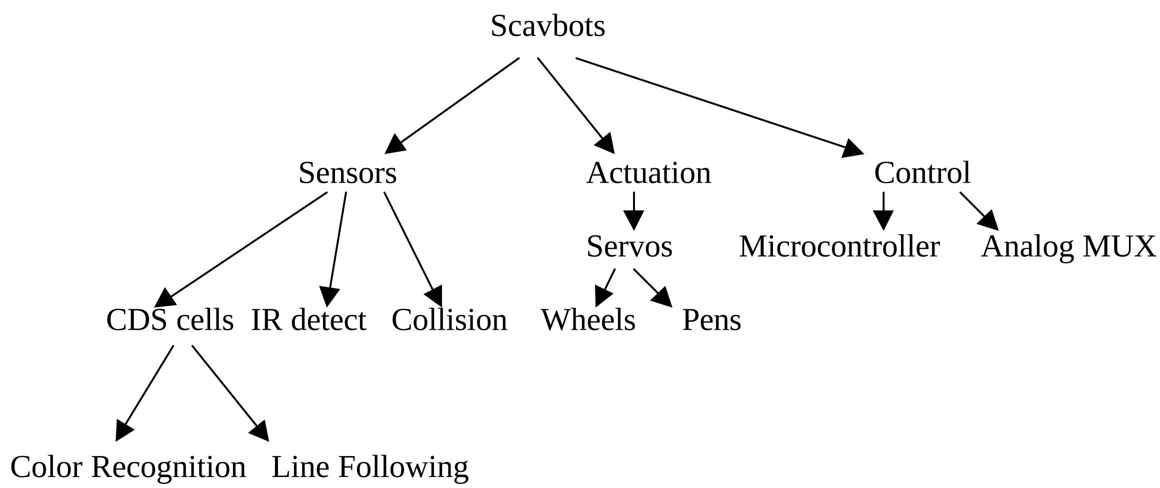
The Scavbots have no way of navigating save by their trails, so discovery of food and hive must first be done randomly. During their searches, the Scavbots practice basic collision avoidance behavior using both IR and contact switch collision detection. For my experiments, the hive and food have been directly across from each other, and the lines between them straight. I have developed contingency routines for robots colliding on trails. In this case, a Scavbot carrying food gets right of way, and the other Scavbot must do an arc around the food-laden bot, hoping to reconnect with the line further along the way. I have not developed contingencies for collisions near walls or near targets.

CONCLUSIONS

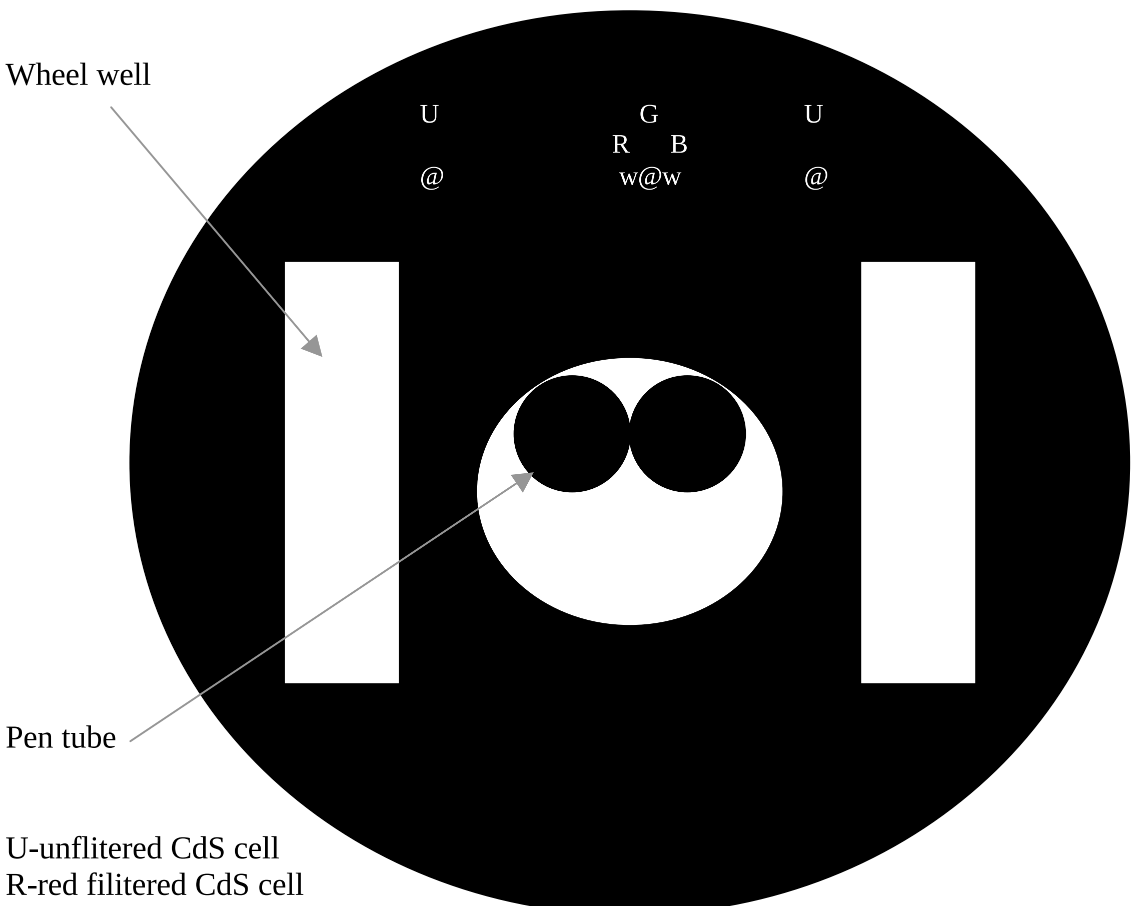
In this project I succeeded in developing limited color recognition for optically simple surfaces, such as colored paper. I succeeded in making robots which are autonomous, mobile, and use this color recognition sense to augment their behavior. I don't believe I succeeded in reaching one of my primary goals, which was to have Scavbots alter the environment by drawing new trails, because the could not analyze the color of the pen inks.

If I started over, I would more carefully explore different inks and light sources. I would also investigate the use of solenoids to control my pens movements, because this might be a more elegant solution, and would avoid the types of pwm interference I had to work around. I might also find a more durable material than Styrofoam with which to mount my CdS cells and LED's.

This diagram shows a breakdown of the Scavbot system organization



\/ **SCAVBOT** bottom view \/



U-unflitered CdS cell
R-red filitered CdS cell
G-Green Filtered CdS cell
B-Blue Filtered Cds cell
@-green LED w@w-white LED

Robot1	--	G	--	R	B	G<=195	G<=212	R<=35	B<=195	
BOARD	101	235	134	66	230	F	F	F	F	
White paper	86	232	117	53	228	F	F	F	F	
Yellow paper	64	227	77	44	222	F	F	F	F	
Green paper	47	223	61	20	215	F	F	T	F	
Red paper	17	167	17	25	175	T	T	T	T	
Blue paper	27	205	35	17	202	F	T	T	F	
Black paper	7	145	7	7	138	T	T	T	T	
						G<=210	R<=47	B<=210	3<=105	
Orange EXPO ink	80	220	98	68	222	F –15	F 0	F –8	T	
Red EXPO ink	55	198	78	60	205	T -37	F –6	T –25	T	
Red Avery ink	66	230	95	55	216	F -5	F –10	F –14	T	
Blue EXPO ink	88	225	115	25	225	F –10	T –40	F –15	F	
Blue Avery ink	62	229	93	24	223	F –16	T –40	F –7	T	
Black EXPO ink	39	205	75	54	189	T –30	F –12	T –40	T	
Black Avery ink	80	229	110	42	230	F -6	T -24	F 0	F	

Sample data of values from CdS cells (before normalization). One can see both how the different filters block different colors of light, and how different inks and surfaces result in vastly different CdS cell values.