Carnegie Mellon University

Robotics Capstone Project

Concept Design

Friction Force Explorers:

Don Zheng
Neil Jassal
Yichu Jin
Rachel Holladay

supervised by
Dr. David Wettergreen

Version 2.0
November 10, 2016

Contents

List of Figures

1 System Description

The goal of our project is to develop a multi-agent robotic drawing system. Using a team of homogenous robots, we can efficiently parallelize the dull task of drawing. Our system is motivated by the miles of paint needed to maintain airport runways and sports arenas, as well as the widespread use of sidewalk chalk. Below, in Fig.1, we model our overall system graphically, with a more detailed description following. We then enumerate each subsystem in Sec. 3.

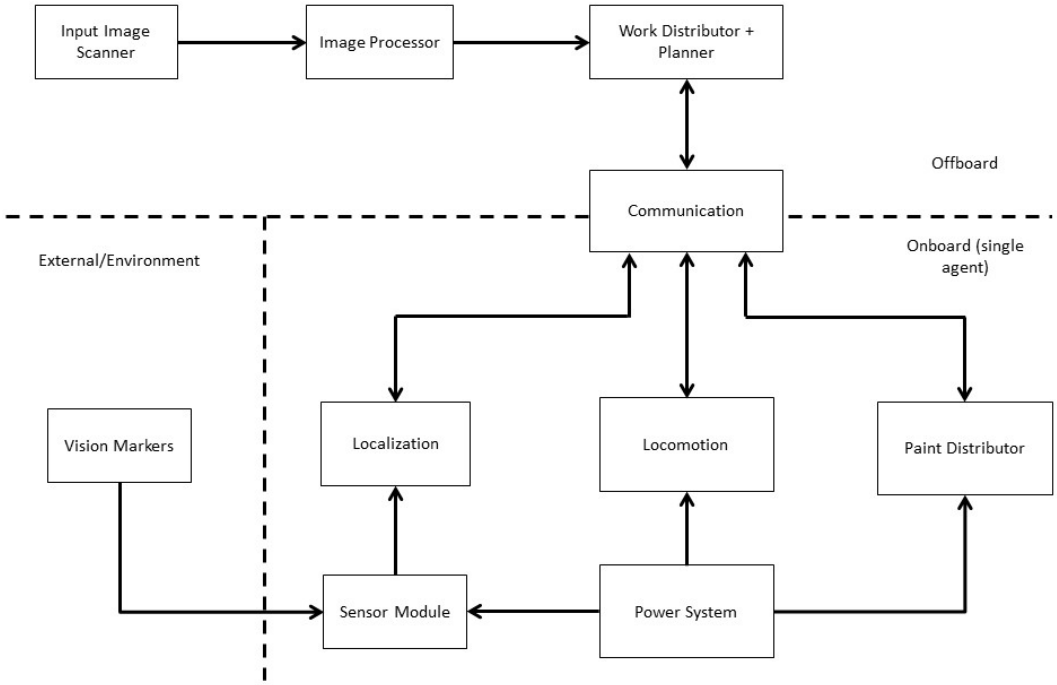

Figure 1: High Level Architecture Diagram

Our system encapsulates three sections, shown in Fig.1 through dotted lines. First, there is a single offboard unit that handles the coordination, planning and user interface. The initial drawing process begins in the offboard unit, where users will generate an input image . Image processing will then occur, where the input is processed into a format compatible by the planner and work distribution subsystem Sec. 8. Moving forward, the planner module will take in updates to current robot progress and use it to update commands it sends to robot agents.

The second section are the robot agents, marked in Fig.1 as "onboard". While the diagram shows one onboard agent, this schema is identical for each agent. Commands are sent to these agents. Communication protocols are able to send information both to and from the offboard work planner to robot agents (Sec. 9). From here, onboard systems are separated into three subsystems: Localization (Sec. 6), Locomotion (Sec. 5), and Paint Distribution (Sec. 4). The paint distributor is involved with all commands regarding engaging and disengaging the writing implement to output paint onto the writing surface. Locomotion is the drive system, which engages wheels and motors used for movement. This subsystem also includes any motion-related safety mechanisms, such as quickly stopping. The Localization subsystem works to determine the position and orientation of an individual robot agent.

Finally our third component comes from environmental setup, which is mainly the situational landmark tools need to localize the robot.

2 Concept Operation

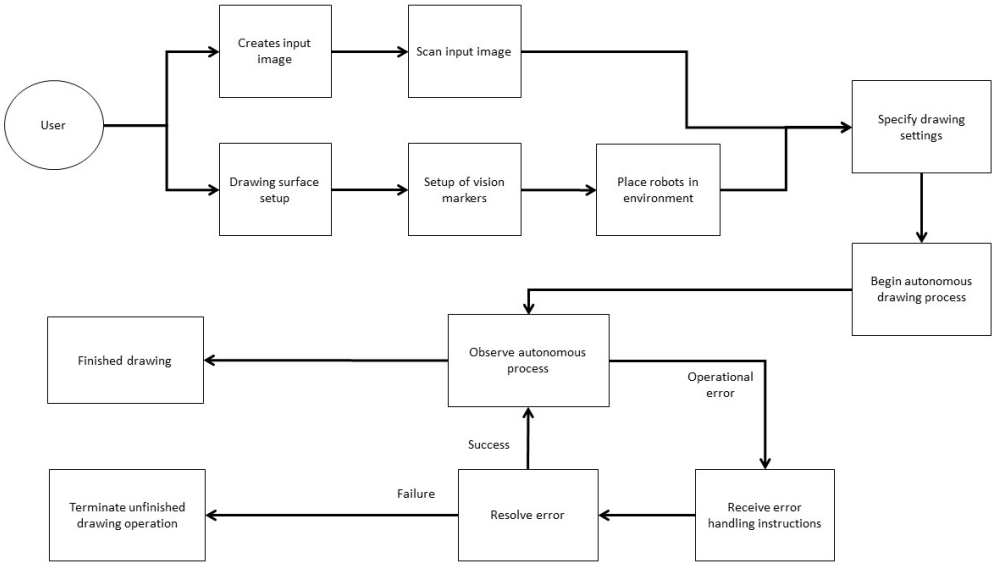

Figure 2: State Machine of Concept of Operations

Above, in Fig.2 we outline the user experience with the system. This is detailed below and our user interface is fully explained in Sec. 10.

The initial setup contains two operations that can be performed simultaneously: adding the image to be drawn to the system, and setup of the drawing surface. Adding the input image involves generating the image, and then scanning it into the system. This satisfies requirements for input the drawing plan (Requirements Specification 5.1, FR9), and a user interface (Requirements Specification 5.1, FR12). Setup of the system involves placing the drawing surface, placing and calibrating vision markers, and finally placing the robot agents within the bounds of the drawing surface. Once both steps are done, the user can enter any required settings for their use, and begin operation. Processing of the input image is done automatically by the system and is invisible to the user (Sec. 7).

Once the autonomous drawing process begins, the user simply observes the robots until they complete the task. Errors will be reported to the user, who then has the option of fixing the issue to continue operation, or terminating the drawing process. The autonomous process satisfies functional requirement FR2.

3 Subsystem Listing

The remainder of this document is split based on the various susbystems. We briefly describe each subsystem here.

Writing Implement (Sec. 4): The mechanism that uses the writing tool to deposit material onto the writing surface.
Locomotion (Sec. 5): This system involves robot motion across the writing surface.
Localization (Sec. 6): All pieces of the robot system involved with determining position and orientation of the robot agents within the drawing space.
Image Processing (Sec. 7): Takes a user-created image and processes into a format usable by the work scheduler (Sec. 8).
Work Scheduling, Distribution, and Planning (Sec. 8): Determines trajectories for individual robots to complete the drawing, based on the user's input image.
Communication (Sec. 9): This subsystem involves all communication between robot agents and the offboard system.
User Interface (Sec. 10): A unified system for user input and interaction into the system.
Power System (Sec. 11): Supplies power to all necessary parts of the robot system.

4 Writing Implement

The writing implement mechanism deposits writing material onto the working surfaces. It manages reloading of the writing implement and ensures that the implement is properly secured. This subsystem can also draw discontinuous strokes and draw in various widths. In the event of writing material depletion, its sensors will detect the occurrence and alert the communication module (Sec. 9).

The writing implement system is modular, so that various writing tools (such as chalk, markers, pens, etc.) can be easily inserted by users. This involves a fixed motor mechanism on the robot, with mounts for each tool that can be locked into place on the robot. The mounts can optionally connect to two motors: One for linear motion to raise and lower the writing implement, and another to rotate the writing material as described above.

Critical Components: Writing implement housing, actuation mechanism, writing material levels sensor, and reloading mechanism.

4.1 Use Cases

4.1.1 Load/Switch Writing Implement

Description: The robot has the ability to allow quick swapping of writing implements. This may be necessary when the current implement has been depleted, or a new implement with different writing properties is desired. The writing implement actuator itself is modular and detachable, so different kinds of writing implements (markers, chalk, etc.) can be used in the same robot.

4.1.2 Actuate Writing Implement

Description: The robot can use its writing implement to make markings on the working surface. It can vary drawing properties such as line continuity and stroke thickness.

4.2 Trade Study

To ensure the robot is suitable for a wide range of drawing applications, we consider three different drawing materials: chalk, markers, and paint. For chalk drawings, the robot can either utilize a traditional chalk or a liquid chalk marker. Markers are usually applied through markers, for example Sharpie Permanent Markers. Typical ways to apply paint is either using a roller or a brush or using spray paint can. The mentioned drawing materials will be evaluated against system requirements in the context below.

As non-functional requirements NFR5 and NFR7 indicate, the robotic system needs to reliably paint drawings that closely match the input image. Compared to chalks and markers, methods for applying paint is often more complex, which makes it difficult to ensure stroke quality. Also, drawing with paint often involve dripping and uneven coatings. Such drawbacks will further defeat our requirement on drawing quality (NFR5.) Therefore, we plan to pursue chalk and markers for drawing materials.

There are two ways of applying chalk drawings: using traditional chalk and using a liquid chalk marker. These two methods are evaluated against criteria including mechanism complexity and drawing quality. Since traditional chalk shortens and flattens while drawing, the designed mechanism needs to account for this potential change in length and provides an extra rotational degree of freedom to unify line width. On the other hand, the mechanism for liquid chalk markers does not need to account for the marker length and requires only one translational degree of freedom. Hence, in terms of mechanism complexity, a liquid chalk marker is better than traditional chalk. However, in terms of drawing quality, both methods are uncertain. Therefore, a prototype will be used to reveal the difference in drawing quality between the two methods and to help us determine which one to use.

4.3 Artistic Sketch

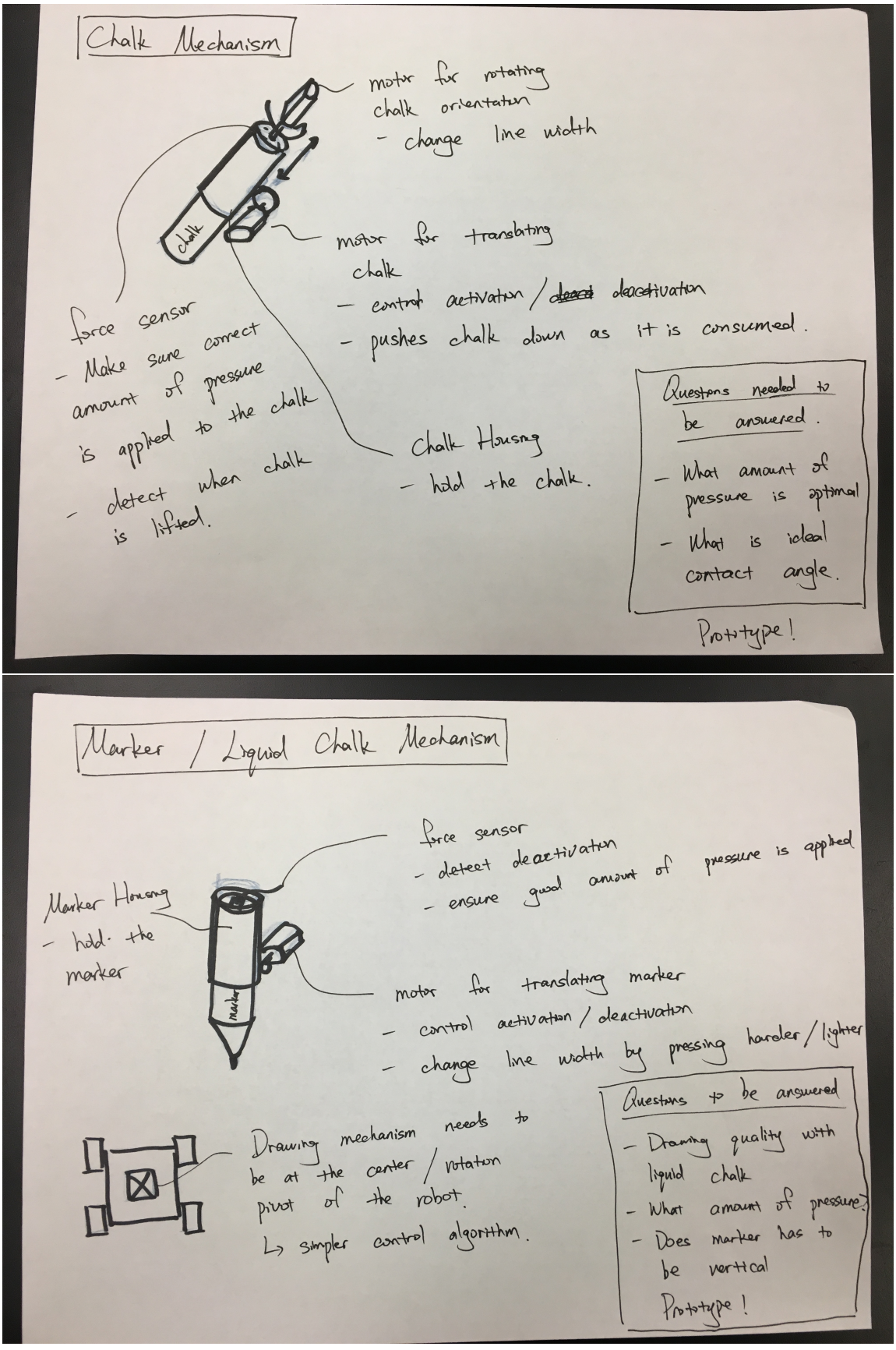

Figure 3: Writing Implement Sketch and Design Justifications

4.4 Requirements Fulfilled

Our modular implement system allows us to easily insert (FR5), remove (FR6) and replace (FR7) tools quickly (NFR14). The motorized control allows the robot to turn the writing tool on and off (FR10). The size of the system is designed to be within weight (NFR3), size (NFR4) and budget (NFR10) restrictions. Through continued prototyping process, we will consider reliability (NFR8) and quality (NFR6) and how we can build in error handling (NFR2).

5 Locomotion

Description: The robot's locomotion system propels each individual robot across the working surface. The robot uses a holonomic locomotion system, it can move in any direction, which allows the robots to execute intricate designs. Mechanum wheels will allow the robots that flexibility, while at the same time preserving stability and ease of control. They also provide enough clearance for the writing implement to work effectively. A driving system that allows for multidirectional movement satisfies motion-related requirements (Requirements Specification, 5.1, FR7). Additionally, the locomotion system contains encoders whose data is used by the localization module (Sec. 6).

Critical Components: Wheels or treads, motors and encoders.

5.1 Use Cases

5.1.1 Locomote

Description: The robot can use its array of motors and wheels to propel itself across the flat working surface. It can make arbitrarily sharp turns and acute curves in order to allow input drawings that incorporate such components.

5.2 Trade Study

One of our functional requirements with the highest priority (FR1) is to be able to move in specified directions with a high degree of accuracy. Additionally, a medium high priority functional requirement (FR 9) is to have a drive control system that enables this movement. Therefore, it is clear from our requirements that locomotion is critical to our robot's performance. When considering locomotion options there are three large categories: aerial, legged and wheeled.

Our drawing occurs on the 2D plane. Therefore, any flight based system would have to constrain one translational degree of freedom and two rotational degrees of freedom. Doing so would require precise control algorithm and large power input. Practically speaking, this overcomplicates the problem for little benefit, and as such rules out airborne locomotion systems.

Therefore, locomotion options are limited to a legged or wheeled system. While a legged system would have the ability to traverse uneven terrain, such capability is not necessary for this project due to the assumption (A3) that the drawing surface is flat and homogenous. Compared to wheeled systems, legged systems usually involve more complicated control algorithms and electrical and mechanical components. In addition, legged systems often fail to ensure stability in the robot body, which for this system holds the writing implement. Considering system complexity and locomotion stability, it can be concluded a wheeled system is best.

In continuing to prioritize functional requirement (FR1), we investigate other possible wheeled drive systems. Four different drive systems are evaluated: differential drive, Ackerman steering, four-wheel steering, and holonomic drive. Each of the four drive systems are evaluated based on mobility, motion accuracy and amount of damage done to the drawing surface.

Differential drive systems typically consist of two independently driven wheels and a non-actuated wheel. Such a drive system has good mobility due to its minimum number of driving wheels. However, the non-actuated wheel often creates unwanted resistance while turning, which both compromises the drawing accuracy and damages the drawing surface. Occasionally, the non-driven wheel can fall into a singularity and ruin drawing quality.

Next is Ackerman steering, where the back pair of wheels is fixed in orientation but the front pair can pivot. This drive system does not suffer from singularities and does not damage drawing surfaces. However, turning with ackerman steering often requires a large turning radius dependant on the wheel

base, which makes sharp corners difficult to draw. Despite a high accuracy of motion and minimal damage to the drawing surface, Ackerman steering has a low mobility and is not feasible.

Another option is four-wheel steering, which is similar to Ackerman steering but allows the back wheels to pivot as well. Even though such drive system can achieve in-place turn, doing so often requires the wheels to pivot in place, which can potentially damage the drawing surface. Hence, four-wheel steering ranks high in mobility and motion accuracy, but low in damage to drawing surfaces.

Holonomic drive involves independent control of four omnidirectional wheels. Such drive system can easily turn in place and does not cause any potential damage to drawing surfaces. The drive system's motion accuracy depends on wheel choice. Holonomic drive is often achieved with either omniwheels or mecanum wheels. Omniwheel does not resist any force in its lateral direction, resulting in low motion accuracy due to slippage. Mecanum wheels, on the other hand, resist sideways disturbances and ensure straight motion due to the wheel's obliquely oriented rollers.

In conclusion, the system's locomotion system will be a wheeled holonomic drive system with mecanum wheels, since such combination best satisfies system requirements and performs best in criteria concerning mobility, motion accuracy, and amount of damage to drawing surfaces.

5.3 Artistic Sketch

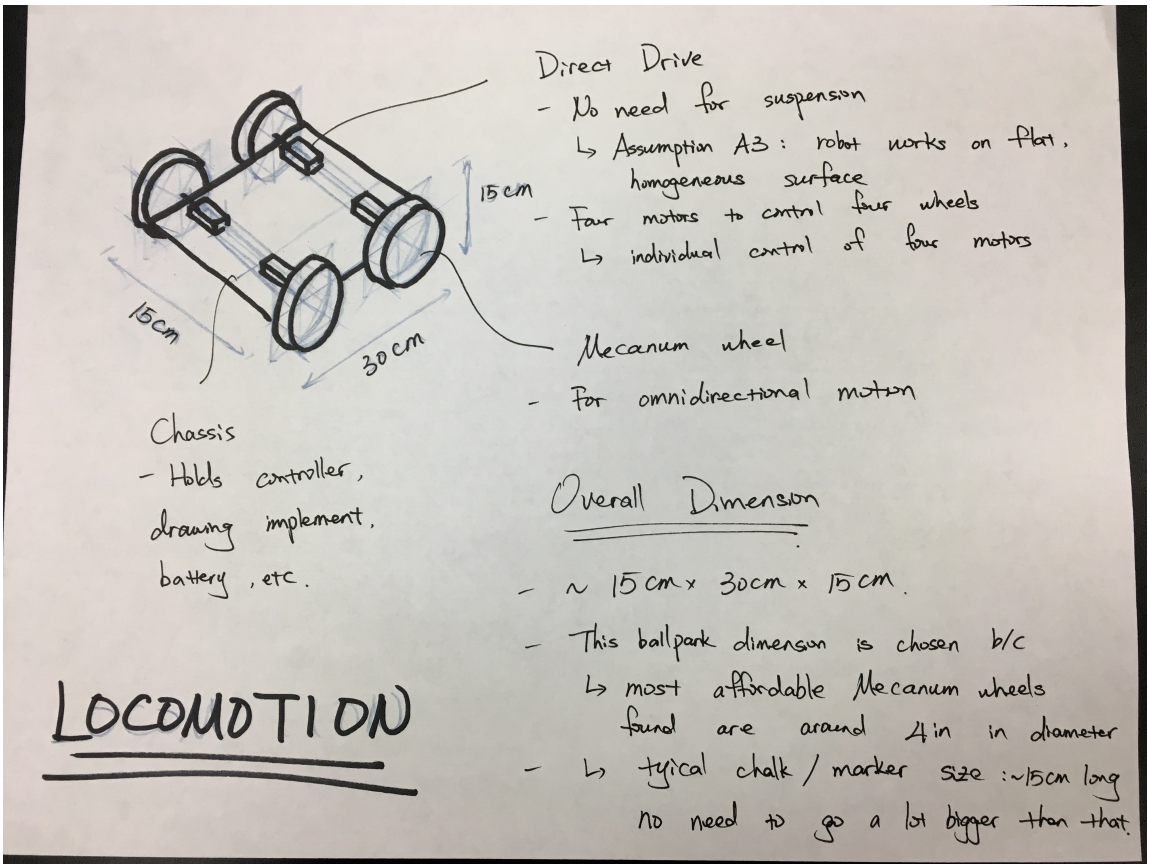

Figure 4: Locomotion Sketch

5.4 Requirements Fulfilled

By using mecanum wheels, we can achieve omnidirectional movement (FR1) and reliable control (FR9). Our emergency stop (FR13) and error handling (NFR2) will be integrated into the drive system through the communication network Sec. 9. Our mecanum wheels have been specificed to stay within our weight (NFR3), size (NFR4) and budget (NFR10) limits. The control of the mecanums will allow us to be efficient (NFR5), due to their omnidirectional motion and speed, while maintaining quality (NFR6), reliability (NFR8) and safety (NFR11). Additionally the omnidirectional control wil allow us positional (NFR12) and rotational (NFR13) accuracy.

6 Localization

The localization system uses an overhead camera and marker (Sec. 6.2) to determine a robot's position and orientation. It uses an overhead-mounted camera with a full, unobstructed view of the workspace and the robot agents. Static markers are mounted in the corners of the workspace, as well as atop the robot workers themselves. The corner markers determine the boundaries of the workspace, to which the input image can be scaled. The system use the robot-mounted markers to determine the position and orientation of each robot in relation to the workspace. It then communicates the positions and orientations of the robots to the scheduling module (Sec. 8). This module directly satisfies local and global localization requirements, as well as indirectly allows for safe and bounded motion from the robots (Requirements Specification, 5.1, FR3, FR4).

Critical Components: Localization algorithm, localization markers.

6.1 Use Cases

6.1.1 Localize

Description: The robot uses a combination of robot-mounted and static environmental markers, as well as a camera mounted above the working surface to determine the locations of each of the robots. This information is used to determine the motion plan of the robot.

6.2 Trade Study

In multi-agent planning, it is important to accurately localize robots' positions and orientations. Keeping in mind limitations in price and ease of use, we come up with two major methods for localization: vision based and marker based. They are described below.

Vision-based localization involves using cameras or other visual sensors to directly obtain information of the environment and localize the robots based on found landmarks in that environment. One example of this is SLAM (Simultaneous Localization and Mapping), often used by autonomous vehicles to simultaneously build maps and localize [1]. With this approach, robots could build small maps of their surroundings and match their locations to features they find in the environment. Benefits of this method include being location agnostic and requiring no additional parts or external setup. Pure vision systems are difficult to calibrate and localization accuracy can depend heavily on static surroundings, but this is relatively easy to guarantee given the requirements of the system (Requirements Specification, 2.3, 2.4 A1).

The other choice of methodology is marker based. Using markers placed around the drawing surface, robot agents can quickly locate these markers and their positions relative to each marker, and consequently triangulate their positions and orientations. While requiring additional setup and more parts to calibrate than vision based localization, existing technology makes it convenient and cheap to get marker based localization working. One example of a marker-based localization system is AprilTags [2], which can be described as 2D barcodes placed in a scene. Marker-based localization can be further classified into two subcategories: passive and active. Passive markers do not output any information and exist for the robot agents to observe and triangulate accordingly. AprilTags is an example of the passive marker system. On the other hand, active markers will "look at" robot agents to determine where the agents are, rather than the robots searching for markers. While less common, active systems behave well in conditions when the markers may not always be easily visible to robot agents [3].

Given the convenience and ease of use of marker-based localization, it is clearly the better choice. Additionally, since the workspace is limited to an indoor, flat, and homogenous area in our requirements

specification (Requirements Specification, 2.3, 2.4 A1) we can reliably count on overhead line-of-sight as opposed to having the robots observe or be observed by the passive or active markers. An indoor space also allows for easy mounting of an overhead camera. Consequently, we have decided to pursue a passive marker system observed from above by a single camera.

6.3 Artistic Sketch

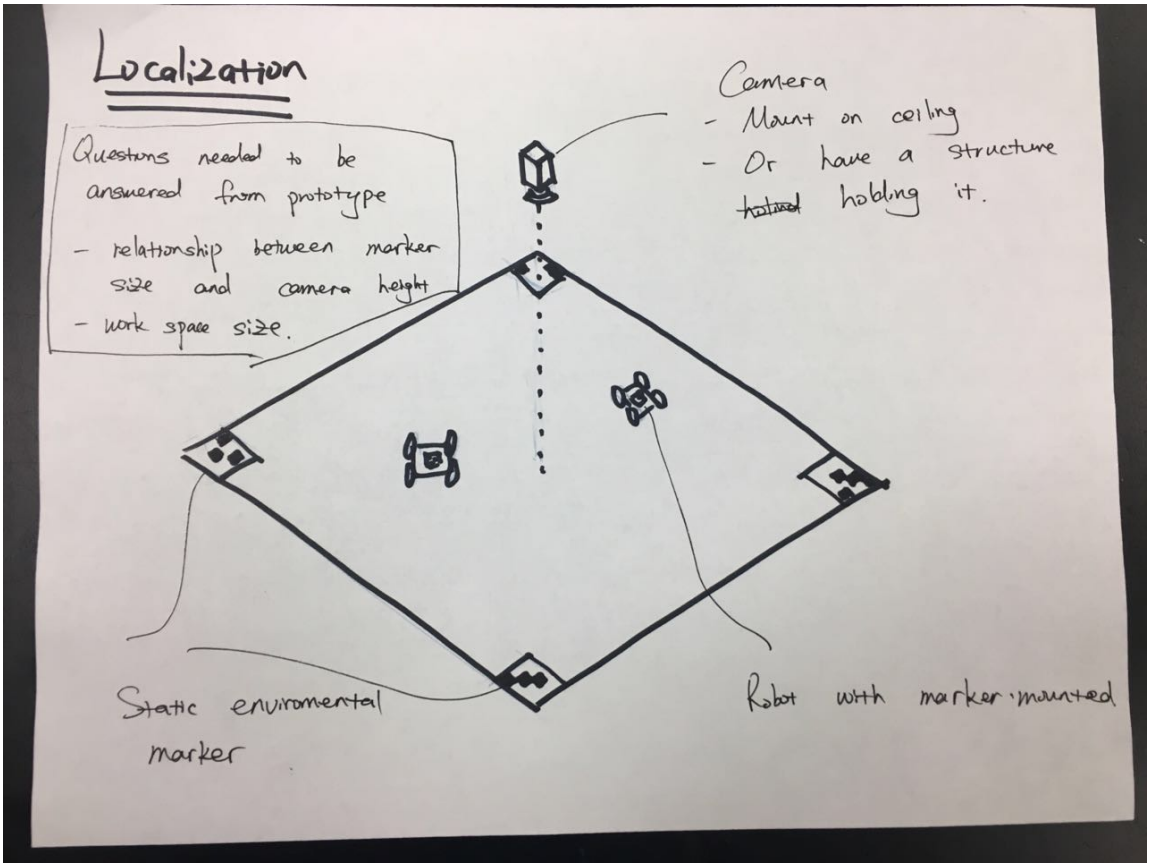

Figure 5: Localization Sketch

6.4 Requirements Fulfilled

Our localization system allows the robot to be autonomous (FR2) and localize (FR3), which gives them the information to stay within workspace boundaries (FR4). Our localization will be refined to account for error handling (NFR2). Precision in localization will enable our robot to produce a quality image (NFR6) with positional (NFR12) and rotational accuracy (NFR13) that is coordinated with the rest of the system (NFR9). As we continue with our overhead camera system we will develop reliability (NFR8) and safety (NFR11) into the system. We believe that our simple overhead camera will be within budget (NFR1).

7 Image Processing

This subsystem takes a user-provided image as input, and processes in such a way that it can be read in and used by the work scheduling subsystem (Sec. 8). The input from the user incorporates a front-end user interface, satifying requirements related to user interaction, which are defined below in Section Sec. 7.3.

The image processor takes input from the image scanner (Sec. 7) and produces information that is recognizable by the planning module (Sec. 8).

Critical Components: User interface, image sensor, Image processing algorithm.

7.1 Use Cases

7.1.1 Input User Image

Description: The system can take in an image provided by the user, and from it determine what the workspace should look like upon completion of the task. The user also inputs parameters such as scale, resolution, and possibly color.

7.2 Software Architecture

This software system was designed based around taking user input in the form of an image, and processing it into a format best suited for use by the work scheduling subsystem (Sec. 8).

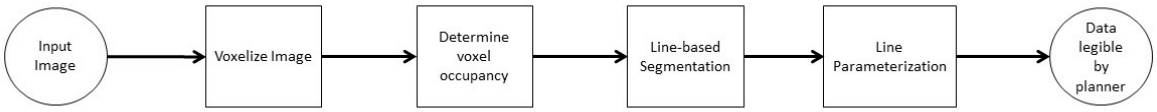

Figure 6: Image Processing Software Subsystem

The image processing pipeline (Fig.6) takes the image scanned by the user, and parses it into a form readable by the work planner. The goal of this software system is to create a series of lines that describes the image. To do this, the system first creates an occupancy grid of the input via voxelization. For a black and white image, the occupancy grid determines black or white. In the case of color, voxels are assigned color based on the image values inside of the voxel.

Once voxelization is complete, lines are formed from the voxels through a nearest-neighbor search. These lines are simply a series of voxel squares that pass through the image. In order to preserve curvature of lines from the input, the grid-delineated lines are reparamterized into paths using splines. These paths are then sent to the work planning module.

7.3 Requirements Fulfilled

Our input system, as seen in Fig.6 allows a user to input an image (FR11). Each component is designed with error handling (NFR2) and reliability (NFR8).

8 Work Scheduling, Distribution, and Planning

Given the output from the image processor (Sec. 7), and parameters such as the number of robots and the size of the workspace, this module determines the work that will be distributed to each robot. The software architecture (Sec. 8.2) section explains the operational flow of the system..

Critical Components: Scheduling and distribution algorithm.
Further Research: Scheduling multiple robots in a loosely coordinated joint task presents an interesting research question that we are continuing to look into.

8.1 Use Cases

8.1.1 Decompose Plan

Description: The system generates an efficient distribution of work to each of the individual robots. The work distribution takes into account the amount of writing material needed, thereby limiting the need for reloading. It is also based on distance that the robots must cover and the speed at which they must travel to ensure that robot agents idle for as little time as possible.

8.1.2 Schedule Lines

Description: The system can divide each robot's specific tasks into subtasks, and determines the order in which they will be completed. This order ensures that the robot spends little time traversing unnecessary distance and completes its tasks in a short amount of time.

8.1.3 Plan Movement

Description: Each robot is capable of determining how to actuate its motors to get from its current location to a desired location. By using its motion model based on the properties of the robot's locomotion system, the system can queue motor values and adjust for noise and error along the way.

8.1.4 Detect Collisions

Description: The system can use a combination of localization sensors and motor encoders to determine if any robot has encountered an obstacle.

8.2 Software Architecture

A key aspect of our software system is distributing and planning the work to the robot agents, modeled in Fig.7. Two of our nonfunctional requirements are to be efficient and have the robots coordinate with each other (NFR4, NFR9). These two objectives inform the planning pipeline. We split the planning and coordination into two separate problems, allowing us to frame coordination as a scheduling problem [4]. Hence our work distribution and planning can be handled offline while our coordinate occurs online.

From our image processing Sec. 7.2, we recieve processed image data. Using this data, we compute the length of every individual line to be drawn. Guided by the assumption that line length correlates to amount of work done and the time to perform that work, we then use those lengths to load balance when assigning which lines should be drawn by each robot. Once each line has been assigned to each robot, each robot has a complete picture of their work and can be assigned an ordering to their lines. To limit wasteful movement, the robots order lines greedily: having completed one line, they pick their next lines (from their assigned set) based on which line has the closet endpoint to the line they have just finished. Having ordered lines, we now can determine each robot's set of paths.

Next, we briefly describe a sketch of our coordination mechanism. Each robot has a queue of paths, based on the set ordering, to execute. Given that a robot has not finished all of it's assigned paths, it removes a path from the queue. The path is then uniformly timed and converted into a trajectory. Given timing information, the system can check for path collision between any trajectories currently being executed. If the trajectory is not at risk of collision, then it can be executed - and is sent to the robot via the protocol mentioned in Sec. 9.2.

If the trajectory is in collision with a currently executing trajectory, then the trajectory being processed defers to the one being executed. The timing of the processing trajectory is adjusted by adding a pause to avoid collision. Given this modified timing we can then execute it.

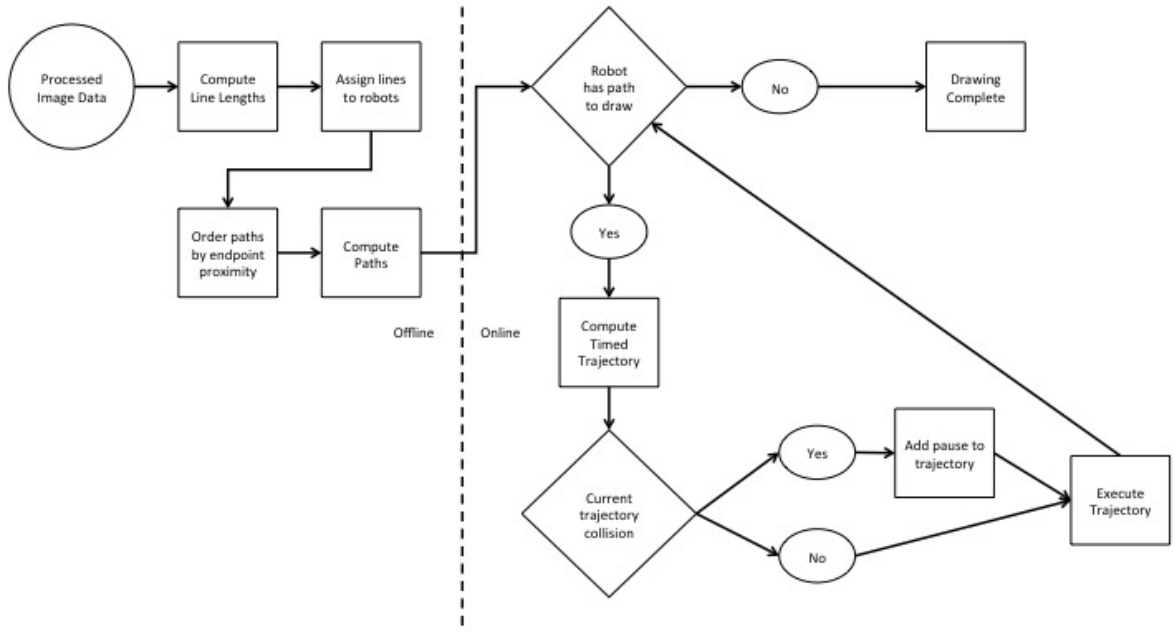

Figure 7: Planning and Scheduling Software Model

Once all paths have been drawn, the drawing is completed.

8.3 Requirements Fulfilled

Our planning framework allows the system to be autonomous (FR2) and will enforce that the robots stay within bounds (FR4). The planning framework will also control the writing implement on/off switch (FR10) and keep track of total system progress (FR12). Our planner is designed to parition work evenly for efficiency (NFR5) and carefully plan for quality (NFR6) and reliability (NFR8). The online component of Fig.7 coordinates the robots to avoid collision(NFR9). Additionally as we flesh out the rest of the system we will build error handling into the software (NFR2).

9 Communication

The communication module is the link between the offboard system and the individual robots. To facilitate real-time changes in the working schedule, communication will be speedy and reliable. Potential communication protocols include WiFi and Bluetooth. Communication between the offboard system will allow individual robots to know their progress relative to the entire drawing (Requirements Specification, 5.1, FR10).

Critical Components: Antennae, wireless communication protocol.

9.1 Use Cases

9.1.1 Communicate and Parse Data

Description: Individual robots report sensor readings to the offboard central computer, and the central computer sends updated localization information and schedules back to the robots.

9.2 Software Architecture

The communication diagram (Fig.8) describes how an individual robot communicates with the offboard path planning system. The system can be described by communication in four categories: motion commands (green), writing tool commands (blue), error handling (red), and localization (yellow).

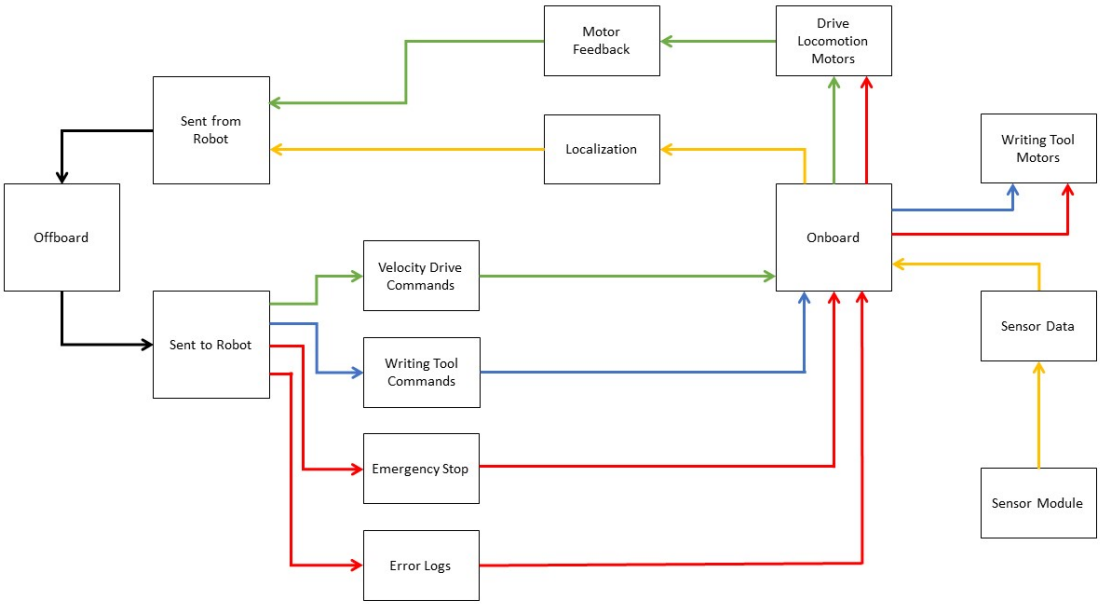

Figure 8: Communication Software Diagram. Green are the motion commands, Blue are the writing tool commands, Red is the error handling system, and Yellow is the localization system

The offboard system will use the planning algorithm to determine motion plans for the robot systems. It determines and then sends velocity commands to the onboard system. The onboard system passes these commands to the drive motors for motion.

To command the writing tool, the offboard system also decides how to move the writing implement based on localization of the robot systems. Similar to velocity commands, the onboard system receives commands for the writing tool, and commands the system accordingly.

Localization receives input from the sensor module, and computes position and orientation. This data is then sent back to the central offboard module to update planning and scheduling.

Error logs are generated by the offboard system based on localization and planning, and sent to each robot's onboard system for appropriate reaction. This includes the emergency stop, which will immediately shut down writing tool and drive motors.

9.3 Requirements Fulfilled

We believe that by partitioning communication as seen in Fig.8, we will have reliable communication (FR8) and by using an offboard system for communication, each robot will know the progress of the drawing (FR12). Also included in software architecture is the provision for an emergency stop (FR 13) and error handling (NFR2). As we iterate on this design we will continue to consider reliability (NFR8), budget (NFR10) and safety (NFR11).

10 User Interface

The user interface provides a unified system for user input. This is the system with which the user specifies the input image and monitors the robots' progress. The system will also display any error messages or anomalies detected, and how the user should address them. There will also be a system-wide kill switch in case of emergencies.

Critical Components: Screen, input device.
Planned Prototype: We plan to sketch our a prototype of our user interface.

10.1 Use Cases

10.1.1 Display Information

Description: The system is able to show a user information pertinent to system operation, including but not limited to the location of the robots, their battery level, amount of the task completed, estimated time of completion, and any existing obstructions or anomalies.

10.2 Requirements Fulfilled

Our prototype will be designed to allow the user to input a drawing plan (FR11), cancel progress in case of emergency (FR13) and provide an intutive experience (FR14). Additionally this user interface must come complete with documentation (NFR1) and error handling (NFR2) and be reliable (NFR8) and within budget (NFR10). A low priority requirement is also to develop a user interface through a mobile app (NFR7).

11 Power System

The power system supplies power to the rest of the system. Each robot has an onboard battery that is small and light enough to satisfy the size and weight requirements (NFR3, NFR4), but also provides enough uptime to last a entire drawing session. The power system will satisfy battery life and contribute to portability requirements.

Critical Components: Battery, voltage regulator modules.
Planned Trade Study: We are continuing to investigate the power system and will have an evaluation of our research soon.

11.1 Requirements Fulfilled

In prototyping our power system, we will have to satisify the kill switch requirement (FR13) while also using a battery that will allow each robot to operate for at least half an hour (FR15). We will also balance having a power system that allow for smooth error handling (NFR2), is within our weight and size restrictions (NFR3, NFR4), budget (NFR10), and is reliable (NFR8) and safe (NFR11).

12 Installation

In addition to the robot system itself, the system installation will require:

- Adequately sized indoor drawing surface
- Localization markers
- Overhead camera mount
- Writing implement

We will be using an indoor obstacle-free surface in accordance with our requirements specification (Requirements Specification, 2.3, 2.4 A1) which placed outdoor, non-homogenous drawing surfaces and obstacles out of scope.

We will set up the localization markers on the corners of the working surface, as well as atop the robots themselves. The system can then identify and calibrate to the localization markers to track the working surface and the robots. We will also adjust the camera mount so that it allows the system's camera a full, unobstructed view of the working surface. Finally, we will place the robots at set locations on the working surface and generate the input image.

13 Requirements Table

Requirement	Section
FR1	Sec. 5
FR2	Sec. 6, Sec. 8
FR3	Sec. 6
FR4	Sec. 6, Sec. 8
FR5	Sec. 4
FR6	Sec. 4
FR7	Sec. 4
FR8	Sec. 9
FR9	Sec. 5
FR10	Sec. 4, Sec. 8
FR11	Sec. 7, Sec. 10
FR12	Sec. 8, Sec. 9
FR13	Sec. 5, Sec. 9, Sec. 10, Sec. 11
FR14	Sec. 10
FR15	Sec. 11
NFR1	Sec. 10
NFR2	Sec. 4, Sec. 5, Sec. 6, Sec. 7, Sec. 8, Sec. 9, Sec. 10, Sec. 11
NFR3	Sec. 4, Sec. 5, Sec. 11
NFR4	Sec. 4, Sec. 5, Sec. 11
NFR5	Sec. 5, Sec. 8
NFR6	Sec. 4, Sec. 5, Sec. 6, Sec. 8
NFR7	Sec. 10
NFR8	Sec. 4, Sec. 5, Sec. 6, Sec. 7, Sec. 8, Sec. 9, Sec. 10, Sec. 11
NFR9	Sec. 6, Sec. 8
NFR10	Sec. 4, Sec. 5, Sec. 6, Sec. 9, Sec. 10, Sec. 11
NFR11	Sec. 5, Sec. 6, Sec. 9, Sec. 11
NFR12	Sec. 5, Sec. 6
NFR13	Sec. 5, Sec. 6
NFR14	Sec. 4

References

[1] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.

[2] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *ICRA*, pp. 3400–3407, IEEE, 2011.

[3] R. Cassinis, F. Tampalini, and R. Fedrigotti, "Active markers for outdoor and indoor robot localization," *Proceedings of TAROS*, pp. 27–34, 2005.

[4] P. A. O'Donnell and T. Lozano-Pérez, "Deadlock-free and collision-free coordination of two robot manipulators," in *ICRA*, IEEE, 1989.