



Université du Québec
École de technologie supérieure



DESIGN REPORT

IGVC 2004



Table of content

Introduction 3
This year’s improvements 3
Organization and design process 3
 Documentation standards 3
 Administration standards and procedures..... 4
 Communication tools 4
 Design process..... 4
Safety 4
Mechanical overview 5
Electrical overview..... 5
 Electrical box..... 5
 Technologies 6
 CAN BUS..... 6
 Embedded PCs 6
 DUAL power source type (DC/AC)..... 6
 Motors / drives 6
 Microcontrollers..... 6
Problems and solutions..... 7
 Reliability 7
 Signal integrity 7
 RF Interference..... 7
 Motor’s drive power handling 7
Sensors 8
 GPS..... 8
 Compass..... 8
 Range finder 8
 Camera 8
Efficient use of power 8
Motion controller..... 8
Software overview..... 9
 Process and standards..... 9
 CMMI model..... 9
 Caasi development process and lifecycle 10
 Phases and milestones 10
 Task scheduling and team management 10
 Path planning and obstacle avoidance 11
 Waypoint Navigation 11
 Finding an optimized list of waypoints 11
 Advising the robot on its trajectory 12
 Archie 12
 Video library abstraction layer 13
 Vision algorithms 14
 Line detection..... 14
 Obstacle detection 14
 Response time 14
Vehicle value..... 15
Organization chart..... 16



INTRODUCTION

Capra and the “École de technologie supérieure” are proud to present its newest version of “Mentis”. Numerous technological additions and complete re-design of both the electrical systems and software architecture brings “Mentis” to a new level.

THIS YEAR’S IMPROVEMENTS

A new artificial intelligence architecture, named “Archie”, was developed to enable modularity, platform independence and scalability. Several abstraction mechanisms have been introduced in this year’s software architecture. Designs are now completely independent of the technology used. An interface always protects the core application from making hardware or API specific calls.

In order to improve the reliability of inter-device communications, we have decided to implement a standard used across the automotive industry: the CAN BUS. This technology allows us to inter-connect an almost unlimited number of devices and have them communicate efficiently. Because of the mobility of the robot and an increasing need for reliability, virtually all internal connections and circuits have been moved to printed circuit boards. This enables quick repairs/replacements and cleaner design. The electrical system was also moved to an aluminum casing reducing interference coming from computers, power supply and external signals.

ORGANIZATION AND DESIGN PROCESS

Documentation standards

To facilitate maintenance and to accommodate new members, a rigorous documentation standard was adopted. Every new program must now comply with the Doxygen documentation standard which allows automatic generation of the documentation. Using Doxygen, we have accurate, up to the minute, documentation of our source code available in several formats including HTML, RTF and text.



Administration standards and procedures

In order to fulfill our responsibilities as an autonomous student club, several forms and procedures have been developed to keep track of the different administration tasks. We have a thorough record of all financial transactions and every meeting.

Communication tools

Communication between developers was improved by a new forum, available on our website. This tool helps us keep an archive of all discussions that resulted in decisions. They are therefore well documented and available to all. Also, the forum allows members to work remotely without being cut from the decision making process. Developers can easily search or browse through the discussions to extract the information she or he needs.

Design process

As experience showed us, experimental software and electrical systems are remarkably complex to design. In order to cope with this fact, we opted for the iterative approach. We first identify and document the requirements of this year's improvements. Secondly, we prioritized them to facilitate selection. Once this operation is completed, solutions for each problem are identified and examined. The most appropriate one is selected and broken in small steps. Each step is independent and complete, as it will be the starting point of the next one. This insures that high risks decisions will be validated early in the development process. This design philosophy is also applied during the solution selection process: this year's design needs to be next year's foundation. Therefore, all of our software and electrical components are designed to be reusable and scalable.

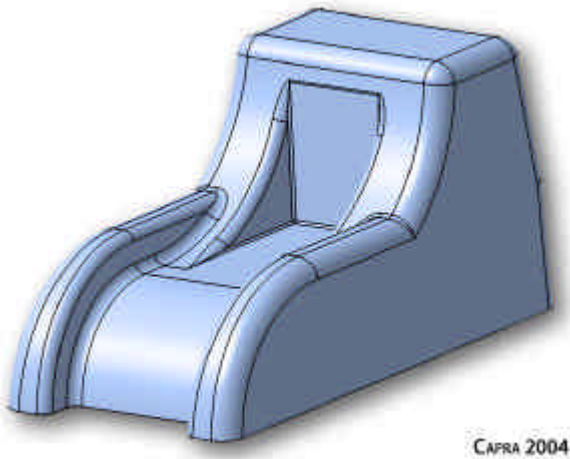
SAFETY

As we acknowledge the importance of safety, measures were implemented to ensure the safety of the developers and of the robot itself. The first and most important measure is the hard-wired emergency stop. This push button controls the relay that feeds current to the drives and motors. If the emergency stop is activated the alimentation to the motors is immediately cut. Each of our motors is equipped with a built-in mechanical break. This spring activated break is release when power is supplied, so when power to the motors is cut, the breaks are automatically applied, thus stopping the robot almost immediately. In addition to this safety measure, a simple circuit, coupled with a remote car starter allows us to safely stop the robot from a distance of up to 500 feet.



Another safety feature is our controller watchdog. If the controller stops responding, the emergency stop relay is immediately activated to prevent erroneous data to be sent to the drives, resulting in a complete stop of the robot.

MECHANICAL OVERVIEW



To improve the mechanical efficiency of our robot, we noticed that a new fiberglass hull was necessary in order to answer all of our requirements. We started the design by drawing a new shell using CATIA. Students from St-Jérôme College, who specialized in composite structure, fabricated the shell.

Overall robot dimensions and characteristics:

- Length: 51 inches
- Width: 24.5 inches
- Height: 34.5 inches (60 with camera)
- Weight: 200 pounds

The vehicle was designed to climb a ramp with a gradient of 15%. To ensure that we would have enough torque, wheel chair motors coupled with a reducing gearbox and relatively small wheels were used. When tested, the robot successfully climbed a 20% slope. This limitation was due to the smoothness of the surface used so we can expect better performance on rough terrain.

ELECTRICAL OVERVIEW

Electrical box

To protect the electrical components from external elements and interferences, we conceived an electrical box. The enclosure has a dozen of connectors on the back while the front section holds the power switches and the PC I/O ports. This casing has helped us increase the reliability of the electrical components.

CAN BUS

This year, we chose to implement a new communication network for some of our sensors. This network, which will be our standard for connecting new devices, uses the CAN Bus protocol. We changed the standard DB9 connectors for the more user friendly RJ45 and we used the 2 extra wires of the CAT5 cable to bring power to our sensors. The bus runs at 1Mbps. This bus was selected for its high fault tolerance and its performance in harsh conditions. Both embedded PCs are connected to the bus using cards designed by our team (one ISA, one PC104). The cards use the SJA1000 CAN Controller from Phillips, which is very popular in manufacturers of CAN cards.

Embedded PCs

To efficiently process the data produced by the camera, the range finder, the compass and the GPS, a powerful computer is needed. For this task, the Matrox 4Sight, a P-III 550MHz, was selected as an affordable yet powerful solution. It is responsible for managing all sensors and generating the “virtual map” which is then used to take decisions. A second computer, a Pentium 100MHz, is used as the controller. The real time capabilities of this PC combined with its robustness makes it a perfect candidate for controlling the drives and motors.

DUAL power source type (DC/AC)

Two industrial type power supplies are responsible of powering both computers and all of our sensors. The first power supply is an AC/DC computer power supply while the second is a 24VDC industrial computer power supply. They provide the different voltages (+/-5V, +/-12V) needed by the various electronic components. Because the Range Finder requires a 24VDC supply, we are using an 18 to 36V DC-DC converter to provide the necessary voltage.

Motors / drives

Two 24VDC wheelchair motors bring mobility to the robot. Although both of these motors are able to run at 1500 RPM, a gearbox is used to divide by 18.2 the speed of the motor, bringing the maximum speed of Mentis to 3.75 mph. Two encoders, installed directly on the motor’s shaft (before the gearbox), gives us an extremely precise feedback which is controlled by two drives.

Microcontrollers

Some of our systems, including the compass and the controller PC, are connected to the CAN BUS. The compass uses an ATMEL microcontroller to manage the communications. We choose the ATMEGA series of



microcontroller for its amount of RAM and its “in-circuit programmable” feature. To handle the communication over the CAN BUS, we used the MCP2515 IC from Microchip. Both microcontrollers interact through their respective SPI ports.

Problems and solutions

Reliability

At the 2001 event, severe reliability issues came up. In order to avoid such problems, we completely redesigned the robot focusing on this aspect. New battery and drive locking connectors were installed, mechanical and electrical systems were tested thoroughly before being mounted and the all interconnections were reorganized. Also, most parts were design for easy replacement.

Signal integrity

After the initial testing phase of the CAN BUS interface card, some signal integrity problems appeared: the signal was rising too rapidly causing ringing. Back on the design boards, we've identified a mean to lower this rising time while keeping our targeted 1Mbps transfer rate and implemented successfully.

RF Interference

Interference from the remote control caused the remote emergency stop to malfunction. Further research using a spectrum analyzer demonstrated that the emergency stop was operating near the harmonics of the remote control. Considering the importance of this problem, a decision to completely replace our design with a commercial product, which is more robust and uses completely different frequencies, was made. Better RF immunity was added and a new remote control, operating under a different frequency, is now used. The problem is now completely solved and the robot is safe again.

Motor's drive power handling

Previous drive designs limited the continuous current to 12.5A. This prevented the motors to get enough power for the initial startup: it needed a push. The new models now supply up to 30A while not limiting the current peak. Mentis is back on track.

<i>GPS</i>	CMC Electronics graciously supplied the GPS module. It supports DGPS and is connected to the 4Sight using a serial port.
<i>Compass</i>	The electronic compass is used to sense the rotation of the robot. We've interfaced the compass module to the CAN bus using a microcontroller.
<i>Range finder</i>	The range finder is a SICK LMS 291 with fog correction. We've chosen an 8 meter radius scan and a millimeter resolution. The range finder gives us the measured distance at every 0.5° over 180°. The range finder is connected to the vision embedded PC through a serial port running at 38400 bauds.
<i>Camera</i>	The camera used on the robot grab images in 640x480 color format. Those images are sent to the vision PC through a FireWire connection. The camera is equipped with a wide-angle lens. The approximate viewing distance is 10 feet.

Efficient use of power

The robot has two 24V battery packs, one is used to power the motors and the other is used to power the

Devices	Volts				
	24,0 V	12,0 V	5,0 V	-12,0 V	-5,0 V
LCD Monitor		2,1 A			
Matrox 4Sight-II		5,8 A			
Embedded PC+HD+FD			1,0 A		
Sensoray 425			0,2 A		
CAN ISA Card			0,1 A		
Drives (2)		0,5 A	0,5 A		
SICK Range Finder	0,833 A				
GPS		0,15 A			
Digital Compass & uC			0,15 A		
Emerg. Stop (Car Rem. Starter)		0,1 A			
Accessory		0,5 A	0,5 A	0,5 A	0,5 A
Total Ampere:	0,833 A	9,15 A	2,45 A	0,5 A	0,5 A
Total Watt:	20,0 W	109,8 W	12,3 W	6,0 W	2,5 W
Converter Efficiency	70,00%	Typ.			
Watt Needed	28,6 W	156,9 W	17,5 W	8,6 W	3,6 W
Total Watt:	215,1 W				

electronic components. The electronic components of our robot are using a total of 215 Watt of power. This gives us about 2 hours of operating time with 18Ah batteries. The motor's batteries can run up to three hours.

Motion controller

The second embedded PC is named the "motion controller" and is seen as a virtual device to the intelligence architecture. This device "drives" the

robot along the planned trajectory and gives the current position and direction when required. This system is



running in real-time and is mainly a path-planning controller. It manages complex trajectories, sent by the artificial intelligence PC and executes all steps necessary to bring the robot to its destination. Independently of the conditions, it can follow a path within less than 4 inches from the planned line. A path-smoothing algorithm is applied in order to prevent unwanted stiff turns.

The position is calculated using the encoders and a dead reckoning algorithm. Although the precision varies with terrain type, it can be precise when calibrated accordingly. Since the errors are cumulative the position information is validated using the GPS and compass. For the navigation challenge, the GPS is essential to increase the precision along the path.

The control algorithms are based on a linear dynamic model of the vehicle to calculate the precise current to apply. Two separate controllers collaborate to adjust the linear and angular speeds. Because of the “wheel chair like” mechanical design, the robot can make stiff turn and even rotate around a single point.

SOFTWARE OVERVIEW

Process and standards

This year, our team has improved his development process by following known standard of software engineering. Our objective was to use a “systematic, disciplined and quantifiable approach¹” in most activities to achieve better quality in our product with respect of budget and schedule. The software project code name is Caasi and includes the new software architecture and algorithms.

CMMI² model

To improve our process, we choose to follow the CMMI model and try to gain capabilities at level 2. This whole year, we took care to develop each capability to obtain better performance and robustness in our product and reduce inherent risk of software development. Those capabilities are:

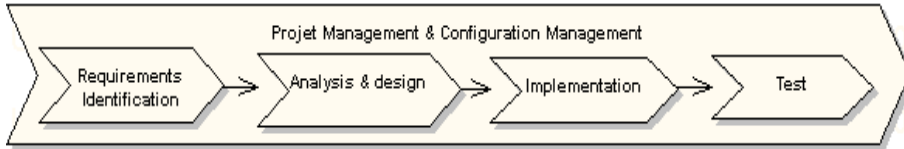
- Requirement Management
- Project Planning
- Configuration management
- Project monitoring and control

¹ IEEE 610.12 – 1990, Standard Glossary of Software Engineering Terminology. Redesignation of IEEE Std 729- 1983

² Capability and Maturity Model Integration by the Software Engineering Institute, <http://www.sei.cmu.edu/cmmi/>

Caasi development process and lifecycle

The software lifecycle that we choose to use is an iterative and incremental process as describe by RUP³. This lifecycle help us constantly review and refine each software discipline involve in our project and insure that

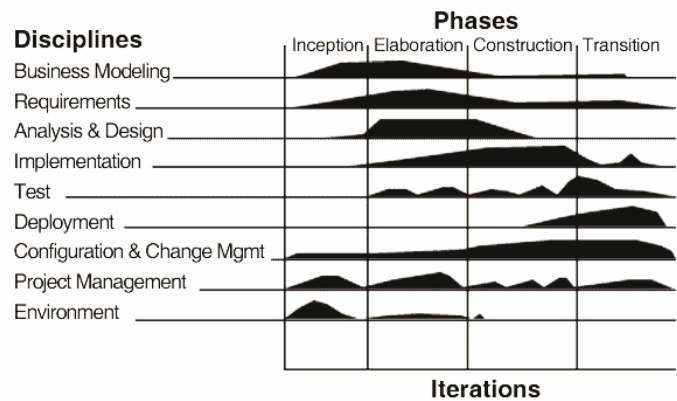


schedule was respected. It also allows us to verify if customer needs was achieved in order to meet requirements.

Iterative process is divided in phase that contains one or more iteration following the process explained in figure 1 and figure 2. All iterations are managed in accordance to project and configuration management activities.

Phases and milestones

Caasi project can be divided in four phases: Inception, Elaboration, Construction and Transition. At the end of a phase, a new milestone is reached. Each milestone evaluates the last phase to know if the selected strategy was correct in consideration of objectives and scope of the project. Here a list of phases and corresponding artefacts produced.



First phase: inception (1 month)

- Vision document
- Software Requirements Specification

Third phase: construction (5 month)

- Implementation of architectural design
- Implementation of vision library
- Implementation of virtual map
- Expert systems oriented blackboard

Second phase: elaboration (2 month)

- Architectural design
- Vision library design
- Choice of technology

Last phase: transition (1 month)

- Outdoor test for vision library
- Additional test for path planning

Task scheduling and team management

The situation at the École de technologie supérieure implies that our team suffer staff change each season. Indeed, a lot of untrained resources want to join our team for specific “short” project. Also, some trained staff



member must quit team to make compulsory internship. Facing this constant change of resources, staffing has been planned for all iteration in accordance to the iterative process. Projects are assigned with respect of individual competencies and availability. For new staff integration, mentoring approach was used. Each week, individual effort was measure in percentage of project completion and in hour of work. With those statistics, project assignment was easiest to do and more accurate for each iteration.

Path planning and obstacle avoidance

The chosen solution for trajectory planning is a blackboard oriented expert system. To evaluate the environment, the raw data collected from sensors must be interpreted by several algorithms to detect barrels, lines and other objects. Those objects are then stored in a multimodal map, used by specialized experts to determine which of the candidate positions would be the best path to choose. To be selected, a candidate point has to be accepted by every experts, hence the expert system.

Waypoint Navigation

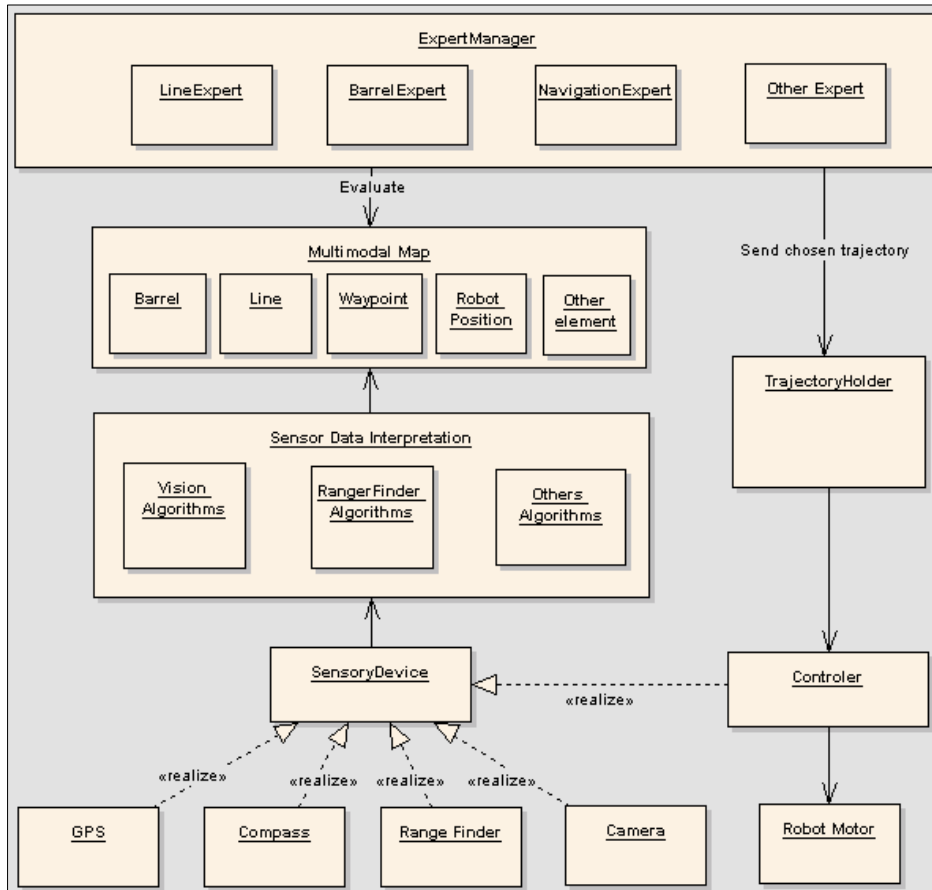
Finding an optimized list of waypoints

In order to reach as many points as possible, the artificial intelligence of the Capra robot has to create an optimized route, to tell the vehicle how to reach those waypoints. That task is given to the waypoint expert, according to the Blackboard system of problem solving. As soon as the expert receives new coordinates to reach, it tries to order the sequence so the route length is minimized. The algorithm used is the Depth-First graph exploration algorithm. First, a solution chart is built with every possible order of reaching waypoints. The result is that the chart gives a tree of $n!$ leaves, where n is the number of waypoints in the course and where every node represents a waypoint. The top of the tree represents the start position of the robot. Once the tree is virtually built, the algorithm starts running into the extreme left branch and calculates the distance the robot would run in the sequence created by the nodes. Once it has done a first head-bottom course, the sequence compares the complete sequence with the next sequence given by the second extreme left course of the tree. A simple algorithm would look at every possible order of waypoints, which means $n!$ possible trajectories. To optimize the algorithm, it compares the on-going trajectory length to the best trajectory calculated so far; it stops when it

³ Rational Unified Process by Rational Software, <http://www.rational.com>

sees the trajectory is getting worse. That way, it avoids any useless calculations. Once the optimized sequence is found, the expert memorizes it and enters the trajectory in the virtual map.

Advising the robot on its trajectory



After it is done, the robot is good to go. During the course the robot puts its location on the map, as in the Blackboard system, in addition to any next location the robot could move. Then, the Waypoint Expert gives its advice on which location is on the way to the next waypoint. This seems to be simple; however, since the course has several obstacles and other considerations to take the Waypoint Expert has to work with the other Experts employed in the Navigation Challenge. As you have noticed, the Waypoint Expert

plays a double role of trajectory planner and advisor. It is a key object of the artificial intelligence used in the Navigation Challenge. A waypoint is considered reached when it is at less than one meter of the robot.

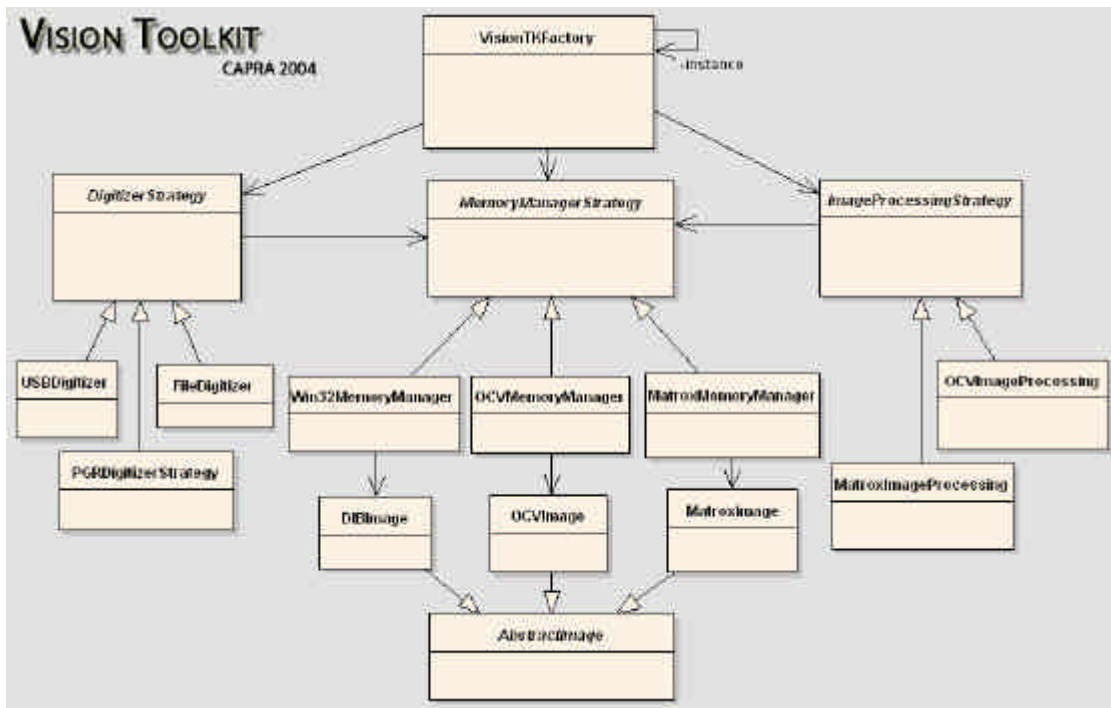
Archie

Archie is the main component of the software architecture. Inspired from the human brain, it mimics the communications between the different sections of the brain. As information is gathered by an expert or device (camera, waypoint expert, range finder, etc.) it is interpreted, classified and stored in a virtual map. This map is then used by the decision experts to analyze the environment and take action. Archie is responsible for routing and transferring information between the different devices and experts. In Archie, each device or expert is called

a “device” and is responsible for a single task. Once devices are declared, communication channels are created (called “connections”). A “device” is only responsible for transforming the information it receives. In other words, it has no information as to which other “device” is interested by its output. Archie is responsible for taking outputs from “devices” and routing it to other “devices” using the relevant “connections”. All the information about the “devices” and “connections” are stored in an XML configuration file to allow easy configuration and maintenance.

Video library abstraction layer

In order to increase portability and flexibility of vision algorithms, we created an abstraction layer that allows us to use different vision libraries. Numerous well known heuristic have been used, such as Factory [GoF95], Interface [Grand98] and Strategy [GoF95]. Strategy patterns provides us with the ability to use multiple variation of an algorithms like Matrox’s MIL and the OpenCV library. Most importantly, it allows us to add our own algorithms to fulfill IGVC needs.

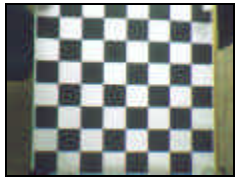


DigitizerStrategy	Abstraction of the material that is used to acquire video data
MemoryManagerStrategy	Abstraction of the memory management used.
ImageProcessingStrategy	Abstraction of vision algorithms implementation for video data analysis

Vision algorithms



Raw image



Corrected image

The first step in autonomous navigation is the ability to measure the elements surrounding the robot. To achieve such a goal, the deformation caused by the wide-angle lens must be reduced. To correct this problem, we use a feature of the OpenCV library, which calculates the parameters of the deformation and corrects the image. Using a chessboard, we point out the intersection points of the squares and match them to their “real” positions.

Once the image is corrected, we need to translate the coordinates into real world positions. Using the same chess board, we specify what is the real size of the squares in three locations. Using linear extrapolation, we are able to analyze the distance of objects using a x-y interpolation of pixel to meter conversion.

Line detection

Line detection is achieved by breaking the image in small sections. For each region of interest, we then use a threshold based on luminosity and the hue. The threshold value is evaluated dynamically by a histogram analysis. Finally, we use a method developed by Hough to calculate the equation of the line. We also apply various algorithms to detect errors and to connect lines together.

Obstacle detection

Obstacle detection is done using two distinct systems: the camera and the range finder. The LMS data is analyzed to detect circular objects. These elements are then validated using visual data. If both detection methods agree, the barrel is added to the virtual map. Also, we use a histogram segmentation technique to identify any objects with a centre of mass in its centre and a constant diameter.

Response time

The response time of the robot is an approximate average between the range finder reaction time and the vision reactions time. We calculated that in most circumstances, Mentis will react within 1000ms.

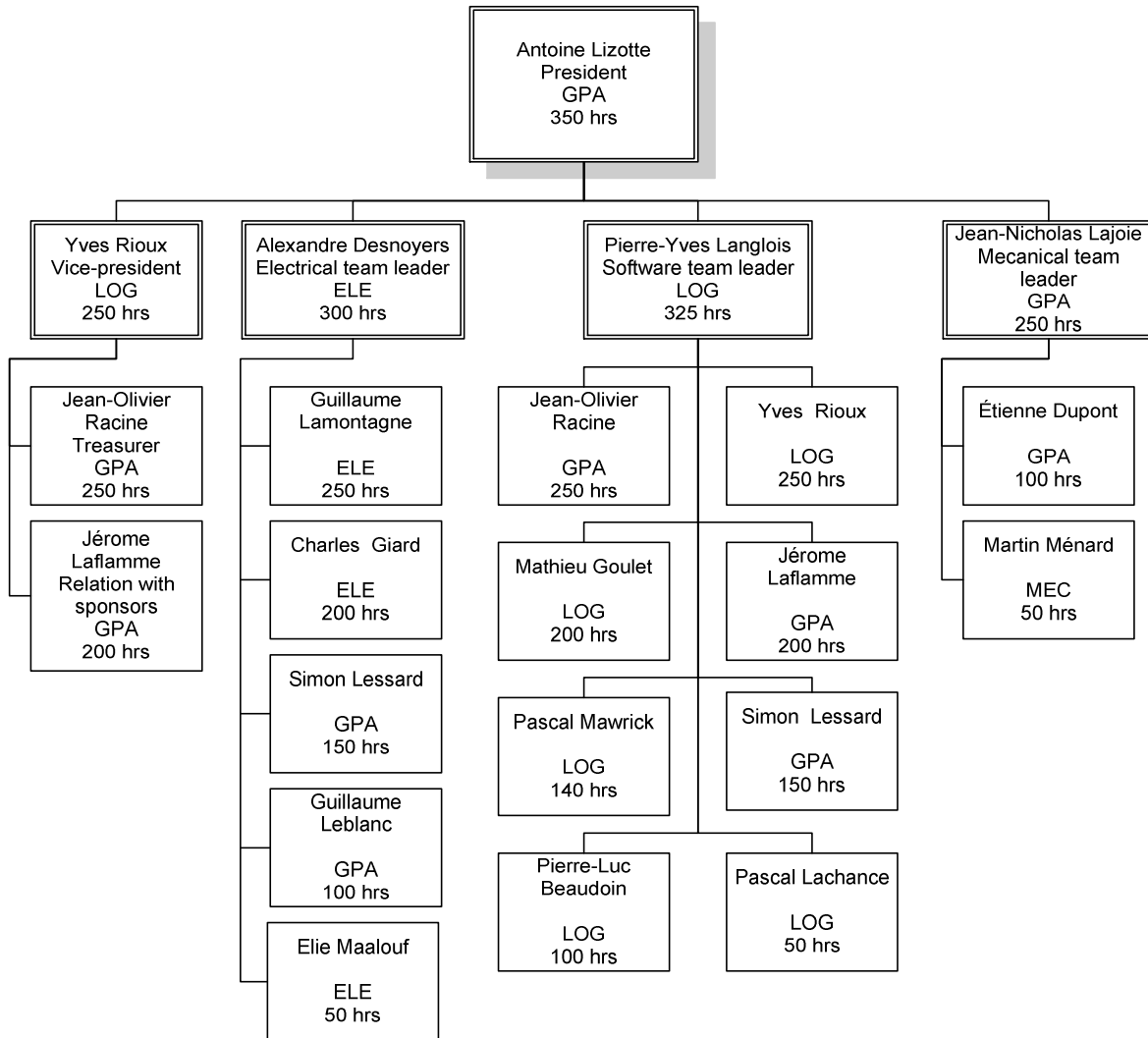


VEHICLE VALUE

Items	Value	Paid cost
Cab	360,00 \$	360,00 \$
Aluminum chassis	50,00 \$	50,00 \$
Carbon fiber floor	150,00 \$	- \$
Tripod	50,00 \$	50,00 \$
Motors and wheels (2)	150,00 \$	150,00 \$
Drive (2)	225,00 \$	225,00 \$
Encoders (2)	100,00 \$	100,00 \$
E-stop relay	25,00 \$	25,00 \$
Wireless e-stop	75,00 \$	75,00 \$
Aluminum electrical enclosure	50,00 \$	50,00 \$
Connectors and wires	150,00 \$	150,00 \$
Control embedded PC	250,00 \$	- \$
Acquisition card Sensoray	300,00 \$	- \$
Power supplies (2)	100,00 \$	50,00 \$
PCB and integrated circuit	600,00 \$	100,00 \$
Camera Point Grey Research	150,00 \$	- \$
Vision PC Matrox 4-sight	1 500,00 \$	- \$
GPS CMC Electronic	400,00 \$	- \$
SICK range finder	6 000,00 \$	5 000,00 \$
Envision flat screen	360,00 \$	360,00 \$
D-Link wireless LAN	220,00 \$	- \$
Wireless keyboard and mouse	75,00 \$	75,00 \$
Total:	11 340,00 \$	6 820,00 \$



ORGANIZATION CHART



Total person-hours: 3315 hours

ELE: Electrical Engineering

MEC: Mechanical Engineering

LOG: Software Engineering

GPA: Automated Manufacturing Engineering

Note: Some members are involved in more than one area of the club.