

UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECATRÔNICA

Note final
9,9 (nove e nove)



IAVL
IDENTIFICAÇÃO E ACOMPANHAMENTO VISUAL DE LINHA

Lucas Felipe dos Santos Maia
Marcelo Almeida Silva

São Paulo
2005

**UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECATRÔNICA**

**IAVL
IDENTIFICAÇÃO E ACOMPANHAMENTO VISUAL DE LINHA**

Trabalho de formatura apresentado à
Escola Politécnica da Universidade de
São Paulo para obtenção do título de
Graduação em Engenharia.

**Lucas Felipe dos Santos Maia
Marcelo Almeida Silva**

Orientador: Jun Okamoto Jr.

**Área de concentração:
Engenharia Mecatrônica**

**São Paulo
2005**

DEDALUS - Acervo - EPMN



31600011836

1494846

FICHA CATALOGRÁFICA

Maia, Lucas Felipe dos Santos
Identificação e acompanhamento visual de linha / L.F. dos S.
Maia, M.A. Silva. -- São Paulo, 2005.
p.

Trabalho de Formatura - Escola Politécnica da Universidade
de São Paulo. Departamento de Engenharia Mecatrônica e de
Sistemas Mecânicos.

1.Segurança de automóveis 2.Visão computacional 3.Programação orientada a objetos 4.Robôs (Controle) 5.Processamento de imagens I.Silva, Marcelo Almeida II.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos III.t.

AGRADECIMENTOS

Aos nossos pais, que mesmo distantes sempre nos apoiaram.

Ao orientador Prof. Dr. Jun Okamoto Júnior pelas diretrizes seguras e permanente incentivo.

A todos que, direta ou indiretamente, colaboraram na execução deste trabalho.

RESUMO

A segurança e a velocidade de uma viagem podem ser aumentadas com a utilização de veículos robotizados. Tais veículos despertam interesse por suas diversas aplicações e potencialidades. Eles diminuem riscos por reagirem a fatores externos em velocidades maiores que a velocidade de reação de um ser humano.

Um dos fatores externos a serem monitorados e controlados é a posição da estrada em relação ao veículo. Isso pode ser feito através do controle servo-visual.

A extração e análise de informações do ambiente através da visão é um dos problemas a serem tratados para que a navegação autônoma com controle servo-visual seja possível. Nesse trabalho utilizamos como informação visual uma faixa marcada no solo sendo que essa informação é adquirida através de um sistema de visão omnidirecional. A interpretação dessas informações é o problema principal a ser resolvido por nosso trabalho. Isso foi alcançado através de uma arquitetura orientada a objetos distribuídos, baseada em CORBA.

ABSTRACT

Travel's security and speed may be increased through utilization of autonomous vehicles. These vehicles lower risks by reacting faster to external factors.

The extraction and analysis of environment's information through vision are some of the problems that must be treated to allow autonomous navigation with visual servoing. Thus, we used line marked in the ground as visual information. This information's interpretation is this project main problem. It was solved through a distributed objects architecture, based in CORBA, and a mobile robot with an onboard omni-directional vision.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1	Introdução	1
2	Veículos autônomos em estradas	3
3	Descrição do ambiente de execução	10
3.1	Hipóteses e simplificações	10
3.2	Equipamentos e software do LPA.....	10
3.2.1	Equipamentos.....	11
3.2.2	Software	15
4	Método de resolução do problema	20
4.1	Algoritmo de resolução do problema.....	20
5	Projeto do software	25
6	Resultados	31
7	Conclusão.....	35
8	Anexo – Cronograma	36
9	Bibliografia	39

LISTA DE FIGURAS

Figura 1 - Robô MagellanPro e sistema de visão omnidirecional	12
Figura 2 - Sistema de visão omnidirecional.....	14
Figura 3 - Exemplo de imagem omnidirecional - com faixa a ser seguida.....	14
Figura 4 – Fluxo de execução esquemático	15
Figura 5 - CORBA - Uma requisição de um cliente para um objeto (CORBA.ORG, 2005)	16
Figura 6 - Exemplo de uma transação em CORBA (WOLFE, 1997).....	18
Figura 7 - Diagrama esquemático da hierarquia de objetos do Mobility.....	18
Figura 8 - Principais componentes do XVision	19
Figura 9 - Região onde faixa pode ser aceita	22
Figura 10 - Diagrama de blocos do controle do robô.....	23
Figura 11 - Erro de orientação do robô	23
Figura 12 – IAVL e bibliotecas auxiliares	25
Figura 13 - Fluxo de execução do software	27
Figura 14 - Diagrama de classes	28
Figura 15 - As funções das classes no projeto IAVL.....	29
Figura 16 - Visão ampla do circuito de testes	31
Figura 17 - Robô seguindo circuito.....	32
Figura 18 - Comparação entre a trajetória real e a trajetória autônoma.....	33
Figura 19 - Interface com o usuário	34

LISTA DE TABELAS

Tabela 1 - Classificação dos artigos conforme abordagem.....	9
Tabela 2 - Relação entre erro de direção e velocidade de translação.....	24

1 INTRODUÇÃO

A navegação autônoma de veículos terrestres propiciará viagens mais rápidas e seguras. O problema central a ser resolvido para que tais viagens sejam possíveis é a navegação.

Desse modo, essa questão básica pode ser dividida em duas categorias:

- Navegação em ambiente não-estruturado
- Navegação em ambiente estruturado

A navegação em ambiente não-estruturado consiste na navegação em ambientes que não apresentam marcações ou padrões, que podem ser utilizados como referência para localização e navegação do veículo. Esta classe de problema apresenta grandes dificuldades que aumentam a complexidade do processamento computacional necessário para identificar o caminho a ser seguido.

Já a navegação em ambiente estruturado, ou seja, em ambientes que apresentam padrões e marcações, é uma categoria de problema mais simples a ser resolvido pois existe um conhecimento *a priori* no ambiente que pode ser usado na criação de hipóteses que diminuem a complexidade do problema .

A solução do problema de navegação em ambiente estruturado consiste na execução de algumas tarefas:

- reconhecimento dos padrões presentes no ambiente a ser navegado,
- tomada de decisão sobre o caminho a percorrer,
- controle do veículo de acordo com a decisão tomada.

Visando a solução do problema apresentado, o projeto IAVL tem o objetivo de fazer um robô navegar num ambiente interno, usando como referência uma faixa colada ao

chão, utilizando visão omnidirecional. Esta faixa simula uma situação análoga as pistas de uma estrada de rodagem. A solução destes dois problemas é semelhante.

Nos itens a seguir, começamos com uma descrição da literatura a respeito do problema de acompanhamento de linha em tempo real, prosseguimos com uma descrição do ambiente de execução do projeto, onde são descritas as ferramentas e equipamentos utilizados, a seguir fazemos uma descrição do algoritmo utilizado para resolução do mesmo, logo em seguida detalhamos o projeto do software utilizado para resolver o problema e apresentamos os resultados obtidos em testes executados num ambiente real.

2 VEÍCULOS AUTÔNOMOS EM ESTRADAS

O problema de navegação autônoma em estradas é tratado por diversos pesquisadores com abordagens distintas.

Apostoloff e Zellinsky (2004) implementaram um sistema para navegação autônoma através da fusão de vários métodos. Eles utilizaram vários métodos para chegar a uma estimativa da posição da estrada. Tais métodos são executados em primeiro e segundo plano. Sendo que os que são executados em primeiro plano têm prioridade no processamento. Assim, é feita a verificação das probabilidades de acerto de cada método a uma freqüência menor que a freqüência de aquisição das imagens e é feita uma verificação para saber quais são os métodos que melhor estão identificando a pista e assim os mesmos são classificados para serem executados em primeiro ou em segundo plano de acordo com a probabilidade de acerto de cada um.

Bertozzi e Broggi (1998) implementaram um sistema para navegação em estradas planas com um método onde o efeito da perspectiva é removido e a estrada/faixa é identificada através da comparação da imagem com padrões, dividindo o processo paralelamente entre alguns computadores. Além disso, com a utilização de duas câmeras posicionadas uma ao lado da outra, foram utilizadas informações da imagem estéreo obtida pela combinação da aquisição das duas câmeras e dessas informações executaram a detecção dos obstáculos presentes na estrada.

Nos artigos “Accurate road following and reconstruction by computer vision” (CHAPUIS; AUFRERE; CHAUSSÉ, 2002), “Real-Time Lane detection for autonomous vehicle” (JEONG; et al., 2001) e “Development of a vision-based lane detection system considering configuration aspects” (HUH; et al., 2005) os autores executam a detecção de linha através do gradiente da imagem e as faixas podem ser

detectadas através de um “treinamento” anterior do programa, como em (CHAPUIS; AUFRERE; CHAUSSÉ, 2002), ou através de um Filtro de Kallmann com em (JEONG; et al., 2001) e (HUH; et al. , 2005)

Foda e Dawoud (2001) utilizaram redes neurais para detectar as faixas das estradas. Nessa abordagem, primeiro a imagem é classificada entre áreas que pertencem à estrada e áreas que não pertencem à estrada. O passo seguinte é procurar pelas linhas no chão e o terceiro passo é separar as faixas que pertencem à faixa de rodagem onde está o carro das outras faixas da pista. E para obter esses resultados, redes neurais e lógica fuzzy são utilizadas para que se consiga atingir o objetivo.

Em “Advanced Lane Recognition – Fusing vision and radar” (GERN; FRANKE; LEVI, 2000) os autores utilizaram um filtro de Kallmann para fundir informações de um radar e do algoritmo de detecção da faixa de rodagem do carro. Assim, a fusão de informações serve para melhorar a estimativa da curvatura da estrada utilizando a informação da posição dos carros à frente do veículo onde o aparato está montado.

Já Kreucher e Lakshmanan (1999) usaram informações do espaço de freqüência da imagem e através do conhecimento que faixas das estradas apresentam componentes de energia dominantes em dada direção, um vetor no domínio da freqüência é obtido para blocos de 8x8 da imagem e tal vetor é verificado contra um modelo da faixa a ser detectada. Feito isso, é determinada uma probabilidade através da semelhança entre os modelos. O próximo passo é encontrar o máximo global de uma função de densidade de probabilidade e nesse máximo global a faixa estará presente.

Porém após esse estudo mais aprofundado três publicações foram selecionadas para serem base de definição da abordagem de resolução do problema e serão detalhadas em maiores detalhes a seguir.

Em “A Fast and Robust Vision Based Road Following Algorithm” (AUFRÉRE; et al., 2000) é utilizado um método de detecção baseado em um reconhecimento recursivo das laterais da pista usando um modelo probabilístico. O reconhecimento é feito em zonas de interesse ótimas que são deduzidas do modelo.

O modelo estatístico é composto de 20 parâmetros de imagens e 5 parâmetros que caracterizam a localização do veículo, a câmera e a geometria da pista. Desta forma o modelo contém um vetor X e sua matriz de covariância CX , onde:

$X = (U_{1r}, \dots, U_{nr}, U_{1l}, \dots, U_{nl}, X_0, W, A, C_i, L)$, com U_{ir} , U_{nr} , U_{1l} , U_{nl} , representando as coordenadas das laterais da pista para as n diferentes faixas da imagem, X_0 a posição do veículo na pista, W o ângulo de curvatura da trajetória do veículo, e A o ângulo de inclinação da câmera, C_i a curvatura da pista e L a largura da pista. A matriz CX define a variação de todos os parâmetros.

Depois define-se as zonas de interesse extraindo-se as menores variâncias encontradas na diagonal de CX , logo após ocorre uma fase de detecção da faixa nas zonas de interesse selecionadas. Isso é feito localizando-se o ponto de máximo gradiente de cada região das zonas de interesse, e traçando-se uma curva aproximando todos os pontos encontrados utilizando o método dos mínimos quadrados.

Outro texto consultado foi “Lane Detection Using Histogram-Based Segmentation and Decision Trees” (GONZALEZ; OZGUNER, 2000). Neste estudo é utilizado um método baseado em segmentação por histogramas. Segmentação é uma técnica de processamento digital de imagens no qual faz-se a subdivisão da imagem a partir de determinadas características da mesma, no caso deste artigo essas características são provenientes da análise dos histogramas de intensidade que são extraídos da imagem.

Assume-se que nas linhas mais baixas da imagem só existe estrada, desta forma é feito um histograma de uma pequena porção da pista com 5 pixels de altura, depois calcula-se a média , o máximo e o mínimo da distribuição de nível de cinza do histograma. Valores acima do máximo dessa distribuição são considerados como “*lane markers*”, e abaixo dos mínimos como sendo objetos (baseado no fato de que outros veículos produzem sombras mais escuras que a pista). Essa divisão da imagem é feita até se atingir o meio da imagem ou o ponto de fuga, pois esse é o ponto onde as informações visuais necessárias para identificação da estrada se tornam imprecisas.

Depois de concluída essa etapa faz-se a limiarização da imagem considerando-se os valores máximos de intensidade encontrados.

A seguir é feita a extração de algumas características dos objetos candidatos a estradas, ou “*lane markers*”. Através do método dos mínimos quadrados, obtém-se a inclinação, a intersecção e o coeficiente de correlação. Calcula-se o centróide do objeto, o ângulo médio entre cada ponto do objeto e o ponto de fuga, o tamanho do objeto, o comprimento máximo do objeto e o ponto onde tal máximo está localizado. A partir de então, são atribuídas notas entre -9 e 5 a cada objeto através de uma árvore de decisão utilizando os parâmetros levantados acima. Objetos que obtém nota maior que 4 são muito provavelmente parte da estrada, os que obtém nota entre 4 e 0 são classificados como estrada apenas se sua posição for compatível com os objetos que já foram admitidos como parte integrante da estrada. Objetos com pontuação negativa são descartados da análise.

O último texto é “Statistical Characterization of the Visual Characteristics of Painted Lane Markings” (KLUGE; JOHNSON, 1995) ,no qual se utiliza segmentação por

Histogramas. Porém a segmentação é feita limiarizando-se a imagem a partir do valor da intensidade do vale entre dois picos do histograma.

Na imagem analisada, existe apenas uma faixa e deseja-se encontrar as retas que determinam suas bordas.

Assim, encontra-se a faixa e suas bordas são aproximadas por retas, sendo que dois métodos são utilizados para tanto. No primeiro assume-se a equação $coluna=m*linha+b$ e considera-se que as bordas da faixa seguem esse padrão, adota-se que no ponto de fuga da imagem “linha” é igual a 0 e encontra-se o valor de b, percorre-se as linhas paralelas da imagem, e linhas que possuem valores iguais de b são eleitas como parte das bordas da pista. A outra forma é utilizar um algoritmo de mínima média de quadrados, determinando a inclinação das bordas da pista e qual seu ponto de fuga, esse método mostrou-se bastante robusto pois seu resultado só é comprometido caso 50% ou mais dos pontos utilizados na estimativa forem espúrios. Tais referências podem ser classificadas de duas formas: uma leva em conta a fonte das informações e a outra leva em conta o método de detecção da faixa ou estrada.

Quanto à fonte de informação, duas são as abordagens:

- Fusão de sensores

São utilizadas informações provenientes de mais de um sensor para determinar a posição da estrada. Essas informações, depois de combinadas são analisadas para determinação da posição da estrada. Com essa abordagem, devido à utilização de mais de um sensor ocorre uma diminuição da incerteza das medições individuais.

- Câmera de vídeo

Nessa abordagem, informações como intensidade dos pixels são utilizadas para determinar a posição da estrada. Combinações de câmeras de vídeos, para formação de visão estéreo também são utilizadas de modo a possibilitar a medição de distâncias.

Já quanto ao método de detecção, as abordagens são as seguintes:

- Estatístico

Com base em dados obtidos da imagem, são executadas análises estatísticas para determinar qual ação será tomada e qual a localização da estrada

- Gradiente

Com base em informações obtidas através da binarização da imagem utilizando seu gradiente, é determinada a posição da estrada e também qual deve ser o comando para o veículo.

Quando utilizando fusão de sensores, os autores executam a detecção da estrada ou da faixa através de métodos estatísticos. Assim fizeram Apostoloff e Zellinsky (2004), Gern, Franke e Levi (2000).

Já quando a detecção utiliza apenas câmeras de vídeo, os dois métodos de detecção são utilizados. Foda e Dawoud (2001), Aufrére; et al (2000), Gonzalez, Ozguner, (2000) e Kluge, Johnson (1995) utilizam métodos estatísticos para determinar a posição da estrada. Já Bertozzi e Broggi (1998), além dos métodos estatísticos, planificam a imagem.

Chapuis, Aufrére e Chausse (2002), Jeong, et al (2001) e Huh, et al. (2005) utilizam informações do gradiente para detectar a posição da faixa.

Tabela 1 - Classificação dos artigos conforme abordagem

	<i>Fusão de Sensores</i>	<i>Câmera(s) de vídeo</i>
Estatístico	Apostoloff e Zellinsky (2004) Gern, Franke e Levi (2000)	Foda e Dawoud (2001) Aufrére, et al (2000) Gonzalez, Ozguner, (2000) Kluge, Johnson (1995) Bertozzi e Broggi (1998)
Gradiente		Chapuis, Aufrere e Chausse (2002) Jeong, et al (2001) Huh, et al. (2005)

3 DESCRIÇÃO DO AMBIENTE DE EXECUÇÃO

Todas as fases do projeto serão desenvolvidas no Laboratório de Percepção Avançada (LPA). Alguns dos equipamentos e softwares a serem utilizados no projeto IAVL necessitaram uma instalação e aprendizado prévios. Além disso, esse ambiente aceita hipóteses simplificadoras descritas no item 3.1 e sua infra-estrutura servirá de base para o desenvolvimento do projeto, conforme descrições do item 3.2.

3.1 Hipóteses e simplificações

Tendo em vista o problema proposto, inicialmente focou-se em encontrar uma forma eficiente de caracterizarmos a faixa no algoritmo que estamos desenvolvendo, levando em conta as particularidades do ambiente onde o programa será executado, o LPA. Esta caracterização deve ser consistente e suficiente para garantir uma movimentação contínua do robô, além de nos proporcionar, futuramente, um modo pouco trabalhoso de introduzirmos no algoritmo comandos que permitam que o robô faça a identificação inicial da faixa a ser seguida de forma completamente autônoma. Uma característica do ambiente de testes que minimiza a dificuldade do projeto é o fato de trabalhar-se em um local no qual temos garantia de que não haverão alterações de luminosidade e também não ocorrerão oclusões da pista.

Além disso, serão adotadas outras duas simplificações, a primeira é que inicialmente o robô sempre será posicionado sobre a faixa a ser seguida. Será assumido ainda que o robô percorrerá sempre trechos planos.

3.2 Equipamentos e software do LPA

Todo o projeto foi desenvolvido com a utilização de equipamentos e software existente no LPA.

Merecem destaque o robô MagellanPRO, o sistema de visão omnidirecional, CORBA, Mobility, XVision que serão descritos em maiores detalhes abaixo. Além deles, ainda existem um computador Pentium 4 HT que será utilizado para desenvolvimento e execução do software, uma rede wi-fi que servirá como meio de comunicação entre o robô e o computador e uma rede ethernet cabeada que conecta todos os computadores do laboratório.

À seguir são apresentados os itens relevantes de hardware e software que serão utilizados no projeto.

3.2.1 Equipamentos

Entre os componentes de hardware relevantes, temos o MagellanPRO, uma plataforma móvel desenvolvida pela empresa iROBOT e que possui vários sensores acoplados, um computador embarcado, interfaces de comunicação e uma plataforma na qual podem ser adicionados componentes personalizados, outro componente relevante é o sistema de visão omnidirecional que consiste em um conjunto formado por um espelho e uma câmera que fornece uma visão em 360° do ambiente observado.

3.2.1.1 MagellanPro

O MagellanPRO é um robô construído pela empresa iROBOT. Tal robô possui 16 polegadas de diâmetro por 10 polegadas de altura. Nesse volume estão montados 16 sonares, 16 infravermelhos e 16 sensores táteis.

Embarcado se encontra um sistema EBX (Embedded Board Exchange) Pentium que oferece todas as funcionalidades de um computador normal e foi adicionado um link wi-fi que conecta o robô à w-lan (wireless lan) do LPA (Laboratório de Percepção Avançada). Além disso, um sistema de visão omnidirecional, com espelho parabólico

e câmera de vídeo com transmissor de sinal foi acoplado à superfície superior do veículo.

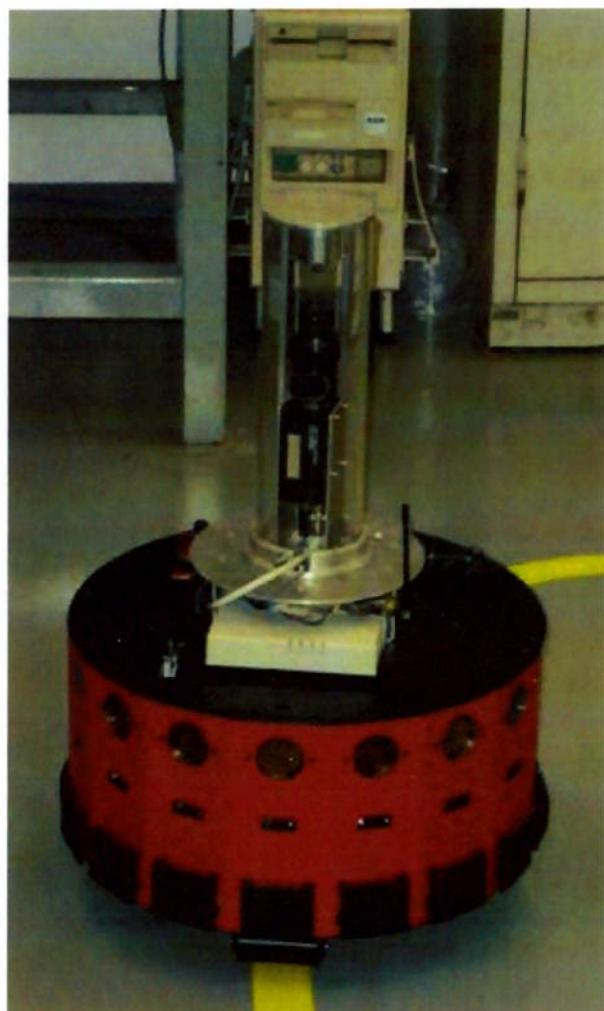


Figura 1 - Robô MagellanPro e sistema de visão omnidirecional

O MagellanPro se move através de duas rodas controladas por motores, e possui dois elementos passivos: um rodízio e um rolo. Sua velocidade máxima de translação é de 1m/s e de rotação é de $2\pi/3$ rad/s.

A interface é feita através de um display com informações básicas sobre o MagellanPro.

Assim, através de suas 2 baterias embarcadas, o robô possui autonomia de cerca de 8 horas.

3.2.1.2 Sistema de visão omnidirecional

Segundo (Grassi; et al, 2001), um sistema de visão omnidirecional pode ser definido da seguinte forma:

“Sistemas de visão omnidirecional produzem imagens de 360° do ambiente podendo ser utilizados em navegação, teleoperação e controle servo visual de robôs. Este tipo de sistema dispensa o movimento da câmera para determinada direção de atenção mas requer processamento não convencional da imagem, uma vez que a imagem adquirida se encontra mapeada em coordenadas polares não lineares. Uma maneira efetiva de se obter uma imagem em um sistema omnidirecional é com o uso combinado de lentes e espelhos. Várias formas de espelhos convexos podem ser utilizadas montando-se uma câmera com o seu eixo óptico alinhado com o centro do espelho.”

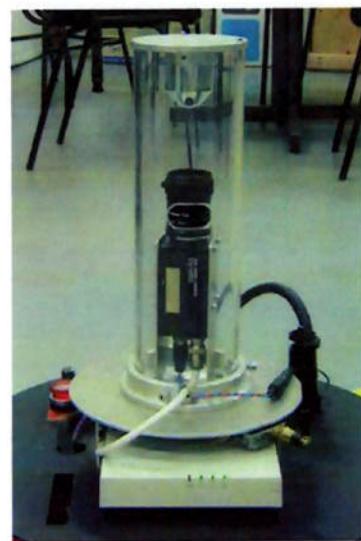


Figura 2 - Sistema de visão omnidirecional

A Figura 3 é um exemplo de imagem omnidirecional obtida no laboratório na qual aparecem o robô MagellanPRO (centro da imagem) a faixa a ser seguida (região mais clara no centro da imagem) e outros objetos menos interessantes.

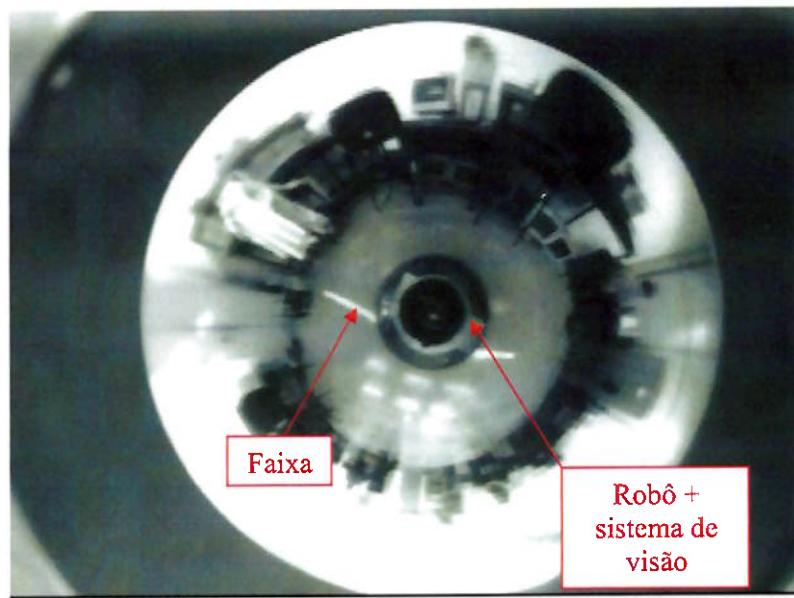


Figura 3 - Exemplo de imagem omnidirecional - com faixa a ser seguida

3.2.2 Software

Entre os programas e especificações a serem utilizados no projeto IAVL, temos o CORBA, o Mobility e o XVision. Sendo que CORBA é uma especificação que aplicações utilizam para trabalhar em rede, o Mobility é a ferramenta desenvolvida pela empresa iRobot, baseada em CORBA, para controle do robô MagellanPRO, já o XVision é um pacote código aberto para aquisição de imagens digitais.

3.2.2.1 Fluxo de execução para o IAVL

A execução do software se inicia com a aquisição dos quadros provenientes da câmera de vídeo embarcada no MagellanPRO. Após essa aquisição, o sistema do IAVL faz a interpretação desses quadros e decide a direção e velocidade do próximo movimento do robô

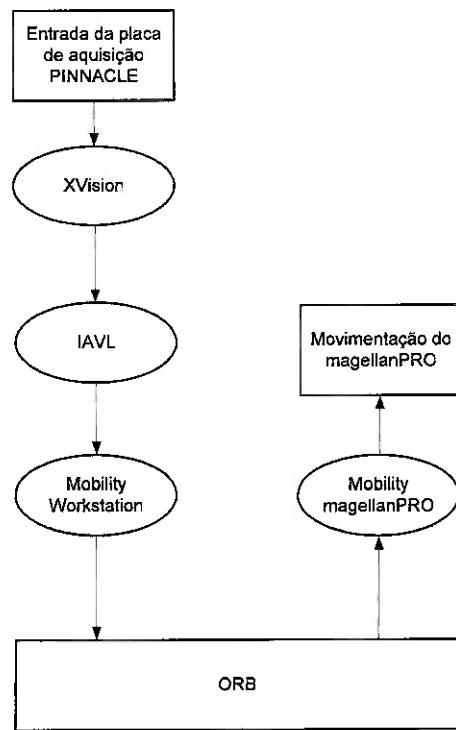


Figura 4 – Fluxo de execução esquemático

No próximo passo o IAVL faz uma requisição através de uma IDL para o objeto que controla o motor do robô, passando parâmetros de velocidade linear e angular. O ORB se encarrega de enviar o comando de movimento ao robô sendo que este, ao receber o comando, executa-o. Essa seqüência se repete até que o MagellanPRO receba um comando de parada, seja via software ou através do botão de emergência do mesmo.

Na Figura 4 é mostrado um esquema do fluxo de execução para o IAVL.

3.2.2.2 CORBA

CORBA (Common Object Request Broker Architecture) é uma arquitetura e infraestrutura aberta que aplicações utilizam para trabalhar em rede. Utilizando um protocolo padrão, um programa baseado em CORBA de qualquer fornecedor, em praticamente qualquer computador, sistema operacional, linguagem de programação e rede, pode se integrar a um programa baseado em CORBA do mesmo ou de outro fornecedor, em praticamente qualquer computador, sistema operacional, linguagem de programação e rede.

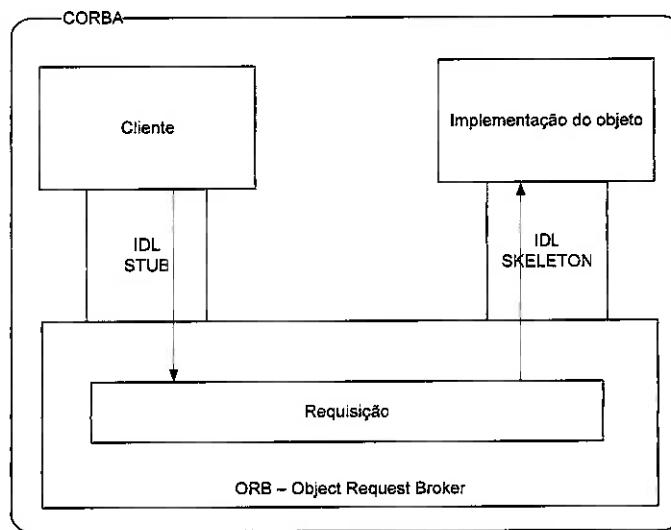


Figura 5 - CORBA - Uma requisição de um cliente para um objeto (CORBA.ORG, 2005)

As aplicações CORBA são compostas por objetos, unidades individuais de software que combinam funcionalidade e dados, e que freqüentemente (mas não sempre) representam algo no mundo real. Usualmente, há muitas instâncias de um objeto de um único tipo - por exemplo, um website de e-commerce teria muitas instâncias de carrinhos de compra, todas idênticas na funcionalidade mas cada uma estaria atribuída a um cliente diferente, e conteria os dados que representariam a mercadoria que o cliente selecionou. Para outros tipos, pode haver somente uma instância.

Para cada tipo de objeto, tal como o carrinho de shopping mencionado, define-se uma IDL. A IDL é a sintaxe do contrato que o objeto do usuário oferece aos clientes que a invocam. Qualquer cliente que invocar uma operação no objeto deve usar esta relação IDL especificando a operação que quer executar e seus argumentos. Quando a requisição alcança o objeto-alvo, a mesma IDL será usada para interpretar os argumentos de modo que o objeto possa executar a operação pedida a ele. A IDL será usada então para devolver os resultados para que o cliente consiga interpretá-los. Esta separação entre interface e implementação, permitida pela IDL, é a essência do CORBA. A relação entre cada objeto é muito bem definida. Porém, a execução de um objeto - seu código e seus dados - é escondida do sistema (isto é, encapsulada) atrás de um limite que o cliente não pode cruzar. Os clientes acessam os objetos somente através de sua IDL, invocando somente as operações que aquele objeto expõe através da mesma, somente com os parâmetros (entrada e saída) que são incluídos na chamada da IDL supracitada.

A Figura 6 mostra um exemplo de requisição entre um cliente e um servidor através do CORBA, nela também estão representados os serviços básicos existentes entre o ORB e a camada cliente-servidor.

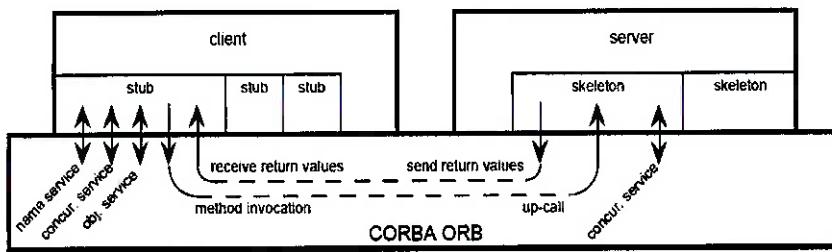


Figura 6 - Exemplo de uma transação em CORBA (WOLFE, 1997)

3.2.2.3 Mobility

Para o acionamento do MagellanPRO é utilizado um pacote de software chamado Mobility, também fornecido pela empresa iROBOT. Nesse pacote é definido o Mobility Robot Object Model (MROM) utilizando CORBA. O MROM define o sistema do robô como um conjunto de objetos hierarquizados e distribuídos. Cada objeto é uma unidade separada de software com uma identidade, uma interface e um estado. Os objetos fornecem um modelo flexível do sistema do robô, que pode ser modificado assim que novos hardwares, novas funcionalidades, e novas aplicações são adicionadas.

Na Figura 7 é apresentada a hierarquia simplificada do MROM.

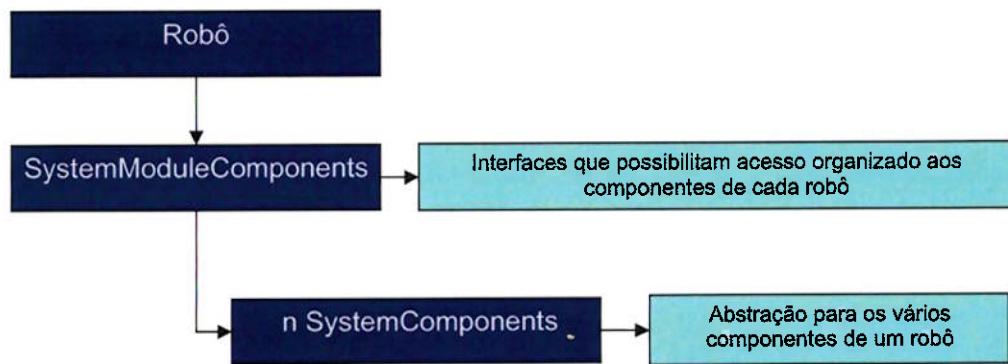


Figura 7 - Diagrama esquemático da hierarquia de objetos do Mobility

3.2.2.4 XVision2

Pacote de software desenvolvido pelo departamento de ciência da computação da universidade JOHNS HOPKINS, e que é uma versão totalmente nova do XVISION original, desenvolvido na Universidade de YALE.

Tal pacote fornece uma interface simples com a placa de aquisição de dados PINNACLE e através desse pacote é possível obter a imagem e armazená-la em um na memória.

Os principais componentes do Xvision são os devices, os consoles e as images. Os devices são os drivers para as placas de vídeo que podem ser utilizadas em conjunto com a biblioteca, as images contém estruturas de dados que representam uma imagem adquirida pelo sistema, e os consoles contém classes para exibição desta imagem através de uma interface LINUX. Na Figura 8 é mostrada a relação entre esses componentes.

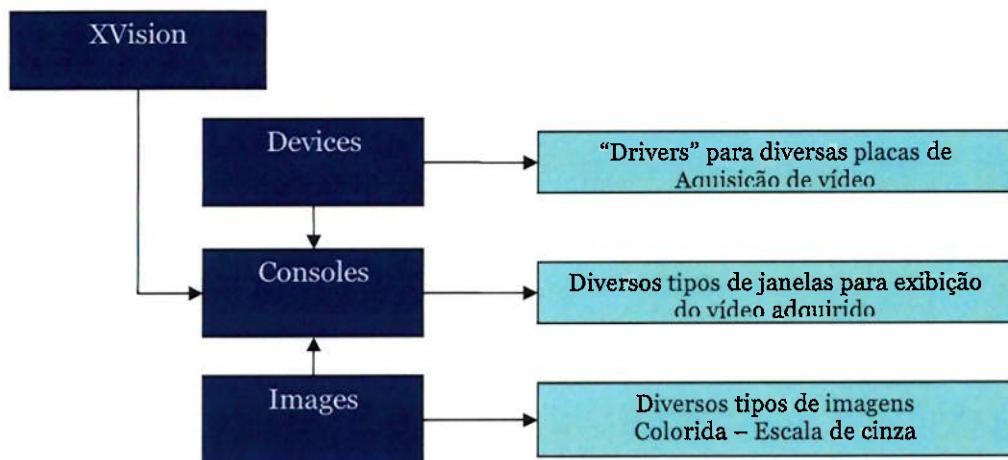


Figura 8 - Principais componentes do XVision

4 MÉTODO DE RESOLUÇÃO DO PROBLEMA

Após o estudo das referências bibliográficas foi tomada a decisão de resolver-se o problema com uma abordagem semelhante à utilizada no artigo “Lane Detection Using Histogram-Based Segmentation and Decision Trees” (GONZALEZ; OZGUNER, 2000), ou seja, o problema foi resolvido utilizando segmentação por histogramas, escolha dos candidatos a faixa e estimativa da posição da estrada utilizando aproximação pelo método dos mínimos quadrados.

4.1 Algoritmo de resolução do problema

Tendo em vista o método escolhido, algumas características particulares do ambiente e do tipo de visão que será utilizado, serão aplicadas algumas modificações no método utilizado por (GONZALEZ; OZGUNER, 2000).

Desta forma, o método utilizado para identificação da faixa será descrito a seguir:

1 – Limitar zona de interesse

Como grande parte da imagem omnidirecional não faz parte da região de interesse, no caso o solo, essa região desinteressante é descartada.

2 – Definir uma região em torno do robô

Através de subjanelas, usualmente com tamanho 3x3, define-se uma região de pixels em torno do robô onde se encontrará a faixa com certeza.

3 – Encontrar o pico de intensidade na região

Utilizando histograma, é determinada a intensidade média máxima em cada subjanela de tal região e o desvio padrão da subjanela que possui a intensidade média máxima.

4 – Identificar candidatos a faixa

Teste de hipótese:

A princípio todas as janelas da imagem são consideradas como pertencentes a faixa, ou seja, a hipótese nula é que a média da janela atual é igual a média da faixa padrão.

A validação desta hipótese é feita através do uso da distribuição t-Student com um nível de significância de 5%. O nível de significância de um teste representa a chance de rejeitarmos a hipótese nula quando ela é verdadeira (erro Tipo I), desta forma ao diminuirmos o nível de significância do teste, estaremos aumentando a possibilidade de aceitação da hipótese nula quando ela é falsa (erro Tipo II). Assim a o nível de significância escolhido também indica o nível máximo de erro tipo II que é possível se cometer sem comprometer o resultado.

5 – Obter a posição dos candidatos e decidir se são faixa

Para que um candidato seja considerado faixa, a distância entre o mesmo e a última faixa não deve exceder uma distância máxima. Na Figura 9 é apresentada a região de aceitação dos candidatos.

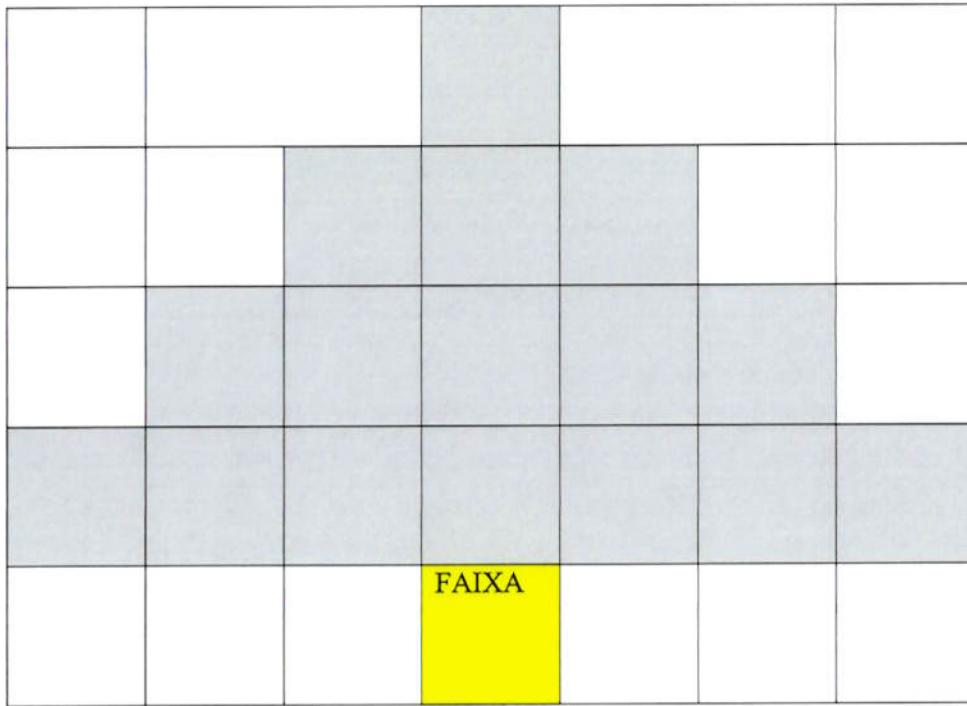


Figura 9 - Região onde faixa pode ser aceita

7 – Estimar parâmetros da reta que aproxima a faixa através do método dos mínimos quadrados.

Foi utilizado o método dos mínimos quadrados para estimar a posição da reta. A Equação 1 apresenta o equacionamento do método.

$$\text{Equação 1 - Mínimos quadrados } y=a + b \cdot x$$

$$\begin{bmatrix} <1,1> & <1,x> \\ <1,x> & <x,x> \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} <1,y> \\ <x,y> \end{bmatrix}$$

8 – Comandar o robô para se movimentar na direção da faixa.

Sabendo qual a direção e sentido para os quais aponta a frente do robô e sabendo qual a direção e sentido para onde a faixa aponta, o algoritmo utiliza uma variável de erro como constante de proporcionalidade para a velocidade de rotação do robô.

A Figura 10 representa o controle descrito acima.

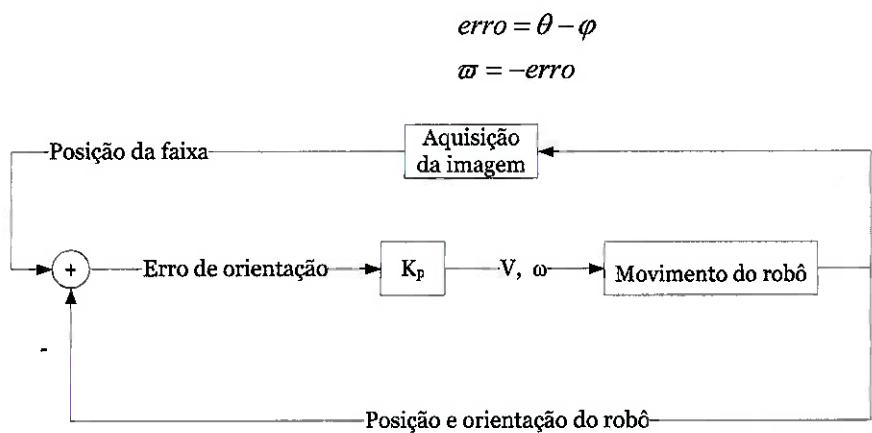


Figura 10 - Diagrama de blocos do controle do robô

Onde,

θ = ângulo formado entre a faixa e o eixo x negativo do sistema de coordenadas centrado no robô

φ = ângulo formado entre a frente do robô e o eixo x negativo do sistema de coordenadas centrado no robô

ω = velocidade de rotação do robô

É importante dizer que existem valores de corte tanto para a velocidade de translação, quanto para a de rotação.

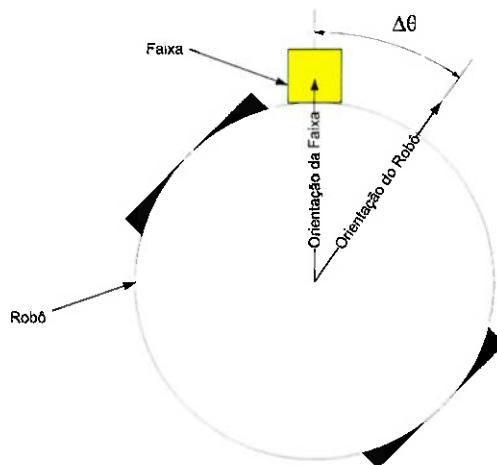


Figura 11 - Erro de orientação do robô

A velocidade de translação é controlada de acordo com a Tabela 2.

Tabela 2 - Relação entre erro de direção e velocidade de translação

<i>Condição</i>	<i>Velocidade de translação</i> <i>[m/s]</i>
$ \text{erro} > 1$	0.3
$0.75 < \text{erro} < 1$	0.4
$0.5 < \text{erro} < 0.75$	0.5
$ \text{erro} < 0.5$	0.6

5 PROJETO DO SOFTWARE

Tendo em vista que a execução do software do projeto IAVL é em tempo real e está sendo implementado em C++, uma linguagem orientada a objetos, foram definidos o fluxo do software, descrito de forma gráfica na Figura 13, e o de classes, mostrado na Figura 14.

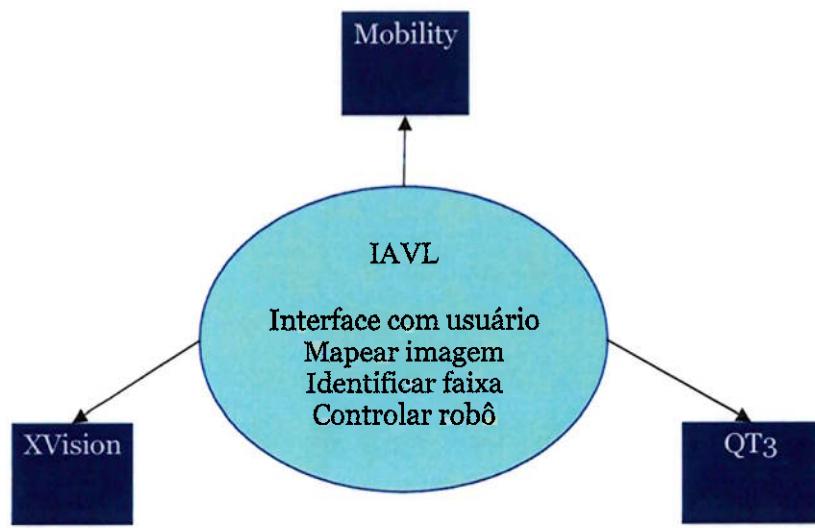


Figura 12 – IAVL e bibliotecas auxiliares

O fluxo de software se divide em três threads, conforme descrito na Figura 13:

threadVisao

Responsável pelo processamento da imagem adquirida com o sistema de visão omnidirecional

threadRobo

Responsável pelo acionamento do robô

threadVelo

Responsável pela atualização do velocímetro do robô

Assim, o programa inicia sua execução com a aquisição e tratamento da imagem e a seguir é feita a sua análise, na qual é feita a extração de parâmetros que definirão se uma área da imagem é faixa, seguida da identificação e parametrização da faixa, que consiste em localizar e definir os parâmetros da reta aproximadora dos pontos tidos como faixa. À partir das informações obtidas no módulo de reconhecimento da faixa, passa-se o comando de movimento ao robô e reinicia-se o ciclo de reconhecimento de faixa. Desse ponto em diante essa rotina se repete indefinidamente até que seja dado o comando de parada via software ou através do botão de emergência do robô, com a atualização da velocidade de cruzeiro do robô sendo feita em paralelo.

O diagrama de classes definido na Figura 14 representa as classes utilizadas no projeto.

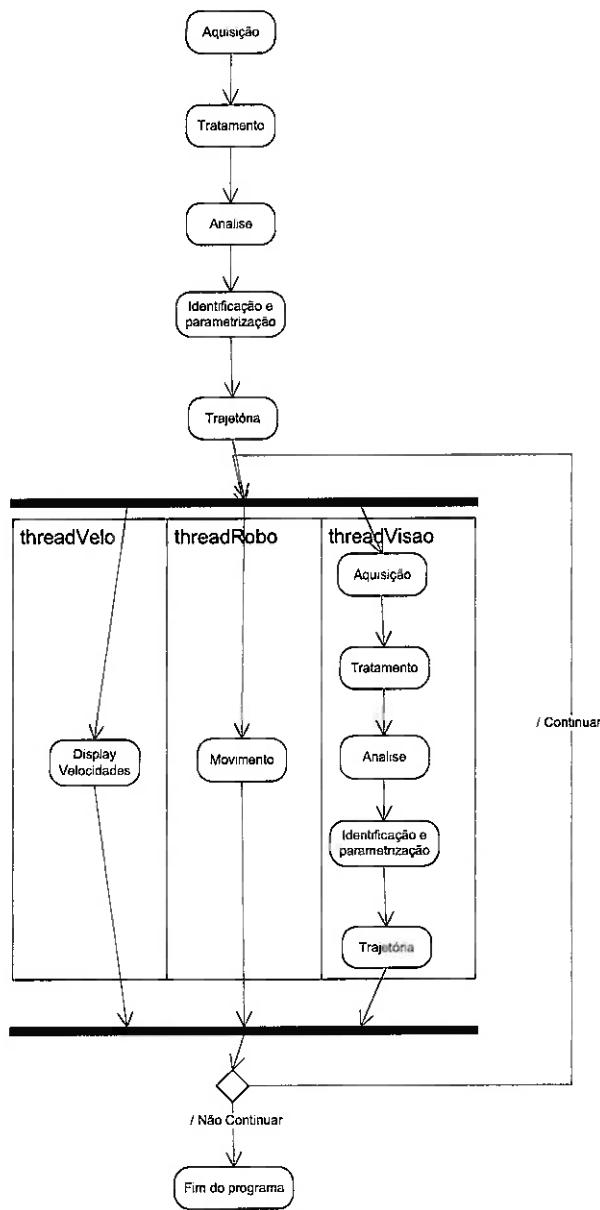


Figura 13 - Fluxo de execução do software

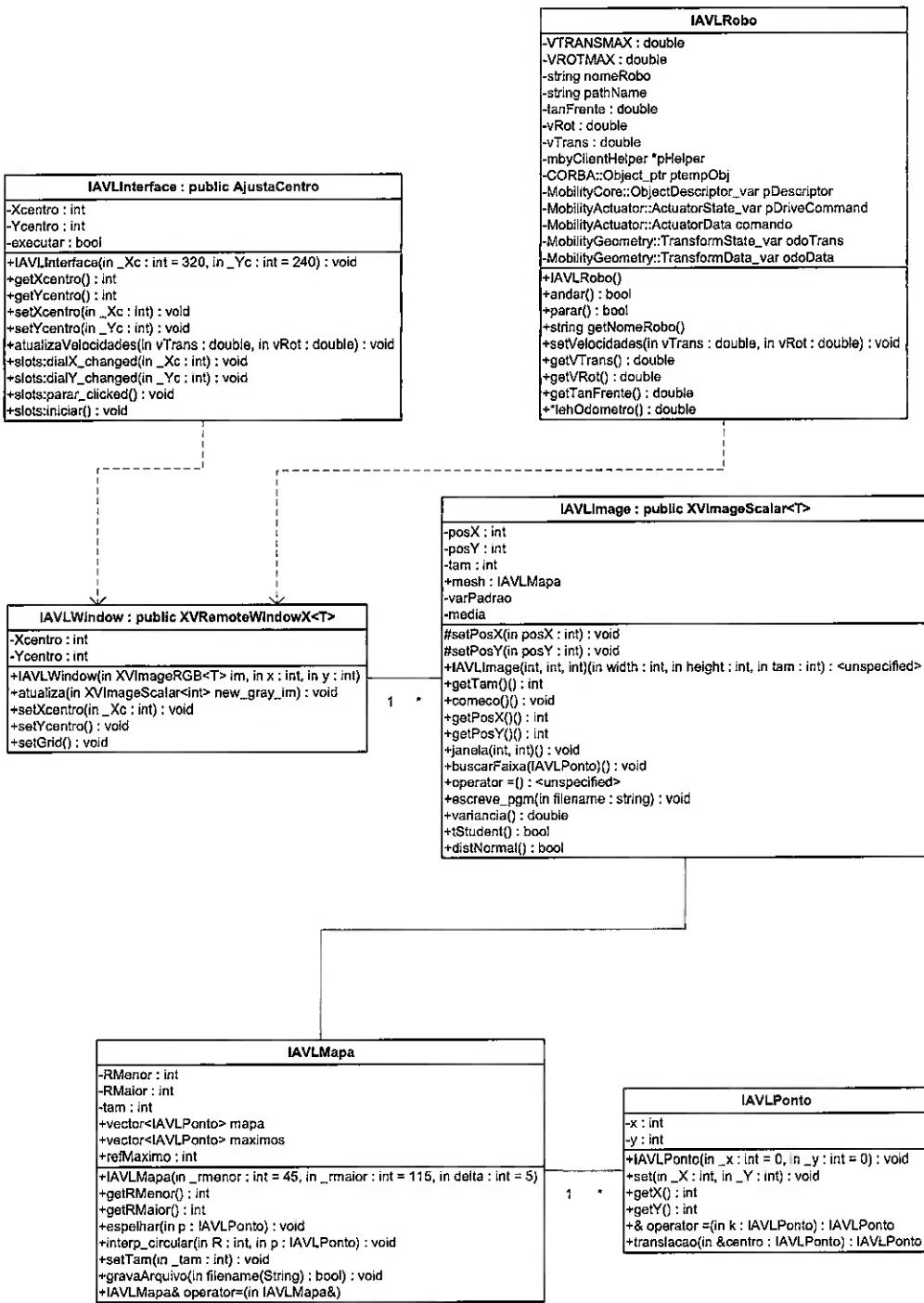


Figura 14 - Diagrama de classes

As classes foram projetadas de modo a representar os objetos componentes do sistema utilizado para resolver o problema proposto.

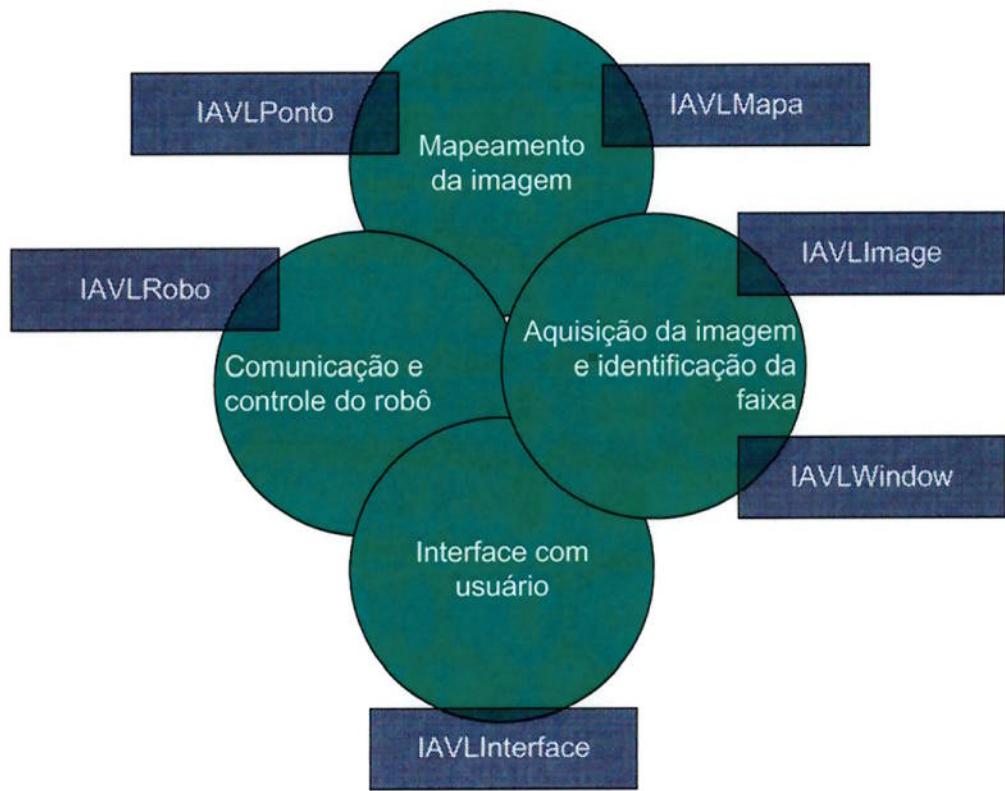


Figura 15 - As funções das classes no projeto IAVL

Assim, IAVLMapa é uma classe utilizada para realizar o mapeamento da região de interesse na imagem e também para guardar informações sobre as possíveis posições da faixa na mesma. Ela utiliza a classe ponto para representar um par ordenado (x,y) que represente um pixel na imagem.

Já a classe IAVLImage, utiliza a classe IAVLMapa para realizar o tratamento da imagem e localizar os candidatos a faixa. Para acessar a imagem, a classe IAVLWindow é responsável pela aquisição e exibição da imagem e através dela a classe IAVLImage consegue acesso aos frames coletados.

IAVLInterface é a classe responsável pela GUI(Graphical User Interface) do projeto, apresentando componentes como display LCD's, botões analógicos de posicionamento do centro e botões de comando do programa.

IAVLRobo é a classe que implementa os métodos e atributos para acionamento do robô . Nela estão implementados limites de velocidade de translação e de rotação, por exemplo.

Na Figura 14 constam todos os métodos e atributos de todas as classes desenvolvidas para o projeto IAVL.

6 RESULTADOS

Para a verificação dos algoritmos de navegação autônoma propostos foram efetuados diversos testes com o robô móvel com visão omnidirecional sobre uma pista definida no chão do laboratório.

Foram testadas diversas curvaturas com diferentes raios e retas com diferentes comprimentos. Foi testada a capacidade do robô se manter sempre sobre a marcação da pista mesmo em situações adversas.

Os testes mostraram o robô seguindo a faixa com velocidade de translação aproximadamente igual a 0,6 m/s durante grande parte do trajeto.

A Figura 19 mostra a GUI do programa. Nela são apresentados todos as informações e comandos relevantes. É interessante notar as subjanelas próximas ao robô e marcadas em vermelho que indicam uma região que contém a faixa.

A apresenta uma visão mais ampla do circuito e mostra que no trajeto planejado o robô teve que passar por curvas a direita, a esquerda, por sinuosidades e diferentes raios de curvatura.

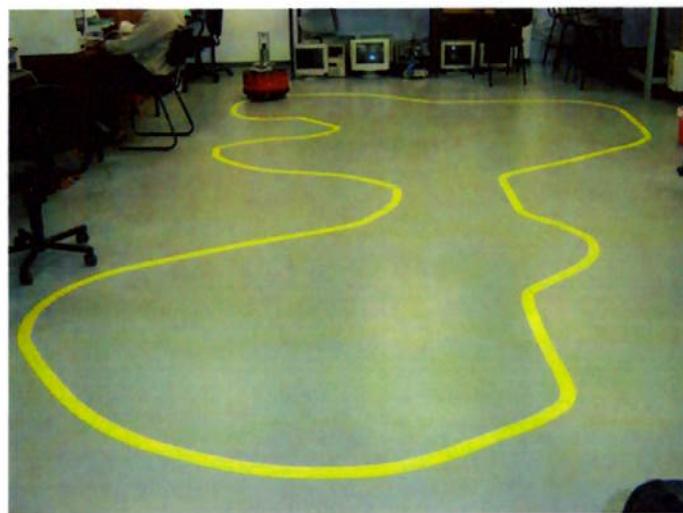


Figura 16 - Visão ampla do circuito de testes

Já na **Error! Reference source not found.** apresenta uma foto do robô seguindo a faixa em um trecho com curvas fechadas onde chega a ser necessário que o robô diminua a velocidade para executar as curvas.

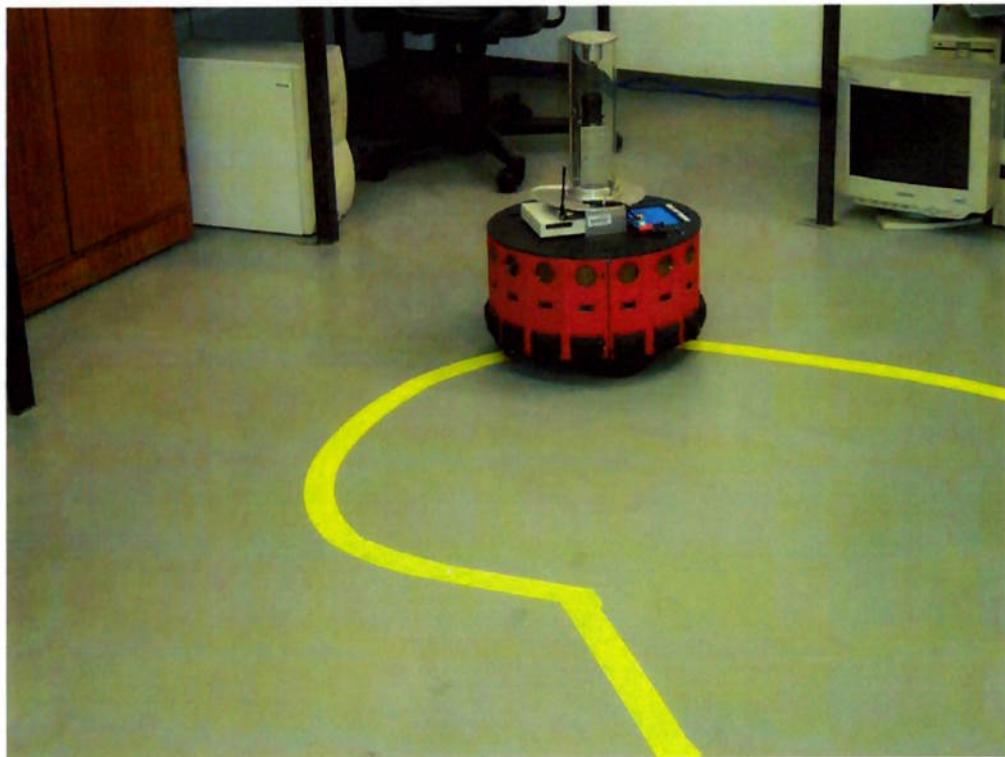


Figura 17 - Robô seguindo circuito

A Figura 18 apresenta duas leitura distintas do odômetro interno do robô. O gráfico nomeado como “*Círcuito*” foi obtido empurrando-se o robô por toda a trajetória de teste enquanto era feita a leitura do odômetro. Já o gráfico nomeado como “*Trajetória autônoma*” foi obtido durante a execução do programa de controle do robô. Essas medições são executadas com informações provenientes de encoders incrementais utilizados pelo robô, a utilização de tais encoders provoca diferentes medidas para diferentes posições iniciais. Assim, o deslocamento notado entre os dois gráficos pode ser atribuído a diferenças na posição inicial das duas medidas.

Além disso, é notável a diferença ocasionada pelo método dos mínimos quadrados, que leva o robô a fazer algumas curvas de modo aproximado.

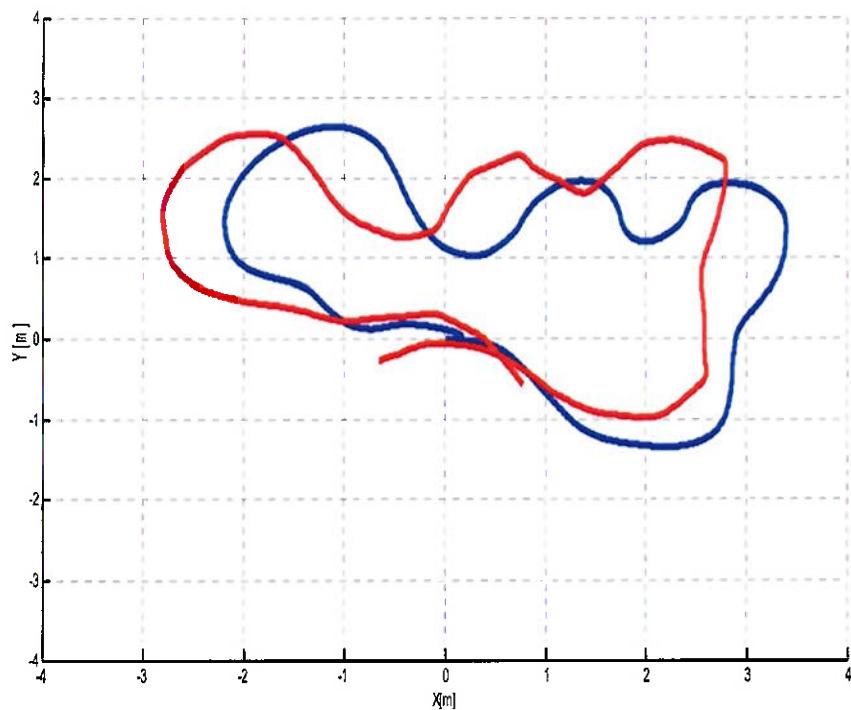


Figura 18 - Comparação entre a trajetória real e a trajetória autônoma

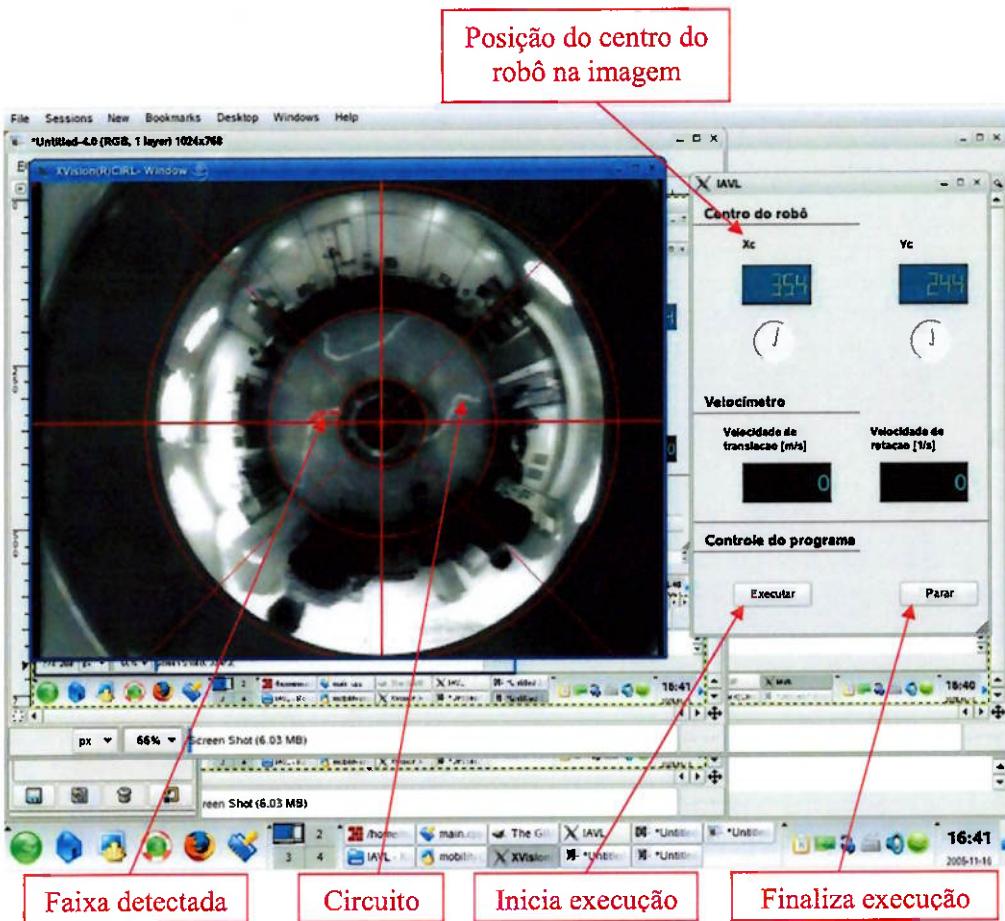


Figura 19 - Interface com o usuário

7 CONCLUSÃO

Esse relatório apresentou os resultados obtidos no projeto IAVL. Foram feitas descrições do método de resolução do problema, do projeto de software, dos testes executados e dos resultados obtidos.

Durante os testes executados em laboratório o robô MagellanPro percorreu um circuito marcado no chão através de uma fita utilizando apenas visão omnidirecional, sem se perder, mesmo quando passando por curvas de pequenos raios. Foram apresentados resultados obtidos com o odômetro do robô, que demonstraram que o robô segue a trajetória desejada.

Os resultados apresentados demonstraram a viabilidade da abordagem e abrem caminho para projetos que visem a navegação autônoma outdoor. Caso a transição para veículos outdoor seja efetuada com sucesso, o sistema poderá ser embarcado em veículos de passeio de modo a permitir que os mesmos percorram auto-estradas sem interferência humana. Essa possibilidade é confirmada pela exposição do projeto, em forma de poster, no XIV Congresso e Exposição Internacionais de Tecnologia da Mobilidade, organizado pela SAE BRASIL e que foi realizado entre os dias 22 e 24 de novembro em São Paulo.

8 ANEXO – CRONOGRAMA

O projeto foi concluído de acordo com o cronograma. Todos os milestones foram atingidos.

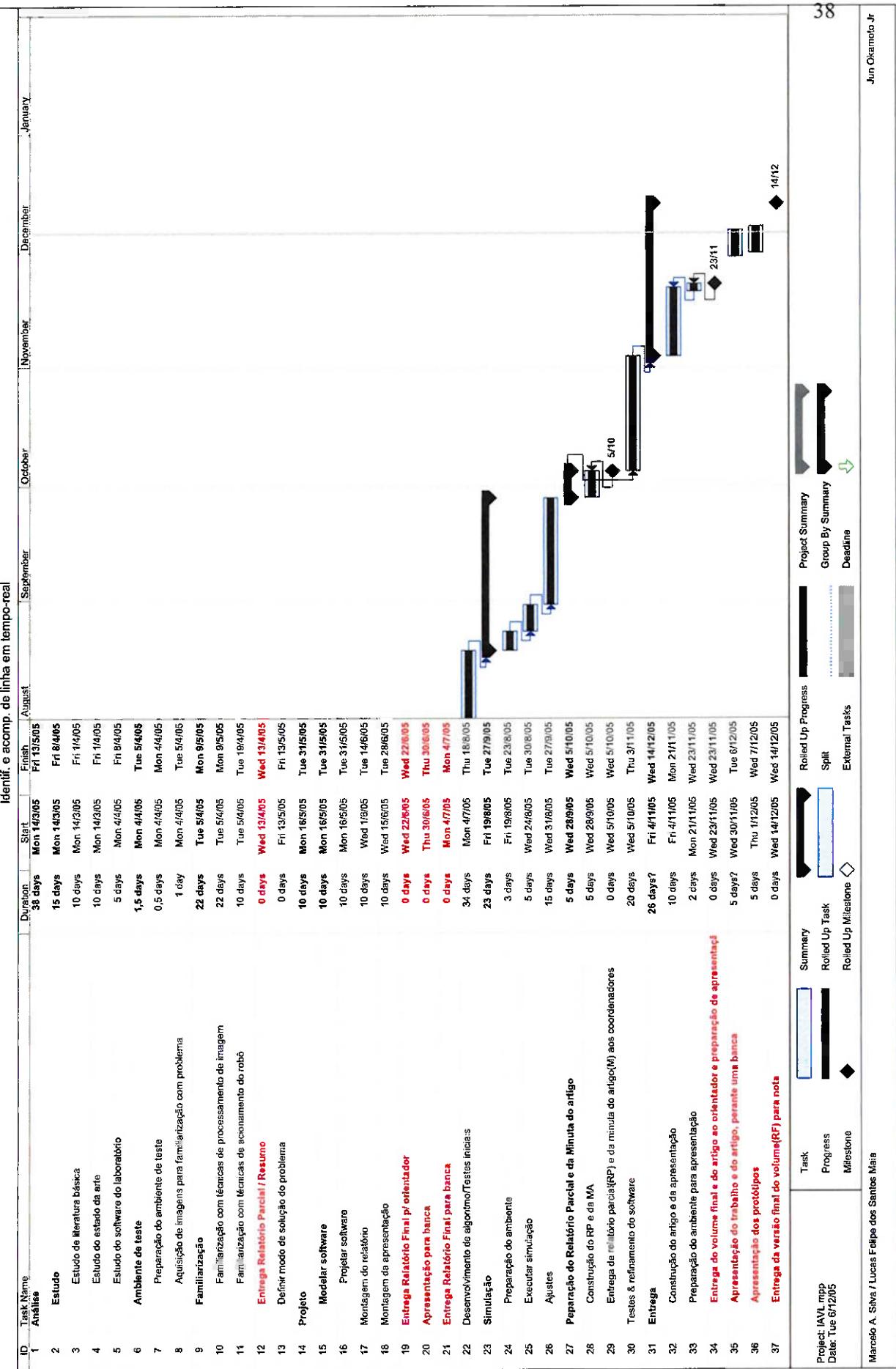
IAVL.mpp

Identif. e acomp. de linha em tempo-real

ID	Task Name	Duration	Start	Finish							
1	Análise	36 days	Mon 14/3/05	Fri 13/5/05							
2	Estudo	15 days	Mon 14/3/05	Fri 8/4/05							
3	Estudo da literatura básica	10 days	Mon 14/3/05	Fri 1/4/05							
4	Estudo do estado da arte	10 days	Mon 14/3/05	Fri 1/4/05							
5	Estudo do software do laboratório	5 days	Mon 4/4/05	Fri 8/4/05							
6	Ambiente de teste	1,5 days	Mon 4/4/05	Tue 5/4/05							
7	Preparação do ambiente de teste	0,5 days	Mon 4/4/05	Mon 4/4/05							
8	Aquisição de imagens para familiarização com problema	1 day	Mon 4/4/05	Tue 5/4/05							
9	Familiarização	22 days	Tue 5/4/05	Mon 9/5/05							
10	Familiarização com técnicas de processamento de imagem	22 days	Tue 5/4/05	Mon 9/5/05							
11	Familiarização com técnicas de funcionamento do robô	10 days	Tue 5/4/05	Fri 19/4/05							
12	Entrega Relatório Parcial / Resumo	0 days	Wed 13/4/05	Wed 13/4/05							
13	Definir modo de solução do problema	0 days	Fri 13/5/05	Fri 13/5/05							
14	Projeto	10 days	Mon 16/5/05	Tue 31/5/05							
15	Modelar software	10 days	Mon 16/5/05	Tue 31/5/05							
16	Projetar software	10 days	Mon 16/5/05	Tue 31/5/05							
17	Montagem do relatório	10 days	Wed 16/6/05	Tue 14/6/05							
18	Montagem da apresentação	10 days	Wed 15/6/05	Tue 28/6/05							
19	Entrega Relatório Final p/ orientadores	0 days	Wed 22/6/05	Wed 22/6/05							
20	Apresentação para banca	0 days	Thu 30/6/05	Thu 30/6/05							
21	Entrega Relatório Final para banca	0 days	Mon 4/7/05	Mon 4/7/05							
22	Desenvolvimento de algoritmo/testes iniciais	34 days	Mon 4/7/05	Thu 18/8/05							
23	Simulação	23 days	Fri 19/8/05	Tue 27/9/05							
24	Preparação do ambiente	3 days	Fri 19/8/05	Tue 23/8/05							
25	Executar simulação	5 days	Wed 24/8/05	Tue 30/8/05							
26	Ajustes	15 days	Wed 31/8/05	Tue 27/9/05							
27	Preparação do Relatório Parcial e da Minuta do artigo	5 days	Wed 28/9/05	Wed 5/10/05							
28	Convenção do RP e da MA	5 days	Wed 28/9/05	Wed 5/10/05							
29	Entrega de relatório parcial(RP) e da minuta do artigo(MA) aos coordenadores	0 days	Wed 5/10/05	Wed 5/10/05							
30	Testes & refinamento do software	20 days	Wed 5/10/05	Thu 31/10/05							
31	Entrega	26 days?	Fri 4/11/05	Wed 4/11/05							
32	Construção do site e da apresentação	10 days	Fri 4/11/05	Fri 21/11/05							
33	Preparação do ambiente para apresentação	2 days	Mon 21/11/05	Wed 23/11/05							
34	Entrega do volume final e do artigo ao orientador e preparação da apresentação	0 days	Wed 23/11/05	Wed 23/11/05							
35	Apresentação do trabalho e do artigo, perante uma banca	5 days?	Wed 30/11/05	Tue 6/12/05							
36	Apresentação dos protótipos	5 days	Thu 1/12/05	Wed 7/12/05							
37	Entrega da versão final do volume(RF) para nota	0 days	Wed 14/12/05	Wed 14/12/05							
Project IAVL.mpp Date: Tue 01/12/05		Task	Summary	Rolled Up Progress	Project Summary	Group By Summary	External Tasks	Deadline			
Marcelo A. Silva / Lucas Felipe dos Santos Maia		Progress	Rolled Up Task	Split							
		Milestone	Rolled Up Milestone	External							

AVL.mpp

Identif. e acompl. de linha em tempo-real



38

Jún Okamoto Jr

Manoel A. Silva / Lucas Felipe dos Santos Maia

9 BIBLIOGRAFIA

APOSTOLOFF, N.; ZELINSKY, A. **Vision In and Out of Vehicles: Integrated Driver and Road Scene Monitoring**, The International Journal of Robotics Research, Vol23, no4-5, Abril-Maio/2004.

AUFRÉRE, R.; CHAPUIS, R.; CHAUSSE, F. **A fast and robust vision based algorithm**; Intelligent Vehicles Symposium 2000, Proceedings of the IEEE; October 3-5 2000. Page(s) 192-197.

BERTOZZI, M.; BROGGI, A. **GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection**; Image Processing, IEEE Transactions on, Volume 7, Issue 1, Jan. 1998 Page(s):62 - 81

CHAPUIS, R.; AUFRERE, R.; CHAUSSE, F. **Accurate Road Following and Reconstruction by Computer Vision**; Intelligent Transportation Systems, IEEE Transactions on; Volume 3, Issue 4, Dec. 2002 Page(s):261 - 270

CORBA.ORG. Disponível em: <<http://www.corba.org/>>. Acesso em: 01 jun. 2005.

FODA, S.G.; DAWOUD, A.K. **Highway Lane Boundary Determination For Autonomous Navigation**; Communications, Computers and signal Processing, 2001. PACRIM. 2001 IEEE Pacific Rim Conference on Volume 2, 26-28 Aug. 2001 Page(s):698 - 702 vol.2

GERN, A.; FRANKE, U.; LEVI, P. **Advanced lane recognition-fusing vision and radar**; Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE, 3-5 Oct. 2000 Page(s):45 - 51

GONZALEZ, J.P.; OZGUNER, U. **Lane detection using histogram-based segmentation and decision trees**, Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE, 1-3 Oct. 2000 Page(s):346 - 351

GONZÁLEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**, São Paulo: Editora Edgard Blücher LTDA, 2000.

GRAND CHALLENGE. Disponível em:
<<http://www.darpa.mil/grandchallenge/>>. Acesso em 27 jun. 2005.

GRASSI JR., V.; DECCÓ, C. C. G.; OKAMOTO JR., J.; PORTO, A. J. V. **Desenvolvimento de um sistema de visão omnidirecional; XVI Congresso Brasileiro de Engenharia Mecânica, Novembro 2001, Uberlândia, MG.**

HUANG, S.; CHEN, C.; HSIAO, P.; FU, L. **On-board vision system for lane recognition and front-vehicle detection to enhance driver's awareness;** Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on; Volume 3, 26 April-1 May 2004 Page(s):2456 - 2461 Vol.3

HUH, K.; PARK, J.; HONG, D.; CHO, D. D.; PARK, J. H. **Development of a vision-based lane detection system considering configuration aspects.** Optics and Lasers in Engineering, 17 February 2005

iROBOT – ROBOTS FOR THE REAL WORLD. Disponível em:
<http://www.rwii.com/>. Acesso em 3 jun. 2005.

JEONG, S. G.; KIMM, C. S.; LEE, D. Y.; HA, S. K.; LEE, D. H. LEE, M. H.; HASHIMOTO, H. **Real-time lane detection for autonomous vehicle;** Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on; Volume 3, 12-16 June 2001 Page(s):1466 - 1471 vol.3

KLUGE, K.; JOHNSON, G. **Statistical characterization of the visual characteristics of painted lane markings;** Intelligent Vehicles '95 Symposium., Proceedings of the, 25-26 Sept. 1995 Page(s):488 – 493

KREUCHER, C.; LAKSHMANAN, S. **LANA: a lane extraction algorithm that uses frequency domain features;** Robotics and Automation, IEEE Transactions on; Volume 15, Issue 2, April 1999 Page(s):343 - 350

NOVELL: SUSE LINUX. Disponível em: <http://www.novell.com/pt-br/linux/suse/>. Acesso em: 5 jun. 2005.

SAE BRASIL. Disponível em: <http://www.saебrasil.org.br>. Acesso em: 10 nov. 2005.

THE XVISION2 PROJECT. Disponível em:
<http://www.cs.jhu.edu/CIRL/XVision2/>. Acesso em: 20 maio 2005.

VIKRAM, S. **Master's thesis.** Disponível em: www.d.umn.edu/~rmaclin/student-publications/svikram-ms.ps. Acesso em: 18 jun. 2005.

WOLFE, V. F.; DIPIPPO, L. C.; GINIS, R.; SQUADRITO, M.; WOHLEVER, S.; ZYKH, I.; JOHNSTON, R. **Real-Time CORBA**. Third IEEE Real-Time Technology and Applications Symposium, Proceedings of the; Montreal, Canada, June 1997. Disponível em:
<<http://citeseer.ist.psu.edu/article/wolfe97realtime.html>>. Acesso em: 5 nov. 2005.