

FABIANO ROGÉRIO CORRÊA

MAPEAMENTO DE AMBIENTES COM SISTEMA
DE VISÃO OMNIDIRECIONAL

Trabalho de Formatura apresentado à Escola
Politécnica da Universidade de São Paulo
para obtenção do título de graduado

SÃO PAULO
2002

9,0 (nae)
9,2 (moodle)
nota final 9,2 (moodle)
nota

FABIANO ROGÉRIO CORRÊA

MAPEAMENTO DE AMBIENTES COM SISTEMA
DE VISÃO OMNIDIRECIONAL

Trabalho de Formatura apresentado à Escola
Politécnica da Universidade de São Paulo
para obtenção do título de graduado.

Área de Concentração:
Engenharia Mecânica

Orientador:
Prof. Doutor
Jun Okamoto Junior

SÃO PAULO
2002

RESUMO

É fundamental para a navegação autônoma de robôs a obtenção dinâmica de informações sobre o ambiente no qual eles estão inseridos. Realizar um mapeamento do espaço e atualizá-lo iterativamente é uma das possíveis soluções para este problema. Tal mapeamento pode ser realizado através da análise e integração dos dados obtidos a partir de um sensor (ou de diversos sensores) embarcados no robô. Sistemas de visão fornecem grande quantidade de informação que pode ser utilizada para o mapeamento de ambientes. A obtenção de informações a cerca da distância entre os objetos que compõem o ambiente e o robô, pode ser realizada por sistemas de visão estéreo compostos por duas ou mais câmeras, ou por uma câmera que se movimenta. No presente trabalho, é utilizado um sistema de visão omnidirecional, com uma única câmera, onde o par de imagens necessário para compor a visão estéreo é obtido pelo deslocamento conhecido de um robô móvel. Para que o mapeamento do ambiente possa ser utilizado por um robô autônomo, este deve possuir certa flexibilidade, devido às incertezas inerentes aos sensores. Neste estudo propõe-se a aplicação de uma ferramenta que possibilite a incorporação dessas incertezas e permita a obtenção de modelos probabilísticos de ambientes desconhecidos, para um sistema com visão omnidirecional. O projeto segue uma linha de pesquisa que vem sendo desenvolvida no Grupo de Percepção Avançada no Laboratório de Automação e Sistemas do Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos da Escola Politécnica da USP. Este grupo trabalha com um robô que possui um sistema de visão omnidirecional e comunica-se com uma rede local de computadores, na qual é feito todo o processamento das imagens obtidas.

SUMÁRIO

RESUMO	1
SUMÁRIO	2
1 INTRODUÇÃO	1
2 SENSORES: VISÃO ESTÉREO OMNIDIRECIONAL.....	4
2.1 Sensoriamento de ambientes	4
2.2 Visão Omnidirecional.....	4
2.3 Visão Estéreo Omnidirecional.....	5
2.4 Extração de Características – SUSAN	6
2.5 Correspondência.....	8
2.5.1 Curvas Epipolares	8
2.5.2 Correlação	11
2.6 Triangulação	12
3 MAPEAMENTO DO AMBIENTE: GRADES DE OCUPAÇÃO	14
3.1 Mapeamento de Ambientes	14
3.2 Grades de Ocupação	16
3.3 Sensoriamento	18
3.4 Modelos do Sensor – Função densidade de probabilidade.....	19
3.5 Inferências sobre o Estado do Mundo – Processo de Estimação Bayesiano	23
3.6 Regra de Decisão – (MAP – máximo <i>a posteriori</i>).....	24

3.7 Atualização dos Mapas.....	24
4 RESULTADOS.....	26
4.1 Ambiente de desenvolvimento Khoros	26
4.2 Sistema de Visão	29
4.3 Grade de Ocupação.....	29
4.4 Ambiente virtual.....	30
4.5 Ambiente Real.....	32
5 CONCLUSÃO	39
6 BIBLIOGRAFIA	40

INTRODUÇÃO

Um robô móvel autônomo deve ser capaz de se deslocar no ambiente baseando-se apenas nas informações por ele adquiridas através das leituras dos sensores que possui. Estas leituras devem ser contínuas para que o robô possa interagir dinamicamente com o ambiente no qual está inserido, pois o próprio ambiente está sujeito a mudanças com o passar do tempo. As informações dos sensores podem ser utilizadas para complementar os dados conhecidos de antemão sobre o local de trabalho (como por exemplo plantas do local) possibilitando a identificação e correspondência de uma entidade real e sua representação disponibilizada pelo robô e assim, podendo localizar-se no ambiente para a realização da tarefa especificada. Pode também, ele próprio criar uma representação do ambiente com as leituras obtidas por seus sensores, e assim ser capacitado a explorar e atuar em locais desconhecidos *a priori*. Esta última abordagem incorpora muito mais o conceito de autonomia em robôs móveis. Nesta linha, o robô realiza uma navegação autônoma e vai acumulando informações sobre o meio em que se encontra e, a partir destas informações, atua para realizar sua missão. Duas formas de abordar a navegação autônoma [14] são: através do comportamento reativo, onde o robô age conforme a interpretação dos dados mais recentes vindos do sensor segundo comportamentos hierárquicos pré-definidos [1][2][15], e outra, através da criação de um modelo ou representação interna do mundo, como por exemplo através da construção de um mapa.

As tarefas realizadas pelo robô podem utilizar informações de vários sensor, tanto do mesmo tipo, como um cinturão de sonares, quanto informações complementares como sonar e visão estéreo. Dentre os diversos tipos de sensores, os sistemas de visão são bastante apropriados para a tarefa de mapeamento por fornecem grande

quantidade de informação em cada leitura, permitindo construir uma representação detalhada do meio. A densidade de pontos localizada é diretamente relacionada à qualidade do mapa construído. Para a construção de mapas, é necessário obter-se dos sensores informações sobre a distância do sensor aos objetos presentes no ambiente. Para isso, utiliza-se visão estéreo que necessita de um par de imagens para extrair a informação de profundidade das imagens planas. Pode ser feita por duas ou mais câmeras “fixas”, ou por uma câmera que se movimenta. O critério de movimentação aqui diz respeito ao modo como o par estéreo é adquirido, não levando em consideração o campo visual que o robô possui. O sistema de visão formado por uma câmera e um espelho hiperbólico, alinhados sobre o mesmo eixo vertical, permite a obtenção de um campo visual de 360°, e pode-se realizar estéreo deslocando-se o robô entre a aquisição de uma imagem e outra.

Para que o mapeamento do ambiente possa ser realizado por um robô autônomo, e utilizado por ele para cumprir tarefas como exploração ou navegação, o mapa deve possuir certa flexibilidade, para acomodar as incertezas inerentes aos sensores e a própria movimentação do robô. Neste estudo propõe-se a aplicação de uma ferramenta que possibilite a incorporação dessas incertezas e permita a obtenção de modelos probabilísticos de ambientes desconhecidos, para um sistema com visão omnidirecional.

Este trabalho segue uma linha de pesquisa que vem sendo desenvolvida no Grupo de Percepção Avançada no Laboratório de Automação e Sistemas do Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos da Escola Politécnica da USP. Este grupo desenvolveu este sistema de visão omnidirecional e utilizou como sensor para a realização de tarefas como controle servo-visual de um robô e rastreamento de um alvo pré-determinado. O laboratório possui um robô (Base Móvel) com este sistema de visão omnidirecional embarcado que comunica-se com uma rede local de computadores, na qual é feito todo o processamento das imagens obtidas.

O presente trabalho tem como enfoque a obtenção de uma representação interna por parte do robô, ou seja, de um modelo do ambiente no qual ele se encontra. Este modelo será construído a partir da utilização de um sensor de visão estéreo

omnidirecional, que faz todo processamento para obtenção das informações de profundidade diretamente nas imagens adquiridas. Este tipo de sensoriamento será modelado segundo uma abordagem probabilística buscando incorporar as vantagens que o sensor estéreo omnidirecional possui para obter os mapas do ambiente dividido em células que indicam seus estados de ocupação, buscando assim incorporar as incertezas presentes na percepção do mundo pelo robô.

SENSORES: VISÃO ESTÉREO OMNIDIRECIONAL

2.1 Sensoriamento de ambientes

A presença de sensores em robôs autônomos em geral, e principalmente naqueles móveis, é essencial para que seja possível adquirir informações sobre o seu estado, sua localização e os arredores onde se encontra. Muitos são os tipos de sensores utilizados, como os sonares, lasers e visão computacional. Cada sensor possui limitações inerentes, nelas incluídas as incertezas do processo de aquisição das informações e na confiabilidade das mesmas. Extrair informações a partir dos dados obtidos através do sensoriamento e encontrar os limites de validade delas é no que consiste a modelagem do sensor, que será feito em duas etapas: o sensor ideal e o real.

Com o crescente desenvolvimento da capacidade de processamento de dados por parte dos computadores atuais, o sensoriamento através de sistemas de visão computacional vêm se tornando cada vez mais atraentes. Além disso, os estudos nesta área vem compondo uma teoria cada vez mais sólida permitindo a realização de processamentos mais adequados e eficientes das imagens.

2.2 Visão Omnidirecional

Sistemas de visão omnidirecional possibilitam a aquisição de imagens com um campo visual de 360° ao redor do robô no qual o sistema é montado.

Um modo de obter-se uma imagem com campo de visão de 360° é através da composição de diversas imagens numa única só, adquiridas por uma câmera que gira com velocidade angular constante ao redor de um eixo vertical. Apesar da imagem

omnidirecional obtida possuir uma alta resolução, este tipo de sistema não é adequado para aplicações onde o ambiente é dinâmico pois a imagem omnidirecional completa não é obtida em tempo real. Outra forma de se combinar imagens é por meio de um arranjo com diversas câmeras apontadas para diferentes direções o que possibilita a aquisição da imagem omnidirecional em tempo real com alta resolução. No entanto, a montagem de tal sistema, além de possuir dificuldades na calibração das câmeras, não é um arranjo compacto para ser usado em um robô móvel.

Uma outra forma de se obter imagens omnidirecionais em tempo real é utilizar uma lente do tipo “olho de peixe” montada em uma câmera fixa.. Entretanto, este tipo de lente possui um custo alto se comparado com lentes convencionais. Além do mais, as imagens adquiridas possuem resoluções distintas, com boa qualidade apenas na região central, embora os objetos do ambiente que são de interesse para diversas aplicações aparecem representados na periferia da imagem, onde a resolução é baixa.

Um sistema de visão capaz de produzir imagens omnidirecionais em tempo real também pode ser obtido através do uso combinado de espelhos convexos e lentes. A este tipo de sistema de visão dá-se o nome de sistema de visão omnidirecional catadióptrico. Neste sistema, uma câmera é montada em frente a um espelho convexo de forma a observar a imagem refletida no espelho. A forma do espelho utilizada pode ser projetada para que se obtenha imagens com características apropriadas para a aplicação que se fará do sistema de visão.

2.3 Visão Estéreo Omnidirecional

Existem diversos sistemas de visão computacional que se distiguem pelo modo como as imagens são obtidas ou pelo número de imagens necessárias para obter as informações do ambiente. O ambiente pode ser mapeado com diversas imagens tiradas de algumas câmeras dispostas em ângulos diferentes ou através da movimentação das mesmas ou do robô que as possui; pode também ser mapeado por meio de uma câmera fixa direcionada para um espelho convexo que reflita o ambiente. Estes sistemas são chamados de catadióptricos.

Uma imagem plana não é suficiente – na maior parte dos casos – para fornecer informações sobre a distância dos objetos à câmera. Para obter esta informação, são utilizadas, no mínimo, duas imagens, sejam elas adquiridas por uma ou mais câmeras, pela movimentação da(s) mesma(s) ou do robô. Usar duas imagens, conhecendo-se o deslocamento do observador na tomada de uma para a outra, para obter a distância dos objetos é no que consiste a visão estéreo.

Utilizando os conceitos de visão estéreo – também conhecido como estereopsia – é possível obter as distâncias dos objetos ao redor do robô. A visão estéreo é obtida com pelo menos um par de imagens que, neste caso, são obtidas por uma câmera fixa que capta imagens conforme o robô desloca-se no plano em determinada direção. O processamento deste par de imagens adota a geometria projetiva e utiliza-se das linhas epipolares para obter estas informações. Com as distâncias, é possível construir mapas que serão armazenados, provavelmente na rede externa de computadores, e atualizados segundo as contínuas imagens captadas pela câmera.

Em [20], foi criado um programa de computador que carrega duas imagens omnidirecionais, seleciona alguns pontos nas imagens e realiza a correspondências dos pontos na imagem para que seja possível encontrar a distância do ponto ao robô, utilizando projeção geométrica, montando assim um mapa com as distâncias obtidas. Foi realizado um trabalho de conhecimento e verificação das técnicas mais adequadas para o sistema de visão omnidirecional. O presente trabalho parte deste conhecimento adquirido, tentando, não apenas reproduzi-lo, mas expandi-lo com o intuito de adequá-lo a ser utilizado para um mapeamento do ambiente e uma possível navegação em tempo real.

O processo obtenção das distâncias dos objetos no ambiente, em relação ao robô, por meio da visão estéreo consiste basicamente em três etapas: a extração de características, a correspondência e a triangulação.

2.4 Extração de Características – SUSAN

Primeiramente, é necessário escolher uma característica da imagem, como as bordas, para se trabalhar. Através do processamento da imagem, procurando uma variação

nas tonalidades de cinza das mesmas, é possível detectar as bordas dos objetos. O presente trabalho utiliza um algoritmo de processamento de baixo nível conhecido como SUSAN (Smallest Univalued Segment Assimilating Nucleus) que usa uma máscara circular que compara a diferença de intensidade de luz entre o pixel do centro da máscara e os do resto da mesma, obtendo assim um valor que caracteriza a quantidade de pontos da máscara que pertencem ao objeto. Quando esta quantidade é a metade da máxima possível (a máscara inteira), significa que o centro da máscara está na borda de um objeto.

Para um sistema de visão estéreo, é necessária a escolha de uma característica, algo que possa ser identificado através de operações matemáticas (algoritmos computacionais) numa imagem. Uma característica pode ser um ponto, uma reta, uma área. A sua escolha está diretamente relacionada ao tipo de mapa do ambiente que deseja-se obter, com relação à sua densidade.

Segundo estudos feitos em [20], escolheu-se como características os pontos nas bordas e quinas dos objetos no ambiente. O algoritmo escolhido foi o SUSAN descrito em [24].

SUSAN é um acrónimo para Smallest Univalued Segment Assimilating Nucleus. O algoritmo funciona da seguinte forma: o núcleo de uma máscara circular percorre todos os pixels da imagem. Em cada pixel, compara-se a intensidade de cinza, $I(\vec{r})$, dos pixels dentro da máscara com a intensidade do núcleo segundo um limiar pré-determinado, t .

$$c(\vec{r}, \vec{r}_0) = \begin{cases} 1, & \text{se } |I(\vec{r}) - I(\vec{r}_0)| \leq t \\ 0, & \text{se } |I(\vec{r}) - I(\vec{r}_0)| > t \end{cases}, \text{ onde } \vec{r}_0 \text{ é a posição do núcleo na imagem e } \vec{r} \text{ a}$$

posição de um pixel da máscara.

$$n(\vec{r}_0) = \sum_{\vec{r}} c(\vec{r}, \vec{r}_0)$$

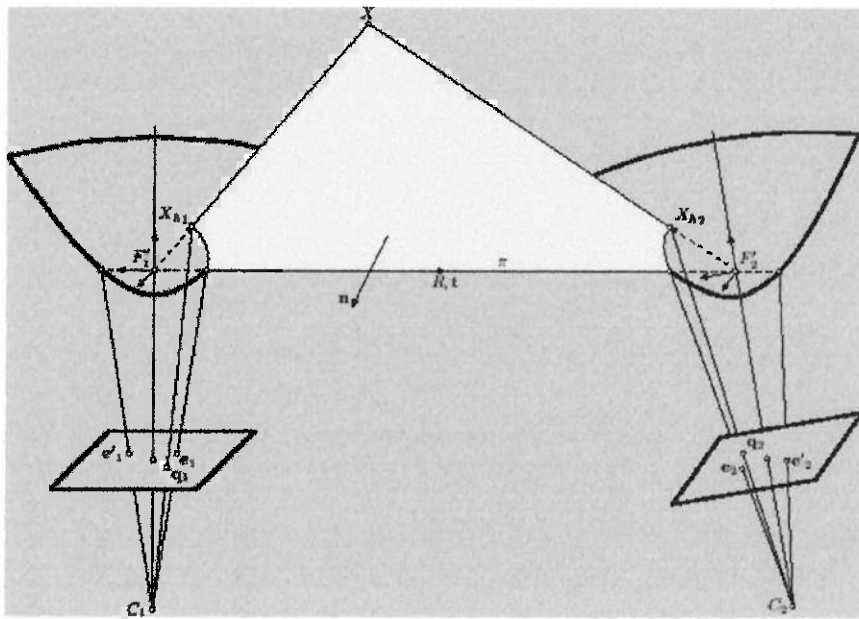
$$R(\vec{r}_0) = \begin{cases} g - n(\vec{r}_0), & \text{se } n(\vec{r}_0) < g, \text{ onde } g = 3/4n_{\text{máx}} \\ 0, & \text{do contrário} \end{cases}$$

2.5 Correspondência

Escolhida uma característica na primeira imagem, é preciso encontrar a característica correspondente na segunda imagem para compor a triangulação, e assim encontrar sua distância em relação ao robô. Existem vários métodos para isto. O adotado neste trabalho visa reduzir a busca no plano para a busca numa curva, a epipolar. Para cada ponto da primeira imagem, é possível traçar uma curva epipolar na segunda sobre a qual encontra-se o ponto correspondente [25]. Fazendo uma busca procurando a correspondência dos pixels da janela centrada nos pontos desta curva, estabelece-se como critério para determinar que o correspondente foi encontrado a maior porcentagem de correspondências exatas entre os pixels na janela de busca.

2.5.1 Curvas Epipolares

Por meio de duas imagens omnidirecionais e do deslocamento da câmera entre a aquisição das imagens, é possível por técnicas de projeção geométrica e álgebra linear, dado um ponto na primeira imagem, traçar uma curva na segunda imagem na qual o ponto correspondente a ele se encontra. Assim, faz-se uma redução no espaço de busca de duas para uma dimensão. Uma propriedade importante da geometria epipolar é que todas as curvas se cruzam em dois pontos específicos na imagem – ou plano de projeção – denominados epipólos. Os epipólos são determinados apenas pela configuração do sistema, ou seja, pela translação entre as duas aquisições da imagem. Os epipólos, junto com um ponto no ambiente, determinam o plano epipolar que “corta” os espelhos dos dois sistemas de visão, gerando uma curva, que projetada sobre a imagem, produz a curva epipolar. Todos os cálculos estão explicitados em [25].



Sistema de visão com a geometria epipolar [25].

Define-se t como sendo a translação da câmera, da posição 1 para a 2. A matriz R é a matriz de rotação entre estas mesmas posições. Os vetores X_{hi} são vetores com centro no foco do espelho e extremidade na superfície do mesmo. A coplanaridade entre os vetores t , X_{h1} e X_{h2} definem o plano epipolar π , que intercepta o espelho produzindo uma cônica no plano de projeção desta intersecção. Esta coplanaridade é escrita da seguinte forma:

$$X_{h2} R(t \wedge X_{h1}) = 0$$

Sendo, $S = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$, a matriz anti-simétrica, pode-se escrever a restrição

de coplanaridade em forma matricial:

$$X_{h2}^T E X_{h1} = 0, \text{ onde } E = R.S \text{ é a matriz essencial.}$$

Uma única cônica é atribuída a segunda imagem a cada ponto q_1 na primeira imagem por meio da equação, $q_2^T A_2(E, q_1) q_2 = 0$.

É necessário encontrar $A_2(E, q_1)$ que normalmente é uma função não-linear da matriz essencial E e do ponto q_1 .

Procurando uma equação da projeção ortográfica do espaço cônico, começa-se por:

$$x_2^T A_{x_2} x_2 = 0$$

O vetor t , X_{h1} e X_{h2} formam o plano π ; o vetor normal a este plano é dado por n_1 .

$$n_1 = t \wedge X_{h1}, e \ n_2 = Rn_1 = R(t \wedge X_{h1}) = RSX_{h1} = EX_{h1}$$

O plano π pode ser escrito, segundo as coordenadas da sistema em 2 como: $px + qy + sz = 0$

Escrevendo a equação da cônica na forma quadrática, tem-se:

$$(p^2b^2 - s^2a^2)x^2 + 2pqb^2xy + (q^2b^2 - s^2a^2)y^2 - 2spb^2ex - 2sqb^2ey - s^2b^4 = 0$$

E comparando-a com outra anterior, tem-se:

$$A_{x_2} = \begin{bmatrix} p^2b^2 - s^2a^2 & pqb^2 & -psb^2 \\ pqb^2 & q^2b^2 - s^2a^2 & -qsb^2 \\ -psb^2 & -qsb^2 & s^2b^4 \end{bmatrix}$$

Admitindo que $s \neq 0$, ou seja, o eixo do espelho não está contido no plano epipolar, tem-se, expressando X_{h2} como uma função linear de x_2 :

$$X_{h2} = \begin{bmatrix} x \\ y \\ \frac{-px - qy}{s} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{-p}{s} & \frac{-q}{s} & 0 \end{bmatrix} x_2$$

$$\text{Então, } \mathfrak{I}^+(R_C^T K^{-1} q_2) R_C^T K^{-1} q_2 = \left(X_{h2} + \begin{bmatrix} 0 \\ 0 \\ 2e \end{bmatrix} \right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{-p}{s} & \frac{-q}{s} & 2e \end{bmatrix} x_2$$

$$\text{E pode-se admitir que: } x_2 \cong N R_C^T K^{-1} q_2, \text{ onde } N = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{p}{2se} & \frac{q}{2se} & \frac{1}{2se} \end{bmatrix}.$$

Finalmente, $q_2^T K^{-T} R_C N^T A_{x2} N R_C^T K^{-1} q_2 = 0$, onde:

$$A_2 = K^{-T} R_C B_2 R_C^T K^{-1}, \text{ e}$$

$$B_2 = N^T A_{x2} N = \begin{bmatrix} -4s^2 a^2 e^2 + p^2 b^4 & pqb^4 & psb^2(-2e^2 + b^2) \\ pqb^4 & -4s^2 a^2 e^2 + q^2 b^4 & qsb^2(-2e^2 + b^2) \\ psb^2(-2e^2 + b^2) & qsb^2(-2e^2 + b^2) & s^2 b^4 \end{bmatrix}$$

2.5.2 Correlação

A correlação é feita buscando-se nas janelas epipolares com um retângulo cujo centróide se encontra sobre a curva. Os pixels desta janela são comparados com os pixels de uma janela de mesmo tamanho centrada do ponto destacado da primeira imagem. A correspondência é encontrada maximizando a equação abaixo:

$$C_{12}(\tau) = \frac{1}{k} \sum_{u_1=-N}^{+N} \sum_{v_1=-P}^{+P} \left(I_1(u_1 + u_0, v_1 + v_0) - \overline{I_1(u_0, v_0)} \right) \left(I_2(u_1 + u_0 + \tau, v_1 + v_0) - \overline{I_2(u_0 + \tau, v_0)} \right)$$

,onde

$$K = (2N+1)(2P+1)\sigma_1(u_0, v_0)\sigma_2(u_0+\tau, v_0)$$

$$\overline{I_1(u_0, v_0)} = \frac{1}{(2P+1)(2N+1)} \sum_{u_1=-N}^{+N} \sum_{v_1=-P}^{+P} I_1(u_1 + u_0, v_1 + v_0)$$

$$\sigma_1^2(u_0, v_0) = \frac{1}{(2P+1)(2N+1)} \sum_{u_1=-N}^{+N} \sum_{v_1=-P}^{+P} \left(I_1(u_1 + u_0, v_1 + v_0) - \overline{I_1(u_0, v_0)} \right)^2$$

2.6 Triangulação

E, finalmente, tendo os pares de pontos correspondentes a um mesmo ponto real, é possível, por meio da construção geométrica de triângulos, determinar a distância do ponto real à câmera.

Com os pontos correspondentes nas imagens, é necessário primeiro transformá-los nos vetores que apontam para o ponto nas coordenadas do referencial do mundo e cujo comprimento vai do foco do espelho à sua superfície. Isto pode ser feito segundo as equações abaixo que indicam transformações de coordenadas:

Mundo-Espelho: $X_M = R_M(X - t_M)$

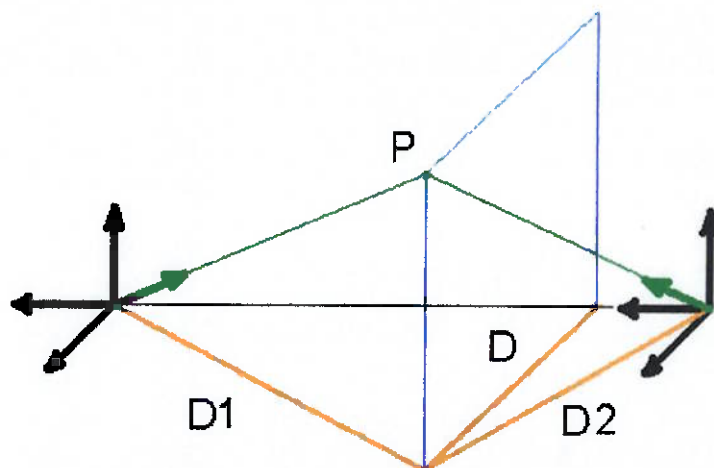
Espelho-Superfície:

$$X_h = \mathfrak{I}^-(X_M) X_M, \text{ onde } \mathfrak{I}^\pm(v) = \frac{b^2(\pm e v_3 + a\|v\|)}{b^2 v_3^2 - a^2 v_1^2 - a^2 v_2^2}$$

Superfície-Câmera (normalizado): $u = (1/z_c) R_c(X_h - t_c)$

Plano normalizado-Pixels: $q = Ku$

A partir dos vetores X_h das duas imagens (indicados em verde na figura abaixo), pode-se encontrar a distância D_2 .



Triangulação para encontrar a distância

A triangulação é feita segundo as fórmulas abaixo (nota-se que este é um caso especial no qual a translação coincide com um dos eixos coordenados. Mas é possível especificar para cada caso um conjunto de equações):

$$\frac{Y_{h_i}}{X_{h_i}} = \frac{D}{D_{x_i}}$$

$$D_{x_2} + (-D_{x_1}) = t$$

$$D = t \frac{Y_{h_1} Y_{h_2}}{X_{h_1} Y_{h_2} + X_{h_2} Y_{h_1}}$$

$$D_i = \sqrt{D^2 \left(1 + \frac{X_{h_i}^2}{Y_{h_i}^2} \right)}$$

MAPEAMENTO DO AMBIENTE: GRADES DE OCUPAÇÃO

3.1 Mapeamento de Ambientes

Para que um robô tenha autonomia, é indispensável que ele consiga obter informações a cerca do ambiente no qual está inserido. Precisa adquirir dados através dos seus sensores e extrair destes dados, informações sobre a estrutura do ambiente ao seu redor, ou sobre os elementos que o compõe, para planejar trajetórias de navegação ou realizar outras tarefas específicas.

No presente contexto, a tarefa de mapeamento concerne ao problema de construção de modelos espaciais de ambientes físicos através de informações adquiridas por meio de sensores presentes em robôs móveis. Tal problema é geralmente considerado como um dos mais importantes na busca da criação de robôs completamente autônomos. Segundo Thrun, tem-se atualmente métodos robustos para mapeamento de ambientes estáticos, estruturados e de tamanho reduzido, mas é ainda um problema em aberto o mapeamento de ambientes desestruturados, dinâmicos e de larga escala.

Entre as décadas de 80 e 90, este campo de pesquisa em mapeamento robótico era largamente dividido em abordagens métricas e topológicas. Os mapas métricos capturam as propriedades geométricas do ambiente, enquanto os topológicos, descrevem a forma como diversos lugares estão conectados ou relacionados. A distinção entre um tipo e outro não é muito clara, pois o mapa topológico baseia-se em informações métricas; na prática, os mapas métricos possuem maior resolução que os topológicos, tendo por isso um custo computacional maior embora possibilitem a solução de várias tarefas complicadas como o problema da correspondência.

A partir de 1990, este campo foi dominado por técnicas probabilísticas e atualmente o estado da arte em mapeamento robótico está quase que completamente nestes tipos de algoritmos. Uma série de trabalhos seminais, como os do Cheeseman [3], introduziram uma poderosa fundamentação estatística para resolver simultaneamente os problemas de mapeamento e de localização do robô no próprio mapa em construção.

O mapeamento é realizado através da informação dos sensores que o robô possui, que estão sujeitos a erros, denominados ruído de medição. Os sensores também apresentam limitações de alcance que torna necessário que o robô se locomova pelo ambiente durante a construção do modelo do mesmo. A movimentação do robô é de extrema importância para a criação deste modelo, mas também está sujeita a erros sendo o controle do robô insuficiente para determinar sua posição (localização e orientação).

Um dos desafios propostos pelo problema do mapeamento está relacionado à natureza dos ruídos de medição que são estatisticamente dependentes, pois as informações sobre o deslocamento do robô, por exemplo, são utilizadas para interpretar as leituras do sensor e compor o mapa do ambiente. Para acomodar este tipo de erro, os algoritmos de mapeamento tendem a ser complexos tanto do ponto de vista matemático quanto de implementação.

Outro aspecto a ser levado em conta é quanto à dimensionalidade do problema. Representar um ambiente, dependendo do tipo de representação escolhida, pode acarretar em cálculos bastante complexos. Pode-se representar um ambiente de maneira topológica (corredores, salas, portas), um ambiente plano de maneira mais detalhada, ou representar todo o espaço. Cada característica é considerada uma dimensão para o problema de estimação por meio de estatística.

A correspondência é um dos problemas mais difíceis de serem solucionados, pois é necessário identificar as mesmas entidades em imagens distantes no tempo e de posições diferentes. As hipóteses utilizadas para encontrar a melhor correspondência crescem exponencialmente com o tempo.

Outro agravante é que os ambientes são dinâmicos, e os objetos mudam de lugar com o tempo, com uma velocidade variada. Não existe um algoritmo que incorpore de maneira significativa a dinâmica do ambiente. O que é feito é adotar a hipótese de ambiente estático, e trabalhar numa curta janela de tempo.

E por último, o problema da exploração, do robô decidir o melhor caminho (segundo algum critério como o ganho de informação, o tempo e a energia gastos para adquiri-la) a seguir para adquirir mais informações para compor o mapa. Em ambientes previamente modelados, esta tarefa possui solução otimizada conhecida, mas a tarefa de exploração implica em trabalhar-se com modelos parciais e incompletos do ambiente.

Formas de abordar este problema são: Filtro de Kalman e uma extensão deste paradigma, o algoritmo de Lu/Milios, os algoritmos de Maximização da Expectativa (EM – Expectation Maximization), uma abordagem híbrida como a máxima verossimilhança incremental (Maximum Likelihood), as Grades de Ocupação, Mapas compostos por objetos (Object Maps), e Dogma, para ambientes dinâmicos.

3.2 Grades de Ocupação

Um robô pode ser capaz de, através da informação obtida de seus sensores, criar um modelo espacial robusto e utilizá-lo para tarefas de planejamento e controle. Assim, o robô poderia interagir de maneira coerente com o ambiente, tanto interpretando os dados disponibilizados pelos sensores com o intuito de obter conclusões apropriadas a cerca do mundo real para decisões a curto prazo, quanto adquirindo e manipulando um modelo complexo de mundo para planejamento e tomadas de decisão a longo prazo.

Tradicionalmente, tem sido dada ênfase a procedimentos de baixo nível de sensoriamento que extraem características geométricas a partir dos dados do sensor, e módulos de interpretação de alto nível que utilizam modelos geométricos primitivos e hipóteses heurísticas sobre o meio ambiente para limitar o processo de interpretação do sensor como base para planejamento de tarefas para o robô. Isto

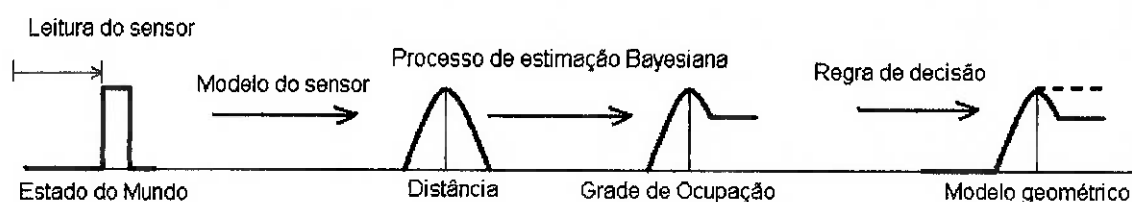
caracteriza o paradigma geométrico que leva a modelos do mundo esparsos e frágeis [7].

Entre os anos de 1973 e 1984, a abordagem da navegação robótica no Mobile Robot Lab's era baseada em visão estéreo com múltiplas linhas de base que rastreava características ao redor para evitar e mapear os obstáculos no ambiente. Este trabalho foi conduzido por Hans Moravec, e entre 1980 e 1984, com o auxílio de Alberto Elfes, na Carnegie Mellon University, mostrou que certos erros cumulativos durante a navegação eram inerentes a abordagem utilizada, que produzia uma representação do ambiente muito suscetível a falhas. Esta abordagem é definida como o paradigma geométrico. Foi então proposto o que ficou conhecido inicialmente como Grades de Ocupação (*Occupancy Grids*, depois sendo denominado *Certainty Grids* e *Evidence Grids*). De um método *ad hoc* passaram posteriormente para uma fundamentação de todo o desenvolvimento na Teoria de Probabilidade.

Uma possível abordagem para o problema da percepção robótica e modelagem do mundo é a utilização de uma representação probabilística da informação espacial obtida através do contínuo sensoriamento do ambiente. Uma ferramenta que atende a estes requisitos é a Grade de Ocupação (*Occupancy Grid*). Uma Grade de Ocupação é um campo multi-dimensional randômico, uma segmentação do espaço físico, que mantém estimativas estocásticas do estado de ocupação de cada célula que compõe o espaço. As estimativas da ocupação das células são obtidas pela interpretação dos dados do sensor (onde o que deseja-se obter é a distância dos objetos ao robô) usando modelos probabilísticos que incorporam a incerteza na informação espacial. Procedimentos de estimativa baseados na regra de Bayes permitem a construção e constante atualização da Grade de Ocupação usando leituras de múltiplos pontos de vista, ou de diversos sensores.

A Grade de Ocupação pode ser definida como $O(C_i) = P[s(C_i)=OCC](C_i)$, ou seja, um processo estocástico de estados discretos definido sobre um espaço segmentado. C_i representa as células nas quais o espaço é dividido e $s(C_i)$ é a variável aleatória binária que assume o valor OCC (ocupado) ou EMP (livre).

Uma aplicação na qual utilizam-se mapas do ambiente é a navegação autônoma. Como todo sensor possui incertezas inerentes, faz-se uma transformação do espaço dos parâmetros (o ambiente) para o espaço de observação através de um modelo probabilístico que busca incorporar as incertezas na leitura do sensor e de todo o processo envolvido na obtenção das distâncias, mais especificamente, uma função densidade de probabilidade condicional (modelo do sensor). Do espaço de observação, com as leituras obtidas pelo sensor, é possível tentar reproduzir o ambiente através de um processo de estimação, o cálculo de uma probabilidade *a posteriori*. Este processo de inferência pode ser feito por meio de uma estimação Bayesiana. Assim, é criada a Grade de Ocupação. A partir dela, é possível reproduzir um modelo geométrico do mundo, com uma regra de decisão, com o máximo *a posteriori* (MAP). A Figura 1 mostra um diagrama do processo:



Esquema do Sistema de Construção do Mapa

3.3 Sensoriamento

Um robô adquire informação sobre o seu ambiente através dos sensores que possui. Cada tipo de sensor apresenta características próprias e se adequam a aplicações específicas. Por exemplo, o sonar tem baixa resolução angular e erros devido a múltiplas reflexões ou reflexões especulares para fora do campo do sensor, enquanto que a visão estéreo detecta e localiza fronteiras e marcas, mas não encontra o interior de superfícies marcadas [8].

Apesar de as Grades de Ocupação serem apropriadas para integrar a informação de distância vindas de vários tipos de sensores (e de diversas leituras do mesmo), este trabalho busca utilizar a visão estéreo omnidirecional como único sensor e um

processamento computacional apropriado para garantir as informações necessárias para futuramente utilizá-las numa navegação autônoma em tempo real.

3.4 Modelos do Sensor – Função densidade de probabilidade

Cada tipo de sensor é modelado de maneira distinta quanto à forma de interpretação espacial do mundo, modelo este que deve incorporar as incertezas inerentes ao mesmo. Este modelo é formado por uma função densidade de probabilidade e uma regra de como utilizá-la para efetuar o cálculo da Grade de Ocupação. Esta regra procura representar a forma como o sensor obtém informações de distância no ambiente em que está inserido. A fórmula de cálculo de probabilidade, o modelo do sensor propriamente dito, incorpora o tipo de erro cometido nas leituras que é inerente ao tipo de sensor utilizado. Estes modelos, as funções densidade de probabilidade, por serem de mesma natureza, são facilmente integráveis para complementar a informação espacial vinda de diversos sensores.

O modelo do sensor consiste de uma função densidade de probabilidade que melhor caracterize o sistema, ou seja, uma probabilidade condicionada de como são as leituras do sensor dado uma configuração de mundo ou ambiente, $p(L|M)$ – onde “L” indica uma leitura do sensor e “M”, uma configuração de mundo. Para um sensor ideal tem-se: $p(l|z) = \delta(l-z)$, representação para o caso unidimensional de grade, que indica que a probabilidade de, dado z , ou seja, a distância real de um objeto, obter-se a leitura l do sensor, no caso de um sistema de visão, da distância vinda da triangulação do par de pontos correlatos das imagens que compõe o estéreo. Mas como todo sensor possui incertezas em suas leituras, é possível modelar este fenômeno através de um ruído gaussiano, uma distribuição normal com média dada pela posição estimada:

$$p(l|z, \theta) = \frac{1}{2\pi\sigma_l\sigma_\theta} \exp\left[-\frac{1}{2}\left(\frac{(l-z)^2}{\sigma_l^2} + \frac{\theta^2}{\sigma_\theta^2}\right)\right], \quad \text{para o caso de uma grade}$$

bidimensional com média dada pela distância encontrada pelo sensor e variância de σ_l^2 e σ_θ^2 para as variáveis l e θ , evidenciando tanto uma incerteza quanto à direção

do objeto localizado quando da distância estimada. Estas funções são utilizadas para o cálculo da grade segundo as regras criadas para cada tipo de sensor.

Como exemplo de regra, pode-se citar o modelo do sonar. Este sensor é um dos mais utilizados com a abordagem de Grades de Ocupação, e foi justamente com ele que toda a teoria foi desenvolvida. O sonar localiza objetos no ambiente emitindo um som e captando a reflexão desta onda no objeto. Pelo tempo de percurso, estima-se a distância. Assim, o sonar obtém informação apenas em uma direção, e por isso, são utilizados cinturões com diversos deles para obter, a cada leitura, informação em todas as direções ao redor do robô que mapeia o ambiente. Outro ponto é que, a partir do momento em que um objeto é localizado em determinada direção, os objetos que estão por detrás dele, na direção do sensor, não são detectados. Apenas através de contínuo sensoriamento, por diversos ângulos de visão, é que o ambiente consegue ser devidamente mapeado. Então, a regra para o sonar seria, dado que sua visibilidade é em apenas um direção, ele identifica como vazio o espaço entre si e o objeto, determina que na distância calculada há um objeto e que a distâncias maiores a ocupação é indefinida. Num modelo ideal de sensor, seria atribuído o valor zero para as células anteriores ao objeto localizado, um para a célula na qual está o objeto e meio para as células posteriores.

Algumas vezes são utilizados sensores de visão estéreo para completar a informação espacial obtida por meio dos sonares, e também como única forma de sensoriamento. A visão estéreo é normalmente adquirida com um par de câmeras presentes no robô. As câmeras também possuem um campo visual restrito, apesar de que mais amplo que o de um único sonar. Mas a visão acaba sendo utilizada apenas para obter informação espacial sobre o ambiente à frente do robô, conforme ele se desloca. A regra de interpretação das leituras deste sensor é semelhante à do sonar. O ambiente em três dimensões é projetado num plano, no da imagem, sendo que apenas os objetos mais à frente são representados. Com o par de imagens obtido a cada leitura, faz uma correspondência entre as características destacadas numa imagem e na outra, e determina-se, por triangulação, as distâncias dos objetos. Tanto a triangulação quanto a correspondência são fontes de erros inerentes ao processo de estéreo e devem ser modelados com o auxílio de uma função densidade de probabilidade

adequada. Para um sensor ideal, as células presentes no campo visual da câmera, anteriores à detecção dos obstáculos, seriam consideradas vazias e teriam probabilidade zero, as células presentes na linha que forma dos objetos localizados teriam probabilidade um, indicando estarem ocupadas, e, ainda dentro do campo visual, as células por detrás desta linha teriam probabilidade meio, evidenciando desconhecimento completo sobre a situação de ocupação das mesmas.

Neste trabalho busca-se modelar a visão estéreo omnidirecional. Para encontrar uma regra que melhor caracterize este tipo de sensor, cabe tentar descrever a forma como ele enxerga o mundo. Assim, com uma leitura deste sensor é possível detectar os contornos ou bordas das superfícies ou objetos no ambiente, com um campo visual de 360° ao redor da câmera, sendo de grande utilidade para evitar colisões com objetos móveis em tarefas de navegação. Apesar de não conseguir, como qualquer outra forma de visão estéreo, localizar o interior dos objetos e superfícies, apenas suas bordas ou contornos, este sensor, devido à sua configuração, não fica limitado a identificar apenas os objetos, em dada direção, mais próximos ao sensor, ou seja, ele consegue localizar objetos que estão atrás de outros, dependendo da altura e distância daqueles que estão à frente. Uma regra poderia então possuir critérios diferentes para o primeiro ponto localizado em dada direção e os que estão além deste. Para o primeiro, o critério seria o mesmo que para a visão estéreo convencional, só que com um campo visual de 360° . Uma grande área ao redor do robô seria considerada como vazio, até o encontro da linha que une os primeiros obstáculos em cada direção. A partir daí, as linhas subsequentes funcionariam, pensando no cálculo segundo o modelo de probabilidades, como na integração de informações de diferentes sensores. A regra seguinte seria a de que não se tem conhecimento algum sobre o estado de ocupação das células anteriores e nem posteriores a nova linha analisada, e através da integração das diversas linhas, o cálculo iria convergir para a forma como este sensor estéreo omnidirecional enxerga o mundo.

Esta regra, ou conjunto de regras, deve modelar as incertezas em alguns processos principais na obtenção das informações de distância.

O processo de extração de características pode ser influenciado basicamente pela variação da luminosidade do ambiente, mas o algoritmo escolhido detecta inúmeros pontos, evidenciando a presença de todos os objetos em cena, através das suas bordas. O fato de que um ou outro ponto não foi destacado, sinal de erro no processo, é irrelevante dada a quantidade de pontos encontrados. A correspondência é um processo sujeito a muitos erros, de correlações erradas ou de não encontradas. Em parte isto ocorre devido ao algoritmo de procura que não possui a robustez necessária para a tarefa. A triangulação também está sujeita a erros que devem ser incorporados. Alguns pontos a serem destacados são:

1. Limitações das regiões visíveis para o sensor, dependente da direção movimentação, devido à utilização da geometria epipolar, que para cada ponto fornece a equação de uma curva onde o ponto corresponde se localiza. Estas curvas são projeções no plano da imagem da intersecção do plano epipolar formado pelo ponto no espaço e pelo vetor deslocamento do robô com a superfície do espelho na aquisição da segunda imagem. Tais projeções podem ser impróprias, quando próximas da linha de deslocamento;
2. Utilizando a noção de arco capaz, pode-se estabelecer um critério para o erro cometido segundo o ângulo de visão, já que a informação de distância depende deste ângulo formado pelo deslocamento entre uma imagem e outra;
3. A busca do ponto correspondente com janelas sobre as curvas epipolares, que podem estar sujeitas a pequenos desvios em sua posição. O espaço percorrido é sempre uma faixa, e não apenas a linha;
4. Existe também uma imprecisão que deve muito ser levado em conta em tarefas de navegação e mapeamento autônomo, qual o deslocamento real do robô, que pode possuir erros sistemáticos de quantidade e direção;
5. A influência da resolução do espelho ao longo de sua superfície, em função da precisão de sua usinagem, fato que pode ser identificado através de testes reais com as imagens adquiridas;

6. E a distância entre a aquisição de uma imagem e outra, procurando encontrar como o erro varia em função deste deslocamento;

3.5 Inferências sobre o Estado do Mundo – Processo de Estimação Bayesiano

A partir das leituras do sensor, que tentam captar informações sobre o mundo, é possível utilizar um modelo para tentar reconstruir o mundo a partir destas informações.

Como é possível apenas conhecer o ambiente ao redor do robô por meio do sensor, temos agora apenas uma função densidade de probabilidade da qual temos de extrair a probabilidade de que cada célula da grade de ocupação (que segmenta o espaço) esteja ocupada ou livre.

Para isto, é necessário escolher um processo de estimação. O processo escolhido foi o Bayesiano. É uma função densidade de probabilidade, que calcula o valor *a posteriori*. Esta probabilidade é obtida através do teorema de Bayes:

$$p(s(C_i) = OCC | r) = \frac{p(r | s(C_i) = OCC) \cdot P(s(C_i) = OCC)}{\sum_{s(C_i)} p(r | s(C_i)) \cdot P[s(C_i)]}.$$

O termo $P(s(C_i) = OCC)$ representa a probabilidade das configurações *a priori*, é o seu conhecimento do mundo antes de uma nova leitura do sensor. Como a intenção é trabalhar com ambientes não estruturados dos quais não se tem conhecimento algum *a priori*, estabelece-se que $P(s(C_i) = OCC) = P(s(C_i) = EMP) = 0,5$, o que significa utilizar a probabilidade de máxima entropia. Levando em consideração o fato de que as células da grade são variáveis aleatórias independentes, é possível escrever que pelo teorema de Kolmogoroff:

$$p[r | s(C_i) = OCC] = \sum_{\{G_{s(C_i)}\}} p[r | s(C_i) = OCC, G_{s(C_i)}] P[G_{s(C_i)} | s(C_i) = OCC], \quad \text{onde}$$

$$P[G_{s(C_i)} | s(C_i) = OCC] = \prod_{\forall j, j \neq i} P[s(C_j) = s_j] = \frac{1}{2^{N-1}}, \text{ sendo } N \text{ a dimensão da grade e}$$

$G_{s(C_i)} = (s(C_1) = s_1, \dots, s(C_{i-1}) = s_{i-1}, s(C_{i+1}) = s_{i+1}, \dots, s(C_n) = s_n)$ são os possíveis estados da grade, desconsiderando a célula C_i .

A probabilidade $p[r | s(C_i) = OCC, G_{s(C_i)}]$ é retirada do modelo de interpretação espacial. Para o caso do sensor ideal, tem-se que [9]:

$$P[r_j | s(C_i) = OCC] = \begin{cases} \frac{1}{2^{n-1}} \frac{2^{n-1}}{2^j} = \frac{1}{2^j} \text{ para } j < i \\ \frac{1}{2^{n-1}} \frac{2^{n-1}}{2^{j-1}} = \frac{1}{2^{j-1}} \text{ para } j = i \\ \frac{1}{2^{n-1}} \cdot 0 = 0 \text{ para } j > i \end{cases}$$

3.6 Regra de Decisão – (MAP – máximo *a posteriori*)

Para transformar a Grade de Ocupação num modelo geométrico do ambiente, é necessário escolher uma regra para determinar a melhor estimativa dentre a distribuição obtida. Uma opção é o máximo *a posteriori* (MAP), que considera o ponto de maior probabilidade como sendo o ponto a ser escolhido. Pode-se escolher a média, a mediana ou a moda. Para distribuições gaussianas, estas três regras caem no mesmo ponto.

3.7 Atualização dos Mapas

Elaborar um modelo ou mapa do ambiente através das informações da leitura dos sensores é um problema que exige a aplicação de restrições, devido às limitações intrínsecas do sensoriamento. A abordagem deste problema segundo as Grades de Ocupação é a utilização de sensoriamento adicional para resolver as ambigüidades e incertezas. É uma composição da informação obtida por diversos sensores ou pelo sensoriamento de pontos de vista diferentes.

A atualização dos mapas é feita segundo uma formulação de atualização sequencial do teorema de Bayes e permite a convergência dos modelos do sensor com ruído para o modelo ideal.

$$p(OCC | \{R_{t+1}\}) = \frac{p(\{R_{t+1}\} | OCC).P(OCC | \{R_t\})}{p(\{R_{t+1}\} | OCC).P(OCC | \{R_t\}) + p(\{R_{t+1}\} | EMP).P(EMP | \{R_t\})}$$

Como discutido anteriormente, um dos problemas de se fazer a atualização, está em conseguir compor as informações espaciais corretamente, o que está vinculado com o posicionamento do robô ao construir o mapa, e as incertezas inerentes ao movimento do mesmo, na trajetória feita para adquirir as informações.

Para facilitar a composição dos mapas de regiões que se sobrepõem, é necessário vincular o deslocamento do robô entre a aquisição de uma imagem e outra do primeiro par estéreo, e de um segundo deslocamento, onde a segunda e terceira imagens compõem o novo par. O robô deve deslocar-se de uma quantidade tal que permita a ele ocupar a mesma posição nas células. Por exemplo, do centro de uma célula para o centro de sua vizinha. Assim, é possível realizar os cálculos pois as células coincidem umas com as outras.

O robô utilizado para o mapeamento neste trabalho não muda a orientação de seu sensor conforme se locomove. Ou seja, não há a necessidade de utilização de uma matriz de rotação nas transformações de coordenadas, pois o sistema mecânico é desvinculado da parte em que está montado o sensor. Isso evidencia que durante todo o percurso do robô no mapeamento, há uma mesma direção que não produz informações devido a utilização da geometria epipolar para a realização do estéreo.

RESULTADOS

4.1 Ambiente de desenvolvimento Khoros

O Khoros possui uma interface gráfica, semelhante ao Simulink do MatLab, que permite a montagem de um fluxo de processamento, discretizando o processo.

Implementado no Khoros, o sistema de visão foi discretizado segundo a Figura, nas seguintes funções para este trabalho:

- Extração de características de uma imagem (SUSAN*), que tem como entrada uma imagem omnidirecional no formato *pgm* e como saída a imagem de entrada com as características seleccionadas em vermelho (formato *pnm*) e uma lista em arquivo texto das mesmas (pares de pixels);
- Filtro (FILTER*), que tem como entrada uma imagem omnidirecional (*pgm*) e uma lista de pares de pixels – características encontradas pelo SUSAN – e como saída a imagem de entrada com as características seleccionadas em vermelho (*pnm*) e uma lista em arquivo texto das mesmas;
- Traçar as linhas epipolares na segunda imagem que compõe o par estéreo (EPIPOLAR**), que tem como entrada esta imagem e a lista de pares de pixels (pontos) da unidade anterior, e tem como saída a imagem de entrada com as linhas epipolares traçadas em verde e uma lista com as matrizes que caracterizam estas linhas;
- Correspondência de pontos no par estéreo (CORRELATION**), que tem como entrada a segunda imagem deste par, a lista de pontos vinda do filtro e

a lista das curvas vinda da Epipolar, e como saída a mesma imagem com os pontos correspondentes marcados em verde e a lista dos mesmos;

- Cálculo da distância do ponto ao robô (DISTANCE***), que possui como entrada a lista de pontos vindas da primeira unidade e dos pontos correspondentes, e como saída cria o mapa de distâncias;

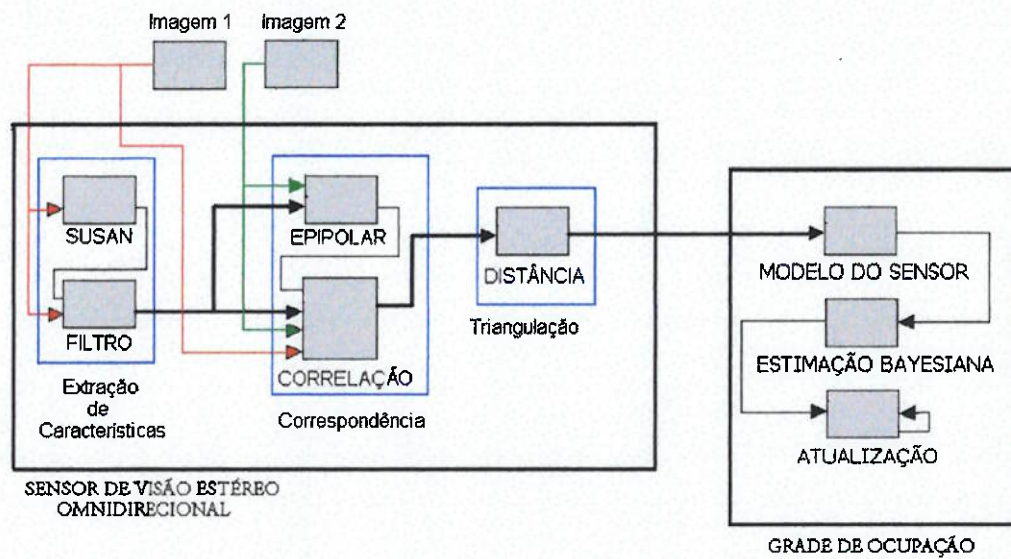
* Compõe o grupo de Extração de Características.

**Compõe o grupo de Correspondência.

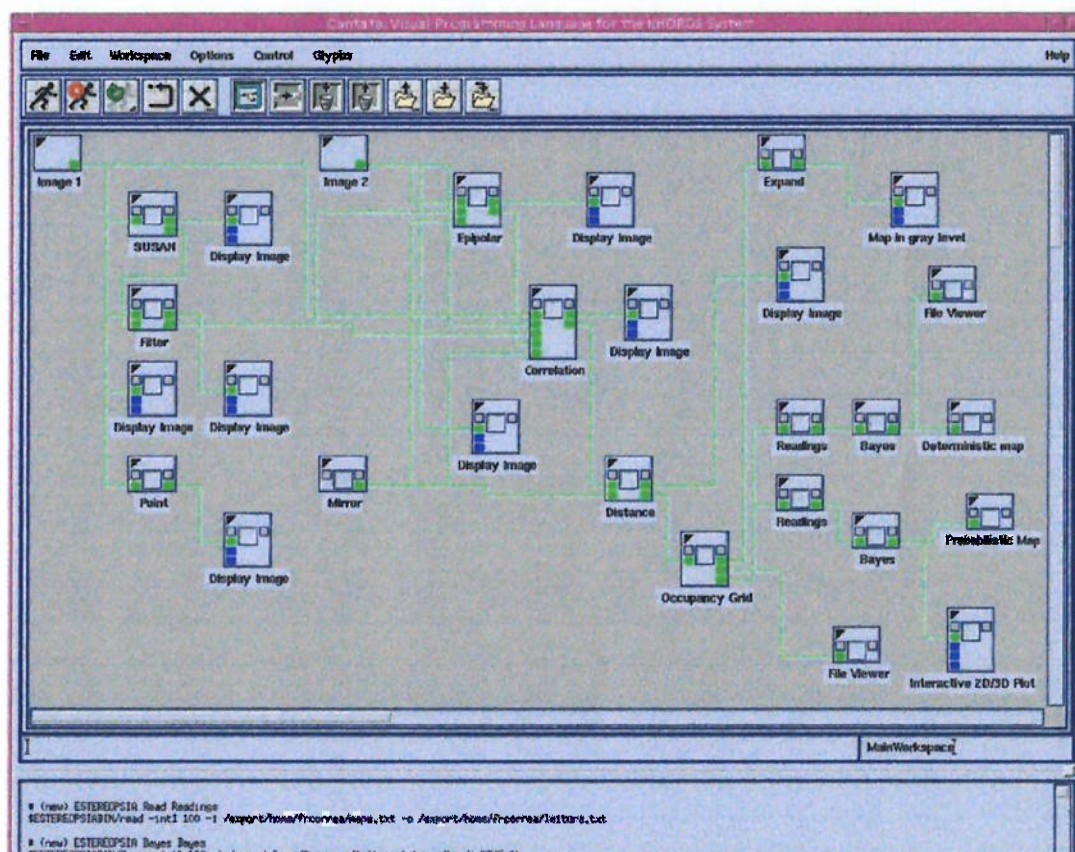
***Compõe o grupo de Triangulação.

Já quanto à Grade de Ocupação, o processo foi discretizado do seguinte modo:

- Cálculo da Grade de Ocupação (OCCUPANCY GRID), onde é feita a transposição dos vetores distância obtidos do sensor estéreo omnidirecional para uma estrutura em células, e cada célula com um ponto marcado em seu interior, produz uma distribuição de Gauss com média nela e um desvio padrão;
- Leituras (READINGS), que faz o acúmulo num arquivo texto das matrizes com os dados das células;
- Atualização (BAYES), que utiliza a informação *a priori* dos mapas compostos previamente, e os atualiza segundo a regra de Bayes;



Esquematização do sistema de visão estéreo



Ambiente de desenvolvimento das rotinas

4.2 Sistema de Visão

O sistema de visão omnidirecional vem sendo estudado para uma implementação correta de [20] e para adquirir conhecimento suficiente com o intuito de melhorar o sistema de mapeamento estático. Está sendo reproduzido o trabalho desenvolvido em [20], previamente implementado em linguagem C, para o ambiente Khoros, programa de processamento de imagens. Aperfeiçoamentos estão sendo realizados para generalizar o sistema, antes restrito ao deslocamento em apenas uma direção, e para aquisição de mapas de distância mais precisos.

Pode-se dividir o trabalho realizado em [20] nas seguintes etapas: Modelagem do Sistema de Visão, Pré-Processamento da Imagem, Seleção de Características, Curvas Epipolares, Correspondência, Cálculo das Distâncias, Mapa de Distância.

O modelo do sistema de visão pode ser descrito como:

- Espelho Hiperbólico (sistema de coordenadas do espelho):

$$\frac{(z + \sqrt{a^2 + b^2})^2}{a^2} - \frac{x^2 + y^2}{b^2} = 1 \Rightarrow z = \sqrt{a^2(1 + \frac{r^2}{b^2})} - \sqrt{a^2 + b^2}$$

,onde os parâmetros a e b são, respectivamente 39,292 e 19,646.

- Ângulo Espacial de Visão (valor máximo):

$$\alpha = \frac{\pi}{2} + \arctg\left(\frac{h - 2e}{r_{top}}\right)$$

,onde $\alpha = 121,3$.

4.3 Grade de Ocupação

O algoritmo responsável pelo cálculo, construção e atualização da Grade está sendo desenvolvido, levando-se em conta os problemas computacionais em se armazenar informação para uma quantidade enorme de células, nas quais o espaço está dividido. No momento, a Grade possui uma divisão de 100x100 células quadradas,

representando um espaço de 5x5 m. O resolução das superfícies de probabilidade estão sendo estudadas para melhor visualização da Grade.

O modelo do sensor adotado foi o de ruído gaussiano, com desvio padrão de 1,2.

O sensor, ao localizar um ponto, identifica como espaço vazio todos os pontos num raio que parte do robô até o ponto localizado, e do ponto até o final do mapa também como vazio. Se por ventura um outro ponto estiver marcado neste caminho livre, o algoritmo o mantém assim. Desta forma, o interior dos objetos fica evidenciando um espaço vazio, cercado por uma barreira ocupada.

4.4 Ambiente virtual

O ambiente virtual foi criado com o auxílio do programa Povray com o intuito de validar os algoritmos de modelagem da visão estéreo. É um ambiente no qual pode-se controlar o efeito de reflexão e difusão, da luminosidade e da presença dos objetos no mesmo. Este ambiente possui uma caixa e uma porta. A seguir será colocado um exemplo ilustrativa da procura de um ponto específico da primeira imagem do par estéreo na segunda.

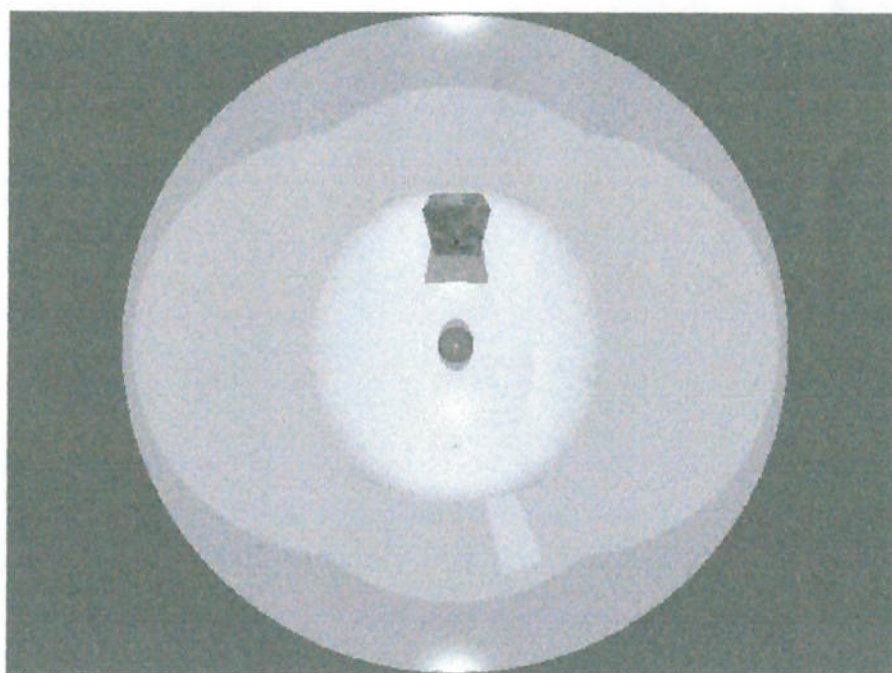
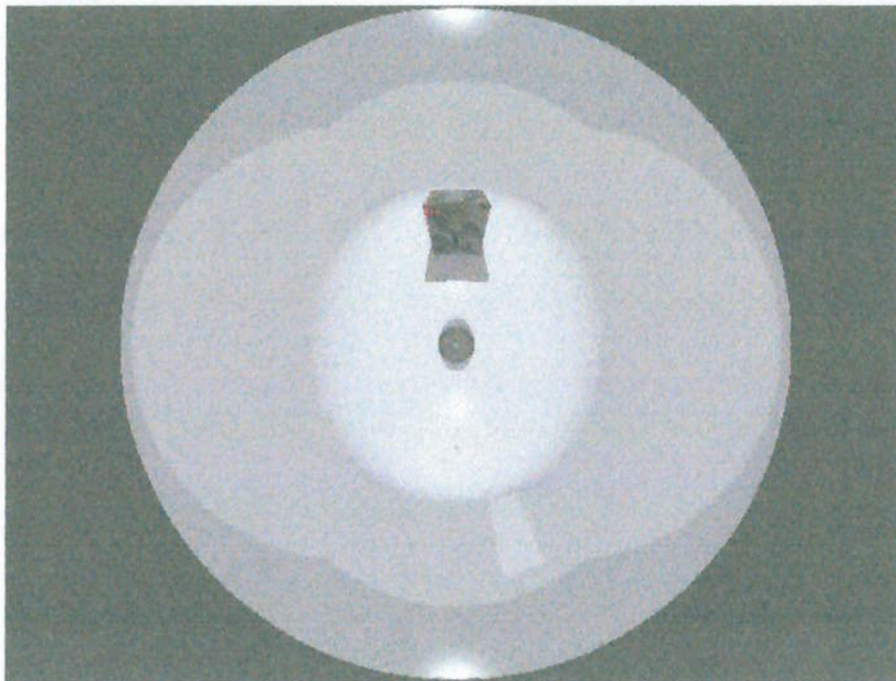
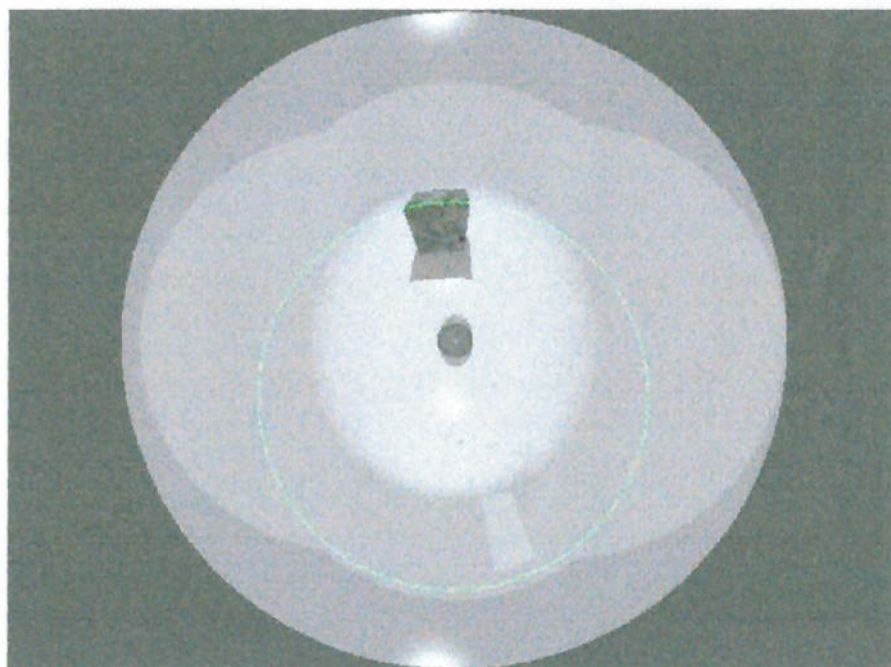


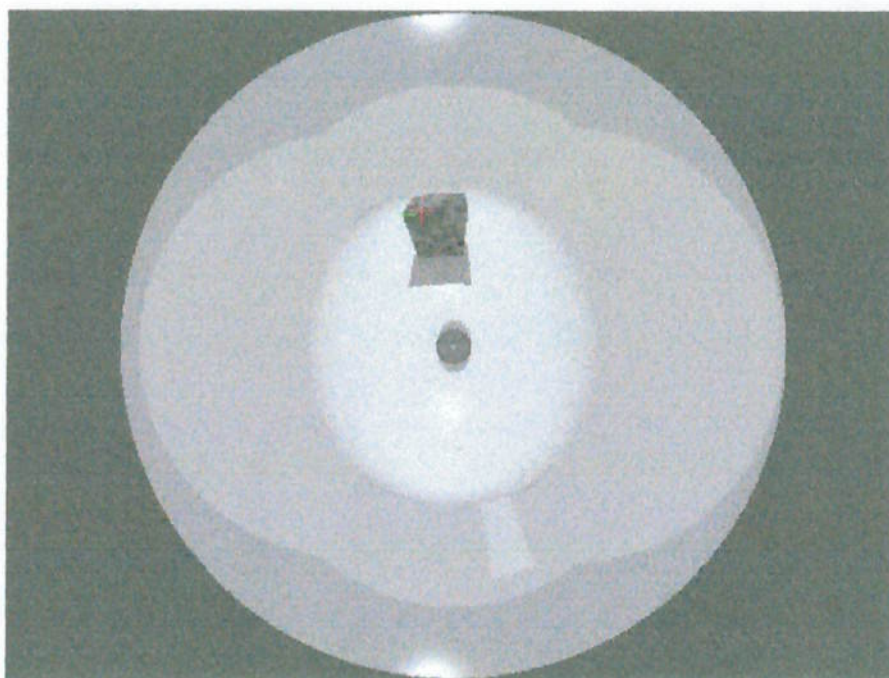
Imagem do ambiente virtual criado para os testes



Um ponto (cruz vermelha) na borda do cubo localizado pelo algoritmo SUSAN



Curva epipolar na segunda imagem passando pelo ponto destacado na primeira



Pontos correlato: em vermelho, a posição do ponto na primeira imagem, e em verde, na segunda.

4.5 Ambiente Real

Após a validação do algoritmo com os teste feitos no ambiente virtual, foram realizados testes no Laboratório de Percepção Avançada.

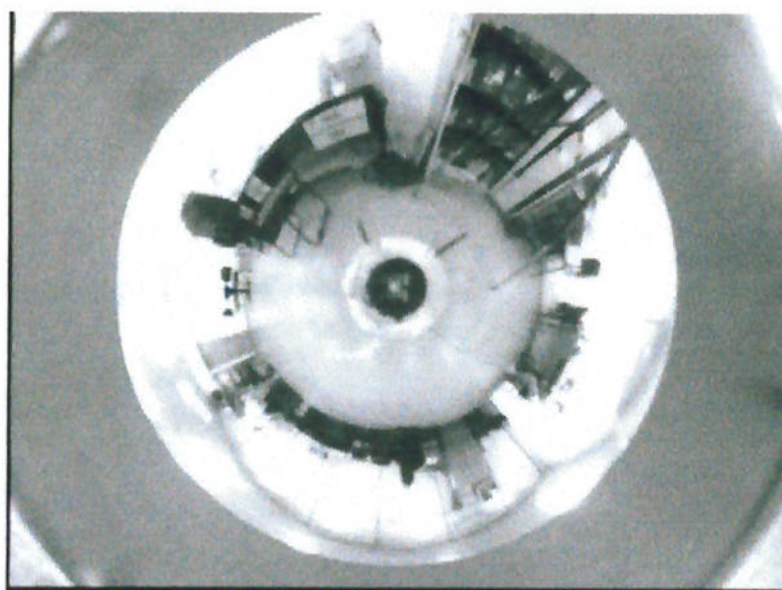
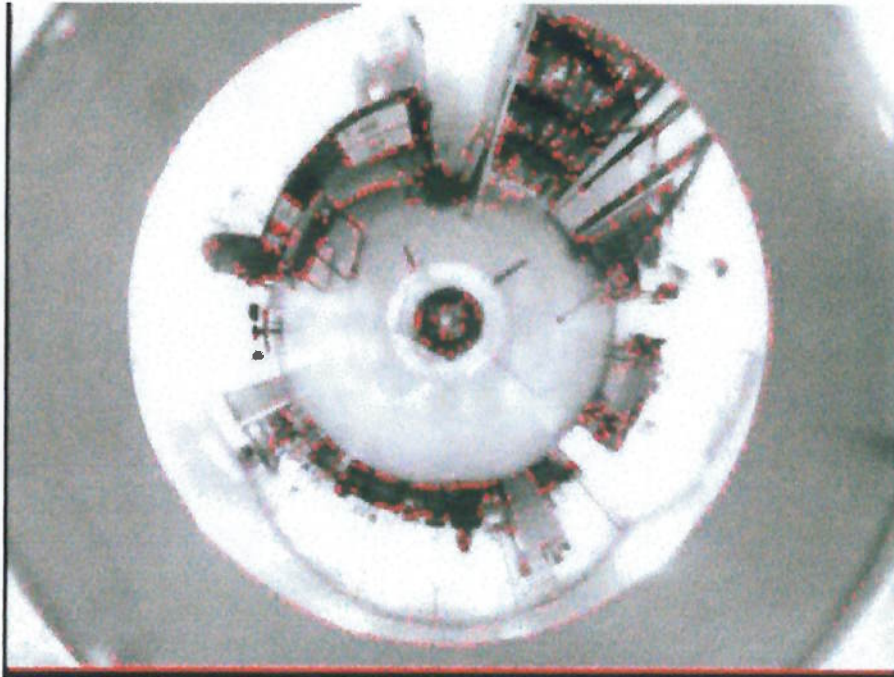
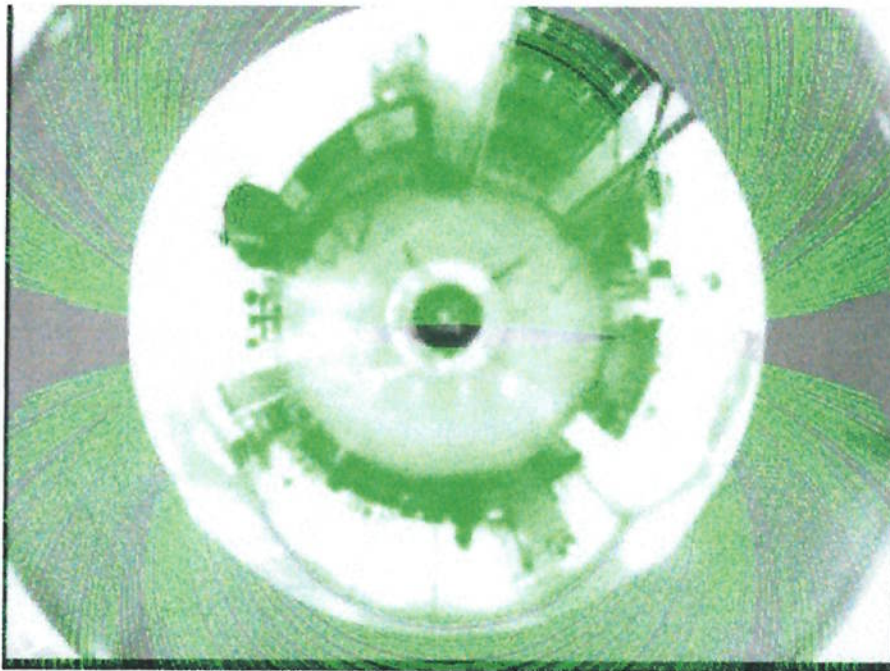


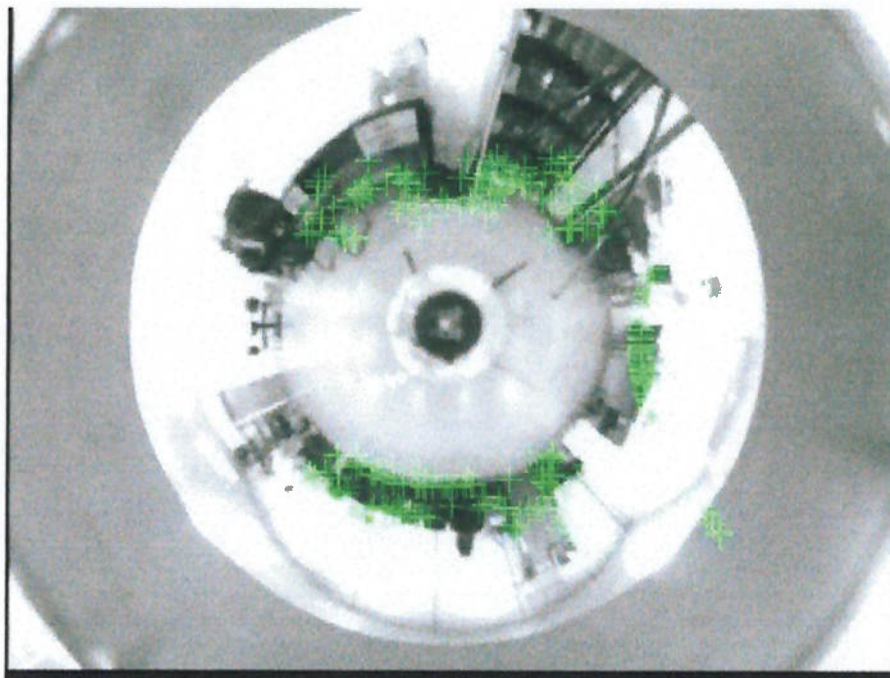
Imagem do ambiente real utilizado para teste



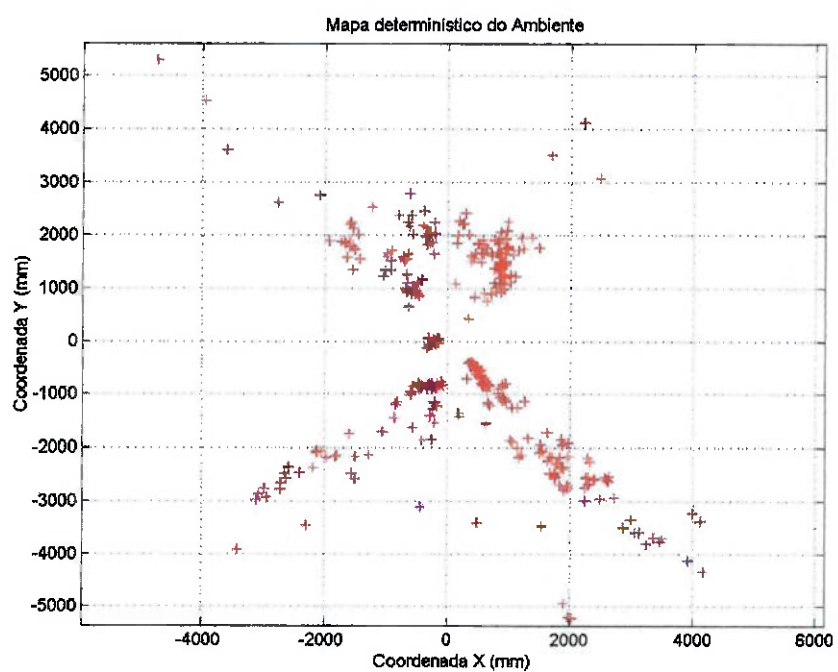
Pontos encontrados segundo o algoritmo do SUSAN



Linhas epipolares correspondentes aos pontos encontrados na primeira imagem

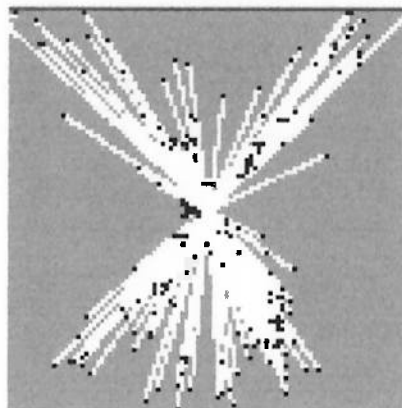


Pontos correlacionados encontrados na segunda imagem procurando sobre as epipolares



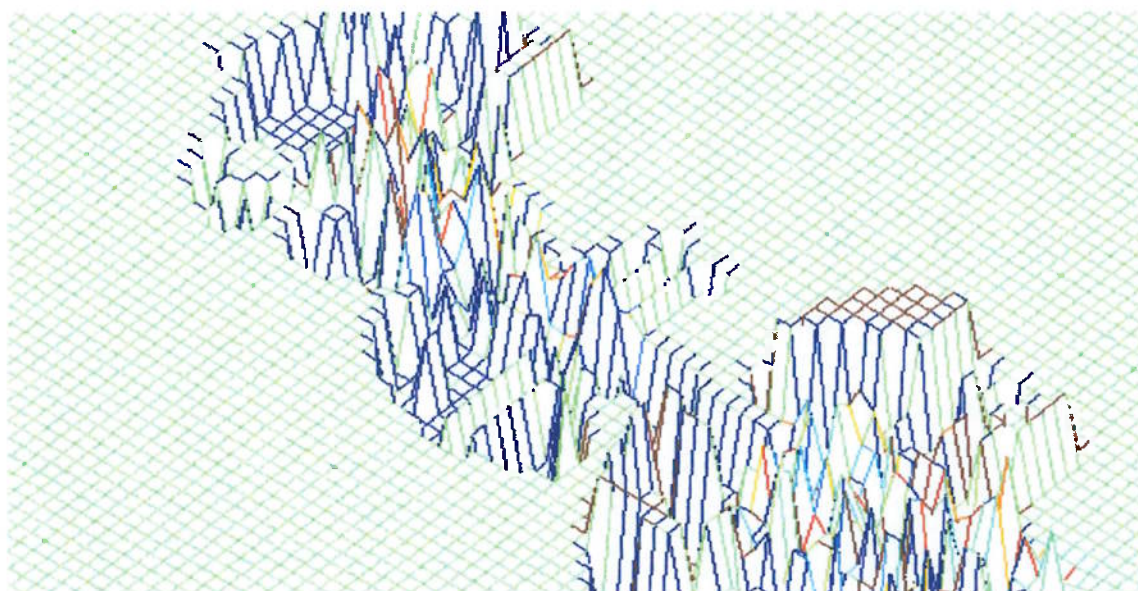
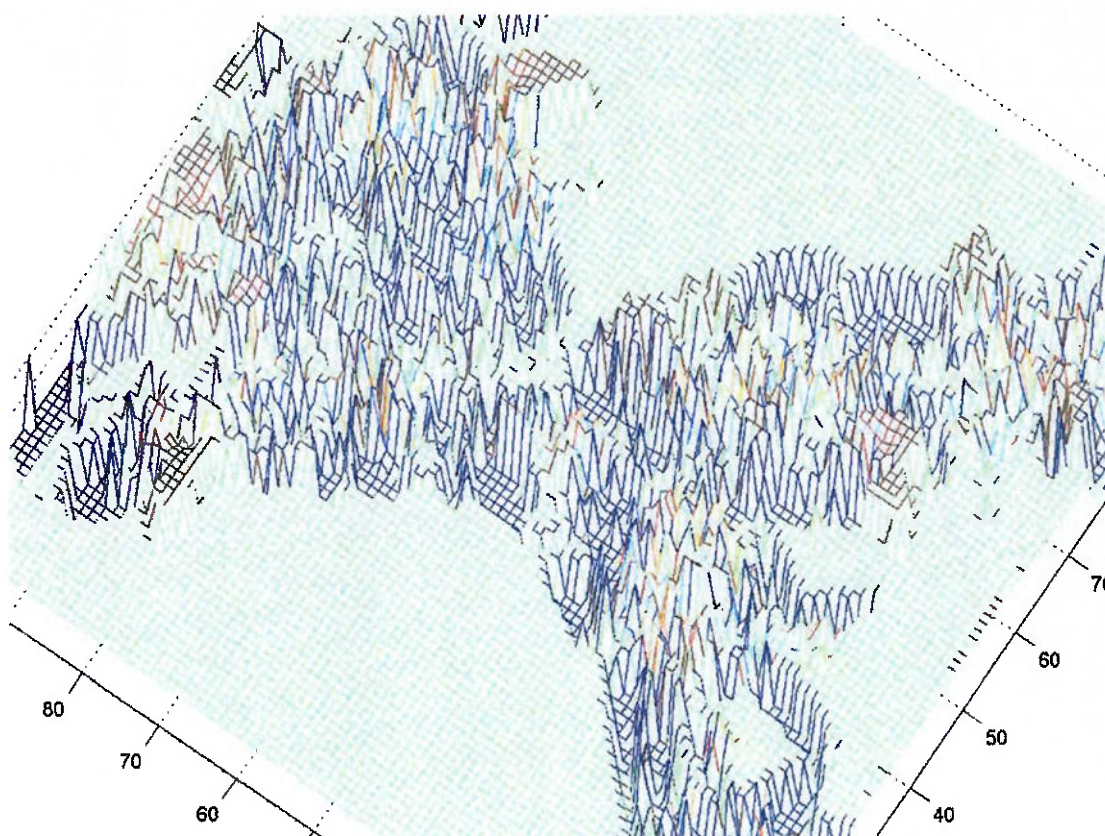
Mapa das distâncias dos objetos à câmera

Uma forma de representar a regra de como o sensor de visão estéreo omnidirecional enxerga o ambiente é utilizar um fundo cinza que representam as áreas de onde não têm-se informação alguma. Nesta representação o robô está no centro. Dele partem raios nas direções dos pontos localizados, marcados em preto, numa primeira leitura do sensor. Entre o robô e os pontos marcados, a área é marcada em branco, mostrando que está desocupada. Note que pontos atrás de outros localizados também são mostrados.

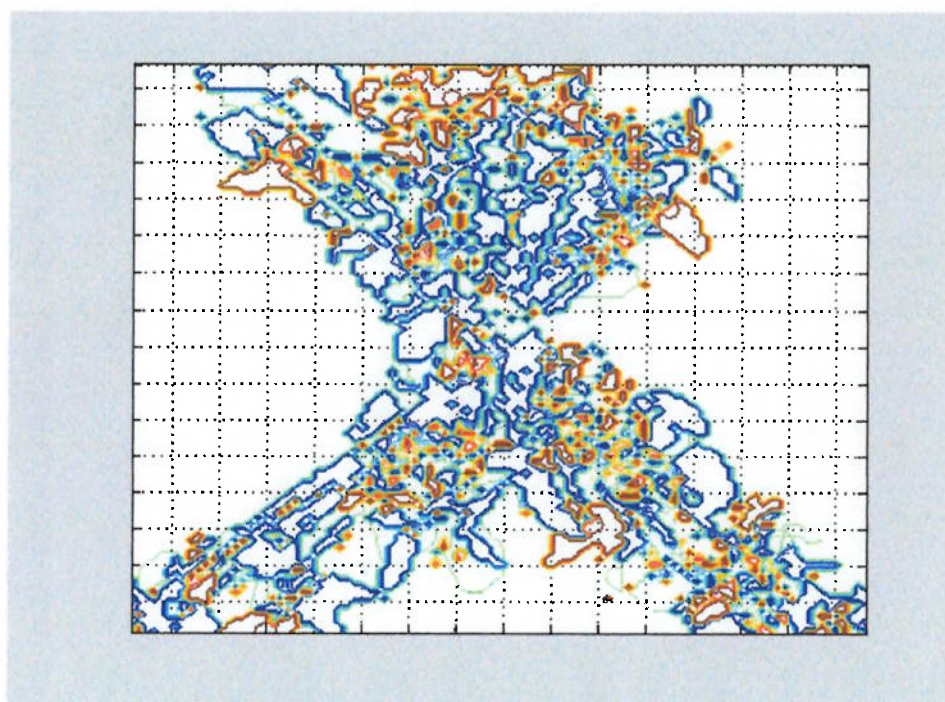


Visão do sensor

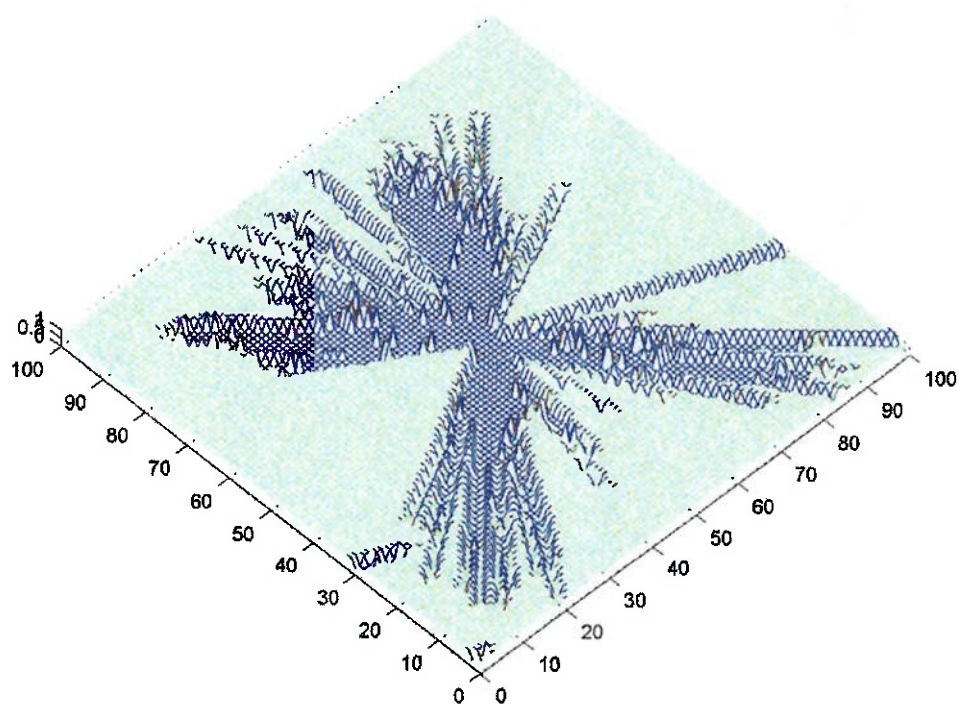
A partir da leitura do sensor, é necessário usar estes vetores para preencher as células que compõem o ambiente. O fato de um ou mais pontos estarem na mesma célula não são levados em conta; isto depende do nível de resolução da grade adotado, que tem a ver com a capacidade computacional de cálculo e armazenamento.



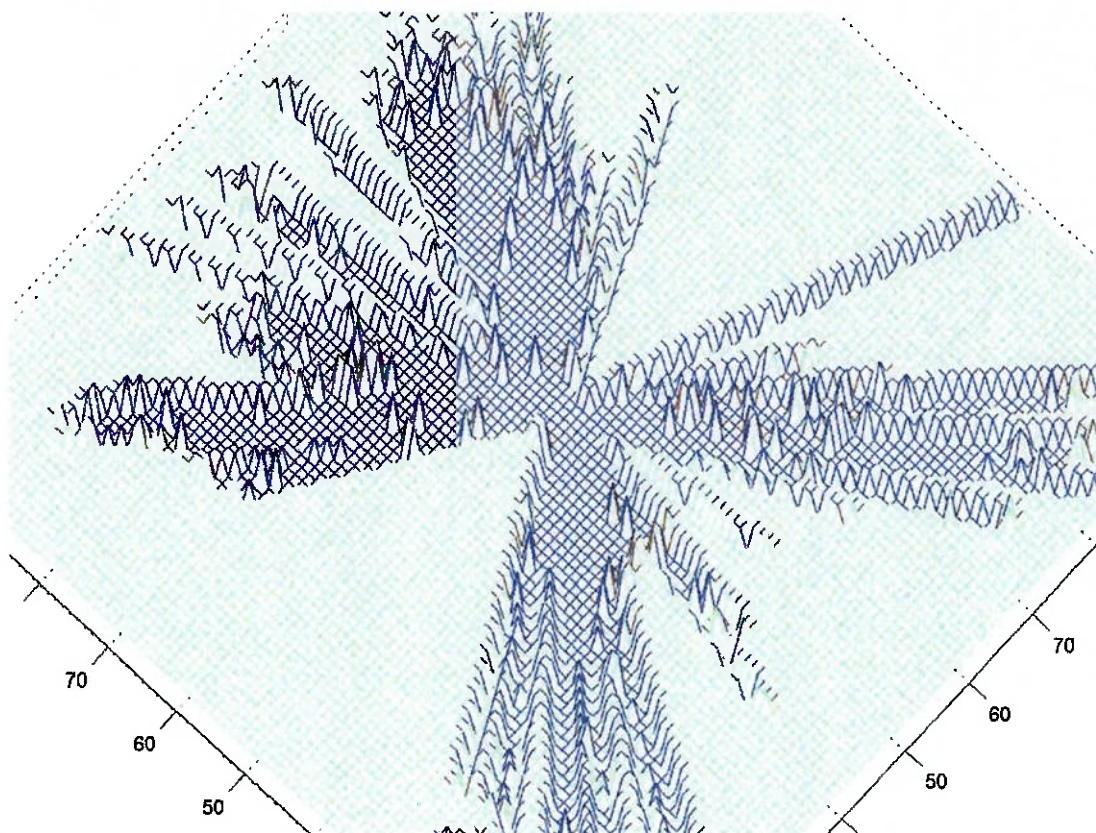
Representações da Grade de Ocupação



Representação topográfica da Grade de Ocupação



Estrutura de leitura do sensor



CONCLUSÃO

Foi proposta uma forma de se obter estéreo através de duas imagens omnidirecionais. Tradicionalmente, em sistemas omnidirecionais, faz-se uma correção das distorções das imagens para facilitar o processamento das mesmas. Neste trabalho, todo processamento foi feito diretamente nas imagens originais, mostrando a viabilidade do sistema.

As funções foram implementadas em linguagem C, num ambiente visual chamado Khoros, que possui uma biblioteca de funções relacionadas ao processamento e visualização de imagens. Foram implementadas as funções de extração de característica (SUSAN e filtro), correspondência (epipolar e correlação) e triangulação (distância) para o sistema de visão, e as funções Grade de Ocupação (Occupancy Grids), Leituras (Readings) e Bayes.

Testes foram realizados num ambiente virtual para validação dos algoritmos em situações mais controladas, e depois utilizou-se um ambiente real.

O sistema ainda necessita de otimizações, principalmente no algoritmo de correlação, muito sensível aos erros de deslocamento entre uma imagem e outra.

Um modelo de sensor foi proposto e implementado, juntamente com um processo de estimação e de atualização.

BIBLIOGRAFIA

- [1] Brooks, R. A., A robust layered control system for a mobile robot. September 1985.
- [2] Brooks, R. A., Elephants don't play chess. *Robotics and Autonomous Systems* 6 (1990)
- [3] Cheeseman, P, In defense of Probability. SRI Internacional.
- [4] Corrêa, F. Mapeamento de ambientes com sistema de visão omnidirecional. Trabalho de formatura, Escola Politécnica da Universidade de São Paulo, 2002
- [5] DeSouza, G. N., Kak, A. C., Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 2, February 2002.
- [6] Elfes, A. Occupancy Grids: A probabilistic framework for robot perception and navigation. Ph. D Thesis. 1989.
- [7] Elfes, A. Occupancy Grids: A Stochastic Spatial Representation for Active Robot Perception.
- [8] Delahoche, L., Pégard, C., Marhic, B., Vasseur, P. A navigation system based on an omnidirectional vision sensor. *IEEE Proc. IROS 97*.
- [9] Faugeras, O., Three-dimensional computer vision: a geometric view point.
- [9] Faugeras, O., Hotz, B., Mathieu, H., Viéville, T., Zhang, Z., Fua, P., Théron, E., Moll, L., Berry, G., Vuillemin, J., Bertin, P., Proy, C., Real time correlation-based stereo: algorithm, implementations and applications, août 1993.

- [10] Gonzalez, R. C., Wintz, P., Digital image processing.
- [11] Joung, I. S., Cho, H. S., An active omni-directional range sensor for mobile robot navigation. Control Engineering Practice 6 (1998).
- [12] Giacomo, G., Iocchi, L., Nardi, D., Rosati, R. Moving a Robot: The KR&R Approach at Work. 1996
- [13] Koyasu, H., Miura, J., Shirai, Y. Recognizing Moving Obstacles for Robot Navigation using Real-time Omnidirectional Stereo Vision. Journal of Robotics and Mechatronics, Vol. 14, 2002 (to appear)
- [14] Martin, C. M., Moravec, H. Robot Evidence Grids. 1996
- [15] Mataric, M. J. Integration of representation into goal-driven behavior-based robots. 1992.
- [16] Matthies, L., Elfes, A. Integration of sonar and stereo range data using a grid-based representation. 1998 IEEE.
- [16] Moravec, H. Robot spatial perception by stereoscopic vision and 3D evidence grids. 1996
- [18] Murray, D., Jennings, C. Stereo vision based mapping and navigation for mobile robots.
- [19] Murray, D., Little, J. Using real-time stereo vision for mobile robot navigation.
- [20] Naccache, F. F., Hachiya, R. K., Visão estéreo a partir de seqüências de imagens geradas por uma camera panorâmica central com espelho hiperbólico. Trabalho de formatura, Escola Politécnica da Universidade de São Paulo, 2001.
- [21] Okutomi, M., Kanade, T., A multiple-baseline stereo. IEEE Transactions on pattern analysis and machine intelligence, vol. 15, no. 4, April 1993.

- [22] Papoulis, A. Probability, Random Variables, and Stochastic Processes
- [23] Pearl, J., Bayesian decision methods.
- [24] Smith, S. M., Brandy, J. M., SUSAN - A new approach to low level image processing.
- [25] Svoboda, T., Pajdla, T., Hlavác, V. Central Panoramic Cameras: Geometry and Design.
1997
- [26] Thrun, S. Robotic Mapping: A Survey. 2002
- [27] Vanmarcke, Random Fields.
- [28] Yagi, Y., Nishizawa, Y., Yachida, M., Map-based navigation for a mobile robot with omnidirectional image sensor COPIS. IEEE Transactions on robotics and automation, vol. 11, no. 5, October 1995.
- [29] Winters, N., Gaspar, J., Lacey, G., Santos-Victor, J., Omni-directional vision for robot navigation. 2000.