

Modelo de Visão Computacional de Baixo Nível Antropomórfica com aplicação em Robótica Móvel

Mathias José Kreutz Erdtmann Christian Emanuel Mapurunga Silvano
Dr.Ing. Marcelo Ricardo Stemmer (Orientador)
Universidade Federal de Santa Catarina
Departamento de Automação e Sistemas
Sistemas Industriais Inteligentes
erdtmann@das.ufsc.br
silvano@das.ufsc.br
marcelo@das.ufsc.br

Resumo

O presente trabalho descreve uma arquitetura para o desenvolvimento de sistemas de visão baseado no antigo modelo antropomórfico de Marr para visão de Baixo Nível. Como exemplo de aplicação deste modelo, um problema de rastreamento e coleta de objetos por um robô móvel será resolvido utilizando um modelo simplificado da arquitetura proposta. São utilizadas técnicas de obtenção de bordas e medições de distâncias para resolver o problema da localização do robô em relação aos objetos a serem rastreados. O objetivo principal do trabalho é mostrar que, através de um modelo simplificado, pode-se obter soluções de baixo custo e com bons resultados. A implementação de um modelo completo poderia resolver problemas complexos e de grande importância prática.

Palavras-Chave: Robótica Móvel, Sistema de Visão, Detecção de Bordas, Medição de Distância

1. Introdução

O termo visão computacional foi pela primeira vez formalizado por David Marr, em 1982, quando o mesmo explicitou as hipóteses assumidas e as definições decorrentes das mesmas[1].

O cerne de sua pesquisa era o relacionamento das técnicas computacionais com a morfofisiologia humana, em parte por acreditar que os problemas que seriam enfrentados pela robótica no futuro refletiriam os enfrentados pelo corpo humano e em parte devido a sua formação técnica.

Baseado no modelo de Marr, infere-se a seguinte subdivisão para a visão computacional:

Baixo Nível Pré-processamento da imagem, cujo objetivo é fornecer informação ao Alto Nível, baseando-se apenas nas informações obtidas pelo sistema de aquisição de imagens.

Alto Nível Utiliza as informações fornecidas pelo Baixo Nível e informações contidas numa base de conhecimento, a qual pode ser consultada e alterada em função do sistema de visão de baixo nível e das funções pré-estabelecidas pelo sistema.

Nota-se que a visão computacional de alto nível é dependente da de baixo nível para obter dados referentes à imagem adquirida. Esta divisão, a princípio um tanto artificial, baseia-se basicamente em um princípio antropomórfico. A visão humana divide-se em dois estágios, sendo o primeiro relacionado às funções realizadas pelo olho humano, e a segunda às funções executadas pelo cérebro.

Marr constatou que os olhos humanos obrigatoriamente realizam uma forma de pré-processamento seguido de uma compressão de dados, transmitindo os dados comprimidos para o cérebro. A justificativa para a obrigação é que as ligações sinápticas entre o olho e o cérebro não suportariam uma taxa de transmissão alta o suficiente para a transferência de informações não comprimidas.

Seguindo o princípio antropomórfico, detecta-se algumas importantes propriedades a serem cumpridas de forma a compreender todas as informações necessárias com a devida compressão, a listar:

1. Reconhecimento das bordas (contorno) dos objetos;
2. Senso tridimensional a partir de imagens bi-dimensionais;

3. Extração de relevos a partir da mudança de luminosidade na superfície dos objetos;
4. Percepção de movimento;

Esta foi a proposta original de Marr e, em parte, razão do insucesso de seu modelo: Marr defendia em sua conjectura que a imagem estática poderia ser totalmente reconstruída apenas a partir das bordas da mesma, o que, mais tarde, provou-se falso por meio de um contra exemplo.

No entanto, por conta desta falha, acabou-se por não creditar o devido valor ao estudo de Marr que inclui, além do modelo, descrições concisas de algoritmos para a implementação completa de visão computacional de baixo nível a partir de seu modelo antropomórfico (apesar da falha na reconstrução a implementação podia ser considerada funcional), incluindo preciosismos que considerava necessário, como a eliminação gradativa de algoritmos iterativos, uma vez que os mesmos são fisiologicamente irrealizáveis pelo corpo humano, mesmo que seja difícil a não adoção de iterações em programas computacionais.

2. Modelo da visão computacional de baixo nível

Dado o fato do modelo de Marr falhar apenas quanto a reconstrução da imagem propõe-se um modelo diferente da camada de visão computacional de baixo nível. O novo modelo permitiria a reconstrução da imagem, baseada em trabalhos recentes, acrescentando uma propriedade a mais na definição de baixo nível – a informação das texturas[3].

Considera-se aqui que textura seria toda a informação da imagem que não pode ser inferida a partir das bordas da mesma, desta forma garantindo a reconstrução por definição (se for possível a compactação da textura aos níveis exigidos).

Desta forma define-se:

Visão de Baixo Nível Fornece toda informação relativa a imagem necessária à Visão de Alto Nível, sendo composta das informações referentes a:

1. Bordas (contorno) dos objetos;
2. Texturas;
3. Distância dos objetos (obtidas a partir de imagens bi-dimensionais);
4. Relevos dos objetos (obtidas a partir da mudança de luminosidade na superfície dos mesmos);
5. Movimentação (obtidas a partir de sequência de imagens);

Nota-se que a imagem em si não é fornecida ao alto nível, a exemplo da estrutura do sistema humano de visão,

mas a mesma pode ser reconstruída a partir das informações de bordas e texturas.

A construção de uma arquitetura de visão computacional desta forma desacopla o problema de visão, uma vez que os algoritmos de alto nível devem ser criados de forma a utilizar diretamente as informações fornecidas pelo baixo nível (apesar de sempre ser possível reconstruir a imagem original, isto não seria muito eficiente).

Este desacoplamento poderia incluir a separação do equipamento de aquisição + sistema de visão de baixo nível e sistema de visão de alto nível. Uma câmera microprocessada, por exemplo, poderia fornecer a um computador toda a informação explicitada anteriormente via rede a um computador remoto, a executar o processamento de alto nível. Esta arquitetura fornece uma facilidade extra para construção de sistemas distribuídos de processamento de imagem.

A analogia com o sistema de visão humana permanece, como pode ser visto nas figuras 1 e 2.

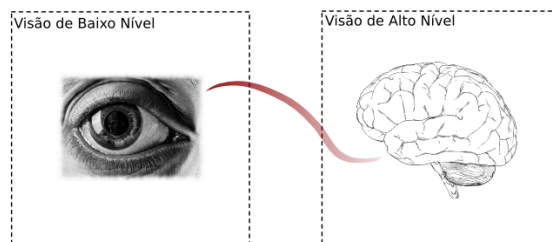


Figura 1. Sistema de visão humana e suas sub-divisões

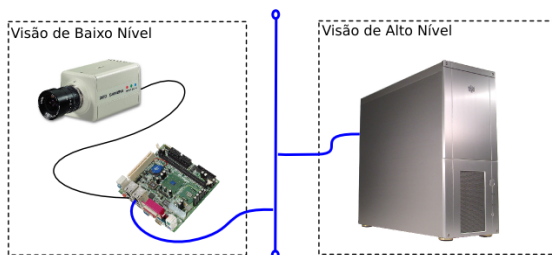


Figura 2. Sistema de visão computacional segundo modelo proposto (exemplo de arquitetura possível com processamento distribuído)

2.1. Definições detalhadas

Para o modelo descrito serão utilizadas definições que não diferem (muito) do usual, mas que, mesmo assim, precisam ser descritas com mais rigor, pois os princípios de aplicação a serem propostos são fortemente baseados nestes termos.

Abaixo segue as definições tomadas neste artigo, que consistem em leves adaptações da literatura.



Figura 3. Imagem de Lena Soderberg, tipicamente utilizada para testes[3]



Figura 4. Bordas da Imagem 3[3]

Imagem Função a duas variáveis ($f(x, y)$), onde o valor da função é proporcional à luminosidade recebida no sensor nas respectivas coordenadas $((x, y))$. Imagens monocromáticas são representadas por funções escalares ($f : \mathbb{R}^2 \rightarrow \mathbb{R}$) e imagens coloridas por funções vetoriais ($f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$). Se a função f é discreta de suporte compacto e os valores diferentes de intensidade luminosa que podem ser assumidos são finitos diz-se que a imagem é digital. (Vide figura 3)



Figura 5. Texturas da Imagem 3[3]

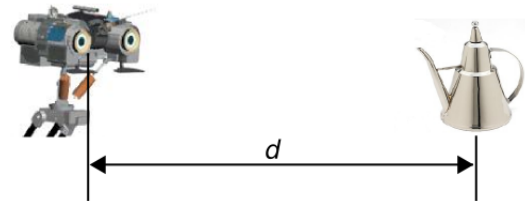


Figura 6. Distância (d) entre observador (esq.) e objeto (dir.)

Bordas Regiões de grande variação da intensidade luminosa. Para imagens diferenciáveis por partes o conjunto de bordas pode ser tomado como sendo a união do conjunto dos pontos de máximo da função ∇f nas regiões diferenciáveis e do conjunto de pontos de descontinuidade. Para imagens digitais, é necessário definir o que seria “grande variação” e/ou utilizar operadores diferenciais. (Vide figura 4)

Texturas Diferença entre a imagem do objeto original e do reconstruído a partir das informações de sua borda. Constituído basicamente de pequenas (em amplitude e extensão) variações de luminosidade existentes numa mesma região, tipicamente formando um padrão aproximadamente uniforme. (Vide figura 5)

Distância do objeto Métrica que determina a distância entre o observador e um ponto fixo do objeto observado. (Vide figura 6)

Mapa de Relevo do objeto Imagem constituída pela diferença entre a distância do observador aos pontos de um objeto e a distância deste objeto. (Vide figura 7)



Figura 7. Relevo da Imagem 3



Figura 8. Movimento observado (dir) do primeiro instante (esq) ao segundo instante (meio). Não foram detectados movimentos nas regiões pretas

Movimentação Diferença na posição e/ou orientação de um objeto em imagens tomadas em instantes de tempo distintos. (Vide figura 8)

3 Princípios de solução para aplicação em robótica móvel

Para exemplificar a estrutura proposta, propõe-se alguns algoritmos para a implementação de um sistema de visão de baixo nível a ser implementado em um robô móvel, que terá que se movimentar em um ambiente estruturado de chão claro e paredes escuras (iluminação homogênea), recolhendo esferas de diferentes cores, cujos tamanhos são relacionados às cores. (Vide figura 9)

Devido a estruturação no ambiente e das habilidades necessárias ao robô, elimina-se alguns dos requisitos do sistema de visão. Além disso, existem as limitações no equipamento, uma vez que o robô só possuirá uma câmera, o que a princípio impossibilita a solução comum para medir distâncias a objetos (via visão estereométrica).

Após as devidas simplificações em baixo nível, o sistema de visão de alto nível receberá informações sobre as bordas e a posição dos objetos no campo de visão relativa à

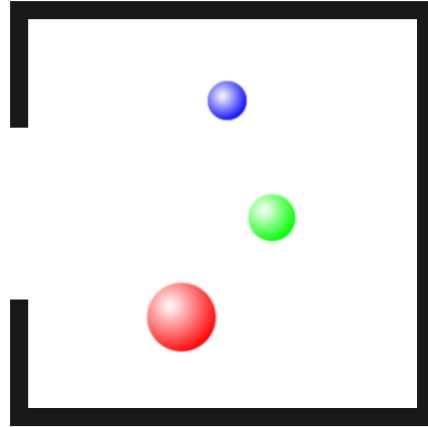


Figura 9. Vista de topo do ambiente estruturado, com as esferas coloridas

posição atual do robô (distância do objeto ao robô). Devido a ausência de uma segunda câmera para constituir uma dupla de imagens para realizar a triangulação, só é possível inferir a distância de uma esfera pelo fato da mesma ter um tamanho constante (dependente somente de sua cor).

Apesar de bastante limitadas, estas informações devem ser suficientes para o funcionamento do robô em um ambiente estruturado da forma proposta. Se o robô possuísse todo o equipamento necessário para implementar completamente toda a camada de baixo nível, o reconhecimento de esferas faria parte do processamento de alto nível, mas da forma aqui descrita, acabou sendo incluído no processamento de baixo nível, para que se pudesse identificar a distância das esferas ao robô.

Como pode ser notado, a arquitetura é flexível e não necessariamente um sistema de visão irá requerir o conjunto completo de habilidades descritas.

3.1 Equipamento a ser utilizado

É requisito de sistema que o processamento ocorra de forma embarcada no robô. Optou-se então, pelo uso de um PocketPC ligado a uma placa de aquisição e um sistema câmera e lentes. O PocketPC, como unidade de processamento lógico terá como funções tanto o processamento de imagens (visão baixo/alto nível) quanto o controle dos atuadores. (vide figura 10)

3.2 Detecção de Bordas

Como bordas são caracterizadas por variações abruptas na intensidade luminosa de um ponto com relação aos pontos próximos, a primeira solução é a utilização de matrizes



Figura 10. Equipamento do sistema

simples de correlação, que relacionam a intensidade de um pixel (a tupla de uma posição e sua respectiva intensidade será chamado pixel, constituindo, para imagens digitas, um átomo de imagem) com a dos pixels vizinhos, por meio de convolução (aqui denotada por $\langle f * G \rangle$, convolução entre a imagem f e o filtro G) [2].

Exemplos destas matrizes são: Sobel (construída a partir da aproximação da soma das componentes do gradiente para imagens discretas) e Laplaciana (construída a partir da aproximação do laplaciano para imagens discretas).

O procedimento para o detector de bordas Sobel é tomar duas matrizes:

$$S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

E então contruir uma imagem f_{bordas} onde $f_{bordas}(x, y) = |\langle f * S_y \rangle(x, y)| + |\langle f * S_x \rangle(x, y)|$

Já o detector de bordas Laplaciano é formado a partir da matriz:

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

E então contruir uma imagem f_{bordas} onde $f_{bordas}(x, y) = \Delta f(x, y) = \langle f * L \rangle$.

Nota-se que todas as matrizes construídas por este processo têm a soma de suas entradas nulas, desta forma regiões onde a intensidade é constante têm intensidade zero na imagem f_{bordas} .

O resultado destes dois procedimentos pode ser conferido nas figuras 12 (Sobel) e 13 (Laplaciano). A imagem original (figura 11) foi construída acrescentada de ruído, para verificar as propriedades dos detectores de bordas quando da presença dos mesmos.

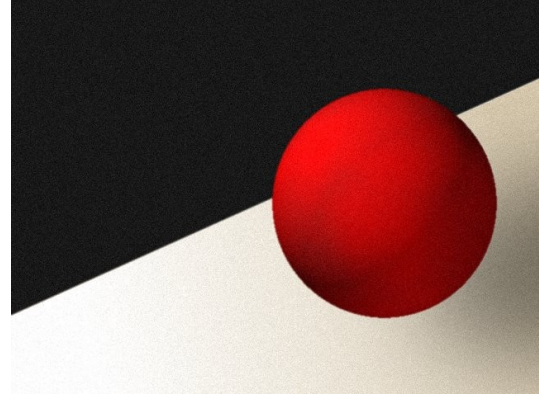


Figura 11. Imagem-teste original

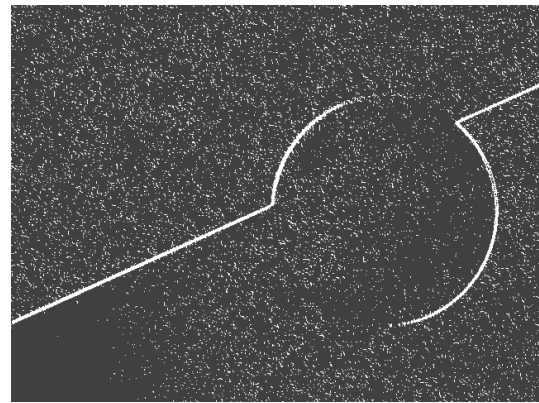


Figura 12. Detector de Bordas Sobel

Percebe-se que as imagens obtidas para estes detectores de borda detectaram as variações devido ao ruído e/ou textura dos objetos como ruído. Isto ocorre porque o ruído é constituído de variações abruptas espúrias na intensidade luminosa.

Uma solução para este problema é a utilização de filtros para amenizar o ruído, dando origem ao detector de bordas via Laplaciano da Gaussiana, o qual filtra o sinal com desfocagem gaussiana (eliminando ruídos brancos) e posteriormente aplica o detector de bordas Laplaciano, como visto acima.

Marr propôs uma variante, onde aplica-se o Laplaciano no núcleo de Gauss e só posteriormente filtra-se a imagem com a matriz obtida, gerando economia de recursos. Para obter melhores resultados, realiza-se este processo duas vezes, uma para um núcleo de Gauss de raio pequeno (3 pixels, por exemplo) e um com raio maior (5 pixels, p.e.), sendo bordas as regiões comuns a ambas as imagens. Este detector é chamado detector de Marr e o resultado de uma aplicação pode ser visto na figura 14.

Um algoritmo mais complexo é proposto por Canny, que utiliza a desfocagem gaussiana, operador derivativo e em

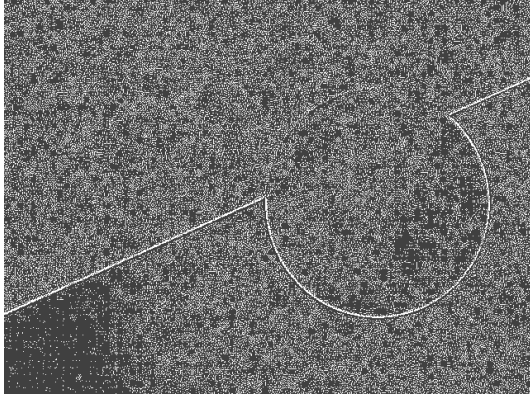


Figura 13. Detector de Bordas Laplaciano

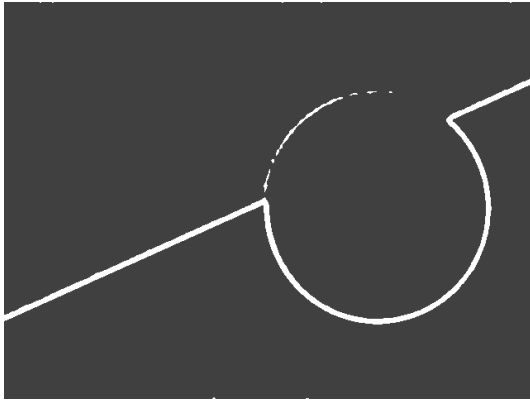


Figura 14. Detector de Bordas Marr

seguida um procedimento para eliminação de pontos não-máximos (espúrios) aprimorado. A aplicação do detector Canny pode ser vista na figura 15. Com este detector obtém-se a eliminação de ruídos, como existia com o detector de Marr, porém com as bordas mais bem definidas, uma vez que as bordas com o detector de Marr ficavam um tanto grossas, aumentando a incerteza de medições que requeiram mais exatidão.

Com exceção do detector de Canny estes algoritmos foram implementados pelos autores em C++, com auxílio da biblioteca OpenCV e de trabalhos anteriores (S2iImage, S2iFilter) realizados no laboratório Sistemas Industriais Inteligentes (S2i) da Universidade Federal de Santa Catarina, também produto dos estudos desenvolvido pelos autores e pelos demais integrantes do laboratório. Todas as bibliotecas desenvolvidas pelo S2i são software livre, sob licença LGPL.

3.3 Medição de Distância

Como observado anteriormente, neste exemplo de aplicação não se utiliza estereometria, sendo portanto ne-

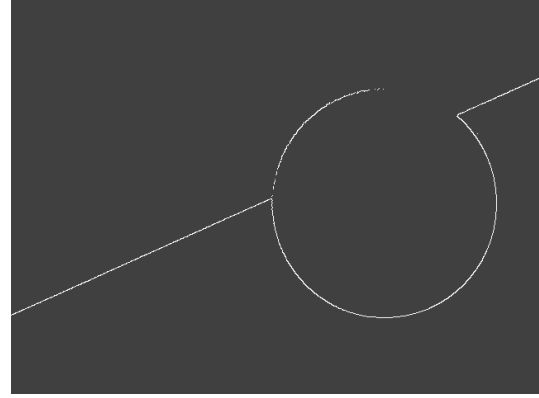


Figura 15. Detector de Bordas Canny

cessário utilizar informações conhecidas da geometria dos objetos presentes no ambiente, bem como da câmera. Assim sendo, a solução adotada para se obter a distância do observador a uma esfera foi modelada em função da distância focal, o raio em pixels da circunferência visualizada e raio da esfera real em milímetros. A geometria do modelo pode ser vista na figura 16

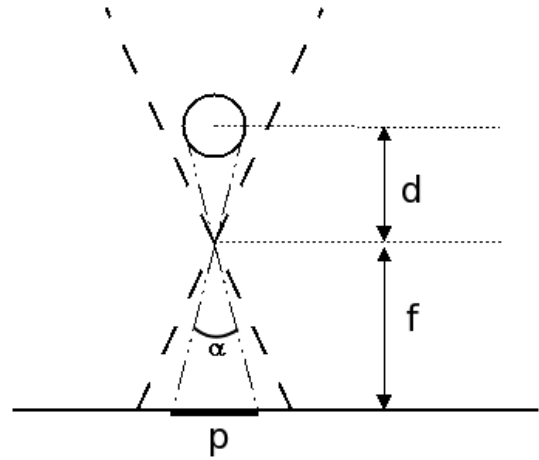


Figura 16. Modelo da imagem projetada no sensor da câmera.

A partir das condições geométricas explicitadas na figura, chegou-se ao modelo:

$$D_t = f \left(1 + \frac{R}{2 \cdot \beta(P) \cdot P} \right)$$

, onde D_t é a distância total do sensor da câmera à esfera, f é a distância focal do conjunto ótico e $\beta(P)$ é uma função que correlaciona o raio da esfera de interesse observado em pixels e em milímetros e foi obtido a partir de uma

interpolação linear de dados obtidos através de ensaios de calibração.

O algoritmo proposto para medição da distância é:

1. Obter a imagem g (vide figura 17);
2. Iniciar o processo de segmentação dos círculos: (vide figura 18)
 - (a) Separar regiões predominantemente vermelha, verde e azul;
 - (b) Processo de eliminação de pontos espúrios por dilatação/erosão; Assim obtendo de 0 a 3 regiões predominantemente vermelha, verde ou azul.
3. Verificar se os objetos encontrados constituem circunferências, a partir d:
 - (a) Para cada região predominante, verificar se a mesma possui ação (norma-1 do sinal) suficiente para constituir um círculo;
 - (b) Verificar se o perfil horizontal e vertical condizem com uma circunferência, sendo o perfil tomado pela ação das linhas (horizontal) e colunas (vertical) da região. Verifica-se a condizência através de comparação de raios máximo horizontal e vertical.
4. Verificar, para cada objeto encontrado, se for uma circunferência, cor e raio do mesmo;(vide figura 19)
5. Caso algum objeto encontrado não seja considerado uma circunferência, ele terá menor prioridade em relação às circunferências encontradas (podem representar esferas parcialmente oclusas ou muito distantes). Em qualquer caso o robô deverá manter a atenção nos objetos próximos e corretamente identificados.
6. Caso nenhum objeto tenha sido encontrado, o robô deverá movimentar-se para buscar melhor campo de visão.
7. A partir do raio de cada circunferência, inferir a distância conforme o modelo proposto (onde conhece-se o raio em milímetros para cada cor de esferas).

3.3.1 Fontes de incerteza

Devido às considerações feitas, as principais fontes de incerteza da medição de distância provêm das características não ideais da lente, do processo de calibração (posicionamento da esfera durante os ensaios de medição) e o processo de interpolação para obtenção da função β .

Todavia, durante a validação do método obteve-se erro máximo de 1 cm para uma distância máxima de 50 cm, podendo este valor ser tomado como a incerteza das medições



Figura 17. Imagem adquirida do ambiente com esfera.

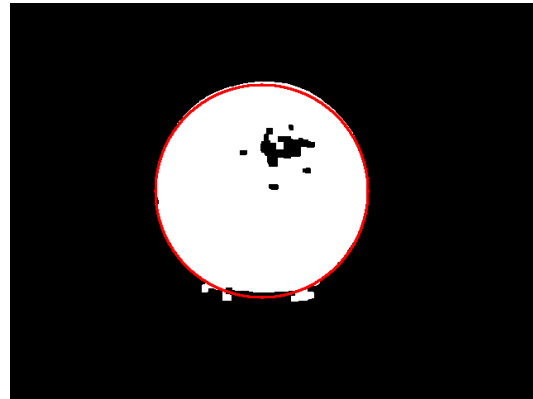


Figura 18. Imagem 17 segmentada.

realizadas por este processo. A interpolação de β só é válida para valores estreitos de distância, limitados pelo processo de calibração do sistema, podendo levar a erros muito maiores para distâncias maiores que a maior distância utilizada durante a calibração.

Tanto os erros quanto a faixa de operação para as medições são satisfatórios para os requisitos da tarefa e viabilizam a navegação do robô e busca pelas esferas.

4 Conclusões

A abordagem proposta torna a implementação de um sistema de visão algo mais estruturado e a solução deverá ser satisfatória para a navegação do robô dentro do ambiente proposto. Apesar das simplificações assumidas no processo de medição, os erros obtidos não deverão influenciar no perfeito cumprimento da tarefa. Como foi utilizado uma arquitetura modular para criação do sistema de visão é possível o aprimoramento de partes do algoritmo mais rapidamente.

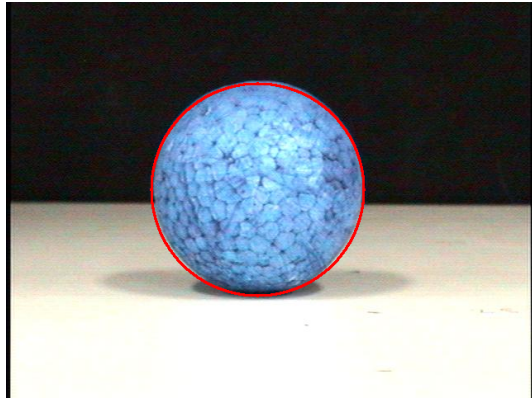


Figura 19. Imagem 17 com circunferência detectada em vermelho.

[3] M. B. Ruskai, G. Beylkin, and R. Coifman. *Wavelets and Their Applications*. Jones and Bartlett, 1992.

5 Agradecimentos

Este trabalho foi realizado com o apoio de várias instituições e pessoas. Sem o apoio destes, esta pesquisa não seria possível.

- WZL/RWTH-Aachen (Werkzeugmaschinenlabor - Rheinisch-Westfälische Technische Hochschule Aachen) - Alemanha, pelo apoio financeiro através de bolsa de iniciação científica e equipamentos.
- CNPQ (Conselho Nacional de Desenvolvimento Científico e Tecnológico) - Brasil, pelo apoio financeiro através de bolsa de iniciação científica.
- CTC/UFSC-Florianópolis (Centro Tecnológico da Universidade Federal de Santa Catarina) - Brasil, pelo apoio financeiro para o custeio das viagens de pesquisa.
- S2i-DAS/UFSC - Florianópolis (laboratório Sistemas Industriais Inteligentes do Departamento de Automação e Sistemas da UFSC) - Brasil, que é o lugar onde tudo acontece!
- Robota-DAS/UFSC - Florianópolis (Grupo de Estudos em Robótica Móvel do DAS), pelo apoio técnico, pois nem só de sistema de visão se faz um robô.
- Aos amigos, parentes, namoradas, pelo apoio e correções em hora de necessidade.

Referências

- [1] Y. Meyer. *Wavelets – Algorithms and Applications*. Society for Industrial and Applied Mathematics, 1993.
- [2] J. R. Parker. *Algorithms for Image Processing and Computer Vision*. John Wiley And Sons, 1997.