

UNIVERSIDADE ESTADUAL DE PONTA GROSSA
MESTRADO EM COMPUTAÇÃO APLICADA

ANDREY ORLOSKI

**PROCEDIMENTO PARA AUTOLOCALIZAÇÃO DE ROBÔS EM CASAS DE
VEGETAÇÃO UTILIZANDO DESCRITORES SURF:
Implementação Sequencial e Paralela**

PONTA GROSSA

2015

ANDREY ORLOSKI

**PROCEDIMENTO PARA AUTOLOCALIZAÇÃO DE ROBÔS EM CASAS DE
VEGETAÇÃO UTILIZANDO DESCRITORES SURF:
Implementação Sequencial e Paralela**

Dissertação de Mestrado apresentado ao Programa de Pós-Graduação em Computação Aplicada, curso de Mestrado em Computação Aplicada da Universidade Estadual de Ponta Grossa, como requisito para obtenção do título de Mestre.

Orientador: Prof. Dr. José Carlos Ferreira da Rocha

PONTA GROSSA

2015

Ficha Catalográfica
Elaborada pelo Setor de Tratamento da Informação BICEN/UEPG

Orloski, Andrey
072 Procedimento para autolocalização de robôs em casas de vegetação utilizando descritores surf:implementação sequencial e paralela/ Andrey Orloski. Ponta Grossa, 2015.
82f.

Dissertação (Mestrado em Computação Aplicada - Área de Concentração: Agricultura), Universidade Estadual de Ponta Grossa.

Orientador: Prof. Dr. José Carlos Ferreira da Rocha.

1.Autolocalização. 2.Agrobot.
3.Algoritmo SURF. 4.Casa de vegetação.
5.Visão computacional. I.Rocha, José Carlos Ferreira da. II. Universidade Estadual de Ponta Grossa. Mestrado em Computação Aplicada. III. T.

CDD: 006.3

Andrey Orloski

**“PROCEDIMENTO PARA AUTO LOCALIZAÇÃO DE ROBÔS EM CASAS DE
VEGETAÇÃO UTILIZANDO DESCRITORES SURF: IMPLEMENTAÇÃO SEQUENCIAL
E PAPALELA”**

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação em Computação Aplicada da Universidade Estadual de Ponta Grossa, pela seguinte banca examinadora:

Orientador:


Dr. José Carlos Pereira da Rocha
UEPG


Dr. Rosane Balate
UEPG


Dr. Emerson Fey
UNIOESTE

Ponta Grossa, 04 de setembro de 2015.

Dedico a toda minha família, em especial aos meus pais,
Estanislau e Ana.

AGRADECIMENTOS

Agradeço à Deus, que está acima de todas as coisas e nos revela a sua sabedoria.

Agradeço à minha família, que me apoiou e incentivou em todos os momentos desta caminhada, me fornecendo forças para seguir em frente.

Agradeço à Barbara Justus, pelo esforço, apoio e compreensão nas horas que mais precisei.

Agradeço ao meu amigo Roberson Fernandes Alves, pela parceria, ensinamentos e apoio nesta longa jornada.

Agradeço ao meu orientador, Prof. Dr. José Carlos Ferreira da Rocha, pela oportunidade dada a mim e por me mostrar o caminho, através de seus conhecimentos.

Agradeço a todos que direta ou indiretamente contribuíram para a conclusão desta pesquisa, em especial aos professores e amigos do Mestrado em Computação Aplicada da UEPG.

RESUMO

Este trabalho descreve um procedimento para autolocalização de *agrobots* móveis e autônomos em casas de vegetação. Isto é, a determinação da posição do robô em relação a um sistema de coordenadas, usando procedimentos e recursos computacionais. O procedimento proposto emprega técnicas de visão computacional para reconhecer objetos marcadores na casa de vegetação e, a partir deles, estimar a coordenada do robô em um plano paralelo a superfície da estufa. A detecção da presença dos marcadores na cena é realizada através do algoritmo SURF. Para viabilizar a estimativa das coordenadas, a partir de dados contidos em uma única imagem, o método de Rahman *et al.* (2008), que consiste em determinar a distância entre uma câmera e um objeto marcador, foi estendido para permitir o cômputo de coordenadas. O desempenho do procedimento proposto foi avaliado em três experimentos. No primeiro experimento, o objetivo foi verificar, em laboratório, a influência da resolução da imagem sobre a precisão. Os resultados indicam que, ao reduzir a resolução da imagem, o alcance do procedimento é prejudicado para reconhecimento dos marcadores. Estes resultados também mostram que, ao reduzir a resolução, o erro na estimativa das coordenadas em relação à distância entre a câmera e o marcador aumenta. O segundo experimento executou um teste que avalia o desempenho computacional do algoritmo SURF, em termos de tempo de computação, no processamento das imagens. Isto é importante pois *agrobots* usualmente precisam executar tarefas que demandam capacidade de processamento em tempo real. Os resultados deste teste indicam que a eficiência do procedimento cai com o aumento da resolução da imagem. Um segundo teste comparou o tempo de processamento de duas implementações do algoritmo. Uma que explora uma versão sequencial do algoritmo SURF e outra que usa uma implementação paralela. Os resultados deste teste sugerem que a implementação paralela foi mais eficiente em todas as resoluções testadas, apresentando uma melhora proporcional quase constante. O terceiro experimento foi realizado em uma casa de vegetação com objetivo de avaliar o desempenho do procedimento proposto no ambiente para o qual foi projetado. Os resultados de campo se mostraram semelhantes aos do laboratório, mas indicam que variações de iluminação exigem ajustes de parâmetros do algoritmo SURF.

Palavras-chave: Autolocalização. *Agrobot*. Algoritmo SURF. Casa de Vegetação. Visão Computacional.

ABSTRACT

This paper describes a procedure for self-localization of mobile and autonomous *agrobots* in greenhouses, that is, the determination of the robot's position relative to a coordinate system, using procedures and computational resources. The proposed procedure uses computer vision techniques to recognize markers objects in the greenhouse and, from them, estimate the coordinate of the robot in a parallel plane to the surface of the stove. The detection of the presence of markers in the scene is performed using the SURF algorithm. To enable the estimation of coordinates, based on data contained in a single image, the method of Rahman *et al.* (2008), which consists in determining the distance between a camera and a marker object has been extended to allow the coordinate calculation. The performance of the procedure was evaluated in three experiments. In the first experiment, the objective was to verify, in the laboratory, the influence of image resolution on accuracy. The results indicate that by reducing the image resolution, the range of the process is impaired for the recognition of the markers. These results also show that by reducing the resolution, the error in estimating the coordinates relative to the distance between the camera and the marker increases. The second experiment ran a test that evaluates the computational performance of the SURF algorithm, in terms of computing time, in the image processing. This is important because *agrobots* usually need to perform tasks that require the processing power in real time. The results of this test indicate that the efficiency of the procedure drops with the increase of image resolution. A second test compared the processing time of two implementations of the algorithm. One explores a sequential version of the SURF algorithm and another uses a parallel implementation. The results of this test suggest that the parallel implementation is more efficient in all tested resolutions, with an almost constant proportionate improvement. The third experiment was performed in a greenhouse to evaluate the performance of the proposed procedure in the environment for which it was designed. Field results were similar to the laboratory, but indicate that lighting variations require parameter settings of the SURF algorithm.

Keywords: Self-Localization. *Agrobot*. SURF Algorithm. Greenhouse. Computer Vision.

LISTA DE ILUSTRAÇÕES

Figura 1: Sequência de processamento de um sistema de visão computacional	19
Figura 2: Exemplo de casa de vegetação	20
Figura 3: Esquema de funcionamento do método de Rahman <i>et al.</i> (2008)	22
Figura 4: Exemplo de detecção de marcador através do algoritmo SURF.....	24
Figura 5: Processamento do algoritmo SURF de maneira sequencial.....	25
Figura 6: Processamento do algoritmo P-SURF.....	27
Figura 7: Pseudocódigo para destacar as maiores semelhanças	30
Figura 8: Esquema com duas possíveis distâncias laterais para o robô.....	32
Figura 9: Dimensões do laboratório; Corredor simulado; Projeção dos pontos de leitura em milímetros	34
Figura 10: Marcadores digitais	35
Figura 11: Objeto marcador no final do corredor.....	35
Figura 12: Exemplos de imagens de marcadores capturadas com variação de profundidade..	38
Figura 13: Corredor simulado na casa de vegetação	40
Figura 14: Gráfico relacionando profundidade em metros e altura em <i>pixels</i> com ajuste de função para equação de segundo grau em imagens de 2MP.	45
Figura 15: Gráfico relacionando profundidade em metros e altura em <i>pixels</i> com ajuste de função para equação de potência em imagens de 2MP.	45
Figura 16: Gráfico comparando a relação θ média com a profundidade em metros com ajuste de função para equação de potência em imagens de 2MP.....	47
Figura 17: Gráfico relacionando o tempo de processamento com o limiar hessiano	51
Figura 18: Relação entre o tempo de processamento, a profundidade máxima obtida e a resolução utilizada para as implementações sequencial e paralela.....	52
Figura 19: Exemplo de leitura na casa de vegetação com profundidade de 5m.....	53
Figura 20: Gráfico relacionando profundidade em metros e altura em <i>pixels</i> com ajuste de função para equação de potência em imagens de 1,28MP.	70
Figura 21: Gráfico relacionando profundidade em metros e altura em <i>pixels</i> com ajuste de função para equação de potência em imagens de 0,72MP.	70

Figura 22: Gráfico comparando a relação média com a profundidade em metros com ajuste de função para equação de potência em imagens de 1,28MP.	72
Figura 23: Gráfico comparando a relação média com a profundidade em metros com ajuste de função para equação de potência em imagens de 0,72MP.	72

LISTA DE TABELAS

Tabela 1: Resultados da profundidade obtidos em relação aos limiares de comparação e hessiano com imagens de 2MP.....	42
Tabela 2: Resultados da profundidade obtida em relação aos limiares hessianos e o limiar de comparação de 0,85 nas resoluções de 0,72MP e 1,28MP.....	43
Tabela 3: Leitura manual da profundidade e leitura em <i>pixels</i> do objeto marcador em imagens de 2MP.....	44
Tabela 4: Valores coletados manualmente para obtenção da equação para cálculo de distância lateral em imagens de 2MP.....	46
Tabela 5: Média, desvio, máximo e mínimo do erro para as estimativas da profundidade e distância lateral em imagens de 2MP	48
Tabela 6: Relação entre o custo computacional de cada resolução com o limiar hessiano em segundos	50
Tabela 7: Resultados das estimativas na casa de vegetação	53
Tabela 8: Resíduos de profundidade para regressão da Equação 6 de segundo grau em imagens de 2MP	65
Tabela 9: Resíduos de profundidade para regressão da Equação 7 de potência em imagens de 2MP	66
Tabela 10: Resíduos de relação θ para regressão da Equação 8 de potência em imagens de 2MP	67
Tabela 11: Resíduos de relação θ para regressão da equação de segundo grau $\theta = 181,2862-1486,3\delta + 3259$ em imagens de 2MP	68
Tabela 12: Leitura manual da profundidade e leitura em <i>pixels</i> do objeto marcador em imagens de 1,28MP.	69
Tabela 13: Leitura manual da profundidade e leitura em <i>pixels</i> do objeto marcador em imagens de 0,72MP.	69
Tabela 14: Valores coletados manualmente para obtenção da equação para cálculo de distância lateral em imagens de 1,28MP	71
Tabela 15: Valores coletados manualmente para obtenção da equação para cálculo de distância lateral em imagens de 0,72MP	71
Tabela 16: Resultados de localização para cada uma das coordenadas de aquisição em imagens de 2MP	73

Tabela 17: Resultados de localização para cada uma das coordenadas de aquisição em imagens de 1,28MP	77
Tabela 18: Resultados de localização para cada uma das coordenadas de aquisição em imagens de 0,72MP	80

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS.....	15
1.1.1	Geral:	15
1.1.2	Específicos:	15
2	REVISÃO BIBLIOGRÁFICA.....	17
2.1	AUTOLOCALIZAÇÃO DE ROBÔS MÓVEIS AUTÔNOMOS.....	17
2.2	CASAS DE VEGETAÇÃO E ROBÔS MÓVEIS AUTÔNOMOS	20
2.3	ESTIMATIVA DE DISTÂNCIAS USANDO MÉTODOS DE PROCESSAMENTO DIGITAL DE IMAGENS	21
2.4	O ALGORITMO SURF.....	24
2.5	COMPUTAÇÃO PARALELA E O ALGORITMO SURF	26
3	METODOLOGIA	28
3.1	PROCEDIMENTO PARA AUTOLOCALIZAÇÃO DE ROBÔS E MÉTODO PARA ESTIMATIVA DE DISTÂNCIAS EM IMAGENS DIGITAIS	28
3.1.1	Treinamento dos parâmetros do método de Rahman	29
3.1.2	Extração de características e detecção de objetos marcadores através do algoritmo SURF	29
3.1.3	Determinação da coordenada do robô	31
3.2	PROTOCOLO EXPERIMENTAL.....	32
3.2.1	Infraestrutura utilizada na realização dos experimentos	33
3.2.2	Aquisição de imagens.....	36
3.2.3	Calibração do sistema.....	36
3.2.4	Experimentos de autolocalização	38
3.2.4.1	<i>Análise dos resultados</i>	39
3.2.5	Experimentos de teste em campo	40
4	RESULTADOS E DISCUSSÃO	41
4.1	RESULTADO DO PRÉ-PROCESSAMENTO.....	41
4.1.1	Treinamento dos parâmetros do algoritmo SURF	41
4.1.2	Obtenção de equações para profundidade e distância lateral	43
4.2	RESULTADOS DE LOCALIZAÇÃO.....	48
4.2.1	Obtenção das coordenadas do robô no ambiente	48
4.3	AValiação DO CUSTO COMPUTACIONAL	49
4.3.1	Utilização de algoritmo paralelo para redução do custo computacional.....	51
4.4	RESULTADOS DE CAMPO	52

4.5	DISCUSSÃO	54
5	CONCLUSÕES	57
5.1	TRABALHOS FUTUROS	59
	REFERÊNCIAS.....	60
	APÊNDICE A	65
	APÊNDICE B.....	66
	APÊNDICE C	67
	APÊNDICE D	68
	APÊNDICE E.....	69
	APÊNDICE F.....	73
	APÊNDICE G	77
	APÊNDICE H	80

1 INTRODUÇÃO

As casas de vegetação são construções que fornecem uma estrutura para o cultivo de plantas que necessitam de um ambiente protegido e o controle de condições ambientais para se desenvolverem. Com o objetivo de reduzir custos e aumentar a produtividade, as casas de vegetação têm passado por um processo de automação. Atualmente, muitas unidades de produção têm dispositivos que permitem controlar a temperatura, a umidade e a luminosidade sem a intervenção humana. Ainda assim, diversas tarefas realizadas em uma casa de vegetação necessitam de mão de obra humana.

Algumas dessas tarefas são repetitivas, exigem mobilidade ou o trabalho em ambiente insalubre, como por exemplo a pulverização de pesticidas e a inspeção para controle de pragas. O desenvolvimento de soluções baseadas em robôs móveis autônomos pode prover redução de riscos e custos associados a estas atividades, pois elimina a necessidade do ser humano estar em contato direto com substâncias nocivas à saúde e reduz a quantidade de mão de obra necessária para realização de tarefas. O emprego de *agrobots* (robôs agrícolas) também pode contribuir para aumentar a precisão na execução de tarefas que atualmente são executadas por pessoas, similarmente ao que acontece com o uso de robôs na indústria. Além disto, os robôs podem trabalhar ininterruptamente por mais tempo sem que seu desempenho seja reduzido significativamente.

Um robô é dito autônomo quando é capaz de determinar sua posição, planejar ações, identificar objetos e se locomover no ambiente em que opera. Neste cenário, a capacidade de determinar sua localização no ambiente é fundamental, pois viabiliza a realização das demais tarefas tais como a navegação e a tomada de decisão como se aproximar de objetos, podar, pulverizar ou monitorar plantas em uma casa de vegetação. Esteves (2005) e Gouveia (2008) colocam que a tarefa de autolocalização de robôs pode ser definida como a determinação da sua posição e da sua orientação, realizada a partir do próprio robô ou de um sistema externo, em cada instante de tempo.

O emprego de métodos baseados em visão computacional tem fornecido uma estratégia para abordar o problema da localização de robôs. Nesta estratégia, a ideia é aplicar métodos que combinam um conjunto de dados coletados por uma câmera com intuito de estimar as coordenadas que especificam a posição do robô no seu ambiente de operação. Isto é, a determinação da localização ocorre com o emprego de procedimentos que detectam e reconhecem objetos em imagens capturadas pelo robô e aplicam algum mecanismo de

raciocínio automático ou geométrico que usa estas informações para estimar a posição deste. Uma abordagem é a utilização de objetos a serem reconhecidos como marcadores dispostos em posições conhecidas da estufa, chamados objetos marcadores. Um objeto marcador é um objeto que se diferencia facilmente no ambiente, com intuito de servir de referência para o sistema de localização.

Rahman *et al.* (2008) apresentam um método para determinar a distância entre uma câmera e um objeto a partir da identificação deste último como um objeto marcador. O método calcula previamente a profundidade e a distância lateral entre o objeto e a câmera, através da aplicação de métodos de visão computacional, esta para encontrar a altura em *pixels* do objeto na imagem capturada. Naquele trabalho, a detecção dos objetos marcadores foi realizada de maneira manual, ou seja, um operador aplicava filtros de processamento digital de imagens para redução de ruídos e definição do contorno dos objetos. Este esquema de processamento não é adequado para um robô autônomo, visto que para este ser autônomo não pode depender da intervenção humana no tratamento dos dados.

O algoritmo SURF¹ (*Speeded Up Robust Features*) permite a identificação de objetos marcadores em imagens pois é um detector e descritor de pontos de interesse invariável à escala e rotação. O algoritmo exige um banco de dados contendo os objetos marcadores a serem verificados nas capturas do robô. Neste algoritmo, a detecção de um objeto marcador é realizada via comparação da similaridade de um conjunto de descritores (pontos de interesse) extraídos da imagem capturada com os descritores do objeto marcador. A similaridade é computada por um escore que calcula uma distância entre os pontos de interesse da imagem e do objeto marcador. Para realizar este cálculo é necessário definir previamente um limiar para o percentual mínimo de proximidade. Quando a distância é inferior a este limiar, o objeto marcador é aceito, portanto, este parâmetro tem grande influência sobre a precisão do algoritmo na identificação do objeto marcador correto. O algoritmo também tem por parâmetro um limiar que regula o número de pontos de interesse a serem detectados na imagem. O valor deste parâmetro influencia a precisão e o custo computacional do sistema.

Considerando o exposto, este trabalho propõe estimar a localização de um *agrobot* em uma casa de vegetação, empregando procedimentos que extraem informações por processamento digital de imagens. Mais detalhadamente, objetiva-se detectar objetos que servem de marcadores no ambiente em que o robô trabalha, a partir de imagens coletadas por uma câmera embarcada no mesmo utilizando o algoritmo SURF. Além disto, pretende-se

¹ SURF = Detector Robusto Acelerado de Características

estender o método proposto por Rahman *et al.* (2008), de forma que ele possa estimar as coordenadas do robô na casa de vegetação. Vale frisar que este trabalho não aborda a orientação do robô.

Dado este objetivo, é importante destacar que a autolocalização de robôs exige o processamento de dados em tempo real. No entanto, os procedimentos de visão podem ter um alto custo computacional, em termos de tempo de processamento, dificultando seu emprego em sistemas de tempo real.

Uma forma de reduzir o tempo de computação é empregar a computação paralela. Considerando isto, este trabalho também propõe o uso de uma solução de programação paralela através de GPU² (*Graphics Processing Unit*) na implementação do método proposto. Busca-se, com isso, aumentar a eficiência do procedimento de autolocalização sem reduzir a precisão do mesmo. Os desempenhos das soluções baseadas em GPUs e das implementações sequenciais são comparados.

1.1 OBJETIVOS

1.1.1 Geral:

- a) Desenvolver um procedimento baseado no método proposto por Rahman *et al.* (2008) para determinar a autolocalização de *agrobots* em casas de vegetação a partir da detecção de objetos marcadores em imagens com o algoritmo SURF;

1.1.2 Específicos:

- a) Estender o método de Rahman *et al.* (2008) para identificar as coordenadas de aquisição da imagem no ambiente;
- b) Elaborar um procedimento de autolocalização que integre o algoritmo de detecção de objetos SURF a versão estendida do método de Rahman *et al.* (2008);
- c) Determinar experimentalmente, para o algoritmo de detecção proposto, a distância máxima em que um objeto marcador é identificado, conforme a resolução das imagens analisadas;

² GPU = Unidade de Processamento Gráfico

- d) Mensurar o erro no cálculo das coordenadas obtidas em relação às coordenadas reais através do procedimento proposto neste trabalho;
- e) Desenvolver uma versão do procedimento proposto neste trabalho que explora recursos de programação paralela;
- f) Comparar experimentalmente o tempo de computação das versões sequencial e paralela do procedimento proposto;
- g) Avaliar o desempenho do sistema em uma casa de vegetação.

2 REVISÃO BIBLIOGRÁFICA

Este trabalho aborda o problema da autolocalização de robôs em duas etapas. Na primeira etapa, procedimentos de visão computacional são usados para detectar objetos marcadores na cena. As dimensões dos objetos marcadores são conhecidas e esta informação é explorada na segunda etapa do procedimento, que é a de calcular a distância do robô ao objeto marcador detectado na primeira etapa. A distância é calculada com o método proposto por Rahman *et al.* (2008) e a partir dos dados de distância determina-se a localização do *agrobot* na casa de vegetação.

Este capítulo apresenta os principais elementos do problema proposto e descreve os procedimentos usados no desenvolvimento do projeto. Assim, a seção 2.1 apresenta os principais conceitos e elementos do problema de autolocalização de robôs em ambientes fechados e aborda o uso de visão computacional na sua solução. A seção 2.2 apresenta um resumo sobre casas de vegetação e sua importância para a agricultura. Esta seção destaca os avanços advindos do uso da robótica na operação de casas de vegetação. A seção 2.3 descreve o procedimento de cálculo de distância proposto por Rahman *et al.* (2008). A seção 2.4 descreve o algoritmo SURF.

Uma vez que um dos objetivos deste trabalho é comparar o tempo de processamento de uma implementação paralela do procedimento proposto com a implementação sequencial, este capítulo também traz um resumo sobre computação paralela. Mais precisamente, a seção 2.5 descreve conceitos de programação paralela usando a arquitetura CUDA³ (*Compute Unified Device Architecture*), bem como apresenta uma versão paralela do algoritmo SURF.

2.1 AUTOLOCALIZAÇÃO DE ROBÔS MÓVEIS AUTÔNOMOS

Os robôs autônomos são máquinas inteligentes capazes de tomar decisões que objetivam atingir as metas especificadas sem a intervenção de ser humano para controlar suas escolhas (RUSSEL; NORVING, 2004; BEKEY, 2005). Um robô móvel autônomo, por sua vez, é dotado de um sistema de navegação. Isto é, um sistema que, dadas as percepções do ambiente coletadas pelos sensores do robô e uma abstração do espaço de trabalho, planeja sua

³ CUDA = Arquitetura Unificada de Dispositivos de Computação.

trajetória de locomoção (BALTZAKIS; TRAHANIAS, 2002; RUSSEL; NORVING, 2004; GOUVEIA, 2008; NAKHAEINIA *et al.*, 2011; YAGHOUBI *et al.*, 2013).

Um dos pontos de partida para a elaboração e verificação de planos de navegação é a determinação da posição do robô através da autolocalização. A tarefa de autolocalização de robôs pode ser definida como a estimativa da sua posição e da sua orientação, em cada instante de tempo, realizada a partir de um sistema embarcado no próprio robô ou de um sistema externo (ESTEVES, 2005; GOUVEIA, 2008). A autolocalização pode ser obtida de maneira relativa ou absoluta. A autolocalização relativa é realizada considerando um dado referencial inicial, onde o robô necessita sempre de um ponto de partida. A autolocalização absoluta é realizada de modo que o robô tem a capacidade de adquirir o conhecimento de sua localização sem a necessidade de um marco de referência (FOX, 1998).

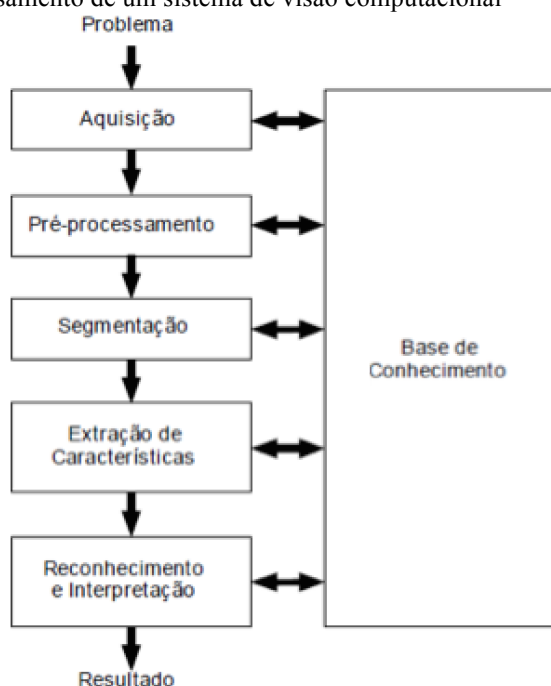
Os sensores usados para a coleta de dados para determinação da localização do robô podem ser passivos ou ativos (MOREIRA, 2005). Os sensores ativos possuem um sistema de radiação, ou seja, não necessitam de uma fonte externa para irradiar seu alvo. Os sensores passivos medem a radiação refletida e/ou emitidas pelos alvos que provém de uma fonte externa, como o sol, pois não possuem radiação própria. Este trabalho considera o uso de câmeras digitais, que são sensores passivos, ou seja, necessitam de uma luz externa refletida, para coleta de informações sobre o ambiente e, métodos de visão computacional para obtenção da localização do robô.

A visão computacional (VC) consiste basicamente no uso de métodos numéricos e algoritmos para transformar os dados obtidos através de câmeras em uma nova representação da informação. A intenção é que as informações obtidas permitam a tomada de decisão com respeito a detecção ou o reconhecimento de objetos (BRADSKI; KAEHLE, 2008). O ciclo de processamento da visão computacional pode ser dividido nas seguintes etapas: aquisição, pré-processamento, segmentação, extração de características, reconhecimento e interpretação (MARQUES FILHO; VIEIRA NETO, 1999). A Figura 1 ilustra a sequência de processamento de um sistema de visão computacional.

Na Figura 1, a etapa de aquisição compreende a captação de imagens do ambiente através de câmeras. O pré-processamento, por sua vez, é realizado com intuito de retirar ruídos provenientes da etapa de aquisição através da aplicação de filtros nas imagens. A segmentação consiste em dividir a imagem em unidades significativas, ou seja, dividir a imagem em regiões ou objetos com intuito de simplificar ou alterar sua representação de acordo com o critério estabelecido. A extração de características visa extrair detalhes da imagem com a finalidade de descrever pontos de interesse comuns, isto é, pode ser definida

como sendo a captura das informações mais relevantes para gerar uma classificação de um dado fornecido como entrada (DEVIJVER; KITTLER, 1982, apud CRUZ; CAVALCANTI; REN, 2008). Por fim, os pontos de interesse são classificados em rótulos conhecidos através da interpretação.

Figura 1: Sequência de processamento de um sistema de visão computacional



Fonte: Marques Filho, Vieira Neto (1999).

Para Hayet, Lerasle e Devy (2007), a visão computacional tem se tornado o principal elemento na navegação de robôs porque tem apresentado resultados com alta precisão e propicia uma descrição detalhada do ambiente ao redor. O uso de técnicas de VC em problemas relacionados à navegação de robôs pode ser abstraído em duas etapas. Na primeira etapa, os procedimentos de VC supracitados na Figura 1 são executados para detectar, identificar e reconhecer características ou objetos específicos na imagem. Na segunda etapa é realizado uma tarefa específica para estimação da posição (localização), rastreamento, inferência de deslocamentos ou identificação de objetos e obstáculos (ROCHA; OLIVEIRA, 2010; HOLANDA *et al.*, 2010).

A tarefa de autolocalização em robôs móveis autônomos encontra alguns obstáculos, dentre eles o custo computacional, as variações do ambiente de trabalho e a precisão. O tempo de processamento é um fator importante no desenvolvimento de sistemas embarcados de visão computacional em robôs móveis que devem realizar tarefas em tempo real. No caso da

autolocalização, esta questão é fundamental porque o robô, idealmente, deve determinar sua posição com precisão em um intervalo de tempo limitado (RAHMAN *et al.*, 2008; LEÃO *et al.*, 2010).

2.2 CASAS DE VEGETAÇÃO E ROBÔS MÓVEIS AUTÔNOMOS

Para Cermeño (1990), uma casa de vegetação consiste em uma estrutura coberta e abrigada artificialmente por materiais transparentes, com intuito de proteger as plantas de agentes meteorológicos exteriores, permitindo o desenvolvimento das culturas nela abrigadas em todo seu ciclo vegetativo. Brito (2000) relata que, com a finalidade de atender às exigências ambientais dos vegetais e prover condições apropriadas para um melhor cultivo e desenvolvimento, surgiram, ao longo do tempo, casas de vegetação com soluções arquitetônicas diversas, cada uma para atender às condições climáticas da região. Tais casas de vegetação permitem hoje um controle parcial ou total das condições de exposição impostas pelos agentes ambientais. A título de exemplo, a Figura 2 mostra uma casa de vegetação localizada no Campus Uvaranas da Universidade Estadual de Ponta Grossa construída com finalidade de estudos científicos.

Figura 2: Exemplo de casa de vegetação



Fonte: Notícias UEPG⁴

⁴ Disponível em: <<http://portal.uepg.br/noticias.php?id=5527>>. Acesso em: abr. 2015

A aplicação de técnicas de automação e controle em casas de vegetação tem buscado adaptar-se às demandas e exigências do mercado (TERUEL, 2010). O desenvolvimento de sistemas de gerenciamento para casas de vegetação vem crescendo nas últimas décadas para as mais diversas situações, incorporando mudanças tecnológicas em todo o sistema, inclusive aspectos relacionados à sustentabilidade destas (MIRINEJAD *et al.*, 2008; PARK *et al.*, 2011; TERUEL, 2010). A automação destas tarefas reduz a necessidade de intervenção humana e consequentemente a exposição das pessoas à agentes e condições ambientais potencialmente nocivas.

A introdução de *agrobots* em casas de vegetação permite o desenvolvimento de sistemas autônomos capazes de realizar tarefas que usualmente exigem mão de obra humana. Isto possibilita redução de riscos e custos e, o aumento de precisão na execução das tarefas (VAN HENTEN *et al.*, 2002; SAMMONS; FURUKAWA; BULGIN, 2005; GONZÁLEZ *et al.*, 2009; SANCHEZ-HERMOSILLA *et al.*, 2013). Por exemplo, Sanchez-Hermosilla *et al.* (2013) desenvolveram um *agrobot* autônomo para pulverização de defensivos, poda e transporte de plantas. Sammons; Furukawa e Bulgin (2005) desenvolveram um *agrobot* autônomo para pulverizar pesticidas. Acaccia *et al.* (2003) propuseram um *agrobot* para inspeção, pulverização e tratamento de plantas em geral. Por sua vez, Van Henten *et al.* (2002) desenvolveram um *agrobot* para colheita de pepino.

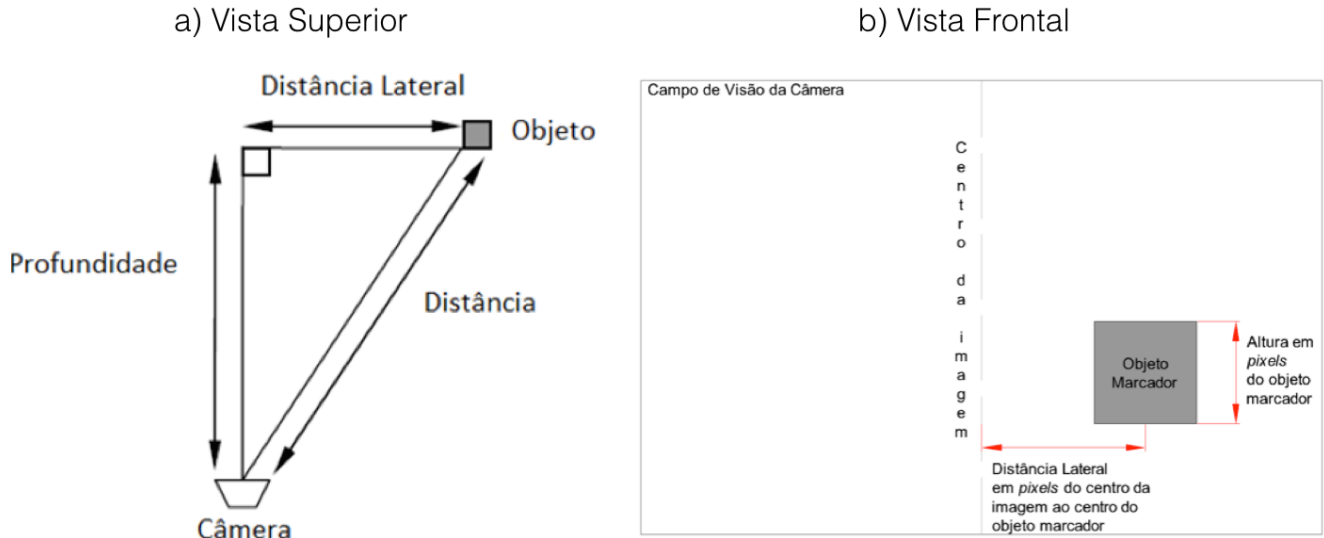
Neste cenário, a autolocalização de robôs pode ser útil, pois permite que o *agrobot* navegue no ambiente e tome decisões sem a necessidade de intervenção humana, da mesma forma que dispensa aparatos adicionais, tais como trilhos, faixas, esteiras, dentre outros.

2.3 ESTIMATIVA DE DISTÂNCIAS USANDO MÉTODOS DE PROCESSAMENTO DIGITAL DE IMAGENS

Rahman *et al.* (2008) desenvolveram um método para calcular a distância de um robô até objetos marcadores pela análise de imagens adquiridas por uma única câmera. A Figura 3a) demonstra os elementos do problema através de uma vista superior. O método proposto pelo autor consiste em medir a altura em *pixels* do objeto marcador e a distância do centro deste objeto ao centro da imagem em *pixels* para, posteriormente, obter a profundidade e a distância lateral deste ao centro de visão do robô. A Figura 3b) mostra a visão frontal da câmera no momento da aquisição da imagem, ilustrando a altura e a distância lateral em

pixels. Com a informação da distância lateral e da profundidade, é possível identificar a que distância o robô se encontra do objeto visualizado através de trigonometria.

Figura 3: Esquema de funcionamento do método de Rahman *et al.* (2008)



Fonte: O autor.

O método exige uma etapa inicial para treinamento do sistema. Para tanto, é necessário posicionar a câmera em frente ao objeto em n posições distintas e realizar aquisição de imagens do objeto marcador. Para cada uma das n posições devem ser medidas a profundidade e a distância lateral distintas ao objeto. Para cada imagem capturada, a altura e a distância lateral em *pixels* são medidas e acrescentadas aos registros da posição. Após a execução destes passos, os seguintes dados estarão disponíveis:

- a) profundidade em metros (denotado por δ);
- b) distância lateral em metros (denotado por λ);
- c) altura do objeto em *pixels* (denotado por ρ);
- d) distância lateral em *pixels* (denotado por β).

As leituras de altura obtidas são usadas para determinar uma equação polinomial de segundo grau que relaciona a altura em *pixels* do objeto com a profundidade da câmera ao mesmo, representada pela Equação 1. Nesta equação, as literais a , b e c são os coeficientes numéricos.

$$\delta = a*\rho^2 + b*\rho + c \quad (1)$$

Para determinar a distância lateral, conforme a Figura 3, primeiramente é preciso aplicar uma relação θ , representada pela Equação 2, entre a distância lateral β na imagem e a distância lateral λ entre o robô e o objeto, para cada profundidade encontrada. Esta relação permite que o método seja usado para identificar a distância lateral para cada profundidade em que o robô se encontrar, pois θ é praticamente invariante à δ , ou seja, independente da variação da profundidade δ , a variação de θ é muito próxima de uma constante.

$$\theta = \beta / \lambda \quad (2)$$

Uma vez determinados os valores da profundidade δ e da relação θ é possível obter uma equação polinomial de segundo grau (Equação 3) que relaciona a profundidade δ com a relação θ . Nesta equação d , e e f são coeficientes.

$$\theta = d*\delta^2 + e*\delta + f \quad (3)$$

Após o treinamento das equações polinomiais para cálculo da profundidade e da relação entre esta e a distância lateral é possível usar estas equações para obter a profundidade δ e a distância lateral λ , a partir dos dados de uma imagem. Para a profundidade, basta saber a altura do objeto marcador em *pixels* e substituir este valor na Equação 1. Desta maneira, obtém-se a profundidade em metros do robô ao objeto.

Para calcular o valor da distância lateral em metros λ , primeiramente é necessário substituir o valor medido anteriormente da altura em *pixels* na Equação 3 para obter o valor da relação θ . Posteriormente, deve-se coletar a distância lateral do centro do objeto marcador ao centro de imagem em *pixels* β e inverter a Equação 2 de modo a isolar a variável λ , como descrito na Equação 4. Por fim, deve-se substituir os valores de θ e β na Equação 4 através do resultado da Equação 3 e da leitura em *pixels* da distância lateral respectivamente. Deste modo, o valor λ poderá ser obtido.

$$\lambda = \theta / \beta \quad (4)$$

2.4 O ALGORITMO SURF

Para calcular a distância de objetos usando o método proposto por Rahman *et al.* (2008), é necessário empregar procedimentos para detectar e localizar em imagens a presença de objetos cujas dimensões sejam conhecidas. O algoritmo SURF provê uma abordagem para isto. Proposto por Bay; Tuytelaars e Van Gool (2006) e, inspirado no algoritmo SIFT⁵ (*Scale Invariant Feature*) desenvolvido por Lowe (1999), o algoritmo SURF é um detector e descritor de pontos de interesse em imagens. Entre as suas propriedades estão a invariância em relação à escala e à rotação.

O algoritmo SURF tem três etapas. Na primeira, ocorre a detecção de pontos de interesse, os quais são responsáveis por identificar características ou padrões específicos do objeto que se quer reconhecer, que são únicos, de modo a servir para posterior rastreamento e comparação. Na segunda etapa, é realizada a descrição destes pontos, ou seja, a vizinhança do ponto de interesse é registrada junto com ele, para que as características ao seu redor possam ser utilizadas para encontrá-lo em outras imagens. Na terceira etapa é realizada uma comparação da proximidade entre os pontos de interesse de duas imagens distintas, através da distância euclidiana, buscando um objeto ou marcador em comum (RIBANI *et al.*, 2013). A Figura 4a) mostra a comparação de proximidade entre os pontos de duas imagens através das linhas em azul entre estas, onde cada linha representa uma comparação. Já a Figura 4b) apresenta o marcador identificado na imagem capturada.

Figura 4: Exemplo de detecção de marcador através do algoritmo SURF



Fonte: O autor.

⁵ SIFT = Detector de Características Invariante à Escala.

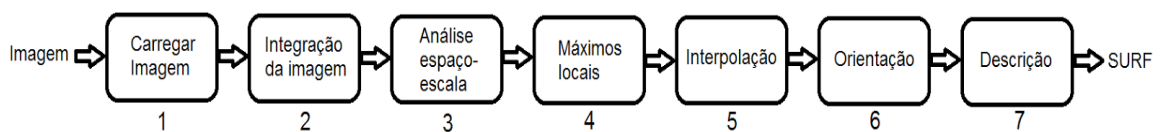
Os parâmetros de entrada do SURF são as imagens a serem confrontadas e suas resoluções e, um valor de limiar sobre uma matriz hessiana. Este valor de limiar permite limitar a quantidade de pontos de interesse a serem extraídos em cada imagem, de modo que quanto maior o seu valor, menor a quantidade destes pontos (PRATES *et al.*, 2012).

O algoritmo exige um banco de dados prévio com imagens dos objetos que se deseja encontrar. Assim, quando uma imagem é capturada, o SURF realiza a detecção e descrição dos pontos de interesse desta, compara com todo o seu banco de dados e informa a conformidade entre cada par de imagens. Neste procedimento, a conformidade entre duas imagens é definida como as distâncias euclidianas entre os pontos de interesse descritos, de modo que o conjunto escolhido será aquele que obtiver o maior número de menores distâncias euclidianas, conforme a Equação 5, onde d_1 representa o ponto interesse da imagem contida no banco de dados, α representa o percentual de conformidade desejado entre as imagens e d_2 representa o ponto de interesse da imagem capturada. A equação é aplicada para todas as combinações de ponto de interesse descritos entre o par de imagens.

$$d_1 = \alpha * d_2 \quad (5)$$

A Figura 5 ilustra o processamento do algoritmo SURF de maneira sequencial, onde (1) compreende a etapa de carregar a imagem, (2) compreende a integração da imagem, ou seja, a soma de todos os *pixels* dentro da região retangular da imagem, (3) realiza a análise espaço-escala, esta sendo a convolução ou superposição da imagem de entrada para cada escala utilizada no espaço-escala, obtendo-se a invariância a escala, (4) identifica máximos locais em todas as escalas, isto é, localiza os pontos de interesse, (5) realiza a interpolação, localizando as coordenadas x e y na imagem dentro de todo o espaço-escala, (6) compreende a atribuição de orientação para cada ponto de interesse, atribuindo invariância a rotação e, (7) realiza a descrição dos pontos de interesse.

Figura 5: Processamento do algoritmo SURF de maneira sequencial



Fonte: O autor.

As imagens do banco de dados devem ser objetos específicos que se diferenciam do ambiente e tem a função de facilitar sua descoberta pelo sistema (MORAIS, 2011; PROCHAZKA *et al.*, 2012). Para sua detecção com o SURF, o objeto necessita estar fixados no local de trabalho em uma posição conhecida.

2.5 COMPUTAÇÃO PARALELA E O ALGORITMO SURF

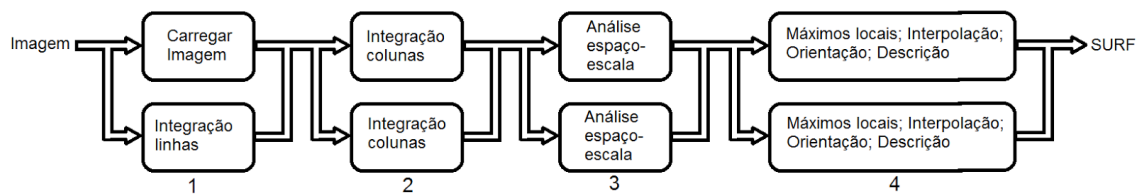
Como destacado anteriormente, os robôs móveis autônomos necessitam executar tarefas de navegação, tais como conhecer sua posição atual, planejar sua rota e se movimentar no ambiente (GOUVEIA, 2008; ZHOU; LIN, 2011). Um dos requisitos para realizar estas tarefas é o processamento em tempo real de maneira robusta e precisa. Por isso, a ideia básica é empregar algoritmos eficientes que explorem as características do problema e do hardware disponível com intuito de reduzir o tempo de computação (GOUVEIA, 2008; LU *et al.*, 2013). Uma abordagem para redução de tempo de processamento é através da computação paralela.

Conforme Almasi e Gottlieb (1989, apud Rodrigues, 2004), a computação paralela consiste em uma coleção de elementos de processamento que se comunicam e cooperam na resolução de problemas de forma rápida e eficiente. Neste contexto, a arquitetura CUDA é uma plataforma de computação paralela de que explora dispositivos compostos por GPUs para permitir a implementação de algoritmos paralelos (NVIDIA, 2015). Uma GPU consiste basicamente em um microprocessador que possui vários núcleos responsáveis pela aplicação de uma instrução para vários dados, permitindo que vários dados sejam processados em paralelo simultaneamente (MIRANDA, 2011).

Diversos pesquisadores e empresas vêm desenvolvendo soluções de propósito geral a partir de algoritmos paralelos através de computação paralela (ARGENTA *et al.*, 2010; LEÃO *et al.*, 2010; MIRANDA, 2011; NVIDIA, 2015). A ideia é aproveitar a alta capacidade de processamento, muitas vezes ociosa, dos múltiplos núcleos existentes no sistema para co-processamento de aplicações de uso geral, visto que a GPU possui um número muito maior de circuitos de processamento do que de armazenamento, aliando uma infinidade de *threads*, que a arquitetura permite executar simultaneamente (HAPP *et al.*, 2011; VIANA, 2012; NVIDIA, 2015).

O algoritmo SURF possui uma implementação paralela, chamada de P-SURF⁶ (*Parallel SURF*) (ZHANG, 2010). A Figura 6 demonstra o método de processamento do algoritmo P-SURF em relação a Figura 5, que demonstra a implementação sequencial, onde: (1) compreende a etapa de carregar e integrar as linhas da imagem, (2) compreende a integração das colunas da imagem, (3) realiza a análise espaço-escala e (4) processa máximos locais, interpolação, atribuição de orientação e descrição dos pontos de interesse.

Figura 6: Processamento do algoritmo P-SURF



Fonte: O autor.

⁶ P-SURF = SURF Paralelo.

3 METODOLOGIA

Este capítulo apresenta um procedimento que combina o algoritmo SURF para detecção de objetos e o método de Rahman *et al.* (2008) para cálculo de distância entre o objeto e o robô. Neste procedimento o algoritmo de Rahman *et al.* (2008) foi adaptado de maneira a permitir a estimativa da localização de um *agrobot* em uma casa de vegetação. São descritas duas realizações do procedimento, uma que utiliza processamento sequencial e outra que emprega processamento paralelo. Também são descritos dois experimentos realizados para obter características do procedimento proposto: um para medir o custo computacional do método proposto e outro para estimar a precisão na determinação das coordenadas do *agrobot*. Por fim, são descritos os procedimentos experimentais para os testes no ambiente experimental.

Este capítulo está organizado da seguinte maneira. A seção 3.1 descreve um procedimento para autolocalização de robôs através de processamento de imagens digitais e a seção 3.2 apresenta o protocolo experimental usado para a execução do procedimento.

3.1 PROCEDIMENTO PARA AUTOLOCALIZAÇÃO DE ROBÔS E MÉTODO PARA ESTIMATIVA DE DISTÂNCIAS EM IMAGENS DIGITAIS

O procedimento de autolocalização proposto neste trabalho utiliza imagens coletadas por uma única câmera acoplada a um robô para extrair as informações necessárias para estimar a posição do robô. A cada captura, o sistema procura e identifica o objeto marcador nesta através do algoritmo SURF. Posteriormente, é aplicado o método de Rahman *et al.* (2008) para encontrar a profundidade e distância lateral do robô ao objeto marcador. A partir destas informações, a extensão deste método proposta neste trabalho estima a posição do robô no ambiente de operação. O procedimento é realizado através de um sistema de processamento externo ao robô, executado de forma assíncrona em relação ao funcionamento deste.

O emprego do método proposto pode ser abstraído em duas etapas, sendo a primeira para treinamento dos parâmetros e a segunda para a execução do procedimento de autolocalização propriamente dito. A execução da autolocalização pode ser dividida em dois passos. No primeiro passo, o algoritmo SURF é usado para identificar o objeto marcador na

imagem capturada pelo robô. No segundo passo, o procedimento de Rahman *et al.* (2008) é empregado e adaptado para encontrar a coordenada do robô no ambiente.

3.1.1 Treinamento dos parâmetros do método de Rahman

A utilização do método de Rahman *et al.* (2008) em um determinado ambiente exige a determinação das constantes das Equações 1 e 3 com o objetivo de relacionar a altura e distância lateral em *pixels* com os seus valores em metros, citado na seção 2.3. O primeiro passo para isso é adquirir imagens do objeto marcador de interesse, sendo necessário coletar pelo menos seis imagens em profundidades e distâncias laterais distintas em relação ao objeto marcador, visto que as equações possuem ao todo seis constantes.

Este relacionamento entre as medidas em *pixel* e em metros pode ser obtido por análise de regressão a partir dos seguintes dados: a) altura em *pixels* do objeto marcador na imagem capturada pelo robô; b) distância lateral em *pixels* do centro do objeto marcador ao centro da imagem capturada pelo robô; c) profundidade em metros do objeto marcador à lente da câmera do robô; d) distância lateral em metros do objeto marcador ao centro da lente da câmera do robô. A rotina de cálculo para as grandezas mencionadas em metros está na seção 2.3, considerando o sistema já com os coeficientes.

3.1.2 Extração de características e detecção de objetos marcadores através do algoritmo SURF

No procedimento proposto, o algoritmo SURF é usado para detectar se algum objeto marcador está presente na imagem capturada pelo robô e, se for o caso, identificar qual é o objeto marcador detectado. Esse algoritmo fornece inúmeros pontos de interesse em cada imagem, muitos desses sendo irrelevantes e responsáveis por dados ruidosos que podem reduzir a precisão dos resultados esperados (OPENCV, 2013). Por isso, é necessário desenvolver um algoritmo que analisa somente as maiores semelhanças entre os pontos de interesse do objeto marcador e da imagem capturada. Neste algoritmo, em cada ponto de interesse comparado de cada imagem, deve ser analisada a distância euclidiana entre estes, de modo que, se um ponto de interesse de uma imagem possuir uma distância próxima ao da

outra imagem, este é considerado como uma maior semelhança. A Figura 7 apresenta o pseudocódigo deste algoritmo, baseado em um modelo disponibilizado pela biblioteca OpenCV.

Figura 7: Pseudocódigo para destacar as maiores semelhanças

```
Algoritmo para comparação de maiores semelhanças
ENTRADA: (pontosInteresseDescritosMarcador, pontosInteresseDescritosImagemCapturada)
{
    correspondências[] <- OPENCV.match(pontosInteresseDescritosMarcador,
    pontosInteresseDescritosImagemCapturada);
    j <- 1;
    PARA (i de 1 até correspondências.tamanho)
    {
        SE (correspondências[i].pontosInteresseMarcador.distEuclidiana < critérioMaiorSemelhança *
        correspondências[i].pontosInteresseImagemCapturada.distEuclidiana)
        {
            maiorSemelhança[j] <- correspondência[i];
            j++;
        }
    }
}
SAIDA: maiorSemelhança[]
```

Fonte: O Autor.

A próxima etapa consiste em verificar a quantidade de maiores semelhanças dentre todas as comparações realizadas, sendo estas feitas entre a imagem capturada pelo robô com cada imagem do objeto marcador pertencente ao banco de dados do sistema. As imagens que possuírem maiores semelhanças entre si devem ser consideradas como melhor análise, com um mínimo de quatro maiores semelhanças, sendo esta registrada e, as demais, descartadas, visto que quatro pontos é o mínimo necessário para encontrar a homografia⁷ do objeto (SANTOS, 2012). Caso nenhuma análise possua quatro ou mais maiores semelhanças, o sistema descarta o ponto de leitura e informa que não será possível utilizar a imagem capturada.

⁷ Homografia = transformação projetiva planar que mapeia pontos de um plano para outro plano.

Após selecionar o conjunto de pontos de interesse que apresentam maior similaridade com a imagem armazenada, é necessário usar um procedimento para traçar o contorno sobre a representação do objeto marcador na imagem. A rotina de detecção de contorno procede da seguinte forma, nesta ordem. Primeiro, ela executa a função *findHomography* (OPENCV, 2013) com intuito de identificar os cantos da imagem do objeto marcador na imagem capturada pelo robô. Em seguida, aplica a função *perspectiveTransform* para encontrar o objeto marcador na imagem capturada. Por fim, desenha o contorno do objeto marcador identificado através da função *line*.

A execução destas três etapas do algoritmo SURF depende da definição de dois parâmetros de entrada. Estes parâmetros consistem nas imagens e suas resoluções, bem como na determinação do valor de limiar para a matriz hessiana do algoritmo SURF.

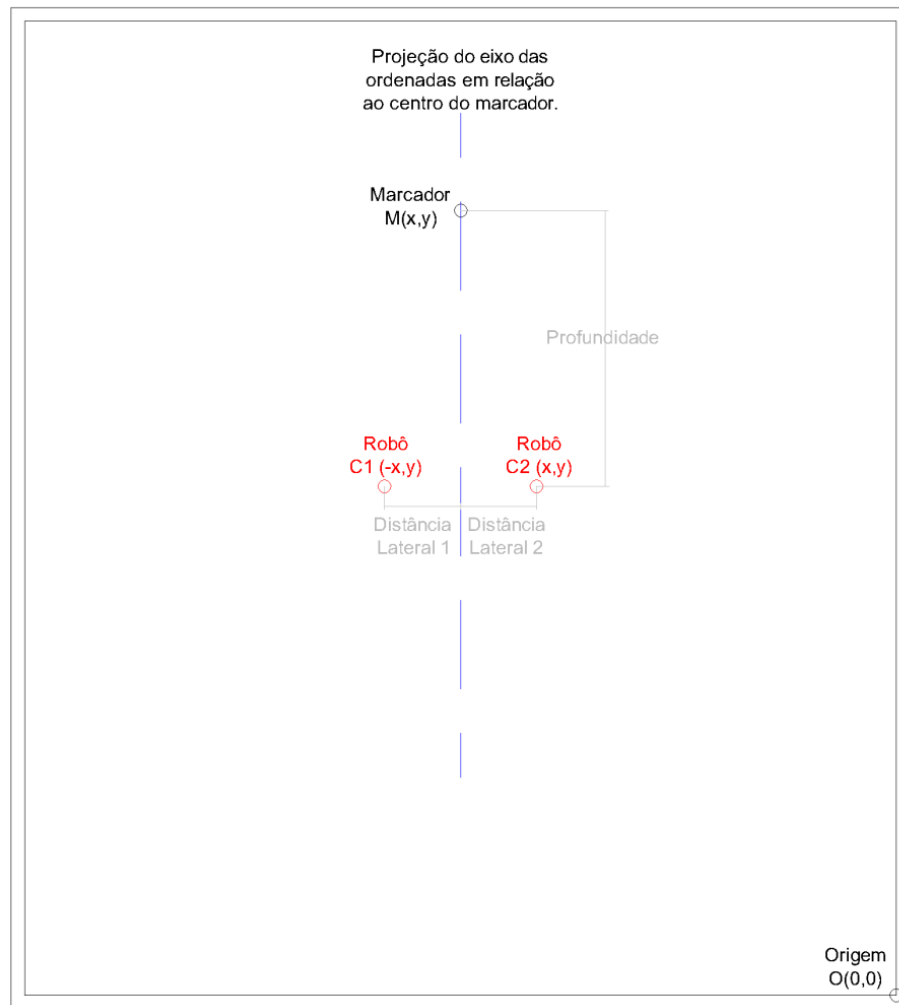
3.1.3 Determinação da coordenada do robô

Uma vez especificados os parâmetros do modelo que representa o objeto marcador, de acordo com o algoritmo SURF, é possível medir a altura e a distância lateral em *pixels* deste objeto.

O método de cálculo de distância proposto por Rahman *et al.* (2008) considera apenas um valor numérico absoluto de distância lateral, de modo a não ser possível identificar se o robô encontra-se à direita ou à esquerda do objeto marcador, ou seja, não informa em qual quadrante o robô se encontra em relação ao eixo das ordenadas relacionado ao centro do objeto marcador. Isto ocorre pois o objetivo do autor é encontrar a distância da câmera ao objeto ao invés da coordenada do robô naquele instante. A Figura 8 ilustra o problema em que o objeto marcador é representado por M e as duas possíveis distâncias laterais são representadas por C1 $(-x, y)$ e C2 (x, y) , deixando indefinido qual a posição correta do robô.

Uma maneira de determinar a coordenada correta é usar o valor numérico relativo fornecido pela Equação 3 para identificar se o robô está à esquerda ou à direita do objeto marcador, ou seja, se o valor numérico do resultado for negativo, o robô estará à esquerda do objeto marcador, caso seja positivo, estará à direita. Assim, um único valor para a abscissa da coordenada de sua posição ao invés de dois será considerado. Dessa forma, é possível obter o par da coordenada do robô em relação ao objeto marcador.

Figura 8: Esquema com duas possíveis distâncias laterais para o robô.



Fonte: O Autor.

3.2 PROTOCOLO EXPERIMENTAL

O desempenho do método de autolocalização proposto neste trabalho foi avaliado em três experimentos. Os dois primeiros foram realizados em um laboratório e avaliaram o desempenho do algoritmo em termos do erro nas estimativas de localização e do tempo de processamento das implementações sequencial e paralela. O terceiro experimento foi um teste de campo em uma casa de vegetação, onde foi avaliada a correção das estimativas para a localização.

A execução dos experimentos demandou a escolha de um ambiente e uma infraestrutura para sua realização. Também foi necessário definir alguns parâmetros do algoritmo SURF que depende das condições materiais e ambientais em que o teste é realizado.

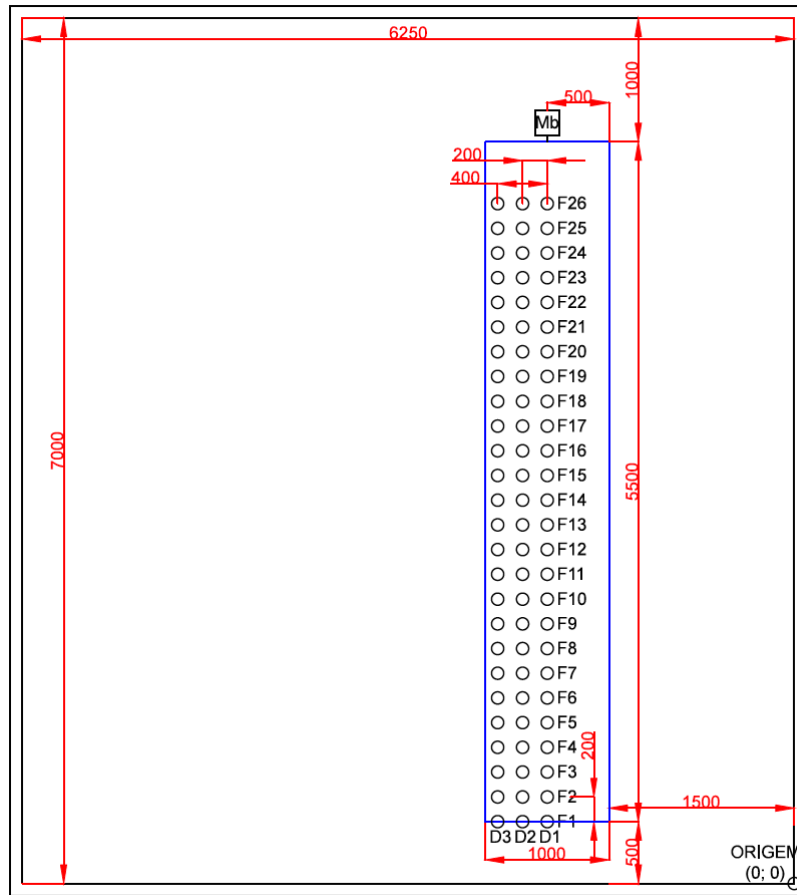
As seções 3.2.1 e 3.2.2 descrevem a infraestrutura e as condições materiais usadas durante o experimento. A seção 3.2.3 descreve o pré-processamento necessário para a execução dos testes. A Seção 3.2.4 descreve os experimentos realizados em laboratório. A Seção 3.2.5 descreve o teste de campo.

3.2.1 Infraestrutura utilizada na realização dos experimentos

Assumindo a localização de robôs em casas de vegetação como um problema similar ao da autolocalização de robôs em ambientes fechados, os testes de desempenho do procedimento proposto foram realizados em um dos laboratórios de pesquisa do Programa de Pós-Graduação em Computação Aplicada da UEPG. O laboratório escolhido, o Laboratório de Pós-Graduação, é uma sala de 6,25m por 7m, com iluminação artificial através de lâmpadas fluorescentes. A decisão de usar uma sala com iluminação artificial teve como objetivo reduzir a variação da iluminação e consequente na captura das imagens ao longo do tempo. Esta opção se justifica visto que os experimentos pretendiam avaliar o desempenho do procedimento de autolocalização e não dos métodos de visão computacional e, como observado por HOLANDA *et al.*, (2010), a redução da variação da iluminação no ambiente contribui para uma maior robustez na execução de métodos de detecção e reconhecimento de objetos em imagens.

Neste ambiente foi simulado um corredor, de modo similar ao que acontece em muitas casas de vegetação. O corredor simulado tem 1m de largura e 5,5m de comprimento. A Figura 9 ilustra a arquitetura para a simulação, onde os pontos $DmFn$ são pontos de leitura (m representa a variação da coordenada para distância lateral e n representa a variação da coordenada para profundidade), detalhados na Seção 3.2.4. O ponto de origem foi utilizado para comparação entre a coordenada real e a coordenada estimada neste estudo.

Figura 9: Dimensões do laboratório; Corredor simulado; Projeção dos pontos de leitura em milímetros

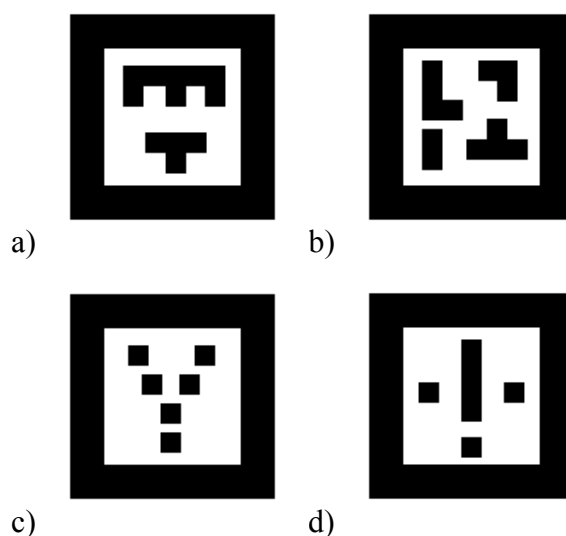


Fonte: O autor.

As imagens de quatro objetos marcadores foram coletadas e armazenadas em formato *Windows Bitmap* (BMP), compondo o banco de dados utilizado pelo algoritmo SURF. A Figura 10 mostra os objetos marcadores, sendo que a Figura 10_b) mostra o objeto marcador, denotado por M_b , que foi disposto no final do corredor conforme indicado no esquema da Figura 9. Este objeto marcador foi impresso em uma folha de papel A4 e suas dimensões são 17cm x 17cm. A Figura 11 mostra o objeto marcador disposto no final do corredor. Os objetos marcadores confeccionados foram armazenados no banco de dados com resolução igual a 160 x 160 *pixels*.

A definição de quatro objetos marcadores pretende simular uma situação real, em que o ambiente de trabalho tem mais de um corredor e mais de um objeto marcador. Neste caso, o procedimento de autolocalização deve identificar corretamente qual o objeto marcador detectado. A identificação incorreta de um objeto marcador fará com que o procedimento erre no cálculo da posição do robô, comprometendo sua navegação.

Figura 10: Marcadores digitais



Fonte: O autor.

Figura 11: Objeto marcador no final do corredor



Fonte: O autor.

Os algoritmos usados nos experimentos foram executados em um computador com processador Intel Core2Duo 2,4 GHz, memória RAM de 4GB, HD de 200GB, equipado com sistema operacional Windows 7 Ultimate SP1 64 bits e placa de vídeo NVIDIA GeForce 210. A implementação do algoritmo SURF, usado para descrição dos pontos de interesse dos objetos marcadores faz parte da biblioteca OpenCV 2.4.6 em linguagem JAVA⁸ (OPENCV, 2013). A biblioteca também implementa as funções que permitem identificar o contorno do objeto marcador na imagem após a descrição de seus pontos de interesse pelo algoritmo

⁸ Java é uma linguagem de programação e plataforma computacional lançada pela Sun Microsystems em 1995, atualmente de posse da Oracle.

SURF, conforme colocado na Seção 3.1.2. A plataforma de desenvolvimento utilizada foi a IDE Eclipse Kepler.

A biblioteca OpenCV dispõe também de uma implementação do algoritmo SURF paralelo (P-SURF), fornecendo as matrizes, métodos de chamada da GPU e retorno para CPU⁹ (*Central Processing Unit*), cálculos e armazenamento de memória. O P-SURF foi usado nos experimentos relacionados à implementação paralela do procedimento proposto neste trabalho.

3.2.2 Aquisição de imagens

As imagens foram adquiridas com a câmera de um iPhone 4S cuja resolução é de 8 *Mega Pixels* (MP), fornecendo imagens com 3264 x 2448 *pixels*. Durante a aquisição das imagens a altura da lente da câmera em relação ao piso foi fixada em 8cm. A escolha desta distância se deve ao fato de que esta é a altura padrão da câmera do robô ROVIO (2008), adquirido pelo Programa de Pós Graduação em Computação Aplicada à Agricultura da UEPG e que poderá ser utilizado na realização de testes futuros.

As imagens adquiridas foram armazenadas em uma base de dados cuja estrutura compreende a imagem capturada e a coordenada (x, y) do robô naquele instante de tempo. As imagens originais capturadas foram copiadas e redimensionadas para três outras resoluções através do software Microsoft Paint 6.1 (2009), sendo 2MP, 1,28MP e 0,72MP, de modo a reduzir a quantidade de pixels, mas representando a mesma imagem. Esta transformação foi realizada para fins de avaliação de precisão e custo computacional.

3.2.3 Calibração do sistema

Esta seção descreve os passos necessários para realizar a calibração, que consiste na definição dos parâmetros para extração de características e identificação do objeto marcador na imagem e, na definição dos parâmetros das Equações do método de Rahman *et al.* (2008).

Para a definição do limiar hessiano do algoritmo SURF, 23 valores para este foram avaliados com intuito de se obter uma faixa de trabalho em que o sistema alcance a

⁹ CPU = Unidade central de processamento.

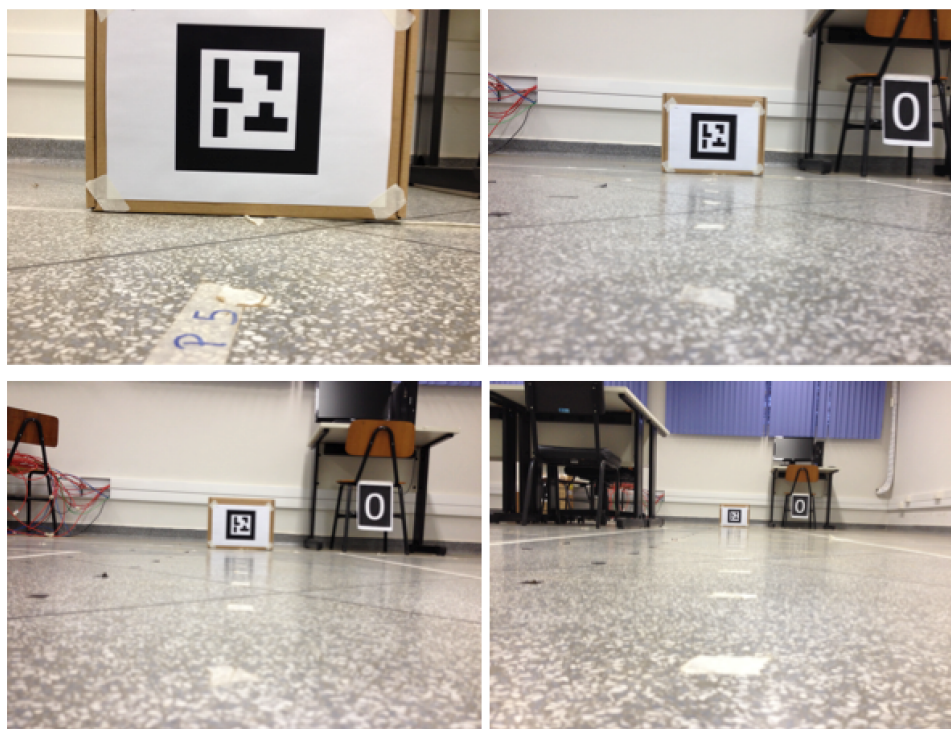
profundidade máxima esperada. Os valores testados foram: 100; 500; 1000; 1500; 2000; 3000; 4000; 5000; 6000; 7000; 8000; 9000; 10000; 11000; 12000; 13000; 14000; 15000; 16000; 17000; 18000; 19000; 20000. A profundidade máxima esperada nos testes foi de 5,5m para 2MP, visto que este foi o limite estabelecido pelo laboratório.

Para análise da maior semelhança entre os pontos de interesse de cada imagem, foram testados sete percentuais de proximidade entre as distâncias euclidianas de cada ponto de interesse. Os limiares de semelhança testados foram: 60%; 65%; 70%; 75%; 80%; 85%; 90%; 95%. O valor de limiar afeta a distância euclidiana mínima entre os pontos de interesse analisados, de modo que limiares pequenos exigem que os pontos de interesse comparados estejam muito próximos, causando um índice de rejeição alto (OPENCV, 2013). Por isso, é necessário definir um limiar mínimo. Neste trabalho, o limiar mínimo foi definido experimentalmente em um nível de 60%.

Para a definição das constantes das Equações 1 e 3, em cada imagem coletada, foi realizada uma contagem manual, através do *software* Microsoft Paint 6.1 (2009), da altura e da distância lateral em *pixels* do objeto marcador. Foram obtidas 78 imagens para profundidade e 52 para distância lateral. Esta diferença no número de análises se deve ao fato de que 26 imagens estão centralizadas com o objeto marcador, ou seja, a distância lateral é zero.

As coordenadas do ponto de leitura e do objeto marcador no ambiente são conhecidas, logo, a profundidade e a distância lateral em metros no ambiente também. Assim, a partir desses dados, foi possível obter as constantes das Equações 1 e 3. Para a regressão dos dados foi utilizado o *software* Microsoft Excel 2007. A Figura 12 mostra quatro imagens capturadas nas profundidades de 0,5m; 2,5m; 4,1m e 5,5m respectivamente, com distância lateral nula.

Figura 12: Exemplos de imagens de marcadores capturadas com variação de profundidade



Fonte: O Autor.

3.2.4 Experimentos de autolocalização

Nesta seção, o procedimento de autolocalização descrito na Seção 3.1 foi avaliado na determinação da localização de um robô em laboratório quanto à precisão.

Foram definidos 26 pontos de leitura sequenciais no corredor, alinhados com o centro do objeto marcador, com 0,2m de distância entre cada ponto de leitura, de modo que o ponto de leitura mais distante está a 5,5m do objeto marcador e o mais próximo a 0,5m deste. A profundidade máxima foi limitada à 5,5m devido às dimensões do laboratório de testes. Da mesma forma, duas linhas adicionais de pontos de leituras foram estabelecidas com intuito de validar a implementação do cálculo de distância lateral, proposto por Rahman *et al.* (2008). A primeira linha foi deslocada 0,2m da linha central e, a segunda, deslocada 0,4m da mesma linha central, totalizando 78 imagens (Figura 9), onde F_n representa a profundidade e D_m representa a distância lateral de cada ponto de leitura.

Para cada imagem coletada, o procedimento descrito na Seção 3.1 foi executado. Com isto, as coordenadas do robô em cada ponto foi estimada pelo método proposto. Da mesma forma, o tempo de processamento médio do sistema foi registrado para as implementações sequencial e paralela, levando em consideração todas as imagens capturadas pela câmera.

3.2.4.1 Análise dos resultados

Esta seção descreve o processo de avaliação da precisão do procedimento na estimativa das coordenadas do robô.

Para cada uma das imagens em que o objeto marcador foi detectado e o procedimento descrito na Seção 3.1 foi executado, calculou-se o erro absoluto da coordenada estimada em relação à coordenada real do ponto em que a imagem foi obtida. A partir desses dados, os valores do maior erro, do menor erro e da média do erro obtidos para cada uma das resoluções foram analisados para determinar se a resolução da imagem afeta o erro do sistema.

Também foi calculado o erro nas estimativas da profundidade e da distância lateral em função da resolução. Novamente, foram computados o maior erro, o menor erro e a média do erro para cada resolução.

Após uma análise descritiva do erro, foram obtidas correlações lineares com o objetivo de verificar o relacionamento entre os elementos:

- a) Erro na estimativa das profundidades *versus* erro na estimativa das coordenadas;
- b) Erro na estimativa das distâncias laterais *versus* erro na estimativa das coordenadas;
- c) Profundidade real *versus* erro na estimativa das coordenadas;
- d) Profundidade real *versus* erro na estimativa das profundidades.

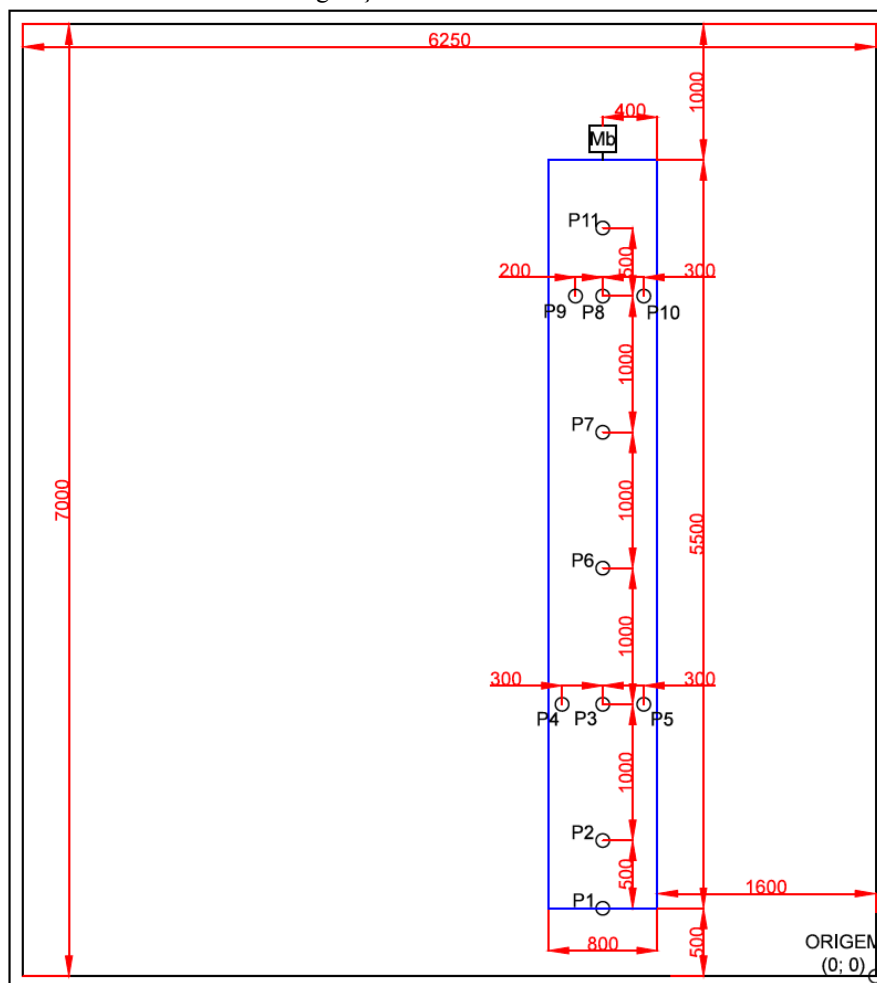
Os dados referentes aos tempos de processamentos foram utilizados para realizar uma análise de custo computacional. Nesta análise, o tempo de processamento para a implementação sequencial foi determinado para diferentes valores do parâmetro limiar hessiano do SURF, em cada uma das resoluções. O objetivo foi comparar o comportamento deste algoritmo para diferentes configurações de execução e, como a resolução afeta o desempenho do procedimento. Os tempos de computação das implementações sequencial e paralela também foram comparadas.

3.2.5 Experimentos de teste em campo

Para realização dos testes de campo foi utilizada a Casa de Vegetação 1 da Universidade Estadual de Ponta Grossa, esta possuindo as paredes e teto de plástico fosco, com iluminação natural. O propósito deste experimento foi realizar a correção das estimativas em relação a precisão da localização em uma situação real.

A ideia básica foi reproduzir o experimento realizado em laboratório na casa de vegetação, de modo que um corredor de 5,5m comprimento foi estabelecido, mas, com largura de 0,8m, devido a limitações do local. A Figura 13 exemplifica o corredor elaborado na casa de vegetação. Foram coletadas onze imagens em pontos distintos conforme pode ser verificado na figura. A partir dessas imagens, foi calculado o erro das estimativas para as coordenada de cada imagem capturada. O maior erro, o menor erro e a média do erro foram usados para avaliação.

Figura 13: Corredor simulado na casa de vegetação



Fonte: O Autor.

4 RESULTADOS E DISCUSSÃO

Esta seção apresenta e discute os resultados obtidos nos experimentos realizados para testar o procedimento de localização descrito neste trabalho. A Seção 4.1 apresenta os resultados da etapa de pré-processamento: treinamento do método proposto por Rahman *et al.* (2008) e configuração dos parâmetros de entrada do algoritmo SURF. A Seção 4.2 mostra os resultados referentes aos experimentos que avaliaram o erro do procedimento proposto na determinação da localização do *agrobot*. A Seção 4.3 apresenta os resultados dos experimentos que compararam o desempenho das versões sequencial e paralela do procedimento de autolocalização. A Seção 4.4 mostra os resultados em campo, na casa de vegetação. Por fim, a Seção 4.5 apresenta a discussão sobre os resultados.

4.1 RESULTADO DO PRÉ-PROCESSAMENTO

4.1.1 Treinamento dos parâmetros do algoritmo SURF

Como descrito na Seção 3.2.3 da Metodologia, o primeiro passo da etapa do pré-processamento foi determinar o limiar hessiano e o limiar de comparação usado pelo algoritmo SURF para quantificar a similaridade entre um padrão armazenado e o objeto marcador observado na cena. A Tabela 1 mostra estes resultados. Nesta tabela, o eixo horizontal indica os valores do percentual de comparação utilizados e o eixo vertical representa os valores de limiar hessiano avaliados. Cada par i, j informa a profundidade máxima entre o objeto marcador e a câmera na qual o sistema foi capaz de identificar o objeto. Os dados da Tabela 1 foram coletados usando uma resolução de 2MP.

Tabela 1: Resultados da profundidade obtidos em relação aos limiares de comparação e hessiano com imagens de 2MP.

limiar comparação / limiar hessiano	60%	65%	70%	75%	80%	85%	90%	95%
500	5,3	5,3	5,3	5,3	5,5	5,5	4,5	2,9
1000	4,7	5,5	5,5	5,5	5,5	5,5	5,3	5,5
1500	4,5	5,3	5,3	3,9	5,5	5,5	5,3	0,9
2000	5,3	5,5	5,5	3,9	5,5	5,5	5,3	2,3
3000	5,3	5,3	5,5	3,9	5,5	5,5	5,3	3,3
4000	4,9	5,3	5,3	3,9	5,5	5,5	5,3	2,9
5000	5,3	5,5	5,3	5,5	5,5	5,5	5,3	2,3
6000	4,9	5,3	5,3	5,5	5,5	5,5	4,9	2,9
7000	5,3	5,3	5,3	5,5	5,5	5,5	5,3	2,9
8000	3,9	5,3	5,3	5,5	5,5	5,5	4,7	2,3
9000	4,9	4,9	5,3	5,5	5,5	5,5	4,9	2,9
10000	4,9	4,9	5,3	5,5	5,5	5,5	5,5	3,9
11000	3,9	3,9	5,5	5,5	5,5	5,5	5,3	3,7
12000	4,1	5,3	5,5	5,5	5,5	5,5	5,5	0,9
13000	5,3	5,3	5,5	5,5	5,5	5,5	4,9	3,9
14000	5,3	5,3	4,9	5,5	4,7	5,5	4,7	3,9
15000	5,3	5,3	4,9	5,5	4,7	5,5	4,7	3,9
16000	5,3	5,3	4,9	4,9	5,5	5,5	4,7	3,9
17000	5,3	3,9	4,9	4,9	5,5	5,5	4,7	4,1
18000	0,9	0,9	4,9	5,3	5,5	5,5	4,7	2,9
19000	5,3	3,9	4,9	5,3	5,5	5,5	4,5	4,5
20000	3,3	3,9	5,3	5,3	4,5	3,9	4,5	2,3

Fonte: O Autor.

Os resultados da tabela indicam que a profundidade máxima de 5,5 metros é atingida, para todos os limiares hessianos do SURF na faixa de 500 a 19.000, quando o limiar de comparação α é igual a 85%.

Esta operação foi repetida para imagens com resolução de 1,28MP e 0,72MP, porém fixando $\alpha = 85\%$. Os resultados mostram que o valor $\alpha = 85\%$ permitiu a identificação dos objetos marcadores a uma distância de 4,7m para a resolução de 1,28MP na faixa de 500 a 18000 e a uma distância de 3,5m para a resolução de 0,72MP na faixa de 500 a 18000. Estes resultados estão listados na Tabela 2. Nesta, a primeira coluna indica o limiar hessiano, a segunda coluna mostra a profundidade obtida para a resolução de 0,72MP e a terceira apresenta a profundidade obtida para a resolução de 1,28MP.

Tabela 2: Resultados da profundidade obtida em relação aos limiares hessianos e o limiar de comparação de 0,85 nas resoluções de 0,72MP e 1,28MP.

Limiar hessiano	Profundidade 0,72 Mega <i>Pixels</i>	Profundidade 1,28 Mega <i>Pixels</i>
500	3,5	4,7
1000	3,5	4,7
1500	3,5	4,7
2000	3,5	4,7
3000	3,5	4,7
4000	3,5	4,7
5000	3,5	4,7
6000	3,5	4,7
7000	3,5	4,7
8000	3,5	4,7
9000	3,5	4,7
10000	3,5	4,7
11000	3,5	4,7
12000	3,5	4,7
13000	3,5	4,7
14000	3,5	4,7
15000	3,5	4,7
16000	3,5	4,7
17000	3,5	4,7
18000	3,5	4,7
19000	2,7	3,5
20000	2,1	3,5

Fonte: O Autor.

4.1.2 Obtenção de equações para profundidade e distância lateral

Os valores obtidos para altura em *pixels* e para profundidade em metros de 26 imagens sem distância lateral estão listados na Tabela 3. Nesta tabela, a primeira e a terceira colunas indicam a profundidade entre a câmera e o objeto marcador e, a segunda e a quarta, representam a altura em *pixels* do objeto marcador na imagem capturada. Os dados da tabela foram obtidos para imagens de 2MP.

Tabela 3: Leitura manual da profundidade e leitura em *pixels* do objeto marcador em imagens de 2MP.

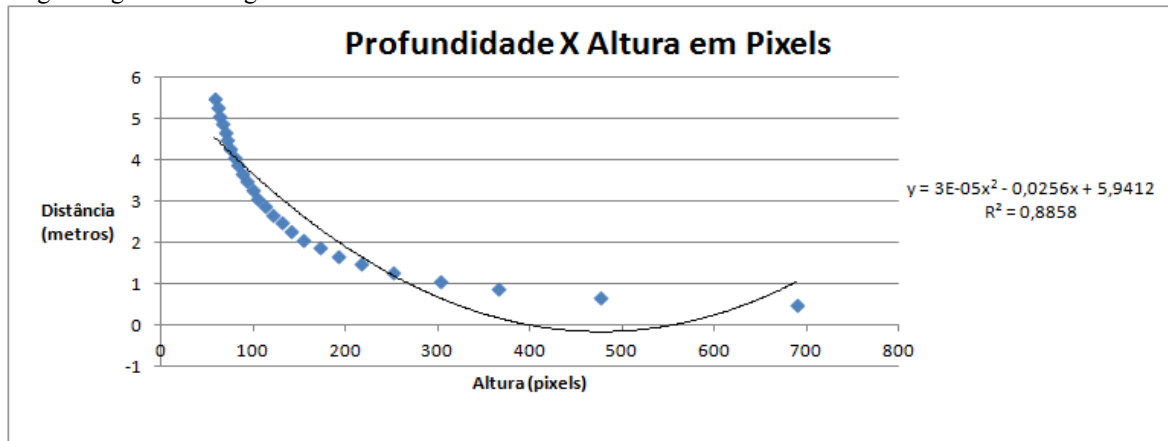
Profundidade	Altura em <i>Pixels</i>	Profundidade	Altura em <i>Pixels</i>
5,5m	58	2,9m	112
5,3m	61	2,7m	120
5,1m	63	2,5m	130
4,9m	66	2,3m	139
4,7m	69	2,1m	154
4,5m	71	1,9m	171
4,3m	74	1,7m	191
4,1m	79	1,5m	216
3,9m	83	1,3m	251
3,7m	87	1,1m	302
3,5m	93	0,9m	365
3,3m	98	0,7m	476
3,1m	104	0,5m	689

Fonte: O Autor.

Rahman *et al.* (2008) propõem que a relação entre a profundidade δ e a altura em *pixels* ρ seja estimada com o emprego de um procedimento de ajuste de funções para determinar os parâmetros da Equação 1 a partir de dados coletados. O resultado deste ajuste, sobre os dados da Tabela 3, é apresentado na Equação 6 e pode ser observado graficamente na Figura 14. O coeficiente de determinação (R^2) desse ajuste foi de 0,8858 e o erro máximo foi de 2,037m para profundidade para os valores residuais. Os dados numéricos do ajuste podem ser consultados no Apêndice A.

$$\delta = 3 * 10^{-5} \rho^2 - 0,0256\rho + 5,9412 \quad (6)$$

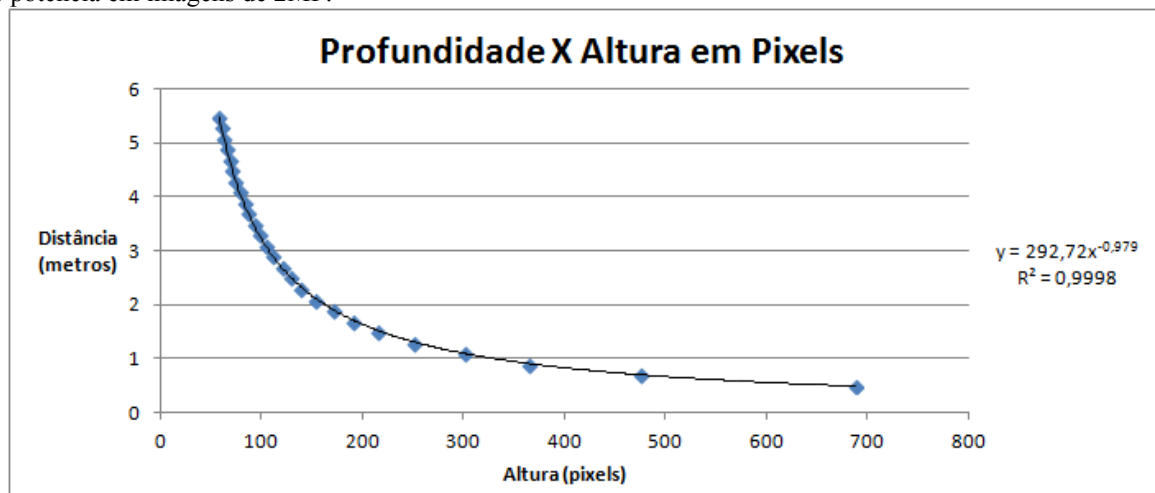
Figura 14: Gráfico relacionando profundidade em metros e altura em *pixels* com ajuste de função para equação de segundo grau em imagens de 2MP.



Fonte: O Autor.

Após uma análise visual dos dados e com o objetivo de reduzir o erro máximo do ajuste, foi efetuada uma regressão da função potência sobre os dados. O resultado desta nova regressão é apresentado na Equação 7 com R^2 igual a 0,9998 e com um erro máximo de 0,057m para profundidade para os valores residuais. A sobreposição da função ajustada aos dados experimentais pode ser observada graficamente na Figura 15. Os dados dos resíduos se encontram no Apêndice B.

Figura 15: Gráfico relacionando profundidade em metros e altura em *pixels* com ajuste de função para equação de potência em imagens de 2MP.



Fonte: O Autor.

$$\delta = 292,72\rho^{-0,979} \quad (7)$$

A Tabela 4¹⁰ apresenta os valores obtidos para cálculo da relação θ entre a profundidade δ e a distância lateral β . Nesta tabela, a primeira e a quinta colunas mostram as profundidades utilizadas, a segunda e a sexta colunas apresentam a distância lateral em *pixels* lida manualmente para imagens capturadas com distância lateral de 0,2m. A terceira e a sétima colunas apresentam a distância lateral em *pixels* lida manualmente para imagens capturadas com distância lateral de 0,4m e, a quarta e a oitava colunas mostram os valores obtidos da relação θ média entre os valores da relação θ para as distâncias laterais de 0,2m e 0,4m, conforme a Equação 2.

Tabela 4: Valores coletados manualmente para obtenção da equação para cálculo de distância lateral em imagens de 2MP

Profundidade	Distância Lateral 0,2m (<i>pixels</i>)	Distância Lateral 0,4m (Em <i>pixels</i>)	Relação (valor médio)	Profundidade	Distância Lateral 0,2m (<i>pixels</i>)	Distância Lateral 0,4m (<i>pixels</i>)	Relação (valor médio)
5,5m	56	117	285,63	2,9m	109	224	552,5
5,3m	58	121	295,63	2,7m	118	239	592,5
5,1m	61	126	309,38	2,5m	128	256	638,75
4,9m	63	130	318,75	2,3m	144	281	709,38
4,7m	66	137	335	2,1m	154	314	776,25
4,5m	69	145	351,88	1,9m	173	341	858,75
4,3m	73	141	369,38	1,7m	192	387	963,13
4,1m	76	159	388,75	1,5m	222	451	1116,88
3,9m	80	165	404,38	1,3m	257	533	1308,13
3,7m	84	174	426,88	1,1m	315	654	1603,13
3,5m	90	186	455,63	0,9m	369	746	1853,13
3,3m	97	197	488,75	0,7m	483	0	2412,5
3,1m	103	213	523,13	0,5m	699	0	3492,5

Fonte: O Autor.

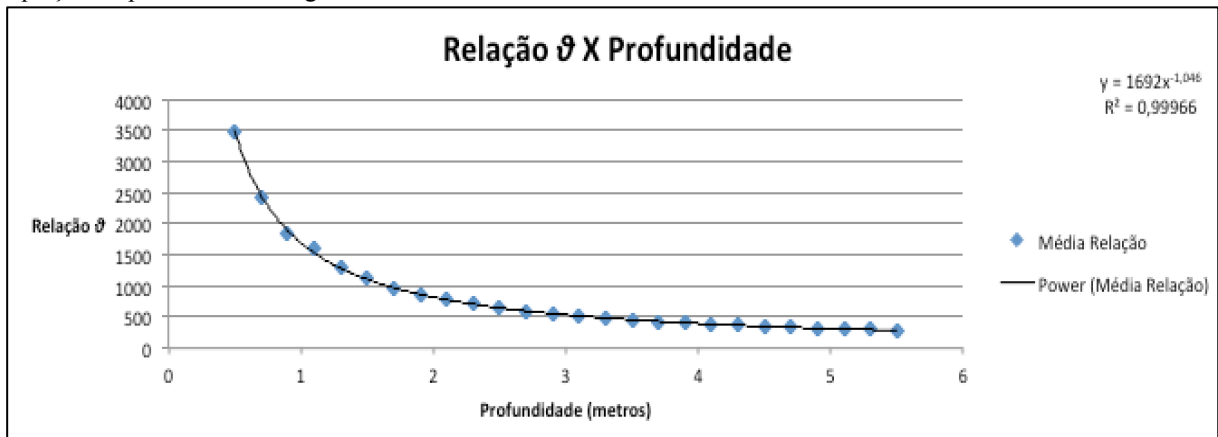
Com os valores médios da relação θ entre a profundidade e a distância lateral estabelecidos, foi efetuada uma regressão da função potência sobre os dados da Tabela 4. O resultado da regressão está apresentado na Equação 8, com R^2 de 0,9997 e um erro máximo de 71,672 para os valores residuais da relação θ . A sobreposição da função ajustada aos dados experimentais pode ser vista na Figura 16. Os resíduos estão listados no Apêndice C. Mais

¹⁰ Para a distância lateral 0,4m e profundidades de 0,5m e 0,7m não foi possível estabelecer a relação, pois o objeto marcador não ficou completamente visível na imagem capturada. Nestes casos, foi considerado apenas o valor da distância lateral 0,2m para o cálculo da relação de cada profundidade.

uma vez a seleção de uma função potência se deve ao fato de que o erro observado no uso de uma regressão quadrática foi superior àquele encontrado com a função potência¹¹.

$$\theta = 1692\delta^{-1,046} \quad (8)$$

Figura 16: Gráfico comparando a relação θ média com a profundidade em metros com ajuste de função para equação de potência em imagens de 2MP.



Fonte: O Autor.

O procedimento foi repetido para as imagens nas resoluções de 1,28MP e 0,72MP, onde as Equações de profundidade 9 e 10 e, de relação 11 e 12, foram encontradas respectivamente às resoluções para a regressão potência. Os resultados detalhados podem ser encontrados no Apêndice E. É possível observar que as equações geradas variam apenas em sua constante proporcional para cada resolução, tanto para profundidade δ quanto para a relação θ . Os resultados das regressões das Equações 9 e 10 apresentaram os mesmos valores de R^2 e consequentemente os mesmos erros máximos para os valores residuais de profundidade que a regressão para a Equação 7. O mesmo comportamento ocorreu para as Equações 11 e 12 em relação aos resultados da regressão para a Equação 8.

$$\delta = 235,27\rho^{-0,979} \quad (9)$$

$$\delta = 177,52\rho^{-0,979} \quad (10)$$

11 A Equação obtida foi $\theta = 181,28\delta^2 - 1486,3\delta + 3259$, com $R^2 = 88,21\%$ e erro máximo de 931,33 no valor θ para os valores residuais. Os resíduos estão listados no Apêndice D.

$$\theta = 1353,6\delta^{-1,046} \quad (11)$$

$$\theta = 1015,2\delta^{-1,046} \quad (12)$$

4.2 RESULTADOS DE LOCALIZAÇÃO

4.2.1 Obtenção das coordenadas do robô no ambiente

O procedimento proposto na Seção 3.1 foi utilizado para calcular as coordenadas de aquisição de cada uma das 75 imagens, com resolução de 2MP. Os resultados estão listados no Apêndice F.

Em relação às estimativas das coordenadas, o maior erro aferido foi de 0,402m para o ponto D2F2, de coordenadas (5,3m; 0,2m) em relação ao centro do objeto marcador. O menor erro foi de 0,006m nos pontos D1F25 e D1F26, de coordenadas (0,7m; 0m) e (0,5m; 0m) respectivamente, em relação ao centro do objeto marcador. A média do erro foi 0,11m em relação às coordenadas reais do robô.

No que se refere às estimativas da profundidade e da distância lateral, os resultados para os valores da média, do máximo e do mínimo do erro, bem como os pontos relacionados, são informados na Tabela 5. Nessa tabela é possível observar que o maior e o menor valor para os erros na estimativa da profundidade coincidem com os pontos de maior e menor erro para a estimativa das coordenadas.

Tabela 5: Média, desvio, máximo e mínimo do erro para as estimativas da profundidade e distância lateral em imagens de 2MP

	Profundidade		Distância Lateral	
	Erro da estimativa (m)	Ponto ID (coordenada)	Erro da estimativa (m)	Ponto ID (coordenada)
Média	0,084	-	0,052	-
Desvio Padrão	0,08	-	0,05	-
Mínimo	0	D1F25 (0,7; 0)	0	D3F6 (4,5; 0,4)
Máximo	0,37	D2F2 (5,3; 0,2)	0,264	D3F20 (1,7; 0,4)

Fonte: O Autor.

Para as 63 imagens com resolução de 1,28MP, os resultados da localização para cada uma das coordenadas de teste estão listadas no Apêndice G. Como pode ser observado a partir da tabela apresentada nesse apêndice, a média do erro das coordenadas foi 0,119m. O erro máximo foi de 0,389m, no ponto D3F9, e o menor erro encontrado foi de 0,007m, no ponto D1F26. Nestes testes a profundidade máxima dos pontos de leitura era de 4,7m.

A média do erro para a estimativa da profundidade foi de 0,1m. Mais uma vez o erro máximo da profundidade, 0,385m, ocorreu no mesmo ponto em que o erro das coordenadas foi máximo. O erro mínimo foi de 0,001m. Para a distância lateral, a média do erro mensurado foi de 0,05m, sendo o erro máximo de 0,146m identificado no ponto D2F5 e o erro mínimo de 0,001m, obtido no ponto D2F14.

Na resolução de 0,72MP, obteve-se resultados para 45 imagens com profundidade máxima de 3,5m. A média do erro das coordenadas estimadas foi de 0,092m, o erro máximo foi 0,269m (Ponto D2F13) e o menor erro foi 0,008m (ponto D1F25).

Já para profundidade, a média do erro foi de 0,072m, sendo o erro máximo de 0,268m (ponto D2F13) e o erro mínimo de 0,004m (ponto D1F25). Para a distância lateral, a média do erro mensurado foi de 0,044m, sendo o erro máximo de 0,116m (ponto D3F16) e o erro mínimo de 0,001m (pontos D2F14 e D2F21). Outra vez, o erro máximo para profundidade coincidiu com o erro máximo da coordenada. Estes resultados estão descritos no Apêndice H.

4.3 AVALIAÇÃO DO CUSTO COMPUTACIONAL

Para avaliar o custo computacional da solução sequencial proposta, foi aferido o tempo de processamento para as mesmas 75 imagens já mencionadas em cada um dos limiares hessianos, para todas as três resoluções empregadas. Os resultados obtidos podem ser visualizados na Tabela 6. A primeira coluna indica o limiar hessiano, a segunda coluna mostra a média do tempo de processamento para imagens de 2MP em cada limiar, a terceira coluna apresenta os resultados para 1,28MP e, a quarta coluna mostra os resultados para imagens de 0,72MP.

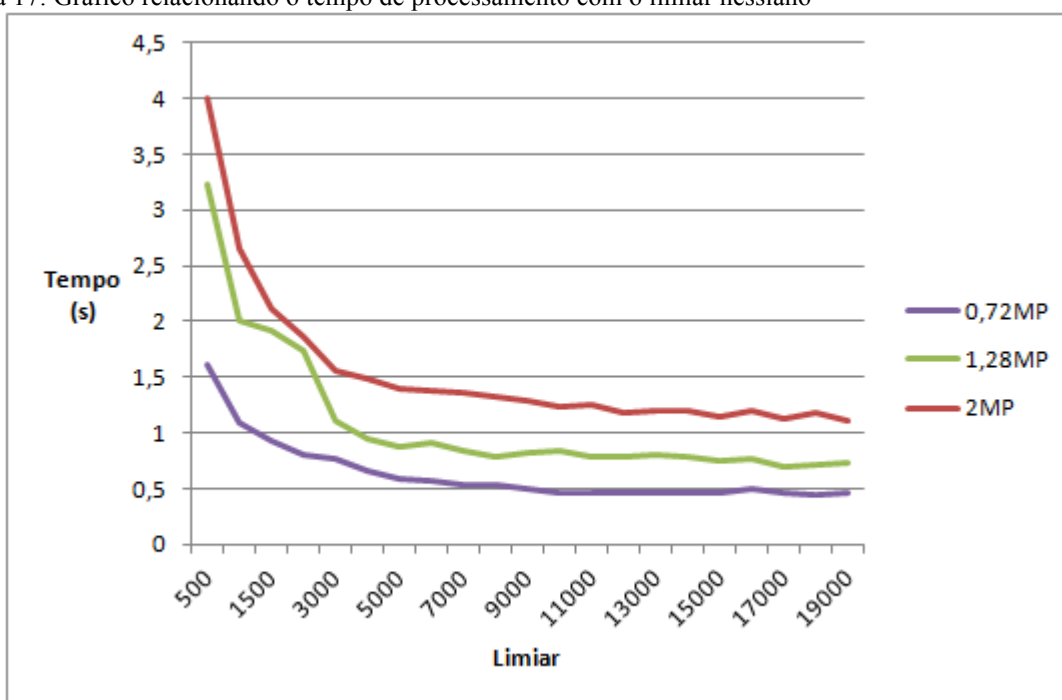
Tabela 6: Relação entre o custo computacional de cada resolução com o limiar hessiano em segundos

Limiar	2MP (s)	1,28MP (s)	0,72MP (s)
500	4	3,23	1,62
1000	2,66	2,01	1,09
1500	2,11	1,91	0,93
2000	1,86	1,73	0,81
3000	1,55	1,11	0,77
4000	1,49	0,94	0,66
5000	1,39	0,88	0,59
6000	1,38	0,91	0,57
7000	1,36	0,84	0,53
8000	1,32	0,79	0,53
9000	1,28	0,83	0,5
10000	1,24	0,84	0,47
11000	1,26	0,78	0,47
12000	1,18	0,79	0,47
13000	1,2	0,81	0,47
14000	1,19	0,78	0,46
15000	1,15	0,75	0,46
16000	1,19	0,76	0,49
17000	1,13	0,7	0,47
18000	1,18	0,71	0,44
19000	1,1	0,74	0,46

Fonte: O autor.

Como pode ser observado, em todas as resoluções, o tempo de computação tende a decrescer com o valor do limiar (Figura 17). Nesta figura, é possível notar que a curva gerada através das imagens com resolução de 2MP possui um custo de tempo maior que as demais curvas de 1,28MP e 0,72MP. Por sua vez, o comportamento observado para a resolução de 2MP se repete para a resolução de 1,28MP, esta possuindo um custo de tempo superior ao da resolução de 0,72MP para todos os valores de limiar.

Figura 17: Gráfico relacionando o tempo de processamento com o limiar hessiano



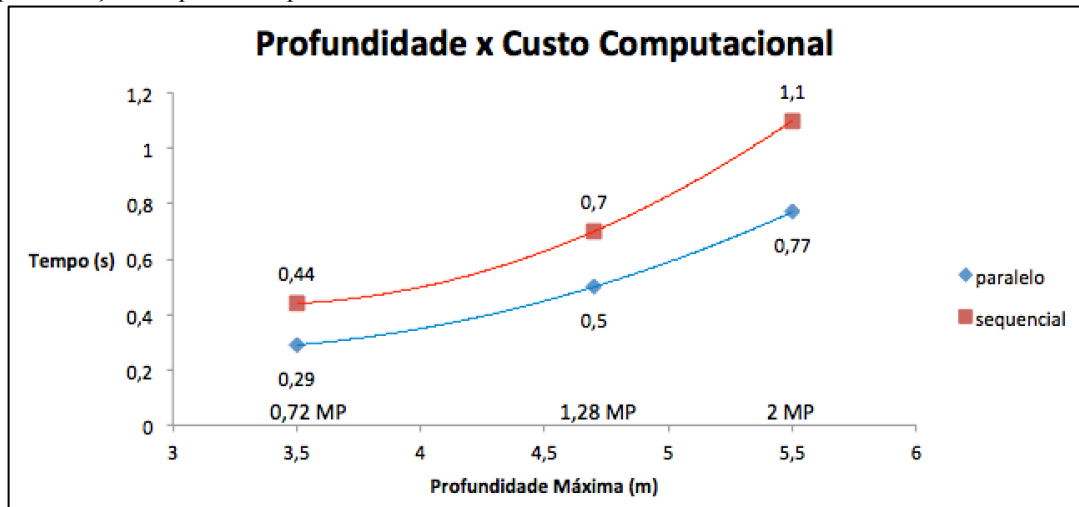
Fonte: O autor.

4.3.1 Utilização de algoritmo paralelo para redução do custo computacional

Para avaliar o impacto no custo computacional da solução, o algoritmo P-SURF foi testado com 25 imagens de cada resolução, computando o custo médio de cada análise. A ideia básica foi manter a profundidade máxima alcançada em cada resolução testada e usar o P-SURF para reduzir o tempo de computação.

Na implementação sequencial, o tempo de computação foi de 1,1s com imagens de 2MP, enquanto que na versão paralela o tempo foi de 0,77s, ou seja, uma redução de 30%. Nas imagens com resolução de 1,28MP, a proposta sequencial consumiu 0,7s para o processamento do SURF, contra 0,5s da versão paralela, uma redução de 28,6% no custo computacional. Por fim, para imagens com 0,72MP, obteve-se na implementação sequencial um tempo de 0,44s e na implementação paralela 0,29s, o que significa uma redução de 34%. Uma relação entre o custo computacional e as profundidades máximas obtidas para imagens de 0,72MP, 1,28MP e 2MP pode ser observada graficamente na Figura 18.

Figura 18: Relação entre o tempo de processamento, a profundidade máxima obtida e a resolução utilizada para as implementações sequencial e paralela.

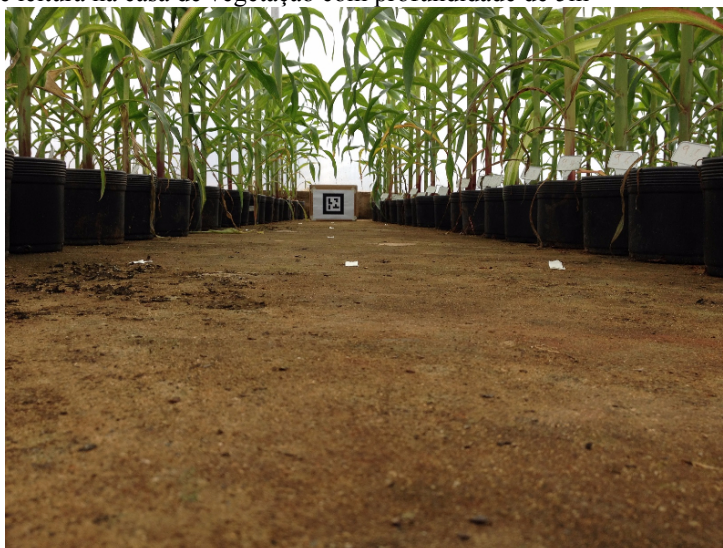


Fonte: O Autor.

4.4 RESULTADOS DE CAMPO

O procedimento realizado em laboratório para obtenção das coordenadas do robô foi repetido em campo em uma casa de vegetação a fim de verificar o erro de localização em uma situação real. A Figura 19 mostra a leitura com profundidade de 5m, com distância lateral nula. No total, onze imagens foram coletadas. O resultado do erro de localização estimados através das imagens coletadas pode ser observado na Tabela 7. A primeira coluna mostra a identificação do ponto de leitura, a segunda coluna indica a resolução da imagem, a terceira coluna mostra a profundidade real em metros, a quarta coluna apresenta a distância lateral real em metros e a quinta coluna indica o erro lido através da distância euclidiana entre a coordenada real e a coordenada estimada, em metros.

Figura 19: Exemplo de leitura na casa de vegetação com profundidade de 5m



Fonte: O Autor.

Tabela 7: Resultados das estimativas na casa de vegetação

ID	Resolução	Profundidade (real)	Distância Lateral (real)	Profundidade (lido)	Distância Lateral (lido)	Erro Coordenada Estimado
P1	2 MP	5,5	0	5,487	-0,157	0,157
P2	2 MP	5	0	5,069	-0,197	0,209
P3	2 MP	4	0	4,012	-0,144	0,144
P4	2 MP	4	0,3	4,061	0,350	0,079
P5	2 MP	4	-0,3	4,102	-0,404	0,146
P6	2 MP	3	0	2,980	-0,035	0,040
P7	2 MP	2	0	2,077	-0,045	0,089
P8	2 MP	1	0	1,119	-0,024	0,122
P9	2 MP	1	0,2	1,077	0,121	0,111
P10	2 MP	1	-0,3	1,104	-0,365	0,123
P11	2 MP	0,5	0	0,576	-0,017	0,078
P3	1,28 MP	4	0	4,059	-0,147	0,158
P4	1,28 MP	4	0,3	3,951	0,348	0,069
P5	1,28 MP	4	-0,3	4,100	-0,404	0,144
P6	1,28 MP	3	0	3,206	-0,037	0,209
P7	1,28 MP	2	0	1,999	-0,046	0,046
P8	1,28 MP	1	0	1,095	-0,026	0,098
P9	1,28 MP	1	0,2	1,054	0,112	0,103
P10	1,28 MP	1	-0,3	1,027	-0,345	0,053
P11	1,28 MP	0,5	0	0,574	-0,019	0,077
P6	0,72 MP	3	0	3,122	-0,034	0,127
P7	0,72 MP	2	0	2,069	-0,046	0,083
P8	0,72 MP	1	0	1,069	-0,026	0,073
P9	0,72 MP	1	0,2	1,047	0,117	0,096
P10	0,72 MP	1	-0,3	1,103	-0,366	0,123
P11	0,72 MP	0,5	0	0,587	-0,017	0,088

Fonte: O Autor.

Os resultados da Tabela 7, para imagens com resolução de 2MP, mostram que foi possível estimar a coordenada do robô em todos os pontos de leitura, mantendo a precisão do sistema para 5,5m. Em contrapartida, para isso, foi necessário alterar o valor do limiar hessiano, limitando-o para valores de 8000 ou abaixo, definido experimentalmente, provavelmente devido a mudança nas condições de iluminação entre o laboratório e a casa de vegetação, visto que tais variações podem reduzir número de pontos de interesse descritos nas imagens analisadas (HUYN *et al.*, 2009). Este comportamento se repetiu para as demais resoluções. Foi mensurado uma média do erro de 0,118m para a coordenada. O erro máximo desta foi de 0,209m e, o menor erro encontrado foi de 0,04m.

Para imagens com 1,28MP, foi possível estimar a coordenada nos pontos de leitura localizados mais próximos que 5m do objeto marcador (4m de profundidade máxima no caso das leituras na casa de vegetação), mantendo o mesmo limite de 4,7m testado em laboratório, totalizando nove imagens. Obteve-se uma média do erro de 0,106m para a coordenada. O erro máximo desta foi de 0,209m e, o menor erro encontrado foi de 0,046m.

Considerando a resolução de 0,72MP, foi possível estimar a coordenada nos pontos de leitura localizados mais próximos que 4m do objeto marcador (3m de profundidade máxima no caso das leituras na casa de vegetação), mantendo o mesmo comportamento do laboratório de 3,5m, totalizando seis imagens. Foi mensurado uma média do erro de 0,098m para a coordenada. O erro máximo desta foi de 0,127m e, o menor erro encontrado foi de 0,073m.

4.5 DISCUSSÃO

Os resultados referentes à precisão das estimativas para as coordenadas do robô mostram que a resolução das imagens tem influência direta sobre a distância máxima que um objeto marcador pode ser reconhecido. Como pode ser observado, a distância máxima de detecção caiu juntamente com a resolução da imagem. Isto indica que a eficácia do método é limitada por esta variável.

Deve ser notado que o erro da média da estimativa das coordenadas não sofreu uma grande alteração, em termos absolutos, em decorrência da mudança na resolução da imagem. A maior diferença absoluta entre as médias do erro é inferior à 3cm (0,72MP; 1,28MP).

Contudo, deve ser notado que esta diferença se deve, principalmente, a uma redução do alcance na detecção dos objetos marcadores.

A análise dos dados sugere que o erro nas estimativas das coordenadas de localização é afetado, principalmente, pelo erro na estimativa da profundidade. Isto pode ser inferido do fato de que há uma forte correlação¹² entre o erro das coordenadas e o erro das profundidades em todas as resoluções (0,88 para 2MP; 0,97 para 1,28MP e 0,94 para 0,72MP). Além disto, as correlações entre o erro das coordenadas e o erro da distância lateral, para cada resolução, pode ser classificada como fraca ou média (0,47 para 2MP; 0,5 para 1,28MP e 0,23 para 0,72MP), indicando que o efeito deste componente sobre o erro de localização é inferior ao impacto do erro sobre a estimativa de profundidade.

Também deve ser notado que a relação entre o erro na estimativa das coordenadas e a distância de aquisição é afetada pela resolução da imagem. O mesmo acontece quando se considera o erro entre a estimativa das profundidades e a distância da aquisição. Neste sentido, os resultados mostram que as correlações entre o erro nas estimativas de posição e a distância de aquisição da imagem, para as resoluções de 2MP, 1,28MP e 0,72MP, foram, respectivamente, 0,43, 0,59 e 0,69. As correlações entre o erro nas estimativas de profundidade e distância de aquisição para as resoluções de 2MP, 1,28MP e 0,72MP foram: 0,39; 0,56 e 0,6, respectivamente. Estes resultados enfatizam o efeito da resolução sobre o erro, pois mostram que quanto menor a resolução, mais rapidamente o erro cresce com a distância de aquisição. Este fato, contribuiu decisivamente para a implementação da versão do procedimento de autolocalização que usa o algoritmo paralelo P-SURF.

A comparação dos resultados com aqueles apresentados por Couto (2012) sugerem que o uso de imagens com maior resolução reduz a média do erro na determinação das coordenadas. Naquele trabalho, o autor desenvolveu um método de localização de robôs móveis autônomos através da obtenção de objetos marcadores no ambiente com o algoritmo SURF, em conjunto com um odômetro para localização do robô no treinamento da rota. O autor compara as coordenadas (x, y) de características paradas nas imagens para obter a localização do robô. O autor obteve uma média do erro de 0,163m utilizando uma câmera de resolução de 640x480 *pixels*.

¹² Correlação = Conforme Dancey e Reidy (2005), a correlação é forte quando possui uma classificação entre $r = 0,7$ e 1 ; moderada quando possui a classificação entre $r = 0,4$ e $0,6$ e, fraca, quando a classificação está entre $r = 0,1$ e $0,3$, considerando r entre 0 e 1 .

Li *et al.* (2014) também desenvolveram um sistema de localização de robôs com *webcams* fixas no ambiente via processamento digital de imagens, informando a coordenada do robô através da posição em que este foi localizado na imagem. O autor binariza a imagem, para segmentar o robô desta, encontrando seu centro de massa e estimando sua localização. O autor obteve como média do erro 0,09m para os dados citados no artigo, com resoluções de 640x480 e 480x360 linhas. Apesar do valor apresentado ser similar àqueles descritos neste trabalho, deve ser notado que as dimensões do objeto marcador (robô) são de 0,54m x 0,54m e os autores não informam o limite de distância de detecção dos objetos marcadores. O autor cita também que o erro da estimativa da coordenada aumenta quando a distância entre a câmera e o robô aumentam, da mesma forma que para este trabalho.

No que se refere ao desempenho computacional, a Figura 17 mostra que as imagens com maior resolução permitem que a detecção dos objetos marcadores seja feita a uma distância maior, o que é útil para que o robô possa tomar decisões com antecedência e prevenir-se de colisões por exemplo (OIKE; IKEDA; ASADA, 2003; FEUSER; IVLEV; GRÄSER, 2005; FLACCO *et al.*, 2012). Contudo, tais imagens exigem mais tempo para serem processadas. A figura também permite notar que o emprego da implementação paralela tem um ganho de escala em relação à implementação sequencial, permitindo que, por exemplo, uma imagem de 2 MP (1,28 MP) seja processada quase tão rapidamente quanto uma imagem de 1,28 MP (0,72 MP).

Em relação ao tempo de computação, Couto (2012) obteve um custo médio de 0,415s, em um computador com características muito próximas ao deste trabalho. O método proposto por Li *et al.* (2014) exige diversas câmeras e um objeto marcador (o robô), ao invés de uma câmera e diversos objetos marcadores. Isto pode exigir um alto custo computacional, este não abordado pelo autor.

5 CONCLUSÕES

Este trabalho apresentou um procedimento de autolocalização de *agrobots* em casas de vegetação que utiliza informações obtidas por processamento digital de imagens para estimar as coordenadas do robô. O procedimento de autolocalização foi feito em duas etapas. A primeira delas emprega o algoritmo SURF para identificar determinados objetos no ambiente de trabalho. Estes objetos foram inseridos no ambiente e possuíam características visuais que permitiam que fossem usados como objetos marcadores, pontos de referência para o *agrobot*, em um ambiente de realidade aumentada.

Na segunda etapa, as coordenadas do robô eram calculadas por um método que estende o procedimento proposto por Rahman *et al.* (2008) para determinar a distância entre a câmera e um objeto conhecido. Assim, como em Rahman *et al.* (2008), o procedimento estendido permite o cálculo de distâncias usando uma única imagem do objeto marcador. Adicionalmente, permite identificar a coordenada do robô em relação àquele objeto marcador, usando valores numéricos relativos ao invés de absolutos obtidos pelas fórmulas descritas neste trabalho.

O procedimento foi avaliado em três experimentos. O primeiro experimento teve como objetivo verificar se a resolução da imagem tinha influência sobre a precisão das coordenadas estimadas pelo procedimento proposto. Também foi realizada uma avaliação empírica a respeito do efeito da resolução da imagem sobre a distância máxima em que o algoritmo SURF foi capaz de identificar os objetos marcadores. Adicionalmente, o tempo de computação foi registrado.

Os resultados indicaram que, ao reduzir o tamanho das imagens de 2MP para 1,28MP, (uma redução de 36% no número de *pixels*) ou de 1,28MP para 0,72MP (43% de redução), a profundidade máxima para detecção dos objetos marcadores cai em 14,5% no primeiro caso e em 25,53% no segundo, isto é, reduz o alcance de detecção dos objetos marcadores. A variação da resolução das imagens analisadas também impactou na precisão da localização. Os resultados mostraram que, ao reduzir a resolução das imagens, o erro na estimativa das coordenadas em relação à distância de aquisição destas aumenta, embora que os erros não cresçam muito em termos absolutos.

Em relação ao tempo de computação, os resultados obtidos para as imagens de 0,72MP e 1,28MP mostram que quando a resolução é reduzida, há um ganho de 36,4% na eficiência. Quando se analisa imagens com 1,28MP e 2MP, o ganho é de 37,1%.

Estes resultados são relevantes do ponto de vista prático porque mostram que há uma relação de custo benefício entre o alcance e o tempo de processamento, uma vez que o problema, a determinação da posição o robô, usualmente está sujeita às restrições de processamento em tempo real. A importância desta informação pode ser vista em um exemplo no qual um *agrobot* se move à velocidade constante de 2 km/h. O uso do procedimento de autolocalização proposto em imagens de 2MP permitiria a detecção dos objetos marcadores a uma profundidade de 5,5m, contudo, durante o tempo exigido para o processamento, o robô teria se deslocado 0,62m. Para imagens de 1,28MP, a profundidade máxima seria de 4,7m e a cada verificação, o robô teria se deslocado 0,39m. Já para uma imagem de 0,72MP, a profundidade máxima de detecção seria de 3,5m e a verificação aconteceria a cada 0,25m.

Uma vez que os resultados do experimento anterior mostraram que a resolução da imagem tem impacto direto sobre o tempo de computação e negativo sobre a precisão das estimativas de posição, o segundo experimento avaliou o desempenho da solução em uma implementação sequencial e paralela com intuito de reduzir o custo computacional. Os resultados indicaram que o procedimento paralelo foi mais eficiente para as três resoluções testadas. A proporção da redução no tempo de processamento foi próxima a uma constante nas três resoluções propostas (em média 30,8%).

O terceiro experimento foi um teste de campo realizado em uma casa de vegetação. O objetivo foi o de avaliar o desempenho do procedimento proposto no ambiente para o qual foi projetado. Para realização do teste foi necessário efetuar ajustes nos parâmetros do procedimento em decorrência de variações na iluminação. Estas variações, como esperado, afetaram o desempenho da rotina de detecção de objetos marcadores através do algoritmo SURF. Após a calibração, os resultados apresentados se mostraram semelhantes aos do laboratório em relação à precisão e ao alcance.

5.1 TRABALHOS FUTUROS

O uso de métodos de visão computacional enfrenta o desafio adicional de que as variações de luminosidade, escala, rotação e contraste podem afetar o desempenho do sistema de localização (MARQUES FILHO; VIEIRA NETO, 1999). Como as estufas agrícolas raramente proveem um ambiente em que a iluminação é controlada, o uso do procedimento proposto exige que ele seja calibrado em cada ambiente. Considerando isto, um trabalho a ser desenvolvido é a seleção e uso de métodos computacionais para pré-processar as imagens a fim de reduzir o efeito da variação da iluminação nas imagens, assim como, melhorar a precisão da detecção dos objetos marcadores pelo algoritmo SURF.

Conforme supracitado, o robô precisa se localizar em tempo real para que possa tomar as decisões que necessita naquele instante de tempo. Neste contexto, um trabalho a ser desenvolvido é utilizar o algoritmo *optical flow* citado por Moraes (2011) como uma maneira para redução do tempo de computação em conjunto com o algoritmo SURF para detecção de objetos marcadores no ambiente.

REFERÊNCIAS

- ACACCIA, G. M. *et al.* Mobile robots in greenhouse cultivation: inspection and treatment of plants. In: **Proceedings of the 1st International Workshop on Advances in Service Robotics**. Bardolino, Italy. 2003.
- ARGENTA, M. A. *et al.* Processamento Baseado em Unidades de Processamento Gráficas (GPU) para Cálculo de Parâmetros Físicos e Geométricos do Tecido Ósseo Trabecular a partir de Dados de Micro Tomografia. **MECOM-CILAMCE 2010**, v. 29, p. 7004-7018, 2010.
- BALTZAKIS, H.; TRAHANIAS, P. Hybrid mobile robot localization using switching state-space models. **Robotics and Automation**, 2002. Proceedings. (ICRA). IEEE International Conference on. IEEE, 2002. p. 366-373.
- BAY, H.; TUYTELAARS, T.; VAN GOOL, L. Surf: Speeded up robust features. **Computer Vision–ECCV 2006**. Springer Berlin Heidelberg, 2006. p. 404-417.
- BEKEY, G. A. **Autonomous Robots: from biological inspiration to implementation and Control**. Cambridge: MIT Press, 2005. 577 p.
- BRADSKI, G.; KAEHLER, A. **Learning OpenCV: Computer vision with the OpenCV library**. 1ª ed. Sebastopol: "O'Reilly Media, Inc.", 2008. 580 p.
- BRITO, A. A. A de. **Casa de Vegetação com Diferentes Coberturas: Desempenho em Condições de Verão**. 2000. Dissertação (Mestrado em Engenharia Agrícola). Universidade Federal de Viçosa, Viçosa, Minas Gerais, 2000.
- CERMEÑO, Z. S. **Estufas: Instalações e Manejo**. 1990: Lisboa. Litexa Editora, Ltda. 355p. 1990.
- COUTO, L. N. **Sistema para localização robótica de veículos autônomos baseado em visão computacional por pontos de referência**. 2012. 82f. Dissertação (Mestrado em Ciências). Ciências de Computação e Matemática Computacional, Universidade de São Paulo, São Paulo, 2012.
- CRUZ, R. M. O.; CAVALCANTI, G. D. C.; REN, T. I. **Análise de técnicas de extração de características para o reconhecimento de dígitos manuscritos**. Centro de Informática. Universidade Federal de Pernambuco, Recife, 2008.
- DANCEY, C. P.; REIDY J. Análise de correlação. In: Dancey CP, Reidy J, organizadores. **Estatística sem matemática para psicólogos**. Porto Alegre: Artmed; 2006. p. 178-218.

ESTEVEES, J. S. **Metodologia de autolocalização absoluta em ambientes quase-estruturados**. 2005. 361f. Tese (Doutorado em Engenharia Eletrotécnica) - Escola de Engenharia, Universidade de Minho, Braga, Portugal, 2005.

FEUSER, J.; IVLEV, O.; GRÄSER, A. Collision prevention for rehabilitation robots with mapped virtual reality. In: **Rehabilitation Robotics**, 2005. ICORR 2005. 9th International Conference on. IEEE, 2005. p. 461-464.

FLACCO, F. *et al.* A depth space approach to human-robot collision avoidance. In: **Robotics and Automation (ICRA)**, 2012. IEEE International Conference on. IEEE, 2012. p. 338-345.

FOX, D. **Markov localization-a probabilistic framework for mobile robot localization and navigation**. 1998. Tese (Doutorado em Ciência da Computação) – Instituto de Ciência da Computação III, Universidade de Bonn, Bonn, Alemanha, 1998.

GONZÁLEZ, R. *et al.* Navigation techniques for mobile robots in greenhouses. **Applied Engineering in Agriculture**, v. 25, n. 2, p. 153, 2009.

GOUVEIA, M. C. M. **Estudo e Implementação de um Algoritmo de Localização baseado na correspondência de mapas**. 2008. Dissertação (Mestrado em Engenharia Eletrotécnica) - Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2008.

HAPP, P. N. *et al.* Uma implementação paralela para segmentação de imagens de alta resolução em GPUs. **Simpósio Brasileiro de Sensoriamento**, v. 15, p. 7713-7720. 2011

HAYET, J.; LERASLE, F.; DEVY, M. A visual landmark framework for mobile robot navigation. **Image and Vision Computing**, v. 25, n. 8, p. 1341-1351, 2007.

HOLANDA, G. C. *et al.* Sistema de medição de distância para robôs móveis utilizando algoritmo surf e visão estéreo. **V CONNEPI-2010**. Maceió, 2010.

HUYNH, D. Q.; SAINI, A.; LIU, W. Evaluation of three local descriptors on low resolution images for robot navigation. In: **Image and Vision Computing New Zealand**, 2009. IVCNZ'09. 24th International Conference. IEEE, 2009. p. 113-118.

LEÃO, C. W. M. *et al.* **Melhorando o desempenho do rastreamento de pontos de interesse em imagens através do paralelismo em GPU**. UFMG, 2010.

LI, I. *et al.* Mobile Robot Self-Localization System Using Single Webcam Distance Measurement Technology in Indoor Environments. **Sensors**, v. 14, n. 2, p. 2089-2109, 2014.

LOWE, D. G. Object recognition from local scale-invariant features. In: **Computer vision**, 1999. The proceedings of the seventh IEEE international conference on. IEEE, 1999. p. 1150-1157.

LU, H. *et al.* Robust and real-time self-localization based on omnidirectional vision for soccer robots. **Advanced Robotics**, v. 27, n. 10, p. 799-811, 2013.

MARQUES FILHO, O.; VIEIRA NETO, H. **Processamento digital de imagens**. Rio de Janeiro: Brasport, 1999. 311 p.

MICROSOFT PAINT. **Getting Started with Paint**. 2009. Disponível em: <http://windows.microsoft.com/en-us/windows7/getting-started-with-paint>. Acesso em: 20 ago. 2014.

MIRANDA, C. S. **A evolução da GPGPU: arquitetura e programação**. Departamento de Projeto Mecânico, Faculdade de Engenharia Mecânica, UNICAMP, 2011.

MIRINEJAD, S. H. *et al.* Design and Simulation of an Automated System for Greenhouse using Lab VIEW. **System**, v. 7, p. 280, 2008.

MOREIRA, M. A. **Fundamentos do Sensoriamento Remoto e Metodologias de Aplicação**. 3. ed. Atualizada e Ampliada. Viçosa: Editora UFV, 2007. 301 p

MORAIS, B. D. S. da. **Realidade aumentada em dispositivos móveis**. 2011. Dissertação (Mestrado em Engenharia Eletrônica e Telecomunicações) – Departamento de Eletrônica, Telecomunicações e Informática, Universidade de Aveiro, Aveiro, Portugal, 2011.

NAKHAEINIA, D. *et al.* A review of control architectures for autonomous navigation of mobile robots. **International Journal of the Physical Sciences**, v. 6, n. 2, p. 169-174, 2011.

NVIDIA. **CUDA**. 2015. Disponível em: http://www.nvidia.com.br/object/cuda_home_new_br.html. Acesso em: 15 fev. 2015.

OIKE, Y.; IKEDA, M.; ASADA, K. High-speed position detector using new row-parallel architecture for fast collision prevention system. In: **Circuits and Systems**, 2003. ISCAS'03. Proceedings of the 2003 International Symposium on. IEEE, 2003. p. IV-788-IV-791 vol. 4.

OPENCV. **Documentation**. 2013. Disponível em: <http://opencv.org/documentation.html> . Acesso em: 21 fev. 2014.

PARK, D. *et al.* A study on greenhouse automatic control system based on wireless sensor network. **Wireless Personal Communications**, v. 56, n. 1, p. 117-130, 2011.

PRATES, R. L. *et al.* Filtros de Correlação e Características Invariantes à Escala para o Reconhecimento de Faces. In: **XXX Simpósio Brasileiro de Telecomunicações**. 2012, Brasília.

PROCHAZKA, D. *et al.* Hybrid SURF-golay marker detection method for augmented reality applications. **Journal of WSCG**, Plzen, v. 20, n. 3, p. 197-204, 2012.

RAHMAN, A. *et al.* An image based approach to compute object distance. **International Journal of Computational Intelligence Systems**, v. 1, n. 4, p. 304-312, dez. 2008.

RIBANI, R. *et al.* Proposal of a Hybrid Algorithm Based on SURF and ORB. IX Workshop de Visão Computacional, 2013, Rio de Janeiro. **Anais**. Rio de Janeiro, FGV, 2013.

ROCHA, C. H.; OLIVEIRA, G. L. Análise de detectores de características visuais em ambientes subaquáticos. **IV MCSUL** (Southern Conference on Computational Modeling), 2010, Universidade Federal do Rio Grande. IV MCSUL, 2010.

RODRIGUES, F. A. **Técnicas de orientação aos objetos para computação científica paralela**. 2004. Dissertação (Mestrado em Física Aplicada) – Instituto de Física de São Carlos, São Carlos, 2004.

ROVIO. **User Manual**. 2008. Disponível em: <<http://www.wowwee.com/en/support/rovio>>. Acesso em: 23 fev. 2014.

RUSSEL, S; NORVING, P. **Inteligência Artificial**. Tradução PubliCare Consultoria. 2 ed. Rio de Janeiro: Elsevier Editora LTDA, 2004. 1021p.

SAMMONS, P. J.; FURUKAWA, T.; BULGIN, A. Autonomous pesticide spraying robot for use in a greenhouse. In: **Australian Conference on Robotics and Automation**. 2005. p. 1-9.

SÁNCHEZ-HERMOSILLA, J. *et al.* Mechatronic Description of a Laser Autoguided Vehicle for Greenhouse Operations. **Sensors**, v. 13, n. 1, p. 769-784, 2013.

SANTOS, M. C. **Revisão de Conceitos em Projeção, Homografia, Calibração de Câmera, Geometria Epipolar, Mapas de Profundidade e Varredura de Planos**. Unicamp, 2012. 49p.

TERUEL, B. J. Controle automatizado de casas de vegetação: Variáveis climáticas e fertigação. **Revista Brasileira de Engenharia Agrícola e Ambiental**, v. 14, n. 3, p. 237-245, 2010.

VAN HENTEN, E. J. *et al.* An autonomous robot for harvesting cucumbers in greenhouses. **Autonomous Robots**, v. 13, n. 3, p. 241-258, 2002.

VIANA, J. R. M. **Programação em GPU: Passado, presente e futuro**. Universidade Federal do Piauí, 2012.

YAGHOUBI, S. *et al.* Autonomous robots for agricultural tasks and farm assignment and future trends in agro robots. **International Journal of Mechanical & Mechatronics Engineering IJMME-IJENS**, v. 13, n. 03, p. 1-6, 2013.

ZHANG, N. Computing optimised parallel speeded-up robust features (p-surf) on multi-core processors. **International Journal of Parallel Programming**, v. 38, n. 2, p. 138-158, 2010.

ZHOU, J.; LIN, H. A self-localization and path planning technique for mobile robot navigation. In: **Intelligent Control and Automation (WCICA)**, 2011 9th World Congress on. IEEE, 2011. p. 694-699, 2011.

APÊNDICE A

A Tabela 8 mostra os valores numéricos dos resíduos de profundidade para a regressão da Equação 6 (segundo grau), em imagens de 2MP.

Tabela 8: Resíduos de profundidade para regressão da Equação 6 de segundo grau em imagens de 2MP

Resíduos (m)	Profundidade Real (m)	Erro (m)
4,557	5,5	0,943
4,502	5,3	0,798
4,458	5,1	0,642
4,382	4,9	0,518
4,328	4,7	0,372
4,275	4,5	0,225
4,211	4,3	0,089
4,116	4,1	0,016
4,033	3,9	0,133
3,941	3,7	0,241
3,830	3,5	0,330
3,721	3,3	0,421
3,603	3,1	0,503
3,460	2,9	0,560
3,301	2,7	0,601
3,129	2,5	0,629
2,962	2,3	0,662
2,718	2,1	0,618
2,441	1,9	0,541
2,146	1,7	0,446
1,818	1,5	0,318
1,411	1,3	0,111
0,950	1,1	0,150
0,596	0,9	0,304
0,551	0,7	0,149
2,537	0,5	2,037

Fonte: O Autor.

APÊNDICE B

A Tabela 9 mostra os valores numéricos dos resíduos de profundidade para a regressão da Equação 7 (potência), em imagens de 2MP.

Tabela 9: Resíduos de profundidade para regressão da Equação 7 de potência em imagens de 2MP

Resíduos (m)	Profundidade Real (m)	Erro (m)
5,496	5,5	0,004
5,274	5,3	0,026
5,108	5,1	0,008
4,843	4,9	0,057
4,670	4,7	0,030
4,509	4,5	0,009
4,330	4,3	0,030
4,087	4,1	0,013
3,893	3,9	0,007
3,695	3,7	0,005
3,480	3,5	0,020
3,289	3,3	0,011
3,103	3,1	0,003
2,898	2,9	0,002
2,697	2,7	0,003
2,503	2,5	0,003
2,336	2,3	0,036
2,120	2,1	0,020
1,907	1,9	0,007
1,711	1,7	0,011
1,521	1,5	0,021
1,312	1,3	0,012
1,095	1,1	0,005
0,909	0,9	0,009
0,701	0,7	0,001
0,488	0,5	0,012

Fonte: O Autor.

APÊNDICE C

A Tabela 10 mostra os valores numéricos dos resíduos de relação θ para a regressão da Equação 8 (potência), em imagens de 2MP.

Tabela 10: Resíduos de relação θ para regressão da Equação 8 de potência em imagens de 2MP

Resíduos	Relação Real	Erro
284,434	285,625	1,191
295,670	295,625	0,045
307,809	309,375	1,566
320,963	318,75	2,213
335,263	335	0,263
350,865	351,875	1,010
367,953	369,375	1,422
386,748	388,75	2,002
407,518	404,375	3,143
430,587	426,875	3,712
456,357	455,625	0,732
485,327	488,75	3,423
518,127	523,125	4,998
555,561	552,5	3,061
598,679	592,5	6,179
648,866	638,75	10,116
708,000	709,375	1,375
778,680	776,25	2,430
864,618	858,75	5,868
971,294	963,125	8,169
1107,156	1116,875	9,719
1285,925	1308,125	22,200
1531,453	1603,125	71,672
1889,134	1853,125	36,009
2457,128	2412,5	44,628
3493,637	3492,5	1,137

Fonte: O Autor.

APÊNDICE D

A Tabela 11 mostra os valores numéricos dos resíduos de relação θ para a regressão da equação de segundo grau $\theta = 181,28\delta^2 - 1486,3\delta + 325$, em imagens de 2MP.

Tabela 11: Resíduos de relação θ para regressão da equação de segundo grau $\theta = 181,28\delta^2 - 1486,3\delta + 325$ em imagens de 2MP

Resíduos	Relação Real	Erro
568,070	285,625	282,445
473,765	295,625	178,140
393,963	309,375	84,588
328,663	318,75	9,913
277,865	335	57,135
241,570	351,875	110,305
219,777	369,375	149,598
212,487	388,75	176,263
219,699	404,375	184,676
241,413	426,875	185,462
277,630	455,625	177,995
328,349	488,75	160,401
393,571	523,125	129,554
473,295	552,5	79,205
567,521	592,5	24,979
676,250	638,75	37,500
799,481	709,375	90,106
937,215	776,25	160,965
1089,451	858,75	230,701
1256,189	963,125	293,064
1437,430	1116,875	320,555
1633,173	1308,125	325,048
1843,419	1603,125	240,294
2068,167	1853,125	215,042
2307,417	2412,5	105,083
2561,170	3492,5	931,330

Fonte: O Autor.

APÊNDICE E

As Tabela 12 e 13 apresentam os valores de altura em *pixels* lidos manualmente, em cada profundidade, para as resoluções de 1,28MP e 0,72MP respectivamente.

Tabela 12: Leitura manual da profundidade e leitura em *pixels* do objeto marcador em imagens de 1,28MP.

Profundidade	Altura em <i>Pixels</i>	Profundidade	Altura em <i>Pixels</i>
5,5m	46	2,9m	89
5,3m	48	2,7m	96
5,1m	50	2,5m	104
4,9m	53	2,3m	111
4,7m	55	2,1m	123
4,5m	57	1,9m	137
4,3m	59	1,7m	153
4,1m	63	1,5m	172
3,9m	66	1,3m	200
3,7m	70	1,1m	241
3,5m	74	0,9m	292
3,3m	78	0,7m	380
3,1m	83	0,5m	551

Fonte: O Autor.

Tabela 13: Leitura manual da profundidade e leitura em *pixels* do objeto marcador em imagens de 0,72MP.

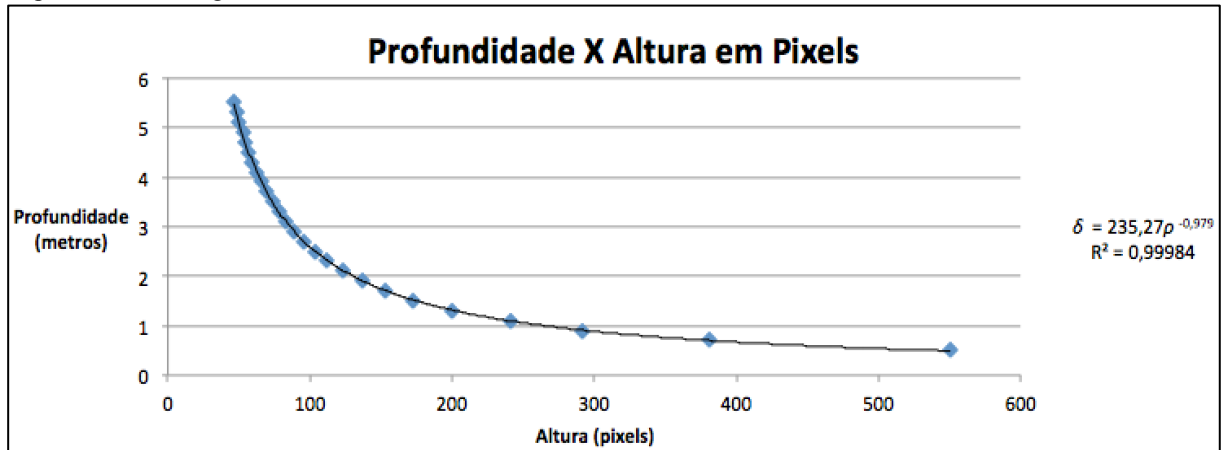
Profundidade	Altura em <i>Pixels</i>	Profundidade	Altura em <i>Pixels</i>
5,5m	35	2,9m	67
5,3m	36	2,7m	72
5,1m	38	2,5m	78
4,9m	40	2,3m	83
4,7m	41	2,1m	92
4,5m	43	1,9m	103
4,3m	44	1,7m	115
4,1m	47	1,5m	129
3,9m	50	1,3m	150
3,7m	52	1,1m	181
3,5m	56	0,9m	219
3,3m	59	0,7m	285
3,1m	62	0,5m	413

Fonte: O Autor.

As Figuras 20 e 21 mostram graficamente a relação entre os valores da profundidade δ e da altura em *pixels* ρ citados nas Tabelas 12 e 13, através de um ajuste da função potência

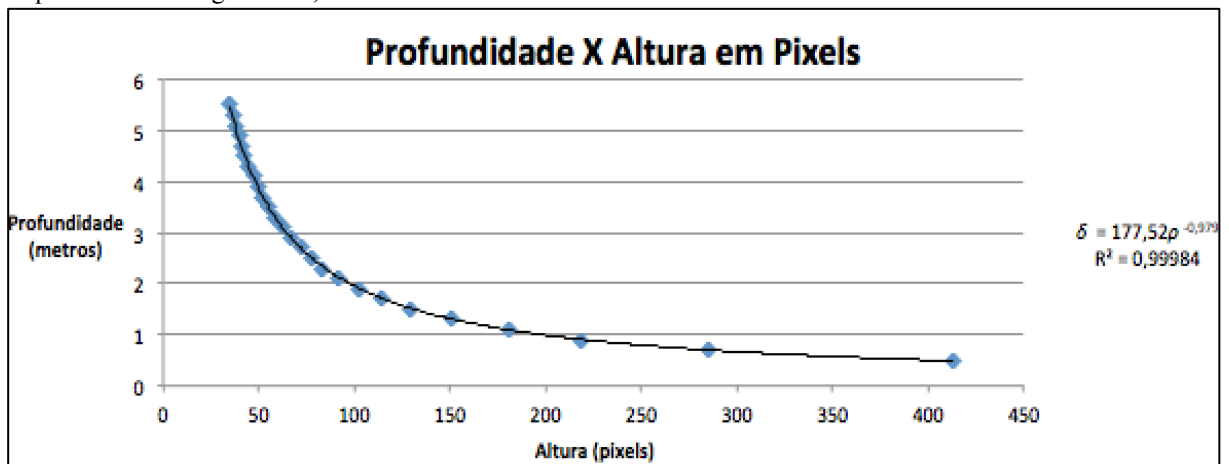
para as resoluções de 1,28MP e 0,72MP, respectivamente. A sobreposição da função ajustada aos dados experimentais também pode ser visualizada.

Figura 20: Gráfico relacionando profundidade em metros e altura em *pixels* com ajuste de função para equação de potência em imagens de 1,28MP.



Fonte: O Autor.

Figura 21: Gráfico relacionando profundidade em metros e altura em *pixels* com ajuste de função para equação de potência em imagens de 0,72MP.



Fonte: O Autor.

As Tabelas 14 e 15 apresentam a relação θ entre a profundidade δ e a distância lateral β , para as resoluções de 1,28MP e 0,72MP, respectivamente.

Tabela 14: Valores coletados manualmente para obtenção da equação para cálculo de distância lateral em imagens de 1,28MP

Profundidade	Distância Lateral 0,2m (Em pixels)	Distância Lateral 0,4m (Em pixels)	Relação (valor médio)	Profundidade	Distância Lateral 0,2m (Em pixels)	Distância Lateral 0,4m (Em pixels)	Relação (valor médio)
5,5m	45	93	228,50	2,9m	87	179	442,00
5,3m	46	96	236,50	2,7m	94	191	474,00
5,1m	49	100	247,50	2,5m	102	205	511,00
4,9m	50	104	255,00	2,3m	115	224	567,50
4,7m	52	110	268,00	2,1m	123	251	621,00
4,5m	55	116	281,50	1,9m	138	273	687,00
4,3m	58	120	295,50	1,7m	154	309	770,50
4,1m	61	127	311,00	1,5m	177	360	893,50
3,9m	64	132	323,50	1,3m	206	426	1046,50
3,7m	67	139	341,50	1,1m	252	523	1282,50
3,5m	72	148	364,50	0,9m	295	596	1482,50
3,3m	78	158	391,00	0,7m	386	0	1930,00
3,1m	82	170	418,50	0,5m	559	0	2794,00

Fonte: O Autor.

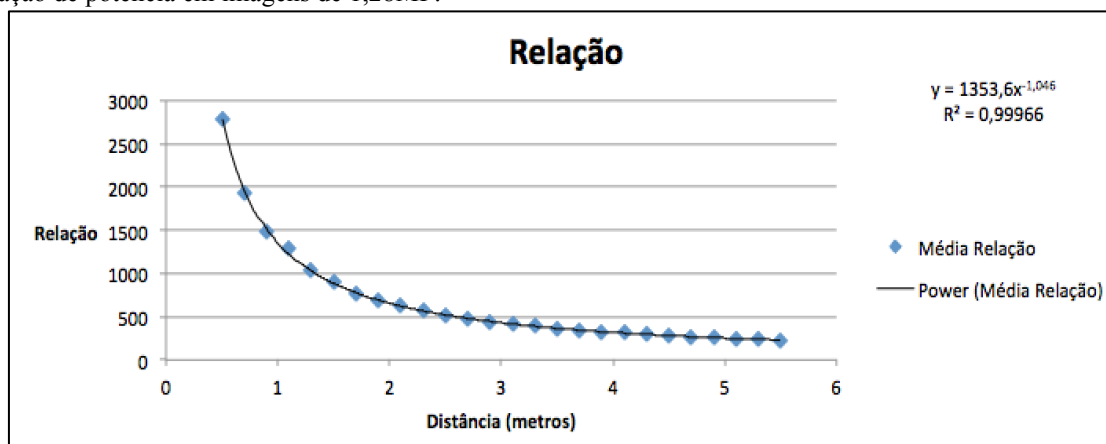
Tabela 15: Valores coletados manualmente para obtenção da equação para cálculo de distância lateral em imagens de 0,72MP

Profundidade	Distância Lateral 0,2m (Em pixels)	Distância Lateral 0,4m (Em pixels)	Relação (valor médio)	Profundidade	Distância Lateral 0,2m (Em pixels)	Distância Lateral 0,4m (Em pixels)	Relação (valor médio)
5,5m	34	70	171,38	2,9m	65	134	331,50
5,3m	35	72	177,38	2,7m	71	143	355,50
5,1m	37	75	185,63	2,5m	77	154	383,25
4,9m	38	78	191,25	2,3m	86	168	425,63
4,7m	39	82	201,00	2,1m	92	188	465,75
4,5m	41	87	211,13	1,9m	104	205	515,25
4,3m	44	90	221,63	1,7m	115	232	577,88
4,1m	46	95	233,25	1,5m	133	270	670,13
3,9m	48	99	242,63	1,3m	154	320	784,88
3,7m	50	104	256,13	1,1m	189	392	961,88
3,5m	54	111	273,38	0,9m	221	447	1111,88
3,3m	58	118	293,25	0,7m	290	0	1447,50
3,1m	62	128	313,88	0,5m	419	0	2095,50

Fonte: O Autor.

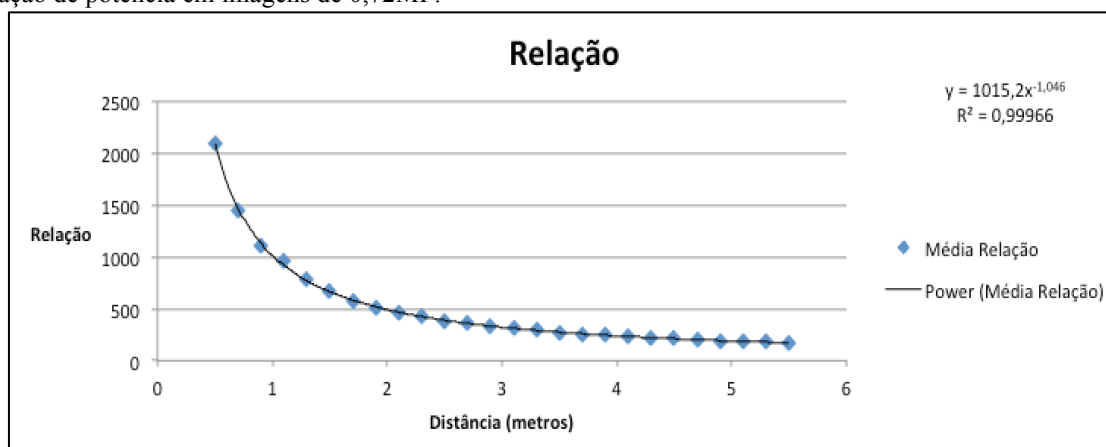
As Figuras 22 e 23 apresentam a regressão da função potência sobre os dados das Tabelas 14 e 15, juntamente com a sobreposição da função ajustada aos dados experimentais, para as resoluções de 1,28MP e 0,72MP, respectivamente.

Figura 22: Gráfico comparando a relação média com a profundidade em metros com ajuste de função para equação de potência em imagens de 1,28MP.



Fonte: O Autor.

Figura 23: Gráfico comparando a relação média com a profundidade em metros com ajuste de função para equação de potência em imagens de 0,72MP.



Fonte: O Autor.

APÊNDICE F

A Tabela 16 mostra os resultados obtidos para cada uma das coordenadas de aquisição em imagens de 2MP.

Tabela 16: Resultados de localização para cada uma das coordenadas de aquisição em imagens de 2MP

ID	Coordenada (real)	Profundidade (real)	Distância Lateral (real)	Coordenada (estimada)	Profundidade (estimada)	Distância Lateral (estimada)	Erro Profundidade	Erro Distância Lateral	Erro Coordenada
D1F1	2; 0,5	5,5	0	2,2; 0,337	5,663	0,200	0,163	0,200	0,258
D1F2	2; 0,7	5,3	0	2,089; 0,645	5,355	0,089	0,055	0,089	0,104
D1F3	2; 0,9	5,1	0	2,091; 0,883	5,117	0,091	0,017	0,091	0,093
D1F4	2; 1,1	4,9	0	2; 1,017	4,983	0,000	0,083	0,000	0,083
D1F5	2; 1,3	4,7	0	1,966; 1,195	4,805	-0,034	0,105	0,034	0,111
D1F6	2; 1,5	4,5	0	1,882; 1,49	4,510	-0,118	0,010	0,118	0,119
D1F7	2; 1,7	4,3	0	1,917; 1,495	4,505	-0,083	0,205	0,083	0,221
D1F8	2; 1,9	4,1	0	1,977; 2,076	3,924	-0,023	0,176	0,023	0,178
D1F9	2; 2,1	3,9	0	1,895; 2,05	3,950	-0,105	0,050	0,105	0,116
D1F10	2; 2,3	3,7	0	1,87; 2,296	3,704	-0,130	0,004	0,130	0,130
D1F11	2; 2,5	3,5	0	1,97; 2,277	3,723	-0,030	0,223	0,030	0,225
D1F12	2; 2,7	3,3	0	1,963; 2,796	3,204	-0,037	0,096	0,037	0,103
D1F13	2; 2,9	3,1	0	1,929; 2,934	3,066	-0,071	0,034	0,071	0,079
D1F14	2; 3,1	2,9	0	1,925; 2,943	3,057	-0,075	0,157	0,075	0,173
D1F15	2; 3,3	2,7	0	1,963; 3,08	2,920	-0,037	0,220	0,037	0,223
D1F16	2; 3,5	2,5	0	1,956; 3,325	2,675	-0,044	0,175	0,044	0,180
D1F17	2; 3,7	2,3	0	1,98; 3,513	2,487	-0,020	0,187	0,020	0,188
D1F18	2; 3,9	2,1	0	1,953; 3,838	2,162	-0,047	0,062	0,047	0,078
D1F19	2; 4,1	1,9	0	1,961;	2,026	-0,039	0,126	0,039	0,132

				3,974					
D1F20	2; 4,3	1,7	0	1,975; 4,251	1,749	-0,025	0,049	0,025	0,054
D1F21	2; 4,5	1,5	0	1,954; 4,484	1,516	-0,046	0,016	0,046	0,048
D1F22	2; 4,7	1,3	0	1,965; 4,672	1,328	-0,035	0,028	0,035	0,045
D1F23	2; 4,9	1,1	0	1,973; 4,918	1,082	-0,027	0,018	0,027	0,033
D1F24	2; 5,1	0,9	0	1,98; 5,07	0,930	-0,020	0,030	0,020	0,036
D1F25	2; 5,3	0,7	0	2,006; 5,3	0,700	0,006	0,000	0,006	0,006
D1F26	2; 5,5	0,5	0	2,005; 5,502	0,498	0,005	0,002	0,005	0,006
D2F1	2,2; 0,5	5,5	0,2	2,197; 0,464	5,536	0,197	0,036	0,003	0,036
D2F2	2,2; 0,7	5,3	0,2	2,359; 0,33	5,670	0,359	0,370	0,159	0,402
D2F3	2,2; 0,9	5,1	0,2	2,119; 0,805	5,195	0,119	0,095	0,081	0,125
D2F4	2,2; 1,1	4,9	0,2	2,221; 1,029	4,971	0,221	0,071	0,021	0,074
D2F5	2,2; 1,3	4,7	0,2	2,199; 1,137	4,863	0,199	0,163	0,001	0,163
D2F6	2,2; 1,5	4,5	0,2	2,228; 1,318	4,682	0,228	0,182	0,028	0,184
D2F7	2,2; 1,7	4,3	0,2	2,14; 1,807	4,193	0,140	0,107	0,060	0,123
D2F8	2,2; 1,9	4,1	0,2	2,174; 1,713	4,287	0,174	0,187	0,026	0,189
D2F9	2,2; 2,1	3,9	0,2	2,2; 2,065	3,935	0,200	0,035	0,000	0,035
D2F10	2,2; 2,3	3,7	0,2	2,158; 2,283	3,717	0,158	0,017	0,042	0,045
D2F11	2,2; 2,5	3,5	0,2	2,168; 2,423	3,577	0,168	0,077	0,032	0,083
D2F12	2,2; 2,7	3,3	0,2	2,162; 2,686	3,314	0,162	0,014	0,038	0,040
D2F13	2,2; 2,9	3,1	0,2	2,211; 2,805	3,195	0,211	0,095	0,011	0,096
D2F14	2,2; 3,1	2,9	0,2	2,184; 3,084	2,916	0,184	0,016	0,016	0,023
D2F15	2,2; 3,3	2,7	0,2	2,157; 3,271	2,729	0,157	0,029	0,043	0,052
D2F16	2,2; 3,5	2,5	0,2	2,094; 3,403	2,597	0,094	0,097	0,106	0,144
D2F17	2,2; 3,7	2,3	0,2	2,098; 3,633	2,367	0,098	0,067	0,102	0,122
D2F18	2,2; 3,9	2,1	0,2	2,164;	2,126	0,164	0,026	0,036	0,044

				3,874					
D2F19	2,2; 4,1	1,9	0,2	2,138; 4,217	1,783	0,138	0,117	0,062	0,132
D2F20	2,2; 4,3	1,7	0,2	2,169; 4,276	1,724	0,169	0,024	0,031	0,039
D2F21	2,2; 4,5	1,5	0,2	2,195; 4,471	1,529	0,195	0,029	0,005	0,029
D2F22	2,2; 4,7	1,3	0,2	2,182; 4,635	1,365	0,182	0,065	0,018	0,067
D2F23	2,2; 4,9	1,1	0,2	2,183; 4,915	1,085	0,183	0,015	0,017	0,023
D2F24	2,2; 5,1	0,9	0,2	2,177; 5,106	0,894	0,177	0,006	0,023	0,023
D2F25	2,2; 5,3	0,7	0,2	2,18; 5,286	0,714	0,180	0,014	0,020	0,024
D3F1	2,4; 0,5	5,5	0,4	2,479; 0,479	5,521	0,479	0,021	0,079	0,082
D3F2	2,4; 0,7	5,3	0,4	2,461; 0,681	5,319	0,461	0,019	0,061	0,064
D3F3	2,4; 0,9	5,1	0,4	2,459; 0,751	5,249	0,459	0,149	0,059	0,160
D3F4	2,4; 1,1	4,9	0,4	2,408; 0,827	5,173	0,408	0,273	0,008	0,273
D3F5	2,4; 1,3	4,7	0,4	2,301; 1,339	4,661	0,301	0,039	0,099	0,107
D3F6	2,4; 1,5	4,5	0,4	2,4; 1,772	4,228	0,400	0,272	0,000	0,272
D3F7	2,4; 1,7	4,3	0,4	2,384; 1,911	4,089	0,384	0,212	0,016	0,212
D3F8	2,4; 1,9	4,1	0,4	2,338; 1,678	4,322	0,338	0,222	0,062	0,231
D3F9	2,4; 2,1	3,9	0,4	2,391; 2,094	3,906	0,391	0,006	0,009	0,011
D3F10	2,4; 2,3	3,7	0,4	2,406; 2,238	3,762	0,406	0,062	0,006	0,063
D3F11	2,4; 2,5	3,5	0,4	2,357; 2,445	3,555	0,357	0,055	0,043	0,070
D3F12	2,4; 2,7	3,3	0,4	2,325; 2,665	3,335	0,325	0,035	0,075	0,083
D3F13	2,4; 2,9	3,1	0,4	2,339; 2,968	3,032	0,339	0,068	0,061	0,091
D3F14	2,4; 3,1	2,9	0,4	2,381; 3,094	2,906	0,381	0,006	0,019	0,020
D3F15	2,4; 3,3	2,7	0,4	2,32; 3,405	2,595	0,320	0,105	0,080	0,132
D3F16	2,4; 3,5	2,5	0,4	2,298; 3,38	2,620	0,298	0,120	0,102	0,157
D3F17	2,4; 3,7	2,3	0,4	2,299; 3,792	2,208	0,299	0,092	0,101	0,136
D3F18	2,4; 3,9	2,1	0,4	2,335;	2,133	0,335	0,033	0,065	0,073

				3,867					
D3F19	2,4; 4,1	1,9	0,4	2,327; 4,146	1,854	0,327	0,046	0,073	0,086
D3F20	2,4; 4,3	1,7	0,4	2,664; 4,298	1,702	0,664	0,002	0,264	0,264
D3F21	2,4; 4,5	1,5	0,4	2,371; 4,527	1,473	0,371	0,027	0,029	0,040
D3F22	2,4; 4,7	1,3	0,4	2,346; 4,748	1,252	0,346	0,048	0,054	0,072
D3F23	2,4; 4,9	1,1	0,4	2,343; 4,973	1,027	0,343	0,073	0,057	0,093
D3F24	2,4; 5,1	0,9	0,4	2,371; 5,025	0,975	0,371	0,075	0,029	0,080

Fonte: O Autor.

APÊNDICE G

A Tabela 17 mostra os resultados obtidos para cada uma das coordenadas de aquisição em imagens de 1,28MP.

Tabela 17: Resultados de localização para cada uma das coordenadas de aquisição em imagens de 1,28MP

ID	Coordenada (real)	Profundidade (real)	Distância Lateral (real)	Coordenada (estimado)	Profundidade (estimado)	Distância Lateral (estimado)	Erro Profundidade	Erro Distância Lateral	Erro Coordenada
D1F5	2; 1,3	4,7	0	1,968; 1,084	4,916	-0,032	0,216	0,032	0,218
D1F6	2; 1,5	4,5	0	1,889; 1,768	4,232	-0,111	0,268	0,111	0,290
D1F7	2; 1,7	4,3	0	1,924; 2,067	3,933	-0,076	0,367	0,076	0,374
D1F8	2; 1,9	4,1	0	1,978; 1,853	4,147	-0,022	0,047	0,022	0,052
D1F9	2; 2,1	3,9	0	1,893; 1,925	4,075	-0,107	0,175	0,107	0,205
D1F10	2; 2,3	3,7	0	1,869; 2,067	3,933	-0,131	0,233	0,131	0,268
D1F11	2; 2,5	3,5	0	1,969; 2,346	3,654	-0,031	0,154	0,031	0,157
D1F12	2; 2,7	3,3	0	1,969; 2,547	3,453	-0,031	0,153	0,031	0,156
D1F13	2; 2,9	3,1	0	1,924; 2,811	3,189	-0,076	0,089	0,076	0,117
D1F14	2; 3,1	2,9	0	1,923; 2,864	3,136	-0,077	0,236	0,077	0,248
D1F15	2; 3,3	2,7	0	1,962; 3,182	2,818	-0,038	0,118	0,038	0,124
D1F16	2; 3,5	2,5	0	1,953; 3,443	2,557	-0,047	0,057	0,047	0,074
D1F17	2; 3,7	2,3	0	1,977; 3,701	2,299	-0,023	0,001	0,023	0,023
D1F18	2; 3,9	2,1	0	1,952; 3,867	2,133	-0,048	0,033	0,048	0,058
D1F19	2; 4,1	1,9	0	1,946; 4,378	1,622	-0,054	0,278	0,054	0,283
D1F20	2; 4,3	1,7	0	1,973; 4,271	1,729	-0,027	0,029	0,027	0,039
D1F21	2; 4,5	1,5	0	1,955; 4,55	1,450	-0,045	0,050	0,045	0,067
D1F22	2; 4,7	1,3	0	1,965;	1,363	-0,035	0,063	0,035	0,072

				4,637					
D1F23	2; 4,9	1,1	0	1,973; 4,888	1,112	-0,027	0,012	0,027	0,029
D1F24	2; 5,1	0,9	0	1,981; 5,071	0,929	-0,019	0,029	0,019	0,035
D1F25	2; 5,3	0,7	0	1,995; 5,325	0,675	-0,005	0,025	0,005	0,025
D1F26	2; 5,5	0,5	0	1,998; 5,507	0,493	-0,002	0,007	0,002	0,007
D2F5	2,2; 1,3	4,7	0,2	2,054; 1,428	4,572	0,054	0,128	0,146	0,194
D2F6	2,2; 1,5	4,5	0,2	2,113; 1,727	4,273	0,113	0,227	0,087	0,243
D2F7	2,2; 1,7	4,3	0,2	2,136; 1,945	4,055	0,136	0,245	0,064	0,253
D2F8	2,2; 1,9	4,1	0,2	2,142; 2,061	3,939	0,142	0,161	0,058	0,171
D2F9	2,2; 2,1	3,9	0,2	2,211; 1,907	4,093	0,211	0,193	0,011	0,193
D2F10	2,2; 2,3	3,7	0,2	2,174; 2,102	3,898	0,174	0,198	0,026	0,200
D2F11	2,2; 2,5	3,5	0,2	2,176; 2,325	3,675	0,176	0,175	0,024	0,176
D2F12	2,2; 2,7	3,3	0,2	2,166; 2,623	3,377	0,166	0,077	0,034	0,085
D2F13	2,2; 2,9	3,1	0,2	2,207; 2,839	3,161	0,207	0,061	0,007	0,061
D2F14	2,2; 3,1	2,9	0,2	2,199; 2,904	3,096	0,199	0,196	0,001	0,196
D2F15	2,2; 3,3	2,7	0,2	2,16; 3,245	2,755	0,160	0,055	0,040	0,068
D2F16	2,2; 3,5	2,5	0,2	2,107; 3,242	2,758	0,107	0,258	0,093	0,274
D2F17	2,2; 3,7	2,3	0,2	2,095; 3,668	2,332	0,095	0,032	0,105	0,110
D2F18	2,2; 3,9	2,1	0,2	2,164; 3,852	2,148	0,164	0,048	0,036	0,060
D2F19	2,2; 4,1	1,9	0,2	2,145; 4,161	1,839	0,145	0,061	0,055	0,082
D2F20	2,2; 4,3	1,7	0,2	2,161; 4,311	1,689	0,161	0,011	0,039	0,040
D2F21	2,2; 4,5	1,5	0,2	2,182; 4,546	1,454	0,182	0,046	0,018	0,050
D2F22	2,2; 4,7	1,3	0,2	2,173; 4,675	1,325	0,173	0,025	0,027	0,036
D2F23	2,2; 4,9	1,1	0,2	2,188; 4,895	1,105	0,188	0,005	0,012	0,013
D2F24	2,2; 5,1	0,9	0,2	2,185; 5,081	0,919	0,185	0,019	0,015	0,024
D2F25	2,2; 5,3	0,7	0,2	2,178;	0,714	0,178	0,014	0,022	0,026

				5,286						
D3F5	2,4; 1,3	4,7	0,4	2,305; 1,305	4,695	0,305	0,005	0,095	0,095	
D3F6	2,4; 1,5	4,5	0,4	2,465; 1,53	4,470	0,465	0,030	0,065	0,072	
D3F7	2,4; 1,7	4,3	0,4	2,398; 1,783	4,217	0,398	0,083	0,002	0,083	
D3F8	2,4; 1,9	4,1	0,4	2,326; 1,739	4,261	0,326	0,161	0,074	0,177	
D3F9	2,4; 2,1	3,9	0,4	2,345; 2,485	3,515	0,345	0,385	0,055	0,389	
D3F10	2,4; 2,3	3,7	0,4	2,403; 2,254	3,746	0,403	0,046	0,003	0,047	
D3F11	2,4; 2,5	3,5	0,4	2,361; 2,431	3,569	0,361	0,069	0,039	0,079	
D3F12	2,4; 2,7	3,3	0,4	2,326; 2,644	3,356	0,326	0,056	0,074	0,093	
D3F13	2,4; 2,9	3,1	0,4	2,358; 2,847	3,153	0,358	0,053	0,042	0,067	
D3F14	2,4; 3,1	2,9	0,4	2,378; 3,123	2,877	0,378	0,023	0,022	0,032	
D3F15	2,4; 3,3	2,7	0,4	2,325; 3,343	2,657	0,325	0,043	0,075	0,086	
D3F16	2,4; 3,5	2,5	0,4	2,288; 3,439	2,561	0,288	0,061	0,112	0,127	
D3F17	2,4; 3,7	2,3	0,4	2,325; 3,639	2,361	0,325	0,061	0,075	0,097	
D3F18	2,4; 3,9	2,1	0,4	2,299; 4,044	1,956	0,299	0,144	0,101	0,176	
D3F19	2,4; 4,1	1,9	0,4	2,348; 4,067	1,933	0,348	0,033	0,052	0,062	
D3F20	2,4; 4,3	1,7	0,4	2,336; 4,285	1,715	0,336	0,015	0,064	0,066	
D3F21	2,4; 4,5	1,5	0,4	2,348; 4,618	1,382	0,348	0,118	0,052	0,129	
D3F22	2,4; 4,7	1,3	0,4	2,348; 4,731	1,269	0,348	0,031	0,052	0,060	
D3F23	2,4; 4,9	1,1	0,4	2,359; 4,938	1,062	0,359	0,038	0,041	0,055	
D3F24	2,4; 5,1	0,9	0,4	2,337; 5,098	0,902	0,337	0,002	0,063	0,064	

Fonte: O Autor.

APÊNDICE H

A Tabela 18 mostra os resultados obtidos para cada uma das coordenadas de aquisição em imagens de 0,72MP.

Tabela 18: Resultados de localização para cada uma das coordenadas de aquisição em imagens de 0,72MP

ID	Coordenada (real)	Profundidade (real)	Distância Lateral (real)	Coordenada (estimado)	Profundidade (estimado)	Distância Lateral (estimado)	Erro Profundidade	Erro Distância Lateral	Erro Coordenada
D1F11	2; 2,5	3,5	0	1,965; 2,598	3,402	-0,035	0,098	0,035	0,104
D1F12	2; 2,7	3,3	0	1,965; 2,907	3,093	-0,035	0,207	0,035	0,210
D1F13	2; 2,9	3,1	0	1,925; 2,878	3,122	-0,075	0,022	0,075	0,078
D1F14	2; 3,1	2,9	0	1,922; 3,05	2,950	-0,078	0,050	0,078	0,093
D1F15	2; 3,3	2,7	0	1,963; 3,118	2,882	-0,037	0,182	0,037	0,185
D1F16	2; 3,5	2,5	0	1,955; 3,395	2,605	-0,045	0,105	0,045	0,115
D1F17	2; 3,7	2,3	0	1,976; 3,631	2,369	-0,024	0,069	0,024	0,073
D1F18	2; 3,9	2,1	0	1,953; 3,827	2,173	-0,047	0,073	0,047	0,087
D1F19	2; 4,1	1,9	0	1,959; 4,018	1,982	-0,041	0,082	0,041	0,091
D1F20	2; 4,3	1,7	0	1,976; 4,251	1,749	-0,024	0,049	0,024	0,054
D1F21	2; 4,5	1,5	0	1,957; 4,437	1,563	-0,043	0,063	0,043	0,076
D1F22	2; 4,7	1,3	0	1,962; 4,678	1,322	-0,038	0,022	0,038	0,044
D1F23	2; 4,9	1,1	0	1,971; 4,937	1,063	-0,029	0,037	0,029	0,047
D1F24	2; 5,1	0,9	0	1,979; 5,114	0,886	-0,021	0,014	0,021	0,025
D1F25	2; 5,3	0,7	0	2,007; 5,296	0,704	0,007	0,004	0,007	0,008
D1F26	2; 5,5	0,5	0	2,004; 5,489	0,511	0,004	0,011	0,004	0,012
D2F11	2,2; 2,5	3,5	0,2	2,18; 2,353	3,647	0,180	0,147	0,020	0,149
D2F12	2,2;	3,3	0,2	2,261;	3,316	0,261	0,016	0,061	0,064

	2,7			2,684					
D2F13	2,2; 2,9	3,1	0,2	2,181; 3,168	2,832	0,181	0,268	0,019	0,269
D2F14	2,2; 3,1	2,9	0,2	2,199; 2,94	3,060	0,199	0,160	0,001	0,160
D2F15	2,2; 3,3	2,7	0,2	2,181; 3,032	2,968	0,181	0,268	0,019	0,269
D2F16	2,2; 3,5	2,5	0,2	2,091; 3,443	2,557	0,091	0,057	0,109	0,123
D2F17	2,2; 3,7	2,3	0,2	2,096; 3,655	2,345	0,096	0,045	0,104	0,114
D2F18	2,2; 3,9	2,1	0,2	2,163; 3,872	2,128	0,163	0,028	0,037	0,047
D2F19	2,2; 4,1	1,9	0,2	2,156; 4,061	1,939	0,156	0,039	0,044	0,058
D2F20	2,2; 4,3	1,7	0,2	2,171; 4,258	1,742	0,171	0,042	0,029	0,051
D2F21	2,2; 4,5	1,5	0,2	2,199; 4,457	1,543	0,199	0,043	0,001	0,043
D2F22	2,2; 4,7	1,3	0,2	2,179; 4,648	1,352	0,179	0,052	0,021	0,057
D2F23	2,2; 4,9	1,1	0,2	2,184; 4,909	1,091	0,184	0,009	0,016	0,018
D2F24	2,2; 5,1	0,9	0,2	2,186; 5,076	0,924	0,186	0,024	0,014	0,028
D2F25	2,2; 5,3	0,7	0,2	2,181; 5,276	0,724	0,181	0,024	0,019	0,030
D3F11	2,4; 2,5	3,5	0,4	2,315; 2,598	3,402	0,315	0,098	0,085	0,130
D3F12	2,4; 2,7	3,3	0,4	2,297; 2,912	3,088	0,297	0,212	0,103	0,236
D3F13	2,4; 2,9	3,1	0,4	2,346; 2,948	3,052	0,346	0,048	0,054	0,072
D3F14	2,4; 3,1	2,9	0,4	2,391; 3,03	2,970	0,391	0,070	0,009	0,070
D3F15	2,4; 3,3	2,7	0,4	2,345; 3,234	2,766	0,345	0,066	0,055	0,086
D3F16	2,4; 3,5	2,5	0,4	2,284; 3,46	2,540	0,284	0,040	0,116	0,122
D3F17	2,4; 3,7	2,3	0,4	2,332; 3,603	2,397	0,332	0,097	0,068	0,118
D3F18	2,4; 3,9	2,1	0,4	2,317; 3,953	2,047	0,317	0,053	0,083	0,098
D3F19	2,4; 4,1	1,9	0,4	2,371; 3,973	2,027	0,371	0,127	0,029	0,130
D3F20	2,4; 4,3	1,7	0,4	2,335; 4,294	1,706	0,335	0,006	0,065	0,066
D3F21	2,4; 4,5	1,5	0,4	2,37; 4,547	1,453	0,370	0,047	0,030	0,055
D3F22	2,4;	1,3	0,4	2,353;	1,284	0,353	0,016	0,047	0,050

	4,7			4,716					
D3F23	2,4; 4,9	1,1	0,4	2,366; 4,926	1,074	0,366	0,026	0,034	0,043
D3F24	2,4; 5,1	0,9	0,4	2,341; 5,089	0,911	0,341	0,011	0,059	0,060

Fonte: O Autor.