Utilização de imagens omnidirecionais catadióptricas para visão computacional de veículos não tripulados

Bianca P. Rocha1, André L. Olivete2

1Discente de Informática – Instituto Federal de Educação Ciência e Tecnologia de São Paulo, Campus Presidente Epitácio

2Docente de Informática – Instituto Federal de Educação Ciência e Tecnologia de São Paulo, Campus Presidente Epitácio

Rua José Ramos Júnior, 27-50 - 19470-000 Presidente Epitácio, SP - Brasil

puertarochabianca@gmail.com, olivete@ifsp.edu.br

**Abstract.** Researches that aimed at reducing the cost of unmanned vehicles allow these technologies to be more accessible and expand the application areas of these vehicles. This project is part of the development of a low-cost remotely controlled vehicle for applications in highly hazardous environments for human beings, allowing navigation and obtaining information about the environment, and aims to study the main methods for rectification and generation of panoramic images from of omnidirectional catadioptric images obtained with cameras and mirrors, as well as the identification of objects in the scenes captured by this camera.

**Resumo.** As pesquisas que visam a diminuição do custo de veículos não tripulados permitem que essas tecnologias sejam mais acessíveis e expandem as áreas de aplicação desses veículos. O presente projeto faz parte do desenvolvimento de um veículo remotamente controlado de baixo custo para aplicações em ambientes de grande periculosidade para seres humanos, permitindo navegar e obter informações sobre o ambiente, e visa estudar os principais métodos para retificação e geração de imagens panorâmicas a partir de imagens omnidirecionais catadióptricas obtidas com câmeras e espelhos, como também a identificação de objetos nas cenas capturadas por essa câmera.

# 1. Introdução

Os veículos autônomos ou remotamente controlados necessitam de sensores para detecção de obstáculos durante sua movimentação no ambiente, como sonares e sensores de distância, porém a visão é o principal sensor para a tomada de decisões em seus movimentos.

A utilização de um sistema de visão omnidirecional facilita a aquisição de imagens com um campo de visão aumentado, de todo o ambiente ao redor do veículo. Esse tipo de imagem pode ser obtida de várias formas diferentes, como: a combinação de várias imagens de uma câmera giratória, por meio da combinação de imagens obtidas por um conjunto de câmeras, ou por meio da combinação de uma câmera e um espelho (Olivete, 2014). Os sistemas formados por câmera e espelho são denominados “sistemas omnidirecionais catadióptricos”.

As imagens omnidirecionais possuem várias aplicações em visão computacional, como a orientação da navegação dos robôs dentro do ambiente e o mapeamento do ambiente identificando os objetos existentes. Para isso podem ser utilizadas imagens panorâmicas geradas a partir de imagens omnidirecionais catadióptricas, como em Spacek e Burbridge (2007).

Outra forma de utilização das imagens omnidirecionais catadióptricas é a retificação através da reprojeção para um determinado plano, gerando uma imagem de vista aérea, também conhecida como *bird’s eye view*. Essa visão é uma projeção ortográfica no plano do solo através de uma transformação radial da imagem omnidirecional, obtida através da correção radial ao redor do centro da imagem (Vassalo et al, 2004).

Com a imagem reprojetada para vista aérea é possível determinar objetos e realizar medições na imagem dentro do raio de visão do veículo, essa imagem pode ser utilizada como uma imagem aérea do local a ser analisado (Gaspar, 2002).

Os veículos não tripulados podem ter toda a lógica de tomada de decisão embarcada sendo necessário carregar equipamentos que consomem grande quantidade de energia, quando comparado com placas de prototipação como o Arduino, porém grande parte desses veículos possuem embarcado apenas a lógica de acionamento dos motores e à comunicação com uma estação base.

Nessa forma de implementação se faz necessário definir protocolos de comunicação entre os componentes do sistema e gera um certo atraso referente a comunicação, porém permite que a lógica para tomada de decisão seja mais elaborada e utilize grande poder computacional, além de possibilitar um maior tempo de navegação dentro do ambiente.

O presente projeto tem como objetivo investigar a utilização de imagens omnidirecionais na orientação da navegação para veículos não tripulados autônomos e/ou remotamente controlados. Para isso, serão investigadas as técnicas e modelos utilizados na retificação e geração de imagens panorâmicas a partir de imagens omnidirecionais e a posterior utilização dessas imagens geradas na detecção de objetos dentro de uma cena.

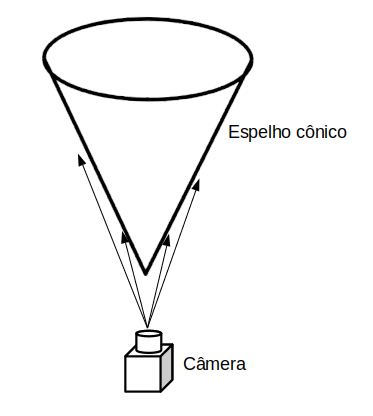
# 2. Fundamentação teórica

Nesta seção é apresentada a fundamentação teórica sobre os principais conceitos abordados para o desenvolvimento do projeto. Iniciando com os sistemas omnidirecionais catadióptricos, a modelagem matemática para o problema, os processos de calibração e retificação e, por fim, as formas de detecção de objetos analisadas para o desenvolvimento do projeto.

# 2.1. Sistemas omnidirecionais catadióptricos

Os sistemas omnidirecionais possibilitam a captura de imagens omnidirecionais com visão horizontal de 360º. Os sistemas de visão omnidirecionais catadióptricos são formados pela combinação de lentes e espelhos, sendo que esses espelhos podem ter vários formatos, como hiperbólico, elíptico, parabólico e esférico (Pereira et al., 2002). A Figura 1 demonstra como é constituído um sistema catadióptrico.

**Figura 1. Esquema de um sistema de visão catadióptrico.**



Fonte: Autores.

As imagens coletadas pelo sistema catadióptrico apresentam distorções por conta do espelho, conjunto de lentes, como também do posicionamento da câmera em relação ao espelho. Para minimizar estas distorções é necessário realizar uma calibração prévia do sistema como um todo, para que as imagens aéreas (*bird’s eye view*) e panorâmicas possam ser geradas, a partir da imagem coletada, e utilizadas para fins de navegação e detecção de obstáculos.

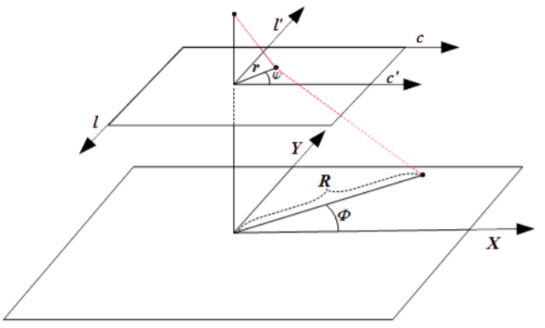
# 2.2. Modelo empírico de mapeamento das coordenadas polares

O modelo de coordenadas polares utiliza uma função para modelar a variação de escala da distância radial na imagem em função da mesma distância no espaço objeto. É uma abordagem bastante comum para mapeamento entre as coordenadas do espaço imagem e as coordenadas do espaço objeto.

As distâncias radiais possuem simetria radial. A razão entre a distância polar no espaço imagem e a distância polar no espaço objeto é uma variação de escala que pode ser modelada por polinômios unidimensionais (Spacek, 2005).

A Figura 2 mostra a relação entre a distância na imagem e a distância no terreno, e que essa relação pode ser mapeada através de um polinômio.

**Figura 2. Relação entre os pontos no espaço imagem e objeto no modelo de distâncias polares.**



Fonte: Olivete & Tommaselli (2016).

A relação entre um ponto no espaço objeto e o seu ponto correspondente na imagem é dado pela equação (1).

(1)

Considerando *F(R)* como sendo um polinômio de grau *n* que modela a relação entre *R* e *r*, apresentado na equação (2).

*F(R) = a0 + a1 . R + a2 . R2 + a3 . R3 + … + an . Rn* (2)

Neste modelo é utilizado também um polinômio para modelar as variações angulares horizontais na imagem em relação ao ângulo no espaço objeto minimizando o problema de alinhamento entre os eixos.

O modelo de coordenadas polares faz a transformação das coordenadas cartesianas para coordenadas polares e utiliza a equação (2) para modelar a relação entre a distância polar no espaço imagem e espaço objeto.

Para corrigir a diferença entre os ângulos do espaço imagem (*θ*) e do espaço objeto (*Θ*) pode ser utilizada apenas uma constante se for considerada apenas uma rotação entre os sistemas. Considerando a falta de alinhamento entre a câmera e o cone, torna-se necessário um polinômio de maior grau para modelar essas diferenças, conforme a equação (3).

(3)

Sendo *θ* o ângulo horizontal a partir do eixo *x* da imagem transladada ao centro do cone, *n* sendo o grau do polinômio para calcular o ângulo horizontal e *b0...bn* são os coeficientes do polinômio.

# 2.3. Modelo para transformação de imagens omnidirecionais em panorâmicas

As imagens omnidirecionais, para a obtenção de características e localização de objetos, podem também ser transformadas em imagens panorâmicas. As imagens panorâmicas permitem definir a posição do objeto dentro da cena, em relação ao veículo, dessa forma permitindo a tomada de decisão em relação a rotação a ser realizada.

As equações utilizadas para realizar nessa conversão são apresentadas em (4).

e (4)

Onde:

* + *x* e *y* são coordenadas da imagem panorâmica;
  + *xm* e *ym* são as dimensões da imagem omnidirecional;
  + *rm* é o raio do espelho;
  + *θ­­i* é o ângulo em radianos do pixel na imagem omnidirecional;
  + *hi* é a distância do centro ao pixel na imagem omnidirecional.

Para determinar as dimensões da imagem panorâmica que será gerada, o valor da altura (ymax) pode ser escolhido e o da largura (xmax) é calculado por meio da equação (5). A altura da imagem panorâmica, sem considerar fator de escala, é a distância entre o centro e a borda do espelho medido na imagem.

(5)

# 2.4. Detecção de objetos na imagem

Há várias técnicas de detecção de objetos. Geralmente a saída destes métodos gera uma caixa retangular que é chamada de *bounding box* e que serve para delimitar a região com maior confiança da imagem, ou seja, a mais provável de conter o objeto procurado. Essas técnicas podem utilizar redes neurais, ou trabalhar exclusivamente com processamento de imagem.

Considerando as redes neurais para detecção de objetos, há duas grandes famílias sendo elas: classificadores de regiões associadas a extratores de características baseados em Redes Neurais Convolucionais (CNN – *Convolutional Neural Network*) e CNN de disparo único (Wangenhein, 2018).

A principal diferença entre essas duas famílias é que a primeira adapta classificadores para a realização da detecção, aplicando um mesmo modelo em uma imagem várias vezes em vários locais e com várias delimitações diferentes dentro da imagem e as regiões com maior pontuação são consideradas regiões com objetos (Farhadi & Redmon, 2018). Na segunda, uma única rede neural é aplicada na imagem como um todo e assim a rede divide a imagem em regiões prevendo as caixas delimitadoras e a probabilidade para estas regiões.

# 2.5. Segmentação por limiarização e detecção de bordas

A segmentação é dita como o processo de divisão de uma imagem em duas regiões (fundo e objeto) para simplificar a representação da mesma facilitando sua análise. Ela é utilizada geralmente na detecção de objetos ou formas em imagens. Como resultado, a segmentação uma imagem binária, com região do objeto em preto e região do fundo em branco (ou o contrário).

A limiarização (*thresholding*) é um processo simples de segmentação e é feita por meio da detecção dos níveis de cores de uma imagem. Para detectar os níveis de cores é necessário realizar a transformação das cores da imagem para uma escala de cinza.

Na limiarização é necessário que a imagem esteja em tons de cinza, tornando mais simples limitar o gráfico de cores de uma linha de pixels. O procedimento se baseia na premissa de que pixels que pertencem a um mesmo objeto, terão tons de cinza muito próximos em uma imagem, assim sendo possível realizar a separação dos objetos e seu fundo.

Posteriormente neste processo, a partir de um limiar definido para cada figura, é possível atribuir a um pixel um entre dois valores, sendo eles acima do limiar ou abaixo do limiar. Esta classificação torna a imagem binária com dois valores possíveis para cada pixel, como mostra a equação (6), sendo *T* o limiar, (0) preto e (1) branco.

(6)

Uma borda é o contorno entre o objeto e seu fundo indicando o limite entre os dois. As bordas são picos de magnitude do gradiente na imagem, ou seja, variações significativamente discrepantes.

De acordo com Wangenheim (2000) os principais algoritmos de detecção de bordas são: Roberts, Sobel, Robinson, Canny e Marr-Hildreth. A maioria destes algoritmos possuem máscaras sobre os pixels da imagem para fim de detectar a descontinuidade da mesma. Cada pixel da imagem e seus vizinhos possuem um valor de nível de cinza que é multiplicado por uma constante. A soma destes valores é a máscara naquele ponto.

# 2.6. Trabalhos relacionados

Chen et al. (2009) também trabalha com o processamento de imagens omnidirecionais em tempo real para sistemas de vigilância e transformando as imagens coletadas em panorâmicas. No trabalho é estudada uma forma melhorada para o processamento de imagens que possa garantir que o processo seja mais rápido considerando o propósito do sistema. Tal implementação é baseada em FPGA (*Field-Programmable Gate Array* - matriz de portas programáveis) e propõe a utilização de sistemas embarcados para realizar os processamentos pesados que são requeridos no processamento da imagem omnidirecional para ser transformada em uma imagem panorâmica.

Rodrigues et al. (2015) descreve a utilização de um sistema catadióptrico para realizar levantamento de regiões por meio de uma imagem omnidirecional retificada gerando quatro imagens a partir da captada que representam os lados da visão do veículo. Posteriormente é feita uma segmentação para detectar as regiões de interesse extraindo o céu da imagem, por exemplo, em cada quadro gerado. O sistema foi projetado para auxiliar na obtenção de informações agrícolas de maneira barata.

Nobre (2021) também desenvolve um método para a retificação de imagens omnidirecionais para um plano, diminuindo as distorções e realizando a detecção de objetos presentes na imagem com a utilização de *Deep Learning*. Porém, há um ponto de diferenciação deste trabalho pois não se tratam de imagens omnidirecionais catadióptricas (captadas por meio de uma câmera e um espelho), mas sim de imagens captadas por câmeras especializadas. Portanto, o método desenvolvido pelo autor não pode ser utilizado no presente trabalho, já que não considera as distorções causadas pelo espelho cônico.

# 3. Desenvolvimento e resultados

Este projeto faz parte de um projeto de maior abrangência, e essa etapa propõe o desenvolvimento de parte da estação base do veículo não tripulado, que consiste em um módulo de visão computacional, que receberá as imagens da câmera omnidirecional catadióptrica e fará a geração de imagens panorâmicas e de vista aérea, como também a detecção de obstáculos a partir dessas imagens.

O projeto utilizará os seguintes materiais e ferramentas durante o seu desenvolvimento:

* Computador pessoal com processador i5 ou i7 com 8GB de memória RAM, 1TiB de HD;
* Imagens omnidirecionais catadióptricas obtidas com câmara de espelho cônico;
* Interpretador Python 3.0;
* Pycharm – ambiente de desenvolvimento para a linguagem Python;
* Biblioteca OpenCV para processamento de imagens em Python.

# 3.1. Atividades desenvolvidas

O projeto iniciou por um levantamento bibliográfico sobre o processamento de imagens digitais e detecção de objetos/obstáculos em imagens e vídeos, para o entendimento das técnicas utilizadas no processamento de imagens digitais.

Com base no levantamento bibliográfico e em uma análise das principais ferramentas utilizadas no processamento de imagens foi definida a linguagem Python em conjunto com algumas bibliotecas de processamento numérico e visão computacional.

Após a definição da ferramenta, foi realizado um estudo detalhado das imagens omnidirecionais catadióptricas, onde foram estudados os principais modelos e técnicas para a retificação das imagens omnidirecionais em imagens de vista aérea e a geração de imagens panorâmicas. E um estudo sobre as principais técnicas de detecção de obstáculos e localização de objetos em cenas estáticas capturadas a partir de um ambiente controlado.

Posteriormente foi implementada a retificação e geração de imagens panorâmicas, onde foram utilizados modelos empíricos em vez de modelos rigorosos, visto que a precisão das imagens geradas com a retificação utilizando modelos empíricos se aproximam muito à utilização de modelos rigorosos.

Após a implementação dos processos de retificação e geração de imagens panorâmicas, a partir de imagens catadióptricas e com o entendimento das técnicas de detecção de obstáculos e objetos, foi realizado um estudo e definição de um processo que utiliza as imagens panorâmicas e de vista aérea na detecção desses objetos em cenas estáticas e em sequências de cenas.

Como última etapa do projeto, foi criado um módulo (servidor) para a estação base do veículo não tripulado que recebe a imagem omnidirecional catadióptrica e retorna para o painel de controle imagens panorâmicas, de vista aérea e com marcações de obstáculos e objetos na imagem.

# 3.2. Preparação da imagem

As imagens captadas pelo sistema omnidirecional catadióptrico possuem, além da paisagem captada, a estrutura do sistema dentro da imagem. Neste sentido, tendo em vista que a câmera está apontada para um espelho, ela também capta a própria imagem.

Desta forma, todas as imagens captadas por este mesmo sistema omnidirecional catadióptrico terão em comum as regiões que mostram a estrutura do sistema. Sendo assim, é possível realizar um processamento em uma quantidade *N* de imagens e selecionar a região comum em todas, sendo feita uma máscara que pode ser aplicada nas demais imagens fazendo a interpolação dos pixels próximos a essa região e tornando a imagem livre destas áreas que podem influenciar negativamente no momento da detecção de objetos.

A forma utilizada para encontrar essas regiões na imagem, considerando que elas nem sempre tem as cores dos pixels idênticas por conta de interferências luminosas, é a análise do desvio padrão dos níveis de cinza entre pixels que se encontram na mesma localização. Por meio desta análise, pode-se concluir que os pixels que tem, em todas as imagens analisadas, mesmo com interferências de iluminação, possuem pouco desvio de cores.

O processo de cálculo do desvio padrão de cada pixel em uma quantidade *N* de imagens adquiridas com o sistema gera uma outra matriz com as mesmas dimensões da imagem que possui em cada posição o desvio padrão das cores relacionadas àquela posição do pixel nas *N* imagens. Então, a matriz com os desvios é normalizada para obter valores que estejam compreendidos entre 0 e 255 (padrão para imagens com apenas um canal).

A partir desta imagem gerada (de desvio padrão), é possível aplicar limiarização na mesma para que seja obtida uma imagem binária (apenas preto e branco, sem níveis). Neste processo, é obtida a imagem mostrada na Figura 3. Para a geração desta imagem, também é aplicado um processo de dilatação para que os pixels dispersos também sejam incluídos na máscara.

**Figura 3. Máscara obtida por meio do processo de limiarização da imagem de desvio padrão normalizada.**

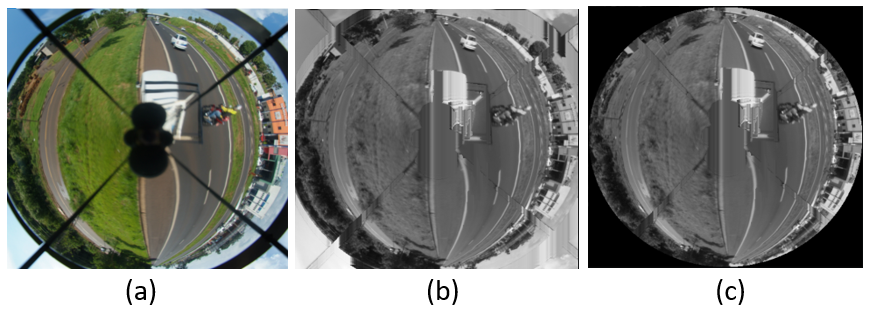


Fonte: Autores.

Por fim, a imagem binária é utilizada como máscara para as imagens que serão processadas, sabendo que são obtidas do mesmo sistema omnidirecional catadióptrico. O preenchimento destes espaços nas imagens é feito a partir de informações obtidas de pixels vizinhos.

Com o método relacionado à geração e aplicação de máscara nas imagens, para minimizar os impactos da estrutura do sistema aparente na imagem, é possível realizar a aplicação de máscara neste momento antes da geração da imagem panorâmica. A Figura 4 (a) mostra uma imagem omnidirecional catadióptrica captada pelo sistema.

**Figura 4. (a) imagem omnidirecional captada. (b) imagem com aplicação da máscara gerada. (c) imagem com aplicação da máscara e remoção das bordas não utilizadas.**



Fonte: Autores.

A Figura 4 (b) mostra a imagem com a máscara já aplicada. No entanto, algumas partes da imagem não são utilizadas no momento da geração da imagem panorâmica. Isto ocorre porque a câmera voltada para o espelho pode captar partes que não estão relacionadas ao espelho.

Sendo assim, é necessário realizar a retirada destas partes para que só a região útil fique visível. A Figura 4 (c) mostra a imagem já sem a região que não será utilizada. Esta região compreende a tudo que está fora do raio do espelho.

Ainda é possível observar na Figura 4 (c) alguns elementos do sistema presentes na imagem. Estes elementos permanecem, pois algumas imagens possuem iluminação diferente nestes pontos e o algoritmo desenvolvido não foi capaz de captar e adicionar estes pontos na máscara.

# 3.3. Transformação da imagem omnidirecional catadióptrica em panorâmica

A implementação foi feita utilizando a linguagem Python e a biblioteca OpenCV para manipulação e processamento de imagem. A Figura 5 mostra a imagem omnidirecional captada por um sistema omnidirecional catadióptrico, concedida por Olivete sobre seu trabalho desenvolvido em 2014.

**Figura 5. Imagem capturada com o sistema omnidirectional catadióptrico.**



Fonte: Olivete (2014).

A máscara pode ser aplicada antes ou depois do processo de retificação da imagem, no entanto, para que a máscara seja gerada corretamente, é necessário que as imagens utilizadas estejam no mesmo padrão. Ou seja, se a aplicação for antes da retificação, as imagens precisam ser omnidirecionais e capturadas pelo sistema, caso seja aplicada depois, as imagens para geração da máscara precisam já estar retificadas.

Para a geração desta imagem, os passos são:

* Criar uma imagem panorâmica sem preenchimento, respeitando a proporção descrita na equação (5);
* Utilização da equação (4) para encontrar o pixel correspondente da imagem omnidirecional para cada pixel da imagem panorâmica gerada;
* Realizar a translação da coordenada do pixel da imagem catadióptrica do centro da imagem para o canto superior esquerdo;
* Preencher o pixel da imagem panorâmica com o pixel correspondente da imagem catadióptrica.

A Figura 6 mostra a imagem panorâmica gerada a partir da Figura 5 por meio do processamento descrito. Como dito na seção anterior, alguns dos elementos do sistema omnidirecional permanecem, mas já é possível ter uma ampla visualização de todo o campo proposto na imagem.

**Figura 6. Imagem panorâmica gerada.**



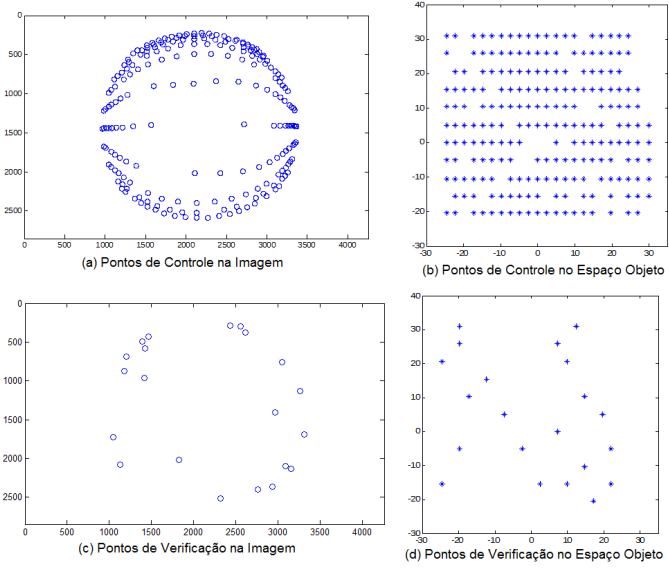
Fonte: Autores.

# 3.4. Transformação da imagem omnidirecional catadióptrica em imagem de vista aérea

Para o processo de calibração é necessário um conjunto de coordenadas medidos no espaço objeto e suas respectivas coordenadas no espaço imagem. Para isso foi necessária a obtenção de uma imagem omnidirecional catadióptrica com o sistema catadióptrico com as mesmas configurações para as demais imagens capturadas. O campo de calibração é uma área plana com marcações no solo. Para esta imagem captada, foi necessário também a captação de medições relativas às marcações feitas no solo.

O projeto de referência utilizado foi o de Olivete (2014), onde foram medidas manualmente as distâncias entre os 210 pontos de controle marcados no local e seus correspondentes na imagem omnidirecional captada. Destes pontos, 21 foram escolhidos para serem os pontos de verificação.

**Figura 7 .** **Distribuição dos pontos de controle e verificação.**



Fonte: Olivete (2014).

A Figura 7 mostra a relação entre os pontos de controle e de verificação no espaço objeto e do espaço imagem. Os pontos de verificação não se encontram no centro da imagem, porém não é necessário pois esta região da imagem possui o sistema físico e será tampada pela máscara.

Para realizar o processo de estimação de parâmetros a partir dos pontos coletados no espaço objeto e no espaço imagem (onde já se sabe o seu correspondente), foi utilizado o modelo das coordenadas polares com polinômios de grau 12 para os parâmetros de distancias e de grau 18 para os parâmetros de ângulos, utilizando as equações (2) e (3).

Para a retificação da imagem, foi utilizada a aplicação das equações (2) e (3) em que são ajustados tanto o ângulo quanto a distância, mas com os parâmetros já calculados.

Na retificação, cada ponto da imagem teve seu centro relativo transformado para o centro real da imagem, considerando que o ponto (0,0) da imagem para processamento se encontra no canto superior esquerdo.

O ponto em coordenadas cartesianas foi transformado para coordenadas polares para que as equações possam ser utilizadas. Assim, tendo-se o ângulo e a distância do ponto na imagem que deve ser gerada, foi possível calcular qual é o ponto em coordenadas polares correspondente na imagem omnidirecional catadióptrica e assim capturar a cor presente no pixel e replicá-la na imagem que foi gerada.

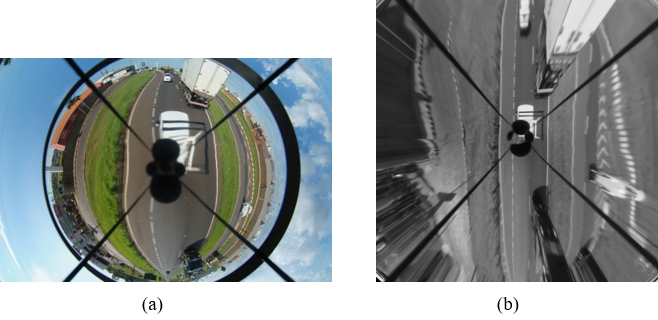
Para a reamostragem da imagem são necessários os seguintes passos:

* Criar uma imagem vazia com os níveis de cinza em 0 (zero), que servirá de base para a retificação (imagem retificada);
* Realizar a translação da coordenada do pixel da nova imagem (que se encontra inicialmente no canto superior esquerdo) para o centro;
* Transformar as coordenadas cartesianas para coordenadas polares (*θ*, *r*) da imagem retificada;
* Aplicar as fórmulas (2) e (3) para obtenção do (*Θ*, *R*) correspondentes à imagem omnidirecional catadióptrica;
* Transformar as coordenadas polares (*Θ*, *R*) da imagem omnidirecional em coordenadas cartesianas;
* Determinar o nível de cinza, pois as equações utilizadas geram números não inteiros, dessa forma não resultando em um pixel na imagem catadióptrica, mas em um pixel aproximado. Dessa forma é necessário utilizar algum método de interpolação para determinar o nível de cinza desse pixel na imagem resultante;
* Por fim, realizar o preenchimento do pixel na imagem resultante com o valor determinado pelo método de interpolação.

O ponto em coordenadas cartesianas foi transformado para coordenadas polares para que as equações possam ser utilizadas. Assim, tendo-se o ângulo e a distância do ponto na imagem que deve ser gerada, foi possível calcular qual é o ponto em coordenadas polares correspondente na imagem omnidirecional catadióptrica e assim capturar a cor presente no pixel e replicá-la na imagem que foi gerada.

A Figura 8 (a) mostra uma imagem omnidirecional catadióptrica utilizada para o teste do algoritmo e a Figura 8 (b) mostra o resultado obtido após a retificação.

**Figura 8. (a) Imagem omnidirecional catadióptrica capturada pelo sistema. (b) Imagem retificada.**



Fonte: Autores.

# 3.5. Detecção de objetos na imagem

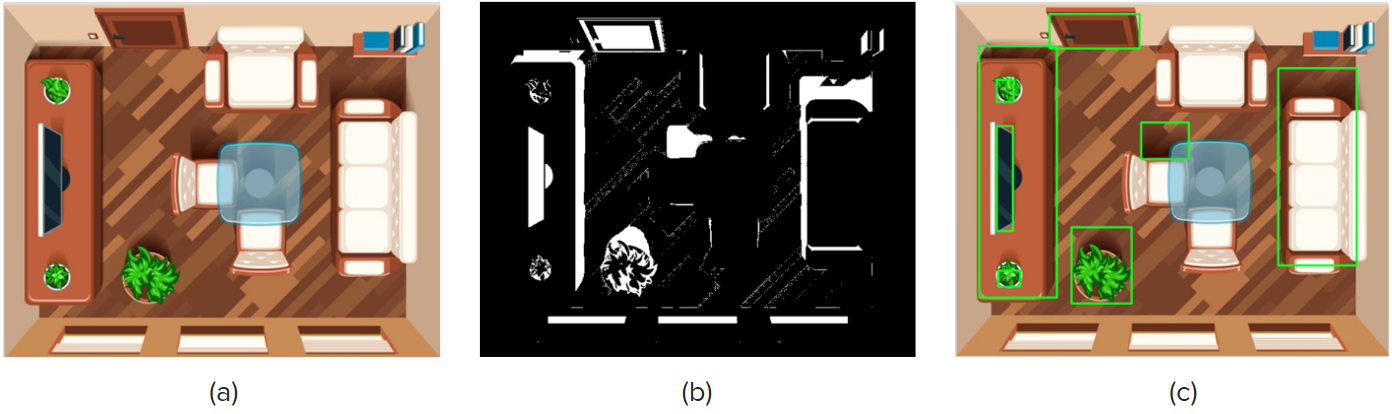
A Figura 9 (b) mostra uma imagem com a aplicação da técnica de *thresholding*. A partir da definição do limiar *T*, o algoritmo classificou cada pixel como sendo preto ou branco o relacionando com o valor, como mostrado na equação (6).

A vantagem de se obter essa imagem binária é que reduz a complexidade da imagem melhorando o processo de reconhecimento de objetos na cena.

Por meio desta limiarização é possível obter contornos bem definidos na imagem e, assim, são detectados possíveis objetos de acordo com o contorno, sendo mais notável a transição de cores diferentes na borda de um objeto.

Quando a distribuição de intensidade entre o objeto e o fundo são diferentes o suficiente, é possível utilizar apenas um limiar global que é aplicável a toda a imagem. No entanto, encontrar o limiar não é uma tarefa trivial e para este processo pode ser usada uma estratégia iterativa ou heurística.

**Figura 9. (a)Figura a ser processada (b)Imagem com a aplicação da limiarização (c)Detecção de objetos.**

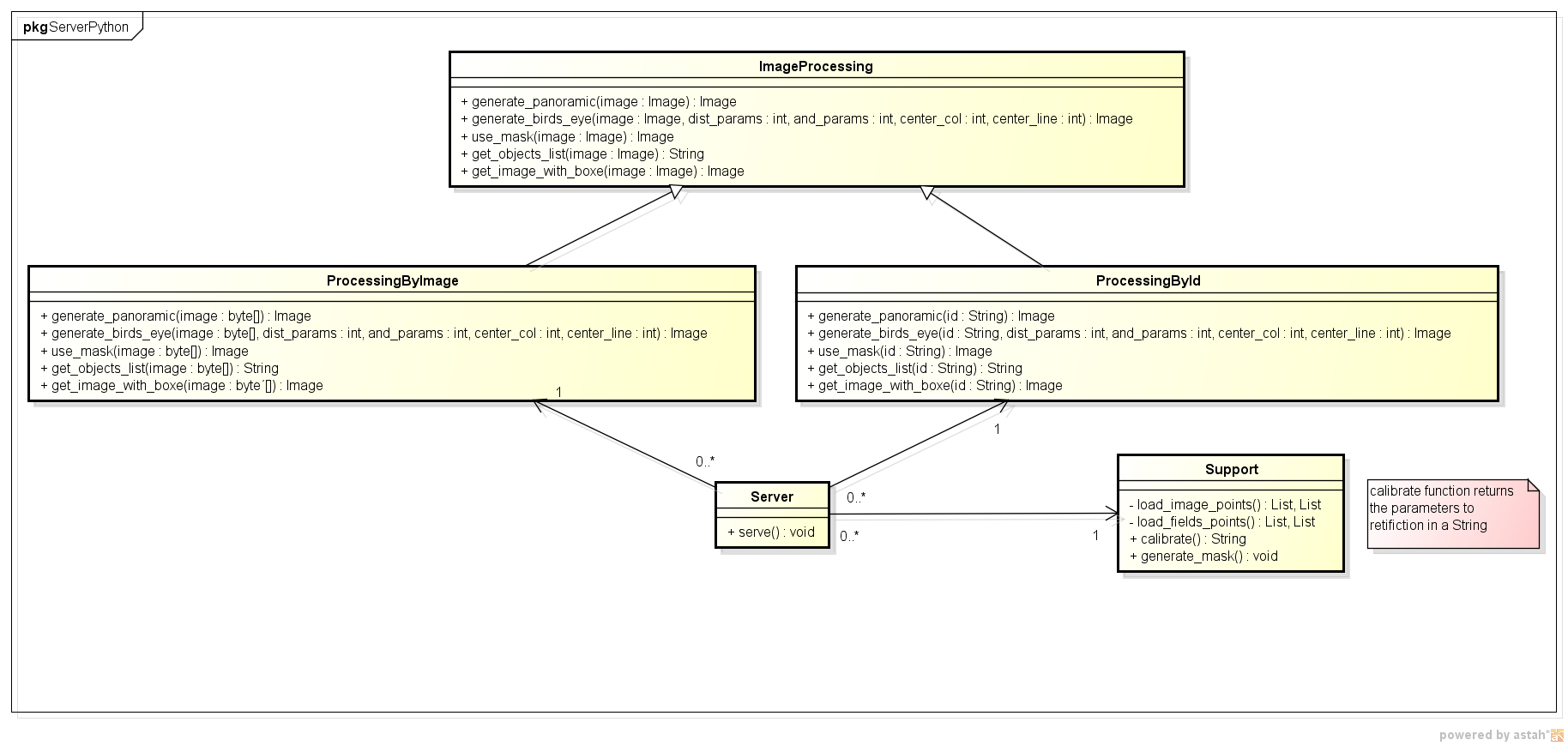


Fonte: Autores.

# 3.6. Servidor de processamento de imagem

O servidor foi implementado em Python e a comunicação é realizada por meio do protocolo de chamada de procedimento remoto XML-RPC. O protocolo XML-RPC utiliza XML para codificação das chamadas de procedimento, permitindo que qualquer software faça chamada de procedimentos remotos em diferentes ambientes, essas chamadas utilizam o protocolo HTTP (Mari, 2004).

**Figura 10. Diagrama de classe do servidor implementado.**



Fonte: Autores.

Na linguagem Python, existe uma biblioteca para utilização do XML-RPC. Sendo assim, as partes em XML são feitas pela biblioteca e as funções ficam encapsuladas facilitando a utilização do protocolo.

A Figura 10 representa a arquitetura do servidor desenvolvido em Python. Existe a classe *ProcessingById*, que processa as imagens que já estão registradas no servidor por meio do id passado por parâmetro. A classe *ProcessingByImage* tem as mesmas funções da classe *ProcessingById*, porém é necessário que seja passada por parâmetro a imagem completa em bytes.

Outra importante classe de processamento presente no servidor é a classe *Support*, que tem por funções: realizar a calibração do sistema por meio dos dados de pontos no terreno e na imagem presentes na pasta “*calibration*” no servidor e também gerar a máscara utilizada com as imagens presentes na pasta “*omni\_images*”. A função de geração de máscara presente na classe *Support* também utiliza as imagens desta pasta para gerar a máscara que é utilizada para remover os elementos dos componentes físicos do sistema capturados pela imagem.

# 3.7. Biblioteca de classes Java para encapsulamento do processo

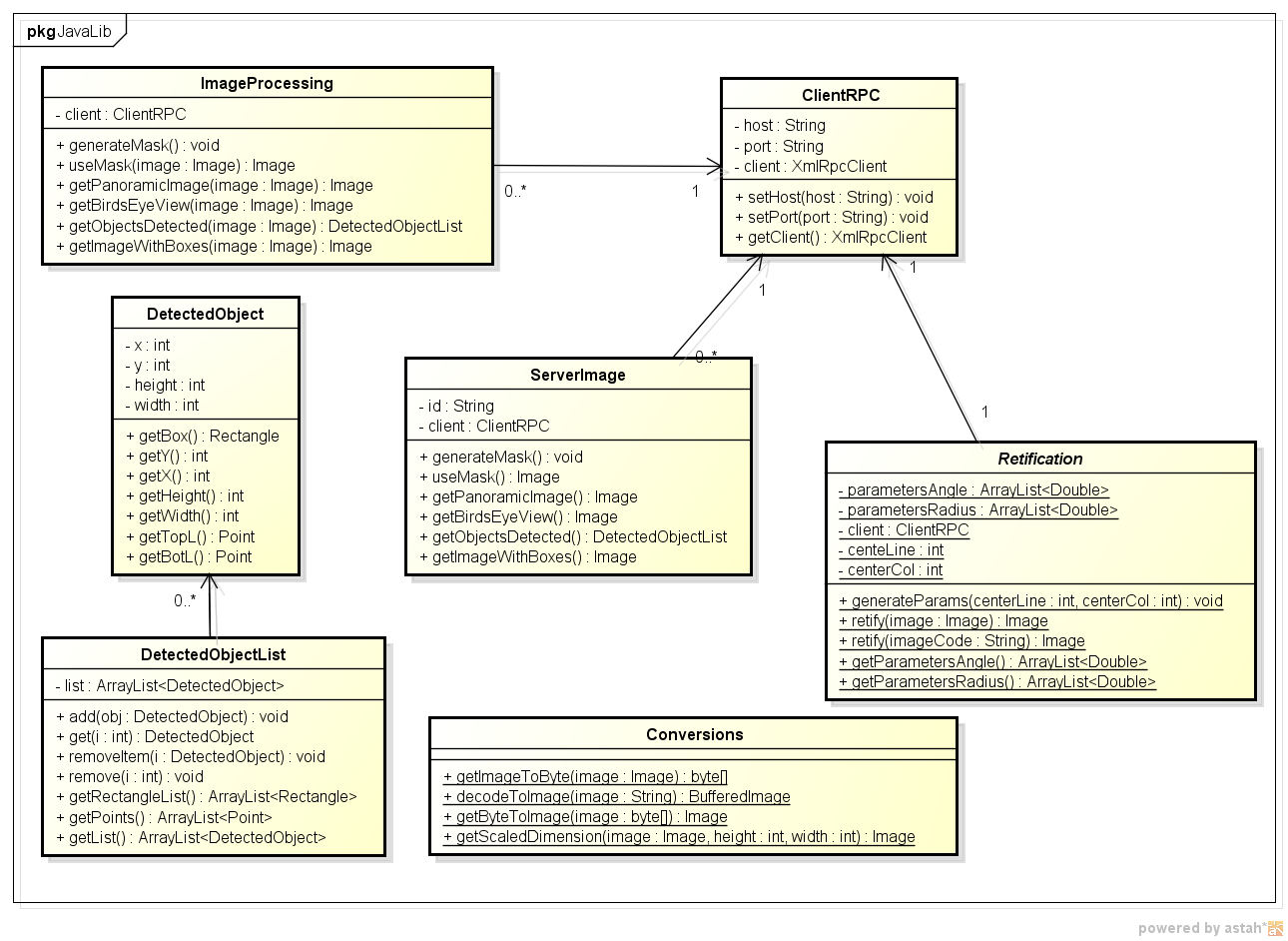
Com o objetivo de otimizar a utilização do servidor de processamento de imagens, foi construído um conjunto de classes em Java para encapsular os protocolos de comunicação utilizados pelo servidor. Sendo assim, essas classes possuem métodos que realizam o envio das imagens e retornam para o cliente a imagem processada pelo servidor. Tornando transparente a comunicação entre a aplicação cliente e o servidor responsável pelos processamentos realizados na imagem.

A Figura 11 apresenta o diagrama de classes da biblioteca desenvolvida em Java para encapsular a comunicação entre o cliente e o servidor.

Foram criadas duas classes de processamento de imagem. Em uma delas (*ImageProcessing*), é possível passar a imagem por parâmetro para as funções que a enviam para o servidor e devolvem a imagem processada ou os dados requisitados.

No caso da classe *ServerImage*, para a criação de uma instancia da mesma, é necessária a passagem da imagem já na criação do objeto. Neste momento, a imagem já é enviada ao servidor e um id é gerado e retornado. A classe armazena o id que é o identificador da imagem no servidor, sendo assim, para a chamada das demais funções, não é mais necessário enviar toda a imagem para o servidor sempre que for necessário um processamento, pode-se apenas enviar o id da imagem que já está salva no servidor.

**Figura 11. Diagrama de classe da biblioteca criada para utilização do servidor.**



Fonte: Autores.

A biblioteca também possui uma classe de retificação, chamada “*Retification*”. Nesta, são armazenados os parâmetros necessários para a retificação das imagens do sistema omnidirecional catadióptrico em questão. A classe é composta por um método de calibração e dois métodos de retificação, um por imagem e um por id. Os métodos de retificação são chamados nas classes *ServerImage* e *ImageProcessing* para processar as imagens.

Para a questão da detecção de objetos na imagem, é possível também fazer esta detecção por uma das duas classes de processamento de imagem. Elas encapsulam os objetos encontrados em *DetectedObject* que, por sua vez, compõem uma lista encapsulada *DetectedObjectList*.

No *DetectedObject* é possível obter as medidas *x*, *y*, *width* e *height*. O ponto (*x,y*) corresponde ao ponto do canto superior esquerdo da imagem, assim sua altura (*height*) e largura(*width*) partem deste ponto. Também é possível obter um objeto da classe Point padrão da linguagem Java, além de um objeto da classe *Rectangle*. Da classe *DetectedObjectList* é possível obter a lista de Points e também lista de *Rectangles*, além da própria lista de *DetectedObjects*.

Por último, a classe *ClientRPC* é a responsável pela conexão com o servidor (em host e porta definidos). As classes podem utilizar uma instância dela para executar as funções necessárias.

# 3.8. Testes realizados

Uma aplicação básica foi construída em Java utilizando a biblioteca de classes para simular o envio de uma imagem omnidirecional catadióptrica e receber a imagem transformada e processada pelo servidor, mostrado na Figura 12.

**Figura 12. Interface da aplicação cliente.**



Fonte: Autores.

Nos testes realizados com essa aplicação, a resposta do servidor para processamento de imagens de 4256x2848px está registrado na Tabela 1 e 2. Este tempo é relativamente alto para aplicações em tempo real, considerando que o tempo de comunicação entre o cliente e o servidor se mostrou em uma média de 0.02 segundos e o processamento da imagem ocupou o restante do tempo, mas deve-se considerar que as imagens utilizadas para os testes foram captadas com câmera de alta qualidade, que não é recomendado para a aplicação em que o servidor será realmente utilizado.

**Tabela 1. Relação entre o tempo e os processos disponíveis no servidor**

|  |  |  |
| --- | --- | --- |
| **Processo** | **Tempo enviando a imagem** | **Tempo utilizando imagem existente no servidor** |
| Geração de imagens panorâmicas | 27.902 s | 23.119 s |
| Geração de imagens de vista aérea | 226.988 s | 223.816 s |
| Detecção de objetos retornando imagem | 3.294 s | 1.253 s |
| Detecção de objetos retornando lista de objetos | 2.195 s | 0.256 s |
| Utilizar máscara | 163.944 s | 146.742 s |

**Tabela 2. Relação entre o tempo e os processos auxiliares disponíveis no servidor**

|  |  |
| --- | --- |
| **Processo** | **Tempo** |
| Calibração do sistema | 0.047 s |
| Geração de máscara | 23.153 s |
| Registrar imagem no sistema | 3.102 s |

A Tabela 1 mostra o tempo medido com relação ao processamento das imagens considerando as funções enviando as imagens inteiras a cada processamento e também considerando o envio apenas do identificador da imagem que já está previamente registrada no servidor.

É possível observar uma diferença significativa ao enviar a imagem inteira comparado ao envio apenas do identificador da imagem já salva no servidor. A Tabela 2 mostra o tempo de processamento dos processos auxiliares que devem ser feitos antes dos processos descritos na Tabela 1. Estes processos são relativamente simples comparados aos processos da Tabela 1, mas são de suma importância para o funcionamento do servidor.

# 4. Considerações finais

Os sistemas omnidirecionais catadióptricos podem ser uma alternativa para a implementação da navegação de veículos autônomos. A necessidade de apenas uma câmera e um espelho fazem com que o sistema tenha um baixo custo de implementação.

Para a utilização efetiva das imagens captadas pelo sistema, é necessário que as imagens sejam processadas, para que as distorções geradas pelo sistema de captura de imagens omnidirecionais catadióptricas sejam amenizadas.

Com a revisão bibliográfica realizada nesse trabalho foi possível verificar que os modelos empíricos, para o problema em questão, possuem resultados muito próximos aos modelos rigorosos, tornando a implementação mais compreensível devido a menor complexidade desses modelos, como também o processo mais rápido.

Com o desenvolvimento do trabalho foi possível verificar que partes da imagem não são úteis para a detecção dos obstáculos, foco principal desse trabalho, dessa forma foi necessário realizar um processamento prévio da imagem antes do processo de retificação com o objetivo de eliminar tudo aqui que não é de interesse o projeto, como os elementos físicos do sistema omnidirecional, como também a área fora do espelho cônico.

Também estudou-se os vários métodos para detecção e classificação de objetos na imagem. Sendo que nessa implementação optou por utilizar a limiarização e detecção de bordas para selecionar os objetos nas imagens processadas por não haver necessidade de classificação dos objetos detectados, porém poderiam ser utilizadas redes neurais para a classificação dos objetos na cena, porém não é o foco do projeto.

Com relação à implementação, foi desenvolvido um servidor que recebe as imagens omnidirecionais catadióptricas e realiza as tarefas de retificação para vista aérea, transformação para vista panorâmica e a o processo de calibração do modelo. Também foi desenvolvida uma biblioteca Java para auxiliar a utilização dos processos disponíveis no servidor sem que o cliente se preocupe com a comunicação, tornando-a transparente.

# References

FARHADI, A.;REDMON, J. YOLOv3: an incremental improvement. Computer Vision and Pattern Recognition. University of Washington. 2018.

GASPAR, J. A. C. P. Omnidirectional vision for mobile robot navigation. IST-Universidade Técnica de Lisboa, p. 150, 2002.

NOBRE, Francisco Lucas Sousa. Detecção e remoção automática de objetos indesejados em imagens utilizando aprendizagem profunda. 2021. Mestrado em Engenharia Informática - Computação Móvel.

MARI, A. D. Desenvolvimento de um servidor HTTP/XML-RPC. 2004. Trabalho de Conclusão de Curso. Universidade Federal de Santa Catarina. Centro Tecnológico. Curso de Ciências da Computação.

OLIVETE, A. L. Utilização de imagens omnidirecionais georreferenciadas como controle de campo para orientação de imagens orbitais. 2014. Tese (doutorado) - Universidade Estadual Paulista, Faculdade de Ciências e Tecnologia. 186p.

OLIVETE, A. L.; TOMMASELLI, A. M. G. Utilização de modelos empíricos para a retificação de imagens omnidirecionais catadióptricas. 2016. Revista Brasileira de Cartografia, v. 68, n. 7.

PEREIRA, F. G.; GAVA, C. C.; VASSALLO, R. F.; SARCINELLI-FILHO, M. Calibração de sistemas catadióptricos e detecção da pose de robôs móveis por segmentação de imagens omnidirecionais. Proceedings of VII SBAI-Simpósio Brasileiro de Automação Inteligente, 8p., São Luís, Brasil, 2005.

SPACEK, L. Omnidirectional perspective and stereopsis with conical mirrors. Research Report CTU–CMP–2003–12, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, 2003.

SPACEK, L.; BURBRIDGE, C. Instantaneous robot self-localization and motion estimation with omnidirectional vision. Robotics and Autonomous Systems, v. 55, n. 9, p. 667-674, 2007.

WANGENHEIM, A. V. Introdução à visão computacional. Encontrando a Linha Divisória: Detecção de Bordas. 2000. p. 9 – 24.