PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E PROCESSOS INDUSTRIAIS – MESTRADO
ÁREA DE CONCENTRAÇÃO EM CONTROLE E OTIMIZAÇÃO DE PROCESSOS INDUSTRIAIS

Charles Varlei Neu

Desenvolvimento de um sistema de reconhecimento de sinais de trânsito utilizando processamento de imagens com OpenCV para um robô humanoide

Santa Cruz do Sul
2014

Charles Varlei Neu

Desenvolvimento de um sistema de reconhecimento de sinais de trânsito utilizando processamento de imagens com OpenCV para um robô humanoide

Dissertação apresentada ao Programa de Pós-Graduação em Sistemas e Processos Industriais – Mestrado, Área de Concentração em Controle e Otimização de Processos Industriais, Universidade de Santa Cruz do Sul – UNISC, como pré-requisito parcial para a obtenção do Título de Mestre em Sistemas e Processos Industriais.

Orientadores: Prof. Dr. Leonel Pablo Tedesco
Prof. Dr. Rolf Fredi Molz

Santa Cruz do Sul
2014

Charles Varlei Neu

Desenvolvimento de um sistema de reconhecimento de sinais de trânsito utilizando processamento de imagens com OpenCV para um robô humanoide

Esta dissertação foi submetida ao Programa de Pós-Graduação em Sistemas e Processos Industriais – Mestrado, Área de Concentração em Controle e Otimização de Processos Industriais, Universidade de Santa Cruz do Sul – UNISC, como requisito parcial para a obtenção do Título de Mestre em Sistemas e Processos Industriais.

Prof. Dr. Leonel Pablo Tedesco
Orientador

Prof. Dr. Rolf Fredi Molz
Coorientador

Prof. Dr. Ricardo Melo Czekster
Examinador - UNISC

Prof. Dr. Rodrigo da Silva Guerra
Examinador - UFSM

Santa Cruz do Sul

2014

AGRADECIMENTO

Agradeço aos meus amigos e familiares pelo incentivo; aos professores e colegas do Programa de Pós-Graduação em Sistemas e Processos Industriais da UNISC pelos ensinamentos, pela amizade e por todo auxílio e apoio prestado; e aos meus professores orientadores, pela sabedoria transmitida e pelo encorajamento na realização de trabalho.

Também agradeço à FAPERGS (Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul) pelo apoio financeiro através de bolsa de estudos.

Agradeço ainda, à *Ostfalia Hochschule für angewandte Wissenschaften*, da Alemanha, que me proporcionou grande aprendizado durante intercâmbio acadêmico de um semestre no em seu programa de mestrado em Ciência da Computação, bem como também, às equipes de assessoria para assuntos internacionais e interinstitucionais da UNISC e da *Ostfalia* que auxiliaram no processo de intercâmbio.

E a todos aqueles que contribuíram na busca deste ideal, meus mais profundos e sinceros agradecimentos.

RESUMO

Esta dissertação apresenta um método de reconhecimento de faixas de travessia de pedestres, aliado aos conceitos teóricos necessários para elaborar um sistema de visão computacional para reconhecer padrões e identificar objetos do mundo real. São utilizadas funções da biblioteca de processamento de imagens OpenCV (*Intel Open Source Computer Vision Library*) para o desenvolvimento deste sistema, que foi projetado para ser executado em um robô humanoide DARwIn-OP (*Dynamic Anthropomorphic Robot with Intelligence - Open Platform)*. Para a avaliação e validação do sistema, foram feitos testes em um ambiente real de trânsito, considerando diferentes condições de visibilidade e luminosidade. Os resultados mostram a eficiência do uso de funções OpenCV para este tipo de sistemas, que exigem um tempo de resposta rápida, em um hardware com capacidades, principalmente de memória e processamento, reduzidas.

Palavras Chave: OpenCV, Visão Computacional, DARwIn-OP, sinais de trânsito.

ABSTRACT

This dissertation presents a recognition method of pedestrian crossing, coupled with the necessary theoretical bases and concepts required to develop a computer vision system to recognize patterns and identify real-word objects. Functions of the OpenCV (*Intel Open Source Computer Vision Library*) *image-processing library are used to develop this system, which was designed to run on a DARwIn-OP* (*Dynamic Anthropomorphic Robot with Intelligence - Open Platform) humanoid robot. Tests were performed in a real traffic environment, in different conditions of visibility and brightness, for system evaluation and validation. Results show how efficient OpenCV functions are in this kind of system, which requires a fast response time on hardware with reduced capabilities, especially memory and processing power.*

Keywords: *OpenCV, Computer Vision, DARwIn-OP, Traffic Signs.*

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

2D	Bidimensional, planar ou de duas dimensões
3D	Tridimensional, espacial ou de três dimensões
BLOB	*Binary Large Object*
CONTRAN	Conselho Nacional de Trânsito
CV	*Computer Vision*
DARPA	*Defense Advanced Research Projects Agency*
DARwIn-OP	*Dynamic Anthropomorphic Robot with Intelligence - Open Platform*
DSP	*Digital Signal Processor*
FPGA	*Field Programmable Gate Array*
FTP	Faixa de Travessia de Pedestres
GPL	*General Public License*
GUI	*Graphical User Interface*
HDMI	*High-Definition Multimedia Interface*
HIGHGUI	*High-level Grafical User Interface*
HSI	*Hue, Saturation and Intensity*
HSV	*Hue, Saturation and Value*
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	*Integrated Development Environment*
LF	Linhas de Fuga
LH	Linha do Horizonte
MLL	*Machine Learning Library*
OpenCV	*Intel Open Source Computer Vision Library*
PDI	Processamento Digital de Imagens
PF	Ponto de Fuga
Pixel	*Picture element*
PV	Ponto de Vista
RAM	*Random-access Memory*
RGB	*Red, Green and Blue*
ROI	*Region of Interest*
UNISC	Universidade de Santa Cruz do Sul
USB	*Universal Serial Bus*
YCbCr	Sistema de cores empregado pelos padrões JPEG e MPEG

YUV *Luminance (Y), blue–luminance (U), red–luminance (V)*

LISTA DE SIGLAS

cm Centímetros

GB *Gigabyte*

GHz *Gigahertz*

IEEE *Institute of Electrical and Electronics Engineers*

MP Megapixels

ms Milissegundos

nm Nanômetros

1 INTRODUÇÃO

A computação está presente em nosso dia a dia para a realização e o auxílio às mais diversas tarefas. Uma das áreas onde cientistas concentram grandes esforços é a robótica, desenvolvendo equipamentos e sistemas autônomos que simulam as habilidades e capacidades humanas. Desta forma, tais equipamentos e sistemas podem ser utilizados para realizar tarefas pertinentes a seres humanos, como por exemplo, nas indústrias, em especial em tarefas críticas e que apresentam risco à integridade física e/ou à saúde de operadores humanos (WANG, 2008; SOUSA, 2007; PEDRINI, 2007; KOSCHAN,2008; FAWCETT,2006).

Entretanto, existem muitos desafios nessa área, como por exemplo, construir sistemas computacionais capazes de simular a capacidade humana de analisar imagens e nelas encontrar informações relevantes, interpretando-as em um tempo pertinente à função. Um sistema de visão computacional deve ser capaz de extrair apenas as informações importantes ou de interesse em um cenário, a partir de imagens, desconsiderando informações irrelevantes para permitir maior eficiência e simplicidade nas análises (GONZALEZ, 2002; GALVÃO FILHO, 2008; SOUSA, 2007; LU, 2008; KOSCHAN, 2008; FAWCETT, 2006; PRATT, 2007).

Um sistema de visão computacional para robôs industriais deve ser projetado visando as limitações na capacidade de processamento e armazenagem de dados que esse tipo de *hardware* possui. A crescente evolução desse *hardware*, com incremento principalmente no poder de processamento e armazenagem de dados, proporciona a implementação de sistemas cada vez mais robustos e eficientes, com respostas em rápidas. Existem várias bibliotecas de PDI (Processamento Digital de Imagens) que oferecem funções otimizadas que podem ser utilizadas nesses sistemas, entre as quais pode-se destacar a biblioteca OpenCV (OpenCV, 2013). Entretanto, é necessário avaliar o desempenho do sistema construído com o uso dessas bibliotecas em tais *hardwares* (LU, 2008; KLAWONN, 2005; GRAVILLA, 2005; OpenCV, 2013; WANG, 2008; SOUSA, 2007; KOSCHAN, 2008; FAWCETT, 2006; CHARETTE, 2009; BURGHOUTS, 2009).

1.1 Tema

O foco desta pesquisa é estudar técnicas de PDI para reconhecimento de padrões, com o objetivo de encontrar informações úteis em imagens digitais. Dentre essas informações,

destacam-se a detecção de objetos de interesse em um cenário de trânsito em um ambiente real a partir de imagens do mesmo.

Junto ao tema dessa pesquisa acrescenta-se a avaliação de desempenho desse sistema em *hardwares* específicos, com processamento e capacidade de memória reduzidos em relação a um microcomputador convencional. Essas abordagens serão aplicadas ao domínio do sistema principal de visão computacional de um robô do tipo humanoide modelo DARwIn-OP.

O presente trabalho inicia um processo de estudos para desenvolver um sistema de visão computacional para controlar robôs autônomos. A partir dos resultados deste, podem ser direcionados projetos futuros para construir um sistema completo de visão computacional para tais equipamentos.

1.2 Problema

Há décadas, cientistas buscam construir máquinas que possam ser utilizadas para auxiliar ou até mesmo substituir o trabalho humano, principalmente em ambientes hostis ou em que a rapidez, a precisão e a corretude na realização das tarefas são essenciais. Com o aumento dos recursos computacionais (de *hardware*), especialistas estão desenvolvendo sistemas para controlar máquinas cada vez mais eficientes e capazes de realizar tarefas complexas de forma autônoma (LU, 2008; GRAVILLA, 2005; OpenCV, 2013; WANG, 2008; KOSCHAN, 2008; FAWCETT, 2006; CHARETTE, 2009; BURGHOUTS, 2009).

As técnicas de PDI apresentam um papel fundamental para possibilitar a construção de sistemas de visão computacional para tais máquinas. Cientistas e pesquisadores vêm desenvolvendo técnicas cada vez mais eficientes e precisas para possibilitar a interação de máquinas no ambiente real de forma autônoma a partir de imagens obtidas por suas câmeras (GONZALEZ, 2002; KOSCHAN, 2008; FAWCETT, 2006; CHARETTE, 2009; BURGHOUTS, 2009).

Entretanto, muitos desafios ainda existem para possibilitar a identificação de objetos do mundo real a partir de imagens digitais em um tempo de resposta rápido. O grande número de detalhes e informações que existem nas imagens, distorções, diferenças de luminosidade, descontinuidades, mudanças de posição, tamanho, orientação, variações ambientais, são apenas alguns fatores e/ou aspectos que representam grandes desafios no PDI.

Um sistema de visão computacional eficiente deve ser capaz de fazer inferências a partir de informações incompletas e também ser capaz de identificar objetos ou características

em uma imagem independentemente desses fatores. Além disso, deve apresentar os resultados do processamento de acordo com os requisitos de tempo do sistema desenvolvido (GONZALEZ, 2002; LU, 2008; GRAVILLA, 2005; OpenCV, 2013; WANG, 2008; KOSCHAN, 2008; FAWCETT, 2006; CHARETTE, 2009; BURGHOUTS, 2009).

1.3 Justificativa

Os veículos autônomos, ou seja, aqueles que são capazes de se autodirigir em uma estrada qualquer ou por ruas de cidades, já estão se tornando uma realidade. Nos estados da Califórnia, Nevada e Flórida, nos Estados Unidos da América, o governo legalizou a circulação desse tipo de veículos em suas estradas, desde que acompanhados por um motorista humano habilitado a dirigir. Grandes empresas investem no desenvolvimento de tecnologias que possibilitem a construção de tais veículos. Nestes veículos, um sistema de visão computacional que interprete corretamente as imagens, não apenas identificando sinais de trânsito, mas estradas, pessoas e objetos em geral, em tempo real, é fundamental (THRUN, 2006; KEVICZKY, 2007; GALLO 2008; ESS, 2009).

Em um grande desafio tecnológico, organizado pela DARPA (*Defense Advanced Research Proj ects Agency*) em 2005, foram reunidos pesquisadores, cientistas, engenheiros e grandes fabricantes de veículos de todo o mundo para uma competição. O objetivo era desenvolver um "carro-robô" capaz de atravessar um deserto de forma autônoma, no menor tempo possível. Nessa competição foi ofertado ao vencedor um grande prêmio em dinheiro (THRUN, 2006).

No campo da computação assistiva, em especial no desenvolvimento de sistemas e dispositivos para auxiliar pessoas com deficiência visual nas mais diversas tarefas, as técnicas de visão computacional também apresentam um papel essencial. Tarefas simples de serem realizadas por pessoas sem deficiência visual tornam-se um grande desafio para pessoas com esse tipo de deficiência. Uma dessas tarefas é atravessar uma rua com segurança, em local seguro, como por exemplo, sobre a FTP (faixa de travessia de pedestres), de forma autônoma e sem a ajuda de outra pessoa (SOUZA, 2007). Segundo o IBGE (Instituto Brasileiro de Geografia e Estatística) 35,8 milhões de brasileiros apresentam deficiência visual (CENSO 2010, 2012), o que representa, segundo (OLIVEIRA, 2012), o equivalente a 18,6% da população.

É neste cenário que o PDI, em especial a visão computacional, aparece como parte integrante deste trabalho de pesquisa e como uma solução para possibilitar a integração de

sistemas e máquinas autônomas em tarefas do nosso dia a dia. Para as indústrias, tais tecnologias podem representar otimização operacional e o desenvolvimento de novos produtos. Por outro lado, tornam-se uma solução para auxiliar no desenvolvimento de sistemas e equipamentos para aplicações no campo da computação assistiva, em especial, para pessoas com deficiência visual.

1.4 Objetivo Geral

Neste trabalho são estudados e apresentados os conceitos e as técnicas necessárias para elaborar um sistema de reconhecimento de padrões e identificação de objetos do mundo real em visão computacional. A biblioteca de processamento de imagens OpenCV disponibiliza várias funções otimizadas para esse tipo de sistema. Na literatura não foi encontrada uma abordagem que utiliza estas funções em um *hardware* específico, como um robô. A partir dessa elaboração, apresenta-se o seguinte objetivo geral:

Utilizar o processamento de imagens e técnicas de visão computacional para extrair características do ambiente que permitam identificar sinais de trânsito e avaliar o desempenho do sistema para verificar se é possível empregar funções OpenCV no robô Humanoide DARwIn-OP em processamentos que exigem um tempo de resposta rápido.

1.5 Objetivos Específicos

Os objetivos específicos deste trabalho de pesquisa foram definidos e divididos conforme os tópicos abaixo:

- Estudar e descrever as principais abordagens empregadas no reconhecimento de padrões e objetos em imagens digitais;
- Testar os métodos propostos em imagens contendo diferentes tipos de padrões, texturas, fundos, condições de iluminação, e outros fatores que interferem na imagem;
- Embarcar o sistema no robô humanoide DARwIn-OP;
- Avaliar a eficiência, a corretude e o desempenho do sistema com a utilização de funções da biblioteca OpenCV no robô humanoide DARwIn-OP em processamentos que exigem um tempo de resposta rápido.

1.6 Organização do documento

No capítulo 2 são explicados os principais conceitos teóricos necessários para um melhor entendimento deste trabalho. O capítulo 3 traz a metodologia adotada. Na sequência, no

capítulo 4, é explicado o sistema desenvolvido. No quinto capítulo são mostrados os resultados obtidos nos testes realizados e no sexto e último capítulo são apresentadas as conclusões e elencadas algumas sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentado o referencial teórico com os conceitos necessários para a realização deste trabalho de pesquisa, visto que são fundamentais para a compreensão e desenvolvimento do mesmo. Além disso, são apresentadas as tecnologias utilizadas e alguns trabalhos semelhantes a este encontrados na literatura.

2.1 Processamento digital de imagens

O PDI implica no processamento de imagens em um sistema de hardware digital, como um microcomputador, um FPGA (*field programable gate array*), DSP (*Digital Signal Processor)*, dispositivos móveis (*smartphones*, *tablets* ou até mesmo aparelhos celulares), entre outros. O objetivo de realizar esse tipo de processamento é extrair informações e/ou características a partir de imagens (GONZALEZ, 2000; GONZALEZ, 2010). A Figura 1 ilustra as etapas básicas do modelo de PDI descrito por (GONZALEZ, 2002; GONZALEZ, 2010).

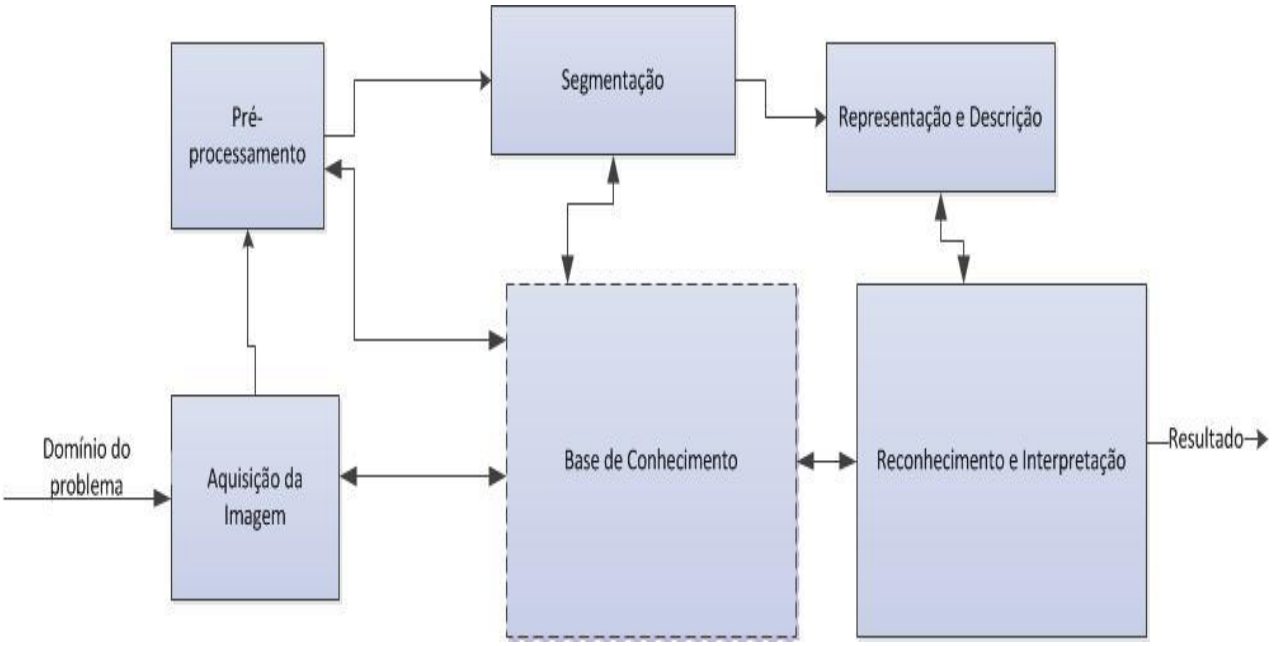


Figura 1:Sistema básico de PDI

Adaptado de (GONZALEZ, 2002; GONZALEZ, 2010)

2.1.1 Aquisição e formação de imagens

O processo de formação de uma imagem digital se inicia com a captura, através do uso de um sensor de imagem, que transforma a energia luminosa em um sinal de tensão. Os

sensores de imagem mais comuns são as câmeras digitais, que, em sua maioria, têm o formato de uma matriz bidimensional, onde cada ponto é um sensor que gera um sinal de tensão proporcional à energia luminosa que incide sobre ele (GONZALEZ, 2000; GONZALEZ, 2002).

Para possibilitar a manipulação de imagens e extrair características das mesmas, é necessário digitalizá-las. Esse processo, denominado digitalização, consiste na transformação das grandezas contínuas em discretas, convertendo as imagens adquiridas em matrizes de números inteiros (GONZALEZ, 2000; GONZALEZ, 2002).

Os elementos desta matriz representam cada pixel (*picture element*), que é o elemento fundamental de uma imagem digital, da imagem. Desta forma, pode-se dizer que uma imagem digital é a representação de um cenário através de uma matriz de pixels dispostos lado a lado. A partir desta representação da imagem, em forma de matriz de pixels, torna-se possível aplicar técnicas de PDI sobre a mesma, através da implementação e utilização de algoritmos computacionais (GONZALEZ, 2000; GONZALEZ, 2002).

As cores em tons de cinza podem ser representados de forma linear, utilizando um único valor. Já para imagens coloridas, deve ser utilizado o modelo tridimensional (GONZALEZ, 2000; GONZALEZ, 2002). Na literatura podem ser encontrados vários modelos de representação de cores, sendo alguns deles descritos resumidamente nas seções seguintes.

2.1.2 Modelos de cores

O processo de percepção e interpretação das cores pelo cérebro humano é um fenômeno fisiopsicológico que ainda não é completamente compreendido. Em 1666, Sir Isaac Newton descobriu através das suas pesquisas que quando um feixe de luz solar atravessa um prisma de vidro, o feixe de luz emergente não é branco, mas consiste em um espectro contínuo de cores que variam de violeta, em uma extremidade, a vermelho, na outra (GONZALEZ, 2000; GONZALEZ, 2002; GONZALEZ 2010).

É possível dividir o espectro de cores em seis grandes regiões: violeta, azul, verde, amarelo, laranja e vermelho, conforme ilustrado na Figura 2, a seguir. Na percepção das cores reais em uma imagem, podemos observar que nenhuma cor do espectro termina de forma abrupta, sendo que cada cor se funde suavemente à outra, conforme pode ser observado na Figura 3, a seguir (GONZALEZ, 2010).

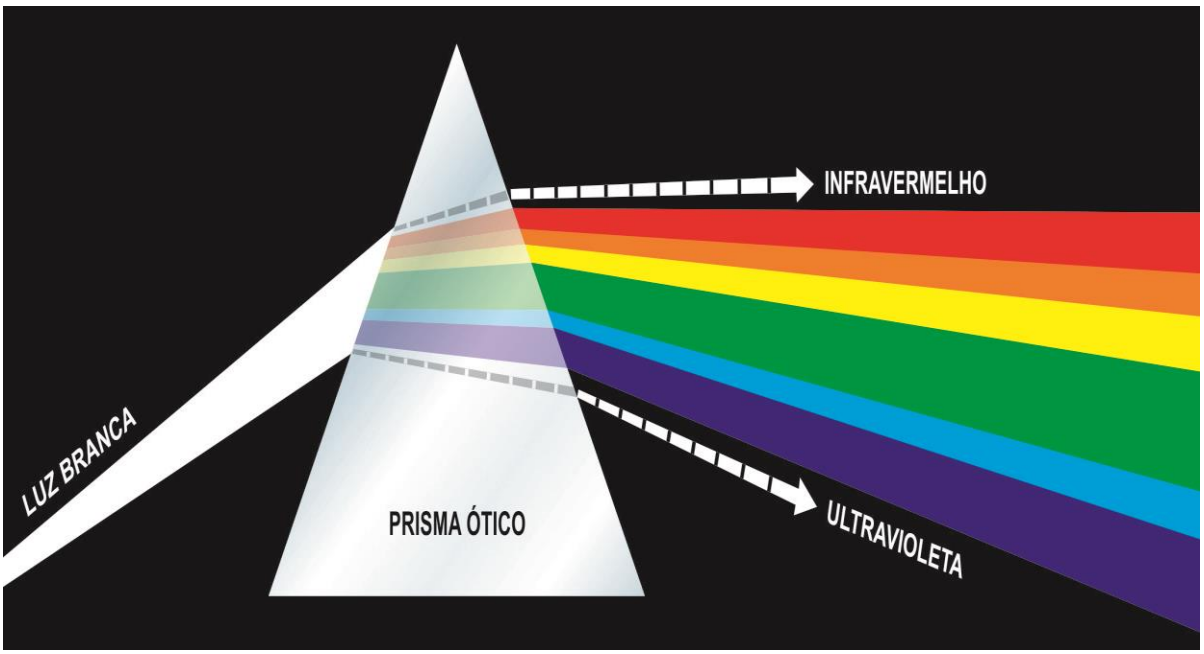


Figura 2: Espectro de cores visto pela passagem da luz branca através de um prisma
Fonte: Adaptado de (GONZALEZ, 2010)

As cores percebidas pelo olho humano em determinado objeto são definidas pela natureza da luz refletida pelo objeto. A luz visível é formada por uma banda de frequências relativamente estreita no espectro de energia eletromagnética, conforme ilustrado na Figura 3, a seguir. Com isso, um corpo que reflete a luz de forma balanceada em todos os comprimentos de onda visíveis é percebido pelo observador como sendo branco. Entretanto, um corpo que favorece a refletância de uma faixa limitada do espectro visível exibe alguns tons de cores. Como exemplo, podemos citar objetos verdes refletem a luz com comprimentos de onda primariamente no intervalo de 500 a 570 nm, enquanto absorvem a maior parte da emergia de outros comprimentos de onda (GONZALEZ, 2010).

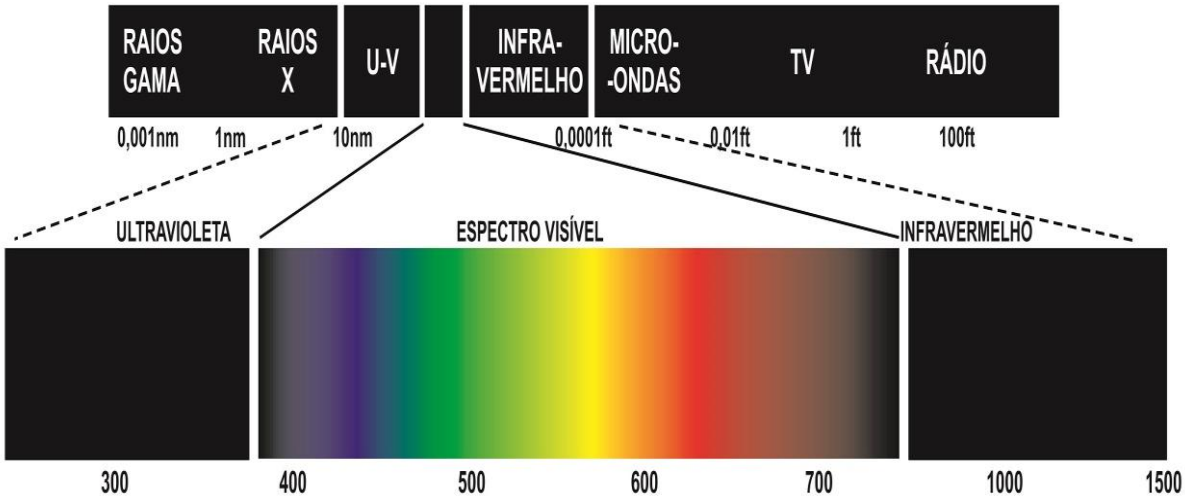


Figura 3: Comprimento de onda englobando a faixa visível do espectro eletromagnético

Fonte: Adaptado de (GONZALEZ, 2010)

Para sistemas onde a ciência das cores é fundamental, como sistemas de visão computacional, a caracterização da luz é fundamental. Se esta for acromática (sem cores), seu único atributo será a intensidade, ou quantidade, que é uma medida escalar que varia do preto, passando pelos diferentes níveis de cinza, até o branco. A luz cromática engloba o espectro de energia eletromagnética de aproximadamente 400 a 700 nm (GONZALEZ, 2010).

Em técnicas de desenvolvimento de sistemas para visão computacional é comum o uso da identificação de cores juntamente com outras características para mapear áreas específicas que são de interesse na imagem. Os modelos de cores descrevem matematicamente como uma determinada cor deve ser representada (KLAWONN, 2005; KOSCHAN, 2008; GONZALEZ, 2002).

Na literatura, são encontrados diversos modelos de cores para representação de imagens digitais, sendo que estes são adotados de acordo com a aplicação. Um dos mais conhecidos e tradicionais, também considerado o modelo padrão para representar cores em imagens digitais, é o RGB (*Red*, *Green and Blue*). Este modelo é usado pela maioria dos dispositivos de aquisição e visualização de imagens, podendo ser visto como um cubo de cores onde os componentes vermelho, verde e azul correspondem aos eixos *x*, *y* e *z*, respectivamente. Os tons de cinza ocorrem quando os três componentes possuem valores iguais, seguindo a diagonal principal do cubo, com a origem (0,0,0), preta, e a extremidade inversa (255,255,255), branca (KLAWONN, 2005; KOSCHAN, 2008; GONZALEZ, 2002). O cubo que representa o modelo RGB é mostrado na Figura 4, a seguir.

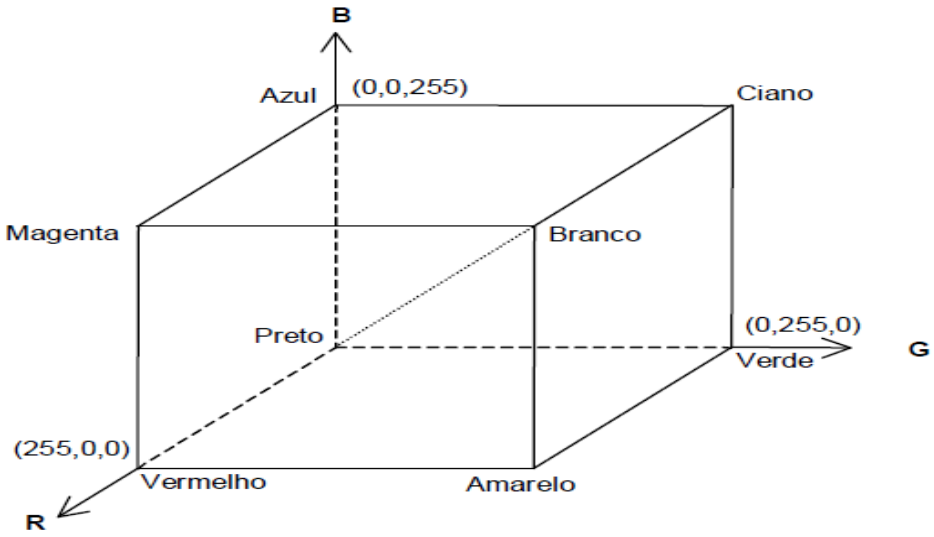


Figura 4:O Cubo RGB

Adaptado de (GONZALEZ, 2000; GONZALEZ, 2010)

Outro modelo bastante utilizado é o HSI (*Hue, Saturation and Intensity*). Pela percepção humana as cores são representadas por coordenadas polares, nas quais o ângulo do vetor corresponde à matiz e o módulo do vetor corresponde à saturação. Enquanto a matiz descreve uma cor pura, a saturação representa sua pureza. Os tons de cinza podem ser percebidos quando a saturação é equivalente a zero. O modelo HSI, ou HSV (*Hue, Saturation and Value*), é uma aproximação da percepção humana de cores, onde H corresponde à matiz e S à saturação. O componente I, ou V, no modelo HSV *(Hue, Saturation and Value),* representa a intensidade da cor.

O valor *H* é o atributo que descreve onde a cor fica ao longo do espectro, ou seja, qual a cor efetivamente representada, como por exemplo, azul ou vermelho. O valor *V* representa o brilho ou intensidade da cor, que pode variar de 0 (cor totalmente preta) e 100. A saturação S é a medida de quão diferente uma cor aparece a partir de uma escala de cinza de mesma luminosidade, o quão pura essa cor é. O valor de S pode variar de 0 a 100, onde um valor baixo indica uma cor neutra e sem brilho e um valor mais elevado indica uma cor forte e pura (KLAWONN, 2005; KOSCHAN, 2008; GONZALEZ, 2002; GONZALEZ, 2010).

O modelo *HSV* pode ser representado através de um hexágono, que é ilustrado na Figura 5 e na Figura 6, a seguir. Neste, pode ser visto, por exemplo, que a cor vermelha é determinada por H = 0 ou 360 e a cor verde por H = 120.

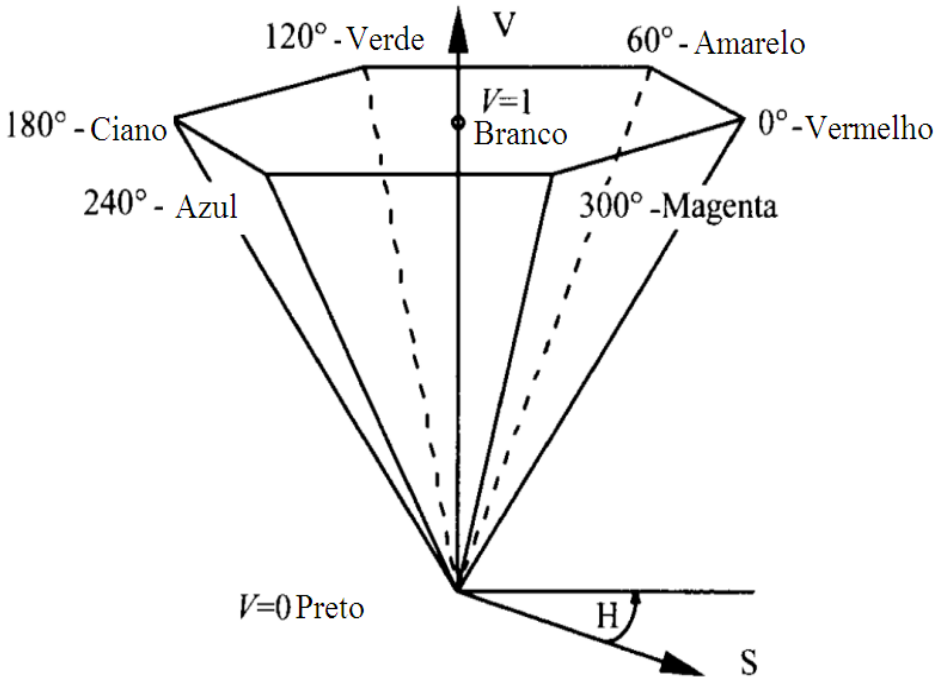


Figura 5: Hexágono que representa o modelo HSV
Adaptado de (GONZALEZ, 2000; GONZALEZ, 2010)

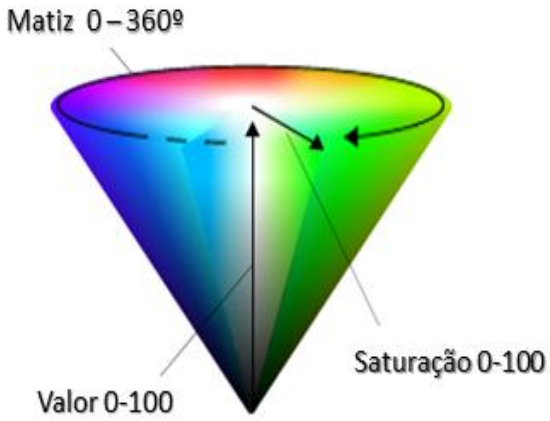


Figura 6: Representação do modelo HSV
Adaptado de (GONZALEZ, 2000; GONZALEZ, 2010)

Os modelos de representação de cores descritos anteriormente utilizam-se de transformações não-lineares para obtenção de seus parâmetros. Para sistemas que possuem pouco poder computacional, pode ser útil a adoção de modelos com transformações lineares, como o YUV (*Luminance (Y), blue–luminance (U), red–luminance (V)*). O modelo YUV é utilizado para codificação de cor nos sistemas de televisão europeu. O valor de Y pode variar entre 0 a 255 e representa a luminância, enquanto os valores *U* e *V* representam a informação

sobre a cor. Este modelo de cor é mais sensível à luminância (mudanças de luz) do que à cor, o que é uma propriedade da visão humana. Ao mesmo tempo que permite transmitir imagens coloridas, neste modelo também podem ser enviadas imagens em preto e branco. No modelo YUV é possível fazer transmissão de cores em um tempo menor que o necessário pelo modelo RGB (GONZALES, 2002; HENNING, 2007; KOSCHAN, 2008).

2.1.3 Amostragem e quantização de Imagens

A saída da maioria dos sensores utilizados para capturar imagens consiste em uma forma de onda com tensão contínua, cuja amplitude e comportamento no espaço estão relacionados ao fenômeno físico que está sendo captado por estes sensores. Para formar uma imagem digital, estes dados contínuos capturados devem ser convertidos para o formato digital, o que envolve dois processos básicos: amostragem e quantização (GONZALEZ, 2010).

Na Figura 7, a seguir, são ilustrados esses processos. No eixo das ordenadas representa os possíveis valores que podem ser atribuídos a uma amostra (entre 0 e 255), enquanto o eixo das abcissas representa as amostragens feitas do sinal original.O sinal original, contínuo, é ilustrado na cor verde. Em vermelho, podem ser observados alguns pontos, que são as amostras (amostragem). Os valores atribuídos a essas amostras são ilustrados em azul, que é o processo de quantização das mesmas.

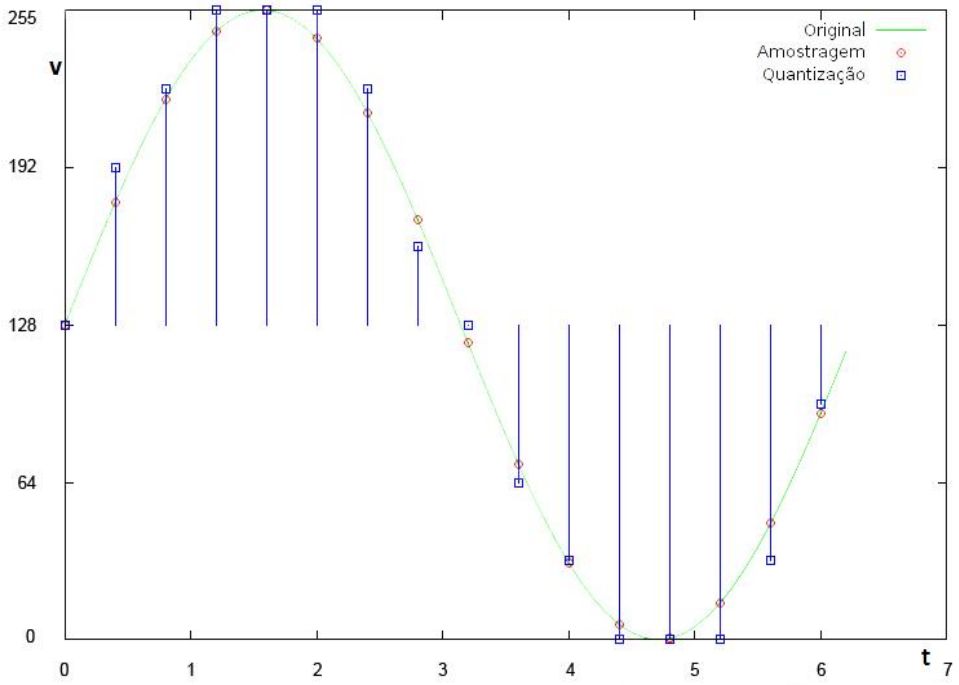


Figura 7: Amostragem e quantização de um sinal contínuo

Fonte: Adaptado de (GONZALEZ, 2010)

A Figura 8, a seguir, ilustra a ideia básica da amostragem e da quantização. A Figura 8 (a) ilustra uma imagem contínua *f* que será convertida para o formato digital. As imagens podem ser contínuas em relação às coordenadas *x* e *y* e também em relação à amplitude. Para converter uma imagem para o formato digital, é necessário fazer a amostragem da função em ambas as coordenadas e também na amplitude (GONZALEZ, 2010).

Na Figura 8 (b) é mostrado um gráfico que representa os valores de amplitude, ou seja, nível de intensidade da imagem contínua ao longo do segmento de reta AB, ilustrado na imagem contínua da Figura 8 (a). Nesta mesma figura pode ser observado algumas variações aleatórias, que se devem ao ruído existente na imagem. Para realizar o processo de amostragem desta função, é necessário coletar amostras igualmente espaçadas ao longo da linha AB, conforme ilustrado na Figura 8 (c) (GONZALEZ, 2010).

Uma pequena marca vertical na parte inferior da figura indica a posição das amostras no espaço. O conjunto destas localizações discretas provê a função de amostragem. Entretanto, os valores das amostras ainda cobrem, verticalmente, uma faixa contínua de valores de intensidade. Estes valores de intensidade devem ser convertidos (quantizados) em quantidades discretas, afim de formar uma função digital. No lado direito da Figura 8 (c) é indicado a escala de intensidade dividida em oito intervalos discretos, que variam do preto ao branco. As marcas verticais indicam o valor específico atribuído a cada um destes oito níveis de intensidade. Os níveis de intensidade contínuos são quantizados, sendo que para cada amostra é atribuído um dos oito valores (GONZALEZ, 2010).

Na Figura 8 (d) são mostradas as amostras resultantes da amostragem e da quantização. Esse procedimento realizado linha por linha, iniciando na parte superior da imagem, produz uma imagem digital bidimensional. Na Figura 8 é mostrado, de forma implícita, que a precisão obtida na quantização está diretamente relacionada ao conteúdo de ruído do sinal da amostragem (GONZALEZ, 2010).

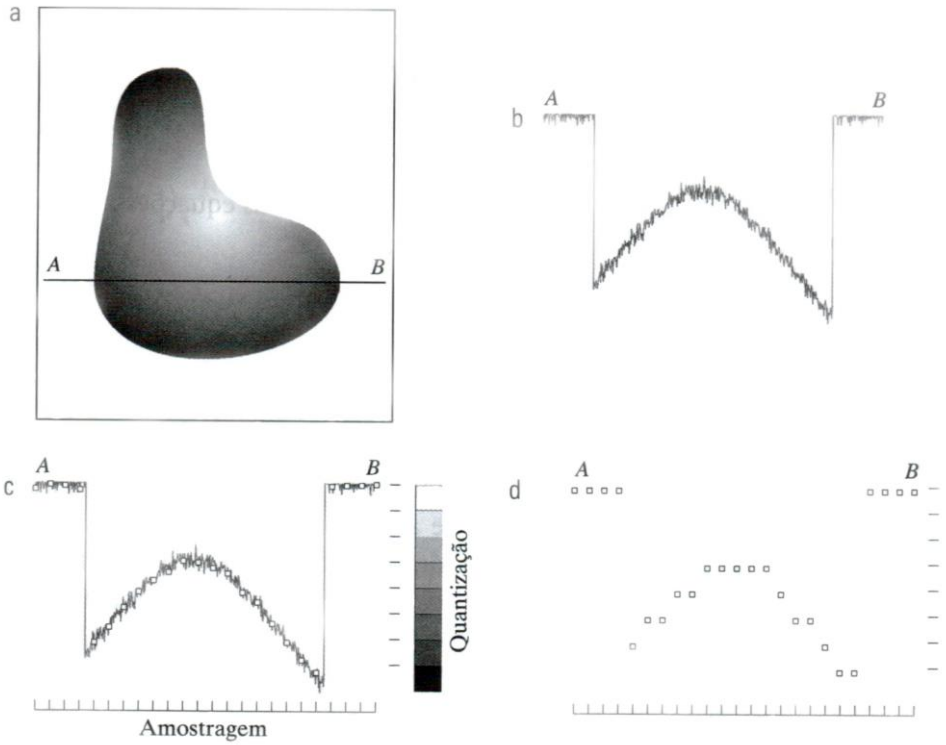


Figura 8: Amostragem e Quantização

Fonte: Adaptado de (GONZALEZ, 2010)

A amostragem consiste na digitalização dos valores de coordenada, ao passo que a quantização consiste na digitalização dos valores de amplitude (GONZALEZ, 2010). A Figura 9, a seguir, mostra o resultado do processo de amostragem e quantização (IMAGEM (B) sobre uma imagem contínua projetada em uma matriz de sensores (IMAGEM (A)).

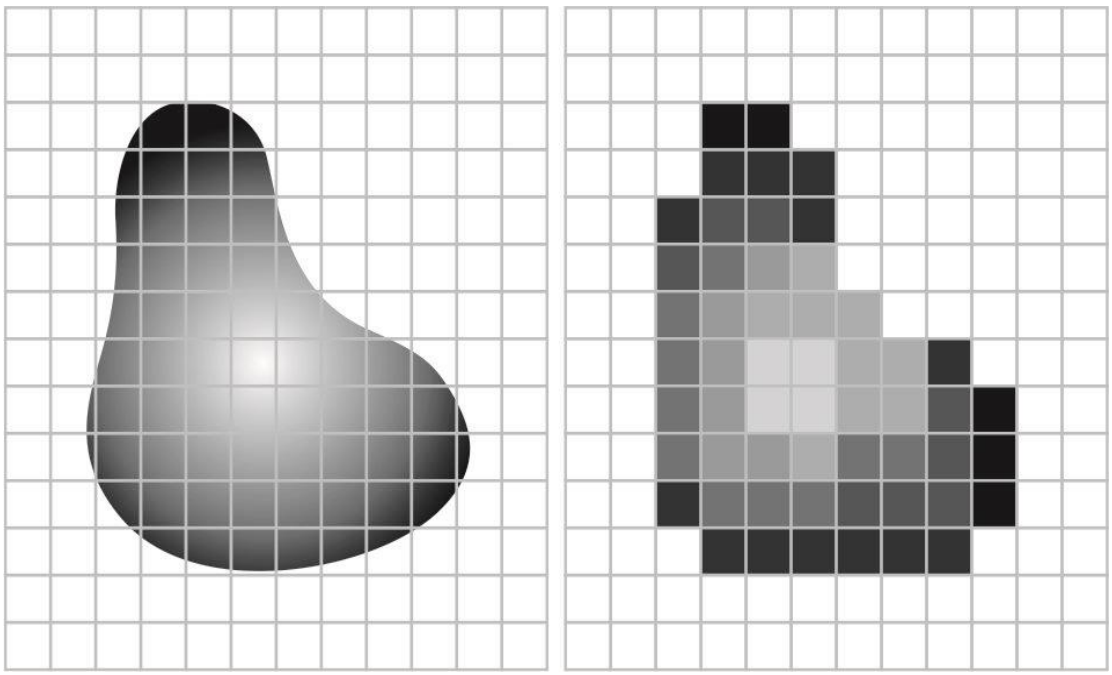


Figura 9: Resultado do processo de amostragem

Fonte: Adaptado de (GONZALEZ, 2010)

2.1.4 Relacionamentos entre pixels

Em uma imagem digital, cada pixel pode conter informações diferentes. Com isso, podem ser estabelecidas várias relações entre estes, a fim de encontrar características nas imagens (GONZALEZ, 2010).

- Vizinhos de um pixel: Um determinado pixel *p*, localizado nas coordenadas (x,y) possui quatro vizinhos horizontais e verticais, cujas coordenadas podem ser expressas por: (x + 1, y), (x – 1, y), (x, y + 1), (x, y – 1).
- Adjacência, conectividade, regiões e fronteiras: O conjunto *V* define os valores de intensidade utilizados para definir a adjacência. Em uma imagem binária, para referir-se à adjacência igual a 1, $V = \{1\}$. Já em uma imagem de escala de cinza, o conjunto *V* terá os valores de 0 à 255, que representam as diversas tonalidades de cinza da cor.
- Medidas de distâncias: Considerando os pixels *p*, *q* e *z*, com coordenadas (x,y), (s,t) e (v,w), respectivamente, podemos definir que D é uma função distância ou medida de distância se:
  a) $D(p,q) \geq 0$ ($D(p,q) = 0$ se $p=q$),
  b) $D(p,q) = D(q,p)$ e
  c) $D(p,z) \leq D(p,q) + D(q,z)$.

Para determinar a distância euclidiana entre os pixels p e q, usa-se a seguinte equação (GONZALEZ, 2010):

$$D_e(p,q) = \sqrt{[(x-s)^2 + (y-t)^2]} \quad (1)$$

2.1.5 Pré-processamento

Em PDI são utilizadas várias técnicas de pré-processamento nas imagens. Essas técnicas são empregadas com o intuito de melhorar a qualidade das imagens, realçando as características consideradas relevantes nas mesmas.

As técnicas de pré-processamento de imagens são fundamentalmente baseadas em métodos que trabalham no domínio espacial e em métodos que trabalham no domínio da frequência. Os métodos que utilizam o domínio espacial trabalham com o processamento das coordenadas de posição dos pixels, utilizando filtros para manipular o plano da imagem. Por outro lado, os métodos baseados em filtros utilizam o espectro da imagem gerado com a aplicação da transformada de Fourier na mesma (KOSCHAN, 2008; GONZALEZ, 2000; GONZALEZ, 2002).

Um processo bastante utilizado é a quantização. Em processamento de imagens, o processo de quantização consiste em uma técnica de redução do número de cores da imagem, formando uma nova imagem que pareça visualmente igual ou muito similar a original. Dessa forma, a nova imagem representa fielmente a original, reduzindo significativamente o tamanho da imagem e a quantidade de dados a serem armazenados e processados (KOSCHAN, 2008; GONZALEZ, 2000; GONZALEZ, 2002).

Esse processo se torna uma opção bastante atraente para sistemas que são executados em equipamentos com poder de memória e processamento limitados (ESS, 2009; GALLO, 2008), que é o caso do robô humanoide utilizado nesse trabalho. Por essa razão, essa técnica será empregada nesse trabalho, otimizando os processamentos realizados e permitindo uma resposta mais rápida do sistema.

2.1.6 Segmentação

A técnica de segmentar uma imagem digital consiste em dividi-la em regiões com características ou atributos similares. Entre as etapas do PDI a segmentação pode ser considerada a mais crítica, pois é nela que são definidas as regiões de interesse da imagem que serão utilizadas nos processamentos e análises posteriores das informações. Quaisquer

erros ou distorções nesta etapa podem influenciar nas etapas seguintes, podendo produzir resultados não desejados e incorretos ao final do processo. Desta forma, o sucesso no processo de reconhecimento de objetos em uma imagem digital depende de uma segmentação efetiva (PRATTE, 2007; GONZALEZ, 2002; SOLOMON, 2011).

O principal objetivo da segmentação é dividir a imagem em regiões que possam ser rotuladas. Os objetos segmentados são denominados de primeiro plano e o restante da imagem de plano de fundo. A segmentação correta de uma imagem depende principalmente dos tipos de objetos ou regiões a identificar. A abordagem central da segmentação consiste em definir qual a relação entre um determinado pixel e os seus vizinhos ou outros pixels da imagem, de modo que este possa ser atribuído a uma região ou outra. As principais abordagens para desenvolver métodos de segmentação consistem na verificação de similaridade entre os pixels e na verificação de descontinuidades entre eles (GONZALEZ, 2000; GONZALEZ, 2002; GONZALEZ, 2010).

Uma das técnicas mais conhecidas baseadas na similaridade é a binarização de imagens, também conhecida como *image trescholding*. Essa técnica é amplamente utilizada em sistemas de visão computacional por ser simples de implementar e eficiente. É utilizada quando a amplitude dos tons de cinza é suficiente para caracterizar os objetos que podem ser encontrados em uma imagem digital. Nessa técnica, são utilizados os níveis de cinza dos pixels que compõem os objetos representados na imagem e o fundo como limiares de separação entre os mesmos. Com a aplicação da binarização é possível obter uma imagem com apenas dois níveis de luminância (branco e preto). Desta forma, a imagem é transformada em uma imagem chamada binária. Nos métodos de binarização existentes, o grande desafio consiste em definir o valor ideal deste limiar. (GONZALEZ, 2000; GONZALEZ, 2002; GONZALEZ, 2010).

As técnicas que utilizam as verificações de descontinuidades entre os pixels, buscam por variações abruptas no nível de luminância entre pixels vizinhos. Desta forma, procuram determinar pelos pixels que delimitam os contornos ou bordas dos objetos presentes na imagem (GONZALEZ, 2000; GONZALEZ, 2002; GONZALEZ, 2010).

2.1.7 Extração de atributos

No processo de extração de atributos, são extraídas as informações úteis da imagem. Na literatura, estes são classificados em atributos da imagem como um todo, como número de

objetos ou a sua área total, por exemplo, e em atributos de região, como a área, o perímetro ou a forma dos objetos independentes, por exemplo (GONZALEZ, 2000; GONZALEZ, 2002).

Com a separação da imagem em regiões correspondentes ao fundo e aos objetos da mesma, na etapa de segmentação, é possível obter agrupamentos de pixels com características semelhantes, denominados BLOBs (*Binary Large Object*). A partir desse agrupamento, na fase de extração de atributos, pode ser realizado o processo de rotulação desses BLOBs, permitindo a identificação dos mesmos. Com isso, os processos seguintes do sistema de PDI podem concentrar-se em cada uma das regiões rotuladas para realizarem a classificação e reconhecimento dos objetos (GONZALEZ, 2000; GONZALEZ, 2002).

A partir da identificação dos BLOBs, é possível obter vários atributos do mesmo, como por exemplo, sua área, seu perímetro, a sua orientação, o retângulo, quadrado ou elipse mínima que envolve o objeto. A partir desses atributos, podem ser aplicados vários processamentos pertinentes às etapas seguinte do PDI (GONZALEZ, 2000; GONZALEZ, 2002).

2.1.8 Classificação e reconhecimento

Nesta etapa do PDI deve ocorrer a identificação dos objetos segmentados na imagem, de forma sistêmica e automática. Em uma imagem digital, objetos podem ser descritos a partir de suas características, tais como tamanho, textura, forma e cor. Na literatura, os autores destacam duas abordagens básicas para classificar os descritores de objetos nas imagens digitais: descritores de contorno e descritores de região (GONZALEZ, 2000; GONZALEZ, 2002).

Na literatura são descritos vários métodos de reconhecimento de padrões, como classificadores bayesianos, que utilizam as propriedades estatísticas dos objetos. Além desses, podem ser utilizadas técnicas que aplicam redes neurais artificiais, através de métodos que utilizam a distância entre os objetos na imagem e suas formas padrões (GONZALEZ, 2000; GONZALEZ, 2002; GONZALEZ, 2010).

Nas técnicas de reconhecimento de padrões podem ser aplicados algoritmos de aprendizado de máquina, de forma a gerar classificadores capazes de prever a classe de novos eventos com as mesmas características. Podem ser utilizados métodos de aprendizado supervisionado, onde a determinação do modelo ou da classe é obtido a partir de exemplos conhecidos (treinamento). Podem ser aplicados também métodos de aprendizado não supervisionado, onde o sistema visa “aprender” a obter agrupamentos seguindo algum critério

de similaridade, buscando encontrar tendências e/ou padrões (GONZALEZ, 2000; GONZALEZ, 2002; GONZALEZ, 2010).

2.1.9 Operadores morfológicos

Para utilizar processamentos morfológicos em imagens, é necessário binarizar as mesmas. Com o uso de operadores morfológicos, é possível alterar a forma das imagens (GONZALEZ, 2010).

Os principais operadores morfológicos utilizados em PDI são a erosão e a dilatação. A erosão consiste basicamente em encolher os objetos da imagem. Esse processo pode ser considerado uma transformação morfológica que combina dois conjuntos A e B usando vetores de subtração. A erosão de "A" por "B", indicada por $A \ominus B$ é definida por:

$$A \ominus B = \{Z \mid (B)_z \subseteq A\} \quad (2)$$

Desta forma, a equação indica que a erosão de A por B é o conjunto de todos os pontos Z de forma que B, transladado por Z, está contido em A. Com o uso desta operação, objetos pequenos tendem a ser eliminados. Este processo é bastante utilizado em processamentos de imagens a fim de identificar dígitos ou caracteres, como por exemplo, placas de automóveis (GONZALEZ, 2010).

O processo de dilatação é o oposto da erosão, ou seja, combina dois conjuntos A e B usando a adição vetorial, ampliando os objetos da imagem. A dilatação de A por B é definida como:

$$A \oplus B = \{Z \mid (B')_z \cap A\} \quad (3)$$

O símbolo $\oplus$ indica a operação de dilatação. A equação é baseada na reflexão de B em torno da sua origem, seguida da translação dessa reflexão por Z. Desta forma, a dilatação de A por B é o conjunto de todos os deslocamentos, Z, de forma que B' e A se sobreponham no mínimo por um elemento. Com esta operação, objetos pequenos tendem a ser eliminados, sendo unidos, fazendo que a área dos objetos seja aumentada. Esse processo pode ser utilizado para restaurar imagens, como por exemplo, linhas de uma faixa de pedestres apagadas parcialmente (GONZALEZ,2010).

As operações anteriores podem ser usadas de forma combinada. As combinações mais tradicionais são open (abertura) e close (fechamento). A operação open consiste da erosão seguida da dilatação, enquanto a close consiste na dilatação seguida da erosão. Em ambas as operações, podem haver "N" ciclos de erosão e "N" ciclos de dilatação. Através da open, é possível separar objetos ligados por poucos pixels e objetos muito pequenos são eliminados da imagem. Na operação close, pequenos "buracos" ou separações entre objetos são eliminados (GONZALEZ, 2010).

Os operadores morfológicos são aplicados em várias técnicas de PDI. Dentre as principais aplicações, podem ser citadas a remoção de pequenas regiões nas imagens, remoção de buracos em regiões, suavização da forma dos contornos e esqueletização (GONZALEZ, 2010).

2.1.10 Projeções em perspectiva

O nome perspectiva tem suas origens no Latim, onde a palavra *perspicere* significa ver através de. É uma forma de representar imagens tridimensionais (altura, largura e comprimento) em uma superfície plana (NOGUEIRA, 2009; HARLEY, 2000; XU, 1996; COXETER,1993; KANNALA, 2010; FAUGERAS, 1993).

Os elementos base da perspectiva são: LH (linha do horizonte), PV (ponto de vista), PF (ponto de fuga) e LF (linhas de fuga). A LH indica o ponto de vista do observador, traçada horizontalmente na imagem. Em representações gráficas da perspectiva, o PV pode ser identificado por uma linha perpendicular a LH, localizando-se exatamente no cruzamento destas duas linhas. Esta linha pode estar localizada no centro da imagem ou em um de seus lados, dependendo do ângulo visual de observação (NOGUEIRA, 2009; HARLEY, 2000; XU, 1996; COXETER,1993; SOBREARTE, 2014; KANNALA, 2010; FAUGERAS, 1993). A Figura 10, a seguir, ilustra a localização da LH e do PV em uma imagem.

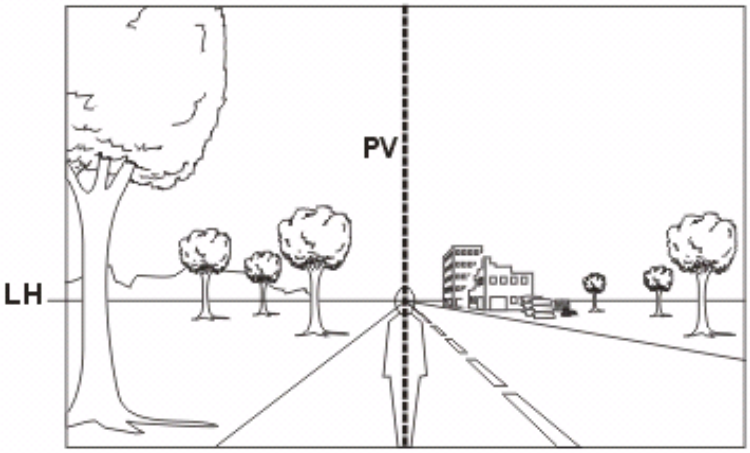


Figura 10: Ponto de vista
Fonte: Adaptado de (SOBREARTE,2014)

O PF é o ponto para onde todas as linhas paralelas convergem, vistas em perspectiva. É localizado na LH. Em perspectiva, podem haver múltiplos PF. As LF são linhas imaginárias que determinam o efeito da perspectiva convergindo para o PF, gerando a sensação visual de profundidade das faces dos objetos em perspectiva (NOGUEIRA, 2009; HARLEY, 2000; XU, 1996; COXETER,1993; KANNALA, 2010; FAUGERAS, 1993).

Ao observar uma estrada de ferro, posicionando-se de frente para ela olhando-a em direção longitudinal, é possível observar que a sua imagem se transforma ao longo da distância, até um ponto que parece estar no infinito. Embora sejam paralelos, os trilhos parecem juntar-se em determinado local. Este local é denominado de horizonte. O que ocorre, de fato, é que as formas são observadas a partir do nosso ponto de vista e segundo as regras que criamos (NOGUEIRA, 2009; HARLEY, 2000; XU, 1996; COXETER,1993; KANNALA, 2010; FAUGERAS, 1993). A Figura 11, a seguir, ilustra essa teoria e mostra os elementos base da perspectiva.

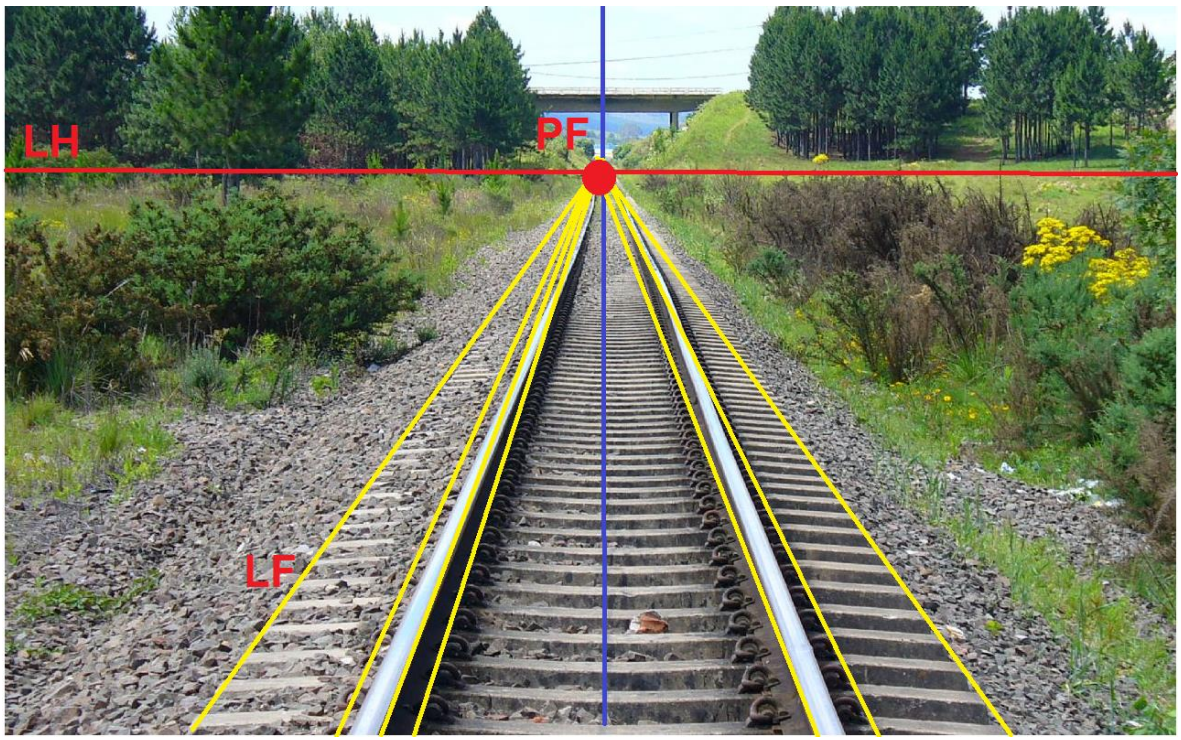


Figura 11: Ilustração dos elementos base da perspectiva

Fonte: O Autor

Nas projeções em perspectiva, retas que não são paralelas ao plano de projeção convergem para um PF. O centro de projeção é um ponto próprio, em coordenadas finitas no sistema tridimensional. Esta projeção deforma a figura, diminuindo os objetos mais distantes e distorcendo os ângulos. Na representação em perspectiva, esse efeito é formado pelas linhas inclinadas (diagonais) que convergem no PF (NOGUEIRA, 2009; HARLEY, 2000; XU, 1996; COXETER,1993).

Na Figura 12, a seguir, são ilustrados os planos da perspectiva. Em verde, é mostrado o plano de terra, ou seja, o chão, onde os objetos estão situados. Em azul, é mostrado o plano horizontal e em rosa o plano de projeção. Nesta imagem são considerados dois pontos de fuga, sendo um a direita e o outro a esquerda do observador.

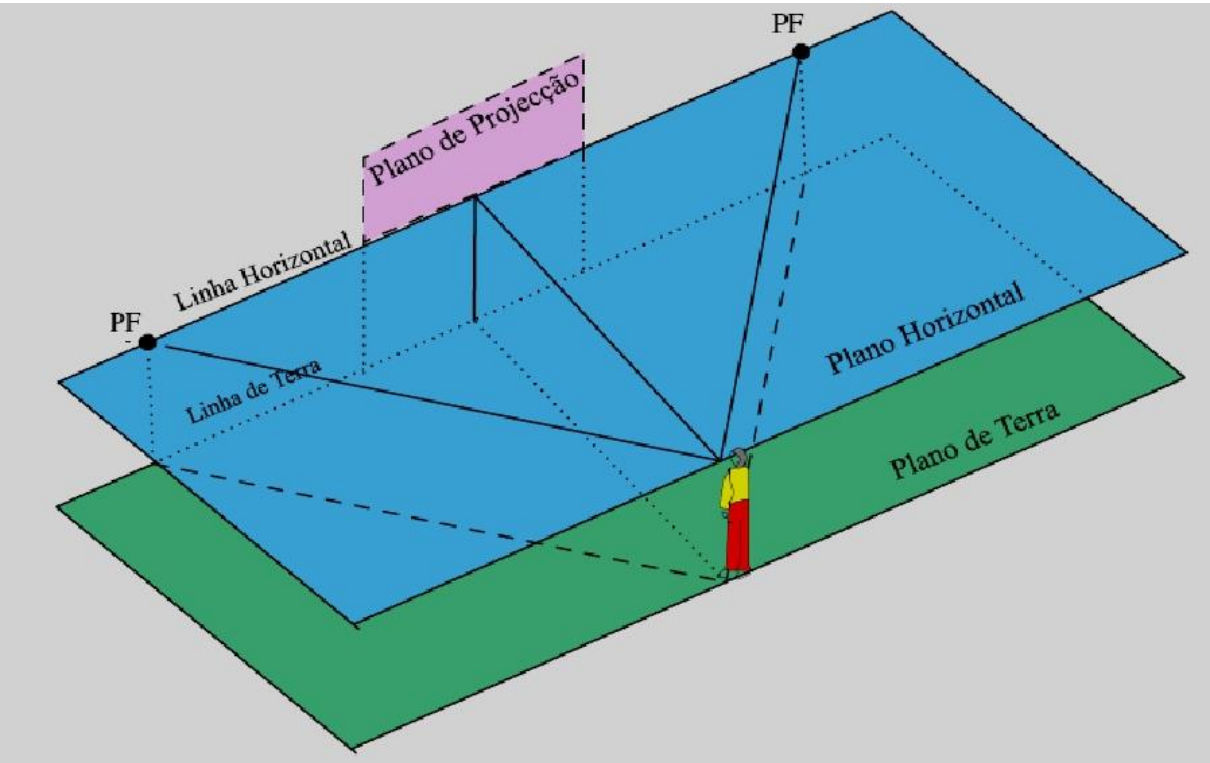


Figura 12: Planos de projeção

Fonte: O Autor

2.2 Visão computacional

A visão computacional é considerada uma subárea do PDI, na qual são desenvolvidos métodos e técnicas para desenvolver sistemas computacionais capazes de interpretar imagens. Os métodos de reconhecimento de padrões são utilizados com o intuito de detectar padrões em imagens digitais com base em exemplos fornecidos, denominados de modelos, máscaras ou *templates*. Além de realizar a detecção, os métodos também podem classificar os padrões em classes ou categorias (GONZALEZ, 2002; CONCI, 2008).

Um sistema de visão computacional deve capacitar uma máquina com habilidades de visão, assim como a visão humana. Desta forma, deve possibilitar a descrição de uma cena contida em uma imagem digital, através do reconhecimento de padrões e objetos nela contidos, independentemente da posição, tamanho, orientação ou variações diversas (GONZALEZ, 2002; CONCI, 2008).

Tipicamente, um sistema de visão computacional possui seis etapas: aquisição da imagem, pré-processamento, segmentação, representação e descrição (extração de características) e reconhecimento. Esse tipo de sistema pode ser classificado, quanto ao grau

de abstração, em baixo, médio e alto, sendo que quanto maior o nível de abstração, maior a redução da quantidade de informações manipuladas (GONZALEZ, 2002; CONCI, 2008).

No nível baixo é gerado um conjunto de dados composto pelos valores dos pixels da imagem original, como a intensidade de brilho, por exemplo. Neste nível estão associadas as etapas de aquisição e pré-processamento. No nível médio ocorre a conversão do conjunto de valores dos pixels em um conjunto de características que descrevem os objetos presentes na imagem. A este nível estão associadas as etapas de segmentação, representação e descrição e reconhecimento. A interpretação da imagem ocorre no nível alto, a partir do conjunto de características obtido (GONZALEZ, 2002; CONCI, 2008).

2.3 Faixa de travessia de pedestres

A FTP é uma sinalização transversal que indica a área das ruas reservada para a travessia de pedestres nas vias, onde estes tem prioridade maior na travessia em relação aos veículos. Estas faixas devem ser instaladas nas vias com demanda de travessia, junto a semáforos, focos de pedestres, no prolongamento das calçadas e passeios, conforme estabelecido no código brasileiro de trânsito – Lei n.º 9.503, de 23 de setembro de 1977, anexo II item 2.2.2 – Marcas transversais, alínea c (ABNT, 2004; CTB, 2008; CONTRAN, 2007).

A resolução nº 160/04 do CONTRAN (Conselho Nacional de Trânsito), através da norma ABNT NBR 9050/2004, define dois padrões de faixas: tipo zebrada (FTP-1) e tipo paralela (FTP-2). Ambos os padrões devem possuir a cor branca e ocupar toda a largura da pista (ABNT, 2004; CTB, 2008; CONTRAN, 2007). O presente trabalho foi desenvolvido com base nas FTP-1, do tipo zebrada, devido a este ser o padrão mais encontrado nas ruas das cidades brasileiras.

O tipo de FTP-1 (zebrada) é mostrado na Figura 13, a seguir. Este padrão pode ser utilizado em vias semaforizadas ou não, onde há um volume significativo de pedestres, nas proximidades de escolas, polos geradores de viagens, em meio de quadra ou onde estudos de engenharia indicarem ser necessário a instalação desse tipo de sinalização. Segundo a norma, a largura das FTP é variável, sendo determinada de acordo com o fluxo de pedestres do local, conforme a equação a seguir:

$$L = \frac{F}{K} > 4 \tag{4}$$

Onde:

L = Largura de faixa, em metros, a qual deve ter mais de 4 metros.

F= Fluxo de pedestres estimado ou medido nas horas de pico.

K = taxa constante de 25 pedestres por minuto.

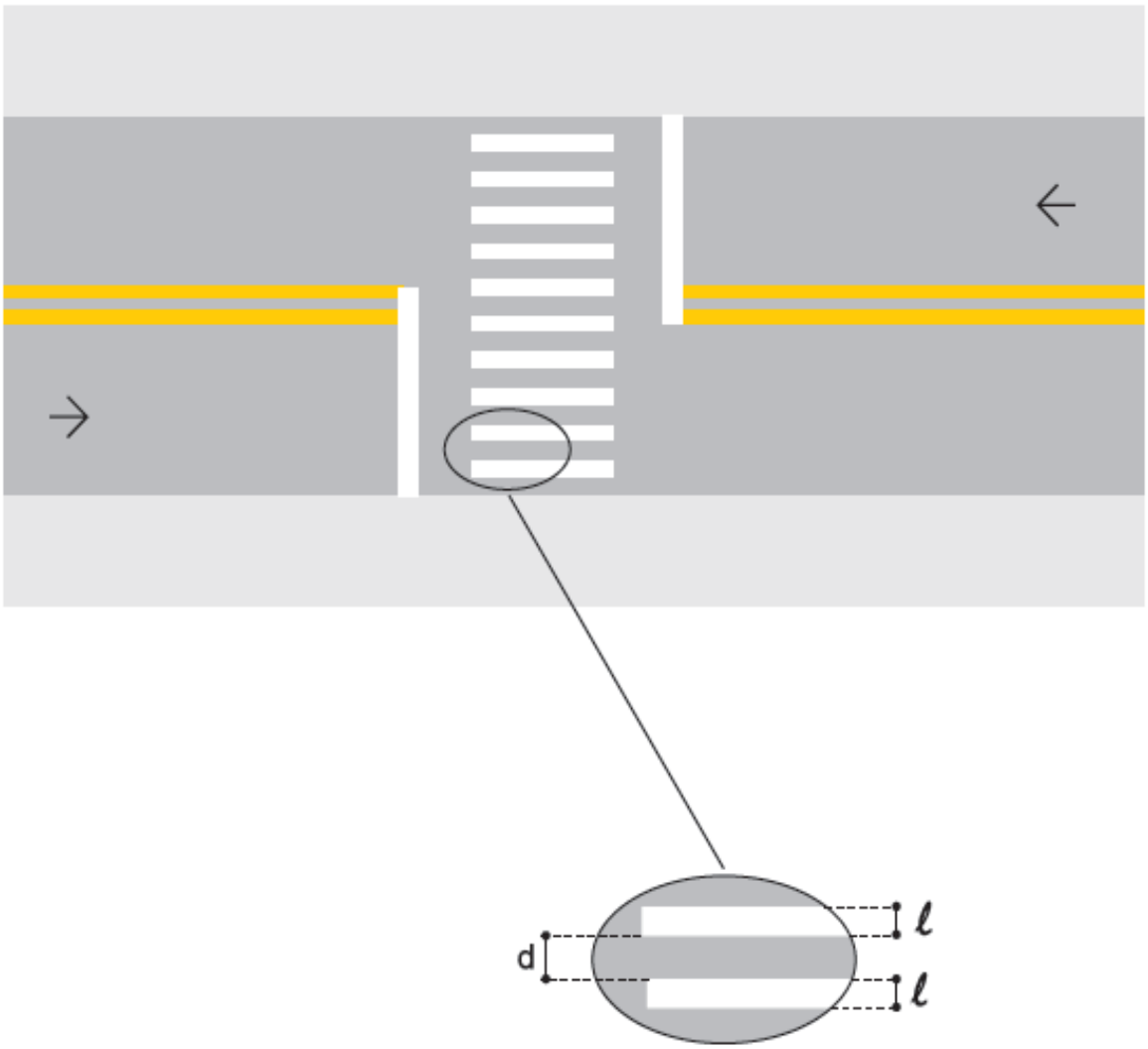


Figura 13: Padrão da FTP-1

Adaptado de (CONTRAN, 2007)

O padrão de FTP-2 (paralela) é mostrado na Figura 14, a seguir. Este tipo de faixa pode ser utilizada apenas em interseções semaforizadas. A largura (l) das linhas pode variar de 0,4 a 0,6 metros. A distância (d) entre as linhas deve ser de 3 metros, no mínimo, sendo recomendada 4 metros (ABNT, 2004; CTB, 2008; CONTRAN, 2007).

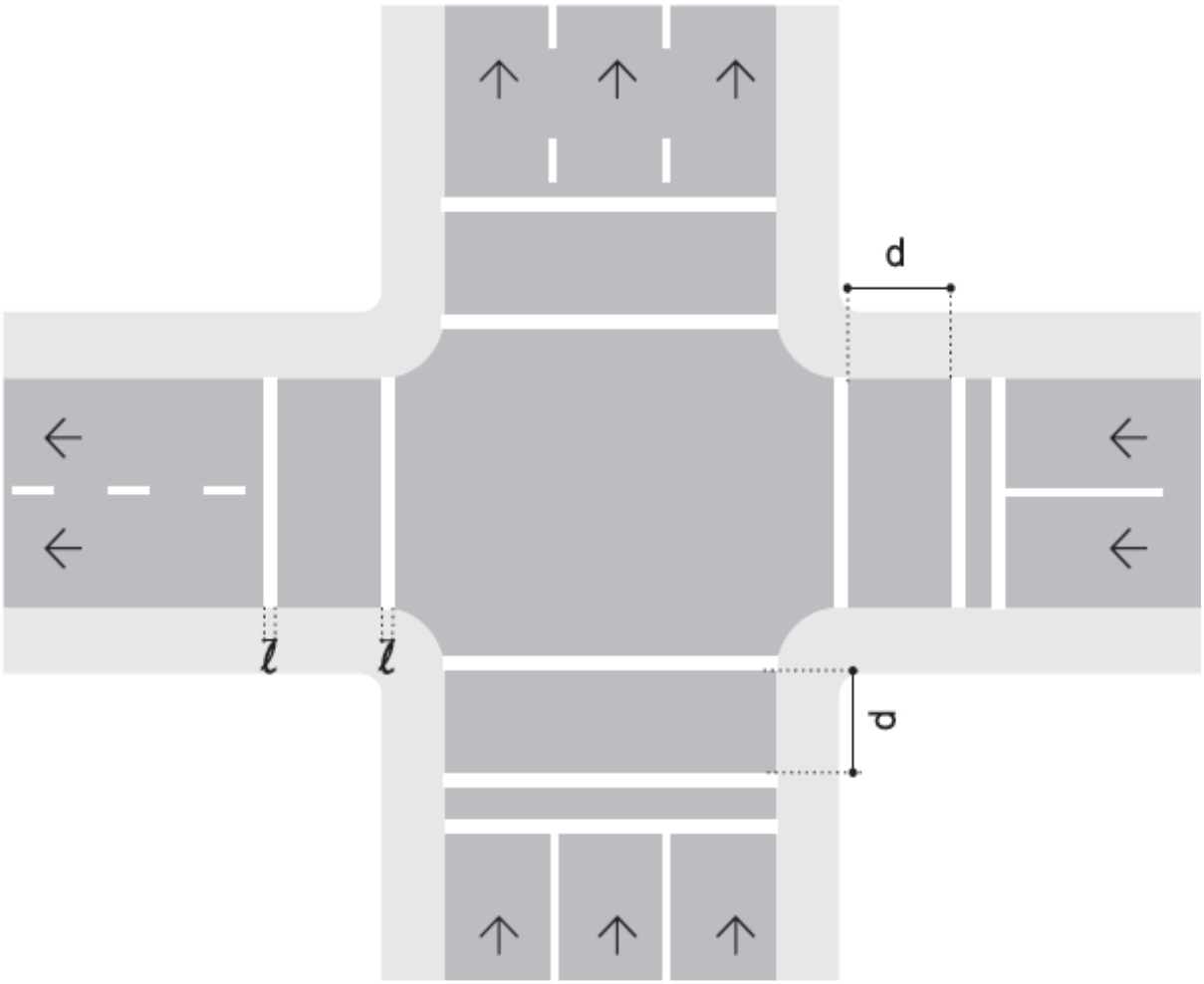


Figura 14: Padrão da FTP-2
Adaptado de (CONTRAN, 2007).

2.4 Tecnologias envolvidas

Para desenvolver o trabalho proposto neste projeto, será necessário utilizar algumas tecnologias. Nas seções seguintes são explicadas as principais tecnologias utilizadas para implementar o sistema proposto.

2.4.1 Linguagem de programação C/C++

Para a implementação dos algoritmos foi utilizada a linguagem de programação C++ para o protótipo, utilizando o compilador GCC em um computador com sistema operacional linux Ubuntu 12.04 TLS x68. A linguagem C++ consiste em uma linguagem de programação de alto nível e de propósito geral. É uma linguagem de programação compilada, fornecendo acesso de baixo nível à memória, estruturada e procedural. Foi criada em 1972 por Dennis

Ritchie para desenvolver o sistema operacional Unix (PUGH, 1990; DARNELL, 1991; DEITEL, 2001).

Para desenvolver um sistema nessa linguagem, podem ser utilizadas várias IDEs (*Integrated Development Environment*), bem como também, um simples editor de texto qualquer. Neste trabalho foi utilizada inicialmente a IDE Netbeans pelas facilidades e funcionalidades que a mesma oferece (NETBEANS, 2012). Entretanto, por conveniência e algumas dificuldades de integração com a biblioteca OpenCV, optou-se por desenvolver os algoritmos em um editor de texto simples (GEDIT).

2.4.2 A biblioteca OpenCV

Para desenvolver o sistema proposto neste trabalho, foram utilizadas algumas funções da biblioteca OpenCV, que disponibiliza funções complexas que podem ser utilizadas de forma simplificada em sistemas desenvolvidos nas linguagens C, C++, Python e Java. Esta foi desenvolvida pela empresa Intel (INTEL, 2012) e implementada nas linguagens C e C++, proporcionando suporte ao multiprocessamento e eficiência otimizada nos processadores desenvolvidos pela empresa (linha de processadores Intel; OPENCV, 2013).

A biblioteca OpenCV foi projetada com o intuito de tornar o desenvolvimento de sistemas de visão computacional mais rápido, eficiente e otimizado em diversas áreas, como a interação homem-máquina, em sistemas de tempo real e robótica. É uma biblioteca de código aberto e inclui recursos avançados para desenvolver sistemas que utilizam o processamento de imagens, efetuar transformações, analisar movimentos, calibrar a câmera, reconstrução 3D, detectar características e aprendizado de máquina. Atualmente, a biblioteca disponibiliza mais de 500 funções de PDI, reconhecimento de padrões e aprendizagem de máquina (OPENCV, 2013).

O pacote de arquivos que compõem a biblioteca OpenCV, bem como manuais de referência, são disponibilizados gratuitamente na internet (OPENCV, 2013). A biblioteca pode ser dividida em cinco grupos de funções básicas, conforme descrito abaixo:

A biblioteca OpenCV possui uma estrutura modular, em que seus pacotes incluem várias bibliotecas estáticas ou compartilhadas. Os principais módulos dessa biblioteca são:

- CORE: possui estruturas básicas de dados, e álgebra linear.
- IMGPROC: possui funções de filtragem linear e não-linear, transformações geométricas, conversões entre modelos de cor, histogramas, entre outros.

- VIDEO: este modulo traz funções para análise de vídeos, incluindo estimativas de movimentos, subtração de fundo e algoritmos de rastreamento de objetos.
- CALIB3D: possui algoritmos básicos de geometria, calibração de câmera e sistema de som, estimativa de posição de objetos e elementos de reconstrução 3D.
- OBJDETECT: possui funções de detecção de objetos e instâncias de classes pré-definidas, como por exemplo, rosto, olhos, pessoas e carros).
- HIGHGUI (*High-level Graphical User Interface*): disponibiliza funções relacionadas a interfaces gráficas com o usuário, além de entrada e saída de vídeos (OPENCV, 2013).

A Figura 15, a seguir, mostra a estrutura básica da biblioteca OpenCV. Esta disponibiliza mais de 500 funções que podem ser utilizadas em processamento de imagens.

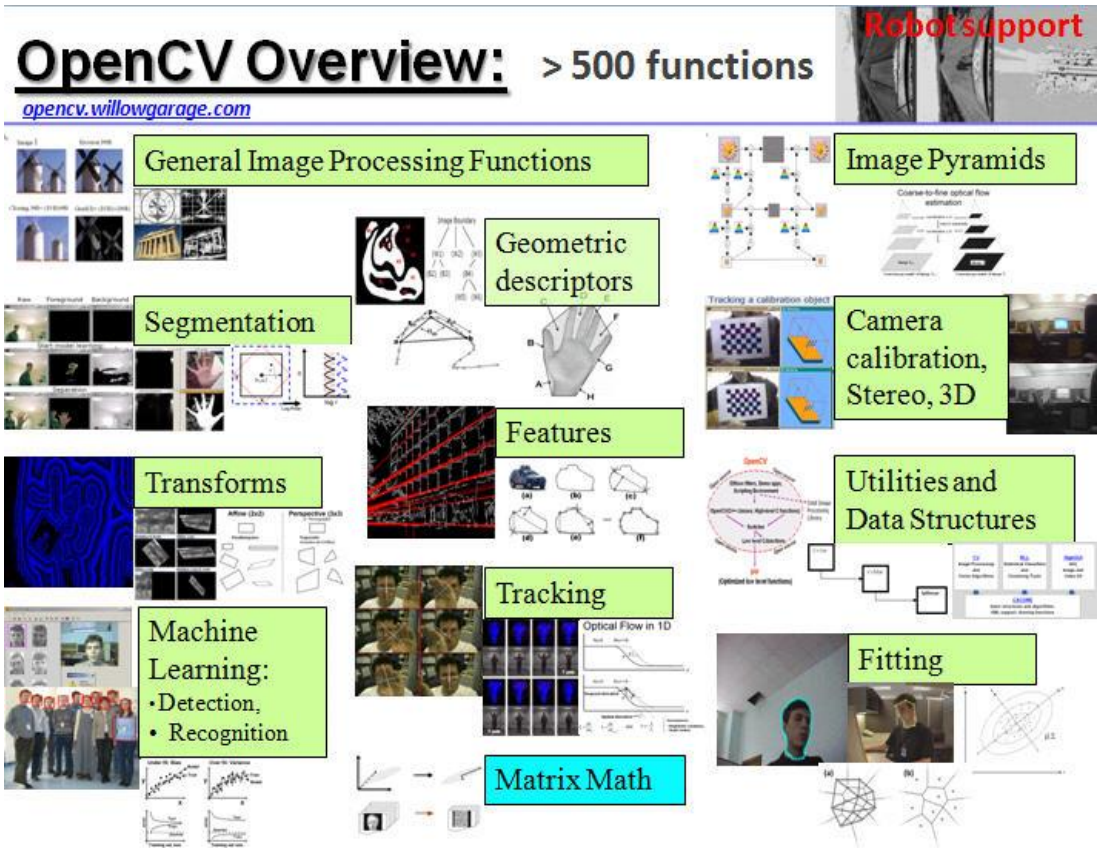


Figura 15: Estrutura da biblioteca Opencv

Fonte: Adaptado de (OPENCV WIKI, 2013)

2.4.3 Oracle VM Virtualbox

O sistema Oracle VM Virtualbox consiste em um sistema de virtualização, permitindo a criação de ambientes virtuais para a instalação de sistemas operacionais distintos dentro do sistema operacional principal do computador. Nestes ambientes virtuais, podem ser simulados ou emulados vários sistemas operacionais que compartilham o mesmo hardware, de forma a não alterar o sistema operacional principal do computador (VIRTUALBOX, 2013). A Figura 16, a seguir, mostra a arquitetura básica utilizada pelo Virtualbox.

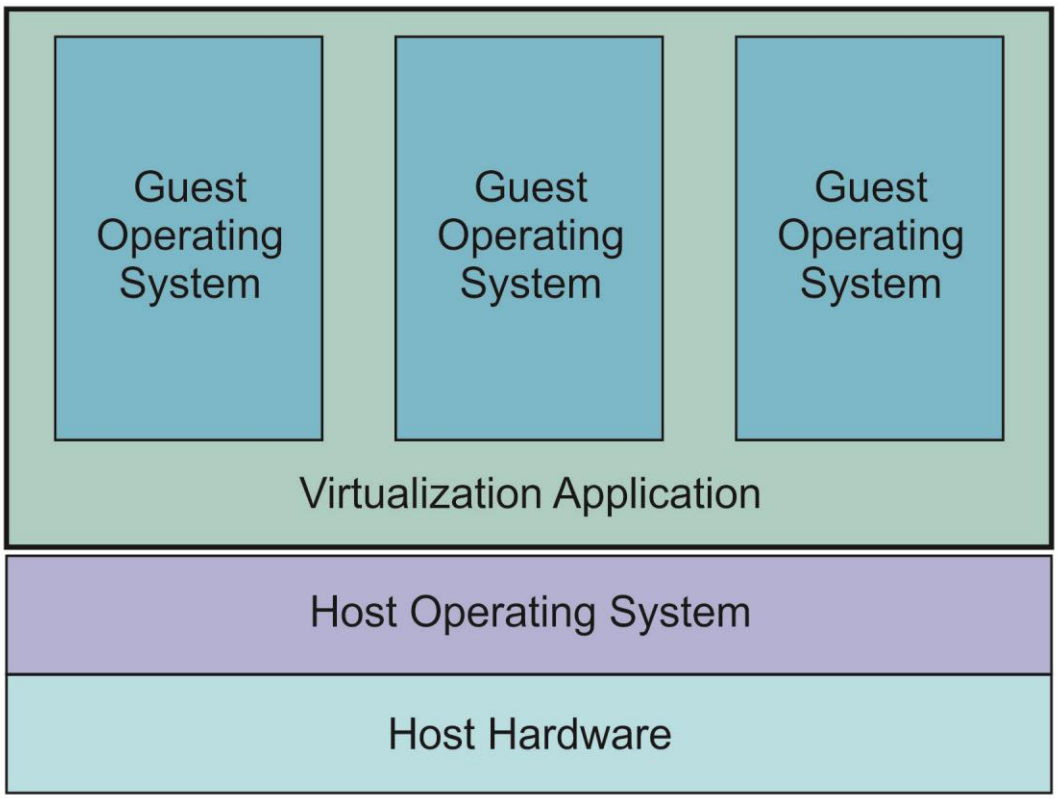


Figura 16: Arquitetura do Virtualbox

Fonte: Adaptado de (VIRTUALBOX, 2013)

Este sistema pode ser adquirido diretamente no site do desenvolvedor, sendo disponibilizado de forma gratuita como um software de código aberto, regulamentado sobre os termos da GNU GPL (*General Public License*). Pode ser utilizado vários sistemas operacionais, tais como as diversas versões do Microsoft Windows, Linux, Solaris, Open Solaris, OS/2 e OpenBSD. (VIRTUALBOX, 2013). Este software se mostra bastante útil para a realização de testes que necessitam de mais de um computador, bem como também, para testar sistemas desenvolvidos em diversas plataformas de sistemas operacionais

(VIRTUALBOX, 2013). A Figura 17, a seguir, mostra a tela principal do Virtualbox e uma máquina virtual instalada neste em execução.

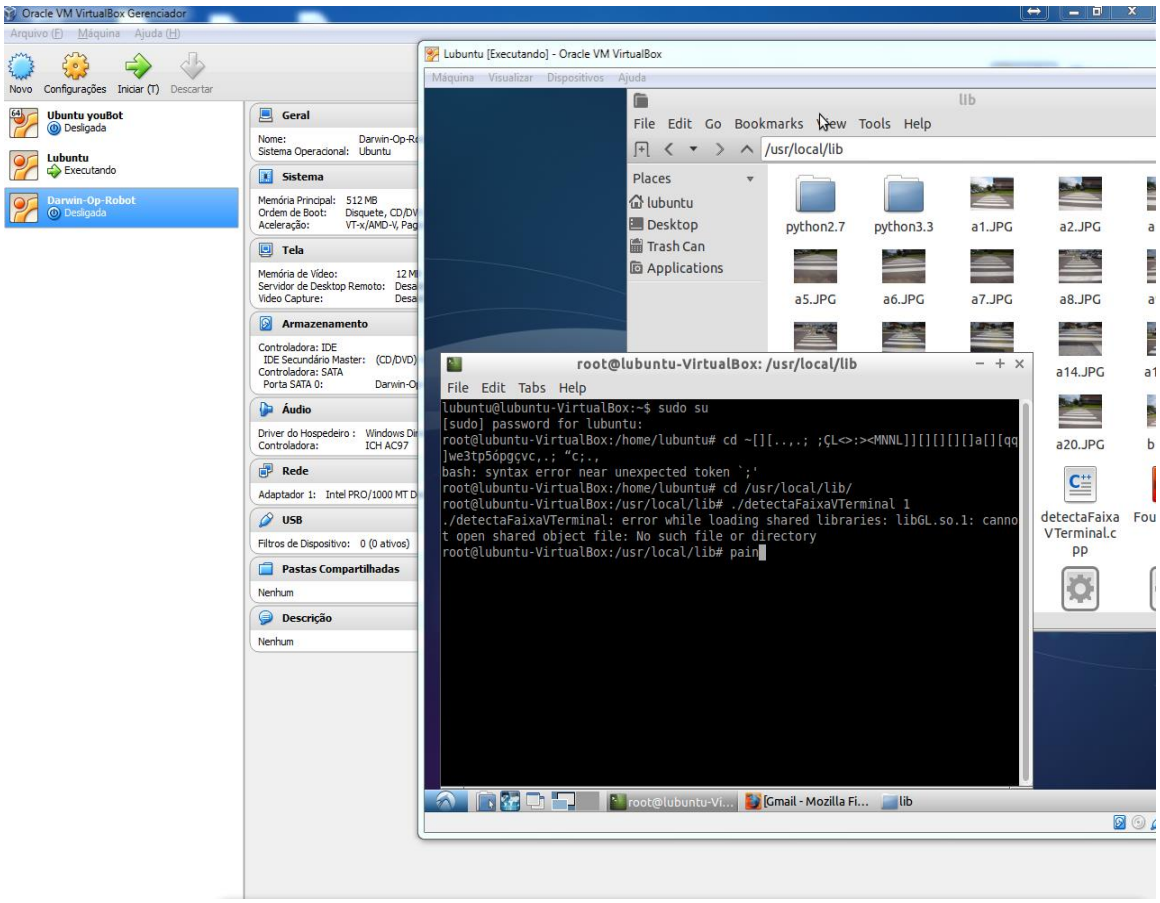


Figura 17: Imagem de uma máquina virtual no Virtualbox

Fonte: O Autor

2.4.4 Robô humanoide DARwIn-OP

O hardware utilizado para testes o sistema proposto será um robô humanoide modelo DARwIn-OP, desenvolvido pela empresa Sul-Coreana Robotis (ROBOTIS, 2012). Esse robô é um modelo educacional de plataforma aberta, voltado ao desenvolvimento de sistemas acadêmicos, em especial, para futebol de robôs (DARWIN-OP, 2012).

O robô humanoide DARwIn-OP possui boa capacidade de processamento, sensores sofisticados e capacidade de realizar movimentos dinâmicos. Sua altura é de 45,45 cm e seu peso de 2,9 kg. Seu controlador principal é composto por (DARWIN-OP, 2012):

- Um processador Intel Atom Z530, com 1 GHz de clock;
- 1 GB(Gigabyte) de memória RAM *(Random-access Memory);*
- 4 GB de capacidade de armazenamento em uma unidade *Solid State* NAND;
- Interface de comunicação de rede ethernet IEEE 802.3 gigabit;

- Interface de comunicação de rede sem fio IEEE 802.11 b/g/n;
- Duas portas USB (*Universal Serial Bus*);
- Uma saída de vídeo HDMI (*High-Definition Multimedia Interface*);
- Entrada e saída de áudio em alta definição;
- 1 câmera de alta definição (2MP Logitech C905).

A Figura 18 ilustra o robô humanoide DARwIn-OP.

Figura 18:Robô humanoide DARwIn-OP
(DARWIN-OP, 2012)

Originalmente, este equipamento possui sistema operacional Linux Ubuntu 9.10. Após algumas tentativas, foi constatado que este não possibilitava a execução do sistema desenvolvido devido à ausência de alguns recursos existentes em versões mais recentes do deste mesmo sistema operacional. Com isso, houve a necessidade de atualizar o mesmo. Após simulado o processo de atualização em uma máquina virtual, com sucesso, foi constatado que seriam necessários aproximadamente 9GB de espaço em disco para o sistema atualizado.

Com isso, observou-se uma limitação de hardware no robô, que possui apenas 4GB de espaço em disco, originalmente. Com isso, após algumas pesquisas, foi optado por uma versão mais simplificada deste sistema operacional, denominada Lubuntu. Essa versão do Ubuntu também foi simulada e testada em uma máquina virtual, na qual o sistema desenvolvido foi executado com sucesso.

2.5 Trabalhos Semelhantes

Durante as pesquisas realizadas para a realização deste trabalho, vários projetos semelhantes foram encontrados. A abordagem utilizada no presente trabalho não é exatamente a mesma da encontrada em outros trabalhos, mas o estudo destes foi relevante para a compreensão de alguns conceitos e diferentes técnicas possíveis. Na sequência, são descritos alguns destes trabalhos.

No trabalho de (SOUSA, 2007) é utilizada a biblioteca OpenCV para desenvolver um sistema de detecção da FTP em imagens. Esse sistema é direcionado para dispositivos móveis com sistema operacional Android na versão 2.3. Inicialmente é detectada a FTP e em seguida é verificada a presença de veículos e/ou pessoas sobre esta faixa para definir se a travessia é segura ou não. Nesta proposta, a interação com o usuário é feita através de comandos de voz e do recurso de vibração do aparelho móvel utilizado. Além disso, as imagens são convertidas em escalas de cinza, diferentemente da proposta apresentada nesse trabalho, onde se pretende realizar os processamentos em imagens coloridas.

Outra proposta de sistema autônomo de reconhecimento de sinais de trânsito em imagens pode ser encontrada em (BARGHAVA, 2010). Nesta, é identificado o semáforo em imagens e verificado o seu estado. A arquitetura proposta consiste na instalação de micro câmeras na armação de um óculos de sol. Estas micro câmeras se comunicam com um dispositivo móvel (aparelho celular, por exemplo) com acesso à internet. Este, por sua vez, envia a imagem para um servidor remoto, o qual realiza o processamento e retorna o resultado para o dispositivo móvel. Logo, o processamento é feito em nuvem e não é local, conforme proposto neste projeto. Para identificar a localização atual do usuário, o sistema utiliza-se de coordenadas de sinal de GPS e se comunica com o servidor Google Maps. O sistema captura a imagem e envia para o servidor, o qual realiza o processamento. Na sequência, o sistema informa ao usuário, em forma de comando de voz no dispositivo móvel, quando o momento é seguro para atravessar a rua (BARGHAVA, 2010).

Em seu trabalho, (KIM, 2011) apresenta um método de identificação da sinaleira de trânsito e o seu estado para sistemas veiculares, em ambiente real, considerando apenas o turno diurno. É utilizado o modelo de cores *YCbYr*, uma extensão do modelo *YUV*. São propostos pré-processamentos para eliminar ruídos na imagem e a área e objetos que não são de interesse, tais como pessoas, carros, prédios, entre outros, trabalhando apenas com a parte da imagem que representa a sinaleira. Após isso, são encontrados os BLOBs, para os quais

existem predefinições para avaliar se os mesmos representam as luzes da sinaleira, de acordo com seu formato, tamanho e cor (KIM, 2011).

O trabalho de (KIM, 2011) se diferencia do proposto neste projeto em vários aspectos. Um deles é o fato do processamento ser feito em um microcomputador normal ao invés de ser utilizado um hardware mais específico, com limitações. Outro aspecto é o fato do sistema ser voltado para veículos autônomos e não para robôs autônomos, como o modelo proposto nesse trabalho. Apesar disso, o estudo do mesmo é de suma importância para a compreensão e entendimento de várias técnicas de processamento de imagens utilizadas, principalmente nos pré-processamentos e na identificação da sinaleira.

Outra proposta semelhante, encontrado na literatura, é apresentada por (GUO-SHENG, 2009). O objetivo dos autores é apresentar um sistema para veículos autônomos capaz de evitar acidentes de trânsito, em especial, atropelamento de pessoas sobre a FTP. Inicialmente, é detectada a FTP, como área de interesse na imagem. Em seguida, é verificada a presença de um pedestre sobre essa faixa. Na sequência é calculado o momento (tempo) em que o veículo alcança a FTP, de acordo com a distância entre a posição atual do veículo, a FTP e a velocidade atual do veículo. É utilizado o método probabilístico kalman (THRUN, 2005) para estimar a posição do pedestre no momento em que o veículo alcança a FTP. A partir dessas informações, o sistema de controle do veículo gerencia a velocidade do mesmo, para que este prossiga na atual velocidade, reduza ou até mesmo pare, para evitar uma colisão com o pedestre, de acordo com o cenário (GUO-SHENG, 2009).

Em (SICHELSCHMIDT, 2010) é apresentada outra proposta de sistema com propósito semelhante ao deste trabalho. Os pesquisadores alemães apresentam um sistema de detecção de FTP do tipo zebrada utilizando a transformada de Fourier, bipolaridade aumentada e a relação de orientação de borda para a classificação. O processamento ocorre em imagens em tons de cinza. O fluxo básico do sistema consiste em: a) capturar a imagem em tons de cinza, através de uma câmera. b) segmentar a imagem. c) extração de características. d) classificação. e) pós-processamento (SICHELSCHMIDT, 2010).

O processo de segmentação é estruturado de forma bastante interessante, sendo feito um processamento específico para compensar o ângulo de inclinação da imagem e determinada a região de interesse. Na sequência, é feita a equalização do histograma para aprimorar o contraste da imagem para que possa ser aplicada a detecção de bordas com maior eficiência. É adotado o algoritmo de *CANNY* para a detecção das bordas, para auxiliar na posterior

identificação das faixas. É adotada também a técnica de dilatação para contornar problemas de FTP irregulares e/ou apagadas parcialmente. O sexto passo é a extração dos contornos. A seguir, é aplicada a transformada de Fourier em cada linha da imagem, visto que o padrão de FTP indica uma determinada largura das linhas brancas da faixa, que é igual ao espaço entre elas. Finalmente é utilizada a abordagem da transformada de Hough, para excluir linhas que não pertencem a FTP (SICHELSCHMIDT, 2010).

Para evitar a taxa de erros, ou seja, reconhecimento incorreto da FTP, é realizada uma etapa de pós-processamento. Nesta, é utilizado um algoritmo de rastreamento baseado no filtro de Kalman, o que exige mais processamento e consequentemente torna o reconhecimento mais lento. Entretanto, com o uso dessa técnica, são diminuídos os reconhecimentos incorretos, tornando o sistema mais confiável (SICHELSCHMIDT, 2010).

Os trabalhos explicados nesta seção apresentaram bons resultados, porém, em sua maioria as imagens são binarizadas ou convertidas para escala de cinza. Além disso, em nenhum deles os testes são realizados com um hardware similar ao utilizado neste trabalho, onde as imagens são ainda tratadas de forma diferente, de acordo com as condições climáticas.

3 METODOLOGIA

Neste capítulo são explicadas as características desta pesquisa quanto aos objetivos, aos procedimentos de coleta e quanto à fonte de informação. Além disso, são apresentados os procedimentos metodológicos utilizados e os passos realizados para alcançar os objetivos propostos.

3.1 Métodos de pesquisa

Este trabalho de pesquisa possui caráter exploratório. Segundo (SILVA, 2005), esse tipo de pesquisa proporciona maior interação e familiaridade com o problema abordado, de forma a torná-lo mais claro e definir hipóteses. É feita a revisão bibliográfica para estudar os principais conceitos teóricos envolvidos, realizadas entrevistas com pessoas ligadas ao problema, estudo de casos, entre outros.

Além do modelo exploratório, este trabalho também possui características do modelo descritivo. Esse modelo de pesquisa tem por objetivo fazer o levantamento e a descrição das características conhecidas, bem como também, estabelecer relações entre variáveis envolvidas. São utilizadas técnicas padronizadas para fazer a coleta de dados e uma observação sistemática, assumindo a forma de levantamento (SILVA, 2005). Para realizar esse tipo de pesquisa, deve ser feito um levantamento bibliográfico detalhado (KOCHE, 2001).

De acordo com o ponto de vista dos procedimentos técnicos, (SILVA, 2005) acrescenta que o modelo de pesquisa adotado neste trabalho pode ainda ser classificado como pesquisa experimental. Esse modelo consiste na determinação de objeto de estudo e na seleção das variáveis que podem influenciá-lo. São definidas as maneiras de controlar e observar os efeitos destas variáveis sobre o objeto (SILVA, 2005).

3.2 Resumo dos procedimentos metodológicos

Para um melhor entendimento e compreensão dos procedimentos propostos e utilizados, é apresentado a seguir um resumo dos processos realizados para o desenvolvimento dessa pesquisa. Nesta seção são apresentados os procedimentos metodológicos utilizados. Inicialmente, o trabalho foi dividido em duas etapas, conforme descrito a seguir.

3.2.1 Primeira etapa – Estudos e levantamentos bibliográficos

Para desenvolver um sistema de visão computacional é necessário estudar e definir algumas técnicas de PDI. Inicialmente, foram pesquisados e estudados outros trabalhos semelhantes a este na literatura, a fim de extrair informações relevantes. Foi feito também um estudo aprofundado das técnicas de PDI existentes na literatura e a definição de quais seriam utilizadas para desenvolver o sistema.

Ainda nesta etapa, foram estudadas as principais funções da biblioteca OpenCV. Isso possibilitou uma avaliação para definir quais as funções podem ser exploradas e utilizadas nos algoritmos desenvolvidos.

3.2.2 Segunda etapa – Preparação da infraestrutura e coleta das amostras

As amostras das imagens foram capturadas por uma câmera digital conectada a um computador. Foram coletadas imagens de um cenário de trânsito, destacando a FTP, sendo necessário:

- Obter imagens onde a FTP aparece, a partir de diferentes alturas;
- Obter imagens onde a FTP aparece, com diferentes luminosidades no ambiente;
- Obter imagens onde a FTP aparece parcialmente apagada/danificada;
- Obter imagens onde a FTP não aparece;
- Obter imagens onde a FTP aparece perfeitamente;
- Obter imagens onde a primeira faixa aparece parcialmente;

A partir destas imagens, foi feita a análise do sistema proposto. Nesta análise, foi definido como pode aparecer uma FTP na imagem, o que a caracteriza e como identificá-la. Além disso, foram definidos os pré-processamentos necessários para a utilização das imagens na etapa de extração das características.

Dentre esses pré-processamentos, podem ser destacados a segmentação da imagem, calibração de cores, eliminação de ruídos, entre outros. Foram definidos também os requisitos de tempo de resposta do sistema.

Ainda nessa etapa, foi montada a estrutura tecnológica com as ferramentas de *hardware* e de *software* necessárias ao desenvolvimento do sistema proposto. Foi realizada a aquisição dos *softwares*, todos de uso acadêmico e/ou gratuito (não havendo a necessidade de investimentos financeiros), e a instalação e configuração dos mesmos.

Foi realizado também um estudo das principais tecnologias utilizadas neste trabalho. Foi definido também a forma de interação do sistema com o usuário e os procedimentos para testes com definição dos resultados esperados.

3.2.3 Terceira etapa – Desenvolvimento do sistema e testes

Nesta etapa, foi desenvolvido um protótipo do sistema para ser executado em um microcomputador normal. Todas as funcionalidades projetadas foram implementadas nesse protótipo, para possibilitar testes completos do sistema. Esse protótipo se faz necessário pois facilita e agiliza a etapa de testes e correção de eventuais falhas. O desenvolvimento desse protótipo foi realizado utilizando praticamente as mesmas tecnologias básicas do sistema embarcado no robô, ou seja, desenvolvido em linguagem C++ com o uso de funções da biblioteca OpenCV.

Após a aprovação do protótipo desenvolvido, os algoritmos do sistema foram portados e integrados ao sistema principal de funcionamento do robô humanoide DARwIn-OP. A partir desta integração, com as adaptações necessárias, o sistema foi testado novamente com imagens de um ambiente real de trânsito. Inicialmente, foi avaliada a corretude do sistema quanto aos resultados apresentados processando as imagens capturadas, verificando a correta detecção da FTP na imagem. Na sequência foi avaliado também o desempenho do sistema quanto ao tempo necessário para os processamentos.

Foi utilizado um robô humanoide DARwIn-OP para efetuar testes em um *hardware* mais limitado. Além disso, o sistema aqui proposto pode ser utilizado para complementar um sistema de visão computacional completo para esse *hardware*. Juntamente com os testes foram feitas as correções das falhas apresentadas pelo sistema, bem como também, as otimizações do mesmo.

4 DESENVOLVIMENTO DO SISTEMA PROPOSTO

Após os estudos e pesquisas bibliográficas realizadas, foi iniciado o desenvolvimento e implementação do sistema proposto neste trabalho. Este sistema é apresentado e explicado em detalhes neste capítulo.

4.1 Algoritmo de detecção da FTP

Os algoritmos de detecção da FTP desenvolvidos, seguem o fluxo mostrado na Figura 19, a seguir. Nesta são ilustradas as etapas básicas necessárias para realizar a detecção da FTP.

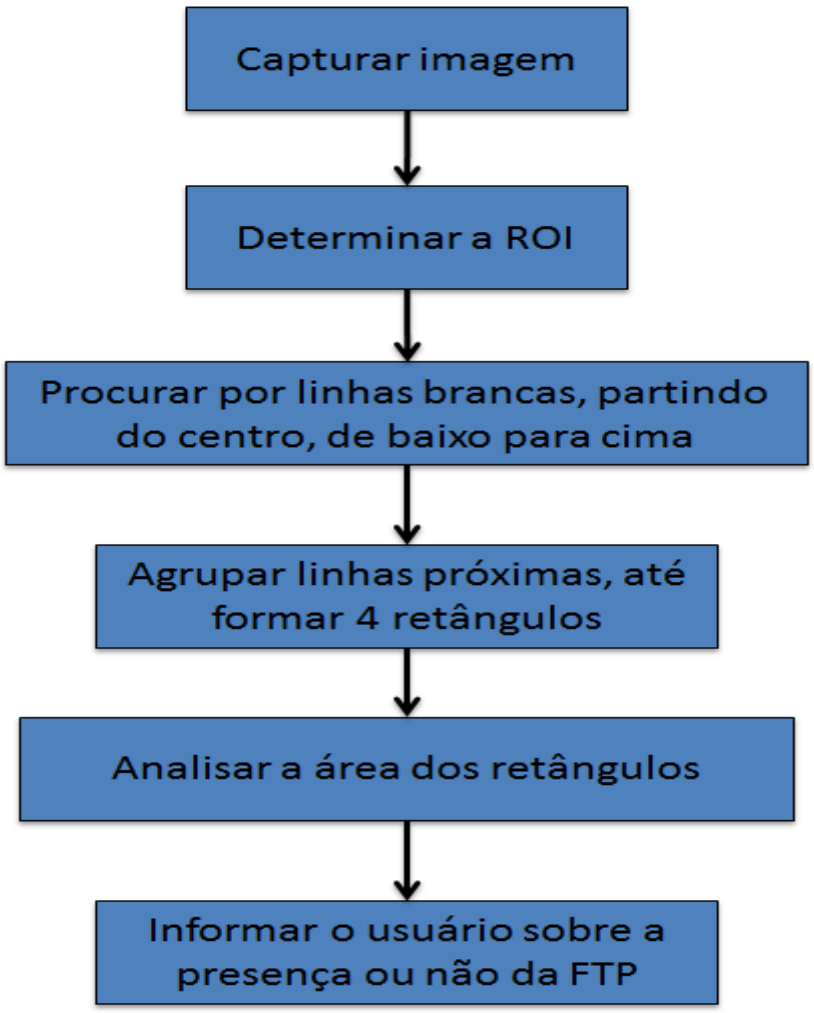


Figura 19: Fluxograma do sistema proposto

Fonte: O Autor

Inicialmente, o sistema faz a captura da imagem a partir da câmera do humanoide DARwIn-OP, através da função OpenCV *cvCaptureFromCAM*. Na sequência é utilizada a

função OpenCV *cvQueryFrame* para extrair um quadro (*frame*), obtendo assim a imagem a analisar. Esta imagem possui o padrão de cores RGB, com resolução de 1152 x 864 pixels. A classe OpenCV *Mat* e a estrutura OpenCV *IplImage* são utilizadas para armazenar a imagem.

Na etapa seguinte, o algoritmo localiza traços individuais de pixels brancos, que agrupados podem representar uma linha (faixa) da FTP. Como existem irregularidades na pintura das FTPs, o sistema não pode procurar apenas por traços onde todos os pixels sejam brancos. Em função disso, são analisados grupos de matrizes M*ij* de pixels.

Se na matriz M*ij* o sistema identificar mais de 50% de pixels brancos, segue para a próxima, até os limites laterais da imagem. O limite mínimo de 50% foi definido empiricamente, sendo que desta forma, o sistema consegue contornar parcialmente problemas de ruído e FTPs cuja pintura está danificada e/ou apagada.

O sistema considera matrizes quadradas, por definição do autor. Além disso, as dimensões devem ser ímpares (3,5,7 ou 9), pois, caso for encontrado um traço válido, este é mapeado na linha central do grupo de matrizes analisado. Por exemplo, em um grupo de matrizes 3 x 3, o traço estaria na segunda linha deste.

Se determinada matriz não apresentar uma quantidade mínima de 50% de pixels brancos, esta é adicionada a um contador. Caso esse contador ultrapasse 5% da largura total da imagem, o sistema considera que ali é uma extremidade do traço, seja a extremidade esquerda ou a extremidade direita. Este limite de 5% também foi definido como valor ótimo baseado em testes. Verificou-se que se usar um limite maior, o sistema apresenta resultados incorretos, pois agrupa regiões de pixels brancos indevidamente. Utilizando um percentual menor, o sistema mostrou-se ineficiente em imagens com ruídos e FTPs parcialmente apagadas ou obstruídas por pequenos objetos.

Para determinar um traço branco, a largura deste traço deve ser maior que X% da largura total da imagem. O valor de “X” é flexível na interface gráfica do protótipo. Baseado em testes, foi verificado que o valor ótimo é 30%, sendo este utilizado no sistema embarcado no robô.

O processamento inicia na parte inferior da imagem, visto que a FTP geralmente começa nessa posição, considerando a sua localização em relação ao usuário. A dimensão da matriz M*ij* e o percentual de pixels analisados que devem ser considerados brancos, em relação a largura total da imagem, foram implementados de forma flexível no protótipo do sistema, permitindo alterar estes valores afim de identificar o padrão mais indicado para o

reconhecimento da FTP. Para identificar a cor do pixel é utilizada a função *at* da classe OpenCV *Mat*. Esta função retorna os valores RGB, que são armazenados em uma estrutura criada pela classe OpenCV *Vec* para serem analisados individualmente.

Ainda nesta etapa, é determina a ROI (*Region of Interest*), que é a região de interesse. Esta região é determinada em tempo de execução, sendo que o sistema processa a imagem apenas até encontrar quatro grupos de traços brancos, que são as possíveis linhas da FTP. Desta forma, é possível otimizar o processamento.

Optou-se por identificar apenas quatro linhas da FTP, pois, baseado em testes, é necessário identificar apenas três linhas para determinar que existe uma FTP. Dependendo da posição do usuário, a primeira linha pode aparecer parcialmente na imagem não sendo reconhecida como pertencente a uma FTP. Além disso, as linhas podem estar parcialmente ocultadas com a presença de veículos ou pessoas. Se esta oclusão ocorrer apenas a partir da quarta linha, o sistema consegue identificar a FTP, pois necessita encontrar apenas três linhas.

A Figura 20, a seguir, ilustra como esses traços brancos são mapeados pelo sistema. Cada "quadro" da imagem representa um pixel da imagem. As linhas verdes definem os agrupamentos de pixels analisados, considerando matrizes com dimensão 3x3.

As setas, na parte inferior, indicam que o processamento inicia no centro em direção às extremidades da imagem. Em vermelho são ilustrados os traços identificados, que são agrupados na etapa seguinte, em que são avaliados individualmente para verificar se correspondem às linhas da FTP. Nestes traços, formados a partir do conjunto de matrizes de pixels analisado, o somatório de pixels brancos deve ser maior que X% (na ilustração, X = 30) da largura total da imagem. Este percentual foi definido com base em testes e auxilia a evitar resultados indevidos, pois, geralmente, a FTP ocupa mais que esse limite definido para a largura das linhas nas imagens analisadas.

Ainda na Figura 20, o traço T1, por exemplo, possui 15, pixels, o que representa 71,4% da largura total, ou ainda, é maior que 6,3 (30% da largura total da imagem). Analisando a quarta matriz partindo do centro para a direita, na formação desse traço, são identificados apenas 2 pixels brancos, equivalentes a apenas 33% dos pixels, por isso, o traço termina no último pixel da matriz anterior (terceira).

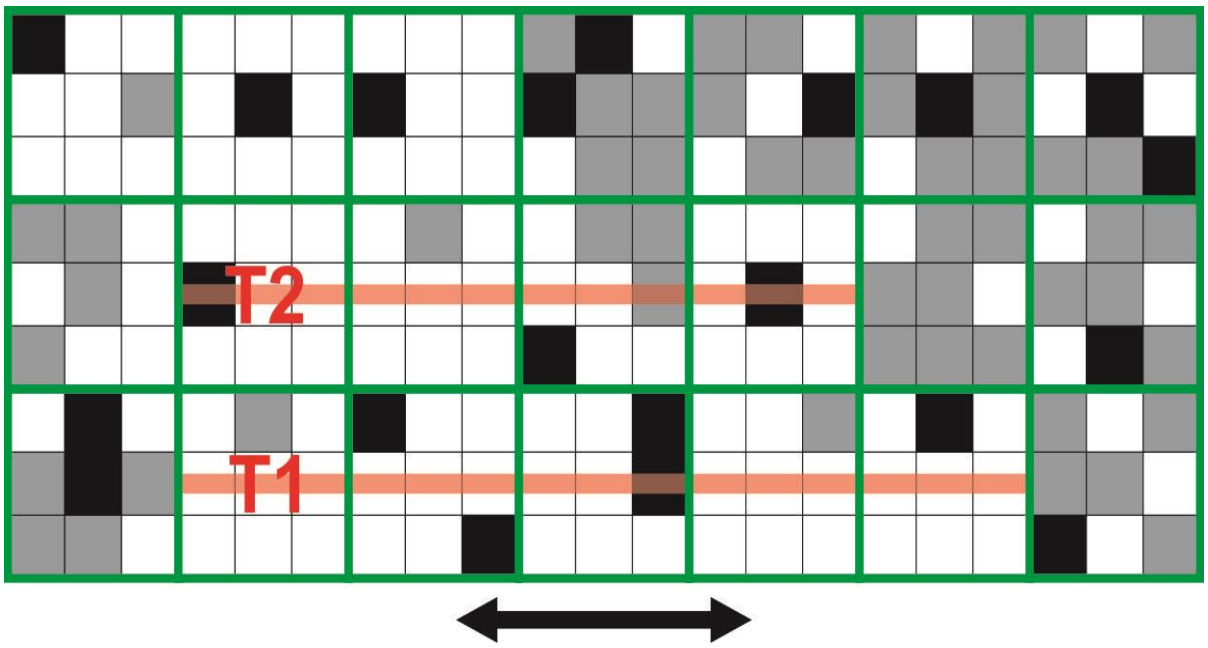


Figura 20: Busca por traços brancos da FTP

Fonte: O Autor

Na sequência, os traços brancos próximos são agrupados em forma de retângulos, que representam as possíveis linhas da FTP. Para traçar esses retângulos, o algoritmo considera o primeiro traço e assume seus extremos laterais como sendo os do retângulo a formar. A seguir, analisa o próximo traço, verificando se o mesmo é adjacente ao anterior. Se o mesmo estiver imediatamente ao lado, é considerado parte do retângulo, e, caso um de seus extremos laterais esteja mais distante do centro que os atuais do retângulo, assume este valor como novo extremo do retângulo.

Esse processo continua até ser encontrado um traço que não é imediatamente adjacente ao traço anterior. Neste caso, é considerado que ali termina um retângulo, o qual é armazenando utilizando a classe OpenCv Rec. É utilizada a classe OpenCv cvPoint, informando o ponto superior esquerdo e o ponto inferior direito do retângulo, formados pela altura e largura obtida com o agrupamento dos traços. O traço analisado que não era imediatamente adjacente ao anterior é então considerado o primeiro traço do próximo retângulo. Este processo é realizado enquanto existirem traços brancos ou até formar quatro retângulos.

A partir da formação dos retângulos, o sistema calcula a área destes para comparar entre os demais retângulos brancos encontrados. A área do retângulo é calculada pela função OpenCV *área*, que pertence à classe *Rect*.

De acordo com (NOGUEIRA, 2009; HARLEY, 2000; XU, 1996; COXETER,1993; KANNALA, 2010; FAUGERAS, 1993), sendo objetos iguais dispostos paralelamente em uma imagem, quanto mais distante o objeto, menor será a sua área na mesma. As linhas da FTP podem ser comparadas aos trilhos de trem, conforme a ilustração feita na seção 2.1.10, através da Figura 11.

É possível ainda, afirmar que a pista forma um plano ortogonal ao plano de projeção e que as linhas são pintadas na pista com orientação paralela ao plano da câmera, considerando que a área seja medida no plano da imagem. Considerando o ponto de vista do usuário (posição da câmera), podem ser traçadas retas imaginárias que convergem em um único PF. (NOGUEIRA, 2009; HARLEY, 2000; XU, 1996; COXETER,1993; KANNALA, 2010; FAUGERAS, 1993).

A Figura 21, a seguir ilustra o PF, a LH e as LF sobre uma imagem contendo uma FTP. Ainda nesta imagem, pode ser observado que as linhas mais distantes aparecem menores, desvanecendo à distância.

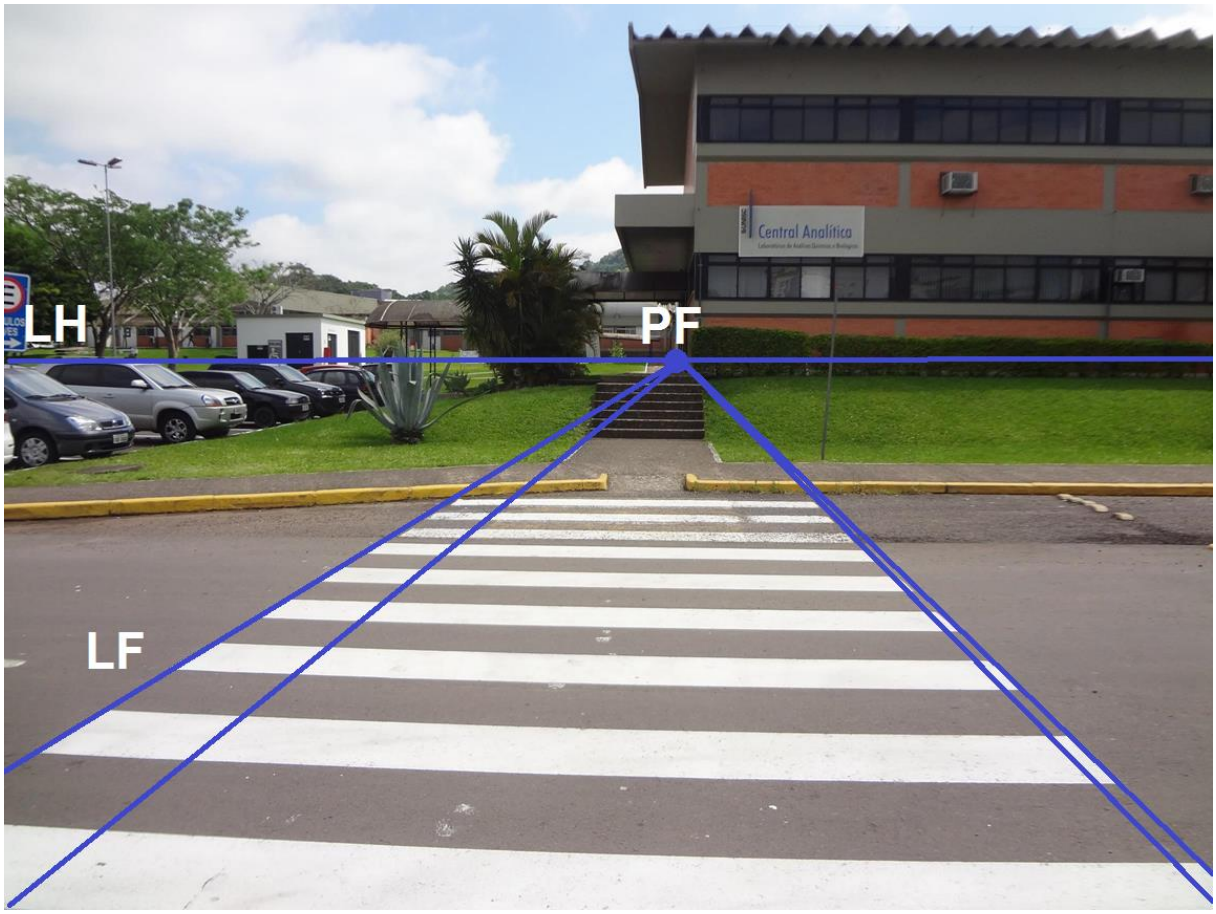


Figura 21: Elementos da perspectiva sobre a FTP

Fonte: O Autor

Considerando as linhas *L* das FTPs como objetos de tamanho e características idênticas dispostos paralelamente na pista, pode-se afirmar que a área da linha seguinte sempre é maior que a área calculada para a anterior.

Em experimentos, observou-se que essa diferença de área é de aproximadamente 30% a 70%, sendo esta considerada nos algoritmos. Desta forma, considerando *L*1 e *L*2 como linhas da FTP, sendo *L*2 a linha seguinte a *L*1, a área de *L2* deve ter entre 30% e 70% da área de *L1* para ser considerada válida, e assim sucessivamente.

4.2 O sistema proposto

Um dos objetivos deste trabalho foi testar um sistema desenvolvido com o uso de funções OpenCV em um hardware limitado. Optou-se em utilizar o robô DARwIn-OP por este estar disponível na universidade e também para iniciar o desenvolvimento de um sistema completo de visão computacional para o mesmo. Além disso, o sistema aqui proposto pode ser utilizado para complementar um sistema de visão computacional completo para esse hardware. Juntamente com os testes serão feitas as correções de possíveis falhas que possam existir no sistema, bem como também, as otimizações do mesmo.

Um dos pontos fundamentais observados através dos experimentos é a posição onde a imagem da FTP for capturada. Inicialmente, supõe-se que o usuário (no caso, o robô) esteja posicionado exatamente em uma via, onde pretende verificar a existência FTP.

O objetivo principal deste sistema é identificar a existência de uma FTP em uma rua qualquer. Inicialmente, foi desenvolvido um protótipo do sistema, com interface gráfica para permitir testes e análises detalhadas do mesmo, sendo executado em um microcomputador e não no humanoide. A Figura 22, a seguir, mostra o fluxo do sistema proposto.

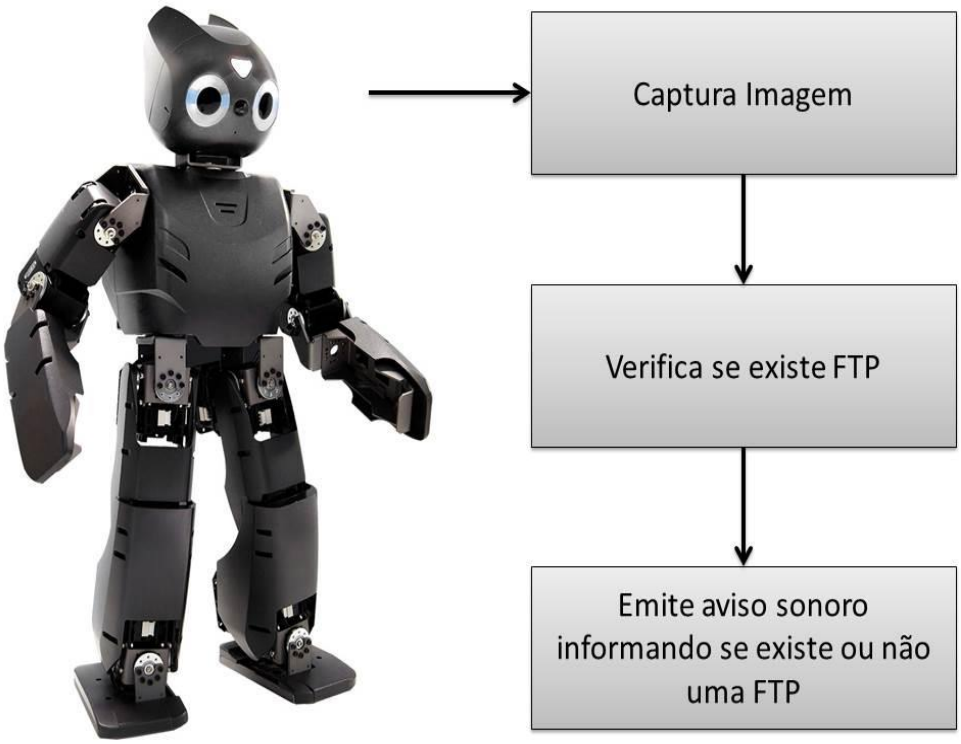


Figura 22: Fluxo do sistema desenvolvido

Fonte: O Autor

O protótipo foi desenvolvido utilizando as mesmas ferramentas de software que o sistema final, com a diferença de possuir uma interface gráfica e não capturar imagens a partir de uma câmera ligada ao sistema. São processadas imagens obtidas por uma câmera fotográfica digital de alta resolução, cujas imagens tiveram suas resoluções reduzidas a fim de aproximarem-se das geradas pela câmera do robô DARwIn-OP. Para realizar esta redução, foi utilizada uma versão gratuita do sistema FOTOSIZER v2.6.0.538 (FOTOSIZER, 2013).

No protótipo desenvolvido, é possível alterar vários parâmetros que interferem diretamente no resultado, através de um painel de controle implementado. A Figura 23, a seguir, mostra a tela deste painel de controle.

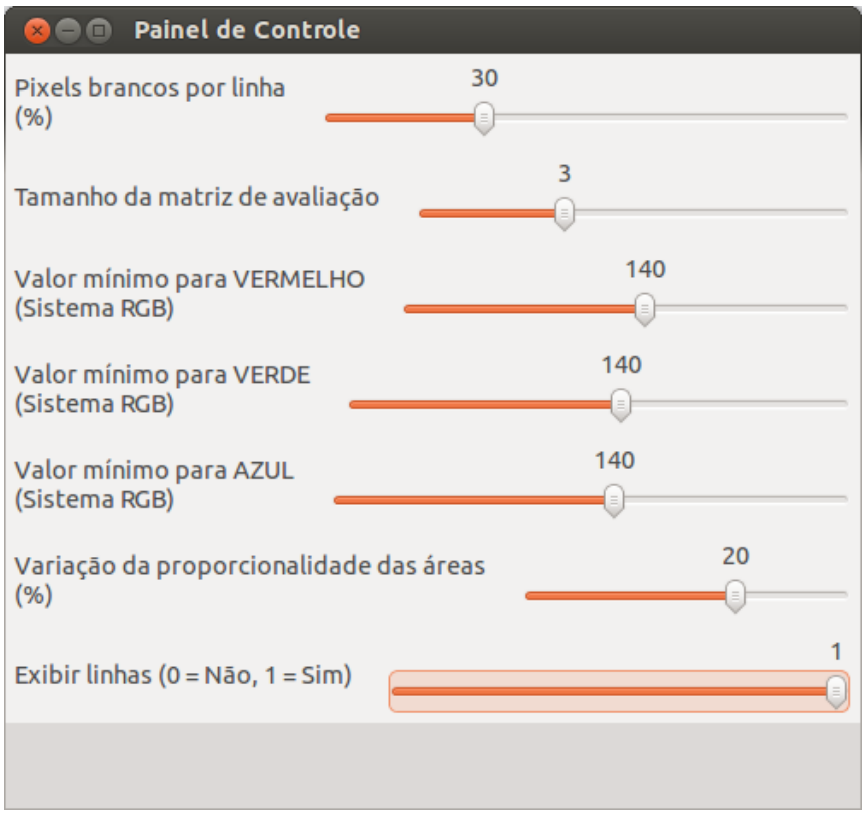


Figura 23: Painel de controle das variáveis no protótipo

Fonte: O Autor

A primeira variável, indicada pela informação “Pixels brancos por linha (%)” na tela do painel de controle, permite definir o percentual de pixels brancos que devem ser encontrados no conjunto de matrizes M*ij* analisadas em relação à largura total da imagem, com o objetivo de identificar os traços de pixels brancos, conforme descrito na seção anterior.

A partir da variável seguinte, indicada pela informação “Tamanho da matriz de avaliação” é possível determinar as dimensões desta matriz, entre os valores 3,5,7 e 9. Nas três variáveis seguintes, é possível definir o valor mínimo de R, G e B para considerar o pixel analisado como branco. Na variável “Variação da proporcionalidade das áreas” é possível configurar a tolerância, em percentual, em relação à regra da área das linhas da FTP. Conforme explicado na seção anterior, esta regra define que a primeira linha deve ser maior que a seguinte, podendo variar em sua área conforme o valor dessa variável, para mais ou para menos. A partir da última variável, “Exibir linhas” é possível configurar a visualização (valor = 1) dos traços mapeados. Se o valor desta for ZERO, o sistema não mostra estes traços.

O sistema informa o usuário sobre a presença ou não da FTP a partir de uma indicação em forma de texto sobre a imagem processada. A Figura 24, a seguir, ilustra uma imagem processada, na interface gráfica do protótipo desenvolvido.

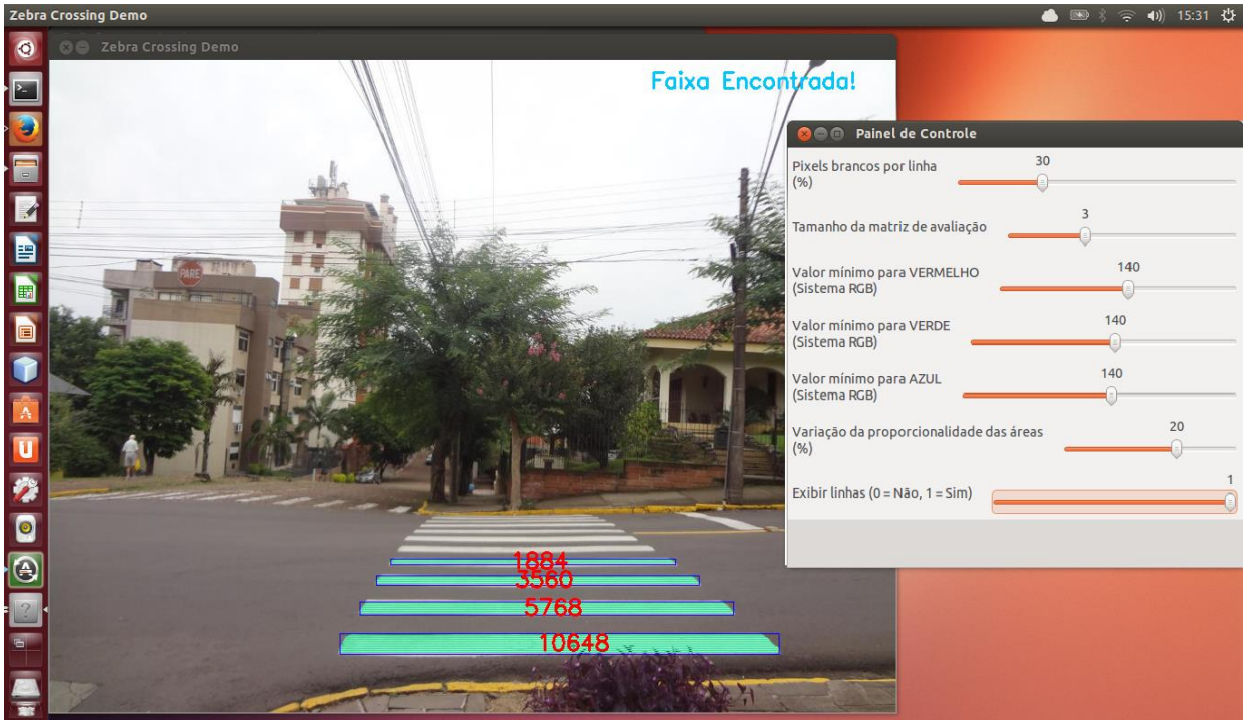


Figura 24: Interface gráfica do protótipo

Fonte: O Autor

Após os testes realizados com imagens através deste protótipo, que validaram o mesmo, foi desenvolvido o sistema a ser embarcado no robô DARwIn-OP. Foi retirada a interface gráfica e o resultado (se encontrou a FTP ou não) passou a ser apresentado em forma de voz. Para apresentar o resultado, foram gravados dois arquivo de áudio, no formato MP3, sendo um com a gravação “FTP encontrada” e outro “FTP não encontrada”. Estes arquivos são executados pelo sistema operacional do robô DARwIn-OP para indicar que a FTP não foi encontrada ou que foi encontrada, respectivamente. O sistema também mostra, em uma tela de terminal, o resultado do processamento, conforme ilustrado na Figura 25, a seguir.

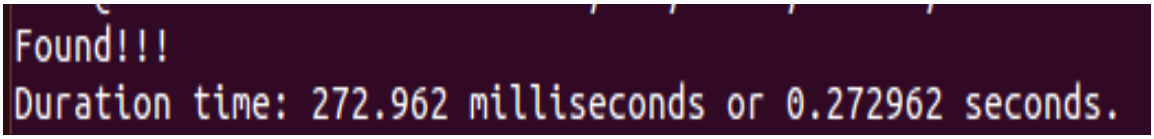


Figura 25: Sistema embarcado mostra o resultado em um terminal

Fonte: O Autor

Para possibilitar a execução do sistema desenvolvido em outro hardware (DARwIn-OP) sem a necessidade de instalar OpenCV, é necessário realizar algumas configurações e copiar

as bibliotecas que são compartilhadas durante a compilação e execução do sistema no computador onde o mesmo foi implementado. Considerando que este sistema foi desenvolvido para ser executado em um sistema operacional Linux, os arquivos a serem copiados (bibliotecas compartilhadas) estão localizados no diretório de instalação do OpenCV (por padrão, /USR/LOCAL/OPENCV/ e /USR/LOCAL/OPENCV2/). O nome destes arquivos inicia com "lib" e termina com "so" (LUBUNTU, 2013; OPENCV,2013).

Estes arquivos devem ser copiados para um diretório do hardware onde o sistema deve ser executado sem OpenCV instalado. Neste trabalho, foi criado o diretório /USR/LOCAL/OPENCV/, para o qual foram copiados estes arquivos. Na sequência, é necessário indicar onde estes arquivos estão localizados. Para isso, deve ser criado um arquivo denominado OPENCV.CONF dentro do diretório /ETC/LD.SO.CONF.D/. Neste arquivo, deve ser informado apenas o diretório onde estão localizados os arquivos (/USR/LOCAL/OPENCV/) (LUBUNTU, 2013; OPENCV,2013).

Após isso, é necessário executar o comando "SUDO LDCONFIG –v" para que as configurações sejam carregadas pelo sistema operacional. O comando LDCONFIG realiza a atualização das bibliotecas compartilhadas pelo sistema e adiciona a informação na memória cache para localizar os arquivos que devem ser carregados e vinculados dinamicamente, no arquivo LD.SO.CACHE. Esse comando deve ser executado, como root (ou usuário com privilégios de administrador). O parâmetro "-v" é usado para procurar por bibliotecas compartilhadas e links em todos os diretórios e mostra-los na tela (LUBUNTU, 2013; OPENCV,2013).

Após isso, basta compilar o sistema e gerar um executável, utilizando o compilador G++, através do seguinte comando (considerando que o nome do programa seja detectaFTP.cpp):

```
g++ -ggdb `pkg-config --cflags opencv` detectaFTP.cpp -o detectaFTP `pkg-config --libs opencv`
```

Para executar o programa, é necessário informar um parâmetro adicional, sendo:

- 1 = Dia ensolarado.
- 2 = Dia nublado/chuvoso.
- 3 = Noite.

A execução do sistema em um ambiente noturno, por exemplo, deve ser feito da seguinte forma, considerando que o nome do programa é “detectaFTP”.

./detectaFTP 3

Para tornar o sistema mais eficiente, foram observadas as variações nas imagens de um mesmo cenário em diferentes condições climáticas (dia ensolarado, nublado, chuvoso, noite). Verificou-se que um modelo genérico para atuar nessas condições não se mostrou muito eficiente, visto a diferença nas características de luminosidade e cor observadas.

Para isso, pensou-se inicialmente em desenvolver um modelo para tratar as imagens de acordo com o turno (noite e dia) baseado no relógio do sistema. Essa abordagem foi descartada devido ao fato que seria inviável detectar se o dia está ensolarado, nublado ou chuvoso. Desta forma, optou-se em criar parâmetros para cada situação, os quais devem ser acionados pelo usuário (através de um botão, por exemplo).

Baseado em experimentos, verificou-se que as principais variáveis que interferem nas imagens obtidas nas três diferentes situações climáticas citadas anteriormente, são os valores de R, G e B, da imagem RGB gerada. Desta forma, o sistema foi implementado para considerar a cor dos pixels analisados como branco (cor da FTP, objeto de interesse) se estes tiverem os seguintes valores:

- Em ambiente de dia ensolarado: R >= 190, G >= 190, B >=190;
- Em ambiente de dia nublado/chuvoso: R >= 140, G >= 140, B >=140;
- Em ambiente noturno: R >= 80, G >= 80, B >=20;

Para o protótipo, que foi desenvolvido para processar imagens gravadas no disco e não captura-las a partir da câmera, deve ser informada ainda a imagem que se deseja processar. Por exemplo, para processar uma imagem de um ambiente noturno, denominada IMAGEM.JPG, localizada no mesmo diretório onde o executável do sistema está localizado, deve se proceder da seguinte forma, considerando que o nome do programa é “detectaFTP”.

./detectaFTP 3 IMAGEM.JPG

5 EXPERIMENTOS REALIZADOS

Para testar o sistema desenvolvido, foram realizados experimentos em diversas situações, conforme descrito na metodologia. O sistema conseguiu fazer a identificação correta das faixas na maioria das situações. Neste capítulo, é explicado como foram realizados estes testes e são apresentados os resultados obtidos.

5.1 Metodologia para testes

Os testes foram realizados seguindo critérios, em diferentes cenários e em diferentes condições climáticas. A seguir são descritos estes cenários:

A) Ambiente diurno, ensolarado onde a FTP aparece bem definida;

B) Ambiente diurno, ensolarado onde a FTP não aparece bem definida, com ruído ou sombra;

C) Ambiente diurno, ensolarado onde a FTP aparece parcialmente obstruída por um veículo;

D) Ambiente diurno, ensolarado, onde existem linhas brancas paralelas na pista, mas a FTP não aparece;

E) Ambiente diurno, nublado onde a FTP aparece bem definida;

F) Ambiente diurno, nublado onde a FTP não aparece bem definida, com ruído ou sombra;

G) Ambiente diurno, nublado onde a FTP aparece parcialmente obstruída por um veículo;

H) Ambiente noturno, iluminado, onde a FTP aparece bem definida;

I) Ambiente noturno, iluminado onde a FTP não aparece bem definida, com ruído ou sombra;

J) Ambiente noturno, iluminado onde a FTP aparece parcialmente obstruída por um veículo;

Na Figura 26, a seguir, são mostrados diferentes cenários testados. Na imagem A) pode ser visto um cenário onde a FTP aparece bem definida; em B) a FTP está com a tinta bastante danificada e a pista é irregular; em C) aparece um veículo sobre a FTP; e em E) aparecem linhas brancas paralelas, que são delimitadores de pista, não FTPs.

Figura 26: Diversos cenários testados

Fonte: O Autor

Além disso, os testes foram realizados coletando as imagens em diferentes alturas:

- 0,45 metros (altura da câmera posicionada na “cabeça” do robô DARwIn-OP).
- 1,2 metros (altura intermediaria, equivalente a uma criança).
- 1,7 metros (altura aproximada dos olhos de uma pessoa adulta).

Inicialmente os testes foram realizados no protótipo, com interface gráfica, para identificar como as diferentes variáveis envolvidas no processo interferem no reconhecimento da FTP. Após a aprovação do protótipo desenvolvido, o sistema foi adaptado para ser embarcado no robô humanoide DARwIn-OP. Após, os testes foram realizados novamente com imagens de um ambiente real de trânsito, obtidas a partir da câmera do robô. Foi avaliado o desempenho (tempo de processamento) e a corretude do sistema quanto aos resultados apresentados. O sistema captura apenas uma imagem (frame) e analisa a mesma, quando acionado.

5.2 Resultados

Nesta seção são explicados os resultados dos experimentos realizados com o sistema para a validação e testes do mesmo. Através do protótipo inicial desenvolvido é possível observar as principais funcionalidades e componentes do sistema, através do qual se pode também alterar o valor de algumas variáveis envolvidas, a fim de observar as diferenças nos resultados.

Os resultados obtidos mostram-se bastante promissores e comprovam a corretude e eficiência do sistema desenvolvido. Foram realizados testes com 39 imagens, contemplando 3 alturas diferentes em 3 diferentes cenários, conforme explicado na seção 5.1. Observou-se que o sistema reconheceu a FTP em 36, o que equivale a 92,3% das imagens testadas. Não foi observada a detecção indevida da FTP, ou seja, em imagens onde a mesma não aparece.

A Figura 27, a seguir, mostra os resultados do reconhecimento correto da FTP. As imagens são da interface gráfica do protótipo em execução, a fim de ilustrar os processamentos realizados e como os resultados foram gerados. Na imagem A), à direita, podem ser observados os retângulos definidos (em azul) e suas respectivas áreas. Na imagem B), à esquerda, podem ainda ser observados os traços formados para a definição dos retângulos, em azul claro na área interna aos retângulos ilustrados em azul. Os resultados gerados pelo protótipo são os mesmos dos gerados pelo sistema final embarcado no humanoide, sendo que ambos os sistemas diferenciam apenas pela interface gráfica, presente apenas no protótipo. A única diferença que pode ser observada em termos de resultado, é o tempo de processamento, maior no protótipo devido a interface gráfica e as ilustrações feitas na imagem processada, quando ambos são executados no mesmo *hardware*.

Figura 27: Reconhecimento correto da FTP

Fonte: O Autor

Na Figura 28, a seguir, é ilustrada uma imagem onde a primeira linha da FTP aparece parcialmente, o que pode ocorrer variando a altura da câmera e/ou a proximidade desta em relação a FTP. Conforme explicado no Capítulo 4, a área da linha da FTP seguinte deve ser menor em relação à sua anterior, o que não ocorre nesta situação.

A área calculada para o primeiro retângulo é igual a 23452, menor que a área calculada para o segundo retângulo (42228). Em função disso, o sistema analisa os demais retângulos formados e verifica se estes podem representar as linhas da FTP. A área calculada para o terceiro retângulo é 17360, o que representa 41,11% do retângulo imediatamente anterior, que tem área igual a 42228. Para o quarto retângulo definido foi calculada uma área de 9400, o que equivale a 54,14% em relação ao seu anterior. Desta forma, é desconsiderado o primeiro retângulo formado e os três seguintes identificados como linhas da FTP, visto que a regra do tamanho das áreas é satisfeita nestes. Assim, o sistema consegue contornar esse tipo de problema, ligado principalmente a altura da câmera em relação a pista, e fazer o reconhecimento correto da FTP nestas condições.

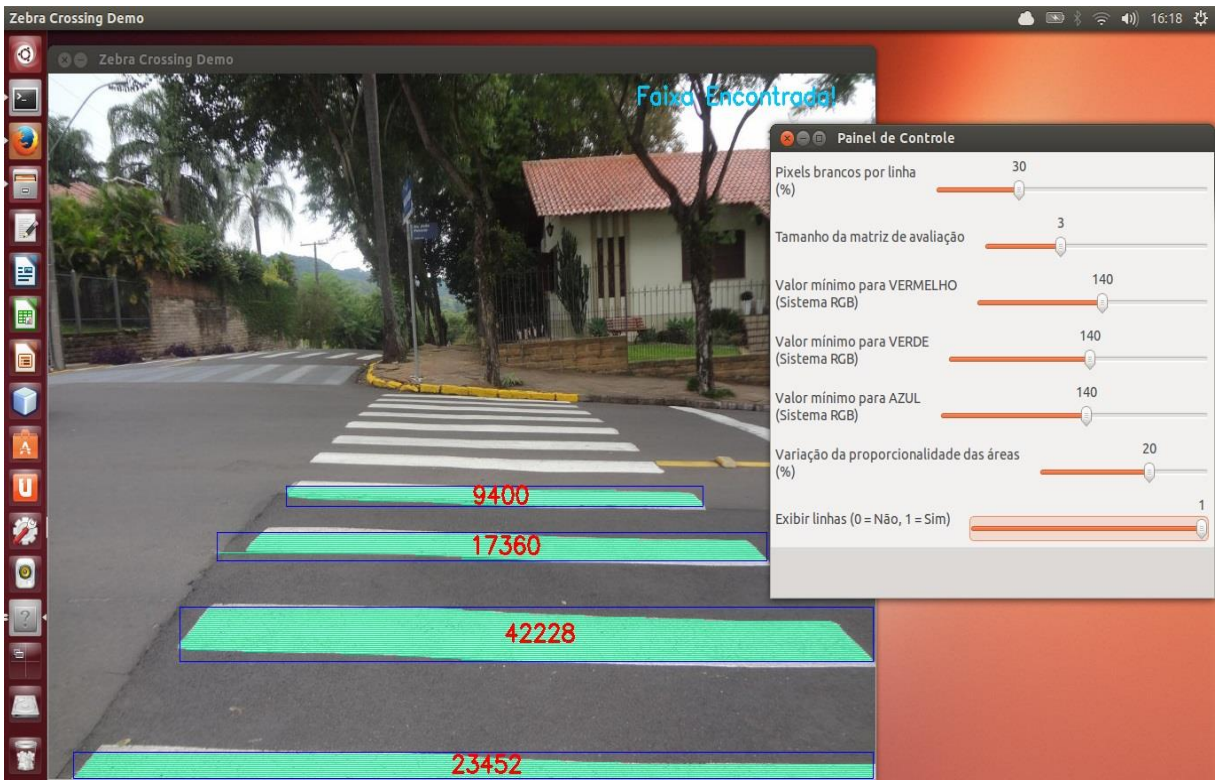


Figura 28: FTP encontrada com a primeira linha parcial

Fonte: O Autor

A importância de calibrar as cores e realizar corretamente o processo de quantização, pode ser observado no resultado de processamento da Figura 29. Na imagem A) pode ser observado o resultado de uma quantização inadequada para os pixels da cor branca em ambiente diurno ensolarado. Uma grande parte da pista foi identificada como branca, fazendo como que o algoritmo definisse o segundo retângulo indevidamente. Isso fez com que o algoritmo não conseguisse identificar a FTP. Na imagem B) foram usadas as definições de cor branca adequadas, fazendo com que o resultado do algoritmo seja correto, ou seja, identificando a FTP.

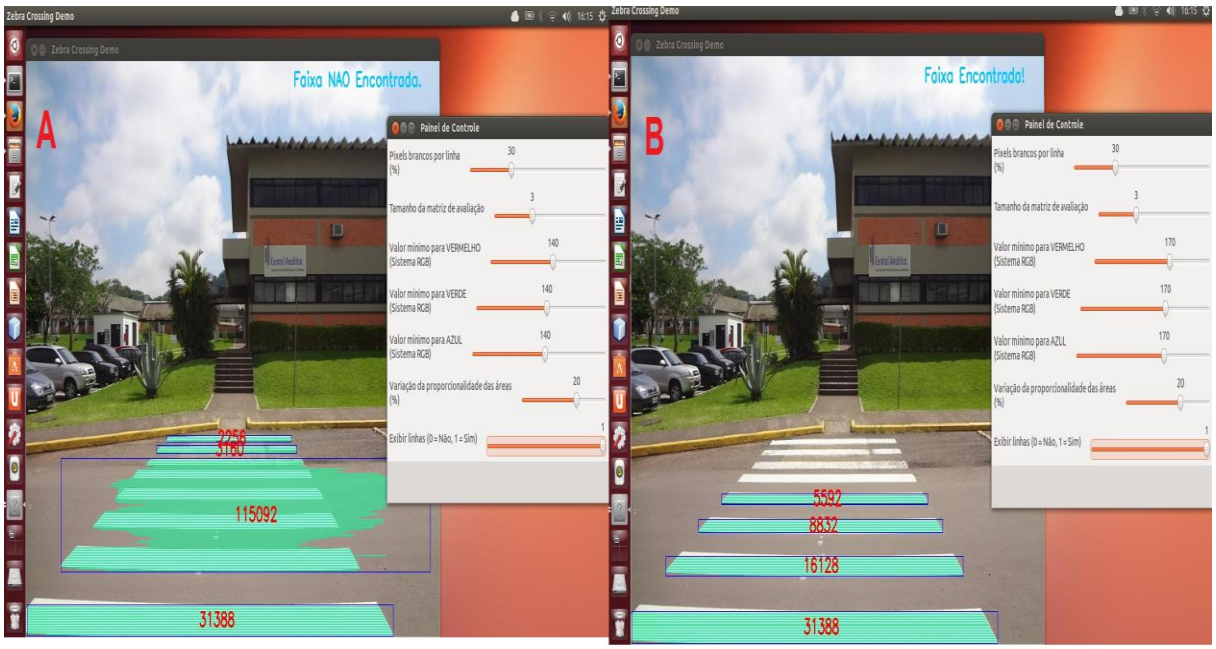


Figura 29: Diferentes valores para pixels brancos

Fonte: O Autor

Outro fator que pode interferir nos resultados é a altura a partir da qual a imagem é obtida (altura da câmera em relação à pista). Na Figura 30, a seguir, é ilustrado o resultado do processamento sobre o mesmo cenário, obtendo a imagem com a câmera posicionada a A) 1,70m; B) 1,20 m; e C) 0,45m de altura em relação a pista. As imagens ilustradas são de ambiente diurno, ensolarado, porém, os resultados são os mesmos em qualquer uma das três condições climáticas testadas.

Figura 30: Detecção da FTP posicionando a câmera em diferentes alturas

Fonte: O Autor

Em imagens noturnas existe uma diferença entre os valores de RGB nas cores dos pixels que devem ser reconhecidos como brancos. Isso se deve ao fato da diferença de luminosidade do ambiente, em relação à luz do dia. Com as definições feitas, tratando as imagens noturnas de forma diferente, foram obtidos resultados bastante promissores. A Figura 31, mostra como o sistema reconheceu a FTP sobre imagens coletadas em uma via à noite. Pode ser observado nas imagens A) e B) as diferentes alturas da câmera em relação ao piso, sendo 0,45m e 1,70m, respectivamente. Nas imagens C) e D) são mostradas diferentes condições de luminosidade.

É possível observar ainda, principalmente na imagem D) da Figura 31, que o sistema identifica as linhas da FTP além da parte pintada, considerando parte do asfalto não pintado como integrante das linhas. Isso se deve ao fato das cores se confundirem devido a intensidade da luminosidade nesta cena. Até o momento, não foi possível contornar este problema devido à similaridade de características dos pixels na região em volta das linhas da FTP neste tipo de cenário. De qualquer forma, esse problema não afetou os resultados.

Figura 31: Resultados em imagens noturnas

Fonte: O Autor

Nos testes em imagens onde aparecem veículos sobre a FTP, a mesma também foi reconhecida pelo sistema. A presença de veículos na cena também pode contribuir como um indicativo ao sistema, de forma que a travessia do deficiente visual tenha um maior grau de segurança, por exemplo. Vale destacar que o sistema apenas consegue identificar a FTP com pessoas e/ou veículos sobre ela, se estes não ocuparem as 4 primeiras linhas da FTP. A Figura 32, a seguir, ilustra o reconhecimento da FTP com um veículo sobre a mesma.

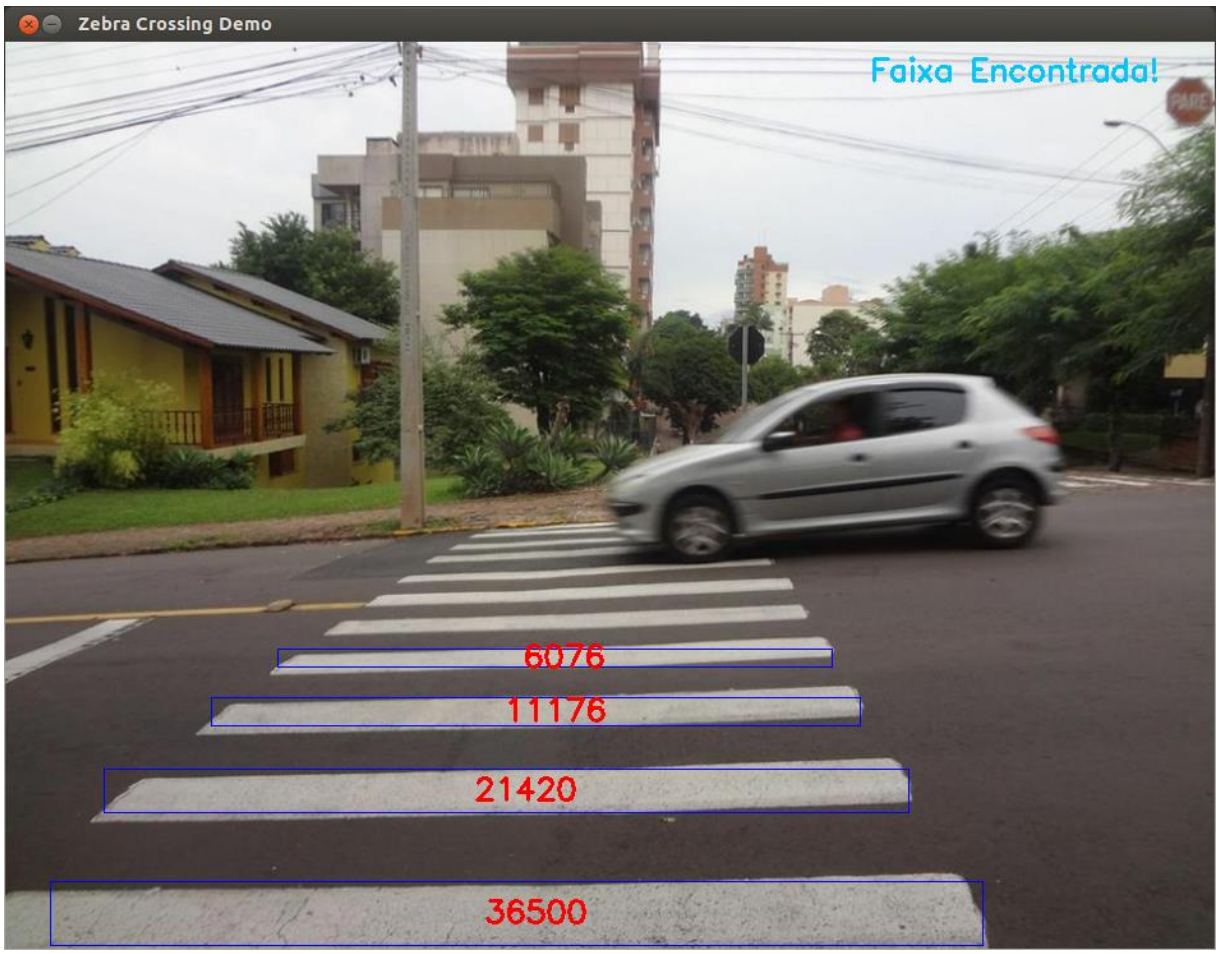


Figura 32: FTP encontrada com veículo sobre ela

Fonte: O Autor

As imagens nas quais a FTP não foi identificada, são de uma via construída com paralelepípedos e onde a tinta está muito danificada e apagada. Foram feitos testes com o sistema ampliando o tamanho da matriz de pixels avaliados para formar as linhas dos retângulos, de forma que um número maior de pixels não brancos pudesse ser tolerado. Entretanto, essa abordagem pode implicar em reconhecimentos falsos, ou seja, identificar FTP onde não existe a sinalização.

Foram feitos ainda, alguns testes utilizando operações morfológicas, como dilatação. Entretanto, os resultados não são satisfatórios para este tipo de cenário. A Figura 33, a seguir, mostra uma imagem em que a FTP não foi reconhecida. Em “A” pode ser vista a imagem original e em “B” a imagem processada, mostrando os retângulos formados. Devido às irregularidades da pista, não há uma continuidade de pixels brancos nas linhas da FTP dentro dos limites estabelecidos no algoritmo, conforme explicado na seção 4.1. Na primeira linha, foram identificados dois retângulos, na segunda um retângulo muito pequeno e na terceira um

retângulo regular. Devido à relação de proporcionalidade entre as áreas dos retângulos formados, também explicada na seção 4.1, estes não foram identificados como faixas da FTP, sendo esta não reconhecida.

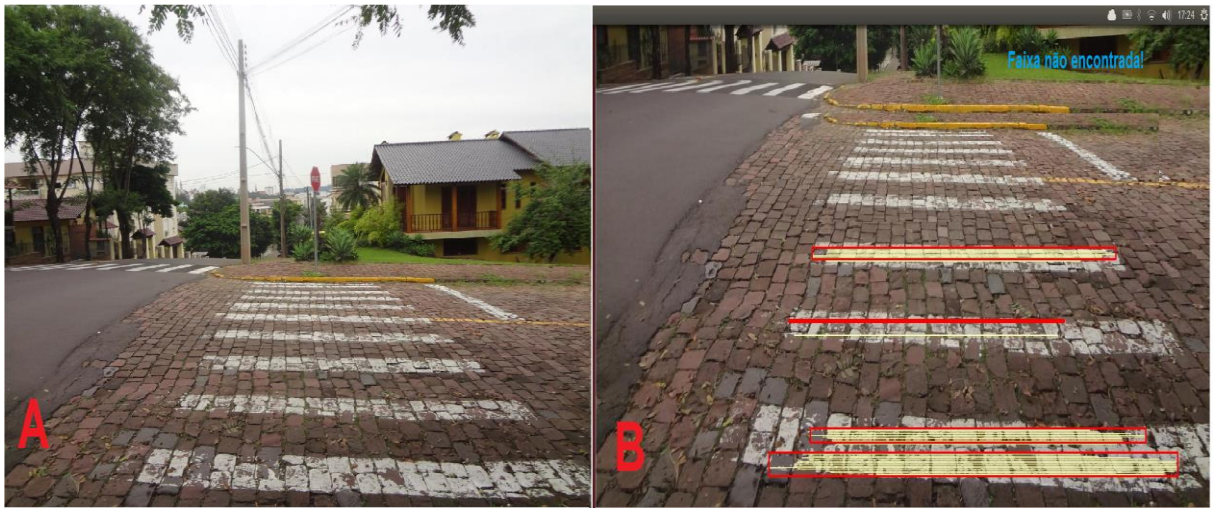


Figura 33: Imagem onde a FTP não foi identificada

Fonte: O Autor

Em comparação com outros trabalhos semelhantes da literatura, é possível ver que a eficiência do sistema proposto se aproximou dos mesmos. É importante destacar que foi utilizado um hardware com maiores limitações, sendo que nas abordagens de (YUZHEN, 2009) e (UDDIN, 2005) não é mencionado o cenário similar ao destacado anteriormente, onde a FTP não foi identificada. Outro fator importante a destacar, é que na abordagem de (UDDIN, 2005) não há distinção das situações climáticas com diferentes luminosidades, sendo analisadas apenas imagens diurnas de ambiente ensolarado (imagens noturnas e de dias nublados não são tratadas), ao passo que neste trabalho contempla estas situações. Além disso, ambas as abordagens citadas anteriormente realizam um processo de conversão das imagens coloridas para tons de cinza. A proposta deste trabalho é trabalhar com as cores originais das imagens coloridas.

A Tabela 1, a seguir, mostra esse comparativo, a partir do percentual de acerto obtido por cada sistema. Nesta, é possível observar também que o sistema apresentado não identificou indevidamente a FTP, o que também não ocorreu nos trabalhos avaliados no comparativo. Esta tabela mostra ainda, que o sistema encontrou a FTP corretamente em 92,3% das imagens analisadas. Este percentual é menor que o dos outros dois trabalhos, pois aborda situações mais diversas em cenários mais complexos, não mencionados nestes.

Tabela 1: Comparativo das taxas de acerto na identificação da FTP

%	Este trabalho	(YUZHEN, 2009)	(UDDIN, 2005)
EFC	92,3	95	94
NEFC	7,7	5	6
EFI	0	0	0
NEFI	100	100	100

Onde:

- EFC = Encontrou FTP corretamente (onde há FTP na imagem)
- NEFC = Não encontrou FTP corretamente (onde há FTP na imagem)
- EFI = Encontrou FTP incorretamente (onde não há FTP na imagem)
- NEFI = Não encontrou FTP onde não há FTP na imagem.

Para detectar uma FTP, o humanoide DARwIn-OP precisou, aproximadamente, 700 ms. Com isso, pode-se afirmar que o processamento se dá em um tempo bom para o propósito do sistema, visto os requisitos de tempo de resposta deste.

Estes tempos foram armazenados nos testes individuais efetuados nos diferentes cenários. Na Figura 34, a seguir, é apresentado um gráfico com o tempo de processamento de 20 cenários analisados, em milissegundos.

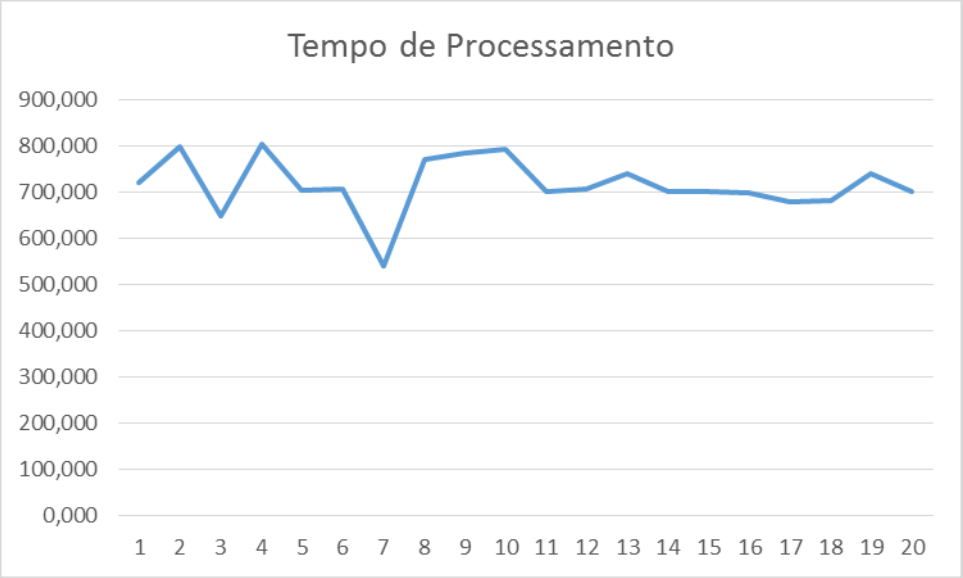


Figura 34: Gráfico do tempo de resposta em ms

Fonte: O Autor

Em comparação com os trabalhos de (SOUSA, 2008) e (BARGHAVA, 2007), o tempo de processamento foi maior. Entretanto, no trabalho de Sousa são utilizadas imagens com resolução menor, o que é uma justificativa para o menor tempo de resposta. Além disso, o hardware utilizado por estes dois pesquisadores é diferente do utilizado neste trabalho. A Tabela 2, a seguir, mostra este comparativo de tempo de processamento, em milissegundos.

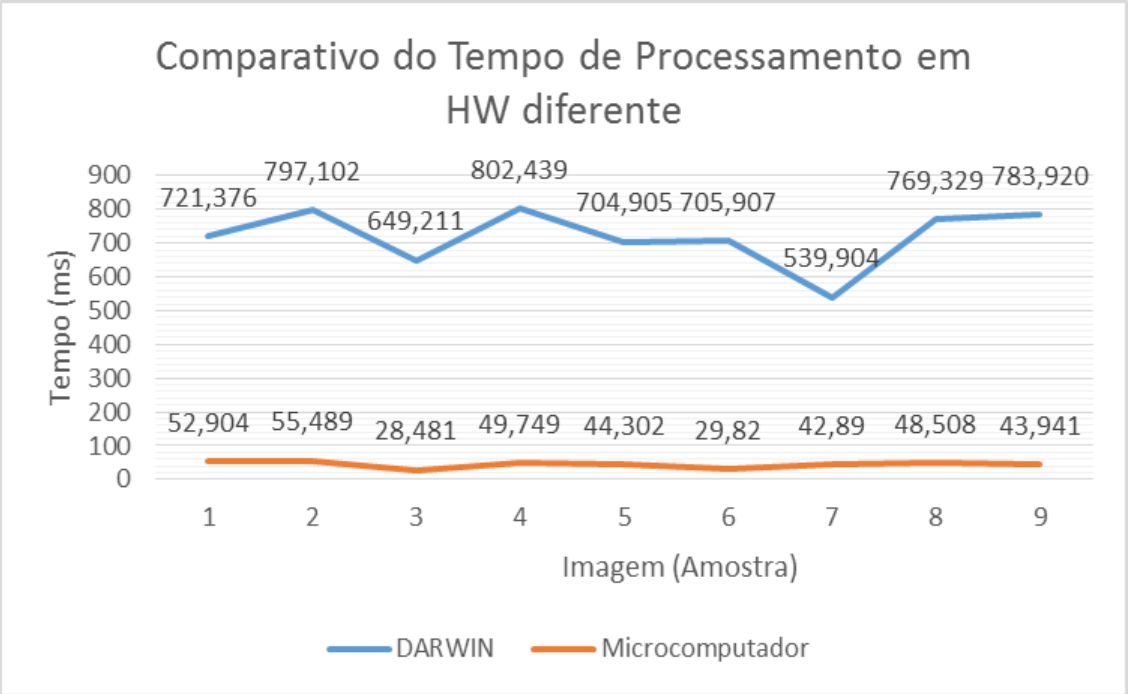
Tabela 2: Comparativo do tempo de processamento para identificar a FTP

	Este trabalho	**(SOUSA, 2008)**	**(BARGHAVA, 2007)**
Tempo médio	700 ms	362 ms	660 ms
Resolução das imagens	1152 x 864	480 x 640	Não informado

Foi realizado outro comparativo de desempenho do sistema. Utilizando amostras de imagens coletadas em diferentes cenários, foi realizado um comparativo do desempenho do sistema sendo executado no humanoide e em um microcomputador de alto desempenho. A tabela mostra os resultados deste comparativo. Na cor laranja, no eixo das ordenadas, são ilustrados os tempos de resposta do sistema sendo executado em um microcomputador de alto desempenho, onde os tempos variam entre 28,481 e 55,489 ms. No eixo das abcissas, em

azul, são mostrados os tempos do mesmo sistema sendo executado no humanoide DARwIn-OP, onde podem ser observados valores bem maiores, entre 539,904 e 802,439 ms.

Tabela 3: Comparativo do Tempo de Processamento em Hardware Diferente



6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentado um sistema para reconhecimento de faixas de pedestres utilizando um hardware específico, com limitações na capacidade de processamento e memória, com a implementação de funções da biblioteca OpenCV. Optou-se por utilizar esta biblioteca, pois esta possui um conjunto amplo de funções otimizadas para aplicar no processamento de imagens, bem como também, para verificar o desempenho de um sistema construído com base em funções OpenCV em um hardware limitado.

Inicialmente foram encontradas algumas dificuldades para embarcar o sistema desenvolvido no robô. Dentre estas, pode ser destacada a necessidade de atualizar o sistema operacional para uma versão mais atual devido à ausência e incompatibilidade de alguns recursos de software utilizados no sistema desenvolvido em relação ao sistema operacional original do robô. Apesar das dificuldades, o sistema foi embarcado com sucesso e mostrou-se eficiente neste hardware.

Na maioria das situações, o sistema reconheceu a FTP corretamente, não sendo observada a detecção da faixa de pedestres em locais onde a mesma não estava presente. Entretanto, em algumas situações o sistema não conseguiu identificar a FTP onde a mesma aparecia, tais como faixa muito apagada ou mesmo obstruída (por pedestres ou carros, por exemplo) nas três primeiras linhas da mesma.

Em testes iniciais, o sistema apresentou uma taxa de acerto na ordem de 92,3%, reconhecendo a FTP em diversas situações em um tempo que pode ser considerado aceitável para o propósito do sistema. Algumas melhorias e implementações se tornam necessárias, tais como, resolver problemas na detecção de FTP em vias irregulares construídas com paralelepípedos, visto que nestas o sistema não conseguiu detectar a FTP.

Outro fator crítico que deve ser considerado é o tempo de resposta, visto que se trata de um sistema no qual o tempo de resposta é crítico. Os resultados apresentados pelos testes mostram que o tempo de processamento está diretamente relacionado ao tipo de hardware utilizado, sendo que no humanoide o tempo foi bastante superior ao apresentado em testes feitos em um microcomputador de alto desempenho. Apesar disso, o tempo de resposta no humanoide pode ser considerado adequado, visto que ficou abaixo de 1 segundo em todas as situações testadas.

Uma das maiores contribuição deste trabalho é o processamento de imagens com as cores mais próximas das originais. Isto possibilitou o processamento de imagens noturnas, o

que não foi observado nos trabalhos semelhantes encontrados. Adicionalmente, as imagens tratadas também são mais complexas e em tamanho (resolução) original. Além disso, é importante destacar a eficiência deste modelo de sistema em hardware específico com limitações de processamento e armazenamento de dados.

Para a sequência deste trabalho, pode-se destacar a ampliação do sistema para reconhecer outros sinais de trânsito, tais como placas de sinalização e sinaleiras de trânsito (semáforos). Com isso, pode-se aliar o reconhecimento destes sinais ao da FTP, tornando o sistema mais completo. Pode-se ainda adaptar o sistema de forma a reconhecer os sinais com foco nos veículos, visto que o sistema desenvolvido possui foco no pedestre.

Outra abordagem interessante, é a integração com um sistema meteorológico, para calibração dos parâmetros de RGB, de forma automática. Também poderia ser aliado a essa integração, o uso do horário do sistema para definir turnos (dia, noite).

REFERÊNCIAS

1. ABNT 2004, Norma Brasileira ABNT NBR 9050, Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos, segunda ed., 2004.

2. BARGHAVA, B., PELIN, A., DUAN, L.. *A Mobile-Cloud Pedestrian Crossing Guide for the Blind*, 2010.

3. BARGHAVA, B., PELIN, A., HEDAL, S.. *A Mobile-Cloud Collaborative Traffic Lights Detector for Blind Navigation*, 2007.

4. BRADSKI, G.; KAEBLER, A. *OpenCv: Computer Vision with the OpenCv Library. United States Of American: O'relly*, 2008.

5. BURGHOUTS, G. J.; GEUSEBROEK, J. M., *Performance evaluation of local colour invariants, Computer Vision and Image Understanding*, 2009.

6. CENSO 2010, Site com as informações do censo 2010, disponível em http://censo2010.ibge.gov.br, acesso em 26 de dezembro de 2013.

7. CHARETTE, R., NASCHASHIBI, F., "*Real Time Visual Traffic Lights Recognition Based on Spot Light Detection and Adaptive Traffic Lights Templates," World Congress and Exhibition on Intelligent Transport Systems and Services*, 2009.

8. CONCI, A.; AZEVEDO, E.; LETA, F. R. Computação Gráfica: Teoria e Prática. v.2, Rio de Janeiro: Campus/Elsevier, 2008.

9. CONTRAN (Conselho Nacional de Trânsito (Brasil)). Sinalização horizontal / Contran-Denatran. 1ª edição – Brasília: Contran, 2007.

10. CTB, Código de Trânsito Brasileiro e Legislação Complementar em Vigor, Ministério das Cidades, Conselho Nacional de Trânsito, Departamento Nacional de Trânsito, 2008.

11. DARNELL, P; MARGOLIS, P. C, a software engineering approach. Springer-Verlag New York Inc., 1991

12. DARWIN-OP, Manual DARwIn-OP, disponível em http://DARwIn-OP.springnote.com/pages/6861909, acesso em 10 de setembro de 2012.

13. DEITEL, H.M., DEITEL, P. J., C++: Como Programar, 3. ed. Porto Alegre: Bookman, 2001.

14. DONY, R.D., WESOLKOWSKY, S. *Edge Detection Images Using RGB Vector Angles, IEEE Canadian Conference on Electrical and Computer Engineering Shaw Conference Center*, 1999.

15. ESS, A., LEIBE, B., SCHINDLER, K., GOOL, L. van, "*Moving Obstacle Detection in Highly Dynamic Scenes," IEEE International Conference on Robotics and Automation*, 2009.

16. FAUGERAS, O., *Three-Dimensional Computer Vision: A Geometric Viewpoint*, *London*: MIT Press, 1993.

17. FAWCETT, T., *An introduction to ROC analysis*, *Pattern Recognition Letters*, 2006.

18. FLUSSER, J.; SUK, T., *Rotation moment invariants for recognition of symmetric objects, IEEE Transactions on Image Processing*, 2006.

19. FOTOSIZER, Website, disponível em www.fotosizer.com, acesso em 12 de Outubro de 2013.

20. GALLO, O., MANDUCHI, R., RAFII, A., "*Robust Curb and Ramp Detection for Safe Parking using the Canesta TOF camera," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008.

21. GALVÃO FILHO, T. A. ; DAMASCENO, L.L. Programa InfoEsp: *Premio Reina Sofia 2007 de Rehabilitación y de Integración. In: Boletín del Real Patronato Sobre Discapacidad, Ministerio de Educación, Política Social y Deporte, Madri*, Espanha. n. 63, 2008.

22. GERKEY, B.P., THRUN, S., and GORDON, G.. *Visibility-based pursuit-evasion with limited field of view. International Journal on Robotics Research*, 2006.

23. GONZALEZ, R. C.; WOODS, R.E. *Digital Image Processing, Prentice Hall*, 2002.

24. GONZALEZ, R.C.; WOODS, R.E. Processamento de imagens digitais. 1.ed. São Paulo: Editora Edgar Blücher Ltda, 2000.

25. GONZALEZ, R.C.; WOODS, R.E. Processamento digital de imagens. 3.ed. São Paulo: Pearson Prentice Hall, 2010.

26. GRAVILA, D. M., FRANKE, U., WOHLER, C., and GORZIG, S., *Real Time Vision for Intelligent Vehicles, IEEE Instrumentation and Measurement Magazine*,Vol. 4, 2001.

27. GUO-SHENG,M., JIAO,Y., XIAO-GUANG, X., *Predict on of the Position of Pedestrian Crossing Road Section Based on Kalman Predictor, International Conference on Measuring Technology and Mechatronics Automation, 2009*

28. HARLEY, R., ZISSERMAN, *A. Multiple View Geometry in Computer Vision. Cambridge University, Press, 2000.*

29. HENNING, P. A., *"Taschenbuch Multimedia"*, 4a ed: Hanser, 2007.

30. INTEL *Company* website, disponível em: http://www.intel.com, acesso em 25 de Novembro de 2012.

31. KANNALA, J., *Models and Methods for Geometric Computer Vision*, University *of OULU*, Finlândia, 2010.

32. KEVICZKY, T., FALCONE P., BORRELLI, F., ASGARI, J., HROVAT, D., *Predictive Control Approach to Autonomous Vehicle Steering, IEEE transactions,* 2007.

33. KIM, Y.K., KIM, K.W., YANG, X., *Real Time Traffic Light Recognition System for Color Vision Deficiencies, IEEE International Conference on Mechatronics and Automation,* 2011.

34. KLAWONN, F. *Grundkurs Computergrafik mit Java 3D*, 1.ed: Vieweg, 2005.

35. KOCHE, J. C. Fundamentos da metodologia científica: Teoria da ciência e prática da pesquisa. 19. ed. Petrópolis: Vozes, 2001.

36. KOSCHAN, A.; ABIDI, M. *Digital Color Image Processing,* 1.ed: A John Wiley & Songs , inc., Publication, 2008.

37. LU, K.; WANG, C.; CHEN, S. *Traffic Light Recognition. Journal of the Chinese Institute of Engineers*, Vol. 31, No. 6, 2008.

38. LUBUNTU, Site do fabricante do sistema operacional Lubuntu, disponível em http://www.lubuntu.net, acesso em 27 de Novembro de 2013.

39. NETBEANS Website. Disponível em http://netbeans.org/. Acesso em 05 de Dezembro de 2012.

40. NOGUEIRA, A.A.M., TRANJAN, C.G..Perspectiva de Observação: Técnicas, exemplos, atalhos e exercícios. Rio de Janeiro, 2009.

41. OLIVEIRA, L. M. B. Secretaria de Direitos Humanos da Presidência da República (SDH/PR), Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência (SNPD), Coordenação-Geral do Sistema de Informações sobre a Pessoa com Deficiência: Cartilha do Censo 2010 – Pessoas com Deficiência, Brasília: SDH-PR/SNPD, 2012.

42. OpenCV manual online. Disponível em: http://docs.opencv.org/. Acesso em 25 de Outubro de 2013.

43. OPENCV web site. Disponível em: http://opencv.willowgarage.com/wiki/. Acesso em: 02 de Setembro de 2012.

44. PEDRINI, H.; SCHWARTZ, WILIAM, R. Análise de imagens digitais: princípios, algoritmos e aplicações, 2007.

45. PRATT, W.K., *Digital Image Processing, 4.ed: A John Wiley & Songs , inc., Publication, 2007.*

46. PUGH, K., Programando em linguagem C, São Paulo: McGraw-Hill do Brasil, 1990.

47. ROBOTIS Website. Disponível em http://www.robotis.com. Acesso em 25 de Novembro de 2012.

48. SICHELSCHMIDT, S., HASELHOFF A., KUMMERT A., Pedestrian Crossing Detecting as a part of an Urban Pedestrian Safety System, 2010 IEEE Intelligent Vehicles Symposium University of California, San Diego, CA, USA, 2010.

49. SILVA, E.L.da, MENEZES, E.M. Metodologia da Pesquisa e Elaboração da Dissertação, 4.ed, Universidade Federal de Santa Catarina, 2005.

50. SOBREARTE, Website com informações sobre desenho em perspectiva, http://www.sobrearte.com.br/desenho/perspectiva/elementos_da_perspectiva.php, acesso em 14 de Janeiro de 2014.

51. SOLOMON, C., BRECKON, *T. Fundamentals of Digital Image Processing – A Practical Approach with Examples in Matlab,* Wiley-Blackwell, 2011.

52. SOUSA, K., MARENGONI, M. Uso de Visão Computacional em Dispositivos Móveis para o Reconhecimento de Faixa de Pedestres, 2007.

53. THRUN, S., BURGARD, W., FOX, D., *Probabilistic Robotics, MIT Press, Cambridge, MA, 2005.*

54. THRUN, S.. et al. *Stanley: The Robot that Won the DARPA Grand Challenge, Journal of Field Robotics, E.U.A., Wiley Periodicals, 2006.*

55. VIRTUALBOX, Site do fabricante do sistema Oracle VM Virtualbox, disponível em https://www.virtualbox.org, acesso em 02 de Dezembro de 2013.

56. WANG, L.; JU, H. *A Robust Blob Detection and Delineation Method, IEEE Computer Society*, 2008.

57. XU,G., ZHANG,Z. *Epipolar Geometry in Stereo, Motion and Object Recognition, A Unified Approach. Kluwer Academic Publishers, 1996*

58. YUZHEN, C., LUSHI, C., SHUO, J., *An Image Based Detection of Pedestrian Crossing, IEEE, CHINA, 2009.*