
Projeto de um sistema embarcado de predição de
colisão a pedestres baseado em computação
reconfigurável

Leandro Andrade Martinez

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Projeto de um sistema embarcado de predição de colisão a pedestres baseado em computação reconfigurável

Leandro Andrade Martinez

***Orientador:* Prof. Dr. Eduardo Marques**

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

USP – São Carlos
Fevereiro de 2012

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

M385p Martinez, Leandro Andrade
 Projeto de um sistema embarcado de predição de
 colisão a pedestres baseado em computação
 reconfigurável / Leandro Andrade Martinez;
 orientador Eduardo Marques -- São Carlos, 2011.
 57 p.

 Dissertação (Mestrado - Programa de Pós-Graduação em
 Ciências de Computação e Matemática Computacional) --
 Instituto de Ciências Matemáticas e de Computação,
 Universidade de São Paulo, 2011.

 1. CIÊNCIA DA COMPUTAÇÃO. 2. COMPUTAÇÃO
 RECONFIGURÁVEL. 3. SISTEMAS EMBARCADOS. 4. CIRCUITOS
 FPGA. I. Marques, Eduardo, orient. II. Título.

AGRADECIMENTOS

Os maiores e mais sinceros agradecimentos ao meu orientador, prof. Eduardo Marques. Sua orientação foi capaz de me fazer trilhar por um crescimento profissional que julgava impossível em tão pouco tempo. Orientador é uma palavra ideal para defini-lo: é sob sua tutela que guio meus passos. Muito obrigado!

Agradeço imensamente à Nádia, minha namorada, por me fazer enxergar meus dons e minhas capacidades, dando-me forças para querer ir sempre além.

A minha mãe e meu pai (in memoriam), que me ensinaram a ter humildade, persistência e confiança na busca pelo conhecimento;

Aos colegas de classe que com muita dedicação ajudaram nos trabalhos em grupo.

À Universidade de São Paulo, pela oportunidade de realização do curso de mestrado;

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico, pela concessão da bolsa de mestrado e pelo apoio financeiro para a realização desta pesquisa;

Ao Instituto de Ciências Matemáticas e de Computação, por colocar à disposição a área experimental e o laboratório;

“Se um homem tem um talento e não tem capacidade de usá-lo, ele fracassou. Se ele tem um talento e usa somente a metade deste, ele fracassou parcialmente. Se ele tem um talento e de certa forma aprende a usá-lo em sua totalidade, ele triunfou gloriosamente e obteve uma satisfação e um triunfo que poucos homens conhecerão.”

Thomas Wolfe

Resumo

Este trabalho apresenta a construção de um sistema embarcado para detectar pedestres, utilizando computação reconfigurável com captura de imagens através de uma única câmera acoplada a um veículo que trafega em ambiente urbano. A principal motivação é a necessidade de reduzir o número vítimas causadas por acidentes de trânsito envolvendo pedestres. Uma das causas está relacionada com a velocidade de resposta do cérebro humano para reconhecer situações de perigo e tomar decisões. Como resultando, há um interesse mundial de cientistas para elaborar soluções economicamente viáveis que venham a contribuir com inovações tecnológicas direcionadas a auxiliar motoristas na condução de veículos. A implementação em hardware deste sistema foi desenvolvida em FPGA e dividida em blocos interconectados. Primeiramente, no pré-tratamento do vídeo, foi construído um bloco para conversão de dados da câmera para escala de cinza, em seguida, um bloco simplificado para a estabilização vertical dinâmica de vídeo. Para a detecção foram construídos dois blocos, um para detecção binária de movimento e um bloco de detecção BLOB. Para fazer a classificação, foi construído um bloco para identificação do tamanho do objeto em movimento e fazendo a seleção pela proporcionalidade. Os testes em ambiente real deste sistema demonstraram ótimos resultados para uma velocidade máxima de 30 km/h.

Abstract

This work proposes an embedded system to detect pedestrians using reconfigurable computing making the image acquisition through a mono-camera attached to a vehicle in an urban environment. This work is motivated by the need to reduce the number of traffic accidents, even with government support, each year hundreds of people become victims thus bringing great damage to the economy. As a result, there is also a global concern of scientists to promote economically viable solutions that will contribute to reducing these accidents. A significant issue is related to the speed of response of the human brain to recognize and or to make decisions in situations of danger. This feature generates a demand for technological solutions aimed at helping people to drive vehicles in several respects. The system hardware was developed in FPGA and divided into interconnected blocks. First, for the pretreatment of the video, was built a block for data conversion from the camera to grayscale, then a simplified block for vertical stabilization dynamic video. To detection, two blocks were built, one for binary motion detection and one for a BLOB detection. To classify, was built one block to identify the size of the object in motion by the proportionality and making the selection. The tests in real environment of this system showed great results for a maximum speed of 30 km / h.

Sumário

1. Introdução	1
1.1. Justificativa	1
1.2. Principais Desafios	2
1.3. Pesquisa e Desenvolvimento.....	5
1.4. Organização do Trabalho.....	8
2. Fundamentação Teórica	10
2.1. Detecção de Pedestres	10
2.2. Tecnologias de Sensores.....	12
2.3. Computação Reconfigurável.....	16
2.4. Síntese das técnicas envolvidas.....	18
2.5. Restrições em relação ao tempo	21
3. Considerações de Projeto	24
3.1. Especificação da Primeira Versão	24
3.2. Especificação da Segunda Versão	38
3.3. Especificação da Terceira Versão	40
3.4. Contextualização do Sistema	40
3.5. Ambientes de Desenvolvimento	42
3.6. Base de Dados de Pedestres	46
4. Resultados Obtidos com a Versão 2	48
4.1. Resultados da Detecção	48
4.2. Estabilização da Imagem	49
4.3. Detecção de Movimento	50
4.4. Extração de Características	51
4.5. Classificação.....	52
5. Conclusão	54
5.1. Contribuições deste Trabalho	55
5.2. Trabalhos Futuros	56
Referência Bibliográfica.....	58
Apêndice A	65
A. Veículos Inteligentes	65
A1. Sistemas Avançados para Auxiliar Motoristas	65
A2. Sistemas Embarcados Automotivos	69

A3. Evolução Histórica.....	70
Apêndice B	73
B. Detalhamento da Implementação do Hardware da Versão 3.....	73
B1. Máquina de Estados do Bloco Estabilizador	73
B2. Diagrama de Hardware do Bloco Estabilizador	74
Apêndice C	75
C. Detalhamento de Módulo para Testes Com Câmera usando NIOS	75

Lista de Figuras

Figura 1 – Variabilidade do Ambiente (McCall, et al., 2006).....	4
Figura 2 - Exemplos de imagens com detecção complexa (McCall, J.C., et al 2006)	5
Figura 3 – Sensor e Placa Mobileye	6
Figura 4 – Interface Mobileye.....	6
Figura 5 – Projeto SENA (USP São Carlos).....	7
Figura 6 – Projeto Carina (LRM, 2011).....	8
Figura 7 - Radar e Scanner a Laser (projeto Autonomes Fahren)	15
Figura 8 - Visão Estereoscópica (projeto Autonomes Fahren)	15
Figura 9 - Exemplo de detecção com visão estéreo - sistema Protector	16
Figura 10 - Cadeia de processamento de imagens (adaptado de Techmer, 2007).....	18
Figura 11 - Etapas básicas de um sistema de visão computacional	19
Figura 12 - Visão Geral para a implementação (Jalalian, 2008 - modificado)	20
Figura 13 - Distância para parar (Alonso, 2005)	21
Figura 14 – Detalhamento da Versão 1.....	25
Figura 15 – Especificação Conversor para Escala de Cinza.....	26
Figura 16 - Espaço de cor RGB.....	27
Figura 17 – Demonstração do Vetor de Estabilização.....	28
Figura 18 – Comparação de Dois frames.....	29
Figura 19 - Vetores Representativos de Movimento (Jalalian, 2008).....	30
Figura 20 – Especificação do Sistema de Detecção.....	31
Figura 21 – Matriz de Pixels em Movimento	33
Figura 22 – Especificação do Bloco de Detecção de Movimento.....	33
Figura 23 – Especificação do Bloco de Remoção de Ruído.....	34
Figura 24 – Detecção de Múltiplos BLOBs na imagem	35

Figura 25 - Tamanho variado de pedestres devido a distância.....	36
Figura 26 – Imagem com Blobs com Junção e filtro por proporcionalidade.	37
Figura 27 – Exemplo da área de risco.....	38
Figura 28 – Detalhamento da Versão 2.....	39
Figura 29 – Detalhamento da Versão 3.....	40
Figura 30 – Placa de Desenvolvimento Altera DE2-70.....	43
Figura 31 - Integração de instrução personalizada a ULA do Nios II	44
Figura 32 – Diagrama de Blocos do Núcleo do Processador Nios II.....	45
Figura 33 – Disposição da Câmera para Captura.....	46
Figura 34 – Exemplo de Quadros Capturados.....	47
Figura 35 – Avaliação do Estabilizador	50
Figura 36 – Avaliador do Tempo para Detecção.....	51
Figura 37 – Avaliação dos Tempos para Detectar BLOB	51
Figura 38 – Avaliação do tempo do Classificador	52
Figura 39 – Avaliação do sistema de Classificação	53
Figura 40 - Identificação de obstáculos laterais (quickcashauto.com)	66
Figura 41 - Sistemas para manter-se centralizado (files.blogter.hu)	67
Figura 42 - Sistemas de auxílio a estacionamento (bosch.com)	68
Figura 43 - Navegação por GPS (gpscopy.com).....	69
Figura 44 – Diagrama do Bloco Estabilizador da Versão 3	73
Figura 45 – Bloco Estabilizador da Versão 3	74
Figura 46 – Integração do módulo de câmera com o NIOS II	75
Figura 47 – Detalhamento do Controlador SDRAM.....	76

Lista de Tabelas

Tabela 1 – Estado da Arte em Detecção de Pedestres (DOLLÁR, P. et al, 2011)	12
Tabela 2 - Comparativo entre CCD e CMOS	13
Tabela 3 – Comparativo entre as diferentes tecnologias.....	17
Tabela 4 – Resultados do Experimento.....	48
Tabela 5 – Resultado da Implementação da Versão 2	49
Tabela 6 – Projeção do Sistema.....	55
Tabela 7 – Saídas da Máquina de Estado do Estabilizador	74

1.Introdução

Propor um novo sistema de detecção de pedestres é um grande desafio, pois existem diversas propostas já apresentadas e algumas comercialmente disponíveis. Contudo, com a complexidade da construção desses sistemas em conjunto com o crescente avanço das tecnologias para sistemas embarcados, existe ainda ampla oportunidade no desenvolvimento de novas abordagens melhorando a eficiência e eficácia das atuais soluções. Portanto, o objetivo deste projeto é construir um sistema de detecção de pedestres em um dispositivo de Computação Reconfigurável (FPGA) que atenda aos principais requisitos e que, posteriormente, possa ser usado de base para a continuação de estudos na área.

Para atingir este objetivo, a primeira versão do projeto faz a validação de um conjunto de algoritmos de processamento de imagem executados em um computador de propósito geral. A segunda versão compreende em embarcar a solução em um dispositivo reconfigurável, executando a aplicação em um processador softcore. Na terceira versão, para atender aos requisitos de desempenho em tempo real, o sistema desenvolvido em software é convertido totalmente em hardware.

1.1. Justificativa

Os dados de interesse nacional na área segurança são obtidos do RENAEST (Registro Nacional de Acidentes e Estatísticas de Trânsito), onde indicam que no Brasil, somente no ano de 2008, ocorreram aproximadamente 116 mil acidentes de trânsito com vítima, sendo quase 17 mil atropelamentos e assim, causando prejuízo às famílias e gerando uma significativa quantidade de despesas médicas e sociais ao País. No contexto Internacional, as estimativas indicam que 24% de todas as mortes no trânsito em todo mundo são consequências de atropelamento, justificando a necessidade de sistemas que visam minimizar esse fato.

Cabe ressaltar, esta estatística do governo: *“Acompanhando o crescimento urbano, o fenômeno trânsito passou a ser visto como elemento de preocupação na gestão urbana,*

principalmente no que se refere à melhoria da qualidade de vida nas cidades. O trânsito é responsável pela ocorrência de 300 mil acidentes por ano, que deixam a expressiva marca de 30 mil mortos e outros 350 mil feridos, representando custo anual de cerca de R\$ 10 bilhões.”
(extraído de <http://www.cidades.gov.br/denatran> - acessado em Abril de 2010)

Os sistemas avançados destinados ao apoio de motoristas (ADAS – Advanced Driver Assistance Systems) pertencem a um importante nicho na área automobilística, no qual é motivado por um grande público e com interesse industrial em melhoria de segurança e conforto. Aplicações para advertir quando o veículo está saindo das faixas de trânsito em estradas, piloto automático de velocidade, sensores para auxiliar estacionamento, câmeras para visão noturna e sistemas de navegação GPS são aplicações normalmente introduzidas em carros modernos. A busca por inovação é contínua e aplicações como detecção de pedestres ou sistemas para auxiliar na ultrapassagem, tem atraído pesquisadores por muitos anos (Techmer, 2007).

Os recentes progressos na área de computação reconfigurável, especificamente em FPGA (Field Programmable Gate Arrays), promoveram um ótimo aumento do desempenho de plataformas de computação baseadas nessa arquitetura, incluindo sistemas de processamento customizado de imagem e vídeo. Por outro lado, as exigências referentes ao tempo de resposta para aplicações de processamento de imagens são grandes, tornando-se um desafio criar soluções que possam equilibrar os requisitos de desempenho com um consumo de energia eficiente.

1.2. Principais Desafios

Os principais desafios para detecção de pedestres em veículos estão relacionados ao uso eficiente do recurso computacional com processadores de Computação Embarcada, considerando a implementação em núcleos de processamento com quantidade de memória limitada e acesso direto à memória principal (DMA – Direct Memory Access).

Outro importante desafio é o desenvolvimento de sistemas multiprocessados e com múltiplos núcleos, pois, com o uso de processamento paralelo é fornecido um ganho de desempenho considerável para processamento em tempo real de detecção de padrões e

identificação (Kirschian, 2008). Este aumento está associado diretamente à eficiência de processamento proporcionado pelos núcleos com a distribuição das tarefas de processamento da imagem e de reconhecimento.

Conforme proposto na justificativa deste trabalho, muitos benefícios podem surgir com a detecção de pedestres. Ainda assim, existem muitos desafios para serem resolvidos e aperfeiçoados para se chegar a uma solução satisfatória. O fato do reconhecimento de imagens ser feito em um veículo em movimento gera uma complexidade maior para a detecção (McCall, et al., 2006). Além do hardware de capacidade limitada, a detecção de pedestres em um cenário real torna-se um desafio devido a várias razões:

As roupas de pedestres podem variar em diversas cores e formatos, confundindo o reconhecimento. Existem também, inúmeras possíveis poses, onde os braços e pernas podem estar em diferentes configurações. Há também, diferentes ângulos de visão que a câmera pode encontrar os pedestres. Esse tipo de problema dificulta a detecção e o rastreamento de pedestres.

Um cenário comum onde há necessidade de detecção de pedestres é em meio ao centro urbano. Neste cenário, as condições do ambiente (Figura 2) podem também dificultar a visão por vários motivos:

Posição Solar: a orientação solar pode causar saturação na imagem capturada pela câmera, ou provocar reflexos especulares;

Sombras: árvores, prédios, pontes entre outros projetam sua sombra sobre os pedestres alterando sua intensidade e textura;

Oclusão: objetos podem obstruir parcialmente ou totalmente a visão da câmera;

Condições Climáticas: fenômenos da natureza (como neblina, chuva ou neve) podem degradar significativamente a qualidade das imagens adquiridas.

Alguns desses fatores são ilustrados em na Figura 1: No primeiro quadro a esquerda pode-se observar a presença de buracos na pista; no segundo quadro diferenças bruscas nas

cores do asfalto; na segunda linha problemas com as faixas de sinalização; e na última linha presença de sombras.



Figura 1 – Variabilidade do Ambiente (McCall, et al., 2006)

Pedestres vistos a distância podem ser facilmente confundidos com outros objetos de fundo, como por exemplo, árvores, placas de trânsito, postes e assim por diante. O movimento do pedestre é importante para sua detecção. Porém, em um ambiente com muitos pedestres em movimento a detecção deve ser tratada de forma diferente. Para isto, existem algoritmos que trabalham com detecção de multidão de pessoas.



Figura 2 - Exemplos de imagens com detecção complexa (McCall, J.C., et al 2006)

1.3. Pesquisa e Desenvolvimento

Atualmente, existem vários modelos de automóvel com algum tipo de sistema para evitar colisão: Acura RL, BMW 750Li, BMW (Série 3), Cadillac DTS, Ford Taurus, Honda Accord, Infiniti (FX e EX), Mercedes-Benz (Classe E), Mercedes-Benz (Classe S), Volvo (S60, V60, XC60, V70, XC70, S80), SsangYong Rexton, Toyota Prius (Série 3), Lexus ES. Tanto a indústria automobilística quanto entidades de pesquisa têm grande interesse na elaboração de dispositivos para detecção na área de visão computacional. Esta seção apresenta algumas dessas entidades e alguns dos produtos que são foco de pesquisa. Outros sistemas relacionados a veículos inteligentes, sistemas ADAS e sistemas embarcados automotivos são detalhados no Apêndice A (ao final deste trabalho).

1.3.1. Sistema Mobileye

A Mobileye® é uma empresa de tecnologia fundada em 1999 que investe no desenvolvimento de sistemas ADAS (Advanced Driver Assistance Systems). Seu objetivo é desenvolver e comercializar sistemas baseados em visão para ajudar motoristas a manter os passageiros em segurança e contribuir na diminuição de acidentes de trânsito em estradas. A empresa oferece também uma ampla gama de soluções de segurança. A Mobileye NV está sediada na Holanda, com um Centro de Pesquisa e Desenvolvimento (P&D) em Jerusalém, Israel, e escritórios de vendas e marketing em Los Angeles, Califórnia; Detroit, Michigan; Nicosia, e no Japão (MOBILEYE, 2011).



Figura 3 – Sensor e Placa Mobileye
(MOBILEYE, 2011)



Figura 4 – Interface Mobileye
(MOBILEYE, 2011)

A Figura 3 e a Figura 4 apresenta a segunda versão de um de seus produtos, o Mobileye EyeQ2™, que consiste em dois processadores de ponto flutuante em hiper-thread 32bit RISC (MIPS32® 34K núcleos), cinco Blocos Internos de Visão Computacional (VCE), três processadores Microcode Vector (VMP™), um Controlador Denali 64bit Mobile DDR, dispositivo 128bit internal Sonics Interconnect, dupla entrada de vídeo de 16 bits e controladores de saída de vídeo com 18bit, 16 canais de DMA e vários periféricos.

1.3.2. Projeto Prevent

O Projeto Integrado PReVENT é uma iniciativa da indústria automotiva europeia co-financiada pela Comissão Europeia para contribuir para a segurança rodoviária, desenvolvimento e demonstração de aplicações de segurança preventiva e tecnologias (PReVENT, 2011).

Os parceiros desse consórcio são muitos dentre eles: DaimlerChrysler AG; Audi; BMW; Ford; Peugeot Citroën; Renault; Volvo; Volkswagen; Delphi Delco Electronics; Philips; Siemens; Navigon AG; ERTICO-ITS Europe.

1.3.3. Processador IMAPCAR

O IMAPCAR é um processador de reconhecimento de imagem digital desenvolvido pela NEC Corporation, proporcionando um desempenho de até 100 GOPs (bilhões de operações por segundo), usando a tecnologia de processamento paralelo através de 128 elementos de

processamento, tendo um consumo de energia menor que 2 watts, o que torna interessante para uso em automóveis (IMAPCAR, 2011).

1.3.4. Projeto SENA

O projeto SENA (Sistema Embarcado de Navegação Autônoma) está em desenvolvimento no Laboratório de Mecatrônica do Departamento de Engenharia Mecânica da Escola de Engenharia de São Carlos (EESC-USP) e visa o desenvolvimento de um veículo de passeio capaz de evitar acidentes (Figura 5 obtida do site <http://www.eesc.usp.br/sena>), assistindo o motorista em situações de risco e auxiliar de forma cooperativa o motorista a melhorar sua capacidade e qualidade de direção. Outro desafio é movimentar o veículo de forma autônoma em ambientes urbanos.



Figura 5 – Projeto SENA (USP São Carlos)

1.3.5. Navegação Autônoma

Um tema de grande interesse para pesquisadores da área de navegação robótica assim como para veículos com detecção de pedestres é a navegação em ambientes externos. Nesse tipo de ambiente não estruturado necessita-se do desenvolvimento de veículos autônomos inteligentes para desviar de obstáculos. O desenvolvimento de veículos totalmente autônomos vem sendo estudado desde os anos 80 e possui diversas aplicações práticas, como a de diminuir o número de acidentes de trânsito. Devido às complexidades da

área, muitas questões ainda continuam em aberto, dando oportunidade a novas pesquisas na área de robótica. Uma das iniciativas nacionais nessa linha de pesquisa é o projeto CARINA (Carro Robótico Inteligente para Navegação Autônoma), desenvolvido no ICMC (Instituto de Ciências Matemáticas e de Computação) da Universidade de São Paulo. O plano de pesquisa inclui também um projeto que propõe o desenvolvimento de um sistema de estacionamento automático para veículos autônomos (LRM, 2011).



Figura 6 – Projeto Carina (LRM, 2011)

1.4. Organização do Trabalho

Capítulo 1: Introdução. São apresentados os objetivos a fim de situar o leitor no domínio do problema abordado. O domínio apresenta a motivação na qual o trabalho está fundamentado, o contexto em que se situa, nas áreas onde se aplica e por fim os principais desafios e propósitos necessários para a execução do projeto.

Capítulo 2: Fundamentação Teórica. São apresentados os métodos e técnicas utilizados no desenvolvimento do projeto. Há também um breve histórico dos métodos aplicados.

Capítulo 3: Considerações do Projeto. São apresentadas informações a respeito do problema abordado. Também há uma análise crítica dos trabalhos e técnicas que abordam

problemas da área em estudo e por final o 'Estado da Arte' com algumas das últimas tecnologias e métodos existentes.

Capítulo 4: Resultados Obtidos com a Verão 2. São relacionados os valores das medidas parciais de cada parte do sistema.

Capítulo 5: Conclusão. São apresentadas as conclusões da implementação do sistema e também são descritas as principais contribuições deste trabalho, bem como sugestões para a continuidade desta pesquisa.

2. Fundamentação Teórica

Um sistema de detecção que usa visão computacional pode reconhecer pedestres em frente de um veículo em movimento e assim, avisar o motorista sobre o risco iminente. Em geral, um sistema que utiliza a visão para detectar pedestres pode ser dividido em três passos consecutivos: detecção de pedestre; reconhecimento de pedestre; rastreamento de pedestre (LI, Z., et al, 2006).

2.1. Detecção de Pedestres

Observa-se que nas últimas décadas houve um rápido crescimento no desenvolvimento de aplicações de sistemas embarcados. Inicialmente, com o desenvolvimento da robótica industrial e a utilização de robôs no chão de fábrica que aumentou significativamente a eficiência dos processos produtivos além de reduzir seus custos (MATARIC, 2007). Os problemas enfrentados no desenvolvimento de sistemas de reconhecimento estão relacionados com a necessidade de interação adequada entre objetos físicos e entidades de um ambiente desestruturado e dinâmico combinados com o poder limitado de processamento de microcontroladores destinados a sistemas embarcados. Em consequência disto, a robótica móvel caracteriza-se como um diversificado campo a ser explorado e possibilita que novas aplicações sejam desenvolvidas (THRUN, 2001).

Existem várias abordagens para a detecção em vídeo-imagem de objetos em movimento, dentre elas (Bertozzi, 2002): Análise de imagens estáticas, Fluxo óptico, Visão Estéreo e Reconhecimento pela forma.

2.1.1. Trabalhos Relacionados

A Tabela 1 apresenta um sumário de um recente trabalho de pesquisa onde os autores DOLLÁR, P. et al (2011) analisaram a maioria dos algoritmos desenvolvidos para o Reconhecimento de Pedestres. Nesta tabela, são listados os principais resultados obtidos para dezesseis abordagens diferentes do problema de detecção. Esse estudo abrange apenas sistemas mono-câmera coloridas, como é o caso deste trabalho de mestrado. Um

dos resultados desta pesquisa identificou a necessidade de melhorias em sete áreas correlacionadas:

- Escalas pequenas (melhor desempenho é necessário);
- Oclusão (o desempenho degrada muito rapidamente mesmo sob pequena oclusão);
- Características de Movimento (utilizado em poucas abordagens);
- Integração Temporal (sistemas que utilizam a sequencias de imagens);
- Contexto (necessárias abordagens mais avançadas para baixas resoluções);
- Combinação de vários métodos;
- Coleta de novas fontes de dados.

Também há uma demanda considerável para pesquisas da melhoria em desempenho. Nos testes realizados pelos autores verificou-se que mesmo nas mais favoráveis condições, as detecções podem falhar de 20% a 30%. Há uma grande degradação do desempenho quanto testado com imagens pequenas de pedestres ou com oclusão parcial.

Em outro trabalho, dos autores LI, Z., et al (2006), as técnicas relacionadas aos sistemas de detecção de pedestres para veículos inteligentes são descritas visando apenas a utilização de câmeras. O objetivo da detecção de pedestres é extrair possíveis regiões da imagem onde o pedestre possa existir. Uma das importantes formas de detectar essas regiões é através do movimento do próprio pedestre. Uma sequência de imagens com uma câmera parada pode fornecer informações suficientes e detectar objetos em movimento, mas essa abordagem não permite detectar os pedestres quando estão parados. É também apresentado um importante estudo para detectar regiões de sequências de imagens contendo movimentos coerentes. Este estudo sugere que cada pixel possa ter um atributo indicando a probabilidade de pertencer a um grande objeto binário em movimento. O termo comum para esses objetos é denominado **BLOB** (*Binary Large Object*). A fim de detectar pedestres ou objetos que se movimentam em um fluxo de imagem, os BLOBs podem ser agrupados pela classificação de atributos como cor, coerência de movimento e posicionamento (Bregler, 1997). Os BLOBs foram cruciais para o desenvolvimento deste trabalho.

Tabela 1 – Estado da Arte em Detecção de Pedestres (DOLLÁR, P. et al, 2011)

Algoritmo	Característica							Detecção		Data
	Gradiente Hist.	Gradientes	Escala de Cinza	Cores	Textura	Similaridade-própria	Movimento	Frames por segundo	Taxa de Erro (log-médio)	Ano de Publicação
VJ			✓					0,447	95%	2004
SHAPELET		✓						0,051	91%	2007
POSEINV	✓							0,474	86%	2008
LATSVM-V1	✓							0,392	80%	2008
FTRMINE	✓	✓	✓	✓				0,080	74%	2007
HIKSVM	✓							0,185	73%	2008
HOG	✓							0,239	68%	2005
MULTIFTR	✓		✓					0,072	68%	2008
HOGLEP	✓				✓			0,062	68%	2009
LATSVM-V2	✓							0,629	63%	2009
PLS	✓			✓	✓			0,018	62%	2009
MULTIFTR+CSS	✓					✓		0,027	61%	2010
FEATSYNTH	✓				✓			--	60%	2010
FPDW	✓	✓	✓	✓				6,492	57%	2010
CHNFTRS	✓	✓	✓	✓				1,183	56%	2009
MULTIFTR+MOTION	✓					✓	✓	0,020	51%	2010

2.2. Tecnologias de Sensores

Diferentes metodologias foram elaboradas para o desenvolvimento de software de controle robótico, indo desde paradigmas baseados em modelos extremamente formais até tratamentos puramente reativos fortemente baseados em sensoriamento (MATARIC, 2007).

Os sensores de navegação, como bússola, GPS, giroscópio, acelerômetros e sensores de velocidade das rodas, permitem que os sistemas mecatrônicos auxiliem o veículo a trafegar de forma segura e controlada, mantendo sua estabilidade dinâmica e o seguimento seguro da trajetória previamente determinada.

2.2.1.Sensores de Espectro Visível

As abordagens baseadas no espectro visível tem atingido um maior interesse nas pesquisas destinadas à detecção de pedestres. Os sensores digitais de imagens que captam a luz no espectro visível, com o comprimento de onda de 0,48 nanômetros a 0,78 nanômetros, são adequados para um grande conjunto de cenários envolvidos na detecção em ambiente real.

As principais vantagens dos sensores digitais de imagem estão relacionadas ao baixo custo, grande quantidade de informações obtidas, tecnologia estável e a transmissão de dados padronizada. Em contra partida, as principais desvantagens são: a incapacidade dos sensores em produzir bons resultados nas áreas de sombras, problemas com reflexos, problemas com variações bruscas na iluminação, chuva, neve, neblina, noite, etc.

Um sensor de imagem digital é um chip eletrônico com dezenas de milhões de transdutores fotossensíveis, cada um deles capaz de converter a energia luminosa de um ponto da imagem agindo analogicamente como a retina dos olhos. A luminosidade das imagens é captada como carga elétrica e mensurada para ser registrada na forma de imagem digitalizada em valores numéricos.

Tabela 2 - Comparativo entre CCD e CMOS

Comparativo entre CCD e CMOS		
	CCD	CMOS
Custo	Caro para ser produzido, pois necessita de métodos especiais de fabricação.	Inexpressivo, pois pastilhas (wafers) são comumente usados em diferentes tipos de semicondutores.
Consumo	Consome 100 vezes mais que um CMOS	Baixo Consumo
Ruídos	Alta qualidade, baixo ruído	Susceptível a ruídos
Maturidade	Produzido há muito tempo; Imagens de alta qualidade, mais pixels	Menos maduro, mas equivalente em intervalos de baixa e media resolução para CCD
Funcionalidade	Tecnicamente viável; outros chips são utilizados	Outros circuitos facilmente incorporados no mesmo chip
Sinal de saída do chip	Voltagem (analogico)	Bits (digital)
Erro de Amplificação	Inexistente	Moderado
Complexidade do sensor	Baixo	Alto

Os sensores digitais mais comuns disponíveis no mercado são: CCD (charge-coupled device) E CMOS (complementary metal-oxide semiconductor). A Tabela 2 faz um comparativo resumido das duas tecnologias.

2.2.2.Sensores Ativos

Um sensor ativo emite energia, que é refletida pelos objetos na cena. A assinatura da energia refletida fornece informações sobre as adjacências do objeto. Os três principais sensores ativos são scanners radar, laser e sonar. O problema neste tipo de sensor é que quando usados em veículos muito próximos, a emissão causa interferência na leitura do veículo mais próximo.

2.2.3.Combinação de vários Sensores

É possível combinar sensores a Laser, Radar e ultrassom, com a informação visual (Beauvais, 2000), (Stiller, 2000), (Guilherme de Lima Ottoni, 2003). A complexidade e a quantidade de processamento necessário para realizar a junção das informações devem ser levadas em consideração, pois diferentes sensores necessitam ser calibrados (Kelber, 2005).

O projeto *Autonomes Fahren* (Becker, 1998), (Simon, et al., 1999), (Simon, 2000), desenvolvido pela empresa Volkswagen e parceiros, utilizaram uma estrutura com vários sensores simultâneos: radar, *scanners* a laser, sistema de visão estéreo, plataforma inercial e GPS diferencial, conforme apresentado na Figura 7 e na Figura 8. Neste caso, é necessário o desenvolvimento de um sistema computacional capaz de criar a fusão dos dados dos diferentes sensores de forma a sincronizar com precisão os vários sensores, minimizando os erros de instrumentação (Becker, 1998).



Figura 7 - Radar e Scanner a Laser (projeto Autonomes Fahren)



Figura 8 - Visão Estereoscópica (projeto Autonomes Fahren)

2.2.1. Visão Estéreo

A visão estéreo possibilita a noção de profundidade e permite a reconstrução de objetos em terceira Dimensão. Suas desvantagens estão relacionadas com o grande custo computacional e complexidade, além ser sensível a calibragem da câmera.

A Figura 9 mostra um sistema de detecção baseado em visão estéreo utilizado no projeto PROTECTOR (Gavrila, 2004). Nesta figura, na área frontal do sensor é mostrada a capacidade de detecção de pedestres de acordo com o tipo de sensor. O aviso é dado apenas se o pedestre estiver no caminho do veículo e na região abrangente do ângulo de visão. Dados fora da região de detecção não são considerados para evitar falsas detecções.

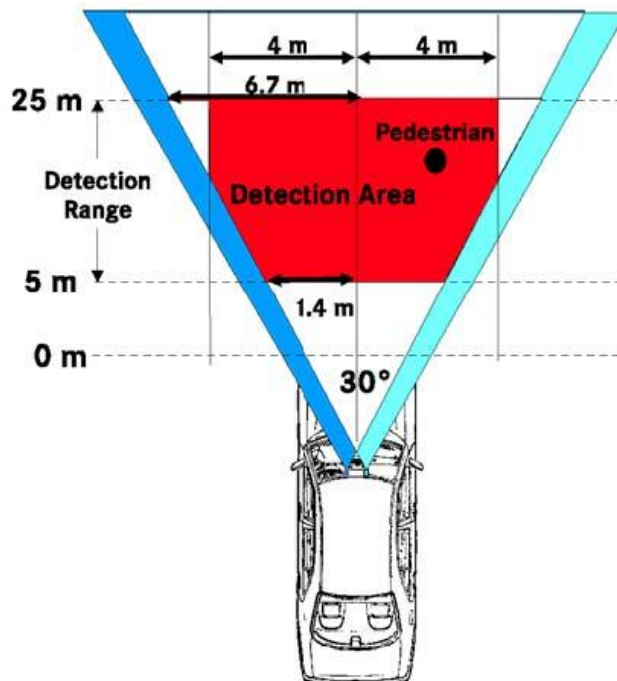


Figura 9 - Exemplo de detecção com visão estéreo - sistema Protector

2.3. Computação Reconfigurável

Após vinte e sete anos de desenvolvimento, os dispositivos FPGA estão consolidados no mercado como uma ferramenta de prototipagem de hardware. Processadores Pentium e Atom da Intel foram sintetizados inicialmente em FPGA (Wang, P. H., 2009). Sua ampla utilização como ferramenta para elaboração de testes e depuração de projetos em hardware expandiu sua capacidade de processamento e de seus elementos lógicos-CLB (Tabela 3).

As principais vantagens da construção de um processador de vídeo customizado usando FPGA são o período de tempo menor para disponibilizar ao mercado (time-to-market), menor investimento em pesquisa e desenvolvimento, e a possibilidade para rápidas modificações adicionais. Em desvantagem estão o alto custo dos dispositivos FPGA (Kirischian, 2008) e sua baixa frequência. Entretanto, o desafio principal está na criatividade e a flexibilidade para reorganizar a aplicação e testar inúmeras formas de projeto de uma maneira muito rápida. Atualmente, os protótipos e produtos ADAS que utilizam câmeras são baseados em dispositivos de alta performance com Processador Digital de Sinais (DSPs), combinados com hardware reconfigurável utilizando-se processamento paralelo (Techmer, 2007).

Uma grande vantagem do FPGA é que ele aceita qualquer tipo e arquitetura de hardware, não se limitando somente a um processador e permite simular os resultados do projeto antes da fabricação de chips em larga escala. Além de ser útil para fazer protótipo, ele pode ser útil quando existe a necessidade de desenvolver um hardware específico do qual não se tem demanda para grande escala, assim somente se produz alguns FPGA's, abaixando o custo.

Dessa forma, grande parte do custo de um processador ou chip qualquer está em seu desenvolvimento. Um simples microcontrolador pode custar vários milhões para ser desenvolvido. Contudo, o custo de produção por unidade é relativamente baixo, de forma que os chips mais vendidos tenham o custo inicial amortizado e passam a ser cada vez mais baratos.

As últimas informações extraídas do site dos dois fabricantes mais conhecidos de FPGA (Xilinx e a Altera) apresentam seus últimos produtos para o mercado com novas tecnologias, consta também para comparativo um dos últimos lançamentos de processadores Intel. De uma maneira muito rudimentar, pode-se intuir que em uma FPGA Virtex 7 caberiam dezoito núcleos de última geração da Intel, levando-se em consideração apenas o número de transistores. De uma forma análoga, admitindo-se o padrão da indústria de semicondutores de que são necessários quatro transistores para a construção de uma porta NAND CMOS, na mesma FPGA caberiam 1.7 bilhão de portas lógicas. Portanto, isto demonstra o enorme potencial desta tecnologia para a implementação de sistemas cada vez mais complexos e maiores.

Tabela 3 – Comparativo entre as diferentes tecnologias

Característica	Processador Intel I7 3960X- Hexa Core	FPGA XILINX Virtex 7 XC7V2000T	FPGA ALTERA Stratix V 5SEEB
Blocos Lógicos CLB/LE		1.954.560	952,000
Transistores	2.270.000.000	6.800.000.000	3,800,000,000
Tecnologia	32nm	28nm	28nm

2.4. Síntese das técnicas envolvidas

De forma geral, os sistemas baseados em câmeras desenvolvidos para ADAS podem ter diferentes tipos de processamento. A Figura 10 ilustra o princípio da cadeia de processamento de um sistemas ADAS (Techmer, 2007) organizado por níveis de complexidade. O paralelismo pode ser encontrado no Nível de Objetos (por exemplo, o rastreamento simultâneo de vários pedestres em suas rotas).

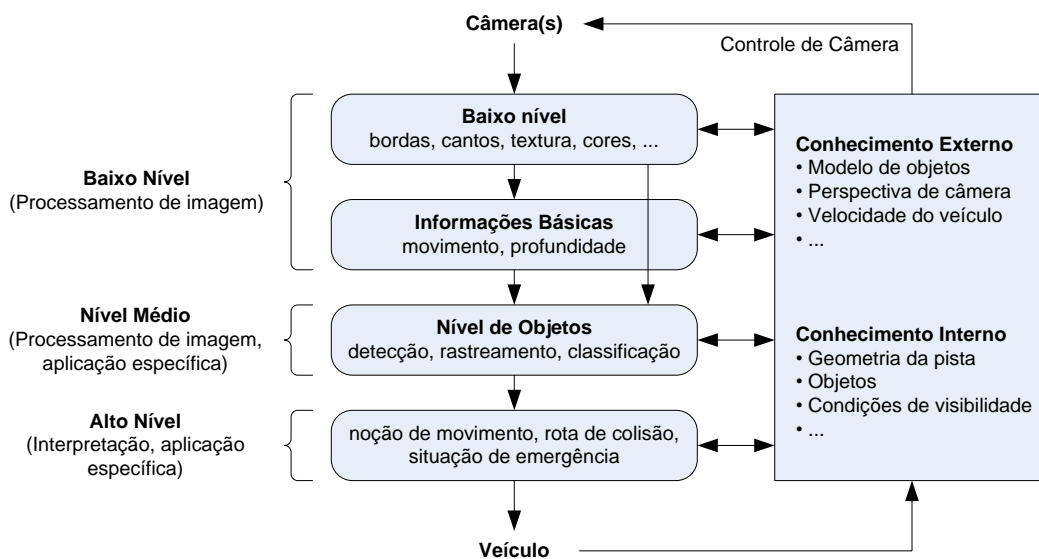


Figura 10 - Cadeia de processamento de imagens (adaptado de Techmer, 2007)

Os primeiros passos desta cadeia de processamento referem-se ao tratamento intensivo dos dados provenientes da câmera, ou seja, aquisição da imagem e filtros. O próximo passo refere-se descoberta de objetos, posteriormente a classificação e rastreamento até a análise da situação específica. Nesta sequência, as informações tornam-se cada vez mais abstratas.

Uma maneira de representar as etapas principais que envolvem a arquitetura de um sistema de visão computacional é mostrada na Figura 11 (Fu, *et al.*, 1987) e (Gonzalez, *et al.*, 2000). Estas etapas são: Sensoriamento (Aquisição), Processamento de Imagens (Pré-processamento), Segmentação, Descrição, Reconhecimento e Interpretação.

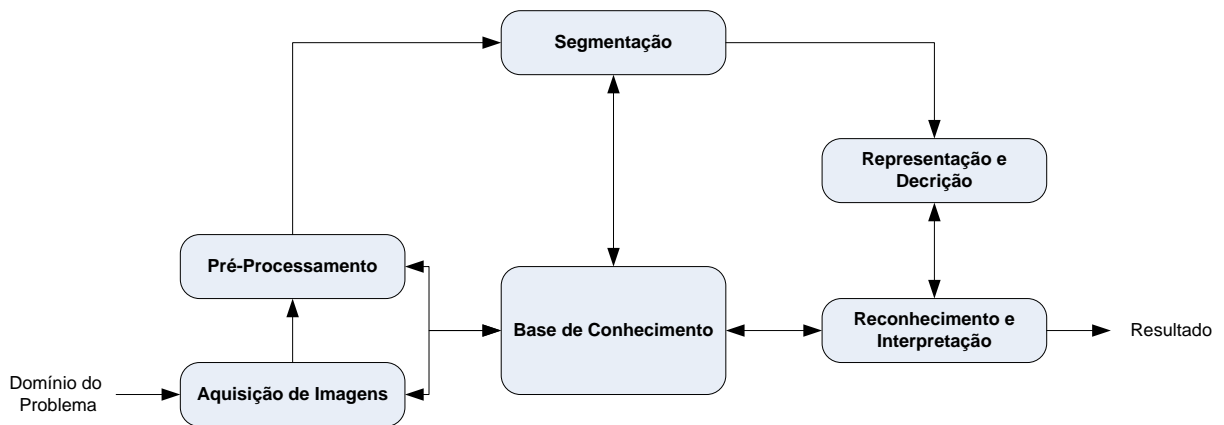


Figura 11 - Etapas básicas de um sistema de visão computacional

Aquisição e digitalização: A imagem do sensor é transformada em uma imagem digital na forma de matriz de valores inteiros (chamados pixels). A natureza do sensor e da imagem que ele produz são determinadas pela aplicação.

Pré-processamento: Corrige defeitos e imperfeições que aparecem durante a aquisição da imagem. Podem ter como causa as características físicas do sistema, condições impróprias de iluminação, etc.

Segmentação: Tem por objetivo dividir a imagem em partes constitutivas, chamadas de regiões. Na imagem há detecção de descontinuidades (contornos) e ou de similaridades (regiões) na imagem.

Representação: Esta etapa agrupa os resultados das etapas precedentes a o armazenamento dos diversos padrões que contem o conhecimento inicial. Também permite medir as formas resultantes da segmentação.

Reconhecimento e Interpretação da Imagem: Esta etapa reconhece os padrões através análise de padrões os elementos da imagem. Os algoritmos exigidos ao final da cadeia são mais complexos, nesta etapa ocorrerá uma interpretação das informações e será dada a resposta aos controles do veículo.

Um requisito muito comum em sistemas embarcados é a execução em tempo real, especialmente quando a aplicação envolve reconhecimento de imagens em veículos

autônomos inteligentes (Bertozzi, 2002). Entretanto, a interpretação de imagens com alta resolução espacial necessita de uma capacidade elevada de processamento de dados.

A detecção de pedestres utilizando câmeras óticas convencionais em um veículo em movimento é bastante complexa. As principais dificuldades estão relacionadas com a estabilização causada pelo movimento do veículo, com interferências no sinal de vídeo e com variações da luminosidade. Contudo, a abordagem de (Jalalian, 2008) introduz uma solução satisfatória, onde os componentes de detecção utilizam redes neurais para criar mapas auto-organizáveis como mecanismo de classificação.

O gráfico da Figura 12 demonstra uma visão geral para implementação de um sistema de detecção de pedestres para uma câmera em movimento.

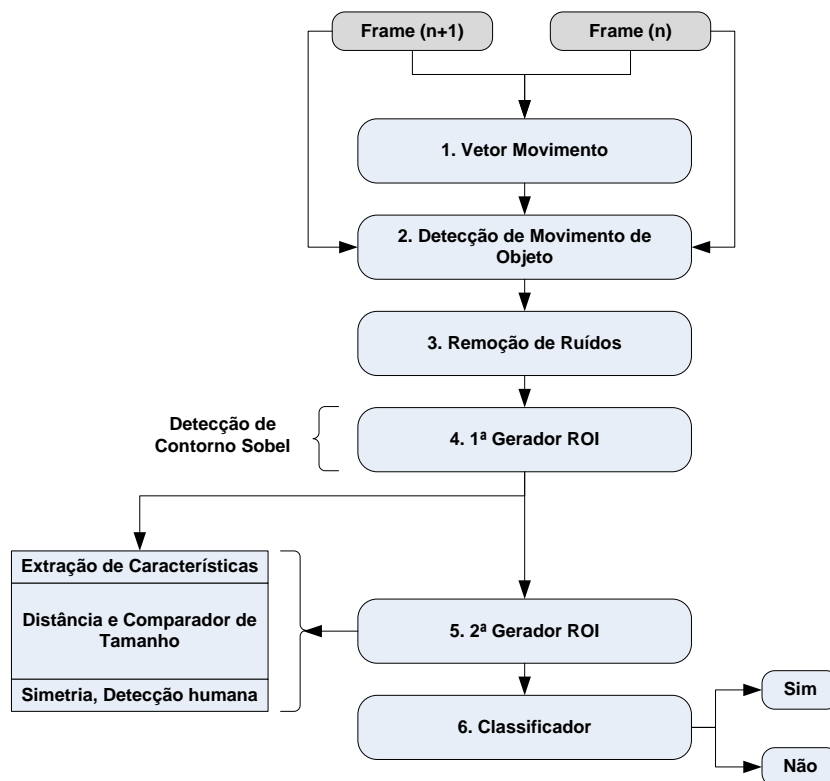


Figura 12 - Visão Geral para a implementação (Jalalian, 2008 - modificado)

Uma das características significantes do processamento através de visão é a grande quantidade de dados envolvidos. Dependendo da abordagem, o volume de processamento para filtro de ruídos de um único quadro pode ser muito grande, exigindo milhares de ciclos por segundo. Neste caso, a necessidade de processamento ultrapassa a capacidade de um microprocessador comum sendo necessário adotar algoritmos compactos que permitam o

uso do paralelismo e a construção de processadores de alta velocidade específicos para a função.

2.5. Restrições em relação ao tempo

Para determinar o limite da capacidade humana na reação aos perigos iminentes, estudos determinaram o tempo de resposta do cérebro humano em responder aos estímulos externos. O requisito de distância mínima exigida para a detecção deste projeto é calculado através da relação entre a velocidade do veículo, o tempo de resposta do cérebro e tempo necessário para frenagem. A detecção do pedestre deve ocorrer antes de chegar ao ponto crítico, onde é encontrada a distância necessária para que o condutor possa reconhecer o aviso de perigo e acionar os freios (Alonso, 2005). A Figura 13 ilustra vários exemplos de velocidades com os respectivos tempos para o reconhecimento do problema e para a frenagem.

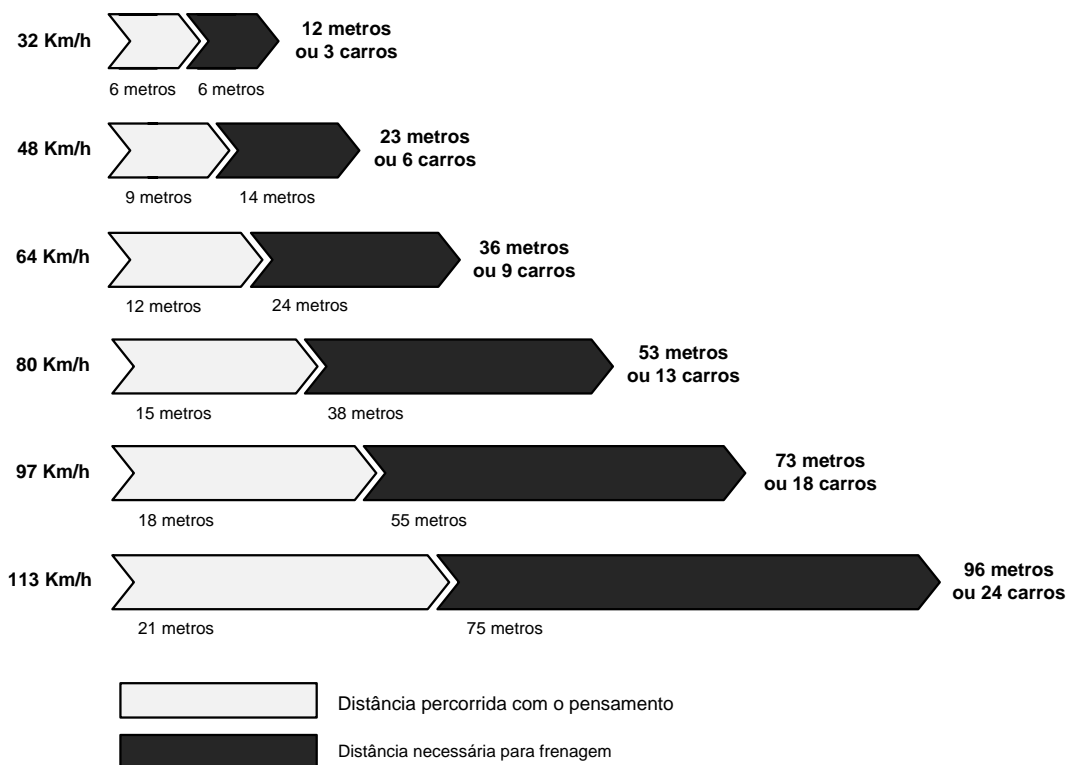


Figura 13 - Distância para parar (Alonso, 2005)

As causas normais para um atropelamento são quatro: a velocidade excessiva, a ausência de reação, a percepção ou reação tardia (sendo que as três primeiras causas

recaem sobre o condutor do veículo atropelador) e a entrada inopinada (atribuída ao pedestre) (Aragão, et al, 2003).

Entre o momento em que o condutor vê um pedestre e aquele em que começa a frear, há um intervalo denominado **tempo de reação**. Assim, o tempo de reação reflete a duração da transmissão do impulso nervoso recebido pelo olho até o envio de comando ao órgão de ação (o pé, que aciona o freio). Este tempo varia de pessoa para pessoa, também pode variar de acordo com o estado de fadiga e também por efeitos químicos ao corpo, como remédios ou consumo de bebidas alcoólicas.

Os tempos de reação considerados nos acidentes de trânsito aqui apresentados utilizam a seguinte graduação de valores mínimos (Tochetto, et al., 2005):

- Evento esperado em condições externas favoráveis: 0,75 segundos
- Evento esperado em condições externas adversas ou evento esperado em condições externas favoráveis: 0,75 segundos a 1 segundo
- Evento inesperado em condições externas adversas: 1,5 segundos

A distância percorrida durante o tempo de reação (DTR) varia em função da velocidade do veículo, conforme a Equação (1).

$$DTR = V_i \times TR \quad (1)$$

Onde:

V_i = Velocidade inicial (m/s)

TR = Tempo de reação (s)

DTR = Distância percorrida durante o tempo de reação (m)

A distância de frenagem (equação (2)) é o espaço entre a posição do veículo no momento em que o condutor aciona os freios e a posição onde efetivamente o veículo para de se deslocar. É importante observar que a distância de frenagem depende do sistema de frenagem, da velocidade e da aderência do veículo sobre a pista. A aderência pode variar em decorrência do sistema de frenagem, das condições dos pneus e das condições da pista que pode estar seca, molhada, com óleo, etc.

$$DF = Vi^2 / 2y \quad (2)$$

Onde:

Vi = Velocidade inicial (m/s)

y = desaceleração (m/s²)

DF = Distância de frenagem (m)

Assim, conforme demonstrado na Equação (3), a distância total de parada é resultante da soma da distância percorrida durante o tempo de reação com a distância de frenagem.

$$DTP = DTR + DF \quad (3)$$

Onde:

DTR = Distância percorrida durante o tempo de reação (m)

DF = Distância de frenagem (m)

DTP = Distância total de parada (m)

Caso identificada uma rota de colisão, o sistema preditivo entrará em ação advertindo o motorista por meio sonoro-visual e em casos de ausência de decisão por parte do condutor o sistema poderá desacelerar (através dos sistemas eletrônicos do veículo) ou acionar freios para evitar a colisão. Tudo isso, mantendo simultaneamente a estabilidade do veículo nas diversas condições de ambiente. Uma vez que os freios forem acionados de forma manual, o sistema concederá prioridade para as decisões tomadas pelo motorista, neste caso, também poderá ser ajudado pelo sistema de freio ABS, sistema anti-derrapagem e demais que o veículo possua.

Com objetivo na utilização urbana, este projeto está preparado para trabalhar com velocidades próximas a 32 km/h. Assim, o requisito de distância mínima é de 12 metros, ou seja, quando o veículo chegar aos 12 metros do pedestre, é necessário que o sistema já tenha detectado a situação de perigo.

3.Considerações de Projeto

Conforme descrito por LI, Z., et al (2006) no item 2.1, os processos de detecção de pedestres baseados em visão podem ser divididos em três passos consecutivos: detecção de pedestre; reconhecimento de pedestre; rastreamento de pedestre. Este projeto focou-se apenas na detecção de pedestre. Em trabalhos futuros os outros passos poderão ser implementados a partir deste projeto.

O projeto de construção do sistema de detecção de pedestres foi dividido em três versões. A primeira versão (Figura 14) teve como principal objetivo de validar os algoritmos de detecção. Para isto, através de uma câmera instalada em um veículo, foi feita uma filmagem de um ambiente urbano real e convertido posteriormente em quadros gerando a base de imagens para o experimento. Com esta base, foi testado o software em um computador de propósito geral. Após a comprovação do funcionamento da lógica, a segunda versão do sistema compreende em embarcar o software em um processador softcore, identificando as modificações necessárias para a adequação à plataforma. Por final, a terceira versão compreende na conversão completa do software para hardware.

3.1. Especificação da Primeira Versão

Para facilitar a conversão de cada algoritmo nas diferentes versões, o sistema foi dividido em blocos. Nesta primeira versão, o primeiro bloco é responsável por converter as cores de saída da câmera para a escala de cinza. Para estabilizar a imagem, o segundo bloco armazena o quadro em memória e o compara com o quadro anterior. Para isto, o algoritmo alterna entre duas memórias, disponibilizando o quadro anterior liberando uma memória para registrar o quadro atual. A detecção de movimento feita pelo terceiro bloco é feita através da subtração pixel a pixel das intensidades. O resultado da detecção de movimento é uma matriz binária contendo pontos que se movimentaram entre os dois quadros. O quarto bloco é utilizado para diminuir o ruído, fazendo com que pontos isolados da matriz de movimento sejam excluídos. Baseando-se nos valores da matriz binária com o filtro de ruído, o quinto bloco cria um vetor contendo os aglomerados de pixels em movimento. Por final,

para classificar esses aglomerados o sexto bloco verifica se estes correspondem às características de pedestres e se este está situado em uma área de risco da imagem. Por se tratar de vídeo contínuo, o ciclo se repete para cada novo quadro (loop frames).

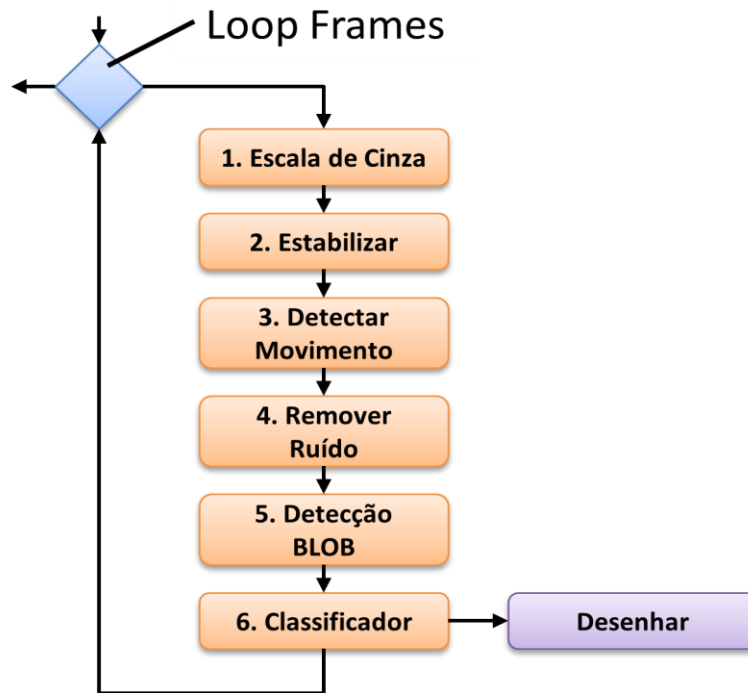


Figura 14 – Detalhamento da Versão 1

Em resumo, as principais características deste sistema são:

- Realizar operações de aquisição e pré-processamento de imagem colorida;
- Construção de um algoritmo que faça a detecção de pedestres;
- Exibir imagem identificando objetos de interesse
- Evitar usar recursos de programação não disponíveis no processador NIOS II
- Fazer aferições de desempenho.

3.1.1. Conversão para Escala de Cinza

O primeiro bloco do sistema trabalha com o pré-processamento da imagem. Apesar das imagens captadas pela câmera CMOS serem coloridas, o projeto foi preparado para trabalhar apenas com a intensidade de cada pixel. Isto diminui a quantidade de informações de trabalho mas mantém dados suficientes para efetuar as detecções. Tanto o software quanto o projeto do Hardware Reconfigurável utilizam com entrada a imagem no formato

RGB. O conversor para escala de cinza mostrado na Figura 15, faz uma varredura por todas as linhas e todas as colunas na figura, ou seja, percorre todos os pixels da imagem. Nesta sub-rotina também é armazenada a somatória dos valores de intensidade dos pixels que será usada no bloco de estabilização.

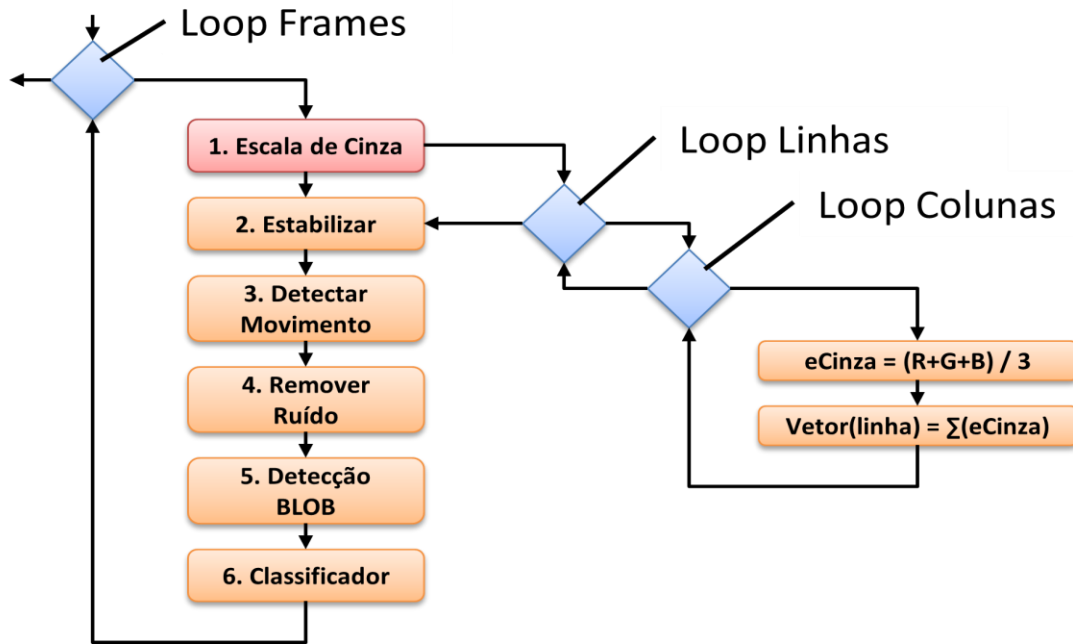


Figura 15 – Especificação Conversor para Escala de Cinza

O espaço de cor RGB é amplamente utilizado como modelo de cor em diversos dispositivos. Neste espaço, cada cor é representada por 3 valores: vermelho (R), verde (G) e azul (B) e posicionados ao longo dos eixos do sistema de coordenadas cartesianas. Os valores de RGB assumem o intervalo de [0,1] ou, em alguns casos, a faixa de [0-255]. Desta forma, o preto é representado como (0, 0, 0), o branco é representado como (1, 1, 1) ou (255, 255, 255). Preto e branco são representadas em cantos opostos no cubo (Figura 16). Os tons de cinza são representados com valores idênticos das componentes R, G, B.

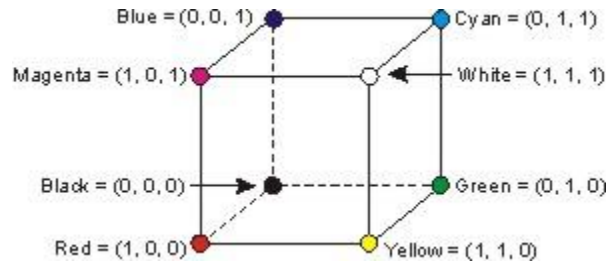


Figura 16 - Espaço de cor RGB

A técnica utilizada para converter os valores RGB para Escala de Cinza foi a média aritmética das primitivas de cada cor. O valor resultante desta fórmula é a intensidade (também conhecido com brilho) do pixel.

3.1.2. Estabilização da Imagem

Neste projeto, as primeiras tentativas de detectar movimento sem estabilização de imagem resultaram em dados imprecisos, pois o conceito de comparação de quadros requer que o maior número de pontos de um quadro de imagem esteja alinhado com o quadro anterior.

A abordagem utilizada para o segundo bloco levou em consideração os dois principais fatores:

- 1) A necessidade de um algoritmo com pouca necessidade de processamento computacional;
- 2) Um algoritmo que possa ser usado na construção de um Hardware que possibilite trabalhar em paralelo com o carregamento dos quadros.



Figura 17 – Demonstração do Vetor de Estabilização

O princípio de funcionamento do Bloco de Estabilização é armazenar em memória a somatória de intensidade de cada pixel de linha e comparar os valores de cada linha com os valores das linhas do quadro anterior até encontrar um valor onde a subtração linha a linha seja a menor possível. O valor resultante é o índice de deslocamento. Pode-se observar que o sistema de estabilização passa a funcionar somente na leitura do segundo quadro. O sistema não visa mostrar a imagem estabilizada, apenas necessita do índice de deslocamento do eixo vertical para ser usado nos procedimentos de detecção de movimento.

A execução do método refere-se a um programa que disponibiliza dois vetores para valores inteiros. Estes vetores têm o mesmo número de linhas da imagem (que neste experimento trata-se de 480 linhas). A Figura 17 mostra um quadro capturado e ao lado direito a representação gráfica do vetor de intensidade. Nota-se que a parte superior do vetor tem valores maiores, pois as linhas superiores da imagem têm intensidades maiores.

A primeira parte do algoritmo executa linha a linha uma somatória da intensidade dos pixels e armazena o valor resultante no elemento do vetor correspondente. Na segunda

etapa, ou seja, na leitura do próximo quadro, os vetores são comparados identificando qual o índice de deslocamento de linha que melhor se adéqua a subtração dos vetores.

A Figura 18 mostra um exemplo de deslocamento. Nesta figura, V1 representa um vetor com a somatória de valores de intensidade do quadro previamente carregado e V2 refere-se à somatória das intensidades do último quadro carregado. Pode-se observar, em destaque, que a imagem do último quadro deslocou-se para baixo, ou seja, a câmera que captura as imagens deslocou-se para cima. A divergência causada pelo movimento para frente é pouco significativa para quadros consecutivos em relação ao deslocamento da câmera para cima ou para baixo causado pelo movimento do veículo. Assim, essa divergência não está sendo levada em consideração neste projeto.

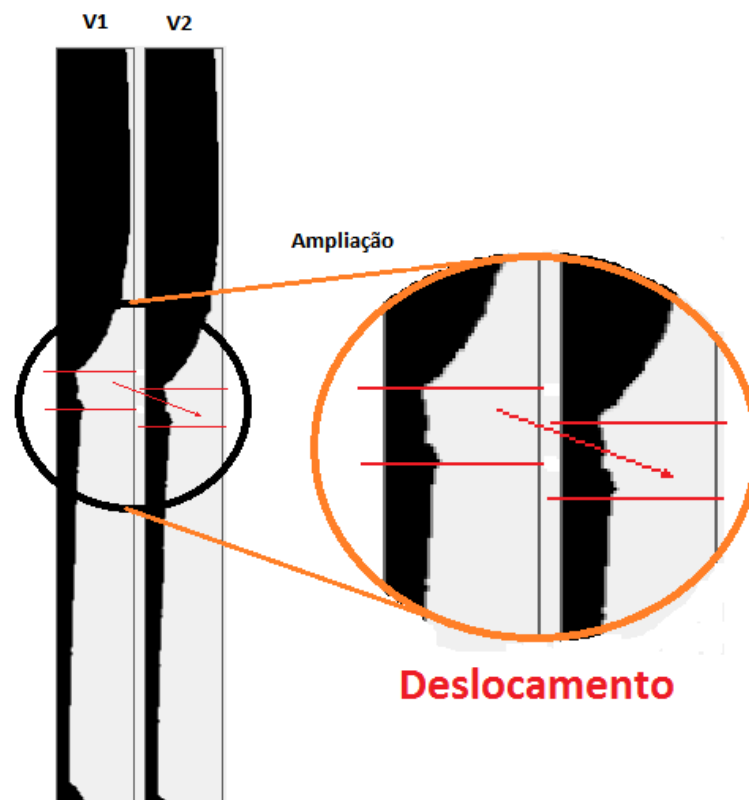


Figura 18 – Comparação de Dois frames

Outra possível solução para resolver este tipo de problema é a criação de vetores de movimento da imagem (Jalalian, 2008). Dessa forma, para minimizar o efeito de movimento da câmera, a imagem é dividida igualmente em quatro regiões e são calculados os quatro vetores representativos de movimento (Figura 19). Com esses vetores é possível tomar dois

quadros de imagem e com os vetores fazer a compensação de deslocamento. Uma vez que os quadros estão alinhados, torna-se possível fazer a diferenciação das imagens para detectar apenas os objetos que se deslocaram e, conseqüentemente, ignorando-se o cenário de fundo.

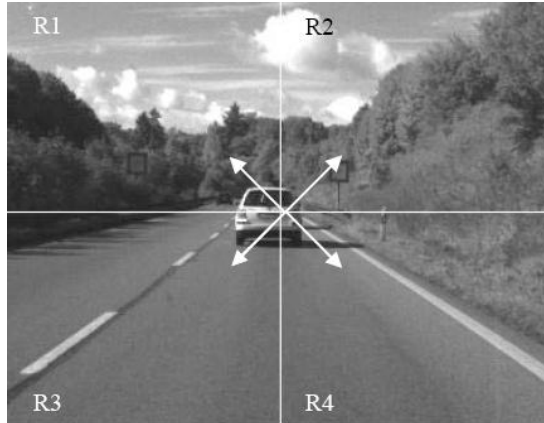


Figura 19 - Vetores Representativos de Movimento (Jalalian, 2008)

Normalmente, o vetor movimento é calculado pela Diferença da Média Absoluta (MAD – Mean Absolute Differencing), conforme a equação (4)

$$MAD = \frac{1}{mn} \sum \sum |F_n(i_1, j_1) - F_{n+1}(i_2, j_2)| \quad (4)$$

Onde F_n é o n-ésimo quadro e F_{n+1} é o (n+1)-ésimo quadro.

E i_1, j_1 são as coordenadas de F_n e $i_2=j_1+y, j_2=j_1+y$ são as coordenadas de F_{n+1} . As variáveis m e n referem-se respectivamente a largura e altura em pixels da imagem.

Um processo bastante usado para detectar objetos em movimento é a diferenciação de quadros de imagem (frames). O algoritmo utilizado para isto tem uma complexidade baixa. Contudo, como visto no item anterior, este algoritmo não funciona caso a câmera esteja em movimento. Assim, com os vetores representativos de movimento da câmera é possível compensar o deslocamento fazendo-se com que os quadros fiquem alinhados.

A equação (5) para diferenciação de quadros é apresentada a seguir:

$$d(i, j) = |f_n(i + x, j + y) - f_{n-1}(i, j)| \quad (5)$$

Onde x e y são os vetores representativos de movimento da câmara.

O algoritmo de estabilização de imagem é rápido pela forma como foi concebido, mas para operação de altas taxas de quadro por segundo (em um sistema real), otimizações são necessárias. Uma das escolhas para utilização deste algoritmo é a possibilidade de execução em pipeline. A principal vantagem dessa abordagem é a simplicidade de sua construção, mas em desvantagem, não há muita precisão.

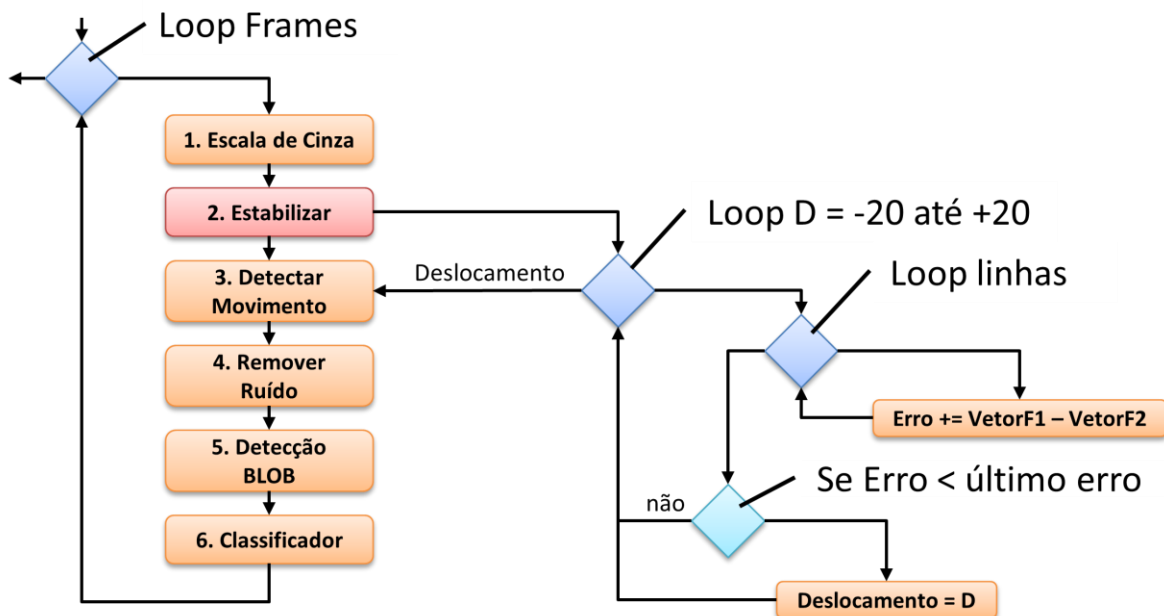


Figura 20 – Especificação do Sistema de Detecção

Para encontrar o melhor índice de deslocamento, o sistema de detecção faz uma varredura entre as 40 variações de linha (Loop D= -20 até 20, onde D é uma variável contento o deslocamento teste). Para isso ele simula o erro resultante da subtração de todas as linhas do quadro atual e do quadro anterior (VetorF1 - VetorF2). Sucessivamente, o algoritmo desloca o índice da linha e faz novamente a subtração até que todas as 40 posições sejam testadas. Ao final o deslocamento que obteve o menor erro é transferido para o próximo bloco como sendo o índice de deslocamento. Este algoritmo pode ser visualizado na Figura 20.

3.1.3. Detecção de Movimento

O princípio de funcionamento da detecção de movimento é armazenar um quadro em um buffer de memória e compará-lo com o quadro subsequente. Neste caso, a detecção passa a funcionar somente na leitura do segundo quadro.

A técnica original proposta por (Jalalian, 2008) consiste em calcular histogramas da imagem, combinando-os para formar o vetor de características. Primeiramente, a imagem é normalizada. Em seguida são calculados os histogramas horizontais e verticais.

Nesta implementação é calculado simplesmente pela soma dos pixels pretos em cada linha da imagem. A equação (6) é apresentada a seguir:

$$H_h(m) = \sum_n f(m, n) \quad (6)$$

Onde **H** representa o vetor de pesos, *m* representa cada linha da imagem e *n* representa o valor de intensidade de cada pixel pertencente a linha *m*. O resultado da diferenciação dos quadros alinhados é mostrado na Figura 21.



Figura 21 – Matriz de Pixels em Movimento

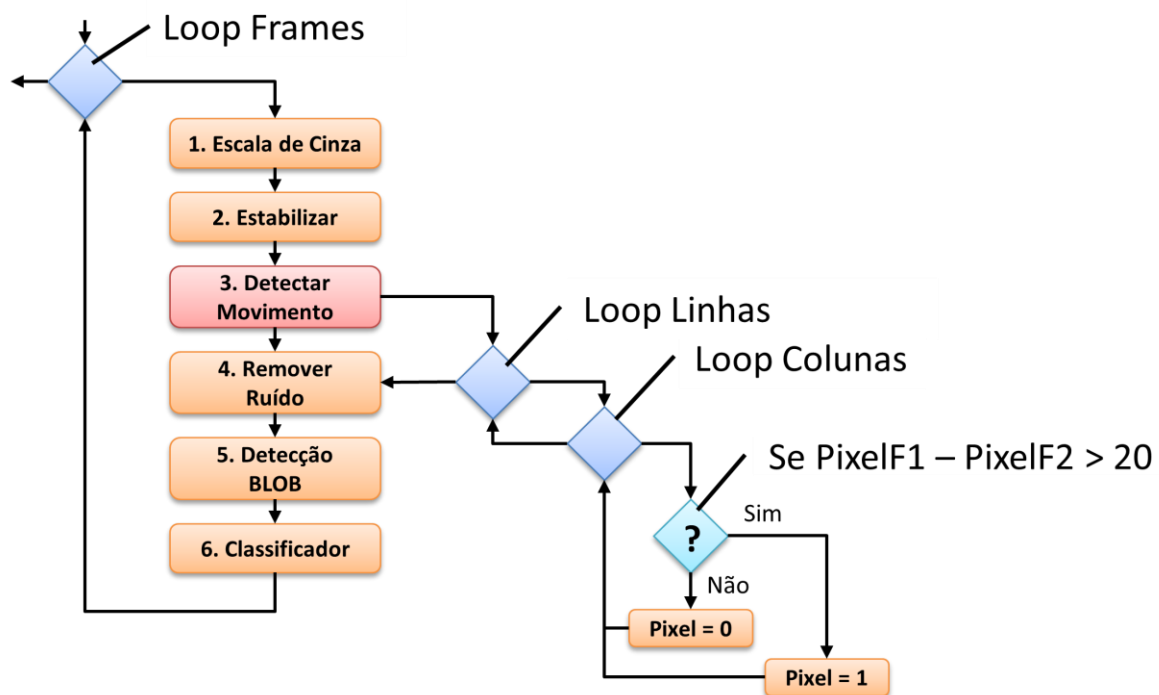


Figura 22 – Especificação do Bloco de Detecção de Movimento

O algoritmo apresentado na Figura 22 faz uma varredura em todas as posições do quadro e faz a diferenciação dos valores de intensidade entre o quadro atual e o quadro anterior. Caso essa diferença ultrapasse a 20 níveis de intensidade (equivalente a 8% de variação) o pixel gerado no mapa de bits será 1 senão o pixel recebe o valor 0.

3.1.1. Remoção de ruídos

Após a diferenciação dos quadros de imagens subsequentes é possível detectar os objetos que na transição se movimentaram. Entretanto, o movimento irregular da câmera pode gerar ruído quando houver a diferenciação. Para resolver este problema, um passo de pré-processamento é necessário para minimizar regiões de ruído. Vários algoritmos para redução de ruído podem ser usados, como por exemplo, o filtro não linear de medianas, que suaviza a imagem sem diminuir a resolução. Neste projeto, visando minimizar o processamento, foi desenvolvido um algoritmo customizado que permite a execução em paralelo ao procedimento de Detecção de Movimento. Este algoritmo é similar ao algoritmo de Erosão para imagens binárias. Este procedimento elimina objetos finos ou pequenos.

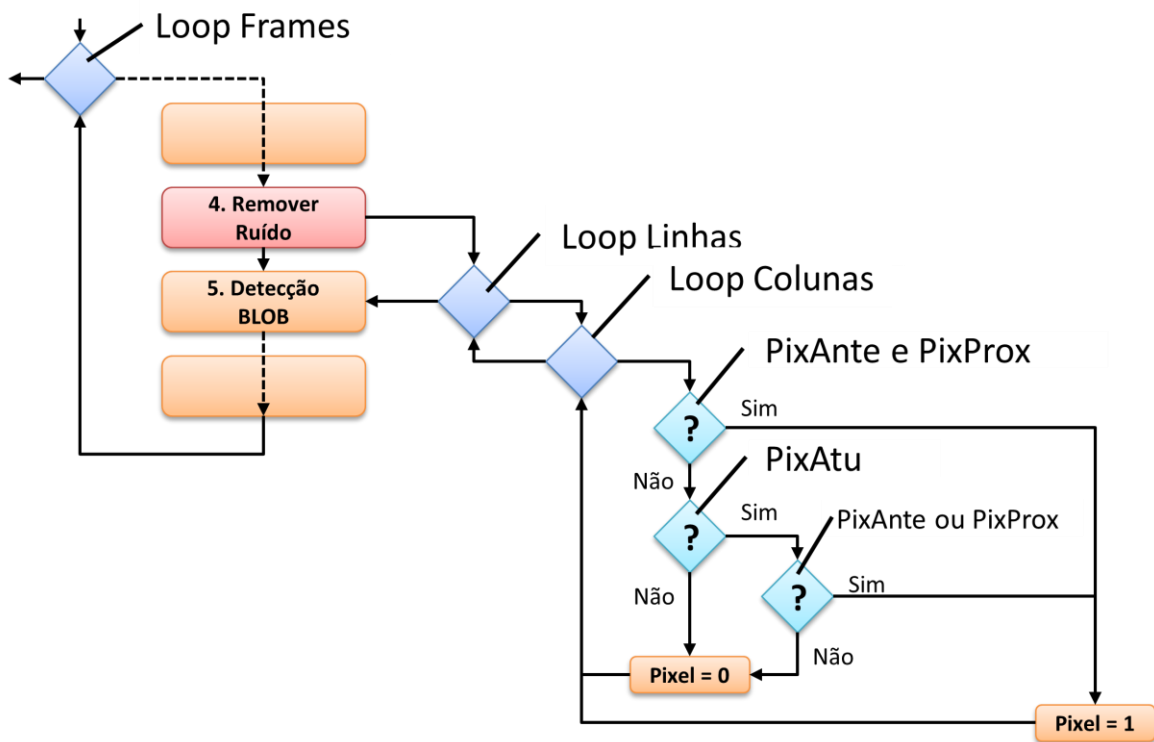


Figura 23 – Especificação do Bloco de Remoção de Ruído

O algoritmo da Figura 23 simplesmente verifica se os pixels vizinhos são do mesmo valor. Em resumo, se o pixel anterior e o próximo pixel tem valor 1 então o pixel atual recebe valor 1 (Isto preenche espaços vazios entre pixels), senão, se o pixel atual for igual a 1 e o anterior ou o próximo também forem iguais a 1 então o pixel atual = 1 senão o pixel atual = 0.

3.1.2. Detecção de Objetos Binários

Para esta aplicação, detectar pontos relacionados em uma imagem binária é um fator fundamental, fazendo-se a segmentação e identificando-se objetos no quadro (frame).

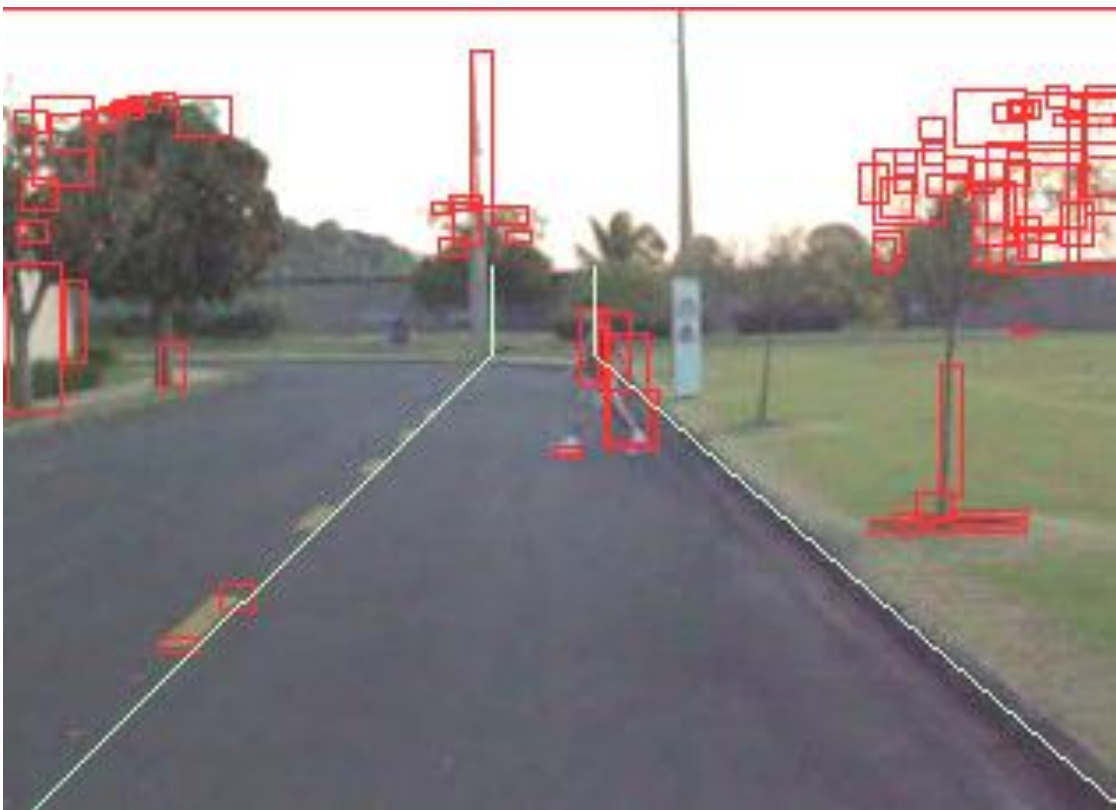


Figura 24 – Detecção de Múltiplos BLOBs na imagem

A ideia do algoritmo de detecção BLOB (Binary Large Objects) deste projeto baseou-se em modelos disponíveis em bibliotecas de processamento de imagem. Adaptações foram necessárias para que o mesmo código pudesse ser convertido para hardware na versão 3. Um exemplo de adaptação foi a não utilização de classes de objeto e de recursividade, ou seja, o algoritmo faz o procedimento em forma serial utilizando-se vetores de

armazenamento. Outro exemplo de adaptação foi a não utilização de comandos *for-next*; ao invés disto as rotinas foram convertidas para iterações (loops). A Figura 24 mostra um dos quadros do experimento com vários BLOBs detectados.

3.1.3. Classificação

A estratégia utilizada para classificação de pedestres utiliza a saída do sistema de detecção de Objetos Binários (BLOB) e valida dois princípios:

a) Proporcionalidade do objeto detectado:

Para exemplificar, sabe-se que um pedestre é sempre mais alto do que largo. Assim, o algoritmo faz uma verificação em todos os BLOBS detectados e descarta os que não se enquadram na regra ($\text{altura} > \text{largura}$). Também são descartados BLOBS muito pequenos que estão isolados.

b) Estipulação da Região de Risco de Acidente

Com a câmera instalada no veículo é possível parametrizar as regiões do frame onde visivelmente existem possibilidades de ocorrer acidentes e as áreas onde é pouco provável. Assim, o algoritmo desconsidera os BLOBS das áreas que se encontram acima da linha estimada do horizonte (ajustada de forma manual quando instalada a câmera), desconsideram-se também as laterais da imagem, criando-se um corredor virtual denominado área de risco. Para melhor visualização, foi adicionado no programa um código para sobrescrever a imagem original desenhando as linhas divisórias da área de risco (Ver linhas verdes destacadas na Figura 27).

A técnica de detecção por proporcionalidade se baseia em estimar o tamanho do pedestre de acordo com a distância onde o objeto foi detectado (Figura 25).

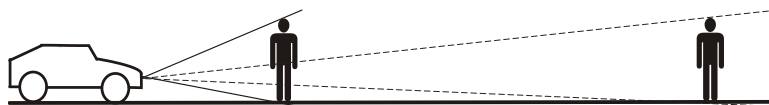


Figura 25 - Tamanho variado de pedestres devido a distância

A Figura 26 mostra o resultado do processo após a utilização do filtro de proporcionalidade, mostrando apenas três candidatos à classificação. No momento do segundo teste (teste da Área de Risco) somente irá permanecer a detecção do objeto do meio (pedestre), pois os outros se encontram fora da trajetória do veículo (fora da área de risco).

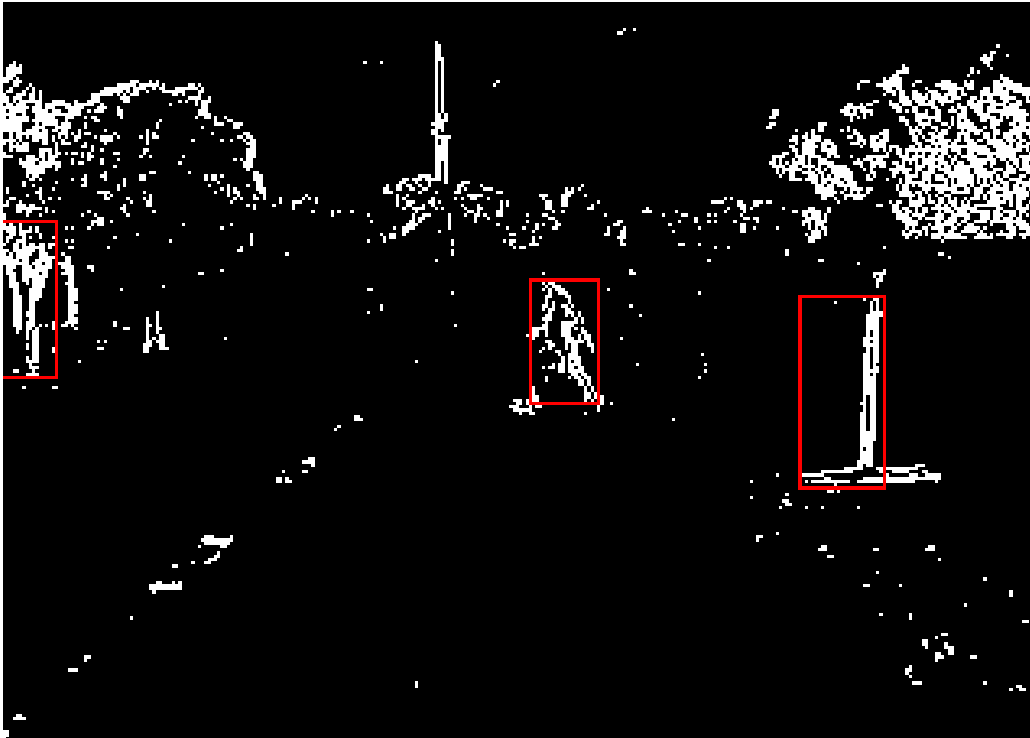


Figura 26 – Imagem com Blobs com Junção e filtro por proporcionalidade.

A área de risco foi determinada manualmente. Através da observação das imagens, é possível identificar o correto posicionamento da câmera. Cada câmera pode ter um ângulo de abertura de visão diferente, assim como a instalação da câmera no veículo pode direcionar o centro da imagem tendendo a um dos lados. A Figura 27 apresenta linhas verdes onde indicam que os objetos detectados entre estas serão classificados como possível pedestre.



Figura 27 – Exemplo da área de risco

3.2. Especificação da Segunda Versão

O sistema da segunda versão é projetado para ser construído em um único circuito integrado. Este conceito é conhecido como SoC (System-on-a-chip). O circuito integrado utilizado para a elaboração desse sistema embarcado é o FPGA (Field-Programmable Gate Array) (Bobda, 2007). Este dispositivo possibilita trabalhar com uma arquitetura reconfigurável, onde é possível executar algoritmos em hardware e software e tornar o sistema flexível para integrar novos módulos quando necessário (K. e Mourad, 1994) (Brown e Vranesic, 2000).

O banco de dados de imagens foi gerado a partir da conversão de um vídeo em figuras individuais (quadro-a-quadro) e gravadas em um cartão SD de memória. Através do barramento o NIOS II acessa o bloco de controle do cartão com portas de entrada/saída (PIO). Um programa no NIOS faz acesso ao sistema FAT do cartão e acessa as informações para serem usadas pelo programa de detecção de pedestres que é executado dentro do processador. O sistema então realiza operações de pré-processamento de imagem

(conversão de cores, filtro, estabilização, detecção de movimento e classificação). Em seguida, o NIOS II transfere as informações para a memória SDRAM onde um hardware específico transfere para um mostrador LCD. Um diagrama mais detalhado, contendo as implementações internas desse sistema é apresentado na Figura 28.

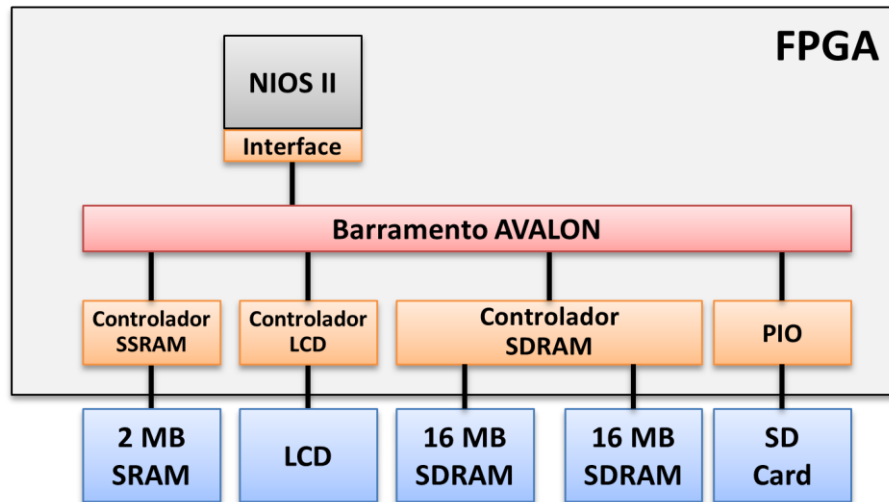


Figura 28 – Detalhamento da Versão 2

Para visualização da detecção, um display de LCD exibe a imagem captada pela câmera e destaca os objetos em movimento. Caso o objeto em movimento seja identificado como um pedestre em área de risco, a informação de detecção positiva será exibida.

Levando-se em consideração todo o processo de construção de uma arquitetura de hardware necessária ao reconhecimento de pedestres foi decidido, em um primeiro momento, trabalhar com filtros de imagem. Esta decisão pode ser justificada com base nos seguintes argumentos:

A aquisição de imagens pelo dispositivo FPGA é o primeiro passo para a construção do sistema, em seguida é possível criar outros blocos de processamento no sistema inicial gerado.

Seguindo a proposta de (Jalalian, 2008), é possível implementar o primeiro bloco gerador de região de interesse (ROI) usando algoritmo de detecção de contornos com filtro Sobel. A abordagem desse trabalho faz a diferenciação de quadros para identificar os elementos móveis da cena.

As principais características da Segunda Versão são:

- Realizar operações de aquisição e pré-processamento de imagem colorida através de um processador NIOS II em FPGA;
- Conversão do algoritmo utilizado na versão 1;
- Exibir imagem identificando objetos de interesse em um display LCD
- Possuir a características de Sistemas Embarcados e SoC.
- Fazer aferições de desempenho através de depuração remota.

3.3. Especificação da Terceira Versão

A terceira versão do sistema (Figura 29) compreende na construção de blocos em hardware dos mesmos elementos disponíveis em software. Nessa abordagem está incluso um bloco de ajuste de luminosidade (PID), já implementado através de um trabalho de mestrado de Rossi, D. et al (2011). Para o pleno funcionamento da versão 3, é necessário ainda a depuração do bloco detector de BLOB e a implementação do classificador.

Mais detalhes sobre os módulos de Estabilização estão dispostos no Apêndice B (no final deste trabalho)

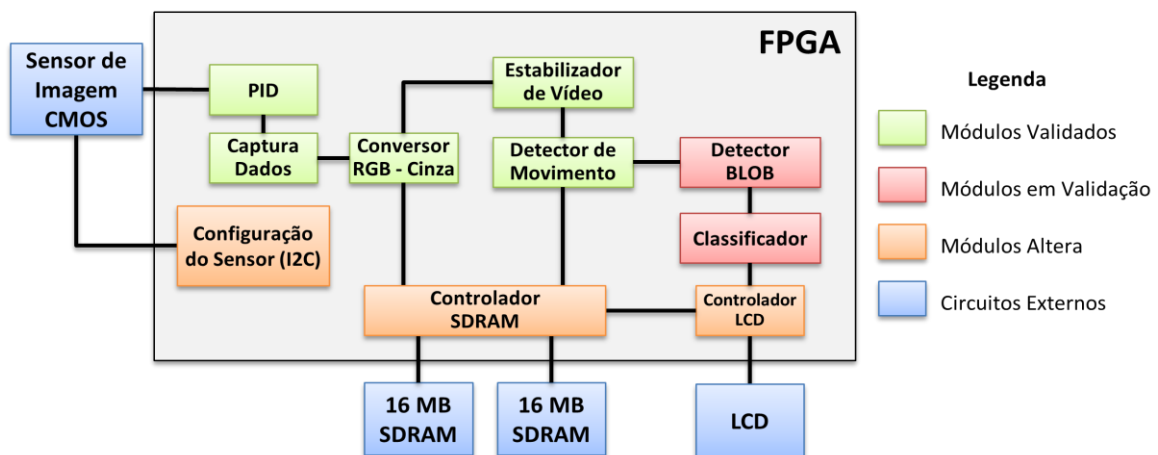


Figura 29 – Detalhamento da Versão 3

3.4. Contextualização do Sistema

O texto a seguir descreve os cenários e as limitações nos quais o sistema está inserido.

3.4.1. Restrições

A técnica utilizada para detectar pedestres restringe a área de sensibilidade à pista (rua) que está à frente do carro, essa abordagem evita que objetos que estão se movimentando ao redor (como árvores, postes, etc.) sejam detectados. Nessa restrição, os objetos em movimento contrastam suas cores apenas com o solo (asfalto), fazendo com que o sistema de detecção destaque os objetos em movimento. O pedestre que se encontra na área de risco será detectado mesmo quando não estiver se movimentando, pois estará em movimento em relação ao veículo. Esta abordagem implica que as intensidades do solo e do pedestre sejam suficientes para que o sistema funcione, ou seja, o sistema não detecta corretamente pessoas ou objetos que estão com as intensidades de cores próximas as cores do solo (roupas da mesma cor). O sistema de detecção de movimento usa um limiar de 8% (valor empírico) na variação da intensidade para considerar que houve movimento. Outra desvantagem deste sistema é que as imperfeições do solo (buracos, manchas, poças d'água, bueiros, obstáculos, etc.) serão detectados pelo sistema.

O sistema foi elaborado apenas para funcionar durante o dia e com uma iluminação difusa, ou seja, sem que a iluminação solar venha diretamente de encontro a câmera. Também não está preparado para situações adversas de clima como chuva, neve, neblina.

A velocidade máxima do veículo para os testes foi de 30 km/h, contudo, velocidades menores terão um maior fator de detecção, pois o pedestre fica maior causando destaque para as cores em movimento. Pedestres localizados a distâncias acima de 50 metros não são detectados, pois o sistema está calibrado para descartar pequenos objetos com intuito de eliminar falsas detecções. Devido às velocidades variáveis entre o veículo, o pedestre e a câmera, e também pelo fato do sistema não possuir um sistema de rastreamento (tracking), as detecções funcionam de forma intermitente e o pedestre não é totalmente acompanhado em sua trajetória.

O sistema de estabilização construído faz apenas a estabilização vertical, ou seja, contempla movimentos de subida ou descida da câmera. Este sistema está limitado a 20 linhas para cima ou para baixo de deslocamento, representando um valor total de 40 linhas de deslocamento. Não está disponível a estabilização horizontal, isto significa que o sistema

não está preparado para funcionar em curvas ou até mesmo para mudanças de faixa, ou seja, funciona apenas em linha reta. Seguindo a mesma abordagem, a proposta para suprir esta característica é a construção de um estabilizador de múltiplas direções, identificando como os pixels do quadro anterior se relacionam no quadro atual. Uma das possíveis soluções para isto é a utilização de fluxo óptico.

3.4.2. Considerações para Avaliação

Para validar o algoritmo de Estabilização de Imagem que foi construído, foram comparados os valores da saída (deslocamento vertical) com os valores de saída de outro programa de estabilização, mas que executa o procedimento de forma exaustiva, comparando assim com o resultado ideal (logicamente a busca exaustiva para chegar ao resultado ideal demandou um alto custo computacional).

A avaliação do classificador contempla as medições de tempo, taxa de acerto, falsos positivos ou falsos negativos e também uma comparação da utilização do algoritmo BLOB sem a junção dos blocos (Merge BLOB). Com o veículo em movimento, o pedestre inicialmente aparecerá pequeno na imagem e gradativamente seu tamanho aumentará, neste caso, a avaliação também identificará a partir de que distância a detecção torna-se mais eficiente.

A avaliação dos requisitos de Sistemas Embarcados apenas identificará os recursos necessários de Hardware e os tempos de execução de cada bloco. O resultado final da execução dos algoritmos é o mesmo, pois utilizam a mesma lógica e a mesma base de dados, tanto para o sistema embarcado quanto para o sistema em PC.

3.5. Ambientes de Desenvolvimento

O experimento foi conduzido em três plataformas diferentes. Primeiramente, a validação dos algoritmos foi executada em um computador PC Intel Pentium Core2Quad Q9300 (2,5 GHz) com 16GB RAM. A plataforma de desenvolvimento foi o Microsoft Visual Studio .Net (2008).

Após a conclusão das simulações do primeiro ambiente, o código fonte foi convertido para o processador NIOS II da Altera. Para isto, utilizou-se um kit de desenvolvimento da Terasic com o FPGA DE2-70 da Altera pertencente ao Laboratório de Computação Reconfigurável da USP (LCR). Para visualizar os experimentos foi utilizado juntamente ao Kit, um display LCD colorido (modelo TRDB-LTM) da Terasic e que permite a conexão com a placa de desenvolvimento. Alguns dos algoritmos foram convertidos para código Verilog para construção de blocos de Hardware. Neste caso foi utilizada a ferramenta Active-HDL da Aldec.

A placa DE2-70 (Figura 30) é projetada para uso em laboratórios, sendo apropriada para uma grande variedade de experimentos. Nesta placa foi utilizado o recurso de Memory Card (SD) para leitura das imagens do banco de dados do experimento. Este recurso foi essencial para a comprovação dos resultados obtidos com os executados no PC.

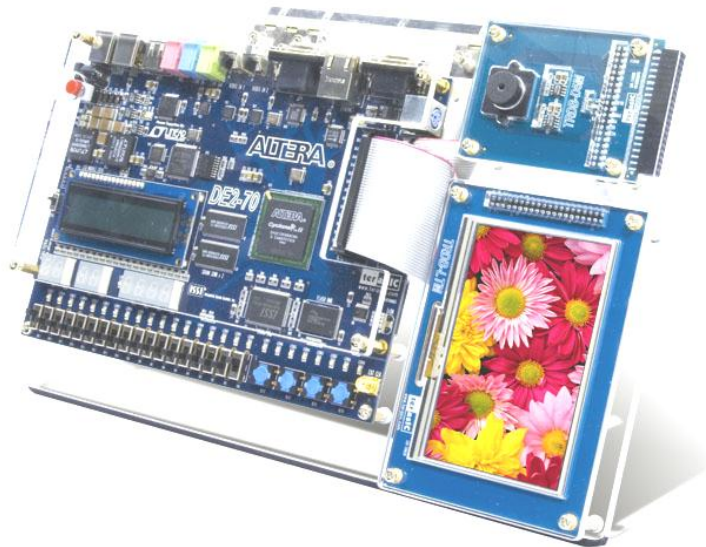


Figura 30 – Placa de Desenvolvimento Altera DE2-70

Dentre as características da placa DE2-70 da Altera pode-se destacar:

- 68.416 elementos lógicos;
- 250 blocos de RAM M4K;

- 1.152.000 bits de RAM
- 150 multiplicadores embutidos;
- 4 PLLs;
- 622 pinos de E/S programáveis;
- Encapsulamento FineLine BGA de 896 pinos.

3.5.1.O processador NIOS II da Altera

A arquitetura reconfigurável a ser utilizada na Versão 2 deste projeto, baseia-se no processador embarcado da Altera, o Nios II. Existem diversas pesquisas desenvolvidas para este processador, em destaque os trabalhos do LCR (Laboratório de Computação Reconfigurável da Universidade de São Paulo) que vem usando a tecnologia com muito entusiasmo (BONATO, et al., 2006), (SACCHETIN, et al., 2006), (MARQUES, et al., 2005). O Nios II é um processador com arquitetura *RISC (Reduced Instruction Set Computing)* de 32 bits altamente personalizável e um dos processadores *softcore* mais utilizados no mundo todo para se implementar *SoPC's* em *FPGA's*. (ALTERA CORPORATION, 2008). O conjunto de instruções do Nios II pode ser estendido e possibilita que os projetistas de sistemas adicionem lógica personalizada a sua unidade lógica aritmética (vide Figura 31). A utilização de instruções personalizadas visa acelerar trechos críticos de algoritmos em software convertendo-se para uma instrução totalmente em hardware (ALTERA CORPORATION, 2008).

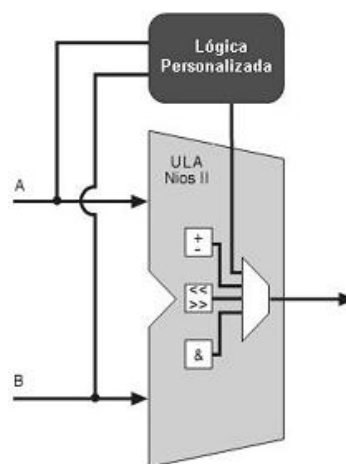


Figura 31 - Integração de instrução personalizada a ULA do Nios II

A ferramenta *SoPC Builder*, integrada ao *Quartus II*, disponibiliza uma interface gráfica e padronizada para criação processadores personalizados baseados no NIOS II associando memórias, periféricos padrão (portas de entrada e saída) e periféricos definidos pelo usuário. Desta forma os desenvolvedores focam-se mais na especificação do projeto lógico do que em construir a topologia do sistema (ALTERA CORPORATION, 2008).

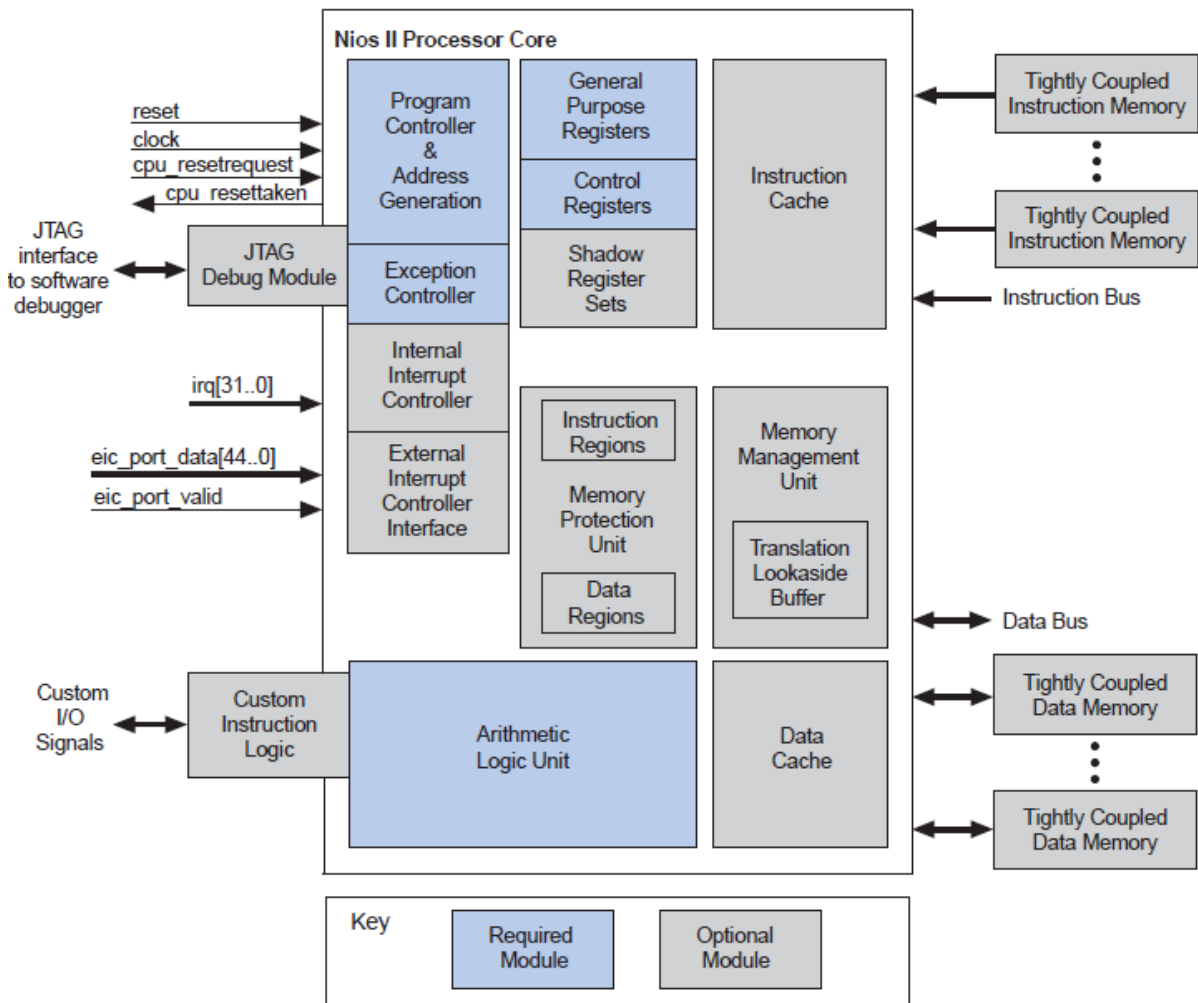


Figura 32 – Diagrama de Blocos do Núcleo do Processador Nios II

O projeto de hardware do processador II Nios disponibiliza um conjunto de instruções que suporta as unidades funcionais descritas na Figura 32. O núcleo do processador não inclui periféricos ou lógica de conexão com o exterior do chip, ele inclui apenas os circuitos necessários para implementar a arquitetura dentro do FPGA.

3.6. Base de Dados de Pedestres

Para obtenção de resultados mais próximos da realidade, a base de dados de imagem do experimento foi gerada em um ambiente urbano. Os vídeos foram capturados através de uma câmera digital Sony modelo Cyber-shot DSC-W90 com tecnologia Double Anti-blur.

Para a captura das imagens, a câmera foi instalada no teto de um veículo (aproximadamente 1,5 metros do solo), onde o veículo não aparece nas cenas, conforme a Figura 33. Foram capturadas diversas sequências de imagens com uma resolução de 640x480 pixels.

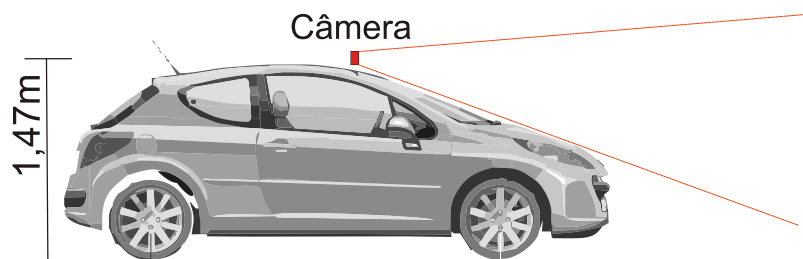


Figura 33 – Disposição da Câmera para Captura

Os testes foram elaborados em ruas pavimentadas (asfalto) e à luz do dia, sem contraposição ao sol, veja a Figura 34. Após a captura, os vídeos foram convertidos quadro-a-quadro para o formato bitmap, onde este é utilizado para os testes na placa de FPGA.

É importante observar como a dimensão (altura em pixels) do pedestre aumenta em relação aos quadros mais recentes. Estudos preliminares indicam que quando o pedestre está mais distante a detecção é menos eficiente. A escolha deste banco de imagens levou em consideração que o fundo da imagem não é uma continuação da rua. O fato de existir uma curva ao final implica que a detecção deverá ignorar o cenário de fundo, atendo-se apenas em detectar o pedestre em movimento. A captura ocorreu em um veículo trafegando a 30 Km/h, o asfalto é de boa qualidade e apresenta poucas imperfeições. A sinalização das faixas é clara.

Este experimento teve-se a testar apenas um pedestre por vez, não contemplando testes com múltiplas detecções simultâneas.

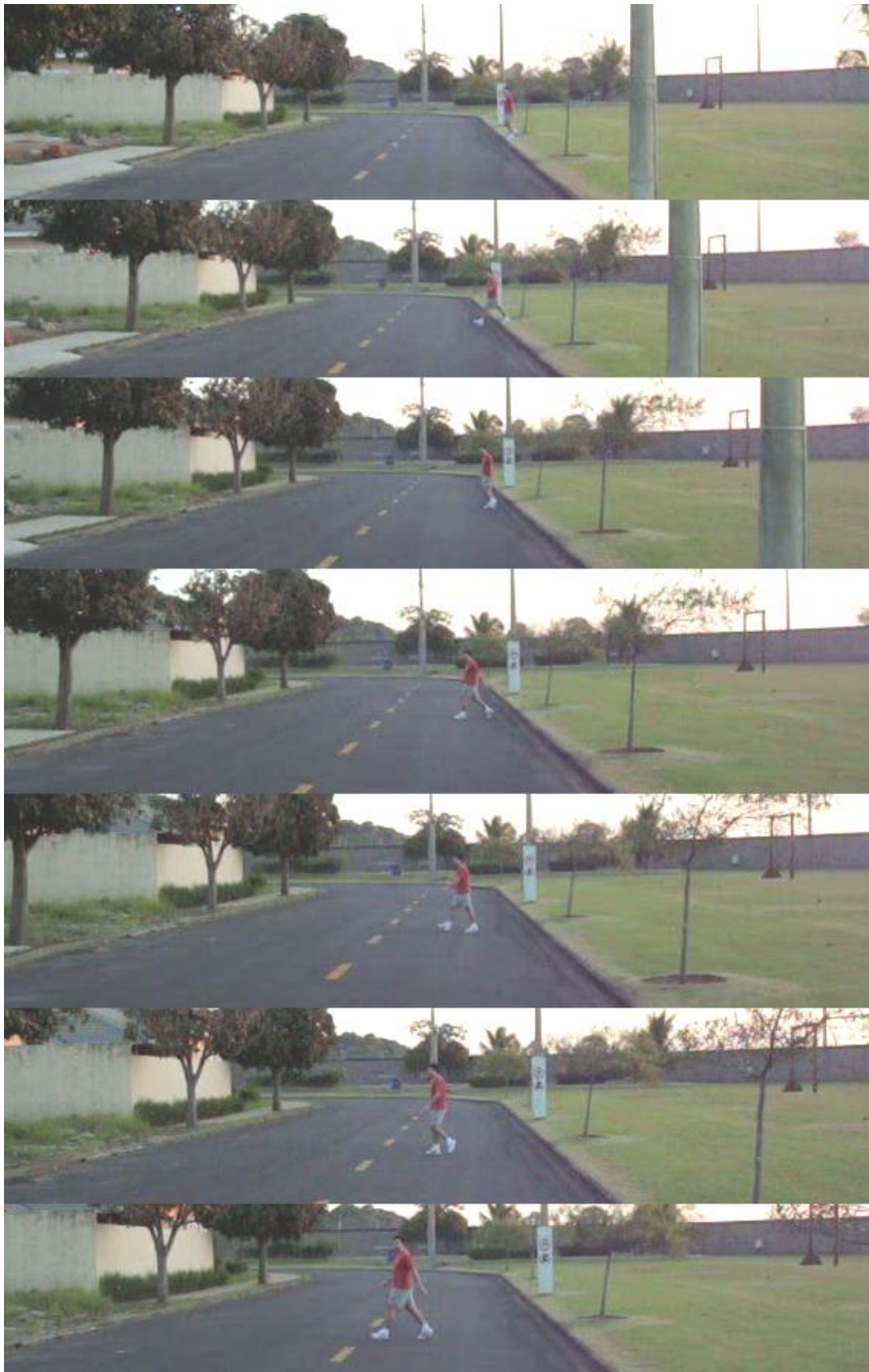


Figura 34 – Exemplo de Quadros Capturados

4. Resultados Obtidos com a Versão 2

Este capítulo apresenta a avaliação experimental de cada um dos os componentes do sistema. Primeiramente é avaliado o algoritmo de estabilização de imagem seguido pelo algoritmo de subtração de quadros, que inclui também um filtro de ruídos. Por fim, os métodos de classificação são aplicados, incluindo a extração de características através do detector BLOB (Binary Large Objects).

Neste trabalho são somente apresentados resultados de implementações da versão 1 (PC) e 2 (NIOS) pois a versão 3 (totalmente em Hardware) ainda encontra-se em fase de depuração.

4.1. Resultados da Detecção

O tempo de execução médio do sistema de detecção, avaliado através de várias sequências é apresentado na Tabela 4:

Tabela 4 – Resultados do Experimento

Algoritmo	Versão 1 Intel Core2Quad	Versão 2 Altera NIOS II
Estabilização	0,45 ms	825 ms
Detecção de Movimento	2488 ms	2435 ms
BLOBs	109 ms	196 ms
Classificação	17 ms	23 ms
Total	2614,45 ms	3479 ms
Quadros por Segundo	0,382	0,287

A forma de se obter esses dados foi através da inserção de um código marcador de tempo dentro do programa. Antes de iniciar cada bloco de execução o marcador era inicializado com um registro de tempo (*timestamp*). Ao final da execução o marcador era finalizado. O tempo de cada bloco foi calculado subtraindo-se o tempo registrado no segundo marcador menos o do primeiro. Após sucessivos testes calculou-se a média de tempo de cada bloco. No caso do processador NIOS II, este possui uma instrução especial para trabalhar com o temporizador trabalhando ao final de forma análoga ao computador.

Os resultados permitem uma análise preliminar indicando que o gargalo do sistema é o bloco de Detecção de Movimento, mas este bloco também é responsável por armazenar a imagem no *buffer* de memória através do barramento Avalon. Os testes iniciais na versão 3 demonstraram que o tempo equivalente para um módulo totalmente em hardware é muito inferior as versões 1 e 2. A detecção de movimento teve um aumento significativo da versão 1 para versão 2, isto deve-se ao fato que na conversão do código do framework.net para a linguagem c do NIOS houve uma conversão do objeto image para um vetor de memória e assim causando grande melhoria de desempenho.

Os resultados da implementação da versão 2 no FPGA utilizando-se o processador NIOS II estão descritos na Tabela 5:

Tabela 5 – Resultado da Implementação da Versão 2

Total de Elementos Lógicos	5928
Total de Funções Combinacionais	5098
Registradores Lógicos Dedicados	3282
Total de Registradores	3617
Total Memória Binária	266176
Multiplicadores Embarcados	4
Total PLLs	2

O número de elementos lógicos necessários para o sistema da versão 2 equivale a apenas 7% da capacidade do FPGA utilizado.

4.2. Estabilização da Imagem

O método para avaliar o sistema de estabilização, foi através da comparação com um segundo método que faz a estabilização de forma exata. Este segundo método faz sucessivas subtrações de dois quadros subsequentes deslocando os quadros até 20 linhas para cima e 20 linhas para baixo. Este número foi definido de forma empírica restringindo-se o deslocamento máximo que o sistema pode captar. Nos experimentos em ambiente real

observou-se que raramente esse deslocamento ultrapassa o número de 10 linhas, conforme apresentado na Figura 35.

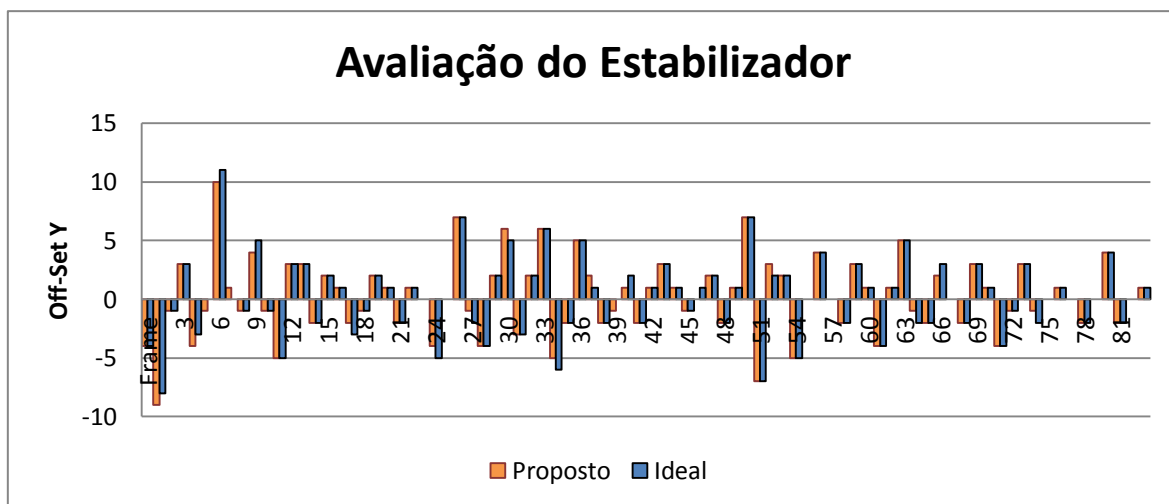


Figura 35 – Avaliação do Estabilizador

A comparação dos dois métodos demonstra que o algoritmo que foi construído retorna resultados muito semelhantes. A taxa de erro média foi de 0,27% em comparação com o método exato. O Estabilizador deste projeto faz somente ajustes de deslocamento vertical, mas sua implementação permite que seja facilmente convertido para fazer a estabilização horizontal. O gráfico mostra no eixo vertical o deslocamento necessário para que o quadro atual fique alinhado com o quadro anterior.

4.3. Detecção de Movimento

A detecção de movimento obteve menor desempenho se comparado aos outros módulos. Apesar da baixa complexidade, o algoritmo tem que passar por todos os pontos da imagem para criar uma matriz binária com os pontos de movimento e transferir as informações para o buffer de memória de imagem. O algoritmo faz a subtração ponto a ponto da imagem do quadro atual com o quadro anterior, utilizando o fator de deslocamento vertical para identificar a linha inicial. Os resultados são apresentados no gráfico da Figura 36. Este gráfico apresenta como eixo vertical o tempo em segundos e no eixo horizontal o número do quadro.

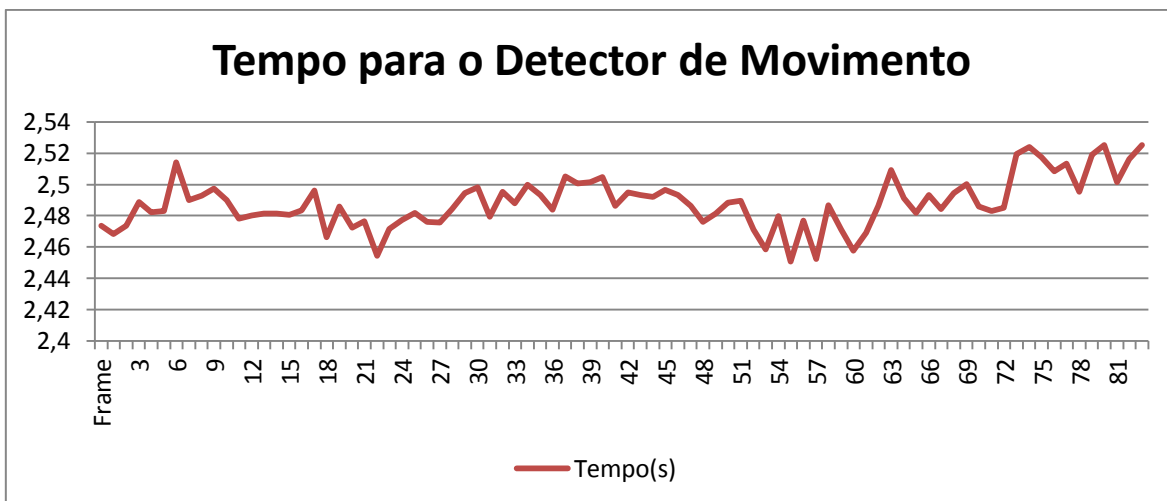


Figura 36 – Avaliador do Tempo para Detecção

4.4. Extração de Características

O algoritmo de detecção de Objetos Binários (BLOB) executa sua operação de forma serial, ou seja, é necessário percorrer toda a imagem para obter os objetos. Em segunda etapa, os objetos BLOB são agrupados, fazendo-se a junção de blocos que estão sobrepostos. Os resultados de tempo de execução para cada quadro são mostrados na Figura 37. O gráfico apresenta os tempo para gerar os Objetos Binários, os tempos para fazer a junção de objetos (Merge) e a soma destes dois tempos (Total).

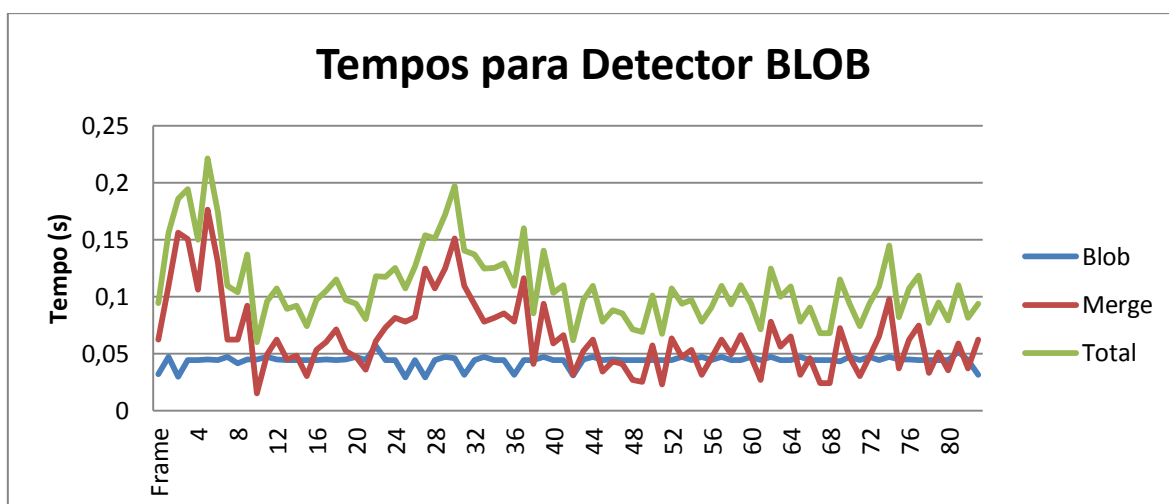


Figura 37 – Avaliação dos Tempos para Detectar BLOB

4.5. Classificação

Uma vez que os BLOBs já foram detectados, o Sistema de Classificação precisa apenas verificar se existem objetos na área de risco. A vantagem desta técnica é que não é necessário fazer o processamento da imagem toda (somente da área de risco), proporcionando um desempenho superior na execução geral do sistema. A técnica utilizada para detectar pedestre foi a proporcionalidade de tamanho dos BLOBs, ou seja, blobs muito pequenos ou muito grandes são descartados. Mesmo com essa abordagem o detector pode captar vários objetos além de pedestres, em contrapartida, o sistema vai detectar qualquer objeto (de tamanho médio) que esteja em na rota de colisão, inclusive pedestres. A Figura 38 mostra os respectivos tempos para classificar de cada quadro. O eixo vertical mostra a escala de tempo em segundo enquanto o eixo horizontal mostra os quadros.

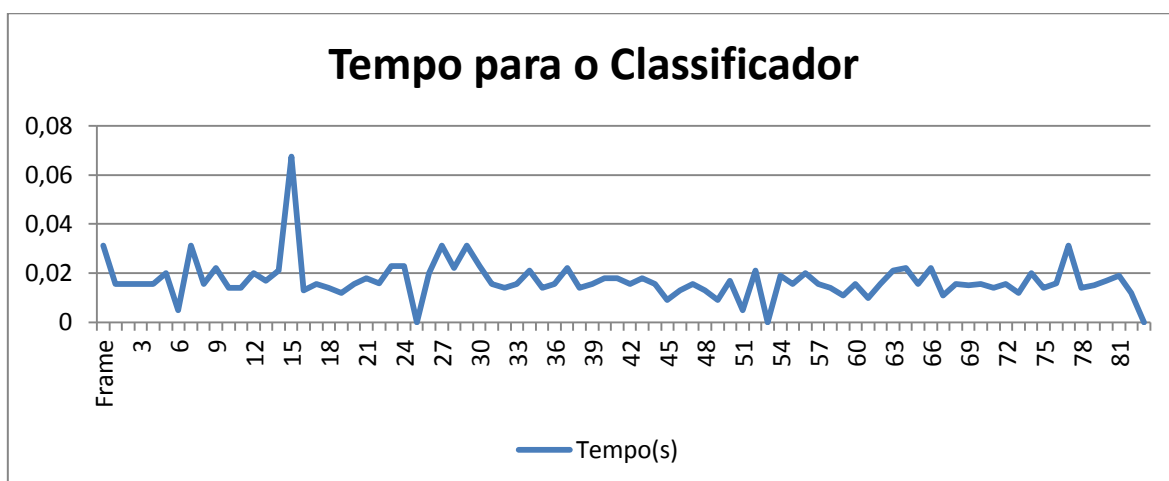


Figura 38 – Avaliação do tempo do Classificador

A Figura 39 mostra o gabarito onde, quando o valor =1 o pedestre está sobre a área de risco. As linhas em vermelho indicam os quadros onde ocorreram erros de detecção. Verifica-se que nos quadros iniciais, onde o tamanho do pedestre na imagem ainda é pequeno, houveram vários falsos positivos.

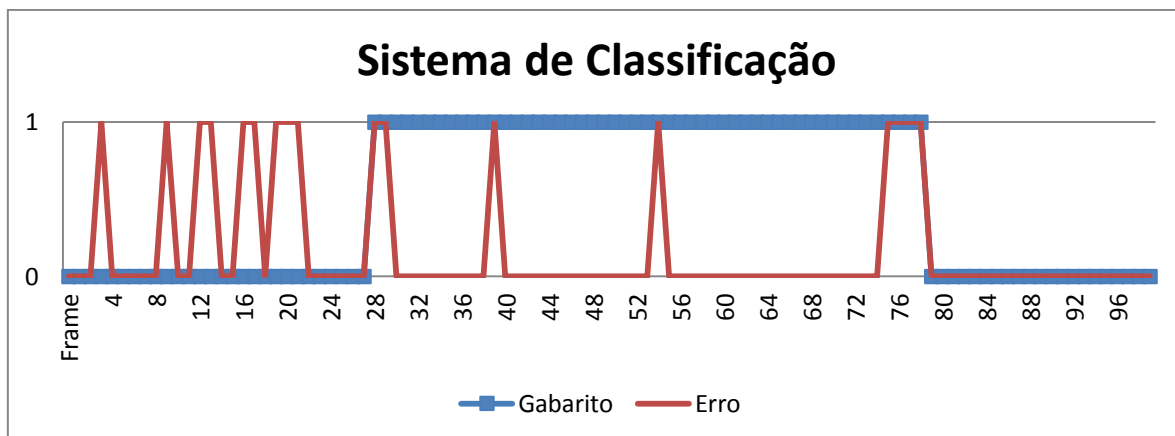


Figura 39 – Avaliação do sistema de Classificação

A taxa média de erros da amostragem é de 17% e a taxa média de falsos positivos é de 9%. Portanto, aproximadamente metade dos erros detectados foi de falsos positivos. Estes resultados indicam que as primeiras detecções tem maior incidência de erros, isto ocorre principalmente devido a distância do pedestre, onde este tem tamanho menor na imagem.

5. Conclusão

A conversão do Sistema de Cores para Escala de Cinza no pré-processamento da imagem é uma operação que em alguns casos o próprio sensor de imagem dispõe diretamente em uma de suas saídas, através da informação de intensidade do pixel. Em todo caso, este algoritmo mostrou-se eficiente e permite que a conversão seja feita em paralelo com o carregamento da imagem, ou seja, para cada pixel lido é possível convertê-lo e em seguida enviá-lo ao buffer de carregamento.

O sistema de Estabilização demonstrou ser suficiente para atender aos requisitos de desempenho. Uma das grandes vantagens dessa abordagem refere-se a sua simplicidade de construção e a pequena taxa de erro (menor que 1%) quando comparada a um algoritmo de busca exaustiva. A execução do mesmo experimento sem o sistema de estabilização de imagem causou um incremento da taxa de erro médio de 17% para 40%. Portanto, verifica-se que o sistema de estabilização foi essencial para resultados mais precisos.

A taxa de acertos médios do classificador foi de 83% e quando comparados com outros trabalhos verifica-se que o sistema enquadra-se na média dos sistemas de detecção de pedestres. Contudo, o tempo médio de execução de todos os módulos no FPGA foi de 3,5 segundos para cada frame, o que inviabiliza seu uso em tempo real quando usado com somente um processador. O maior tempo recomendado seria de 0,03 segundos que equivale a 30 frames por segundo. Ou seja, o sistema da versão 2 opera em média com 0,286 frames por segundos. Uma das possíveis soluções para o problema é a construção da versão 3 deste projeto (versão totalmente em Hardware). Outra possível solução é aumentar o número de processadores NIOS II. Nota-se que o sistema utilizou apenas 7% da capacidade do FPGA e construindo-se processadores NIOS II em paralelo é possível executar o mesmo experimento aproximadamente dez vezes mais rápido. Nesta placa ainda não seria suficiente, mas o problema se resolve caso utilizado com FPGAs mais modernas, conforme apresentado na Tabela 6:

Característica	Virtex 7 XC7V2000T	Stratix V 5SEEB	Altera DE2-70	Projeto Versão 2
CLBs	1.954.560	952.000	68.416	5.928
Estimativa de aumento de Desempenho	X329,7	X160,6	X11,5	X1

Tabela 6 – Projeção do Sistema

Os testes preliminares na versão 3 (apenas até blocos de detecção de movimento) mostraram resultados muito satisfatórios. Usando a lógica de das outras versões, esta apresentou uma taxa acima de 30 quadros por segundo.

Observa-se que o gargalo da versão 2 está justamente na detecção de movimento. Neste caso, é possível utilizar o bloco de detecção da versão 3 integrado com o processador NIOS II para ter um resultado ótimo.

5.1. Contribuições deste Trabalho

De um modo geral, este trabalho contribuiu para o desenvolvimento de um sistema anti-colisão em uma plataforma de Computação Reconfigurável. Este protótipo poderá servir de base de outros sistemas autônomos e ou detecção de pedestres com carros em velocidades superiores a 30 Km/h.

Os grupos de pesquisas da USP do Laboratório de Computação Reconfigurável (LCR) e do Laboratório de Robótica Móvel (LRM), que atuam nas áreas de Sistemas Embarcados, Evolutivos e Robóticos, têm buscado tecnologias no 'estado da arte', e onde possam integrar aspectos de segurança e comodidade para o motorista. Dessa forma, esse trabalho vem a contribuir com alternativas de construção de sistemas de detecção para a Robótica.

Em resumo as contribuições deste trabalho foram:

- Desenvolvimento de um Sistema Digital de Estabilização Vertical de imagem em Software e em Hardware

- Desenvolvimento de um sistema embarcado para validação de técnicas de detecção de pedestres baseado no processador softcore NIOS II da Altera.
- Proposta do mesmo sistema implementado numa versão totalmente em Hardware (sem o uso do processador NIOS II)

5.2. Trabalhos Futuros

É necessário a conclusão da última versão deste projeto (versão 3), que será totalmente em Hardware, tendo o benefício de operar com paralelismo de algumas funções.

Um dos limitadores do sistema é o não funcionamento em curvas ou deslocamentos laterais. Para atender a este requisito, é necessário complementar o sistema de estabilização de imagem para atender ao deslocamento horizontal da imagem.

Um importante elemento para completar este trabalho é a construção do sistema de rastreamento (*Tracking*). Pelo fato do bloco de classificação não detectar corretamente todos os quadros, o sistema identifica situação de risco de forma intermitente. Como a velocidade de resposta em tempo real é alta, um sistema de rastreamento que armazene a última posição do pedestre melhoraria de forma sensível a resposta do sistema.

Uma interessante área relacionada ao trabalho é o Reconhecimento de Pedestres. Normalmente isso é feito através de redes neurais, pelo reconhecimento da forma e pelo reconhecimento do tipo de movimento.

Em um ambiente real, existem situações que devem ser tratadas e que dificultam a detecção deste sistema. Uma destas situações são as Faixas de Travessia de Pedestres. Como a cor branca contrasta com o asfalto, o sistema irá classificá-lo como um objeto.

Outro trabalho importante é modificar os algoritmos para uso em sistemas multiprocessados.

Alguns testes foram realizados utilizando a câmera TRDB-LTB (Terasic) diretamente na placa DE2-70 e tendo acesso via barramento ao NIOS II, os estudos preliminares indicam que

esta construção pode ser útil para testar a placa em um ambiente real assim como foi o caso da câmera digital neste projeto. Mais detalhes podem ser visualizados no Apêndice C.

Foi verificado que uma solução intermediária (entre a versão 2 e a 3) pode ter bons resultados. Esta proposta consiste em trocar o módulo mais lento (detecção de movimento) por um circuito equivalente em hardware e fazer o acesso pelo NIOS através do barramento ou através de instruções customizadas.

Referência Bibliográfica

- ALIX, R., COAT, F. L., AND AUBERT, D. (2003). Flat world homography for non-flat world on-road obstacle detection. In Proceedings of IEEE Intelligent Vehicles Symposium, pag 310–315, Columbus, USA..
- ALONSO, M. ,GARAYO,P., HERRÁN, L. (2005). Defining Safe Speed and Safe Distance towards improved longitudinal control using ADAS: Functional Requirements of the SASPENCE system. 2005.
- ALTERA CORPORATION. (2008). Introduction to the Quartus II Software. [Online] Outubro de 2008. http://www.altera.com/literature/manual/intro_to_quartus2.pdf.
- ALTERA CORPORATION. (2008). Nios II Custom Instruction User Guide. [Online] Outubro de 2008. http://www.altera.com/literature/ug/ug_nios2_custom_instruction.pdf.
- ALTERA CORPORATION. (2008). Nios II Processor Reference Handbook. [Online] Outubro de 2008. http://www.altera.com/literature/hb/nios2/n2cpu_nii5v1.pdf.
- ALTERA CORPORATION. (2008). Nios II Software Developer's Handbook. [Online] Outubro de 2008. http://www.altera.com/literature/hb/nios2/n2sw_nii5v2.pdf.
- ÁLVARES, L.O. E SICHMAN, J. (1997). Introdução aos Sistemas Multiagentes. XVII Congresso da SBC - Anais JAI'97. Brasília, DF. 1997.
- ARAGÃO, R. F. (2003) Acidentes de Trânsito aspectos técnicos e jurídicos. 3ª ed. Campinas: Millennium.
- ASSUMPCAO, J. WOLF, D. MARQUES, E., (2007) Towards a hardware accelerated obstacle avoidance system for mobile robots using monocular vision, in IEEE Second International Symposium on Industrial Embedded Systems - SIES, pp. 365_368.
- BEAUVAIS, M. AND LAKSHMANAN, S. (2000). Clark: a heterogeneous sensor fusion method for finding lanes and obstacles. Image and Vision Computing, pag 397–413.
- BECKER, J. SIMON, A., SÖHNITZ, I., GÖLLINGER, H., SCHUMACHER, W. (1998). A Decentralized Path Planning and Control Structure for an Autonomous Vehicle. In Proceeding of the IV '98. IEEE International Conference on Intelligent Vehicles.
- BERTOZZI, M., BROGGI, A., CELLARIO, M., FASCIOLI, A., LOMBARDI, P., AND PORTA, M. (2002). Artificial vision in road vehicles. Proceedings of the IEEE. 2002 pag 1258–1271.
- BOBDA, C. (2007) Introduction to reconfigurable computing - architectures, algorithms and applications. Dordrecht, The Netherlands: Springer, pag 375.

BONATO, V., HOLANDA, J. A. E MARQUES, E. (2006). An Embedded Multi-Camera System for Simultaneous Localization and Mapping. International Workshop on Applied Reconfigurable Computing (ARC 2006).

BONATO, V. (2004) “Projeto de um módulo de aquisição e pré-processamento de imagem colorida baseado em computação reconfigurável e aplicado a robôs móveis” – Defesa e Dissertação Apresentada ao Programa de Pós-Graduação em Ciência da Computação do ICMC-USP para obtenção do título de Mestre, São Carlos, S.P..

BREGLER, C. (1997) “Learning and recognizing human dynamics in video sequences”, Proc. IEEE Conf. Comp. Vision and Pattern Recognition, San Juan, Puerto Rico, 1997.

BROWN, S.; VRANESIC, Z. (2000) Fundamentals of digital logic with vhdl design. Toronto: Mc Graw Hill.

CHUANJIN, L., XIAOHU, Q., XIYUE, H., YI, C., AND, Z. X. (2003). A monocular-vision-based driver assistance system for collision avoidance. In Proceedings of IEEE International Conference on Intelligent Transportation Systems, pag 463–468.

COLLADO, JM., HILARIO C., ARMINGOL, J., DE LA ESCALERA, A. (2003) “Visión por Computador para Vehículos Inteligentes”, XXIV Jornadas de Automática, León, España.

DEMONCEAUX, C. AND KACHI-AKKOUCHE, D. (2004). Robust obstacle detection with monocular vision based on motion analysis. In Proceedings of IEEE Intelligent Vehicles Symposium.

DOLLAR, P., WOJEK, C., SCHIELE, B., AND PERONA P. (2011). Pedestrian Detection: An Evaluation of the State of the Art. IEEE Transactions on Pattern Analysis and Machine Intelligence

DUDEK, G. AND JENKIN, M. (2000). Computational Principles of Mobile Robotics. Cambridge University Press, Cambridge, UK.

DUGARRY, A. (2004). PhD Thesis Cranfield University School of Engineering Applied Mathematics and Computing Group. Advanced Driver Assistance Systems Information Management and Presentation.

FASCIOLI, M. B. A. AND BROGGI, A. (1999). Performance analysis of a low-cost solution to vision-based obstacle detection. In Proceedings of IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, pag 350–355.

FU, K. S., GONZALEZ R. C., LEE, C. S. G. (1990) “Robotics: Control, Sensing, Vision and Intelligence”, McGraw-Hill, pag 1-10.

FRITSCH, J., MICHALKE, T., GEPPERTH A., BONE S., WAIBEL F., KLEINEHAGENBROCK M., GAYKO J., GOERICK, C. (2008). “Towards a Human-like Vision System for Driver Assistance” 2008

IEEE Intelligent Vehicles Symposium Eindhoven University of Technology Eindhoven, The Netherlands, Junho 2008

GARCIA, A. C. B. E SICHMAN, J. (2003). Agentes e Sistemas Multi Agentes. Sistemas Inteligentes: Fundamentos e Aplicações (Coord. Solange Rezende). SP : Manole.

GAVRILA, D., GIEBEL, J., AND MUNDER, S. (2004). Vision-based pedestrian detection: the protector system. In Proceedings of IEEE Intelligent Vehicles Symposium. 2004.

GAVRILA D., MUNDER S. (2006). "Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle" International Journal of Computer Vision Springer Science

OTTONI, G. L., LAGES, W. F. (2003). Navegação de Robôs Móveis em ambientes desconhecidos utilizando sonares de ultra-som. Revista Controle & Automação. 2003, Vol. 14, 4.

GONZALEZ, R. C., WOODS, R. E., (2000) "Digital Image Processing", Prentice Hall; 2nd edition.

HEINEN, FARLEI J. (2002). Sistema de Controle Híbrido para Robôs Móveis Autônomos. Unisinos – PIPCA – Dissertação de Mestrado em Computação Aplicada. 2002.

HORN, B. K.; SCHUNCK, B.G.(1980) Derermining optical flow. Relatório técnico AIM 572, Cambridge, MA, USA.

IMAPCAR (2011) http://www.nec.com/global/onlinetv/en/society/imapcar_l.html acesso em Novembro 2011.

WANG, P. H., COLLINS, J. D., WEAVER, C. T., KUTTANNA, B. SALAMIAN, S., CHINYA, G. N., SCHUCHMAN, E., SCHILLING, O., DOIL, T., STEIBL, S., WANG, H. (2009) Intel ® Atom™ Processor Core Made FPGA-Synthesizable. FPGA '09 Proceeding of the ACM/SIGDA international symposium on Field programmable gate arrays.

JALALIAN, A. , FATHY, M. (2008). Pedestrian Detection from a Moving Camera with an Advanced Camera-Motion Estimator. Third International IEEE Conference on Signal-Image Technologies and Internet-Based System.

JANTA, J., KUMSAWAT, P., ATTAKITMONGCOL, K., SRIKAEW, A. (2008) Pedestrian Detection using Color Symmetry Phases. Robotics & Automation Research Unit for Real-World Applications, Intelligent System Group School of Electrical Engineering Suranaree University of Technology

JIANG, G.Y., CHAI, T.Y., HANG, S.K., BAE, W., SONG, B.S. (2000) "Lane and obstacle detection based on fast inverse perspective mapping algorithm." In: IEEE International Conference on Systems, Man and Cybernetics. Vol. 4.

- K., P.; MOURAD, S. (1994) Digital design using field-programmable gate arrays. Prentice Hall, 1994.
- KASTRINAKI, V., ZERVAKIS, M., AND KALAITZAKIS, K. (2003) A survey of video processing techniques for traffic applications. *Image and Vision Computing*, pag 359–381..
- KELBER, C. R., JUNG, C. R., HEINEN, F. J., OSÓRIO, F. S. (2005). *Computação Embarcada: Projeto e Implementação de Veículos Autônomos Inteligentes*, XXV Congresso da Sociedade Brasileira da Computação – SBC, pag 1358 - 1406.
- KELBER, C.R., ET AL. (2004). Active Steering Unit with integrated ACC for X-by-Wire vehicles using a joystick as H.M.I.. In *Proceeding of the 2004 IEEE Intelligent Vehicles Symposium*; Parma, Itália, pag 14-17.
- KELBER, C.R., ET AL. (2005). *Industrial Electronics, 2005. ISIE 2005. Proceedings of the IEEE International Symposium on Volume 4*, Pag 1623 - 1628.
- KIRISCHIAN, V., GEURKOV, V., KIRISCHIAN, L. (2008). Cost Effective Reconfigurable Architecture for Stream Processing Applications. *CCECE/CCGEI Niagara Falls. Canada 2008 IEEE*. 2008.
- KUANG, P., ZHU, Q., LIU, G., (2004) “Real-time road lane recognition using fuzzy reasoning for AGV vision system.” In: *Communications, Circuits and Systems, ICCAS 2004. International Conference. Vol. 2*, pag 989-993.
- LRM (2011) Laboratório de Robótica Móvel – Projeto CaRINA <http://lrm.icmc.usp.br/wiki/index.php/CaRINA>, acesso em Novembro 2011
- MAES, P. (1994). *Modelling Adaptive Autonomous Agents*. *Artificial Life Journal*. Edited by C. Langton, Vol. 1, No. 1 & 2, pag 135-162, MIT Press. 1994, Vol. 1.
- MARQUES, E., BONATO, V. E FERNANDES, M. M. (2005). A Fault Tolerant Gesture Recognition System for Mobile Robot. *International Workshop on Applied Reconfigurable Computing (ARC 2005). Proceedings of the International Workshop on Applied Reconfigurable Computing 2005*.
- MATARIC, MAJA. (2007). *The Robotics Primer*. Cambridge MA, London – England : MIT Press, 2007.
- MCCALL, J.C. E TRIVEDI, M.M. (2006). Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *Intelligent Transportation Systems, IEEE Transactions on Intelligent Transportation Systems*. 2006, Vol. 7, 1.
- MEDEIROS, ADELARDO A.D. (1998). A Survey of Control Architectures for Autonomous Mobile Robots. *JBCS - Journal of the Brazilian Computer Society, special issue on Robotics*. 1998, Vol. 4, 3.

MobileEye (2011) <http://www.mobileeye.com> acesso em Novembro de 2011

PARK, J.W., LEE, J.W., JHANG, K.Y. (2003) "A lane-curve detection based on an LCF." In: Pattern Recognition Letters, Vol. 24, Issue 14, pag 2301-2313

PREVENT (2011) <http://www.prevent-ip.org> acesso em Novembro de 2011

RAMESH, J.; RANGACHAR, K.; SCHUNCK, B. G. (1995) Machine vision. New York: McGraw-Hill.

RENAEST (Registro Nacional de Acidentes e Estatísticas de Trânsito). (2008). DENATRAN. Estatísticas do Sistema RENAEST 2007. [Online] Agosto de 2008. <http://www2.cidades.gov.br/renaest/detalheNoticia.do?noticia.codigo=245>.

ROSSI D., LIMA J., BONATO V. (2011) "A PID Controller Applied to the Gain Control of a CMOS Camera Using Reconfigurable Computing", to appear at Reconfig 2011 Conference

SACCHETIN, MARCELO CARVALHO, ET AL. (2006). Analysis and Implementation of Algorithms for Localization and Mapping of Mobile Robots Based on . II Southern Conference on Programmable Logic (SPL2006).

SENA (2011) <http://www.eesc.usp.br/sena/url/pt/index.php> acesso em Novembro 2011.

SIMON, A. E BECKER, J. (1999). Vehicle Guidance for an Autonomous Vehicle. In Proceeding of the IEEE International Conference on Intelligent Transportation Systems.

SIMON, A., SÖHNITZ, I., BECKER, J., SCHUMACHER, W. (2000). Navigation and Control of an Autonomous Vehicle. In Proceeding of the 9th IFAC Symposium on Control in Transportation Systems.

STEIN, G. P., MANO, O., AND SHASHUA, A. (2000). A robust method for computing vehicle ego-motion. In Proceedings of IEEE Intelligent Vehicles Symposium.

STEIN, G. P., MANO, O., AND SHASHUA, A. (2000). Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE Pag 362-368.

Stiller, C., Hipp, J., Rössig, C., and Ewald, A. (2000). Multisensor obstacle detection and tracking. Image and Vision Computing, pag 389-396.

TECHMER, A. (2007). Application Development of Camera-based Driver Assistance Systems on a Programmable Multi-Processor Architecture. Proceedings of the 2007 IEEE Intelligent Vehicles Symposium Istanbul.

Terasic (2010) <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=53&No=296> acesso em Abril de 2010

THOMANEK, F., DICKMANN, E. D., AND DICKMANN, D. (1994). Multiple object recognition and scene interpretation for autonomous road vehicle guidance. In Proceedings of IEEE Intelligent Vehicles Symposium.

THRUN, SEBASTIAN. (2001). Is Robotics Going Statistics? The Field of Probabilistic Robotics. Communications of the ACM.

TOCHETTO, D.; SPÍNDULA, A. (2005) Criminalística procedimientos e metodologia. 1ª ed. Porto Alegre: [s.n].

TSUGAWA, S. (1994) "Vision-based vehicles in Japan: machine vision systems and driving control systems", IEEE Transactions on Industrial Electronics, Vol. 41, Issue 4, pag 398-405.

WISCHOFF, L., ET AL. (2003). SOTIS - A self-organizing traffic information system. Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual. 2003, Vol. 4

WONG, S. M. AND XIE, M. (1999) "Lane geometry detection for the guidance of smart vehicle," In Proceeding . IEEE Int. Conf. Intelligent Transportation Systems, 1999, pag 925–928

Glossário

ADAS - Advanced Driver Assistance Systems

ABS - Anti-lock Braking System

DSP – Digital Signal Processor

EDA – Eletronic Design Automation

FPGA – Field Programmable Gate Array

GPS – Global Positioning System

LASER – Light Amplification by Stimulated Emission of Radiation

MuVE - Multiprocessor Vision Engine

RADAR - Radio Detection And Ranging

RENAEST – Registro Nacional de Acidentes e Estatísticas de Trânsito

RISC – Reduced Instruction Set Computing

RMAs - Robôs Móveis Autônomos

SoPC – System on Programmable Chip

ULA – Unidade Lógica Aritmética

Apêndice A

A. Veículos Inteligentes

A1. Sistemas Avançados para Auxiliar Motoristas

A criação de veículos inteligentes proporciona ao condutor uma vantagem em segurança, mas pode também auxiliá-lo em várias tarefas cotidianas, como por exemplo: dirigir na estrada mantendo-se centralizado na faixa correta, manter automaticamente uma distância segura entre veículos, controlar a velocidade conforme as condições do trânsito, fazer ultrapassagens seguras, encontrar um caminho mais curto para o destino ou estacionar facilmente em qualquer vaga no ambiente urbano.

Com objetivos cada vez mais desafiadores em melhorar o conforto, aumentar a segurança, permitir maior estabilidade e melhorar o rendimento de consumo dos veículos automotores, a área industrial tem investido intensamente em sistemas eletrônicos embarcados. Para auxiliar na condução de veículos, diversas soluções inovadoras estão sendo desenvolvidas nos últimos anos, algumas das quais já são consideradas comuns e assim muitas vezes não são notadas pelo usuário, como o caso de freios ABS, computadores de bordo, controle de direção, injeção eletrônica, sistema de estabilização e os sistemas de localização via GPS.

Com a evolução da automação, diversos sistemas mecânicos são substituídos por soluções eletromecânicas, criando um novo estágio para as soluções tecnológicas, chamadas de *drive-by-wire*¹ (C.R. Kelber, 2005).

A tecnologia *drive-by-wire* consiste em uma unidade de controle estruturada em níveis hierárquicos de controle, onde se destacam os Sistemas Mecatrônicos Embarcados e Sistemas de Apoio ao Motorista, desenvolvidos a partir de soluções de computação aplicada.

A interface de comunicação entre o condutor e o veículo automatizado necessita de estudos aprofundados, pois, com a tecnologia *drive-by-wire*, o carro pode ser guiado a partir de comandos eletrônicos, como por exemplo através de *joystics*, sem a necessidade da interface convencional de volante ou pedais (C. R. Kelber, 2004).

Os sistemas avançados destinados ao apoio de motoristas (ADAS) são classificados em cinco categorias (Dugarry, 2004): sistemas de controle lateral; sistemas de controle longitudinal, assistência a estacionamento; sistemas avançados de visão e sistemas de adaptação de velocidade. São exemplos desses sistemas:

Sistemas para identificação de obstáculos laterais: Auxiliam o motorista em manobras de ultrapassagem, indicando a existência de algum veículo onde não há visibilidade pelos espelhos retrovisores. Estes sensores baseiam-se em imagens obtidas através de câmeras de vídeo ou de scanners a laser instalados na lateral do veículo.

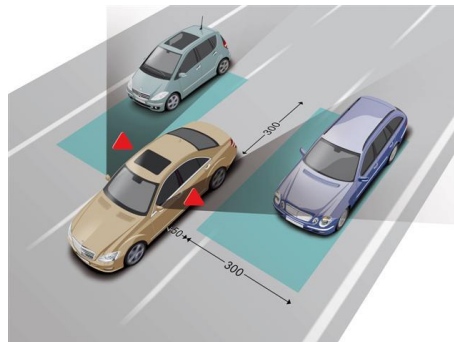


Figura 40 - Identificação de obstáculos laterais (quickcashauto.com)

Sistemas para manter-se centralizado na faixa da pista: Através de uma câmera de vídeo no pára-brisa ou sensores infravermelhos instalados no pára-choque frontal, são identificadas as bordas da pista e uma eventual tendência a sair da pista pode ocasionar um alerta ao motorista ou a direção será corrigida automaticamente. A desvantagem é que tais sistemas necessitam de uma boa pintura das faixas nas rodovias.

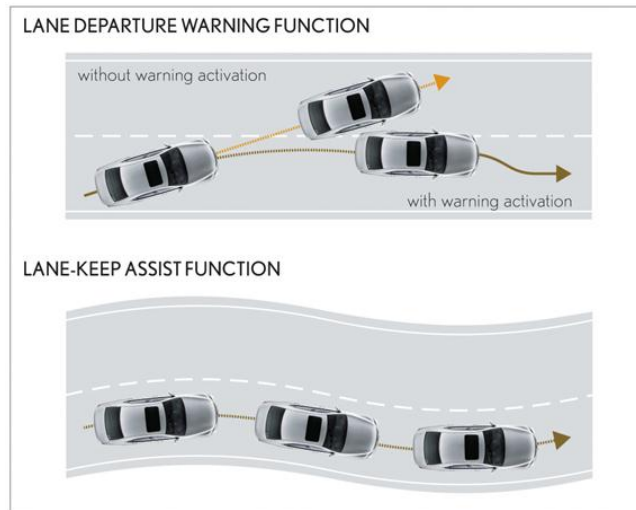


Figura 41 - Sistemas para manter-se centralizado (files.blogter.hu)

Controle de velocidade de cruzeiro avançado: Além de manter a velocidade do veículo constante em um valor previamente determinado, torna-se possível adequar o controle de velocidade do veículo às condições de tráfego da estrada. A informação gerada por sensores de distância instalados no pára-choque permitem ajustar o valor de referência da velocidade com o veículo à frente.

Sistema Anti-Colisão: Quando um obstáculo é detectado a frente do veículo em uma rota de colisão, o sistema avisa o motorista para que o mesmo possa tomar uma ação evasiva. Com objetivo de minimizar a gravidade do impacto da colisão, o sistema atua sobre diversos dispositivos de segurança: caso não exista resposta do motorista para o alerta, o sistema poderá acionar os freios automaticamente e o encosto da cabeça é ajustado para ficar mais próximo da cabeça e por fim os air-bags são preparados para o uso. Tudo isso, baseando-se em informações de radares ou *scanners* a laser. O principal desafio é garantir que o motorista não possa ser impedido ou prejudicado de realizar uma manobra evasiva.

Sistemas “Parar e Continuar”: do inglês “Stop and Go”, são sistemas similares ao controle de cruzeiro, mas especificamente projetados para o trânsito urbano para baixas velocidades, onde sem intervenção do motorista o sistema faz com que o veículo siga o próximo veículo à frente conforme a movimentação do engarrafamento.

Sistemas de Alerta a detecção de pedestres: O sistema irá alertar o motorista caso detectar que um pedestre ou um objeto vulnerável entrou no caminho do veículo. Existem diferentes tecnologias que podem ser utilizadas para construir este sistema e serão analisadas neste trabalho.

Sistemas de auxílio a estacionamento: com auxílio de uma câmera de vídeo ou sensor ultra-sônico instalado na parte traseira do veículo as informações de distancia são transmitidas a um visor localizado no painel do carro e são apresentadas linhas de referência, que auxiliam o motorista durante a manobra.

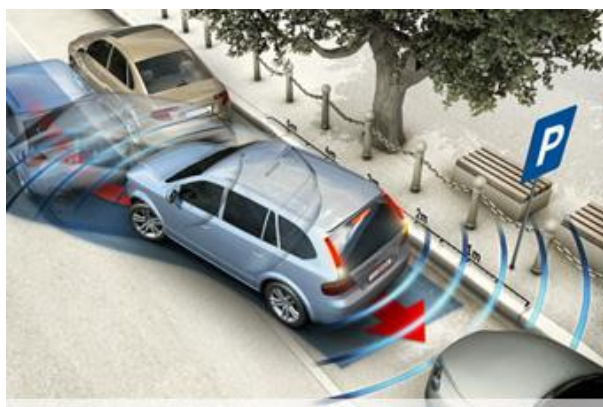


Figura 42 - Sistemas de auxílio a estacionamento (bosch.com)

Sistemas de visão noturna: Com o auxílio de câmeras infravermelhas ou varredura térmica do ambiente é possível mostrar ao motorista através de monitor ou projetando imagens no vidro pára-brisa, uma visão que realça animais ou objetos que podem passar despercebidos em condições de chuva ou pouca luminosidade.

Sistemas de leitura de placas de trânsito: Com o auxílio de câmeras o sistema detecta placas de sinalização de velocidade e compara com a velocidade exercida pelo veículo, avisando o motorista em caso de não conformidade.

Sistemas de navegação por GPS: Com o auxílio do GPS e baseando-se em um mapa digital da região, o sistema ajuda o motorista a planejar seu trajeto até o destino indicado. A desvantagem é que são necessários mapas atualizados da região e uma comunicação sem interferência com os satélites do sistema GPS.



Figura 43 - Navegação por GPS (gpscity.com)

Sistema de comunicação: Os sistemas inteligentes de comunicação inter-veicular permitem alertar de forma automática a ocorrência de acidentes ou impedimento de tráfego (Wischoff, et al., 2003). Dessa forma os motoristas podem trafegar de forma cooperativa, auxiliando o controle de tráfego, reduzindo-se assim, a ocorrência de engarrafamentos e acidentes.

Todos estes sistemas também podem ser classificados como passivos e ativos. No primeiro caso o dispositivo eletrônico identifica os cenários envolvidos e alerta o motorista, que toma as providências necessárias ao volante. No caso dos sistemas ativos, o sistema trabalha como um co-piloto eletrônico e pode interagir na condução do veículo, auxiliando o motorista, mas não interfere nas responsabilidades do condutor.

A2. Sistemas Embarcados Automotivos

Os Sistemas Embarcados, especificamente os mecatrônicos, são formados por conjuntos eletro-mecânicos e são responsáveis por funções específicas no funcionamento do veículo. Pode-se destacar o sistema de injeção de combustível e o sistema de freios ABS. Esses sistemas são controlados por uma unidade eletrônica, que se comunica com as outras através de uma rede interna ao veículo. No setor automotivo é comum o uso da rede CAN para tais aplicações.

Diversas soluções estão disponíveis nos veículos atuais, principalmente nos automóveis de maior valor agregado. Dentre eles os mais comuns atualmente são (Kelber, 2005):

ABS (Anti Blocking System): Impede que as rodas bloqueiem quando houver uma freada forte, evitando derrapagem do veículo.

EBD (Electronic Brake Distribution): distribui força de frenagem a cada roda, aumentando a estabilidade do veículo em manobras rápidas e em terrenos irregulares.

BAS (Braking-Assistant): Ao pisar bruscamente no pedal do freio o sistema identifica como uma colisão iminente e dessa forma, amplifica a intensidade do freio com objetivo de garantir uma desaceleração máxima.

ESP (Electronic Stability Program): Sistema anti-derrapagem. Quando o veículo perde-se nas direções longitudinal e transversal, o sistema atua de forma orientada e independente sobre o freio de cada uma das rodas, evitando que o veículo derrape.

ABC (Active Body Control): Controle dinâmico sobre a carroceria do veículo, evitando que a mesma se incline.

TCS (Torque Control System): Sistema de controle de torque de tração. Realizado de forma eletrônica para cada uma das rodas de tração, evitando que alguma roda derrape e perca o contato seguro com o solo.

CDC (Continuous Damping Control): Sistema de suspensão ativa, que possibilita uma calibração contínua do amortecedor. A partir de comandos eletrônicos, é possível regular a suspensão do veículo para viagem ou de forma esportiva.

A3. Evolução Histórica

Com o desenvolvimento de componentes eletrônicos cada vez menores, a computação embarcada em veículos vem constantemente crescendo, hoje veículos mais modernos podem ter até centenas de dispositivos embarcados. Em conjunto com esse avanço, a área da visão computacional vem pesquisando soluções para atender aos requisitos dos problemas relacionados com a Visão Computacional.

O desenvolvimento de tecnologias direcionadas a veículos inteligentes tem evoluído de forma progressiva nos últimos anos. O interesse em reduzir riscos, tempo de traslado e

consumo de energia trazem grande motivação aos interesses do mercado atual. Dessa forma, vários grupos de pesquisa estão voltados ao estudo das técnicas de direção automatizada, assim, podem-se destacar alguns importantes projetos que estão contribuindo para a evolução da área (adaptado de Tsugawa, 1994):

- Em 1970, um dos primeiros veículos inteligentes construídos, oferecia um sistema de detecção de obstáculos baseado em um sistema de visão e um sistema de navegação que mostrava constantemente a posição do veículo no ambiente através da estimativa das suas coordenadas cartesianas. As velocidades atingidas eram da ordem de 10 km/h.
- No ano 1980, também baseado num sistema de visão, eram capturadas as linhas centrais e laterais onde o objetivo era evitar obstáculos. Este sistema também foi baseado num sistema de visão usando um par estéreo de câmeras, obtendo sucesso a velocidades entre 10-30 km/h.
- Em 1995, o protótipo *VaMP*, desenvolvido pela Universidade de Bundeswehr München (*UBM*), conseguindo manter-se na estrada seguindo as linhas e sinais, controle longitudinal, desviar de obstáculos, e fazer manobras de mudança de estrada. O módulo de detecção baseava-se num processo de reconhecimento onde foram implementados dois submódulos: o de detecção e seguimento de um só objeto (*single object detector and tracker – SODT*) e o de detecção e seguimento de múltiplos objetos (*multiple object detector – MODT*). Ambos com base em um sistema de visão composto por duas câmeras. A estimativa do estado de vetores que descreve o modelo dinâmico do movimento do veículo e da forma da estrada foi realizada através de um algoritmo de filtro de Kalman¹.
- Também em 1995, o sistema *RALPH* (*Rapidly Adapting Lateral Position Handler*), desenvolvido pela Universidade Carnegie Mellon, baseado em um sistema de

¹ Rudolph E. Kalman (1960). O filtro de Kalman (FK) utiliza equações matemáticas que implementam um estimador preditivo de estados, buscando corrigir interativamente a resposta de um determinado sistema através de múltiplas variáveis relacionadas a ele.

visão composto por um par estéreo de câmeras e equipado com um computador portátil, câmeras, um receptor GPS, um sistema de detecção de obstáculos.

- Em 1998, o veículo experimental *ARGO*, desenvolvido no Departamento da Engenharia da Informação da Universidade de Parma, Itália, usou um sistema baseado em visão para captura de imagens e também realizado através de duas câmeras (par estéreo) extraindo informações da estrada e do ambiente por onde se movimenta o veículo.
- Detecção geométrica de linhas para guiamento de veículos inteligentes (Wong, et al., 1999)
- Detecção de linhas e obstáculos baseados em um algoritmo rápido de transformação de perspectiva “fast inverse perspective mapping algorithm (FIPMA)”, (Jiang, et al., 2000).
- Reconhecimento de linhas de estradas e movimentos de veículos usando sistemas de visão (Park, et al., 2003).
- O trabalho *IVVI (Intelligent Vehicle base on Visual Information)* baseado em visão por computador, concentrou-se em sistemas de detecção de sinais de trânsito, outros veículos, pedestres e limites de estrada (Collado, et al., 2003).
- Reconhecimento de linhas na estrada em tempo real usando lógica fuzzy para sistemas de visão de veículos guiados automaticamente (*AGV*), (Kuang, et al., 2004)
- Em 2007 o trabalho “Application Development of Camera-based Driver Assistance Systems on a Programmable Multi-Processor Architecture” (Techmer, 2007) já era específico para para sistemas ADAS.
- Recentemente: Sistemas ADAS com sistema de visão baseada na visão humana (Fritsch, J., et al, 2008)

Apêndice B

B. Detalhamento da Implementação do Hardware da Versão 3

B1. Máquina de Estados do Bloco Estabilizador

O diagrama apresentado na figura Figura 44 foi gerado a partir do algoritmo da versão 1. A grande diferença está na inclusão entradas para verificar se o quadro é valido (iFval) e se a linha do quadro é válida (iLval). Essas duas entradas estão de acordo com um projeto de exemplo de utilização da câmera CMOS que vem com o kit da placa DE2-70.

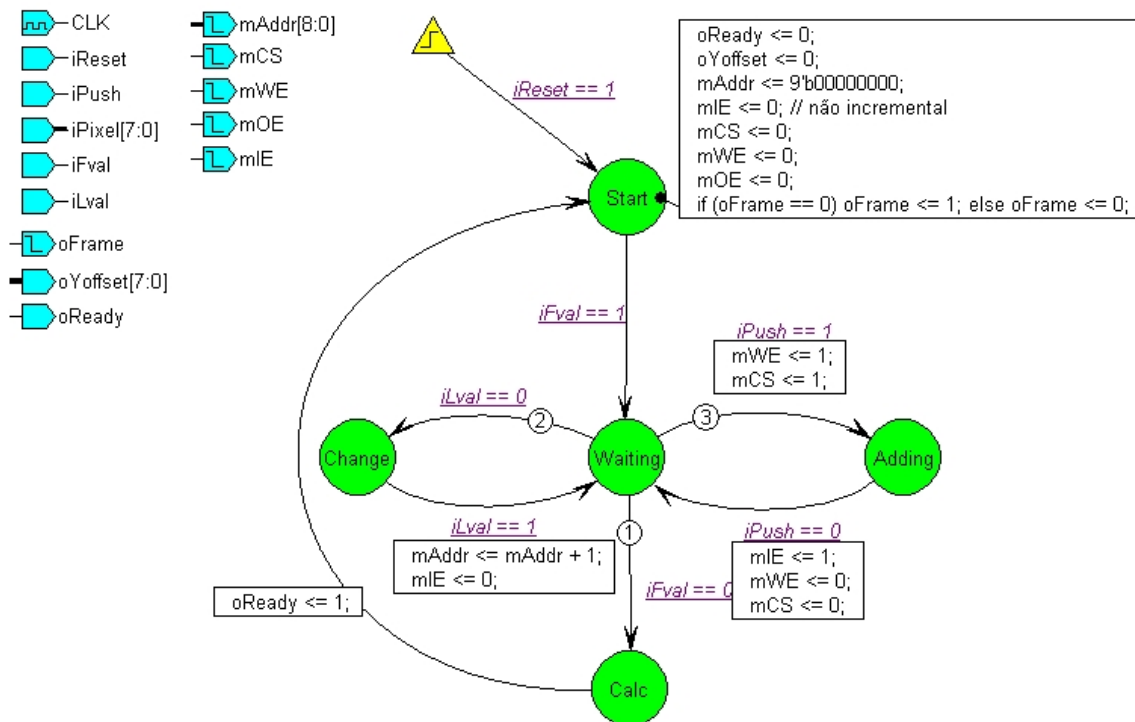


Figura 44 – Diagrama do Bloco Estabilizador da Versão 3

As saídas desta máquina de estados são descritas na Tabela 7:

Tabela 7 – Saídas da Máquina de Estado do Estabilizador

mAddr	Endereço da memória de registro do vetor de deslocamento
mCS	controlador do bloco de memória para identificar Acesso
mWE	controlador do bloco de memória para identificar Escrita
mOE	controlado do bloco de memória para identifica Saída de Dados
mIE	controlado do bloco de memória para identifica Entrada de Dados
oFrame	contador de quadros
oYoffset	fator de deslocamento
oReady	indicador de que o fator de deslocamento está disponível para uso

B2. Diagrama de Hardware do Bloco Estabilizador

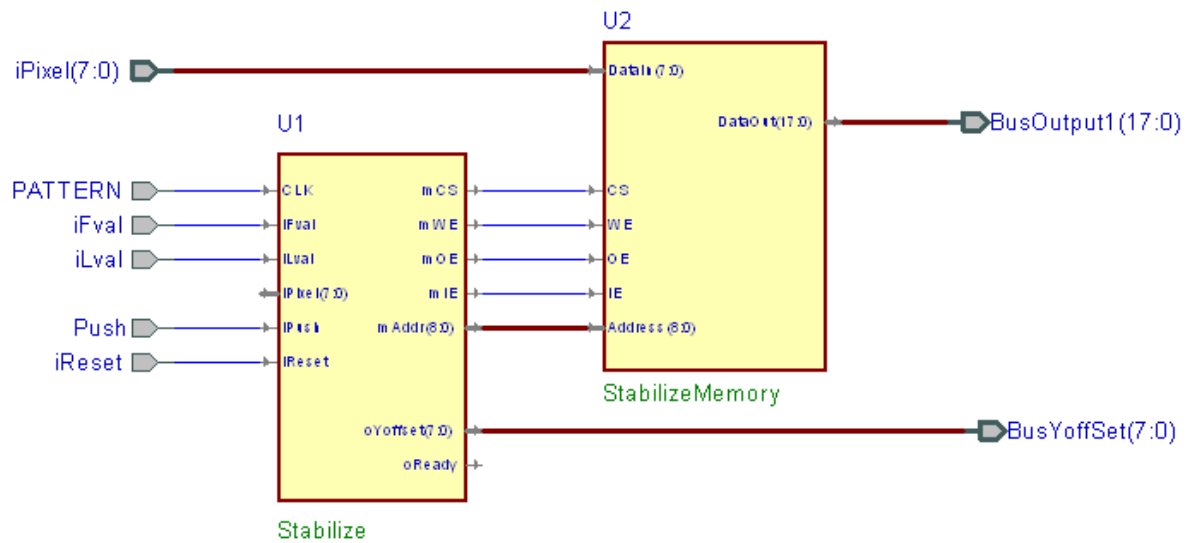


Figura 45 – Bloco Estabilizador da Versão 3

O bloco em hardware do sistema de estabilização da versão 3 descrito na Figura 45 utiliza a máquina de estados do Apêndice B1 em conjunto com um bloco de memória. A saída deste sistema é o índice de deslocamento vertical.

Apêndice C

C. Detalhamento de Módulo para Testes Com Câmera usando NIOS

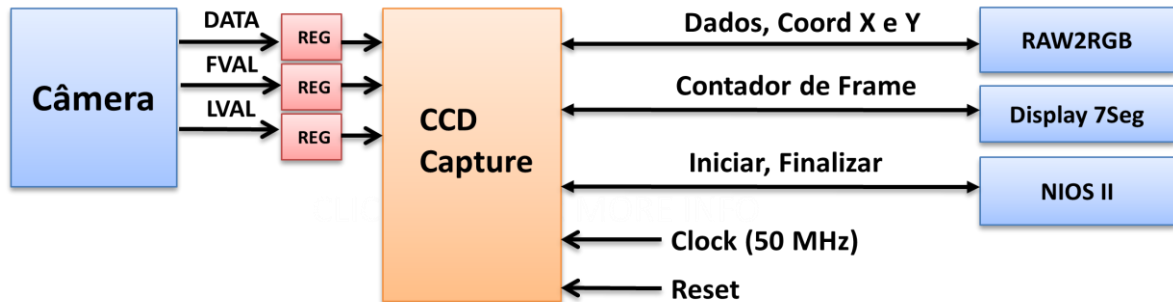


Figura 46 – Integração do módulo de câmera com o NIOS II

A Figura 46 apresenta a interligação da Câmera TRDB-LTM (disponível no kit da DE2-70) com o sistema NIOS. Inicialmente a câmera envia as informações de quadro válido (FVAL – Frame Valid) e linha válida (LVAL – Line valid). Quando essas duas informações estão no nível alto a câmera transfere as informações através do canal DATA (Dados). Esses dados ficam temporariamente armazenados em registradores para depois serem usados pelo módulo de captura CCD. Este módulo é responsável por controlar os estados da câmera e enviar as informações em coordenadas X e Y para um conversor RGB. Através de uma interface de Barramento Avalon o processador NIOS pode controlar a câmera pelo módulo CCD Capture. Por fim, o módulo ainda disponibiliza um contador de quadro que está ligado a um display de 7 segmentos.

A Figura 47 descreve o módulo de controle da memória SDRAM. Os dados enviados pela câmera são convertidos do formato interno da mesma para o formato RGB. O módulo de conversão RGB está diretamente ligado ao controlador de memória que captura a informação e armazena. Como o clock do circuito da câmera é mais lento que o controlador de memória, este alterna em gravar as informações do bloco de conversão e enviar dados da memória para o controlador do LCD.

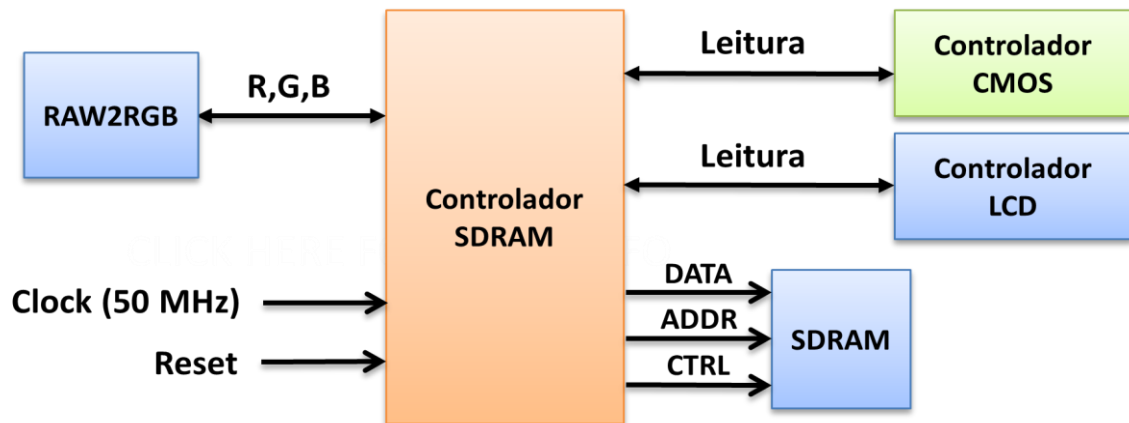


Figura 47 – Detalhamento do Controlador SDRAM

ⁱ *Drive-by-wire* - traduz os comandos do motorista em impulsos elétricos.

Pipeline - Técnica de hardware (permite que a CPU realize a busca de uma ou mais instruções além da próxima a ser executada)