



UNIVERSIDADE DO SUL DE SANTA CATARINA
GUILHERME SCHMITT DA SILVA

**DETECÇÃO DE MOVIMENTO NA AUTOMAÇÃO DOS TESTES DE PRODUÇÃO
DE CÂMERAS CFTV**

PALHOÇA

2011

GUILHERME SCHMITT DA SILVA

**DETECÇÃO DE MOVIMENTO NA AUTOMAÇÃO DOS TESTES DE PRODUÇÃO
DE CÂMERAS CFTV**

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia Elétrica
Telemática, da Universidade do Sul de Santa
Catarina, como requisito parcial para obtenção do
título de Engenheiro Eletricista.

Orientador: Prof. Robinson Pizzio, M. Eng.

Co-orientador: Prof. Marcelo Daniel Berejuck, M. Eng.

PALHOÇA

2011

GUILHERME SCHMITT DA SILVA

**DETECÇÃO DE MOVIMENTO NA AUTOMAÇÃO DOS TESTES DE PRODUÇÃO
DE CÂMERAS CFTV**

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Engenheiro Eletricista e aprovado em sua forma final pelo Curso de Graduação em Engenharia Elétrica Telemática, da Universidade do Sul de Santa Catarina.

Palhoça, 09 de dezembro de 2011.

Prof. e Orientador Robinson Pizzio, M. Eng.
Universidade do Sul de Santa Catarina

Prof. Marcelo Daniel Berejuck, M. Eng.
Universidade do Sul de Santa Catarina

Eng. Robson Veronezi Campos, M. Eng.
Universidade Federal de Santa Catarina

Dedico esse trabalho aos meus pais João e Eni, que sempre me deram o amor e carinho incondicional necessário para minha formação. Para a minha esposa Karen, pela paciência, afeto e incentivo nesta jornada.

AGRADECIMENTOS

Primeiramente a Deus, que me deu força e vitalidade para conseguir alcançar meus objetivos.

Ao meu co-orientador Prof. Marcelo Daniel Berejuck, que aceitou o convite de orientação deste trabalho, bem como sugeriu o tema de pesquisa, apoiando em todos os momentos com sua experiência e sabedoria.

Ao meu orientador Prof. Robinson Pizzio, sempre disposto a auxiliar e sugerir o melhor caminho para a concepção deste trabalho.

Ao Eng. Marcos Roberto da Silva pelo incentivo e disposição em sanar dúvidas a qualquer momento.

A Prof^a. Sheila Santisi Travessa, pelas orientações.

“Pedras no caminho? Guardo todas, um dia vou construir um castelo”
(PESSOA).

RESUMO

Este Trabalho de Conclusão de Curso tem por objetivo propor um sistema automatizado para agilizar o processo de testes finais em câmeras de CFTV (acrônimo de Circuito Fechado de Tele Visão). Tal sistema é baseado na teoria da detecção de movimento, onde o sistema percebe que o usuário finalizou a conexão das câmeras no ambiente de testes e, inicia a avaliação da qualidade das câmeras em questão. O sistema é composto por um kit de desenvolvimento do fabricante Altera® com quatro entradas, três para conexão das câmeras a serem testadas e uma para verificar que não há mais movimentação no cenário de testes e assim iniciar o teste e um canal de saída serial padrão RS232, na qual irá repassar ao usuário o resultado do teste. Foi utilizada a linguagem de síntese VHDL em conjunto com um processador NIOS® para compor o sistema em questão.

Palavras-chave: Câmeras CFTV. Detecção de Movimento. VHDL.

ABSTRACT

This Course Conclusion Work has as goal propose an automated system to expedite the process of final testing of CCTV cameras (acronym for Closed Circuit Tele Vision). This system is based on the theory of motion detection, where the system senses when the user concluded the connection of cameras in the test environment and start assessing the quality of the cameras in question. The system consists of a development's kit by Altera® manufacturer with four inputs, three for cameras connection to be checked and other one to be verified if there no more movement in the test's scenario and so start the test and a standard serial output channel RS232, in which will pass to the user the test result. We used the VHDL synthesis language together the NIOS® processor to compose the system in question.

Key words: CCTV Cameras. Motion Detection. VHDL.

LISTA DE ILUSTRAÇÕES

Figura 01 – Configuração do sistema proposto	15
Figura 02 – Sistema proposto de Detecção de Movimento utilizando FPGA	19
Figura 03 – Estrutura interna de um CCD	22
Figura 04 – Diagrama em blocos ADV7180	24
Figura 05 – Estrutura genérica de um FPGA	25
Figura 06 – Modelo de UART	28
Figura 07 – Estrutura interna da UART	29
Figura 08 – Comunicação utilizando barramento I ² C	30
Figura 09 – Estado das linhas DAS e SCL	31
Figura 10 – Momento das transições	31
Figura 11 – Transferência com o MSB	32
Figura 12 – Detecção de Movimento	34
Figura 13 – Detecção do objeto na imagem	36
Figura 14 – Aplicação do algoritmo de difusão de calor anisotrópico	38
Figura 15 – Kit de desenvolvimento Altera®	40
Figura 16 – Estrutura da câmera utilizada no sistema proposto	41
Figura 17 – Softwares para tratamento dos frames	42
Figura 18 – Algoritmo para coleta dos frames	43
Figura 19 – Algoritmo para comparação dos frames	43
Figura 20 – Sistema proposto em sua totalidade	44
Figura 21 – Diagrama em blocos do sistema proposto em sua totalidade	44
Figura 22 – Cálculo de similaridade entre as cenas	45
Figura 23 – Cenário de referência do sistema proposto	48
Figura 24 – Imagem de referência do sistema proposto	48
Figura 25 – Cenário parcialmente escuro	49
Figura 26 – Imagem do sistema parcialmente escuro	50
Figura 27 – Cenário com objeto entre a câmera e o quadro branco	51
Figura 28 – Imagem com objeto entre a câmera e o quadro branco	51
Figura 29 – Gráfico Comparativo Limite Real e Limite com Luminosidade	54

LISTA DE TABELAS

Tabela 01 – Comparação dos Frames coletados no Cenário A	46
Tabela 02 – Comparação dos Frames coletados no Cenário B	52
Tabela 03 – Comparação dos Frames dos cenários A e C	53

LISTA DE SIGLAS

Acknowledge - Confirmar

BS - *Background Subtraction*

CCD - *charge-coupled device*

CFTV - Circuito Fechado de Televisão

CIF - *Common Intermediate Format*

clock - Relógio

CoDec - Codificador / Decodificador

Core - Núcleo

CPLD - *Complex Programmable Logic Device*

dB - Decibel

Dev C++ - Ferramenta para linguagem de programação

DSP - *Digital Signal Processor*

EEProm - *Electrically-Erasable Programmable Read-Only Memory*

FPGA - *Field Programmable Gate Array*

Frame - quadro

Hz – *hertz*

I/O - *Input / Output*

IC - *Integrated Circuits*

Mhz - *Megahertz*

MSB - *Most Significant Bit*

NTSC - *National Television Standards Committee*

PAL - *Phase Alternating Line*

RS232 - *Recommended Standard 232*

SA - *Simulated Annealing*

SCL - *Serial Clock Line*

SDA - *Serial Data Line*

SECAM - *Séquentiel Couleur à Mémoire*

SPLD - *Simple Programmable Logic Device*

UART - *Universal Asynchronous Receiver/Transmitter*

UNISUL – Universidade do Sul de Santa Catarina

VHDL - *VHSIC Hardware Description Language*

VHSIC - *Very High Speed Integrated Circuits*

Vpp - Volts pico a pico

LISTA DE SÍMBOLOS

® - Marca Registrada

SUMÁRIO

1 INTRODUÇÃO	15
1.1 DEFINIÇÃO DO PROBLEMA	16
1.2 OBJETIVOS	16
1.2.1 Objetivo geral	17
1.2.2 Objetivos específicos	17
1.3 JUSTIFICATIVA	17
1.4 METODOLOGIA	18
1.5 ESTRUTURA DO TRABALHO	20
2 FUNDAMENTAÇÃO TEÓRICA	21
2.1 CÂMERAS	21
2.2 PADRÕES DE IMAGEM	23
2.3 CODEC DE IMAGEM	24
2.4 DISPOSITIVOS LÓGICOS PROGRAMAVEIS E SUA LINGUAGEM DE HARDWARE	25
2.5 BARRAMENTOS	27
3 DETECÇÃO DE MOVIMENTO	33
3.1 MOVIMENTO NA CENA	33
3.2 DETECÇÃO DO MOVIMENTO	34
3.3 ESTUDO DE CASO	37
4 DESENVOLVIMENTO DO PROJETO PROPOSTO	39
4.1 KIT DE DESENVOLVIMENTO DO FABRICANTE ALTERA®	39
4.2 CÂMERA UTILIZADA NO SISTEMA PROPOSTO	40
4.3 SOFTWARES DESENVOLVIDOS PARA COLETAR E COMPARAR AS AMOSTRAS	41
4.4 LUMINOSIDADE	45
5 RESULTADOS OBTIDOS	47
5.1 CENÁRIOS DE TESTES	47
5.2 COMPARAÇÃO ENTRE O CENÁRIO A COM O CENÁRIO B	52
5.3 OBJETO NO CENÁRIO DE REFERÊNCIA	53
6 TRABALHOS FUTUROS	56
7 CONSIDERAÇÕES FINAIS	57

REFERÊNCIAS58

1 INTRODUÇÃO

A utilização de câmeras de vídeo em nosso cotidiano é cada vez mais comum e os segmentos em que ela pode atuar são os mais variados possíveis, podendo estar em atividades monótonas para o ser humano ou, ainda, chegando a tarefas perigosas ou inacessíveis (resgate de objetos no oceano) (OLIVEIRA, 2003).

Os sistemas de segurança baseados em dispositivos conhecidos como Circuito Fechado de Televisão (CFTV), são constituídos pelo uso de equipamentos de tecnologia de vigilância visual e são projetados para monitorar diversos tipos de ambientes, fazendo uso intensivo de câmeras de vídeo. São denominadas de “câmeras CFTV” e que, segundo fabricantes de equipamentos de segurança, tem aumentado o seu consumo significativamente nos últimos anos no Brasil.

O aumento na produção de câmeras de CFTV exige que o processo de produção seja cada vez mais eficiente, mantendo um bom padrão de qualidade e produzindo um número maior de câmeras por hora de produção. Neste contexto, este trabalho propõe a realização de um sistema de teste de modo que o usuário, ou operador do sistema de testes, tenha apenas que conectar câmeras no ambiente de testes e, quando o sistema não detectar mais o movimento das mãos do usuário, inicia-se a validação das câmeras de CFTV, informando ao usuário um resultado de aprovado ou não.

A Figura abaixo demonstra a elaboração do sistema:

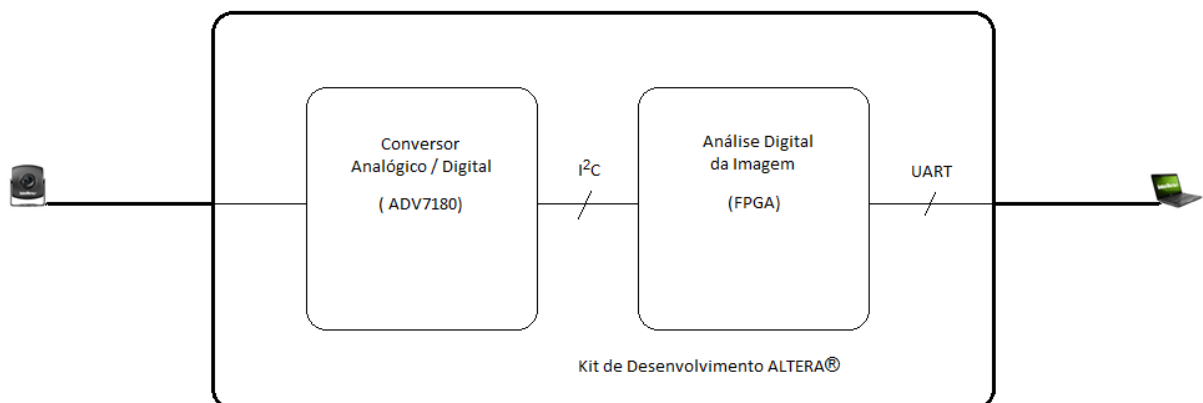


Figura 01 – Configuração do sistema proposto
Fonte: Elaboração do autor, 2011.

1.1 DEFINIÇÃO DO PROBLEMA

O controle de qualidade de câmeras CFTV ainda depende muito da percepção humana, conseqüentemente, os erros inerentes a esta análise acabam prejudicando todo o processo de validação em uma indústria. Um lote de câmeras produzidas e testadas de forma equivocada gera para a empresa um retorno negativo, tanto da sua imagem diante do mercado, quanto para os clientes que buscam na concorrência a solução para suas necessidades.

O processo de controle de qualidade para estes produtos apresenta um custo relativamente alto, exemplo disto é o equipamento chamado Vectorscope, o que inviabiliza o investimento em algumas empresas, já que além da aquisição de equipamentos para este fim, há também o custo de manutenção. Com isso, utilizar apenas a análise humana para avaliar a qualidade de uma câmera CFTV é, em um projeto de desenvolvimento, um risco a ser considerado.

No entanto, a proposta deste Trabalho de Conclusão de Curso é buscar desenvolver um sistema baseado na detecção de movimento utilizando dispositivo de lógica programável – FPGA (acrônimo de *Field Programmable Gate Array*) que automatize os testes nas câmeras CFTV em uma linha de produção, diminuindo assim os possíveis riscos inerentes a qualidade em um projeto de desenvolvimento e produção.

1.2 OBJETIVOS

Os objetivos deste trabalho podem ser separados em objetivo geral e objetivos específicos.

1.2.1 Objetivo geral

Desenvolver um sistema que baseado na detecção de movimento do usuário / operador, inicie a avaliação das câmeras CFTV utilizando para isto um dispositivo de lógica programável – FPGA.

1.2.2 Objetivos específicos

Para o presente trabalho almeja-se atingir os seguintes objetivos específicos:

- Buscar referências bibliográficas no que tange os assuntos câmeras, detecção de movimento e processamento digital da imagem;
- Buscar desenvolver um sistema de detecção de movimento utilizando dispositivo de lógica programável – FPGA;
- Definir uma métrica que sirva de parâmetro para detectar o movimento ou não.

1.3 JUSTIFICATIVA

A percepção visual de um usuário / operador não deve ser considerada um parâmetro na avaliação de câmeras CFTV, no entanto, devido aos custos elevados em equipamentos para este fim, opta-se por uma análise humana.

O processo de validação de câmeras CFTV é um quesito importante em um projeto de desenvolvimento, pois estes produtos são destinados, na maioria dos casos, para uso de segurança patrimonial o que exige um bom nível de confiabilidade das imagens captadas.

Face ao exposto, podemos citar as seguintes justificativas para este trabalho:

- Minimizar a interação / percepção humana no cenário de validação de câmeras CFTV;
- Implementar um sistema de detecção de movimento em FPGA com custo acessível e adaptado as necessidades de uma empresa;
- Diminuir o tempo para avaliar câmeras CFTV em uma linha de produção;
- Aumento da confiabilidade dos testes destinados a validar câmeras CFTV.

1.4 METODOLOGIA

Primeiramente buscaram-se referências bibliográficas em torno dos assuntos pertinentes a elaboração deste trabalho, tais como câmeras, detecção de movimentos e processamento digital de imagem. Após esta fase, o propósito do trabalho foi definido.

Na fase de desenvolvimento do trabalho, foi realizado o estudo do kit de desenvolvimento do fabricante Altera®, de modo a detalhar os recursos a serem utilizados. Na sequência, foi analisado como capturar e analisar a detecção de movimento do usuário / operador visando iniciar os testes de validação de câmeras CFTV.

O software para realizar esta detecção de movimento foi escrito utilizando linguagem C, através da ferramenta NIOS II. Assim, a câmera irá analisar o cenário de testes e no momento em que não houver movimento do operador, o sistema irá iniciar a validação das câmeras CFTV, seguindo os procedimentos descritos por Gesser (2011).

A Figura 2 descreve com detalhes as etapas para elaboração deste trabalho:

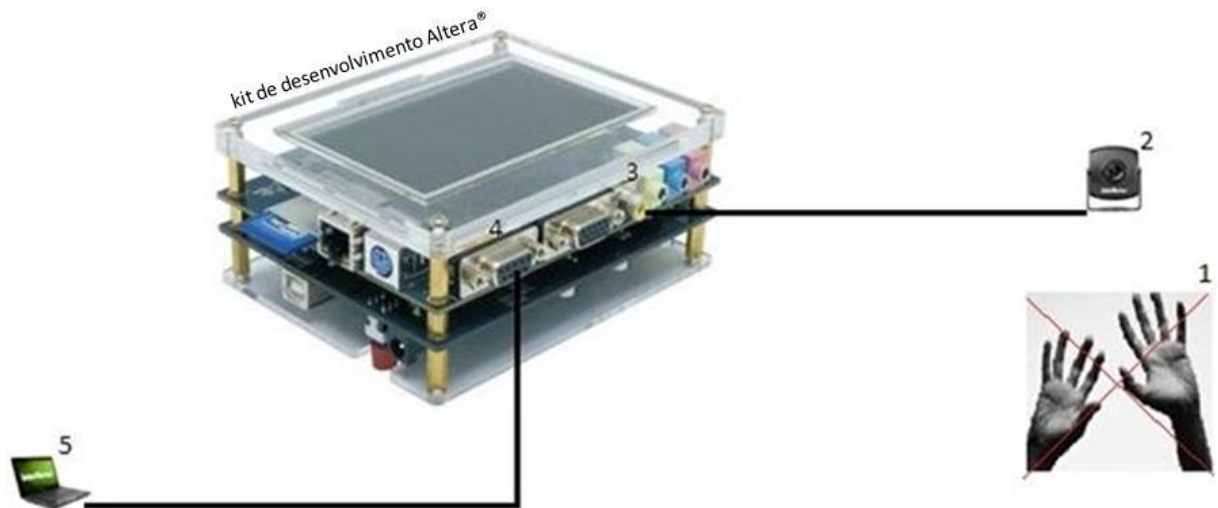


Figura 02 – Sistema proposto de Detecção de Movimento utilizando FPGA
 Fonte: Elaboração do autor, 2011.

Onde:

- 1) Movimentação do usuário / operador para inserir as câmeras a serem avaliadas no controle de qualidade, processo este descrito por Gesser (2011);
- 2) Câmera que irá analisar o cenário e, tão logo não haja movimento na cena, inicia o processo de validação;
- 3) Entrada de vídeo do kit de desenvolvimento do fabricante Altera®. Esta entrada está diretamente conectada ao Codificador / Decodificador ADV7180, no qual irá transformar o sinal analógico para sinal digital que, por sua vez, será enviado para o FPGA que irá realizar a análise digital da imagem para checar se houve ou não movimento no cenário de validação das câmeras;
- 4) Interface serial RS232 utilizado para enviar o resultado da análise de detecção de movimento para o usuário, deixando-o ciente que a validação será iniciada;
- 5) Computador utilizado para apresentar os resultados da detecção de movimento.

1.5 ESTRUTURA DO TRABALHO

Este trabalho está dividido em sete capítulos. No Capítulo 1 é apresentada a introdução, a definição do problema, os objetivos gerais e específicos, a justificativa e a metodologia deste trabalho. Já no Capítulo 2 abordam-se definições para a concepção do projeto tais como câmeras, padrões de imagem, CoDec (acrônimo para Codificador/Decodificador), dispositivos lógicos programáveis e barramentos. O Capítulo 3 é destinado a descrever a detecção de movimento em si, ou seja, a fundamentação em torno deste assunto e as técnicas utilizadas. Já no Capítulo 4 aborda-se como o projeto foi desenvolvido, os dispositivos utilizados para obtenção do sistema proposto, bem como a lógica embarcada no sistema. O Capítulo 5 apresenta os resultados obtidos. O capítulo 6 sugere trabalhos futuros tomando este sistema como referência. Por último, o Capítulo 7 aborda as considerações finais do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo abordará conceitos importantes para o entendimento do projeto em questão. A Seção 2.1 descreve os conceitos em torno do assunto câmeras. A Seção 2.2 trata dos padrões de imagem utilizados e qual o adotado para este tipo de aplicação. Já a Seção 2.3, trata sobre as questões que envolvem o CoDec (Codificador/Decodificador) de imagem. A Seção 2.4 aborda o assunto que envolve os dispositivos lógicos programáveis, bem como a linguagem de hardware utilizada para estes dispositivos. Finalizando o capítulo, a Seção 2.5 trata dos barramentos utilizados, como por exemplo, I²C, RS232 e UART.

2.1 CÂMERAS

De acordo com Vahid e Givargis (2002), uma câmera digital é um dispositivo eletrônico de consumo popular que pode capturar imagens, ou “tirar imagens”, e armazenar elas em formato digital. As câmeras digitais não contêm filme, mas sim um ou mais circuitos integrados que possuem processadores e memórias. As câmeras digitais não foram possíveis a uma década atrás em virtude dos circuitos integrados não possuírem processamento rápido e armazenamento o suficiente para torna isto viável. O advento da memória do sistema em um dispositivo de alta capacidade e memórias flash tornou tais câmeras possíveis.

Do ponto de vista do desenvolvedor, uma câmera digital simples executa duas tarefas fundamentais. A primeira tarefa é de captura / processamento de imagens e armazenamento das mesmas na memória interna, já a segunda tarefa é a de descarregar as imagens em série a um computador.

A tarefa de captura / processamento e armazenamento da imagem é iniciada quando o sistema recebe a luz refletida desta imagem em suas lentes. Neste momento, a imagem é capturada e convertida para o formato digital por um *charge-coupled device* (CCD). Então, a imagem é processada e armazenada na memória interna.

O CCD é um sensor especial que captura uma imagem, ou seja, é um dispositivo de estado sólido de silício sensível a luz composta de células pequenas. A luz incidindo sobre uma célula é convertida em uma pequena quantidade de carga elétrica, que é medida pelo sistema eletrônico do CCD e armazenada como um número. O número geralmente varia de 0, ou seja, sem luz, para 255 ou 65.535, o que significa luz muito intensa por pixel. A Figura 03 ilustra o interior de um CCD.

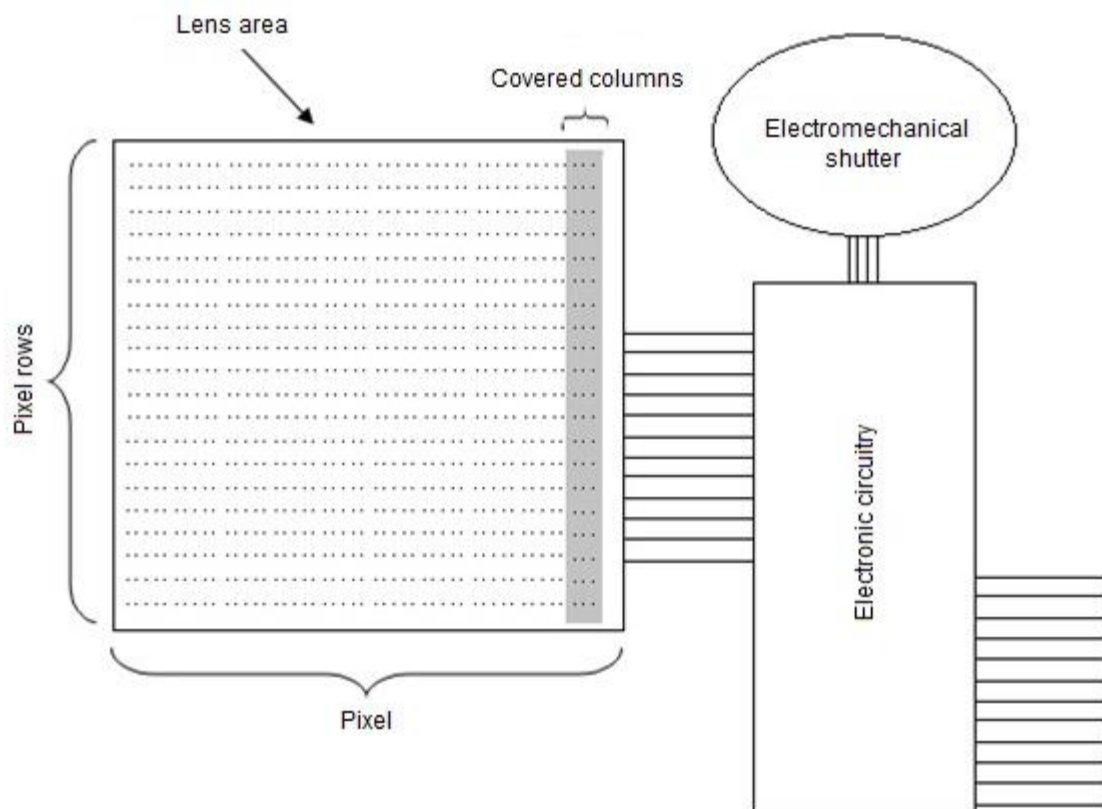


Figura 03 – Estrutura interna de um CCD.

Fonte: Vahid e Givargis (2002, p. 181) e adaptado pelo autor

Em sua periferia, um CCD é composto por um disparador mecânico. Esta é uma tela que normalmente bloqueia a incidência da luz na superfície sensível a mesma. Quando ativada, a tela se abre momentaneamente e permite que a luz atinja a superfície sensível à luz, carregando as células com energia elétrica proporcional à quantidade de luz que atingiu a superfície. A tela normalmente fica atrás de uma lente ótica que focaliza a cena observada através do visor para a superfície sensível à luz do CCD. O CCD tem também um circuito interno que mede a carga elétrica de cada célula, converte para um valor digital, e oferece uma interface para saída de dados.

Devido a erros de fabricação, células sensíveis à luz de um CCD podem medir a intensidade da luz um pouco acima ou abaixo do valor real. Este erro, o chamado erro de "ponto zero", é geralmente o mesmo em colunas, mas diferentes entre as linhas.

Por esta razão, algumas das colunas mais à esquerda de células sensíveis à luz são bloqueadas por uma faixa de tinta preta. A intensidade efetiva registrada por essas células bloqueadas deve ser zero. Portanto, uma leitura diferente de zero indica o erro de "ponto zero" para essa linha. A Figura 03 mostra células cobertas.

2.2 PADRÕES DE IMAGEM

Segundo Harwood (2008), existem três padrões de vídeo dominantes em uso ao torno do mundo, NTSC, PAL e SECAM. Esses três formatos foram desenvolvidos em diferentes partes do mundo por razões históricas. No entanto, no Brasil foi adotado o padrão NTSC.

Ainda de acordo com este autor, o NTSC (*National Television Standards Committee*) é o comitê que define as normas do sistema de televisão para os Estados Unidos e Japão. Criado nos Estados Unidos foi o primeiro sistema de televisão a cores. Este sistema possui 525 linhas de varredura com frequência de campo igual a 60Hz (vertical) e frequência de linha igual a 15.743Hz (horizontal). A largura de banda do vídeo e a portadora de áudio são 4,2MHz e 4,5MHz, respectivamente.

No entanto, nem todas as 525 linhas ficam ativas, ou seja, visíveis na tela. Algumas linhas são utilizadas para equalização de sincronização vertical ou ainda são praticamente invisíveis devido ao efeito overscanning (área em torno das quatro bordas da imagem).

Para efeito de comparação, Damjanovski (2005) demonstra as diferenças entre os referidos padrões de TV:

- PAL: 625 linhas / 50 imagens entrelaçadas por segundo;
- NTSC: 525 linhas / 60 imagens entrelaçadas por segundo;
- SECAM: 625 linhas / 50 imagens entrelaçadas por segundo.

2.3 CODEC DE IMAGEM

Neste trabalho será utilizado o codec ADV7180 do fabricante Analog Device, pois trata-se de um dispositivo versátil e que detecta e converte automaticamente os padrões de vídeo PAL, NTSC e SECAM. Neste trabalho estaremos adotando o padrão NTSC, já que o modelo adotado no Brasil.

A Figura 04 mostra o diagrama em blocos do codec ADV7180:

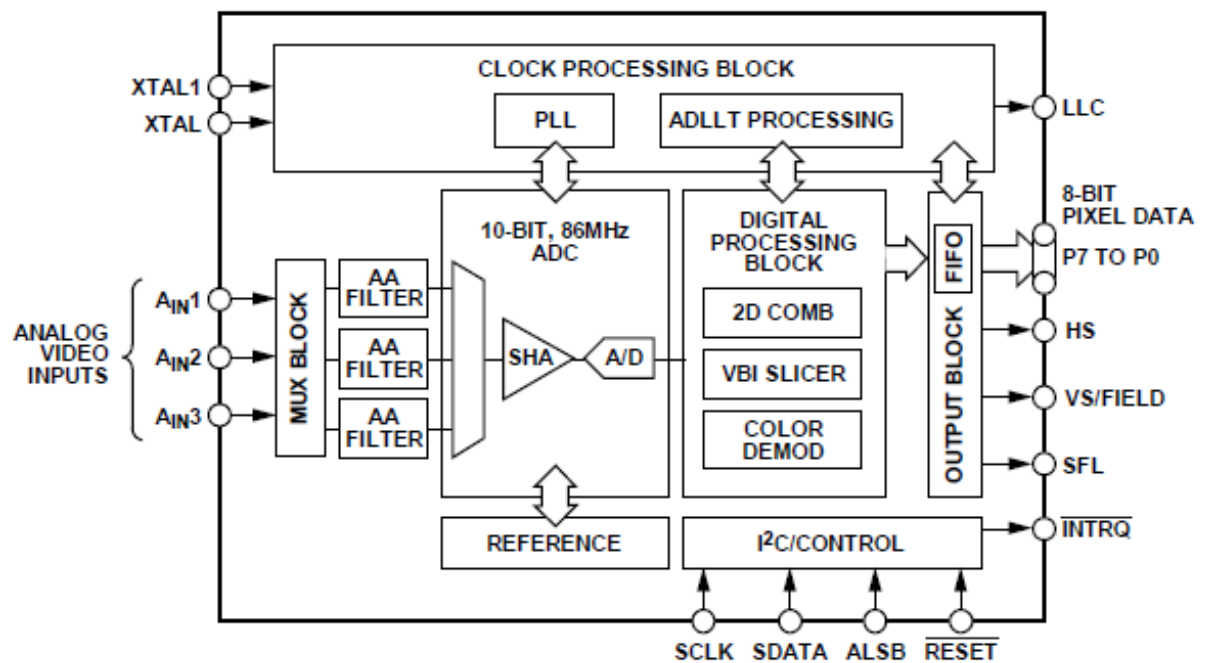


Figura 04 – Diagrama em blocos ADV7180.
Fonte: Analog Device

2.4 DISPOSITIVOS LÓGICOS PROGRAMÁVEIS E SUA LINGUAGEM DE HARDWARE

Conforme Brown e Zvonko (2008), pelos padrões modernos, um circuito de lógica com 10.000 portas (circuitos lógicos que operam “n” níveis lógicos de entrada, porém obtém como resultado apenas um nível lógico (0 ou 1) (VAHID; GIVARDIS, 2002) não é grande.

Ainda segundo estes autores, para implementar circuitos maiores, é conveniente usar um tipo diferente de circuito integrado (CI) que tem uma maior capacidade de lógica. O *Field - programmable gate array* (FPGA) é um dispositivo lógico programável que suporta implementações de circuitos lógicos relativamente grandes. FPGAs são bastante diferentes de SPLDs (*Simple Programmable Logic Device*) e CPLDs (*Complex Programmable Logic Device*) porque FPGAs não possuem planos AND ou OR.

Em vez disso, FPGAs fornecem blocos lógicos para execução das funções exigidas. A estrutura genérica de um FPGA é ilustrada na Figura 05.

O FPGA contém três tipos principais de recursos: blocos lógicos, blocos I/O (*Input/Output*) para conectar os pinos do conjunto, e os fios de interligação e switches.

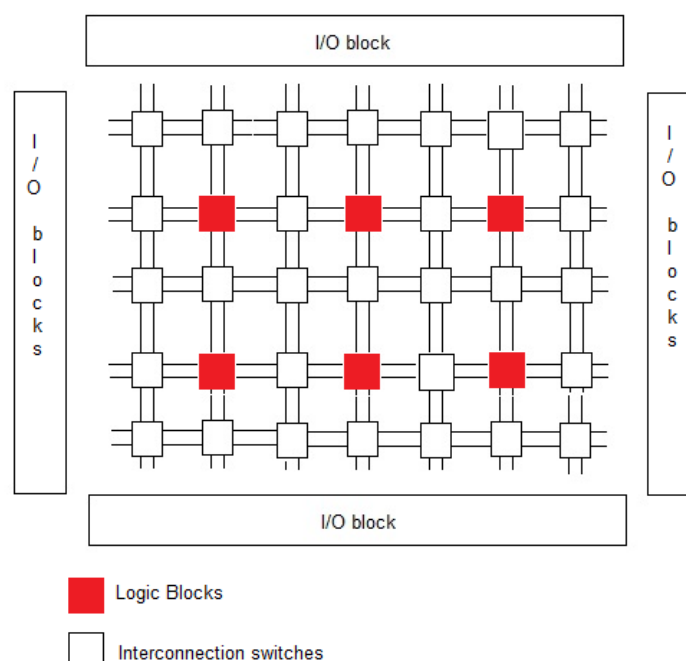


Figura 05 – Estrutura genérica de um FPGA
Fonte: Brown e Zvonko (2008, p. 110).

Contudo, para realizar a programação deste dispositivo, faz-se necessário uma linguagem de hardware. Uma das linguagens utilizadas denomina-se VHDL (*VHSIC Hardware Description Language*), onde VHSIC significa *Very High Speed Integrated Circuits*. Assim, VHDL é uma linguagem de descrição de hardware que pode ser usada para descrever e modelar sistemas digitais, conforme Murdocca e Heuring (2001).

Conforme estes autores, esta linguagem de hardware modela características temporais de forma natural, e permite o controle da progressão de eventos através do tempo. Ao contrário da maioria das linguagens procedimentais em uso comum, VHDL implementa execução concorrente e é ativada por eventos (*event driven*).

- Execução Concorrente: significa que, a não ser quando explicitado, todos os comandos em uma especificação VHDL são executados em paralelo. Esta é a modelagem natural de um sistema digital, pois os mesmos funcionam “em paralelo”;
- Sistemas ativados por Eventos: HDL trata da propagação de sinais através de sistemas digitais, e portanto implementa de forma lógica e natural o conceito de mudanças de estado como função do tempo. Conseqüentemente, implementam conceitos como “depois”, “até que” e “espere por”. Como um sistema ativado por eventos, começa sua execução iniciando variáveis, então identifica todas as mudanças de valores de sinais de 0 para 1 e de 1 para 0 que ocorrem nas entradas e saídas dos componentes.

Essas mudanças ou eventos são armazenados em uma fila ordenada por tempo, conhecida como fila de eventos. Esses eventos são examinados, e se um evento tem algum efeito em algum componente, este efeito é avaliado. Se o efeito causa a ocorrência de outros eventos, o simulador coloca os novos eventos na fila de eventos, e o processo continua até que não haja mais eventos para serem processados.

Em VHDL a especificação de quaisquer componentes é dividida em duas partes, uma parte denominada entidade e uma parte denominada arquitetura.

Estas partes correspondem aproximadamente às partes sintáticas e semânticas de especificação da linguagem. A parte denominada entidade descreve

a interface do componente sem dizer nada sobre sua estrutura interna. A parte denominada arquitetura descreve o comportamento interno do componente.

2.5 BARRAMENTOS

A transferência de dados em modo serial pode ser realizada de duas maneiras: através de comunicação síncrona ou assíncrona. Na comunicação síncrona existe um sinal de relógio (usualmente chamado pela palavra inglesa *clock*) que é utilizado como referência para a transmissão dos dados entre os dispositivos que se comunicam através deste tipo de interface serial. Ou seja, existe uma dependência do sinal de *clock* para haver comunicação entre os dispositivos.

Já para realizar uma transmissão serial de um dado de forma assíncrona (independência de *clock*) é necessário que ambos os lados da linha de transmissão haja um dispositivo capaz de decompor cada caractere bit a bit e prover a inclusão de bits adicionais que são usados para sincronizar o início e o fim da comunicação, conhecidos como bits START/STOP. Ainda, quando dispositivos estão usando comunicação assíncrona devem ter suas taxas de transferências previamente estabelecidas, por exemplo, 2400 kbps. Um dispositivo muito comum, usado em microcomputadores e que faz parte da maioria dos circuitos integrados de entrada/saída, denomina-se UART (*Universal Asynchronous Receiver/Transmitter*), transmissor/receptor universal assíncrono (MONTEIRO, 1996).

A UART é um mecanismo de comunicação serial assíncrono que transforma em serial a saída paralela do barramento do computador, cuja função básica é a decomposição e composição de um caractere em bits e vice-versa, conforme mostrado esquematicamente na Figura 06.

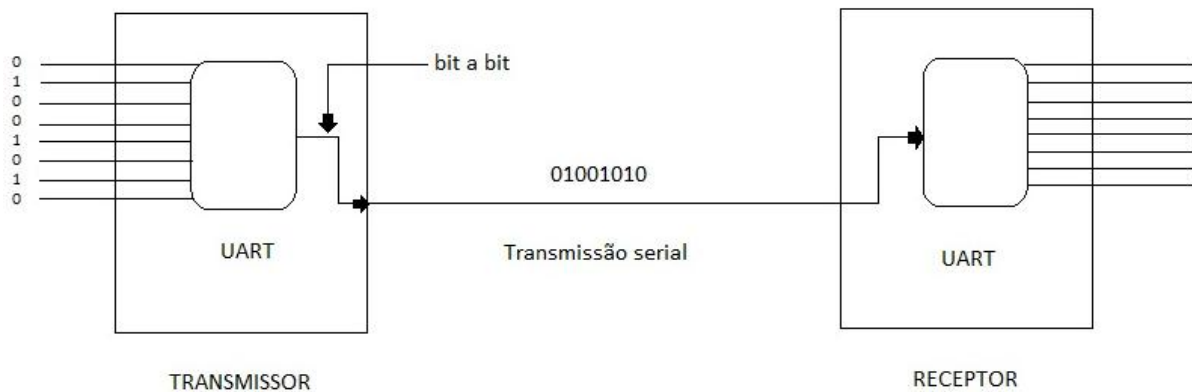


Figura 06 – Modelo de UART.
 Fonte: Monteiro (1996, p. 347)

Ainda segundo Monteiro (1996), dentre os principais elementos que compõe uma UART, podem ser destacados:

- Buffer de saída de dados a serem transmitidos: recebe os “n” bits do caractere (podem ser 5, 6, 7 ou 8 bits, dependendo do dispositivo que está sendo usado) e os envia para o registrador de transmissão;
- Registrador de transmissão: desloca os bits do caractere uma a uma (*shift register*) para a linha de saída. Este deslocamento é realizado a cada pulso de relógio da UART;
- Registrador de recepção e buffer de saída de dados recebidos: funcionam de modo semelhante, porém em sentido inverso ao dos dois registradores já descritos. O caractere é recebido bit a bit no registrador de recepção, que efetua o deslocamento de cada bit até completar todo o caractere e, então, o encaminha para o buffer de saída;
- Unidade de controle: permite que a UART funcione de modo diferente, conforme a escolha do usuário, ou seja:
 - Opção de paridade: se será par ou ímpar;
 - Opção de STOP: 1 ou 2 bits.
- Registrador de estado (semelhante ao flag dos microprocessadores): possui um bit para indicar algumas ocorrências durante o funcionamento da UART, tais como: erro de paridade, erro de sincronização (a UART perdeu o bit START), dados disponíveis (para que o processador leia o caractere);

- Relógio: divide a frequência de transmissão (taxa de *bands*) para permitir o deslocamento de cada bit dos registradores e deslocamento.

Os blocos aqui descritos estão representados na Figura 07:

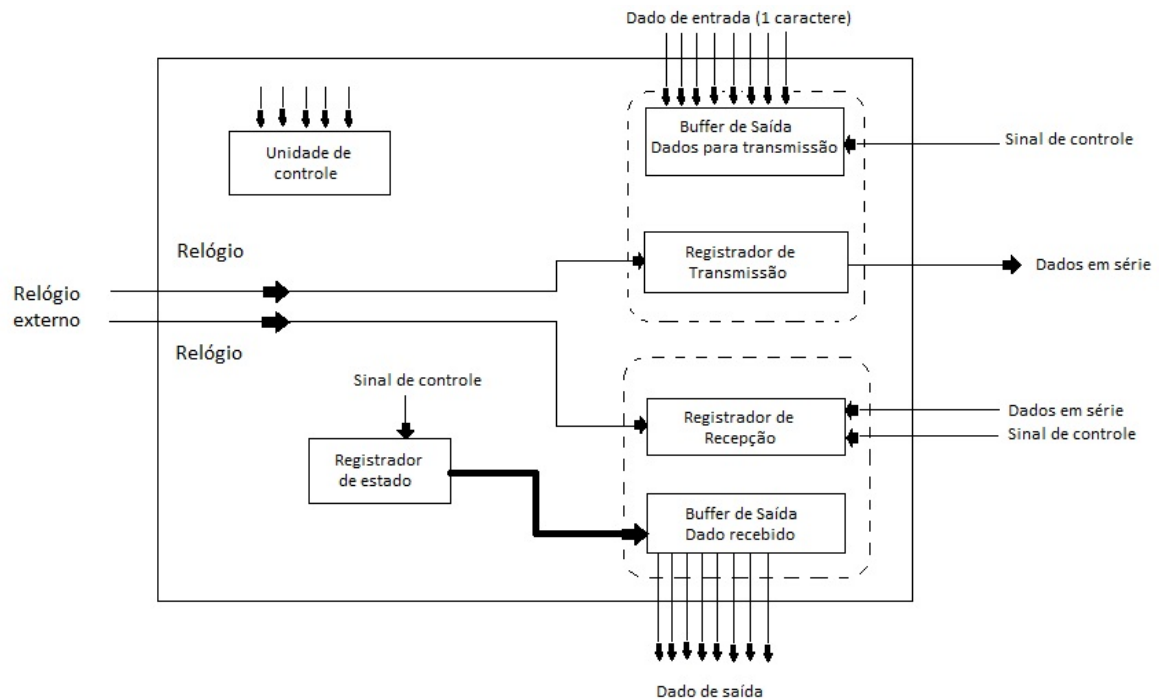


Figura 07 – Estrutura interna da UART
Fonte: Monteiro (1996, p. 348)

Barramentos continuam sendo melhorados e aperfeiçoados, conforme a aplicabilidade da solução proposta, exemplo disto foi o barramento que a Phillips Semicondutores desenvolveu denominado I²C, (UM10204, 2007).

O I²C é uma forma de comunicação serial síncrona que utiliza dois fios para controlar circuitos integrados (IC – sigla inglesa que significa *Integrated Circuits*). Neste modo de comunicação apenas duas linhas de barramento são necessárias:

- SDA (Serial Data Line): Linha de Dados Serial;
- SCL (Serial Clock Line): Linha de Clock Serial.

A Figura 08 apresenta um exemplo de comunicação entre o microcontrolador A e B, utilizando o barramento I²C:

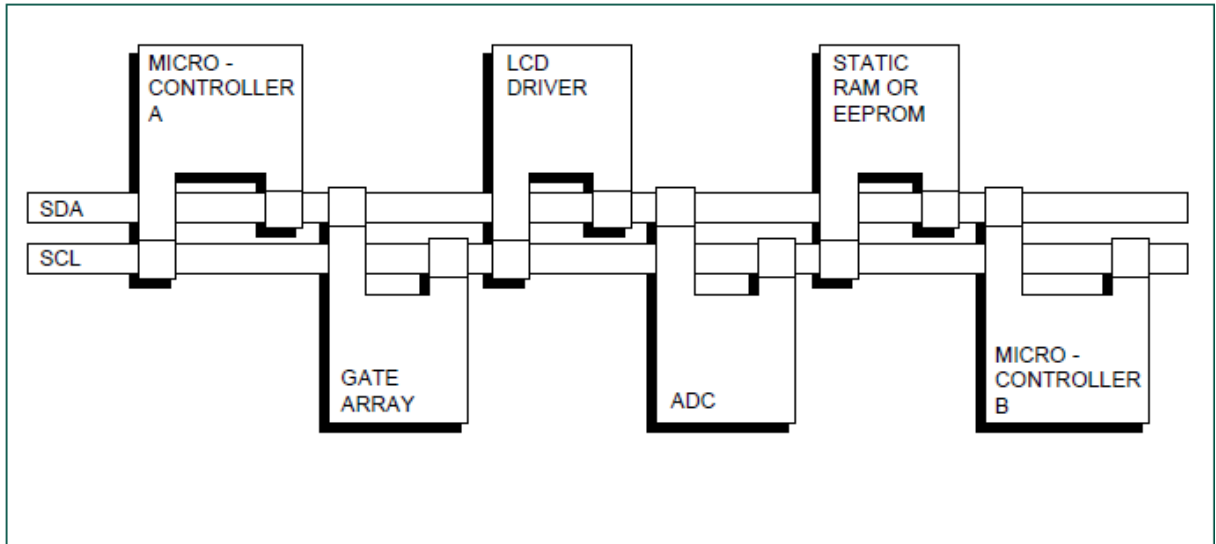


Figura 08 – Comunicação utilizando barramento I²C
 Fonte: UM10204 (2007, p. 10)

Suponha que um microcontrolador A quer enviar informações para o microcontrolador B, então:

- O microcontrolador A (mestre), aborda o microcontrolador B (escravo);
- O microcontrolador A (transmissor-mestre), envia dados para o microcontrolador B (receptor-escravo);
- O microcontrolador A termina a transferência.

Agora, se o microcontrolador A quer receber informações do microcontrolador B:

- O microcontrolador A (mestre) aborda microcontrolador B (escravo);
- O microcontrolador A (mestre-receptor) recebe dados do microcontrolador B (transmissor-escravo);
- O microcontrolador A termina a transferência.

Os dados na linha SDA deve ser estável durante o período de alta do *clock*.

O estado de ALTO ou BAIXO da linha de dados só pode mudar quando o sinal de *clock* na linha SCL é BAIXA (ver Figura 09). Um pulso de *clock* é gerado para cada bit de dados transferidos.

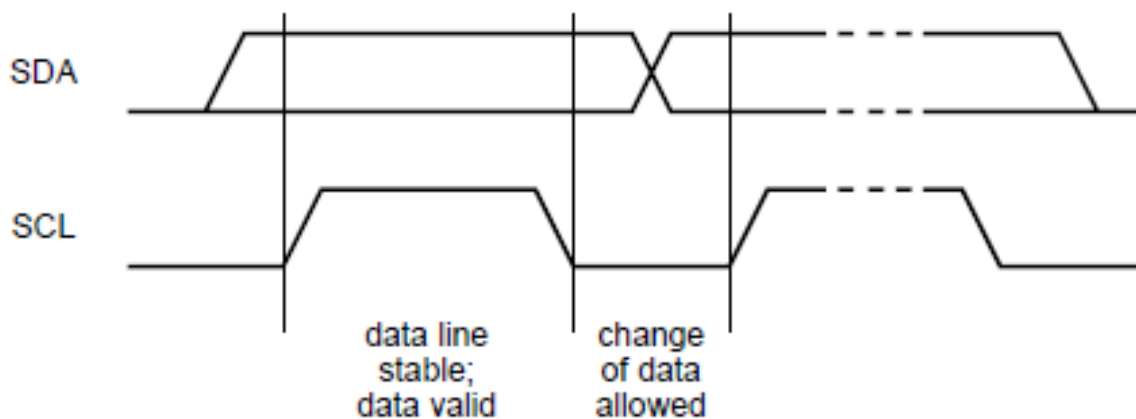


Figura 09 – Estado das linhas DAS e SCL
 Fonte: UM10204 (2007, p. 9)

Todas as transações começam com um START (S) e pode ser encerrado por um STOP (P) (ver Figura 10). A transição ALTO para BAIXO na linha SDA enquanto SCL está ALTO define uma condição de START.

Já a transição de BAIXO para ALTO na linha SDA enquanto SCL está ALTO define uma condição de STOP.

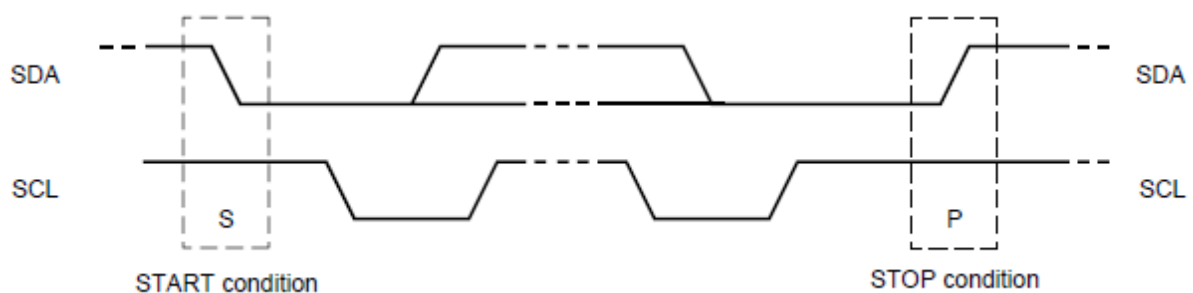


Figura 10 - Momento das transições
 Fonte: UM10204 (2007, p. 9)

Cada byte inserido na linha SDA deve ser de 8 bits. O número de bytes que podem ser transmitido por transferência é irrestrito. Cada byte tem de ser seguido por um bit *Acknowledge* (palavra inglesa que significa confirmação).

Os dados são transferidos com o Bit Mais Significativo (*MSB – Most Significant Bit*) primeiro (veja Figura 11). Se um escravo não pode receber ou transmitir outro byte completo de dados até que tenha realizado alguma outra função, por exemplo, uma interrupção de serviço interno, pode segurar a linha de clock SCL BAIXO, com isso o mestre em vigor entrará em estado de espera.

A transferência de dados, em seguida, continua quando o escravo está pronto para outro byte de dados e libera linha de *clock* SCL.

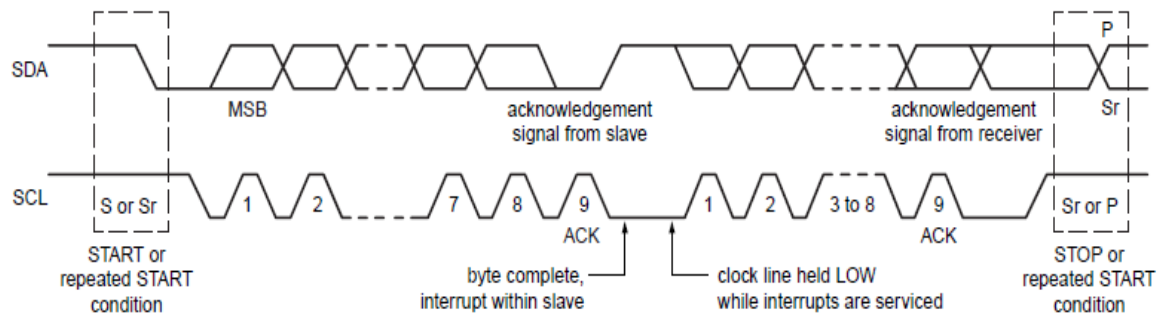


Figura 11 – Transferência com o MSB
Fonte: UM10204 (2007, p.10)

3 DETECÇÃO DE MOVIMENTO

Este capítulo aborda conceitos relacionados a detecção de movimento, suas técnicas e a aquisição da imagem no instante que ocorre o movimento na cena. A Seção 3.1 aborda a importância da detecção do movimento em uma cena e as razões desta detecção em relação ao processamento de vídeo. Já a Seção 3.2, destaca como podemos realizar a detecção do movimento no ambiente desejado. A Seção 3.3 traz alguns estudos de caso em torno deste assunto e o que hoje temos de pesquisa desenvolvida para este ramo da tecnologia.

3.1 MOVIMENTO NA CENA

De acordo com Bovik (2000), a sequência de imagem é uma fonte muito mais rica de informações visuais quando comparado a uma imagem estática, isto ocorre principalmente devido a captura de movimento, pois uma imagem estática fornece um instante de uma cena, já uma sequência de imagens registra a dinâmica do acontecimento.

O movimento é igualmente importante para o processamento de vídeo e compressão por duas razões. Primeiro, o movimento carrega várias informações sobre as relações espaço-temporais entre os objetos da imagem e esta informação pode ser usada em aplicações como monitoramento de tráfego ou vigilância de segurança, assim, pode-se identificar objetos que entram ou saem de uma cena por exemplo. Em segundo lugar, as propriedades da imagem como a intensidade ou as cores tem uma alta correlação na direção do movimento, ou seja, eles não mudam significativamente quando rastreados na imagem, por exemplo, a cor de um carro não muda quando o carro se move na cena. Isto pode ser utilizado para a remoção de redundância de vídeo temporal, isto é, em uma situação ideal apenas a primeira cena e o movimento subsequente deveriam ser transmitidos.

As aplicações acima exigem que os pontos de imagem sejam identificados como movimento ou não (no caso da vigilância, por exemplo) ou ainda que seja medido como o objeto se move (compressão e filtragem da cena). A

primeira tarefa é muitas vezes referida como detecção de movimento enquanto a segunda é apresentada como estimativa do movimento.

Estaremos enfatizando o estudo no primeiro caso, ou seja, a detecção do movimento e suas particularidades.

3.2 DETECÇÃO DO MOVIMENTO

Apesar de várias diferenças que caracterizam um determinado cenário, o movimento da imagem é uma informação crucial para a maioria deles, em particular, a detecção de movimento é relevante em aplicações onde o sistema de visualização é estático e o objetivo é identificar se houve movimento na cena observada (FORESTI; MÄHÖNEN; REGAZZONI, 2000).

Assim, o resultado de uma análise de detecção de movimento é uma máscara binária que indica a presença ou ausência de movimento para cada pixel da imagem.

A Figura 12 exemplifica a análise de um movimento em um determinado cenário:



Figura 12-a

Figura 12-b

Figura 12-c

Figura 12 – Detecção de Movimento
Fonte: Foresti; Mähönen; Regazzoni (2000, p. 59)

A Figura 12-a representa a cena de referência, ou seja, o ambiente no qual iremos detectar ou não o movimento de um objeto, já a Figura 12-b indica a cena corrente, isto é, o exato momento em que três veículos cruzam a cena de referência, e por último a Figura 12-c demonstra a diferença absoluta de intensidade

de luminância entre a Figura 12-a e 12-b relatando isto através do aglomerado mais intenso de pixels brancos.

Sendo assim, uma vez que objetos em movimento geram mudanças na intensidade da imagem, a detecção de movimento é altamente relacionada com a detecção de alterações temporais, no entanto, esta relação não é única.

Por um lado, as mudanças temporais na intensidade da imagem podem ser geradas através de ruído ou de outras causas externas, como desvios de iluminação ou sombras, por exemplo. Por outro lado, objetos em movimento geram perturbações nas mudanças temporais, um exemplo disto é a área referida como fundo descoberto, que não pertence a um objeto em movimento, mas geralmente é detectado como uma mudança temporal.

Uma área crítica em uma determinada imagem de referência ocorre quando há a sobreposição de duas imagens sucessivas de um objeto. Esta área é difícil de ser detectada como alterada, já que o objeto não é suficientemente texturizado. Os dois últimos problemas são menos críticos quando as mudanças temporais são computadas entre a imagem atual e sequencial com um quadro de referência que representa apenas o fundo da cena.

Em muitas aplicações, este cenário simplificado geralmente é viável, uma vez que muitas vezes é possível escolher a imagem de referência quando não há objetos em um primeiro plano. Outra maneira de reconstruir a imagem de fundo é integrando as informações de fundo a partir de imagens sequenciais. O uso da imagem de referência tem uma vantagem interessante para aplicações destinadas a vigilância, já que com este conceito é possível detectar novos objetos na cena, mesmo que de repente eles parem de se mover ou ainda, também é possível detectar objetos que foram removidos da cena.

Estas são características importantes na maioria das aplicações onde se trabalha com a câmera estática. No entanto, com este conceito, as sombras e reflexos gerados pelos objetos em movimento ficam destacados na cena e, frequentemente, mostram as mesmas propriedades à medida que o objeto em movimento muda de posição, com isso, há a degradação da precisão espacial dos resultados. Com o objetivo de reduzir este efeito, é possível combinar os resultados obtidos utilizando uma cena de referência e os quadros consecutivos do movimento em questão.

Ainda segundo os autores, a implementação do método proposto para análise de imagens em aplicações de vigilância está em um estágio bem avançado, sendo que a primeira etapa (detecção de movimento da imagem) é realizada quase que em tempo real (6 quadros por segundo) para sequências no formato *CIF* (sigla inglesa para *Common Intermediate Format*, cujo resolução é de 352 x 288) utilizando um computador Pentium II com processador de 300MHz. O tempo de execução exigida para a segunda etapa (estimativa do movimento) depende do conteúdo da sequência do vídeo.

Uma vez que cada objeto no vídeo seja submetido a etapa de estimativa do movimento separadamente de outros objetos, um processamento paralelo estará sensivelmente diminuindo o tempo de execução. Por exemplo, para os objetos mostrados na Figura 13, esta etapa teve um tempo médio de processamento de 0,2 segundos.

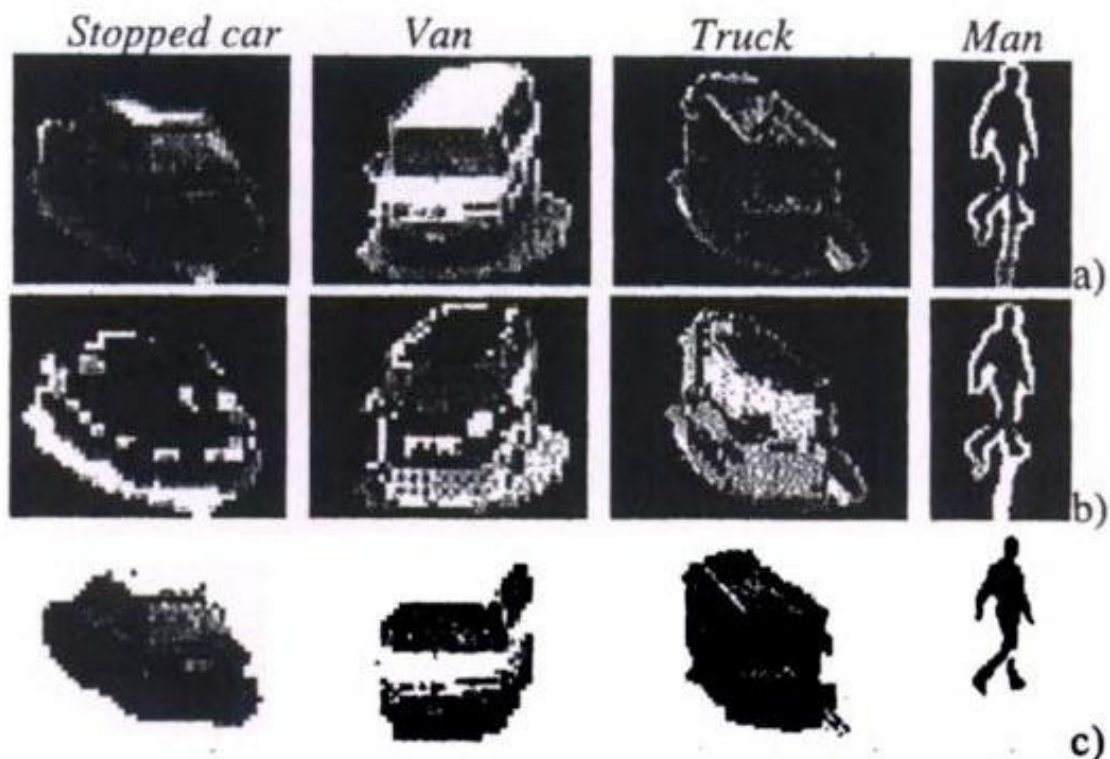


Figura 13 – Detecção do objeto na imagem
Fonte: Foresti; Mähönen; Regazzoni (2000, p. 66)

A Figura 13-a demonstra as áreas nas cenas onde houve a detecção de movimento, consequentemente, a Figura 13-b enfatiza a obtenção do agrupamento

dos pixels nas imagens e, por último, a Figura 13-c demonstra o resultado final após o processo de estimativa do movimento.

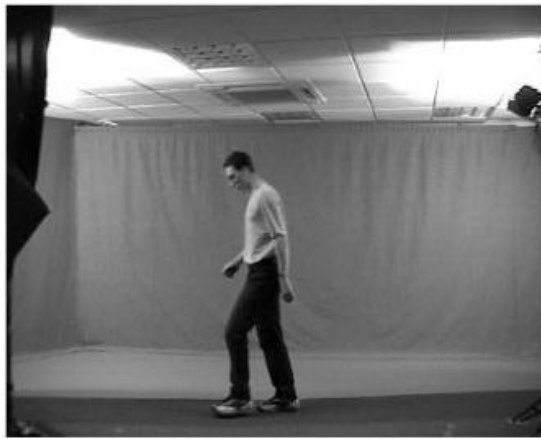
3.3 ESTUDO DE CASO

Atualmente algumas pesquisas em torno deste assunto, abordam melhorias a serem realizadas para atingir resultados mais rápidos e com mais qualidade.

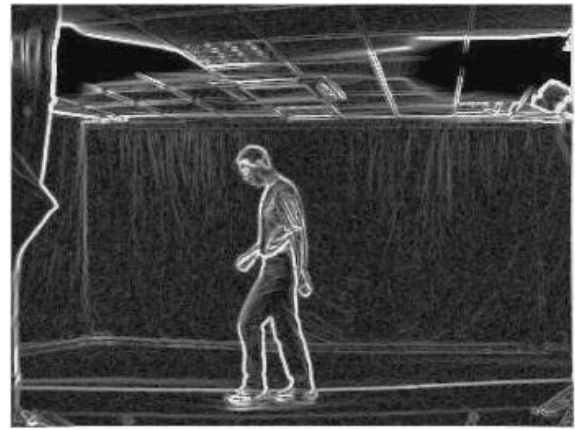
O estudo recente apresentado por Bahadir Karasulu e Serdar Korukoglu (Julho, 2011) da Universidade de Ege – Turquia, realiza um método chamado de *Background Subtraction* (BS – sigla inglesa que significa *subtração do fundo*) onde a ideia basicamente é a de subtrair o quadro atual a partir de uma imagem de fundo e assim, classificar cada pixel como primeiro ou segundo plano da imagem, comparando a diferença resultante com um limiar, porém adicionando a isto uma técnica denominada *Simulated Annealing* (SA – sigla inglesa que significa *simulação temperada*) que otimiza ainda mais a área analisada da imagem.

Os resultados obtidos pelos autores foram significativos, pois demonstram que a técnica SA embutida com a técnica BS é mais robusta e eficaz quando realizamos a detecção de movimento pura e simples de uma cena.

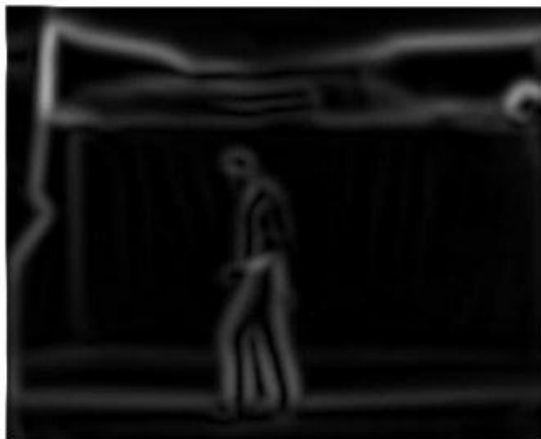
Outro trabalho publicado recentemente foi dos autores Cem Direkoglu e Mark S. Nixon (Janeiro, 2011) da Universidade de Southampton – Reino Unido, aborda a detecção de movimento baseado no fluxo de calor, onde essa técnica é utilizada para a suavização da imagem e aprimoramento. O algoritmo apresentado inicia com a difusão de calor anisotrópico no domínio espacial para remover o ruído intrínseco do dado analisado e aperfeiçoar os limites da região de borda com o propósito de obter uma alta qualidade da imagem. A Figura 14 apresenta a aplicação deste algoritmo, comparando com o calor isotrópico.



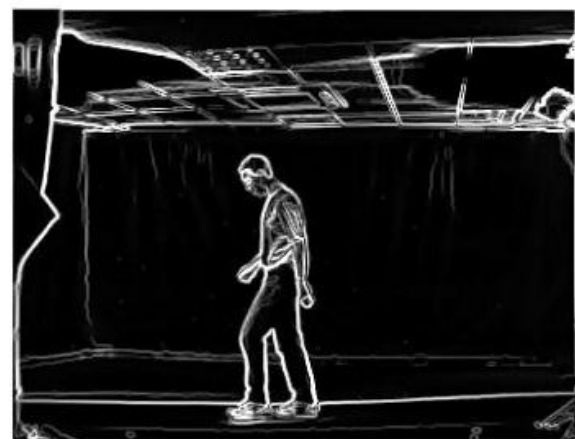
(a) Grey-scale image



(b) Original Sobel edge map



(c) Sobel edge map after isotropic diff.



(d) Sobel edge map after anisotropic diff.

Figura 14 – Aplicação do algoritmo de difusão de calor anisotrópico
 Fonte: Direkoglu e Nixon (2011)

A Figura 14-a demonstra a imagem natural, já a Figura 14-b detecta as bordas na imagem e as Figuras 14-c e 14-d enfatizam a diferença quando aplicado o algoritmo isotrópico¹ em relação ao anisotrópico², sendo este último muito mais eficiente e robusto em relação ao primeiro de acordo com os mesmos autores.

¹ Diz-se de qualquer meio transparente em que a luz atua igualmente em todas as direções.

² Diz-se de qualquer meio transparente em que a luz não atua igualmente em todas as direções.

4 DESENVOLVIMENTO DO PROJETO PROPOSTO

Este capítulo descreve o hardware implementado neste trabalho bem como a lógica embarcada. A Seção 4.1 aborda o kit de desenvolvimento do fabricante Altera® e suas interfaces. A Seção 4.2 descreve a câmera utilizada para este sistema. Já a Seção 4.3 aborda os softwares implementados para que a coleta e comparação dos frames pudessem ocorrer. Por último, a Seção 4.4 aborda a variável luminosidade e se a mesma pode ou não influenciar no cenário de testes.

4.1 KIT DE DESENVOLVIMENTO DO FABRICANTE ALTERA®

Para realizar o desenvolvimento do sistema proposto, se fez necessário primeiramente uma análise / estudo no kit de desenvolvimento do fabricante Altera®.

Neste estudo observou-se que o kit de desenvolvimento em questão possui todas as interfaces necessárias para a implementação e desenvolvimento do referido sistema. A entrada de vídeo disponível no kit permite a entrada do sinal de vídeo padrão NTSC, assim a câmera responsável pela detecção de movimento será conectada nesta entrada de vídeo.

A saída RS232 será utilizada para conectar o computador responsável pela análise e comparação entre os frames, verificando assim se houve ou não movimentação na cena.

A Figura 15 demonstra o kit de desenvolvimento do fabricante Altera® modelo EP3C25F324.

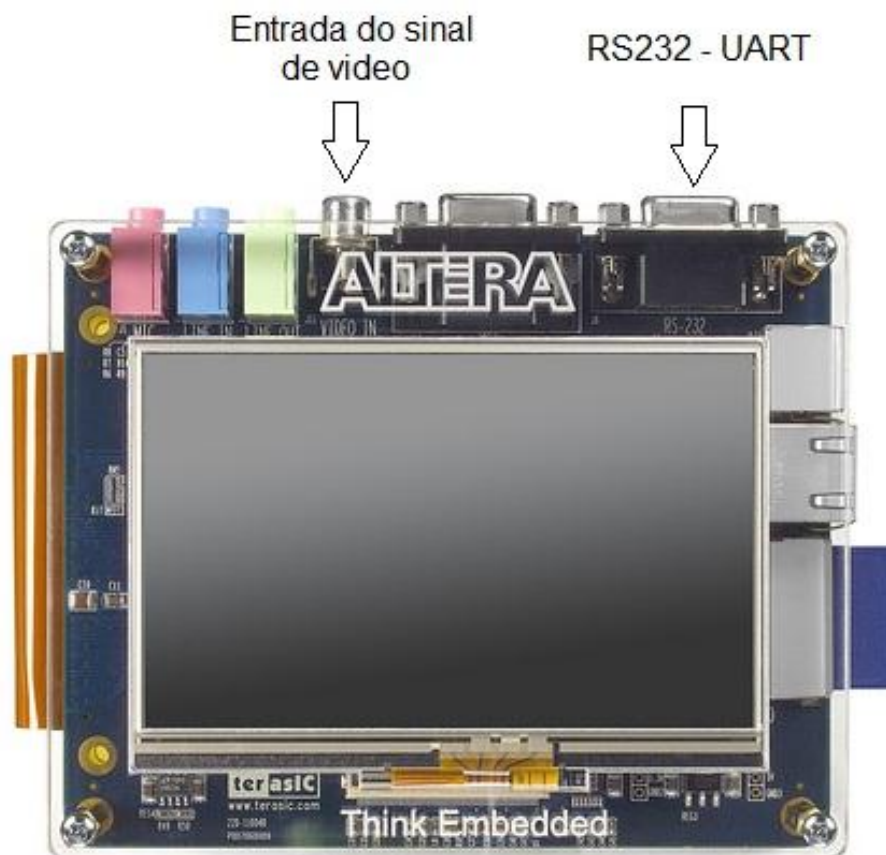


Figura 15 – Kit de desenvolvimento Altera®
 Fonte: Altera e adaptado pelo autor (2011)

4.2 CÂMERA UTILIZADA NO SISTEMA PROPOSTO

A câmera utilizada para realização do sistema proposto foi aprovada pelo sistema de qualidade descrito por Gesser (2011), com isso a mesma foi considerada apta para a utilização.

O diagrama em blocos mostrado na Figura 16 ilustra como a câmera é constituída. A imagem é colhida pelo CCD através da lente, em seguida a mesma é processada pelo DSP, sendo este componente responsável pelo processamento de imagens. Como circuitos de apoio, o DSP conta com o cristal e a *EEprom*, sendo eles responsáveis pela geração de *clock* e armazenamento do software da câmera, respectivamente.

Por último, o DSP possui uma saída do sinal de vídeo (padrão NTSC), no entanto com uma amplitude relativamente baixa. Em virtude disto, o circuito amplificador de vídeo é utilizado, elevando o sinal NTSC a um ganho de 6 dB,

tornando assim o sinal com um nível desejado para o referido padrão (em torno de 1Vpp).

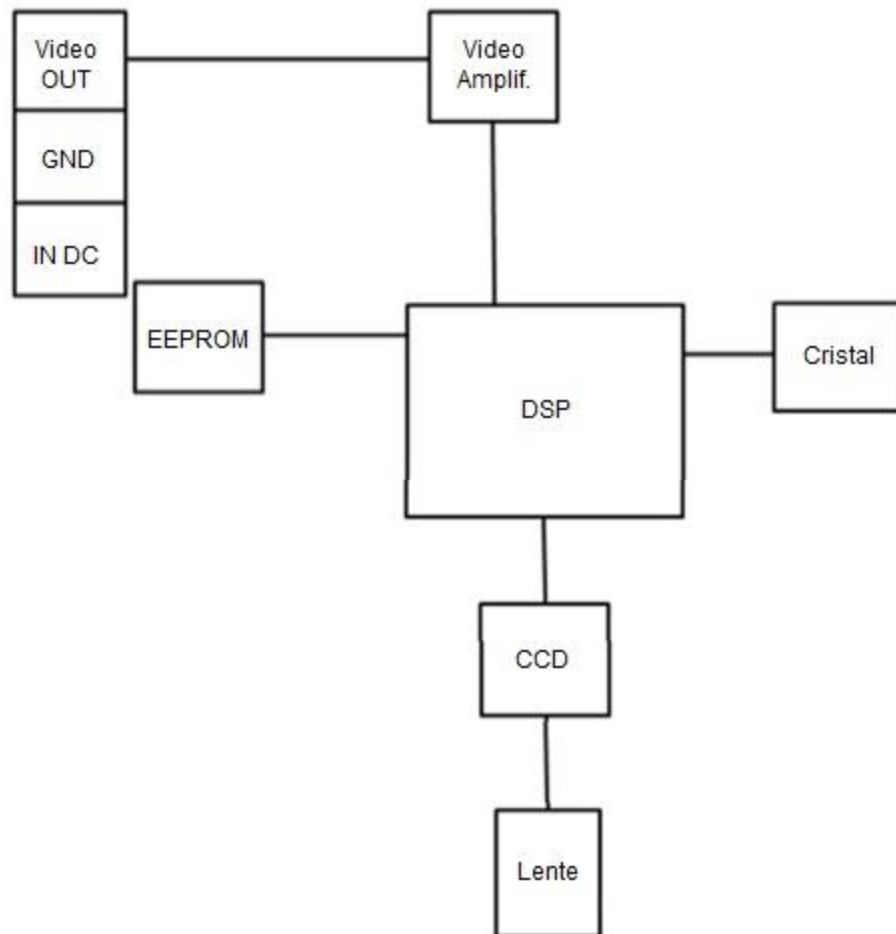


Figura 16 – Estrutura da câmera utilizada no sistema proposto
Fonte: Gesser (2011, p. 60)

4.3 SOFTWARES DESENVOLVIDOS PARA COLETAR E COMPARAR AS AMOSTRAS

Para realizar a coleta e comparação das amostras, foram utilizados dois programas desenvolvidos para este fim, sendo ambos escritos na linguagem de programação C. O software responsável pela coleta das amostras foi utilizado pelo *Soft Core Nios II* implementado no FPGA do Kit de desenvolvimento e o software responsável pela comparação foi utilizando em um computador através da ferramenta Dev C++. A Figura 17 ilustra a organização dos softwares em questão.

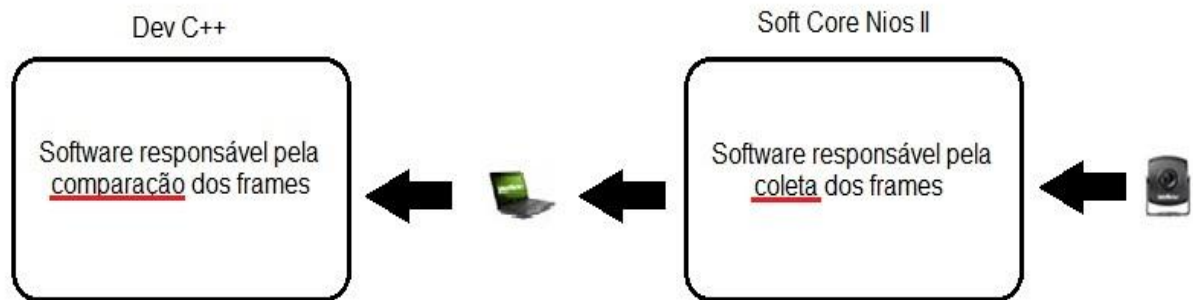


Figura 17 – Softwares para tratamento dos frames
Fonte: Elaboração do autor (2011)

O desenvolvimento destes dois programas foi necessário para otimizar o tratamento dos frames dos cenários de testes, assim a análise se tornou mais confiável e rápida para alcançar o propósito deste trabalho. Além disso, a flexibilidade dos softwares foi algo imprescindível para a coleta e comparação das amostras, pois a quantidade de pixels a ser colhida/analísada pode ser configurada conforme a necessidade do operador/usuário alterando o parâmetro Buffer.

As Figuras 18 e 19 ilustra o trecho do software utilizado pelo Soft Core Nios II e pela ferramenta Dev C++, respectivamente.

```

#define Buffer 10000 //1024
int x, temp[Buffer];
int Frame_1[Buffer];
int Frame_2[Buffer];
int Frame_Result[Buffer];
int count, per;

// OUTPUT Data Image

for(x=0;x<Buffer;x++)
{
    temp[x] = IORD(0x00100000,x);
}

for(x=0;x<Buffer;x++)
{
    printf("%x", temp[x]>>24);
    printf("%x", temp[x]>>16);
    usleep(100);
    printf("%x", temp[x]>>8);
    printf("%x", temp[x]);
    usleep(100);
}
usleep(1000 * 2000 * 2000);

```

Figura 18 – Algoritmo para coleta dos frames
Fonte: Elaboração do autor (2011)

```

//Calculo da porcentagem eh: porcentagem = (count / 9999) x 100

count = 0.0;
for(i=0;i<10000;i++)
{
    if(getc(in1) != getc(in2))
    {
        count++;
        per = ((count/9999) * 100);
        out=fopen("destino.dat", "wt");
        fprintf(out, "Arquivos %.1f por cento diferentes\n", per);
        fclose(out);
    }
    else
    {
        out=fopen("destino.dat", "wt");
        fprintf(out, "Arquivos iguais!\n");
        fclose(out);
    }
}

```

Figura 19 – Algoritmo para comparação dos frames
Fonte: Elaboração do autor (2011)

Com isso, o sistema proposto coletava e comparava as amostras, assim a prova de conceito utilizando um dispositivo de lógica programável – FPGA era alcançado. A Figura 20 demonstra o sistema proposto em sua totalidade e a Figura 21 ilustra cada etapa através de um diagrama em blocos.

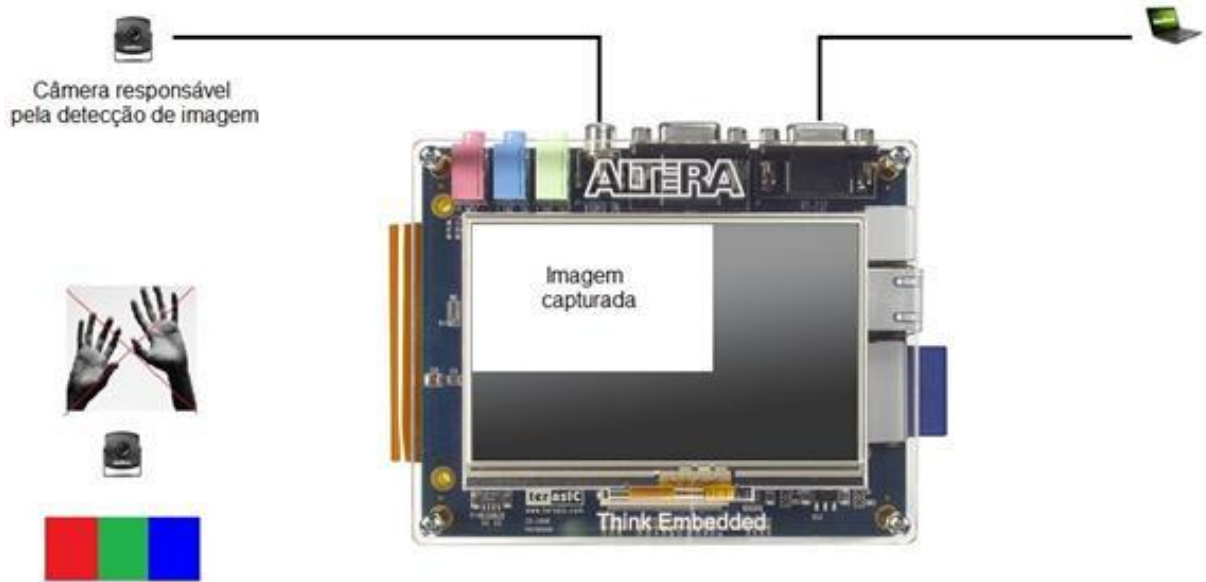


Figura 20 – Sistema proposto em sua totalidade
Fonte: Elaboração do autor (2011)

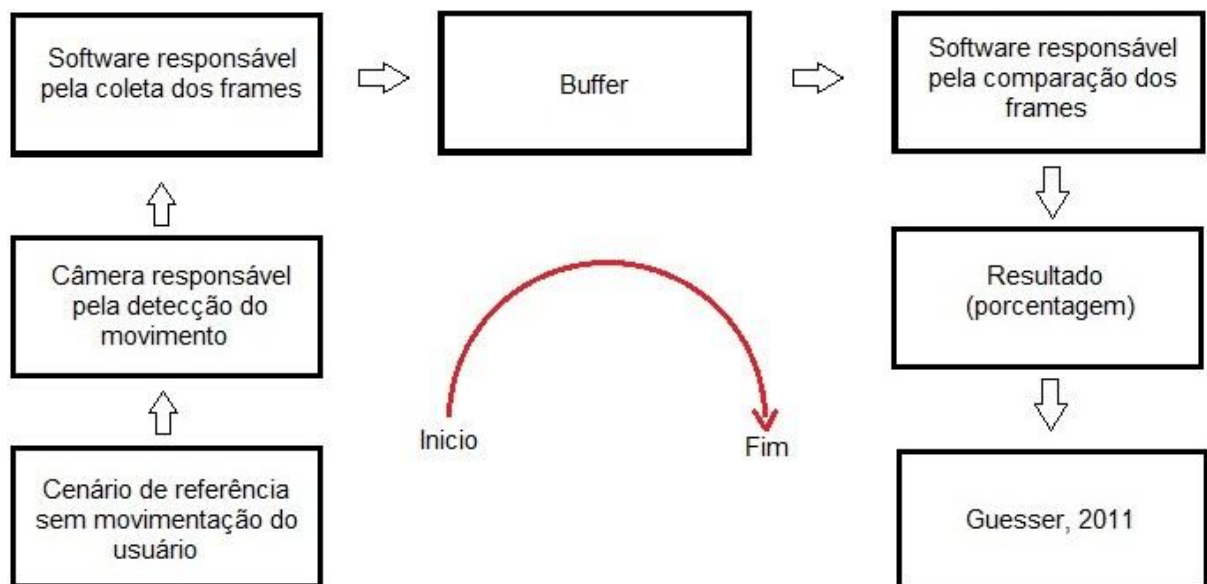


Figura 21 – Diagrama em blocos do sistema proposto em sua totalidade
Fonte: Elaboração do autor (2011)

Sendo assim, a lógica embarcada no software para realizar o dimensionamento da similaridade dos frames foi realizar a subtração de uma cena

com outra subsequente, de modo que o resultado obtido é dividido pelo número de pixels colhidos, então é possível alcançar o limite percentual para cada comparação. A Figura 22 demonstra a lógica do sistema.

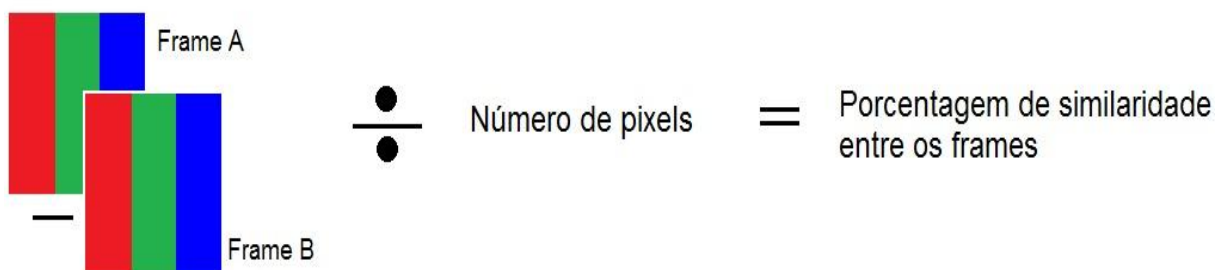


Figura 22 – Cálculo de similaridade entre as cenas
Fonte: Elaboração do autor (2011)

4.4 LUMINOSIDADE

De acordo com Guimarães (2004), o olho humano tem alguns instrumentos para compensar a diferença de luminosidade ambiente e manter a boa recepção da informação. Este controle é realizado pela íris. Quando a luz é insuficiente, os músculos da íris se contraem, alargando a pupila, e, quando a luz é mais forte, distendem-se, restringindo o seu diâmetro, variando a entrada da luz em até 30 vezes. Com isso, explica-se por que as imagens com maior iluminação exigem menor esforço da visão do que imagens com baixa iluminação.

Em virtude disto, segundo Cieszynski (2007), as câmeras eletrônicas possuem circuitos que simulam a íris, ou seja, estes circuitos tem o objetivo de ajustar o tempo de carga das células do CCD de modo a atender o nível médio de incidência de luz fazendo com que a luminosidade seja a mais ideal possível. Então, foi realizado neste trabalho uma análise baseado em amostras colhidas de um cenário de referência, ou seja, um ambiente que contenha uma boa luminosidade.

De posse das amostras coletadas, foi realizada a comparação entre os frames usando para isto um software escrito em linguagem C desenvolvido para este fim. A Tabela 1 demonstra as informações obtidas desta comparação.

Tabela 01 – Comparação dos Frames coletados no Cenário A

COMPARAÇÃO – CENÁRIO A	
AMOSTRAS	PORCENTAGEM DE DIFERENÇA (%)
1	9.3
2	9.5
3	9.3
4	9.0
5	9.7

Fonte: Elaboração do autor (2011)

A metodologia adotada para comparar os frames foi realizar a análise da amostra 1 com a amostra 2, em seguida a amostra 2 com a amostra 3 e assim por diante, até finalizar a análise comparando a amostra 5 com a amostra 1.

Com os dados da Tabela 1, foi encontrada a média de variação da luminosidade para o cenário de referência, expresso da seguinte maneira:

$$\frac{\sum \text{Porcentagem}}{\text{quantidade de amostras}} = \frac{46,8 \%}{5} = \mathbf{9,36\%} \quad \dots (1)$$

Assim, o valor obtido na Equação 1 equivale a variação, mesmo que pequena, da luminosidade para o referido cenário, ou seja, o quanto a luminosidade afeta o ambiente de referência.

5 RESULTADOS OBTIDOS

Este capítulo abordará a metodologia utilizada para a realização do trabalho proposto. A Seção 4.1 aborda os cenários no qual o sistema foi submetido, ou seja, as alterações de imagem no qual o sistema foi testado para verificar se houve ou não movimentação. A Seção 4.2 compara a variável de ambiente luminosidade com um ambiente parcialmente escuro. Já a Seção 4.3 aborda a comparação de uma imagem denominada de referência com outra imagem que possui um objeto inserido neste contexto.

5.1 CENÁRIOS DE TESTES

Visando extrair o máximo de informações para o sistema proposto a fim de obter dados suficientes para a realização de uma análise crítica sobre o tema, foram realizados três cenários distintos, conforme podemos destacar abaixo:

Cenário A: A câmera responsável pela detecção de movimento foi apontada para um quadro branco. Este ambiente denominado de “*cenário de referência*” tem o objetivo de colher os primeiros resultados e servir como base de dados para comparações futuras.

As Figuras 23 e 24 demonstram o cenário em questão e a imagem detectada pela câmera respectivamente através do kit de desenvolvimento do fabricante Altera®.

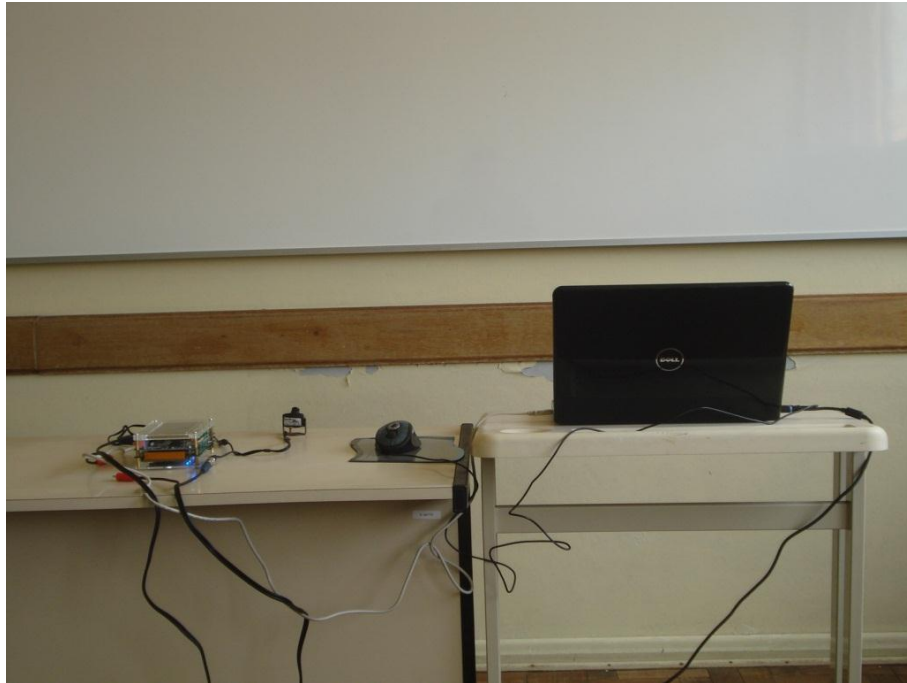


Figura 23 – Cenário de referência do sistema proposto
Fonte: Elaboração do autor (2011)

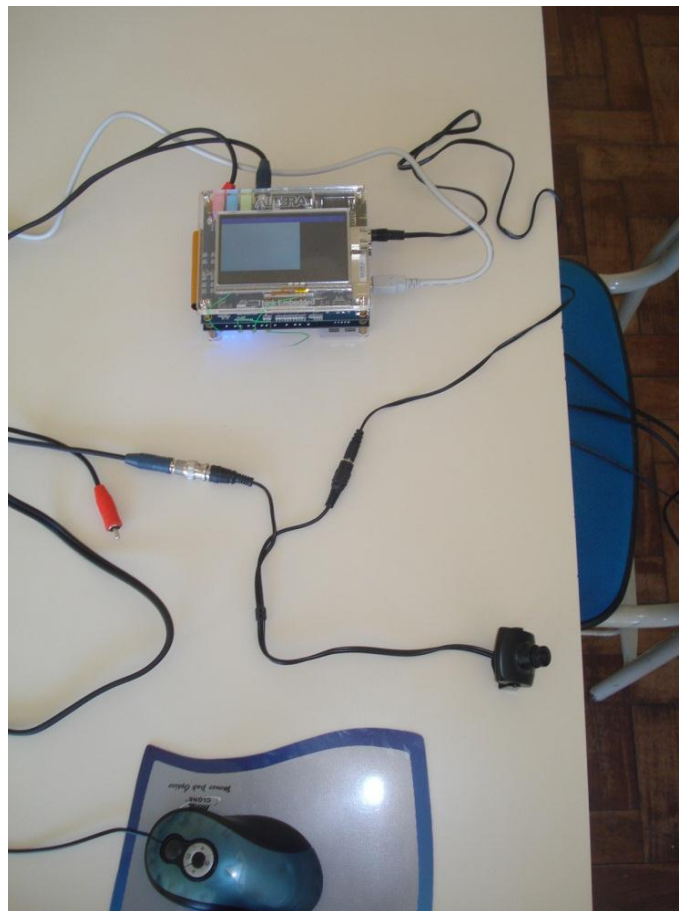


Figura 24 – Imagem de referência do sistema proposto
Fonte: Elaboração do autor (2011)

Cenário B: Foi inserido um objeto muito próximo à câmera responsável pela detecção do movimento, de modo a obstruir a incidência de luz no CCD. Com isso, foi possível obter um ambiente parcialmente escuro, simulando um cenário antagônico ao cenário A.

A Figura 25 e 26 ilustram o ambiente sugerido bem como a imagem obtida pela câmera.

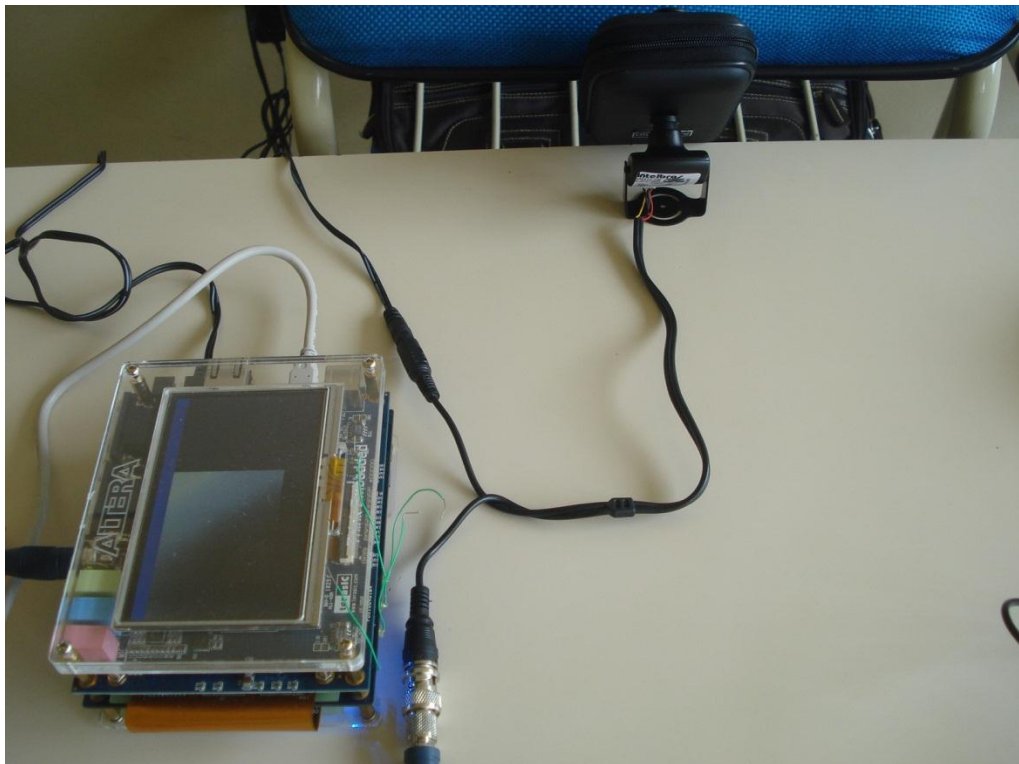


Figura 25 – Cenário parcialmente escuro
Fonte: Elaboração do autor (2011)



Figura 26 – Imagem do sistema parcialmente escuro
Fonte: Elaboração do autor (2011)

Cenário C: Para este caso foi colocado um objeto (cadeira) entre a câmera e o quadro branco de modo a obter uma terceira condição de análise. Este cenário visa checar o comportamento do sistema mediante a uma real movimentação quando comparado com o cenário de referência (Cenário A). A Figura 27 e 28 demonstra o cenário proposto bem como a imagem captada pela câmera.



Figura 27 – Cenário com objeto entre a câmera e o quadro branco
Fonte: Elaboração do autor (2011)



Figura 28 – Imagem com objeto entre a câmera e o quadro branco
Fonte: Elaboração do autor (2011)

Foram extraídas cinco amostras de cada cenário de testes (A, B e C), onde cada amostra possui a quantidade de 10.000 pixels que são armazenados em um buffer para, posteriormente, serem analisados em um computador. O objetivo desta coleta é alcançar uma média aritmética das medidas efetuadas para fins de

cálculos e comparações futuras. Esta amostragem foi necessária, pois ao realizar os experimentos, observou-se uma variação entre os frames quando comparados.

5.2 COMPARAÇÃO ENTRE O CENÁRIO A COM O CENÁRIO B

Seguindo o mesmo raciocínio da Seção 4.4, foi realizada a comparação de um cenário de referência, representado pelo cenário A, com um cenário que contenha uma luminosidade parcial, representado pelo cenário B.

Assim, foi analisado o comportamento do sistema quando submetido ao cenário B, de modo a verificar se a falta parcial de luminosidade irá afetar os frames coletados. A Tabela 2 demonstra a relação de comparação dos frames colhidos no cenário B.

Tabela 02 – Comparação dos Frames coletados no Cenário B

COMPARAÇÃO – CENÁRIO B	
AMOSTRAS	PORCENTAGEM DE DIFERENÇA (%)
1	3.2
2	3.6
3	2.9
4	3.3
5	3.8

Fonte: Elaboração do autor (2011)

Foi realizada a mesma estratégia de comparação do cenário A, assim a média de variação da luminosidade para o cenário B é expressa na equação 2:

$$\frac{\sum \text{Porcentagem}}{\text{quantidade de amostras}} = \frac{16,8 \%}{5} = 3,36\% \quad \dots (2)$$

Através do valor obtido na Equação 2, pode-se concluir que a falta de luminosidade no ambiente favorece a relação de igualdade entre os frames, ou seja, as amostras coletadas possuem uma margem de erro de 3,36% umas das outras,

enquanto para o cenário A no qual possui uma luminosidade maior esta margem de erro passa para 9,36%.

5.3 OBJETO NO CENÁRIO DE REFERÊNCIA

Após realizar a análise dos cenários A e B, verificando assim a questão da luminosidade, voltou-se o foco para o cenário C com as cinco amostras coletadas anteriormente. O procedimento realizado tem como objetivo comparar as imagens extraídas do cenário A (referência) com as imagens do cenário C de modo que seja possível rastrear um limite percentual entre as duas cenas, estipulando assim um critério para a detecção de movimento ou não.

O cálculo deste percentual foi realizado comparando a amostra 1 do cenário A com a amostra 1 do cenário C, a amostra 2 do cenário A com a amostra 2 do cenário C e assim por diante, até concluir o cálculo com a amostra 5 de cada cenário. A Tabela 3 evidencia a relação percentual das amostras do cenário A e cenário C:

Tabela 03 – Comparação dos Frames dos cenários A e C

COMPARAÇÃO		
AMOSTRAS – CENÁRIO A	AMOSTRAS – CENÁRIO C	PORCENTAGEM DE DIFERENÇA (%)
1	1	59.2
2	2	59.4
3	3	59
4	4	59.5
5	5	60.2

Fonte: Elaboração do autor (2011)

Com base nos números obtidos, realizou-se a média aritmética para estipular o limite de sensibilidade do sistema. A Equação 3 demonstra esta medida:

$$\frac{\sum \text{Porcentagem}}{\text{quantidade de amostras}} = \frac{297,3 \%}{5} = 59,46\% \quad \dots (3)$$

Através dos resultados obtidos acima, pode-se perceber que apesar do grau de sensibilidade do sistema ser considerado alto (59,46%), o sistema proposto detectou a movimentação da cadeira no cenário C em relação ao cenário A (referência) e assim a comparação entre os frames tornou-se válida, tendo em vista a diferença expressiva entre as cenas.

Com base nas informações colhidas após a conclusão do sistema, no estudo bibliográfico realizado e no conhecimento adquirido com o kit de desenvolvimento do fabricante Altera®, é possível realizar uma análise sucinta e criteriosa do ambiente em que o sistema foi submetido.

O limite percentual de detecção de movimento para os cenários apresentados, apesar de ser um número expressivo ($\cong 59\%$), não está considerando a variável luminosidade conforme demonstrado na Seção 4.4. O Gráfico 1 mostra com exatidão o real limite percentual quando comparamos o cenário de referência puro com outro cenário que contém um objeto neste mesmo cenário de referência.

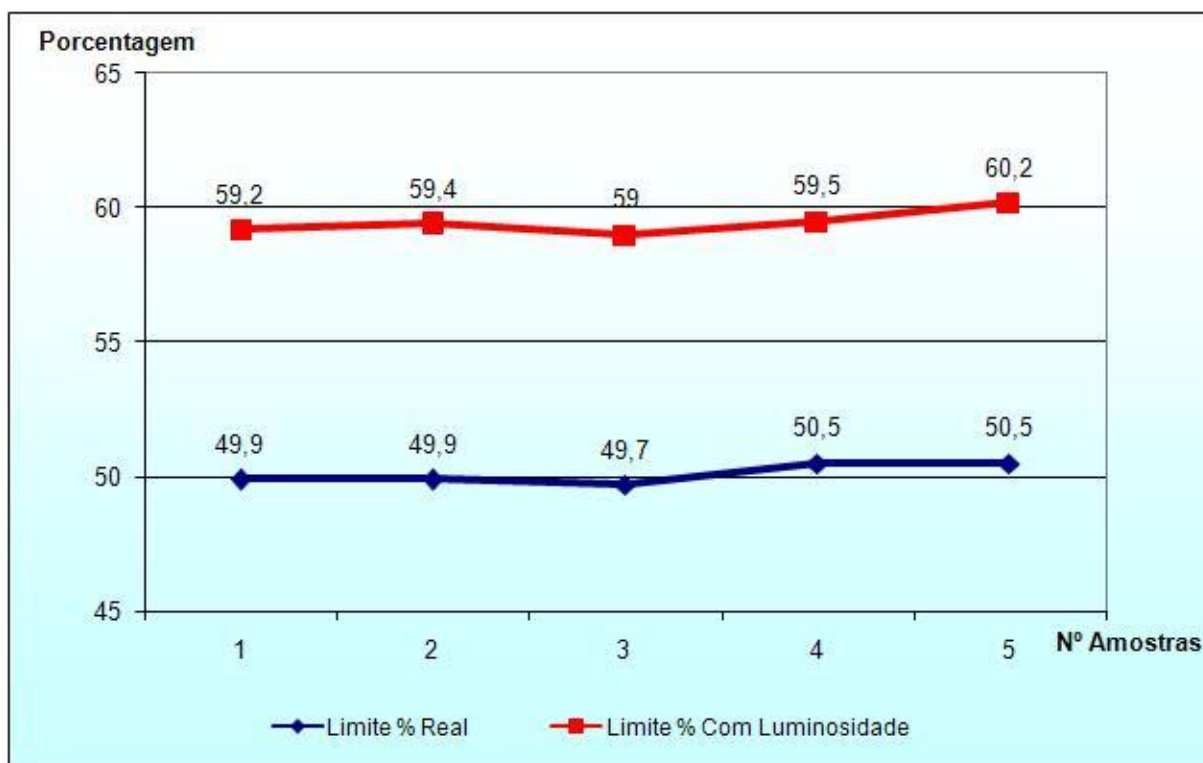


Figura 29 – Gráfico Comparativo Limite Real e Limite com Luminosidade
Fonte: Elaboração do autor (2011)

Em suma, o limite percentual próximo de 50% para a detecção de movimento é aceitável para o ambiente de testes, já que em uma linha de produção teremos

movimentos acintosos e que, conseqüentemente, serão detectados para este nível de sensibilidade.

6 TRABALHOS FUTUROS

A seguir as sugestões para trabalhos futuros que podem ser desenvolvidos a partir deste:

- Realizar a verificação do comportamento do sistema quando submetido a variação do tamanho do objeto sob análise;
- Desenvolver uma metodologia de testes capaz de diminuir o limite percentual para a detecção de movimento, fazendo com que o sistema seja mais sensível a qualquer alteração do cenário de testes;
- Estudar outra forma além da subtração de frames para que o movimento em uma cena seja percebido, abrindo assim um leque de possibilidades para melhorias futuras.

7 CONSIDERAÇÕES FINAIS

Os objetivos deste trabalho foram atingidos, já que a busca de referências bibliográficas sobre o tema possibilitou embasamento para que a técnica de subtração entre os frames pudesse ser aplicada, além disto, o conhecimento adquirido com o kit de desenvolvimento do fabricante Altera® e a vivência com a linguagem C de programação para a construção dos softwares foi de suma importância para o crescimento acadêmico e profissional, já que para este nicho de mercado no qual o trabalho foi desenvolvido é necessário uma aprendizagem em diversas áreas do conhecimento.

A metodologia de pesquisa, testes e verificação dos resultados é outro ponto que merece destaque. Através dos procedimentos descritos neste trabalho foi possível analisar o comportamento do sistema proposto para os cenários de testes sugeridos onde concluiu-se que: a luminosidade é uma variável de ambiente que influencia no cenário no qual se deseja trabalhar e que, apesar do limite percentual para a detecção do movimento ser um número expressivo. Este sistema pode ainda ser melhorado com a realização de mais testes, como por exemplo, variando o tamanho do objeto a ser detectado ou ainda, checar se a resolução da câmera pode ou não influenciar na detecção do movimento.

Com isso, futuramente, tal sistema pode tornar-se um produto para as indústrias com um custo relativamente menor se comparado com outros equipamentos existentes no mercado, como por exemplo, o equipamento chamado Vectorscope, que possibilita a mensuração de níveis de cores, entre outros parâmetros.

No mais, a prova de conceito proposta neste trabalho para detecção de movimento utilizando um dispositivo de lógica programável – FPGA foi concluída e validada, conforme destacado no capítulo 5.

REFERÊNCIAS

ANALOG DEVICE, Application Note. Disponível em: <http://www.analog.com/static/imported-files/data_sheets/ADV7180.pdf>. Acesso em: 28 jul. 2011.

BOVIK, Al. **Handbook of image and video processing**. Canada: Academic Press, 2000.

BROWN, Stephen; VRANESIC, Zvonko. **Fundamentals of digital logic with vhdl design**. 3. ed. EUA: McGraw-Hill, 2008.

DAMJANOVSKI, Vlado. **cctv networking and digital technology**. EUA: Elsevier, 2005.

CIESZYNSKI, Joe. **Closed circuit television**. 3. ed. Publisher: EUA: Elsevier, 2007.

DIREKOGLU, Cem; NIXON, Mark S. Moving-Edge Detection via Heat Flow Analogy, **Pattern Recognition Letters**. Jan./2011, v. 32, n. 2, p. 270-279.

FORESTI, Gian Luca; MÄHÖNEN Petri; REGAZZONI, Carlos S. **Multimedia video-based surveillance systems: requirements, issues and solutions**. EUA: Kluwer Academic, 2000.

GESSER, Kelly Cristini. **Sistema de verificação de cores no modelo RGB para câmeras de CFTV utilizando dispositivos de lógica programável**. 2001. Trabalho de Conclusão de Curso – Universidade do Sul de Santa Catarina, Brasil, 2011.

HARWOOD, Emily. **Digital CCTV: a security professional's guide**. EUA: Elsevier, 2008.

HEURING, Vincent P.; MURDOCCA, Miles J, **Introdução a arquitetura de computadores**. Rio de Janeiro: Campus, 2001.

KARASULU, Bahadir; KORUKOGLU, Serdar. **A Software for performance evaluation and comparison of people detection and tracking methods in video processing**. Disponível em: <http://efe.ege.edu.tr/~karasulu/vicampev/SuppMaterial_AppendixA.pdf>. Acesso em: 29 jul. 2011.

MONTEIRO, Mario Antonio. **Introdução a organização de computadores**. 5. ed. São Paulo: LTC, 1996.

OLIVEIRA, Isaura Nelsivânia Sombra. **Metodologia para detecção rápida de movimentos em sequências de imagens**. 2003. 191 f. Tese (Doutorado Física Aplicada) Universidade de São Paulo, Brasil, 2003.

PESSOA, Fernando. **Pedras no caminho? Guardo todas, um dia vou construir um castelo**. Disponível em: <<http://www.fotolog.com.br/angelprazeres/35871264>>. Acesso em: 29 set. 2011.

UM10204, Application Note. Disponível em: <http://www.nxp.com/documents/user_manual/UM10204.pdf>. Acesso em: 29 jul. 2011.

VAHID, Frank; GIVARDIS, Tony. **Embedded SYSTEM DESIGN**: a unified hardware/software introduction. EUA: Wiley, 2002.