

Jhony Andreatta Custódio da Silva

Sistema Embarcado *Lowcost* de Contagem Automática de Pessoas

Vitória - ES, Brasil

09 de Janeiro de 2018

Jhony Andreatta Custódio da Silva

Sistema Embarcado *Lowcost* de Contagem Automática de Pessoas

Parte manuscrita do Projeto de Graduação do aluno **Jhony Andreatta Custódio da Silva**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Universidade Federal do Espírito Santo - UFES

Centro tecnológico

Programa de Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Jair Adriano Lima Silva

Vitória - ES, Brasil

09 de Janeiro de 2018

Jhony Andreatta Custódio da Silva

Sistema Embarcado *Lowcost* de Contagem Automática de Pessoas/ Jhony Andreatta Custódio da Silva. – Vitória - ES, Brasil, 09 de Janeiro de 2018.

Orientador: Prof. Dr. Jair Adriano Lima Silva

– Universidade Federal do Espírito Santo - UFES

Centro tecnológico

Programa de Graduação em Engenharia Elétrica, 09 de Janeiro de 2018.

I. Jair Adriano Lima Silva. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Sistema Embarcado *Lowcost* de Contagem Automática de Pessoas.

CDU 621.3

Jhony Andreatta Custódio da Silva

Sistema Embarcado *Lowcost* de Contagem Automática de Pessoas

Parte manuscrita do Projeto de Graduação do aluno **Jhony Andreatta Custódio da Silva**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Trabalho aprovado. Vitória - ES 22 de Dezembro de 2017.

Prof. Dr. Jair Adriano Lima Silva
Orientador

Prof. Dr. Hélder Roberto de Oliveira Rocha
Universidade Federal do Espírito Santo

Ms. Pablo Rafael Neves Marciano
Universidade Federal do Espírito Santo

Vitória - ES, Brasil
09 de Janeiro de 2018

Agradecimentos

Primeiramente a Deus por ter permitido que tudo isso acontecesse, por ter me dado saúde e força para superar as muitas dificuldades.

Agradeço a minha mãe Renilda Maria, meu pai Rogério Custódio e minha namorada Karina Assini, pelo apoio incondicional, por ter me incentivado e me ajudado nos momentos mais difíceis, por sempre me manter motivado a continuar, apesar das dificuldades.

A todos meus amigos da Universidade, do projeto Vitória Baja, da Empresa CTjúnior, do CEFD , da igreja e aos companheiros de trabalhos que fizeram parte da minha formação.

À equipe Empresa Sinales - Sinalização do Espírito Santo pela oportunidade, apoio e suporte na elaboração deste trabalho.

Ao Prof. Dr. Jair Adriano Lima Silva, pela solicitude, boa vontade e empatia.

Finalmente, a todos que direta ou indiretamente fizeram parte da minha formação.

Resumo

O presente projeto aborda o desenvolvimento de um sistema de *software* e sua implementação em *hardware*, para a contagem automática de pessoas, utilizando-se de processamento e análise de imagens capturadas por um computador de baixo custo *Raspberry pi - model B*. Na contra-mão da maioria dos sistemas encontrados nas literaturas e aplicações comerciais, os quais são baseados na utilização de sensores fotoelétricos, infravermelhos ou sensores de pressão, o sistema projetado neste trabalho baseia exclusivamente na análise de câmeras de vídeo instaladas e configuradas no local desejado.

Tal implemetação, até o momento, consiste em um *hardware* (câmera, computador e conexões) e um *firmware*. Testes realizados com o protótipo até agora sugerem a implementação bem sucedida do *firmware* no computador utilizado para coleta dos dados provenientes da câmera

Palavras-chave: *Raspberry Pi*. monitoramento de pedestres. Protótipo. Contagem de pessoas.

Abstract

This project addresses the development of a software system and its implementation in hardware, for the automatic counting of people, using processing and analysis of images captured by a low-cost computer Raspberry pi-model B. In contrast to most of the systems found in commercial literature and applications, which are based on the use of photoelectric sensors, infrared sensors or pressure sensors, the system designed in this work relies exclusively on the analysis of video cameras installed and configured at the desired location.

This implementation, so far, consists of a hardware(camera, computer and connections) and a firmware. Testing with the prototype so far suggests the successful implementation of firmware on the computer used to collect data from the camera

Keywords: Raspberry Pi. Pedestrian detector. Prototype. People counter.

Lista de ilustrações

Figura 1 – Ambiente de análise	13
Figura 2 – Escala de cinza na forma matricial	18
Figura 3 – Detecção da região de bordas	20
Figura 4 – Imagem Limiarizada <i>Threshold 196</i>	21
Figura 5 – (acima à esquerda) Imagem original. (acima à direita) Distância. (abaixo à esquerda) <i>Background</i> . (abaixo à direita) <i>Foreground</i>	22
Figura 6 – Resultado da aplicação do Filtro da Mediana	24
Figura 7 – Resultado da aplicação do Filtro de Sobel	25
Figura 8 – Exemplo de aplicação de um Filtro Homomórfico	27
Figura 9 – Medidas Antropométricas	28
Figura 10 – Base de sistemas de computação	30
Figura 11 – <i>Raspberry Pi Model B</i>	31
Figura 12 – <i>Raspberry Pi Camera</i>	34
Figura 13 – Conexão via <i>flat cable</i> placa/câmera	34
Figura 14 – Adaptador HDMI - VGA	35
Figura 15 – Circuito do Adaptador HDMI - VGA	35
Figura 16 – Teclado	36
Figura 17 – <i>Mouse</i>	36
Figura 18 – <i>Circuito externo de acionamento</i>	38
Figura 19 – Cartão de Memória	39
Figura 20 – Fonte chaveada	40
Figura 21 – Exemplo de aplicação da escala de cinza	44
Figura 22 – Exemplo da aplicação do Filtro Gaussiano	45
Figura 23 – Linhas bases de referência usadas na detecção da direção do movimento.	46
Figura 24 – Ambiente - Teste 1	48
Figura 25 – Acurácia - Teste 1	49
Figura 26 – Ambiente - Teste 2	50
Figura 27 – Acurácia - Teste 2	50
Figura 28 – Ambiente - Teste 3	51
Figura 29 – Acurácia - Teste 3	52

Lista de tabelas

Tabela 1 – Exemplo de Filtro da Mediana	24
Tabela 2 – Comparação entre as principais plataformas de desenvolvimento	33
Tabela 3 – Resultados do primeiro teste realizado	49
Tabela 4 – Resultados do segundo teste	49
Tabela 5 – Resultados do terceiro teste realizado	51
Tabela 6 – Custo do projeto (excetuando a mão de obra e pesquisa)	53
Tabela 7 – Comparação entre produtos similares disponíveis no mercado	53

Lista de abreviaturas e siglas

SINALES	Sinalização do Espírito Santo
UFES	Universidade Federal do Espírito Santo
GPIO	General Purpose Input/Output
LED	Light Emitting Diode
SAE	Sociedade de Engenheiros da Mobilidade
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
HDMI	<i>High-Definition Multimedia Interface</i>
VGA	<i>Video Graphics Array</i>
REV	Revisão
CEFD	Centro de Educação Física e Desportos
CSI	<i>Camera Serial Interface</i>
GPRS	<i>General Packet Radio Service</i>
FPS	<i>Frame Per Second</i>
MRF	<i>Markov Randomm Field</i>
USP	Universidade de São Paulo
LCAD	Laboratório de Computação de Alto Desempenho
FBBB	Filtro Passa Banda de <i>Butterworth</i>
DIY	<i>Do it Yourself</i>
EEPROM	<i>Electrically-Erasable Programmable Read-Only Memory</i>
RAM	<i>Random Access Memory</i>
PWM	description <i>Pulse Width Modulation</i>
SPI	<i>Serial Peripheral Interface</i>
WIFI	<i>Wireless Fidelity</i>

Lista de símbolos

α	Letra grega minúscula alfa
β	Letra grega minúscula beta
γ	Letra grega minúscula gama
μ	Letra grega minúscula mi
Ω	Unidade de medida da resistência elétrica (Ohm)

Sumário

1	INTRODUÇÃO	13
1.1	Contextualização	13
1.2	Objetivo	15
1.3	Descrição do Problema Proposto	16
1.4	Organização da monografia	16
2	REFERENCIAL TEÓRICO	18
2.1	Introdução a Teoria de Imagens	18
2.2	Segmentação	19
2.3	Deteccção e Análise de Movimento em Vídeos	21
2.4	Filtragem	22
2.4.1	Filtros lineares	23
2.4.2	Filtros não-lineares	24
2.4.3	Filtragem Homomórfica	26
2.5	Atributos humanos	27
2.5.1	Modelo de Dimensões	28
2.5.2	Média de Velocidade	28
3	IMPLEMENTAÇÃO	30
3.1	Hardware	30
3.1.1	Raspberry Pi Model B	31
3.1.2	Raspberry Pi Camera	32
3.1.3	Adaptador HDMI para VGA	34
3.1.4	Teclado	36
3.1.5	Mouse	36
3.1.6	Circuito de Acionamento	38
3.1.7	Cartão <i>microSD</i>	39
3.1.8	Fonte de alimentação	40
3.1.9	Bateria	41
3.2	Firmware	41
3.2.1	Captura do Sinal de Acionamento	42
3.2.2	Captura dos <i>frames</i> via <i>Raspberry Pi</i>	43
3.2.2.1	Captura	43
3.2.3	Processamento	44
3.2.3.1	Lógica	45
3.3	Software	46

4	TESTES E RESULTADOS	48
4.1	Primeiro Teste	48
4.2	Segundo Teste	49
4.3	Terceiro Teste	50
5	CONCLUSÃO	53
	REFERÊNCIAS	55

1 Introdução

1.1 Contextualização

Este relatório aborda a problemática de contabilizar por estimativas a quantidade de pessoas que passam por um determinado local ao longo do dia, utilizando-se de variadas técnicas como processamento digitais de imagens, análise de vídeos por quadro de imagens, entre outras. O sistema aqui abordado foi criado com a intenção de capturar e contabilizar o fluxo de pedestres em uma via urbana, porém para evitar a burocracia de fechamento de uma via urbana para realização de teste, os testes serão feitos em um ambiente controlado (Figura 1).

Figura 1 – Ambiente de análise



Fonte: Produção do próprio autor.

Alguns dos principais desafios nos ambientes de detecção são:

- Pessoas com necessidades especiais (como cadeira de rodas);
- Casais;
- Crianças acompanhada dos pais;
- Sombras que variam de acordo com o passar do dia;
- Reflexos;
- Pessoas com carrinhos, ou caixas grandes;
- Ciclistas.

A parte majoritária dos projetos realizados nesta área são baseados em técnicas de subtração de plano de fundo do quadro, para que assim possa ser detectado o movimento, para isso é usado a limiarização de imagens baseados em alguns autores renomados que estudam processamento digitais de sinais.

- Bhuvaneshwari, 2004
- Papanikolopoulos, Nikolaos, 2001

Estimativa de fluxo de pedestres em tempo real, visa principalmente atender a demanda crescente no setor empresarial, visto que cada dia mais se dispõe de novos aparelhos e dispositivos com grande capacidade de processamento e qualidade de imagens, pode-se evoluir para o sistema de processamento digital de imagens e vídeos em contramão as tecnologias ultrapassadas comumente usadas como sensores fotoelétricos, sensores ultrassônicos e de pressão como exemplos.

Em alguns estabelecimentos, tais como, *shoppings*, lojas varejistas, centros comerciais, sua função principal seria fornecer aos interessados informações sobre o fluxo de pessoas em determinados pontos, podendo assim usar destas informações para um melhor gerenciamento de serviços, propaganda, controle de velocidade de escadas rolantes, entre outros.

Em áreas externas pode usar a tecnologia de contagem de pessoas para várias aplicações, como :

- Sistema de controle de tráfego inteligente;
- Contagem de pessoas que atravessam fora da faixa de pedestres;
- Local ótimo para instalação de faixa de pedestres;

- Quantidade de pedestres que cruzam a via, para uma possível estimativa do tempo para controle semafórico.

Com uma obtenção de dados precisa, *softwares* e câmeras de detecção confiáveis, aliado a uma base teórica sólida, as possibilidades de criação de produtos e/ou processos mais eficientes em diversas áreas de aplicação são inúmeras.

O presente trabalho foca em engenharia de tráfego, sendo este realizado com o intuito de contagem de fluxo de pessoas em tempo real em vias urbanas, bem como passarelas e túneis. O trabalho baseou-se no uso de técnicas consideravelmente simples, que seriam:

- Subtração de plano de fundo;
- Filtro Gaussiano;
- *GrayScale*;
- Redimensionamento de quadros.

Os testes foram realizados e apresentam um resultado satisfatório em relação aos métodos antigos e ultrapassados de contagem existente e comumente usadas na indústria, mostrando assim que o sistema é promissor e a acuracidade da contagem tem sido em média de 87,17 % sendo esta uma versão protótipo, como mencionado anteriormente, cabe citar que o criador tem plena ciência que para produção comercial, várias melhorias e aperfeiçoamentos deverão ser implantados, visando assim aumentar o valor agregado ao produto bem como os serviços oferecidos.

Neste projeto são apresentados conceitos de visão computacional, que nada mais é uma ciência que modela e procura replicar a visão humana através de *softwares* e *hardwares*.

1.2 Objetivo

Ao final deste projeto, pretende-se avaliar os dados através do *software Microsoft Excel*, que serão adquiridos através de testes realizados no ambiente controlado da empresa SINALES, visando melhorar processos arcaicos usados até então. Os objetivos são:

1. Desenvolver o *firmware* via *python*, que será implementado em um computador de uso geral *Raspberry Pi model B*, conectado a uma Câmera Pi NoIR.
2. Desenvolver um *software* para captura e análise de imagens.
3. Montar e instalar no local de testes o protótipo.

4. Desenvolver um modelo de produto.
5. Realização dos testes em um local controlado.
6. Análise dos resultados através do *Microsoft Excel*.

1.3 Descrição do Problema Proposto

Esta monografia propõe o desenvolvimento e construção de um modelo embarcado de contagem de pessoas em tempo real, para uso diário com sistema de baterias, podendo ser alterado para uso semanal ou mensal com auxílio de painéis fotovoltaicos ou carregadores de baterias conectados a uma rede de tensão alternada comum.

Para que o projeto possa ser realizado algumas restrições foram inseridas com o intuito de garantir a confiabilidade das medições, bem como assegurar que o prazo de execução seja cumprido. São estas as restrições impostas:

- O instalador deve possuir pleno conhecimento do produto que será instalado;
- A iluminação deve ser natural, evitando-se fixar o produto em áreas com iluminação forte incidente;
- O sistema não deve ser instalado em locais de multidões;
- A superfície deve ser fosca ou opaca e sem a presença de reflexos;
- O produto deve respeitar a altura e ângulo de captura máximo de acordo com especificações previamente definidas.

1.4 Organização da monografia

- No [Capítulo 2](#) é apresentada uma revisão bibliográfica sobre introdução a teoria de visão computacional, filtros, captura de sinais, conceitos de sistemas embarcados e plataformas de desenvolvimento como *Raspberry Pi*.
- No [Capítulo 3](#) é apresentada a estrutura da implementação do sistema embarcado no local escolhido. Discorre sobre todos os componentes eletrônicos utilizados: plataformas, módulos, baterias, estrutura e conexões. É apresentado também o desenvolvimento do *firmware* e o tipo de *software* utilizado para análise dos resultados.
- No [Capítulo 4](#) é apresentada a validação experimental com a descrição dos experimentos, resultados e a discussão.

- No [Capítulo 5](#) são apresentados as conclusões, as contribuições desta monografia e os trabalhos futuros.

2 Referencial Teórico

Com o crescente aumento nos sistemas de visão computacional, processamento digital de sinais e imagens, está ocorrendo um aumento significativo de tecnologias na área, aumentando assim a sua relevância, bem como aumento do uso dessas tecnologias. A visão computacional procura processar o conteúdo de uma imagem, visando transformá-la em um sinal bidimensional ou multidimensional para que possam ser usadas em sistemas de processamento.

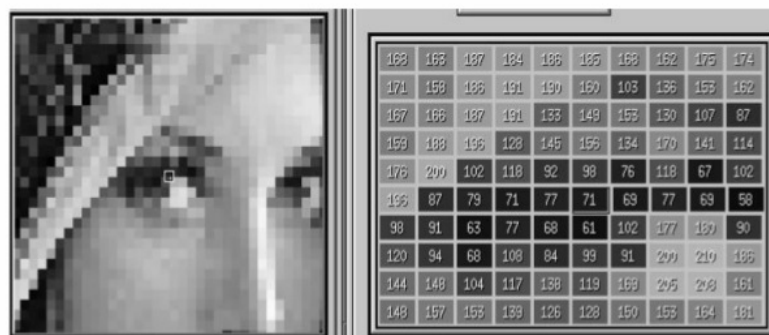
Grande parte dos estudos e trabalhos feitos nesta área são para extrair dados através de métodos matemáticos. Outra importante área de estudo em processamento digital de imagens é o uso de filtros, para que a qualidade das imagens continue mantida.

Neste capítulo, está descrita uma revisão bibliográfica para o estudo e análise da parte teórica quanto ao funcionamento dos sistemas de processamentos digitais e filtros.

2.1 Introdução a Teoria de Imagens

Na maioria dos casos de processamento de imagens (excetuando-se o reconhecimento de cores), são usados tons de cinza, ou escala de cinza (do inglês *Grayscale*), que nada mais é do que uma função matemática bidimensional de intensidade de luz que funciona em função e proporcionalmente ao brilho da imagem em questão, mostrado matricialmente a seguir (Figura 2):

Figura 2 – Escala de cinza na forma matricial



Fonte: Próprio autor.

Para quadros ou imagens em escala de cinza, a resolução será feita através de números binários. Já em imagens coloridas, vários sistemas foram criados para representar, tais como:

- RGB *Red, Green e Blue*
- XYZ onde o "Y" é dado como o valor de luminância e "X" e "Z", valores cromáticos.
- HSV ou HSB *Hue, Saturation e Brightness*.

Neste trabalho serão analisadas imagens coloridas, usando o método RGB (do inglês RED, Green, Blue), da teoria de cores, sabe-se que qualquer cor pode ser obtida a partir destas 3 cores primárias. Geralmente são usados 8 bits (0 a 255) em cada cor.

2.2 Segmentação

Quando o assunto é processamento digital de imagens, segmentar, significa subdividir a imagem em partes ou objetos constituintes de interesse. (Item 1). As vezes um quadro possui objetos de interesse com a segmentação pode-se isolar os *pixels* que não fazem parte desses objetos que serve para a análise em questão. Os métodos de segmentação são divididos em 3 grandes grupos:

1. Conhecimento global (*Histogram-based*);
2. Baseado em bordas (*Edge-based segmentation*)
3. Baseado em regiões (*Region-based*)

Foco na descontinuidade entre regiões.

Similaridade entre regiões.

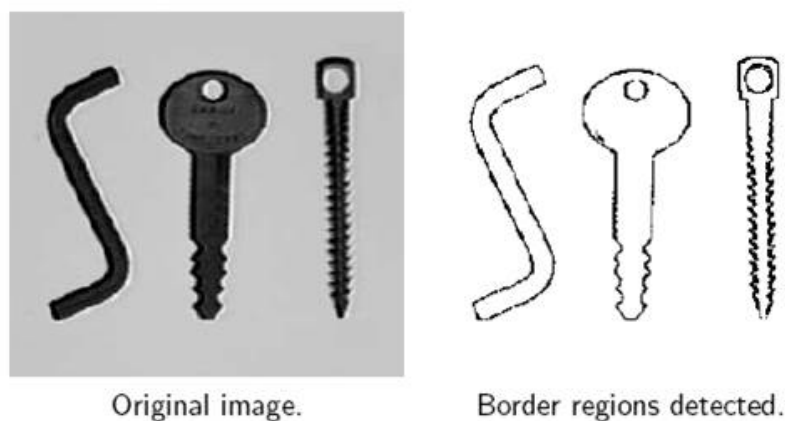
Os métodos mais recentes para segmentação são:

- Algoritmos Genéticos;
- Redes Neurais;
- Métodos estocásticos (MRF);
- Segmentação por textura;
- Morfologia Matemática;
- Segmentação por Textura;
- Segmentação por Cor;

Como vários algoritmos e métodos foram desenvolvidos, cada projeto escolhe e usa um ou vários métodos que mais lhe seja útil, não havendo assim uma solução geral para todos os casos.

Na figura seguinte pode-se observar um exemplo de segmentação já executado.

Figura 3 – Detecção da região de bordas



Fonte: LCAD - USP (2008).

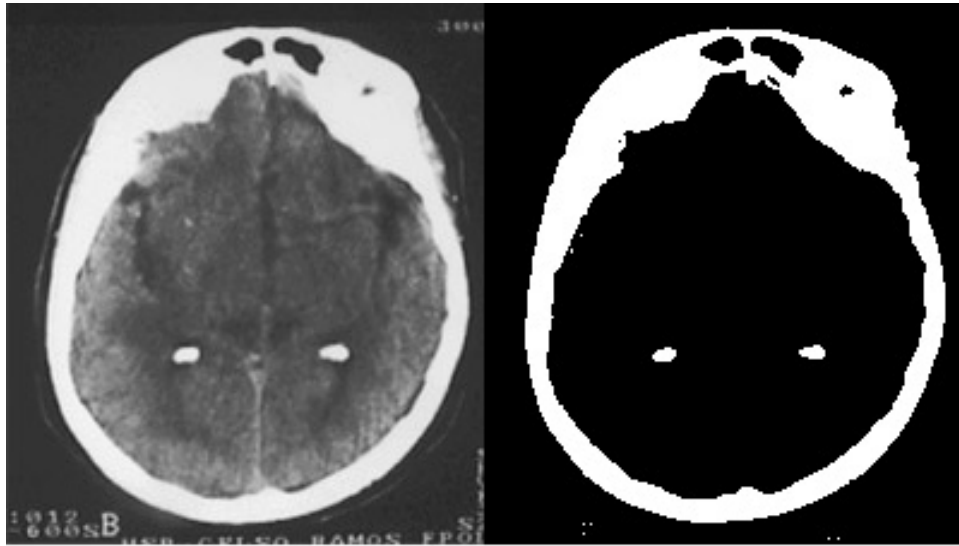
Anteriormente foram explicitados os métodos mais "recentes", porém quando se fala em segmentação é importante falar sobre Segmentação Limiar em tons de cinza (*Grey Scale Thresholding*), que é a mais antiga técnica de segmentação, porém seu uso é limitado a algumas aplicações específicas.

Grey Scale Thresholding é usado quando o objeto de uma determinada cena exibe valores de brilho contra um fundo (que não é o desejado), com outros valores diferentes dos valores do objeto. Esse processo de limiarização pode ser interativo ou heurístico, sendo estes escolhidos após a análise da curva do histograma.

As principais aplicações para este método de limiarização são:

- Área médica;
- Área industrial aplicada a controle de qualidade;
- Radares e satélites.

A figura a seguir ([Figura 4](#)) mostra uma limiarização feita a partir da técnica de *Theshold*.

Figura 4 – Imagem Limiarizada *Threshold 196*

Fonte: Próprio Autor.

2.3 Detecção e Análise de Movimento em Vídeos

Em um vídeo ou uma sequência de imagens, quando se necessita detectar objetos em movimento ou novos objetos que surgem no quadro, é de extrema importância que se conheça bem duas coisas: Subtração de fundo e diferença temporal entre imagens.

A subtração de fundo refere-se a diferença espacial entre o modelo do quadro e a imagem atual captada.

A diferença temporal entre imagens consiste em calcular a diferença entre *frame* corrente a o *n*-ésimo *frame* anterior, bem como fornecer o contorno do objeto em movimento. Geralmente é indicado para ambientes dinâmicos e que apresentam buracos no interior.

As principais técnicas de segmentação de plano de fundo ou *background* são realizadas pelos seguinte métodos:

- Baumberg and Hogg

Modelam o *background* usando um filtro de mediana para cada *pixel*. É usado um limiar feito sob a diferença absoluta entre quadros para identificar o *foreground* (tudo que não é *background*).

- Haritaoglu *et al.*;

Modelam a intensidade dos *pixels* do *background* usando valores de máximos e mínimos entre cada quadro.

- Ridder *et al.*

Usa o método de Filtro de Kalman em cada *pixel* para modelar a intensidade e assim prever o *background*.

- Stauffer and Grimson.

Usa o método de misturas de Gaussianas para modelar a história de cada *pixel*.

Na figura a seguir (Figura 5) fica claro o conceito de *background* e *foreground*.

Figura 5 – (acima à esquerda) Imagem original. (acima à direita) Distância. (abaixo à esquerda) *Background*. (abaixo à direita) *Foreground*.



Fonte: UTFPR (2017).

2.4 Filtragem

Uma definição sucinta e objetiva de filtragem é: Conjunto de técnicas de transformações que visam corrigir, suavizar ou realçar alguma característica de uma imagem ou sequência de imagens em uma aplicação específica.

Quando técnicas de segmentação são usadas em imagens, normalmente se tem como resultado alguns grupos de *pixels* espalhados e isolados, que aqui serão chamados de ruídos. Para que a qualidade do resultado seja aumentada, utiliza-se técnicas de filtragem *pixel a pixel*.

Existem 2 domínios da filtragem de imagens, que são:

- Domínio Espacial;

Estas técnicas são operadas diretamente sobre os *pixels* da imagem, considerando sua forma original.

- Domínio da Frequência

Este domínio opera sobre a Transformada de Fourier da imagem original.

2.4.1 Filtros lineares

Estes tipos de filtros são os que geralmente realçam a imagem ou alvos que constam em imagens digitais e podem aumentar a nitidez das transições entre regiões. Os filtros brevemente citados a seguir são exemplos dos filtros lineares mais comuns, lembrando que estes filtros não alteram a média da imagem.

1. Filtro Passa-Baixa;

Este Filtro possui a característica de manter na imagem apenas as frequências espaciais que se encontram em um intervalo pré determinado. Com um produto da transformada de Fourier de uma imagem, sendo esta de função de transferência $H(u,v)$ é possível obter o realce de uma imagem no domínio da frequência, onde a transformada inversa do resultado obtido será a imagem transformada.

Um filtro Butterworth passa-banda bidimensional, pode ser realizado usando a equação a seguir(Equação 2.1).

$$H(u, v) = 1 - \left(1 / (1 + (DW/D^2 - D_o^2)^{2n}) \right) \quad (2.1)$$

onde:

- D = Distância do centro da imagem (na frequência zero);
- D_o = Frequência de Corte ;
- n = ordem do filtro;
- W = largura da banda.

2. Filtro Passa-Alta;

Este tipo de filtro permite a passagem apenas de frequências acima da sua frequência de corte. Geralmente é usado quando se procura tratar problemas de descontinuidade entre regiões, como este atenua baixas frequências permite detectar contornos, bem como mudanças lentas de cores.

3. Filtro Passa-banda;

Este filtro permite apenas a passagem de uma banda de frequências. Geralmente é usado quando se procura tratar similaridade entre regiões.

2.4.2 Filtros não-lineares

Estes tipos de filtros geralmente são usados para minimizar ou realçar ruídos e suavizar ou realçar bordas, alterando geralmente a média da imagem. Os principais e mais comuns serão brevemente explicitados a seguir:

1. Filtro da Mediana

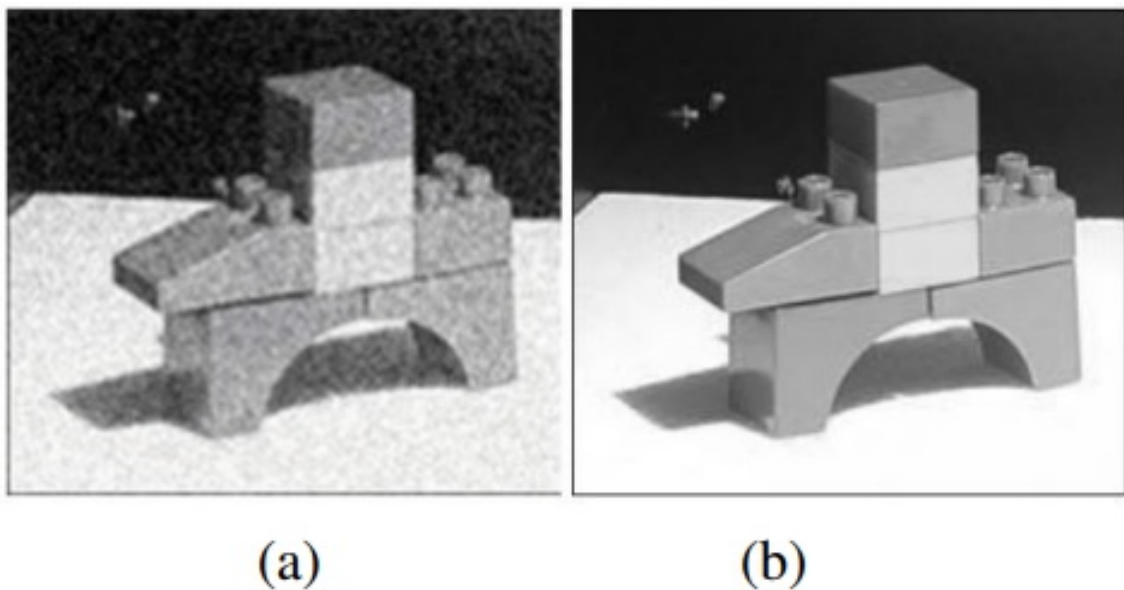
É um filtro muito efetivo na remoção do ruído com os componentes tipo espigada (*Spike*, que é quando ocorre aleatoriedade de valores brancos e pretos). Este tipo de operador preserva bem as bordas.

Tabela 1 – Exemplo de Filtro da Mediana

10	20	20
20	15	20
25	20	100

Na tabela ([Tabela 1](#)), como Mediana = 20 , então substitui-se "15"por "20". A figura a seguir mostra visualmente o resultado da aplicação de um filtro deste tipo.

Figura 6 – Resultado da aplicação do Filtro da Mediana



Fonte: UFF - Aura Conci - 2015.

2. Operador de Sobel

Este tipo de filtro é utilizado geralmente em aplicações onde busca-se detecção de contornos, usando operador de diferença finitas, resultando assim numa aproximação do gradiente da intensidade dos *pixels* do quadro em questão.

É considerado mais sofisticado que o Operador Roberts, quando usado para detectar bordas, baseia-se no método de aplicação de máscaras na circunvizinhança do pixel em questão.

É do tipo Passa-alta e como colocado anteriormente, não linear no domínio espacial. Este operador é dado pela seguinte expressão:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.2)$$

Onde G é a magnitude do gradiente e G_x e G_y são as saídas dos filtros.

Figura 7 – Resultado da aplicação do Filtro de Sobel



Fonte: PUC-Rio - Raquel Guilhaon - 2013.

3. Filtro de Ordem

Neste tipo de filtro são analisadas as intensidades dos pontos da vizinhança de um dado *pixel* (x, y) , dentro da janela de imagem, estes são em consequente ordenados com novo valor $g(x, y)$ mais alto da vizinhança. Vários filtros de ordem são bem famosos e amplamente usados.

4. Filtro de Moda

Neste tipo de filtro são analisadas as intensidades dos pontos da vizinhança de um dado *pixel* (x, y) , dentro da janela de imagem, estes são em consequente ordenados com novo valor $g(x, y)$, diferente do filtro anterior, este usa o valor mais frequente

da vizinhança, ao invés do valor máximo. Geralmente são usados para homogeneizar imagens temáticas, bem como reduzir ruídos mantendo assim o máximo possível de informação no quadro.

5. Filtro detectores de Borda

São usados para identificar transições ou mudanças bruscas na função $f(x, y)$.

O método mais utilizado para obter o gradiente é a derivada do *pixel* tomando como base seus vizinhos.

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.3)$$

Sendo que para calcular o tamanho deste vetor é usada a seguinte equação (Equação 2.4):

$$|\nabla f| = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{\frac{1}{2}} \quad (2.4)$$

2.4.3 Filtragem Homomórfica

O filtro Homomórfico foca na luminância e reflectância de imagens, assim dividindo cada *pixel* em componentes distintas de luminância e reflectância.

A reflectância chamada de $i(x, y)$ é a medida que mostra o quanto de luz incide sobre cada *pixel*, esta medida apresenta uma variação de baixa frequência (lenta). Quando se trata da reflectância, aqui chamada de $r(x, y)$ tem-se a medida de quanto a luz incidente em cada *pixel* é refletida, esta medida, ao contrário da anteriormente citada, apresenta variações de altas frequências. Sabendo isto uma boa representação de um *pixel* é (Equação 2.5) :

$$f(x, y) = i(x, y)r(x, y) \quad (2.5)$$

Com essas definições bem estabelecidas pode-se definir como objetivo principal de um filtro homomórfico, a correção de uma possível má iluminação de uma cena. Por isso como visto na equação anterior (Equação 2.5), atua-se separadamente nas componentes de iluminância e reflectância, como :

$$\mathfrak{F}(i(x, y)r(x, y)) \neq \mathfrak{F}(i(x, y))\mathfrak{F}r(x, y) \quad (2.6)$$

deve-se aplicar algumas operações para ser possível a execução do filtro.

$$z(x, y) = \ln(i(x, y)r(x, y)) = \ln(i(x, y)) + \ln(r(x, y)) \quad (2.7)$$

$$S(u, v) = \mathfrak{F}(z(x, y)).H(u, v) \quad (2.8)$$

$$s(x, y) = \mathfrak{F}^{-1}(S(u, v)) \quad (2.9)$$

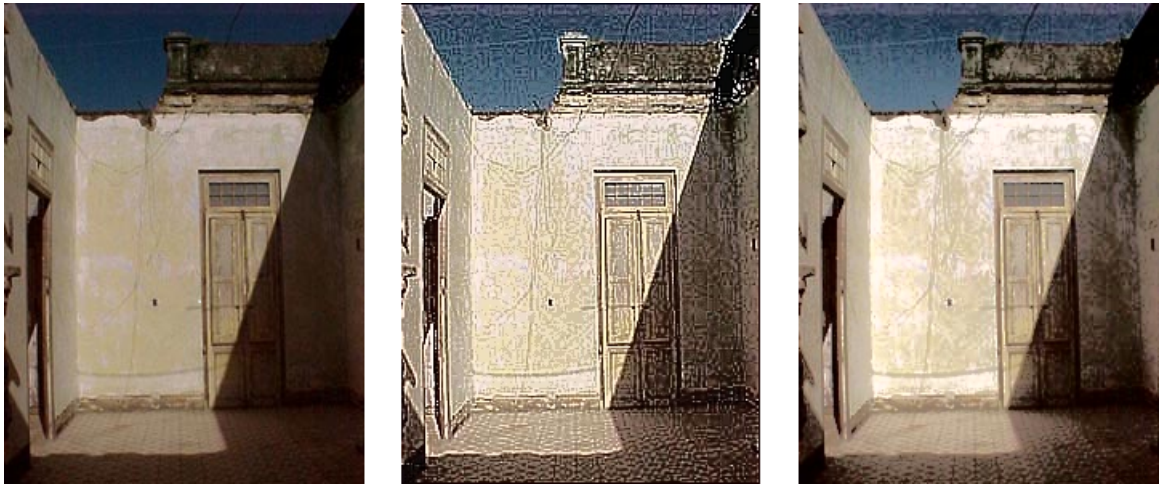
Sendo assim a imagem filtrada fica da seguinte forma:

$$g(x, y) = e^{s(x, y)} \quad (2.10)$$

e a componente $H(u, v)$ é o filtro em questão (Homomórfico) na frequência. Esta componente pode ser descrita da seguinte forma:

$$H(u, v) = (\gamma H - \gamma L) \left(1 - e^{-c \left(\frac{D^2(u, v)}{D_0^2} \right)} \right) + \gamma L \quad (2.11)$$

Figura 8 – Exemplo de aplicação de um Filtro Homomórfico



Fonte: Rafael Heitor (2005).

2.5 Atributos humanos

Com base na bibliografia e outros trabalhos anteriores, serão definidos a seguir algumas medidas e valores característicos mais comuns em seres humanos, para que, com auxílio desses parâmetros, seja possível.

- Largura média entre os ombros.
- Altura média.
- Velocidade média.

- Profundidade média do tórax.

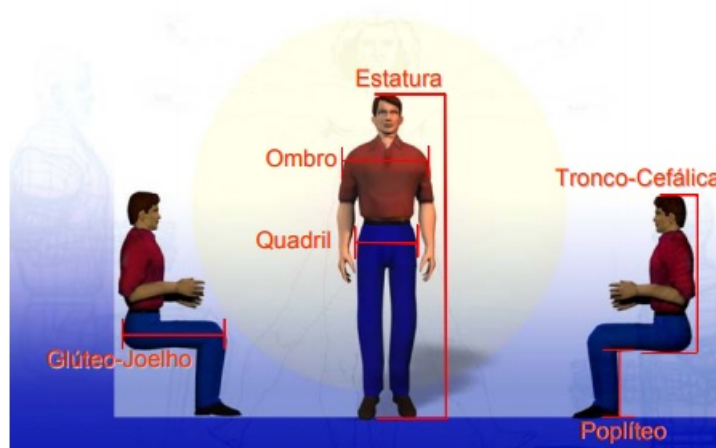
Essas informações específicas serão usadas para gerar um modelo de dimensões corporais humanas.

2.5.1 Modelo de Dimensões

Segundo o "Levantamento do perfil Antropométrico da População Brasileira" (MONTEIRO, Wallace - 2009) as medidas mais usadas para estudos são:

Geralmente quando se discute sobre engenharia, arquitetura e desenho industrial, é necessário obter um modelo simplificado, este modelo será descrito e discutido a seguir.

Figura 9 – Medidas Antropométricas



Fonte: MONTEIRO, Wallace - 2009.

Os dados encontrados no *paper* anteriormente descrito foram os seguintes:

- Ombros 45 a 50 cm.
- Quadril 38,7 cm de média.
- Estatura média 173,1 cm

2.5.2 Média de Velocidade

A velocidade média de um ser humano em vias urbanas sofre influências de diversos fatores, tais como:

- Idade.
- Massa corpórea.

- Qualidade do ambiente de caminhada.
- Hora do dia.
- Possível carga.
- *Crowd* - Quantidade de pessoas no local de trânsito.
- Calçado utilizado.
- Gênero.

Assim um número preciso desse valor torna-se uma tarefa bem complicada, porém estudos sugerem alguns valores médios que serão usados como base para análises e modelagens posteriores.

- Pessoas normais

1,22 m/s

- Idosos do sexo masculino

0,99 m/s

- Idosos do sexo feminino

0,76 m/s

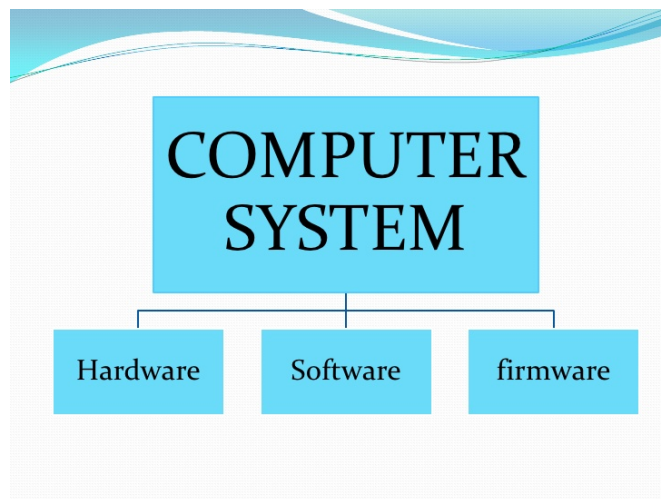
Porém esses valores podem variar de acordo com os itens citados anteriormente, assim estes serão apenas usados como base.

3 Implementação

A implementação deste projeto se divide de acordo com as etapas seguintes:

1. *hardware*
2. *firmware*
3. *software*

Figura 10 – Base de sistemas de computação



Fonte: Próprio Autor.

3.1 *Hardware*

O *Hardware* é a parte física do sistema, sendo assim um conjunto de aparelhos eletrônicos e equipamentos usados no projeto. O *hardware* do projeto consiste nos componentes eletrônicos listados:

1. *Raspberry Pi Model B*;
2. *Raspberry Pi Camera Rev 1,3*;
3. Adaptador Hdmi para VGA
4. Teclado;
5. *Mouse*;

6. Circuito de Acionamento;
7. Cartão *microSD*;
8. Fonte de Alimentação.

3.1.1 *Raspberry Pi Model B*

O *Raspberry Pi* é uma fundação inglesa que desenvolveu um "computador" pequeno e portátil chamado pelo mesmo nome da fundação. O objetivo inicial dessa placa era o promover o ensino de computação e engenharia em escola básicas, porém visto o grande leque de possibilidades de uso, bem como sua facilidade com o sistema *raspian* o *Raspberry pi*, passou a ser mundialmente conhecido e vastamente usado em diversos tipos de projetos, tanto industriais como *DIY*(do inglês *Do it Yourself*, projetos caseiro "faça você mesmo).

Figura 11 – *Raspberry Pi Model B*



Fonte: *Site Raspberry Pi Foundation*.

Projetos desenvolvidos utilizando o *Raspberry pi* são incríveis e alguns valem a pena ser citados neste relatório.

- *Mini Web Browser*

Um método que permite usar uma televisão grande como computador comum e "surfear" na *internet*.

- *Arcade Pi*

Para entusiastas de *games* é possível criar um ambiente *arcade* para emulação de diversos jogos.

- *Mini Câmera instantânea*

É um projeto que já está sendo vendido em alguns sites, de uma mini câmera que fotografa e logo em seguida imprime sua miniatura de foto.

- Toca fitas com *Spotify*

Utilizando um *case* de antigos tocas fitas onde as fitas possuem um *TAG* embutido que permite uma busca no aplicativo *Spotify*, esse aparelho faz grande sucesso com o publico jovem entusiasta da moda *Vintage*.

Neste projeto de pesquisa a placa microcontroladora é responsável pelos itens numerados a seguir. Seu modelo é *Raspberry Pi Model B 2011.2012*.

1. Ler o botão de liga/desliga.
2. Capturar e processar os sinais oriundos da câmera.
3. Gravar os dados oriundos da câmera no cartão de memória micro que será conectado na entrada adequada já embutida na placa microcontroladora utilizada.
4. Guardar o quantitativo de pedestres nas duas direções.

A tabela a seguir mostra uma comparação entre as principais plataformas de prototipagens:

Através da análise da tabela anterior ([Tabela 2](#)), bem como a disponibilidade de bibliografia, simplicidade de programação e uma *interface* agradável de programação, a plataforma *Raspberry pi* foi escolhida.

3.1.2 *Raspberry Pi Camera*

A Câmera usada neste relatório técnico será um módulo próprio para uso em *Raspberry pi, Pi NoIR Rev 1.3*. Como pode-se observar no nome do módulo, este é fabricado pela mesma empresa/fundação onde é fabricada a plataforma *Raspberry Pi*.

Tabela 2 – Comparação entre as principais plataformas de desenvolvimento

Plataforma	Arduino Uno	Raspberry Pi	Beagle Bone
Modelo Analisado	R3	Model B	Rev A5
Preço médio	U\$22.00	U\$ 35.00	U\$ 89.00
Tamanho	2,85"x 2,10"	3,37"x 2,125"	3,4"x 2,1"
Processador	ATMega 328	Arm11	ARM Cortex - A8
Clock	16 Mhz	700Mhz	700 Mhz
RAM	2KB	256MB	256MB
Flash	32KB	(Sd Card)	4GB(Micro Sd)
EEPROM	1KB	-	-
Tensão de entrada	7 - 12 V	5V	5V
Potência mínima	0.3W	3.5W	0.85W
GPIO	14	8	66
Entradas analógicas	6	-	7
PWM	6	-	8
TWI/I2C	2	1	2
SPI	1	1	1
UART	1	1	5
Dev IDE	Arduino Tool	IDLE, Scratch, Squeak, Raspian	Python, Scratch, Debian
Ethernet	-	10/100	10/100
USB master	-	2 - USB 2.0	1 - USB 2.0
Saída de Video	-	HDMI	-
Saída de Audio	-	HMDI/Analógica	Analógica

Produção próprio autor

Características do módulo:

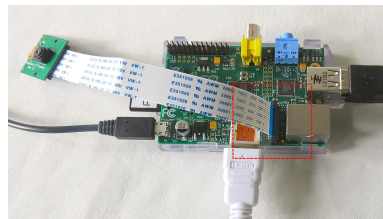
- Lentes de foco fixo *on-board*;
- Sensor de resolução de 8MP, capaz de imagens estáticas de 3280 X 2464;
- Vídeos de 1080p30fps - 720p60fps - 640x480p90fps;
- Conexão direta á placa por cabo *flat*.
- Compatível com várias versões do *Raspian*¹
- Peso 3,8 g.

A figura à seguir ([Figura 13](#)) mostra a conexão câmera-*raspberry* que é feita diretamente via um cabo *flat* o que simplifica a ligação, bem como não ocupa outras saídas da placa, podendo assim ampliar suas funcionalidades.

¹ *Raspbian* é uma variante do *Debian* baseada no *ARM hard-float*, sendo um porte da arquitetura *Wheezy*, otimizada para o conjunto de instruções ARMv6 do hardware do *Raspberry Pi*.

Figura 12 – *Raspberry Pi Camera*

Fonte: Próprio autor.

Figura 13 – Conexão via *flat cable* placa/câmera

Fonte: Próprio autor.

Essa câmera será responsável pela captura dos quadros, sendo assim a peça fundamental do projeto deste protótipo. Várias outras câmeras os módulos podem ser usados, porém este foi o que preencheu melhor os pré-requisitos necessários.

O aumento da qualidade de detecção esbarra na capacidade de processamento e armazenamento da plataforma, desta forma, optou-se por uma câmera de qualidade satisfatória.

3.1.3 Adaptador HDMI para VGA

A placa *Raspberry pi*, tem em sua entrada de vídeo, um conector do tipo HDMI (*High-Definition Multimedia Interface*). Este tipo de conector é o atualmente mais usado nos aparelhos de vídeo, porém os monitores que usam este tipo de conector, são usualmente mais caros e grande parte frágeis.

Visto que o protótipo visa num futuro próximo ser comercializado como uma solução *Lowcost*, optou-se por usar um monitor mais comum e robusto para programação e apresentação da interface, visto que estes monitores utilizam conectores do modelo VGA

(*Video Graphics Array*), um adaptador HDMI-VGA (Figura 14) é necessário para fazer esta conexão.

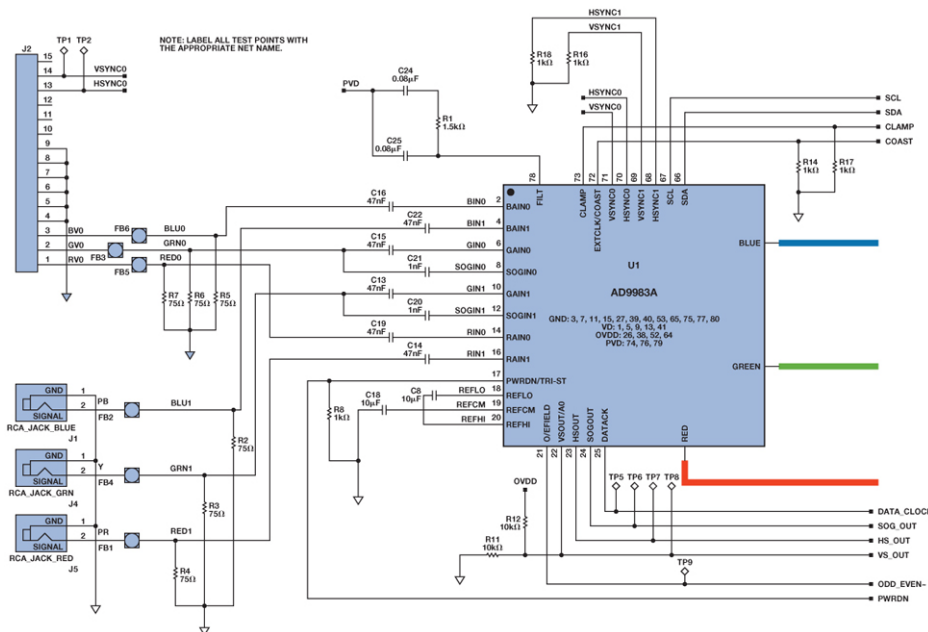
Figura 14 – Adaptador HDMI - VGA



Fonte : Próprio autor.

Inicialmente optou-se pelo desenvolvimento de um adaptador pelo próprio autor, o circuito idealizado pode ser observado a seguir (Figura 15), porém o a partir da análise de benefício/custo, levanto em conta, tamanho, robustez, dificuldade na compra de alguns componentes, simplicidade, preço e confiabilidade optou-se pela compra de um adaptador pronto (Figura 14).

Figura 15 – Circuito do Adaptador HDMI - VGA



Fonte: *Site AnalogDialogue.com*

A tela, bem como o adaptador (Figura 14), não fazem parte do protótipo, apenas serão usados no desenvolvimento, não sendo estes instalados em futuros testes ou vendas.

3.1.4 Teclado

A plataforma *Raspberry pi* possui em seu *hardware* duas entradas USB, onde serão conectados um teclado e um *mouse* durante o desenvolvimento.

O teclado é de modelo simples, padrão ABNT 2; conector tipo USB, na cor preta (este assim que plugado, não necessita de instalação, pois o sistema operacional embarcado reconhece e instala automaticamente o *drive* respectivo).

Figura 16 – Teclado



Fonte : Próprio autor.

O teclado ([Figura 16](#)) não faz parte do protótipo, apenas será usado no desenvolvimento, não sendo este instalado em futuros testes ou vendas.

3.1.5 Mouse

O *mouse* é de modelo simples, tipo óptico; Conector tipo USB; na cor preta; possui 7 botões para navegação, sendo um destes o *scroll* (este assim que plugado, não necessita de instalação, pois o sistema operacional embarcado reconhece e instala automaticamente o *drive* respectivo).

Figura 17 – Mouse



Fonte : Maplin - 2015.

O *mouse* (Figura 17) não faz parte do protótipo, apenas será usado no desenvolvimento, não sendo este instalado em futuros testes ou vendas.

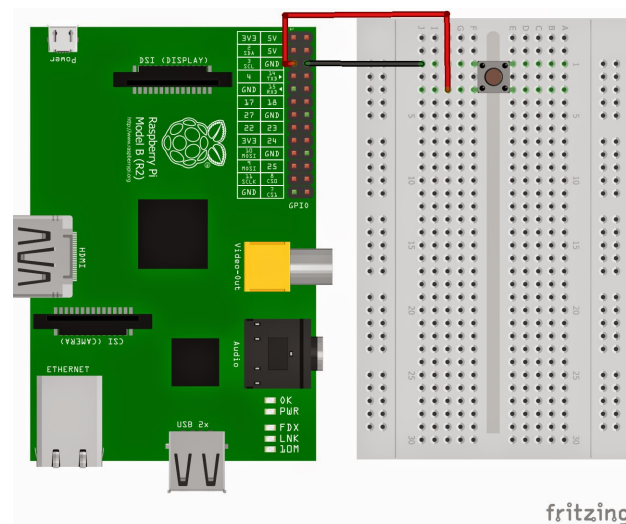
3.1.6 Circuito de Acionamento

O circuito de acionamento externo visa a facilidade de operação do protótipo. Quando ainda em fase de testes esse recurso não é tão válido, visto a experiência do autor em eletrônica e programação, entretanto como o foco do sistema é ser concebido como um produto é imprescindível pensar em simplicidade de execução.

Como o protótipo visa ser num futuro próximo um produto, este deve ser feito pensando em um consumidor final desprovido de conhecimentos avançados de programação, pois com é sabido profissionais que podem subir em alturas elevadas necessitam ter certificação NR35 e para atuação em equipamentos elétricos, certificação NR10, porém quando esses profissionais são dotados dessa certificação seu salário é incrementado geralmente de 30 50%, ou seja, poucas são as empresas que utilizam profissionais de engenharia com essas certificações.

O circuito de acionamento é feito com um botão simples e uma técnica com resistor de *pull-up* interno da *Raspberry Pi*. Assim que conectado com fonte a placa irá aguardar o comando do botão para iniciar a captura e o processamento de imagens, bem como armazenamento dessas informações, como visto na [Figura 18](#).

Figura 18 – *Circuito externo de acionamento*



Fonte : Acionamento-fritzing - 2017.

Foi analisado a possibilidade de implementação de um sistema *wireless*, porém este foi descartado a partir dos motivos listados a seguir:

- Aumento do custo do projeto.
- Inserção de possíveis erros de transmissão.
- Aumento da complexabilidade de desenvolvimento, bem como instalação.

- Necessidade de dispositivos para transmissão de dados (que elevaria o custo) como por exemplo:
 - rede *Wifi* com antenas de longo alcance.
 - rede GPRS com uso de cartão SIM.
- Velocidade de transmissão.
- Aumento da solicitação de processamento do *hardware*.

3.1.7 Cartão *microSD*

Este cartão usa uma tecnologia (UHS-I) que foi projetada para aumentar o desempenho dos modelos SDHC I, oferecendo taxas de transferência de até 50MB/s, lembrando que estes são limites teóricos, podendo assim variar com limitações até mesmo da própria plataforma.

Figura 19 – Cartão de Memória



Fonte : Próprio autor.

Pensando em gravações mais prolongadas, foi realizado uma tentativa de "burlar" o sistema de leitura do *Raspberry pi* usando um *software* chamado *Sd Formatter*, com o intuito de mudar o tipo de tabela de alocação de arquivos bem como o tamanho da *storage* do cartão de 64Gb para 32GB (pois a *Raspberry pi model B* somente aceita cartões até 32Gb) e assim conseguir usar um cartão de 64GB. O processo funcionou, porém com a qualidade da câmera, não foi necessário o uso deste cartão.

Após essa breve introdução, o cartão será formatado com o sistema FAT ou FAT32. Após a formatação foi instalado o sistema operacional *Raspian* com o auxílio da ferramenta NOOBS.

3.1.8 Fonte de alimentação

Como pode-se observar na [Tabela 2](#) a placa que mais drena potência da fonte (3.5W), na sua entrada é a que está sendo usada nesta monografia, sendo este valor de potência um pouco elevado, quando se trata de fontes simples chaveadas, dessa forma foi escolhida essa fonte que será brevemente explicitada a seguir.

Fonte modelo SYS1089-1305-W2 da empresa TML *Technology Co LTDA*.

Figura 20 – Fonte chaveada



Fonte : Próprio autor.

- Tensão de entrada 100-240V;
- Corrente de entrada máxima 1,0A (aproximadamente);
- Tensão de saída 5V;
- Corrente de saída 2,6A;
- Faixa de frequência de entrada 50 a 60 Hz;
- Faixa de potência aparente 28-40VA;

Foi necessário uma pequena adaptação no conector (substituição do *plug*) de ligação da fonte chaveada, para que está pudesse ser conectada ao *Raspberry pi*.

3.1.9 Bateria

Geralmente o ponto onde será analisado o tráfego de pessoas possuem fios de alimentação em tensão alternada comuns (127V ou 220V), sendo assim não é necessário o uso de baterias para uso do protótipo, todavia em um caso de necessidade, foi realizado um estudo sobre diversas tipos de baterias caso houvesse uma necessidade urgente.

A bateria que melhor atendeu as especificações foi a bateria de tipo LIPO as quais possuem alguns atributos que levam vantagens quando comparadas com outros modelos, tais como: Leveza, capacidade de descarga, durabilidade e velocidade de recarga.

Algumas características da bateria escolhida como caso urgente de necessidade:

- Mais leve que as convencionais Ni-MH (peso igual a 337g);
- Elevada capacidade de descarga mesmo após de 200 ciclos;
- Não sofrem efeito memória²;
- Pode ser carregada sem precisar estar descarregada;
- É necessário carregador específico para Lipo;
- Tensão: 5.0 V ;
- Capacidade de carga: 700 mAh;
- Capacidade de descarga: 25 C (Coulombs) contínuos.

3.2 Firmware

Firmware pode ser definido como um conjunto de instruções operacionais que são programadas diretamente no *hardware* de equipamentos eletrônicos.

Geralmente são escritos nas memórias PROM ou EPROM, porém na plataforma adotada, o código fica escrito no próprio cartão de memória (caso esse seja removido) o *firmware* não permanece escrito na *raspberrypi*.

Sabendo isso será apresentado a seguir o *firmware* que foi construído neste protótipo.

² As baterias que são propensas ao efeito, tendem a adquirir uma capacidade de carga cada vez menor. A bateria deixa de ser carregada por completo, mesmo o seu marcador indicando o contrário. Isso se deve principalmente a modificações químicas sofridas pelos materiais utilizados na confecção das células, e também ao ato de recarregar a bateria com ela não totalmente descarregada. Também conhecido como vício nas baterias

3.2.1 Captura do Sinal de Acionamento

Como dito na [subseção 3.1.6](#), através de uma chave de acionamento momentâneo o *firmware* começa a ser executado.

A lógica de programação segue a seguinte ordem:

- Primeiramente insere-se um código para rodar automaticamente após a energização da *raspberrypi*;
- Esse código lê e aguarda o acionamento de um botão momentâneo;
- Assim que pressionado o botão inicia um arquivo em *Python*;
- Este arquivo iniciado é o responsável pela captura, processamento e armazenamento dos dados gerados;

Para que um código inicie junto com a energização da *raspberrypi* é necessário seguir os seguintes passos no terminal do sistema *raspian*.

É necessária a edição do arquivo `rc.local` (para execução desta tarefa usa-se):

```
1 sudo nano /etc/rc.local
```

No arquivo será inserido em uma de suas linhas um comando que indica qual o arquivo deve ser executado, bem como sua localização no diretório.

```
1 sudo /home/pi/pg2/inicializacao &
```

O uso do caractere `&` tem como finalidade que o programa seja executado em *background*, também protege a placa de um *loop* de inicialização, pois libera a finalização deste processo.

O código da leitura de botão é bem simples sendo desta forma possível explicitá-lo aqui.

```
1
2 #Programa : inicializacao
3 #Autor : JACS
4
5 contador = 0
6
7 import RPi.GPIO as GPIO
8 import time
9
10
11 GPIO.setmode(GPIO.BOARD)
12
```

```

13
14 #Define o pino do botao como entrada
15 GPIO.setup(18, GPIO.IN)
16
17 while(1):
18 #Verifica se o botao foi pressionado
19     if GPIO.input(18) == True:
20         execfile('pg2.py')

```

Deste modo, logo após o acionamento do botão, o código desejado é executado, dando inicio a análise. É importante salientar que todo este processo vem após uma calibração do instrumento, bem como seu posicionamento correto.

3.2.2 Captura dos *frames* via *Raspberry Pi*

Uma vez que o sistema já foi configurado e os periféricos habilitados e configurados, bem como o botão de acionamento acionado a próxima etapa é a leitura dos dados oriundos do módulo câmera, bem como sua captura e processamento.

O processamento digital de imagens é uma área altamente complexa da engenharia e sua implementação sem auxílio de bibliotecas prontas a torna inviável para alunos de graduação, pois contam com processamentos complexos e utilização de filtros bem específicos. E citando o ex-Diretor de cultura da empresa Ivoti, agora diretor da Casa de Cultura Mario Quintana: "Nós não queremos reinventar a roda". Assim como uso de uma biblioteca específica mostrou-se promissor no início e satisfatório ao final, esta foi usada.

Todo o processamento desse plataforma foi realizado com o auxílio da biblioteca *OpenCv* que é um código aberto focado em visão computacional. Esta biblioteca possui uma estrutura modular, desta forma o pacote é disposto de várias bibliotecas.

3.2.2.1 Captura

Como descrito na [subseção 3.1.2](#), o vídeo será feito com uma câmera do mesmo fabricante da plataforma, conectada através de um *flat cable*.

Uma classe para captura de vídeo comumente usada em processamento de quadros, ou sequência de quadros, é a seguinte:

```

1 camera = cv2.VideoCapture("test2.mp4")

```

Como pode ser visto o argumento "test2.mp4" trata-se de um vídeo no formato "mp4", armazenado no mesmo diretório que o arquivo "pg2.py". No caso mostrado nesse relatório, como se trata de um sistema que visa processar imagens em tempo real, o argumento será mudado para "0" que é o valor *default* (assim que instalado um pacote *imutils*) do barramento de pinos do conector do *flat cable*.

Comando de instalação do *imutils*, no terminal do *raspian*.

```
1 $ pip install imutils
```

Após a instalação do pacote anteriormente citado, a comando de captura fica da seguinte forma:

```
1 camera = cv2.VideoCapture(0)
```

3.2.3 Processamento

Para o processamento de imagens, as vezes é necessário o redimensionamento dos quadros, para que a imagem possa ter largura (*width*) e altura (*height*), sendo assim proporcional. Para alcançar este formato é usado o comando descrito abaixo. Com este comando utilizado é possível observar que o quadro ficará com a sua altura igual a sua largura.

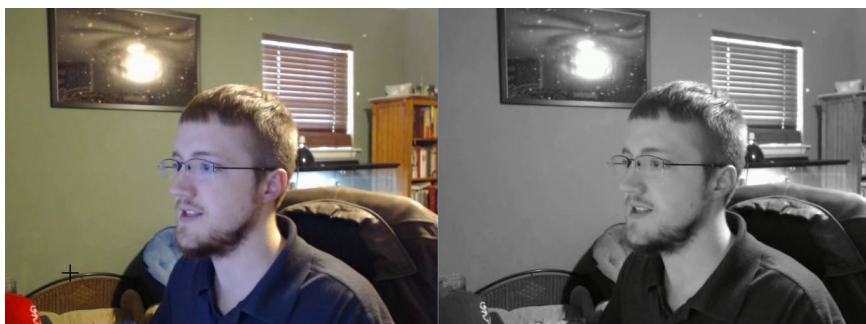
```
1 frame = imutils.resize(frame, width=width)
```

Neste protótipo o valor usado para *width* é 800 *pixels*. Que é um valor que proporcionou uma boa velocidade de processamento, e não excluiu nenhum objeto importante da cena.

Como o protótipo não foca, muito menos pretende identificar cores, a imagem é mudada de escala comum de cores para uma escala de cinza (*GrayScale*). A ideia de mudar a escala de cores se dá para que a complexidade do processamento diminua significativamente. O seguinte comando foi usado e um exemplo de resultado pode ser visto na [Figura 21](#).

```
1 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

Figura 21 – Exemplo de aplicação da escala de cinza



Fonte Youtube.com - 2016

Um filtro gaussiano é um filtro linear é geralmente usado para desfocar a imagem ou para reduzir o ruído. Se você usa dois deles e subtraindo, você pode usá-los para

unsharp masking (detecção de borda). O filtro gaussiano sozinho apagará bordas e reduzirá o contraste.

Uma vantagem dos filtros gaussiano sobre um filtro mediano é que este é mais rápido (poupando assim esforço computacional) porque usa apenas multiplicação e adição, sendo por esses motivos o escolhido para uso neste projeto.

O pacote *OpenCv* já conta com este recurso que pode ser usado com o comando a seguir:

```
1 gray = cv2.GaussianBlur(gray, (21, 21), 0)
```

O seguinte comando foi usado e um exemplo de resultado pode ser visto na [Figura 22](#), para melhor visualização e entendimento, foi inserido uma comparação do tratamento de ruído do filtro Gaussiano ao lado de uma aplicação de Filtro Mediano.

Figura 22 – Exemplo da aplicação do Filtro Gaussiano



Fonte MINES ParisTech - 2002 .

3.2.3.1 Lógica

A parte de lógica deste relatório trata como é feita a detecção e cálculos das mudanças na sequência de quadros, tendo como resultado que um objeto está em movimento e assim dependendo do seu tamanho, classifica-lo como um humano e deste modo incrementar uma variável.

Como visto em [seção 2.5](#) foi mostrado alguns dados de velocidade média e dimensões de humanos.

Analisando *frame* a *frame* é possível, calculando a diferença absoluta, detectar diferença de valores em quadros, assim consequentemente decretar que o dado objeto está se movendo.

```
1 frameDelta = cv2.absdiff(firstFrame, gray)
```

A linhas bases podem ser vários locais, que no caso é uma das partes mais importantes deste relatório, pois estas linhas definem o resultados das análises.

- Entrada de um *Shopping*;

Para saber qual das várias entradas do *shopping* é mais utilizada.

- Entrada de uma casa de *Shows*;

Pode ser usado para saber se a quantidade de pessoas que entrou pela entrada principal foi a mesma quantidade que saiu (detecção de penetras) .

- Entrada de uma escada rolante;

Para comparar qual escada é mais utilizada, podendo ser analisado para diferente horários do dia.

Figura 23 – Linhas bases de referência usadas na detecção da direção do movimento.



Site embarcados.com - 2016

Após as linhas bases definidas, o próximo passo é fazer um teste de interseção, que dirá se o movimento foi na direção de "entrada" ou "saída". Assim que um dado objeto cruzar as duas linhas, neste instante, detecta-se a sua direção de movimento, visto que é ficou gravado em qual linha este passou primeiro e em qual cruzou por ultimo.

Para realização deste teste foram utilizados os seguintes comandos:

```

1  if(testeIntersecaoEntrada((x + x + w) / 2, (y + y + h) / 2)):
2      textEntrada += 1
3
4  if(testeIntersecaoSaida((x + x + w) / 2, (y + y + h) / 2)):
5      textSaida += 1

```

3.3 Software

Foi analisado a possibilidade da criação de um *software* na linguagem C ou C++, para que a plataforma apenas gravasse os dados e após a gravação os dados fossem analisados em um computador de melhor poder de processamento.

A partir da análise dos primeiros testes, foi visto que realmente o processamento estava consumindo praticamente 100% do processamento disponível na *raspberrypi*, porém

com a análise dos dados, verificou-se que acurácia estava em um patamar satisfatório, visto isso optou-se por embarcar o processamento diretamente na placa, sem a necessidade de processamento posteriores.

Alguns dos motivos de não usar um método de pós-processamento foram:

- Necessidade de um computador para pós processamento.
- Como foram definidos alguns critérios anteriormente, como tamanho do quadro (após o redimensionamento), bem como qualidade da imagem capturada, foi observado que seria preciosismo não fazer o processamento diretamente embarcado.
- Como a análise é feita em ambiente ventilado e quando o *case* do produto for desenhado haverá a presença de um *cooler*, o problema de aquecimento devido ao esforço de processamento é mitigado.
- O não uso de pós-processamento dá mais autonomia ao sistema e diminui significativamente o custo que um dos principais objetivos deste projeto.

4 Testes e Resultados

Os testes em terra foram realizados na SINALES - Sinalização do Espírito Santo às 07 horas do dia 18, 19 e 20 de Dezembro de 2017. Foi usado um modelo de faixa emborrachado que se encontra na entrada do principal galpão da empresa.

É importante salientar que os itens restritores, descritos na [seção 1.3](#) foram observados. Caso estes itens não sejam observados a eficácia do projeto pode diminuir.

Antes da apresentação dos resultados vale ressaltar que todos os testes foram realizados em ambientes abertos com vento praticamente constante o que mitiga todo tipo de problema se sobreaquecimento.

4.1 Primeiro Teste

Este teste foi realizado dia 18 de Dezembro de 2017, pela manhã que é quando ocorre o maior tráfego de pessoas, visto que esta é a hora de liberação das equipes que fazem ronda diária

O primeiro teste foi o com melhor resultado. Analisando o motivo dessa melhora pode-se observar que o principal motivo foi a regularidade do tráfego de pessoas

Figura 24 – Ambiente - Teste 1



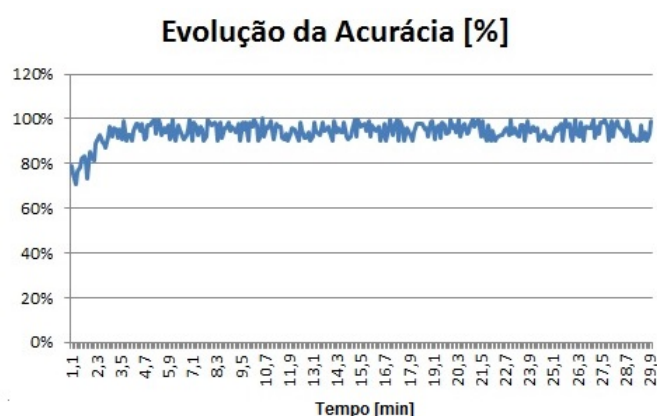
Fonte : Próprio autor.

Tabela 3 – Resultados do primeiro teste realizado

TESTE 1	PESSOAS ENTRANDO	PESSOAS SAINDO
Sistema Desenvolvido	121	13
Contagem visual	132	15
ACURÁCIA	91,67%	86,67%

Fonte : Próprio autor.

Figura 25 – Acurácia - Teste 1



Fonte : Próprio autor.

Pode-se observar na [Figura 25](#) que com o passar do tempo a curva tende a aumentar a taxa de acerto e se tornar mais estável.

4.2 Segundo Teste

Este teste foi realizado dia 19 de Dezembro de 2017. Realizado pela manhã que é quando ocorre o maior tráfego de pessoas, visto que esta é a hora de liberação das equipes que fazem ronda diária

Tabela 4 – Resultados do segundo teste

TESTE 2	PESSOAS ENTRANDO	PESSOAS SAINDO
Sistema Desenvolvido	155	35
Contagem visual	167	42
ACURÁCIA	92,81%	83,34%

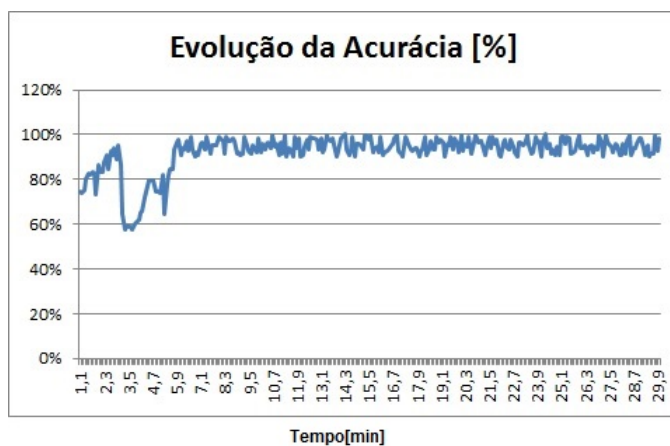
Fonte : Próprio autor.

Figura 26 – Ambiente - Teste 2



Fonte : Próprio autor.

Figura 27 – Acurácia - Teste 2



Fonte : Próprio autor.

Os erros iniciais de alto valor observados na [Figura 27](#) foram causados por algumas pessoas que tiveram problemas como a liberação de notas fiscais e saíram com caminhões atrasados, atrapalhando um pouco a parte inicial da detecção.

Este foi o dia com maior fluxo de pessoas trafegando pela área mesmo assim não houve influência negativa no resultado final.

4.3 Terceiro Teste

Este teste foi realizado dia 20 de Dezembro de 2017. Realizado pela manhã que é quando ocorre o maior tráfego de pessoas, visto que esta é a hora de liberação das equipes

Tabela 5 – Resultados do terceiro teste realizado

TESTE 3	PESSOAS ENTRANDO	PESSOAS SAINDO
Sistema Desenvolvido	79	46
Contagem visual	93	55
ACURÁCIA	84,95%	83,63%

Fonte : Próprio autor.

que fazem ronda diária.

Figura 28 – Ambiente - Teste 3

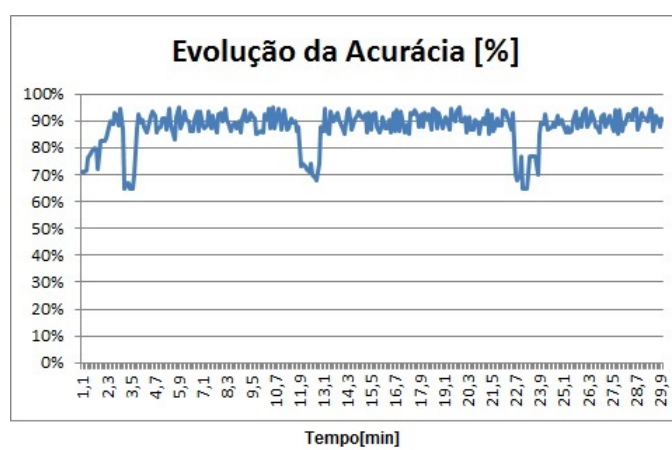


Fonte : Próprio autor.

Neste dia o teste apresentou alguns valores ainda dentro da faixa de satisfação (80% de acurácia), porém destoou dos testes anteriores. De acordo com a experiência adquirida na confecção deste projeto pode-se relatar a principal possível causa dessas imprecisões.

- Grande quantidade de pessoas passando com alguns materiais não comuns (semáforos de grande volume).

Figura 29 – Acurácia - Teste 3



Fonte : Próprio autor.

5 Conclusão

Este trabalho de monografia teve como objetivo desenvolver um sistema embarcado de aquisição e processamento de dados para capturar dados úteis de um módulo câmera, visando o desenvolvimento de um produto final, com objetivo comercial de contagem do fluxo de pessoas em um determinado local.

Seguindo este objetivo, foi desenvolvido o *firmware* necessário para captura do vídeo provenientes da câmera escolhida para este projeto de acordo com valor e disponibilidade de peças e também levando em consideração a base de conhecimento do autor.

Tabela 6 – Custo do projeto (excetuando a mão de obra e pesquisa)

ITEM	Valor
Raspberry Pi Model B	R\$125,69
Raspberry Pi Câmera PI NOIR Rev 1.3	R\$75,23
Adaptador HDMI - VGA	R\$8,34
Teclado	R\$17,90
Mouse	R\$19,90
Circuito de acionamento	R\$2,50
Micro Sd	R\$37,12
Fonte de alimentação	R\$25,28
Monitor - VGA	R\$200,00
Total	R\$511,92

Fonte : Próprio autor.

Concluiu-se que com o valor usado no projeto, comparado aos produtos de mesma finalidade disponíveis no mercado, o preço de venda pode chegar a vinte vezes menor que os produtos de marcas mais conhecidas.

É obvio que a câmera da Empresa AXIS dispõe de mais funcionalidades e confiabilidade, porém a comparação serve para mostrar que é possível a produção de um sistema *lowcost* de processamento digital de imagens com foco em contagem de pessoas.

Tabela 7 – Comparação entre produtos similares disponíveis no mercado

Plataforma	Raspberry Pi	Axis
Modelo	JACS_1	Q1615
Valor	R\$1500,00 ¹	R\$14,883,69

Fonte : Próprio autor.

Para trabalhos futuros seguem as recomendações:

- Mudança no módulo câmera para um modelo mais atual e com melhor qualidade.

A qualidade do módulo usado neste relatório foi satisfatória, porém como é bom sempre evoluir vários outros modelos podem ser explorado e testados.

- Mudança no *firmware* para inserção de laço virtual.

Ao invés das linhas bases de detecção serem fixas, estas seriam desenhadas na hora da instalação de acordo com a necessidade da medição.

- Conexão de um cabo Ethernet.

O uso deste cabo teria o objetivo de acessar o que a câmera está recebendo via protocolo IP.

Referências

- [1] R.C.Gonzalez e R.E.Woods. *Digital Image Processing*.Prentice Hall .*third edition*, 2008.

- [2] Sonaglio,Simara e Moecke ,Marcos. Detecção de movimento para sistema automático de vigilância por vídeo.Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

- [3] Curso de Filtragem de sinais. Disponível em
<http://www.dpi.inpe.br/~carlos/Academicos/Cursos/Pdi/pdi_filtros.htm> Acesso em:
10/28 dez. 2017.

- [4] Melhoria de Imagens por filtragens. Disponível em
<<http://www2.ic.uff.br/~aconci/aula-5-2015-AI.pdf>> Acesso em: 13/28 dez. 2017.

- [5] Filtragem de Imagens. Disponível em
<<http://computacaografica.ic.uff.br/transparenciasvol2cap5.pdf>> Acesso em: 29 dez.
2017.

- [6] Algoritmo de detecção de bordas. Disponível em <http://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/pos-graduacao/273-dissertacao_patricia.pdf> Acesso
em: 12/28 dez. 2017.

- [7] Dimensões Antropomórficas. Disponível em
<http://www2.anac.gov.br/arquivos/pdf/relatorio_final_projeto_conhecer.pdf>
Acesso em: 29 dez. 2017.