

CLAUDIO MARCIO CARDOSO AIRES

**Sistema de Vigilância Baseado em Localização de
Pessoas e Veículos**

São Carlos

2016

CLAUDIO MARCIO CARDOSO AIRES

**Sistema de Vigilância Baseado em Localização de
Pessoas e Veículos**

Trabalho de Conclusão de Curso
apresentado à Escola de Engenharia de São
Carlos, da Universidade de São Paulo

Curso de Engenharia de Computação

ORIENTADOR: Prof. Carlos Dias Maciel

São Carlos

2016

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTA TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

A298s Aires, Claudio Marcio Cardoso
Sistema de Vigilância Baseado em Localização de
Pessoas e Veículos / Claudio Marcio Cardoso Aires;
orientador Carlos Dias Maciel. São Carlos, 2016.

Monografia (Graduação em Engenharia de Computação)
-- Escola de Engenharia de São Carlos da Universidade
de São Paulo, 2016.

1. Rastreamento Autônomo. 2. Compartilhado. 3.
Sistema de Vigilância. 4. Solução Preventiva. 5.
Background Subtraction. 6. Python. I. Título.

FOLHA DE APROVAÇÃO

Nome: Claudio Marcio Cardoso Aires

Título: "Sistema de vigilância baseado em localização de pessoas e veículos"

Trabalho de Conclusão de Curso defendido em 28/11/2016.

Comissão Julgadora:

Resultado:

Prof. Associado Carlos Dias Maciel
(Orientador) - SEL/EESC/USP

Aprovado

Mestre Michel Bessani
Doutorando - SEL/EESC/USP

Aprovado

Mestre Tadeu Junior Gross
Doutorando - SEL/EESC/USP

Aprovado

Coordenador do Curso Interunidades Engenharia de Computação pela
EESC:

Prof. Dr. Maximilian Luppe

*A MINHA MÃE, PAI, PADRASTO E
IRMÃ QUE ATÉ AQUI ME APOIARAM
INCONDICIONALMENTE*

Agradecimentos

Primeiramente, agradeço a Deus por todas as bênçãos e graças obtidas. Agradeço por ter conseguido chegar até aqui e por tudo que ele reservou para o meu futuro.

Agradeço a minha mãe que me apoiou incondicionalmente abrindo mão de tudo por mim. Agradeço porque sempre ter me incentivado a estudar e buscar um futuro melhor. Sou grato por nunca ter me deixado desistir e sempre ter me apoiado em tudo, mesmo quando todos diziam que eu não conseguiria. Mãe, Muito Obrigado!

Agradeço ao meu pai, padrasto e irmã por terem me apoiado e estado ao meu lado em todos esses anos e por tudo que fizeram por mim. Agradeço a minha família por sempre terem estado ao meu lado.

Obrigado aos amigos, aos poucos, porém verdadeiros que Deus colocou em minha vida. Agradeço aos amigos que estão comigo desde a infância por terem me aguentado até aqui. Agradeço aos amigos que fiz na universidade, vocês ultrapassaram a barreira de meros colegas. Agradeço por ter tido a chance de conhecer todos vocês nessa jornada e espero poder tê-los por muitos outros anos. E principalmente, obrigado por nunca terem me deixado desistir.

Agradeço a professora Solange que a mais de 3 anos atrás, convenceu um garoto de que a faculdade estava apenas começando e que ele não deveria abandoná-la. Agradeço por todos os conselhos e orientações. Agradeço também ao Rafael pela ajuda e orientação durante aquele período.

Agradeço ao professor Maciel por ter me orientado e aconselhado nesses últimos anos e por ter me permitido e apoiado no desenvolvimento de projetos que me interessavam, mesmo quando esses estavam muitos distantes da sua área de atuação. Agradeço também pelo apoio durante as tentativas de fundar a minha empresa e principalmente pela sua amizade.

Agradeço ao Daniel L. pela amizade e ajuda durante este trabalho, pela sua disponibilidade e paciência comigo ao responder minhas infinitas dúvidas.

Agradeço ao Daniel A., por ter acreditado em mim e me dado uma oportunidade, por tudo que me ensinou e por ter sido compreensivo com os imprevistos gerados pela faculdade. Agradeço a todos que me receberam nessa nova fase da minha vida e por tudo que me ensinaram e estão ensinando.

Por último, agradeço a Mariani, que entrou na minha vida há uns três anos. Obrigado por estar comigo e por sempre me apoiar.

Sumário

Capítulo 1. Introdução	1
Capítulo 2. Teoria	3
2.1 Inteligência Artificial	3
2.2 Python.....	4
2.3 <i>OpenCV</i>	5
2.4 Remoção do plano de fundo	5
Capítulo 3. Modelagem do Problema.....	9
3.1 Sistema físico.....	9
3.2 Iluminação.....	11
3.3 Compartilhamento.....	15
Capítulo 4. Localização de Pessoas e Veículos.....	17
4.1 Método Background Subtraction	17
4.2 Aplicação Prática do Algoritmo.....	18
Capítulo 5. Resultados e Discussões.....	21
5.1 Sombras	21
5.2 Reflexos.....	24
5.3 Ambiente Controlado	27
Capítulo 6. Conclusões.....	31
6.1 Próximos passos.....	32

Resumo

Aires C. M. C. (2016). **Sistema de Vigilância Baseado em Localização de Pessoas e Veículos**. Trabalho de Conclusão de Curso – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Paulo, 2016.

Nos dias atuais, crimes como sequestros relâmpagos e assaltos passaram a ser realizados diretamente nos portões das residências, tornando a chegada ao lar, um momento de tensão e insegurança. Em vista de tal situação de perigo, as pessoas passaram a buscar por alternativas que lhes garantissem alguma sensação de segurança. Algumas das alternativas adotadas com intuito de diminuir o perigo são o patrulhamento por vizinhos das redondezas, que acaba os expondo ao perigo, e a contratação de empresas para monitoramento 24 horas, que por fazer uso de mão-de-obra humana a torna uma solução cara. Desse modo, o presente trabalho propõe um sistema de vigilância compartilhado, entre vizinhos e/ou moradores de uma mesma vizinhança, que realiza o rastreamento autônomo de pessoas e veículos que estejam se movimentando nas áreas vigiadas, através da análise de imagens com o uso do algoritmo de *Background Subtraction*, implementado em *Python*. Apresentando pequena taxa de distorção e erro nas respostas devido a presença de sombras e reflexos, decorrentes do não uso da técnica de calibragem no ambiente. Por outro lado apresentou as respostas esperadas, nas situações com variação de luz controlada e sob superfície com baixo índice de reflexão. Resultando assim em uma solução preventiva satisfatória e de baixo custo.

Palavras-chave: **Rastreamento autônomo, Compartilhado, Sistema de Vigilância, Solução Preventiva, Background Subtraction e Python.**

Abstract

Aires C. M. C. (2016). **Surveillance System Based on Location of People and Vehicles**. Trabalho de Conclusão de Curso – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Paulo, 2016.

Nowadays, crimes such as express kidnappings and assaults are carried out directly at the gates of the homes, making the arrival at home a moment of tension and insecurity. In view of such a dangerous situation, people began to look for alternatives that would guarantee them some sense of security. Some of the alternatives adopted to reduce the danger are the patrolling by neighbors in the vicinity, which ends up exposing them to danger, and the hiring of companies for monitoring 24 hours, which by making use of human labor makes it a expensive solution. In this way, the present study proposes a shared surveillance system, between neighbors and / or residents of the same neighborhood, that performs the autonomous tracking of people and vehicles that are moving in the surveyed areas, through the analysis of images using the Background Subtraction algorithm implemented in Python. It presents a small distortion rate and error in the responses due to the presence of shadows and reflections, that happen because to the non-use of the calibration technique in the environment. On the other hand, it presented the expected responses, in situations with controlled light variation and under surface with low reflection index. This results in a satisfactory and low cost preventive solution.

Keywords: **Autonomous Tracking, Shared, Surveillance System, Preventive Solution, Background Subtraction and Python.**

Capítulo 1.Introdução

Na atualidade crimes como sequestros relâmpagos, e assaltos passaram a ser realizados diretamente nos portões das residências. Devido a esses eventos, o lar deixou de transmitir uma sensação de segurança, e chegar em sua moradia se tornou um momento de tensão e insegurança. Em vista de tal situação de perigo, vizinhos passaram a se unir [1] em uma tentativa de diminuir os incidentes, como os relatados pela Folha de São Paulo¹. Uma das práticas mais usadas é a troca de mensagens, por meio de grupos em redes sociais, principalmente via WhatsApp. Tais mensagens têm como objetivo informar a situação das redondezas, e assim alertar aqueles que estão chegando em suas residências sobre eventuais riscos.

A fim de se obter tais informações as pessoas acabam se colocando em risco, uma vez que são os próprios moradores que normalmente fazem a patrulha, fazendo com que os patrulheiros se exponham ao perigo. Além disso, esse tipo de vigilância não é feito de modo contínuo, ocorrendo apenas em alguns períodos do dia. Uma das possíveis soluções para este problema é a utilização de câmeras, que evitariam a exposição ao perigo do responsável pela vigia, além de funcionarem 24 horas por dia.

O monitoramento remoto é um ponto vantajoso [2], pois permite a segurança do vigilante, uma vez que este não se expõe ao perigo. Entretanto, depender da intervenção humana para identificar possíveis movimentações suspeitas acaba sendo problemático. O ser humano é sujeito a falhas e distrações, que podem influenciar diretamente a eficácia do sistema [3]. Além disso, a mão-de-obra humana aumenta os custos do projeto, fato que tem motivado a busca por sistemas autônomos [4], como o uso de robôs em ambientes fabris².

Com intuito de obter uma solução confiável (baixo índice de falhas) a um baixo custo, seria preciso desenvolver um sistema que consiga de modo autônomo localizar pessoas e veículos que estejam na área vigiada [5], [6].

¹ *Um morador de SP é vítima de sequestro relâmpago a cada 5h; [Online]. Disponível em: <http://www1.folha.uol.com.br/fsp/especial/arquitet/index.html>, acessado em 31 de Outubro de 2016.*

² *Cada vez mais barato, robô já substitui até trabalhador chinês; [Online]. Disponível em: <http://economia.estadao.com.br/noticias/geral,cada-vez-mais-barato-roboto-ja-substitui-ate-trabalhador-chines,108113e>, acessado em 31 de Outubro de 2016.*

Objetivos

O objetivo geral deste trabalho é a proposição de um sistema de segurança que possua a funcionalidade de localizar pessoas e veículos e que possa ser utilizado de forma compartilhada. A parte de compartilhamento visa permitir que vizinhos compartilhem imagens de diferentes câmeras, desse modo aumentando a área vigiada.

Especificamente, os objetivos deste trabalho são:

- Demonstrar a implementação prática da parte do sistema que realiza a localização de pessoas e veículos.
- Realizar simulações em diferentes ambientes nos quais o sistema possa ser exposto.
- Apresentar descrição de um piloto (sistema) para a realização da tarefa de localização de pessoas e veículos.

Organização textual

O presente trabalho apresentará o projeto da criação de um sistema de segurança baseado em localização de pessoas e veículos. No Capítulo 2, é apresentada uma discussão sobre a teoria que serve como base para o trabalho. Já no Capítulo 3, é apresentada a modelagem do problema. O Capítulo 4 contém a descrição da solução para a localização de pessoas e veículos. No Capítulo 5, são apresentados os resultados: simulações de um sistema com uma *webcam*, que vigiará diferentes ambientes, e que localizará carros e pessoas presentes nos ambientes vigiados. E, por fim, o Capítulo 6 que apresenta as conclusões do trabalho.

Capítulo 2. Teoria

Neste capítulo são apresentadas as teorias que serviram de base para o trabalho desenvolvido. Na Seção 2.1 apresenta-se os conceitos de Aprendizado de Máquina e Inteligência Artificial, que são fundamentais para entender os demais conceitos e ideias apresentadas durante esse projeto (Capítulo 3 e Capítulo 4). Na Seção 2.2 introduz-se o *OpenCV*³, que é a biblioteca de funções utilizadas para o tratamento e processamento de imagens, que foi utilizada neste trabalho. Já na Seção 2.3 apresenta-se o conceito de *Background Subtraction* que foi o método escolhido para realizar a localização de pessoas e veículos nas imagens.

2.1 Inteligência Artificial

A inteligência artificial é um ramo da ciência da computação que estuda modelos matemáticos, de modo a possibilitar ao computador um comportamento inteligente parecido com o humano [7], [8]. Por se tratar de um conceito amplo e abstrato, recebe inúmeras definições [8], entretanto mesmo com tantas definições existem características que devem ser destacadas como pilares por trás do tema [8], como:

- Capacidade de raciocínio: Aplicação de um grupo de regras a um conjunto de dados disponíveis para se chegar a algum tipo de conclusão.
- Aprendizagem: Evoluir com os erros e acertos de modo ser mais eficaz em situações futuras.
- Reconhecimento de padrões: Localização de padrões, tanto os visuais e sensoriais, como aqueles derivados de comportamentos.
- Inferência: Capacidade de aplicação do raciocínio nas situações problemas, que forem apresentadas.

Apesar das diversas definições e diferentes usos, a aplicação da Inteligência Artificial tem sempre em comum o mesmo objetivo. Sendo este objetivo o desenvolvimento pela máquina da capacidade de interpretar situações e compreender os seus conceitos, e a partir dessa compreensão tomar decisões para realização de ações (tarefas) de modo autônomo, ou seja, sem que houvesse a intervenção de controle humano.

³ Vide mais sobre *OpenCV* em <<http://opencv.org/>>.

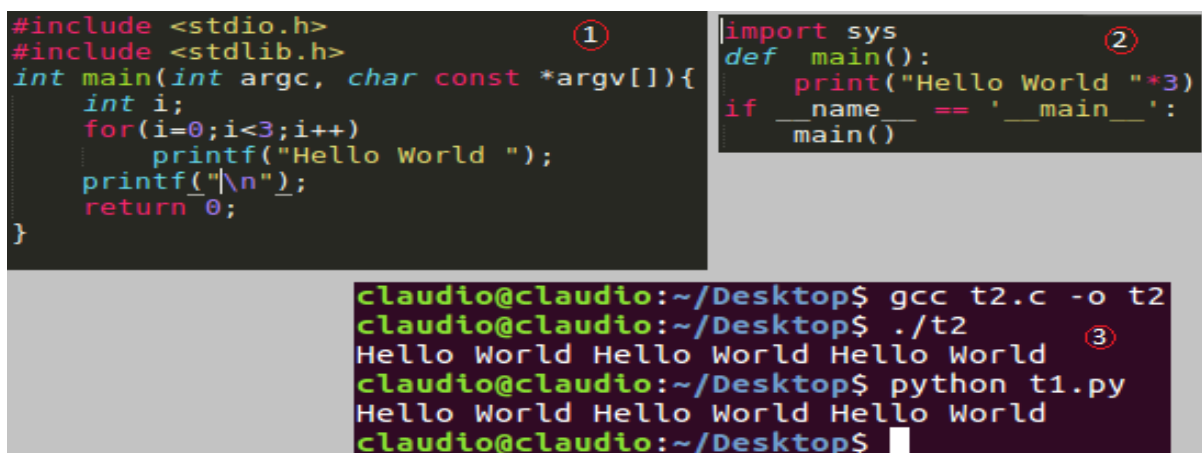
2.2 Python

Python é uma linguagem interpretada de script, de propósito geral de alto nível que foi projetada com a filosofia de enfatizar a legibilidade do código [9]. A linguagem suporta múltiplos paradigmas, incluindo orientação a objetos nos estilos imperativo, funcional e procedural (para mais informações acesse <https://www.python.org/>). Além disso, possui tipagem dinâmica, gerenciamento de memória automático e possui uma compreensiva e extensa biblioteca padrão.

A linguagem combina uma sintaxe concisa e clara com os recursos poderosos de sua biblioteca padrão e por módulos e frameworks desenvolvidos por terceiros. Devido a suas características, ela é principalmente utilizada para processamento de textos, dados científicos e criação de *CGIs* (*Common Gateway Interface*: Tecnologia que permite gerar páginas dinâmicas, permitindo a um navegador passar parâmetros para um programa alojado num servidor web. Logo um *script CGI* será um pequeno programa que interpreta esses parâmetros e gera a página após processá-los) para páginas dinâmicas para web [10].

A linguagem foi escolhida para o desenvolvimento desse projeto, devido a sua facilidade de implementação e sua grande variedade de funções disponíveis, como as do *OpenCV* (ver Seção 2.3). Sua facilidade de implementação pode ser facilmente notada quando se compara a quantidade de linhas necessária para implementar desde as mais simples até as mais complexas tarefas, um exemplo disso pode ser visto na Figura 1, apresentada abaixo.

Figura 1: Apresentação dos códigos em C (1) e em *Python* (2) que realizam o *print* da frase “*Hello World*“, por três vezes. O resultado das compilações dos códigos e suas respectivas chamadas pode ser visto na parte inferior da Figura (3).



```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char const *argv[]){
    int i;
    for(i=0;i<3;i++)
        printf("Hello World ");
    printf("\n");
    return 0;
}

import sys
def main():
    print("Hello World "*3)
if __name__ == '__main__':
    main()

claudio@claudio:~/Desktop$ gcc t2.c -o t2
claudio@claudio:~/Desktop$ ./t2
Hello World Hello World Hello World
claudio@claudio:~/Desktop$ python t1.py
Hello World Hello World Hello World
claudio@claudio:~/Desktop$
```

Fonte: Imagem gerada pelo autor.

2.3 OpenCV

O *OpenCV*⁴ é uma biblioteca de funções, desenvolvida pela Intel, que foi idealizada com objetivo de tornar a visão computacional acessível tanto a usuários como programadores em áreas como robótica e interações em tempo real entre humanos e computadores [11]. O *OpenCV*⁵ é uma biblioteca multiplataforma, de uso livre tanto para usos comerciais quanto para usos acadêmicos bastando seguir o modelo de licença BSD Intel, sendo esse uso livre (gratuito) um dos motivos da sua escolha para o desenvolvimento do projeto [12].

As funções da biblioteca *OpenCV*, podem ser divididas em cinco principais grupos:

1. Processamento de imagens [13];
2. Análise estrutural;
3. Análise de movimento e rastreamento de objetos;
4. Reconhecimento de padrões;
5. Calibração de câmera e reconstrução 3D.

Neste projeto são principalmente utilizados os grupos 1 (Processamento de imagens), 3 (Análise de movimento e rastreamento de objetos) e o 4 (Reconhecimento de padrões). A união desses três grupos forma o núcleo, da parte computacional desse projeto, que tem por objetivo a localização de pessoas e veículos.

A biblioteca foi desenvolvida nas linguagens de programação C/C++. Entretanto também dá suporte às linguagens Java, Python e Visual Basic, possibilitando assim seu uso independente da plataforma.

2.4 Remoção do plano de fundo

Background Subtraction [14] ou remoção do plano de fundo é um modelo para detecção do movimento de objetos em gravações de câmeras estáticas (fixas). A detecção do movimento no vídeo é realizada a partir da diferença entre o atual *frame*

⁴ Eosgarden, *OpenCV for iOS* (2013). Disponível em: <<http://www.eosgarden.com/en/opensource/opencv-ios/overview/>>, acessado em 17 de Outubro de 2016.

⁵ *University of Leeds, Leeds Guide to OpenCV* (2006). Disponível em: <<http://www.comp.leeds.ac.uk/vision/opencv/whatisit.html>>, acessado em 17 de Outubro de 2016.

(quadro) e o *frame* de referência, que nesse caso recebe o nome de “*background image*” (imagem do plano de fundo).

Sendo que essa imagem do *background* precisa ser uma representação do cenário em que não existam objetos se movendo, como apresentado abaixo na Figura 2.1. Tal representação precisa ser atualizada regularmente para que ocorra a adaptação às variações de luminosidade e o reconhecimento de novos objetos estáticos. Desse modo o algoritmo possui um tratamento para situações tanto de modificações provenientes de variações de iluminação quanto as de modificação de cenário, adaptando assim as diversas situações que poderiam ocorrer nos mais variados cenários, de modo a não limitar seu uso a aplicações específicas. Existem ainda modelos mais complexos para o método, cuja abordagem vai muito além do simples conceito de remoção do plano de fundo.

Figura 2.1: Representação de um ambiente sem objetos (pessoas ou veículos) se movendo, obtido através de uma câmera estática (fixa).

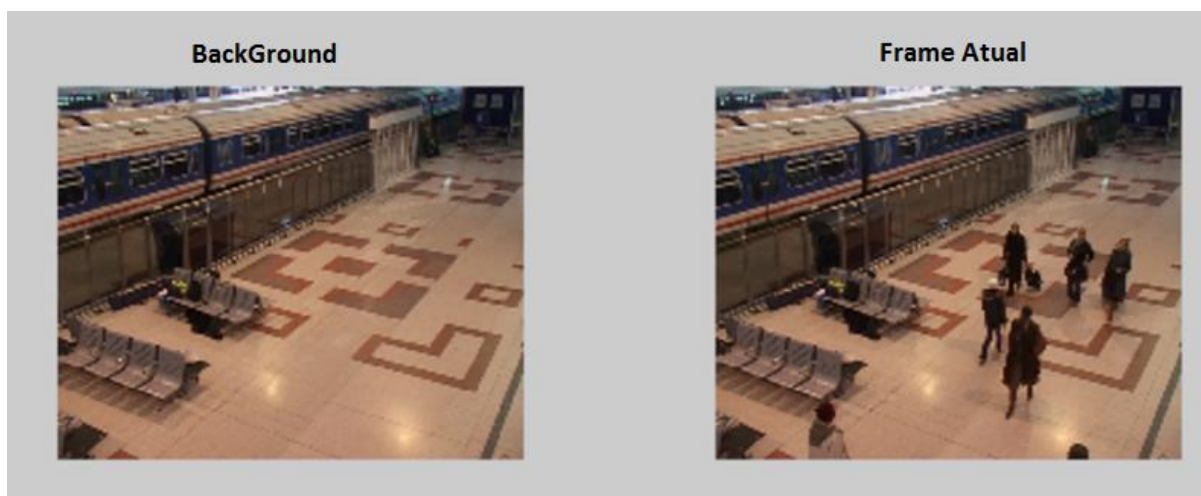


Fonte: Imagem adaptada de vídeo da *BGSLibrary*⁶ (2016).

Como será mais bem detalhado na Seção 3.3.1, neste trabalho foi utilizado uma versão do algoritmo que estima o plano de fundo (*background*) através de uma sequência de frames, ordenados cronologicamente, ou seja, o algoritmo identifica autonomamente a parte estática do cenário, como exemplificado na Figura 2.2 apresentada na sequência.

⁶ *BGSLibrary, Dataset (2016)*. Disponível em: <https://github.com/andrewssobral/bgslibrary>, acessado em 03 de Novembro de 2016.

Figura 2.2: Exemplo de aplicação do método de Remoção do plano de fundo. A direita é apresentada a imagem atual (visualização presente do ambiente), e na esquerda a de referência (visualização do ambiente base para a comparação, ou seja, o *BackGround*).



Fonte: Imagem adaptada de *Pantechsolutions*⁷ (2013).

⁷ *Pantechsolutions, Matlab Code for BackGround Subtraction (2013)*. Disponível em: <<https://www.pantechsolutions.net/blog/matlab-code-for-background-subtraction/>>, acessado em 28 de Outubro de 2016.

Capítulo 3. Modelagem do Problema

Nas grandes cidades a violência já se tornou algo presente na vida da maioria das pessoas, de tal modo que nem mesmo as respectivas residências transmitem sensação de segurança para aqueles que nelas habitam. O simples ato de chegar ao seu lar se tornou extremamente perigoso, e um grande alvo para os criminosos. Sendo o momento no qual a pessoa parou o carro para esperar o portão abrir ou mesmo no momento que esta pessoa desce do veículo para abrir o portão, o momento preferido dos criminosos. Esse período mínimo de vulnerabilidade passou a ser utilizado para realização de crimes como assaltos e sequestros relâmpagos.

Com intuito de aumentar as chances de prevenção desse tipo de situação, é proposto neste trabalho um sistema que consiga autonomamente localizar pessoas e carros que estejam localizados dentro das áreas vigiadas. Pelo fato de ser uma máquina a responsável pela localização e não um humano, o sistema se torna capaz de atingir uma menor taxa média de erro do que o de um sistema baseado em decisão humana. Além de obter capacidade de operação ininterrupta a um custo extremamente baixo quando comparado ao custo da mesma operação envolvendo mão-de-obra humana.

3.1 Sistema físico

Fisicamente o sistema consiste de algumas câmeras espalhadas pela rua, um computador e um roteador ou cabos de conexão entre o computador e as câmeras. O número total de câmeras depende da dimensão da área na qual se deseja vigiar, variando desde uma, para o caso em que se desejaria vigiar apenas a entrada da casa ou um ambiente interno (como a garagem), até 15 ou mais, no caso de grandes condomínios. A quantidade de câmeras poderá ser limitada pelo processamento da máquina, uma vez que o uso de um número de câmeras superior ao recomendado para o *hardware* daquela máquina acarretaria em atrasos na resposta do sistema, que levaria a sua inefetividade. Por não se tratar de um sistema crítico (ou sistema de tempo real) [15], atrasos são tolerados, entretanto apenas dentro de um pequeno intervalo (entre a gravação da imagem e a entrega da análise com a localização das pessoas e dos carros, deve ocorrer em no máximo 30 segundos), caso esse atraso se seja maior que isso, ocorreria quase que uma inutilização do sistema para a situação de prevenção para o qual foi proposto, já que esse atraso pode significar a diferença entre ficar a salvo e ser vítima de alguma ação criminosa.

Abaixo é apresentado na Figura 3, um diagrama do sistema de modo a exemplificar uma possível aplicação do mesmo.

Figura 3: Exemplificação de uma aplicação real do sistema, para vigilância da entrada de uma casa. Em amarelo são representados os fios de conexão entre o computador, com o software, e as câmeras. No computador representado acima, é exemplificado uma possível tela de visualização de resultado, neste caso a rua estava vazia e por isso não ocorre qualquer marcação.



Fonte: Adaptação de imagens retiradas da internet, Seguritecsistemas⁸ (2016), RS Micro⁹ (2015) e Cftv¹⁰ (2014).

⁸ Seguritecsistemas, Câmera de Segurança (CFTV). Disponível em: <<http://www.seguritecsistemas.com.br/Solucoes/10/Cameras-de-Seguranca-CFTV>>, acessado em 26 de outubro de 2016.

⁹ RS Micro, Computador Intel i7 4ª Geração. Disponível em: <<http://www.rsmicro.com.br/produto/11/computador-intel-i7-4-geracao>>, acessado em 26 de outubro de 2016.

¹⁰ Cftv, Fazer instalação de segurança em 2 passos. Disponível em: <<http://www.cftv.tv.br/blog/>>, acessado em 26 de outubro de 2016.

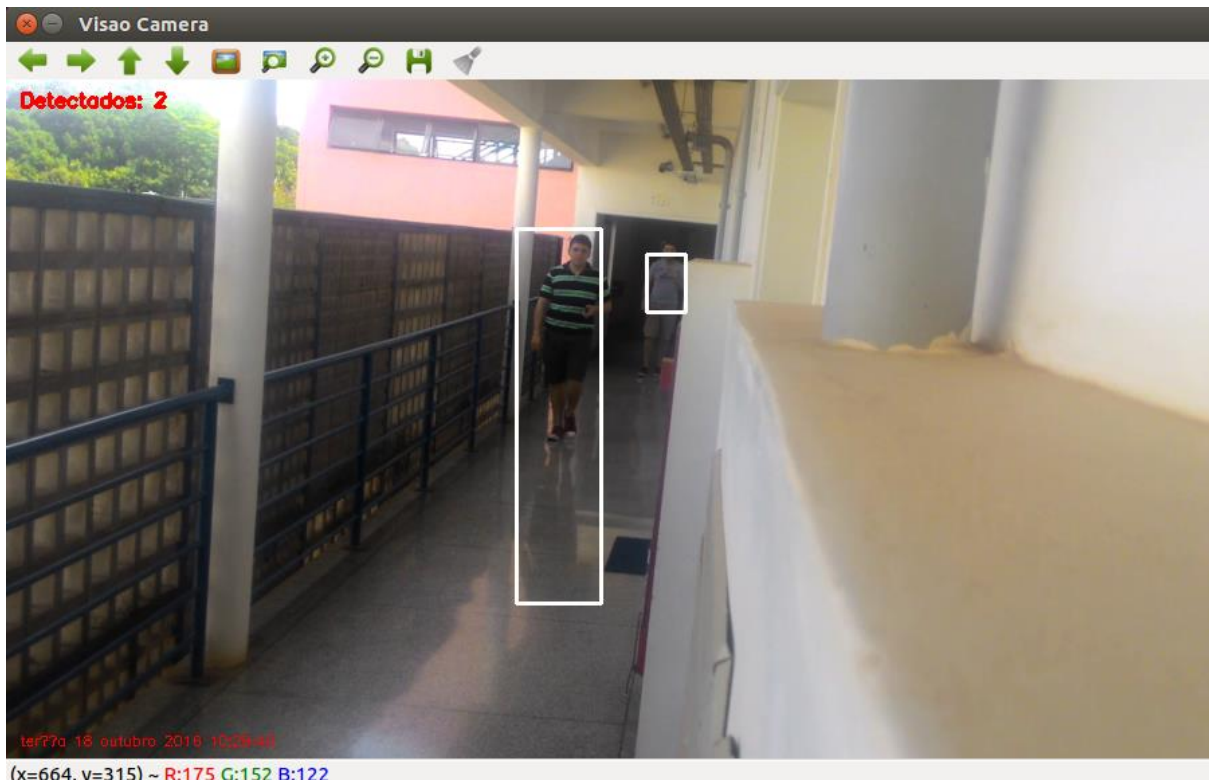
3.2 Iluminação

Por se tratar um sistema que terá seu maior uso em ambientes externos, o tratamento para as variações de iluminação se torna indispensável. De tal modo tratar essas variações é também, senão o maior dos desafios um dos maiores desafios desse projeto.

Dois principais fatores que são diretamente influenciados pelas variações de luz no ambiente, podem ser destacados, o reflexo e a sombra [16]. Mesmo se tratando de dois fatores habituais para qualquer problema que envolva luz, no sistema abordado por este projeto, eles são pilares fundamentais.

Em um caso básico, simplesmente tratar uma situação no qual se tenha uma grande taxa de iluminação, nesse caso ocorreria um excesso de reflexão de luz nas superfícies, o que decorreria de uma deformação na imagem obtida, causando assim uma resposta deformada, como apresentado nas Figuras 4.1 e 4.2 abaixo, uma vez que a lente da câmera poderia ser facilmente atingida por uma alta incidência de luz derivada de uma dessas reflexões.

Figura 4.1: Resposta deformada do sistema, devido ao sistema entender como um único corpo a junção entre uma pessoa e seu reflexo no piso.



Fonte: Imagem gerada pelo autor.

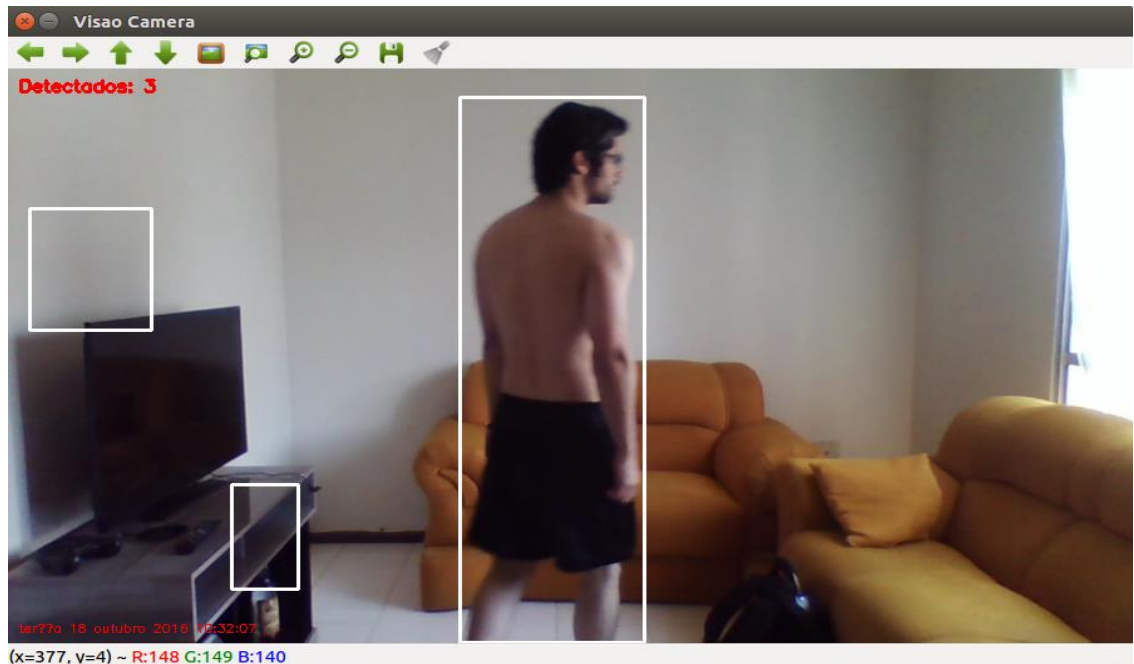
Figura 4.2: Resposta deformada do sistema, na qual ele reconhece incorretamente o reflexo no piso de uma pessoa, como sendo outra pessoa.



Fonte: Imagem gerada pelo autor.

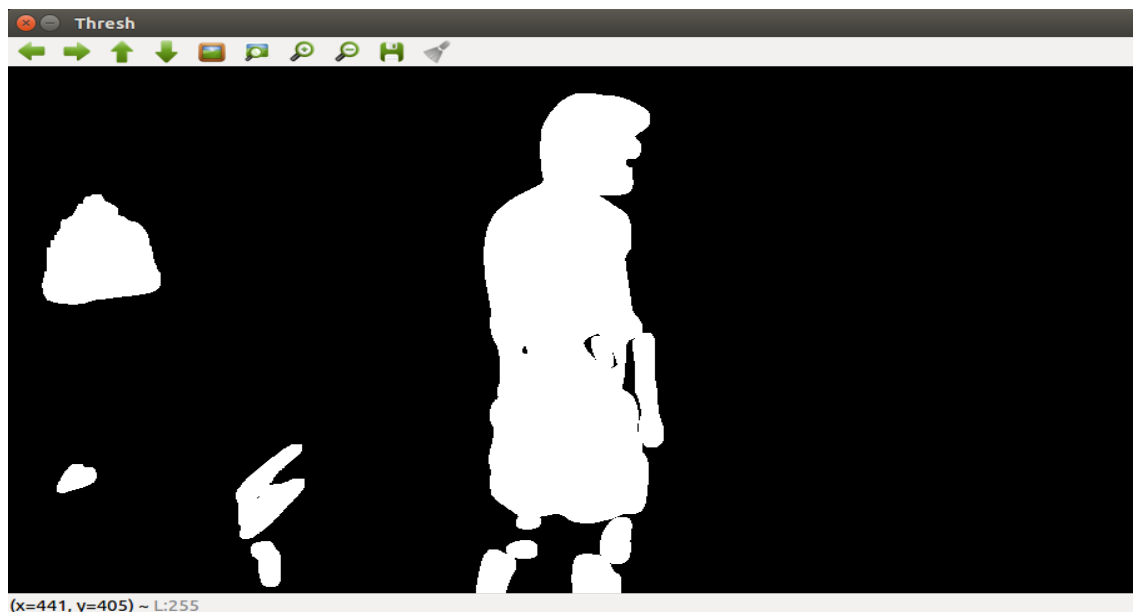
No caso das sombras, a deformação criada nas imagens pela aparição delas, não é tão abrupta, entretanto ainda assim é significativa. Em um caso simples, a aparição da sombra de uma pessoa poderia confundir os algoritmos e fazer com que a máquina considere essa sombra como uma outra pessoa, como mostrado nas Figuras 4.3 e 4.4 abaixo, em outro caso também seria possível a deformação da forma de uma pessoa ou veículo devido à máquina entender que a sombra e a pessoa ou veículo dono dessa sombra são a mesma coisa, assim como apresentado na Figura 4.5 abaixo.

Figura 4.3: Resposta incorreta do sistema, que considera partes da sombra da pessoa no centro da foto, como sendo outras duas pessoas.



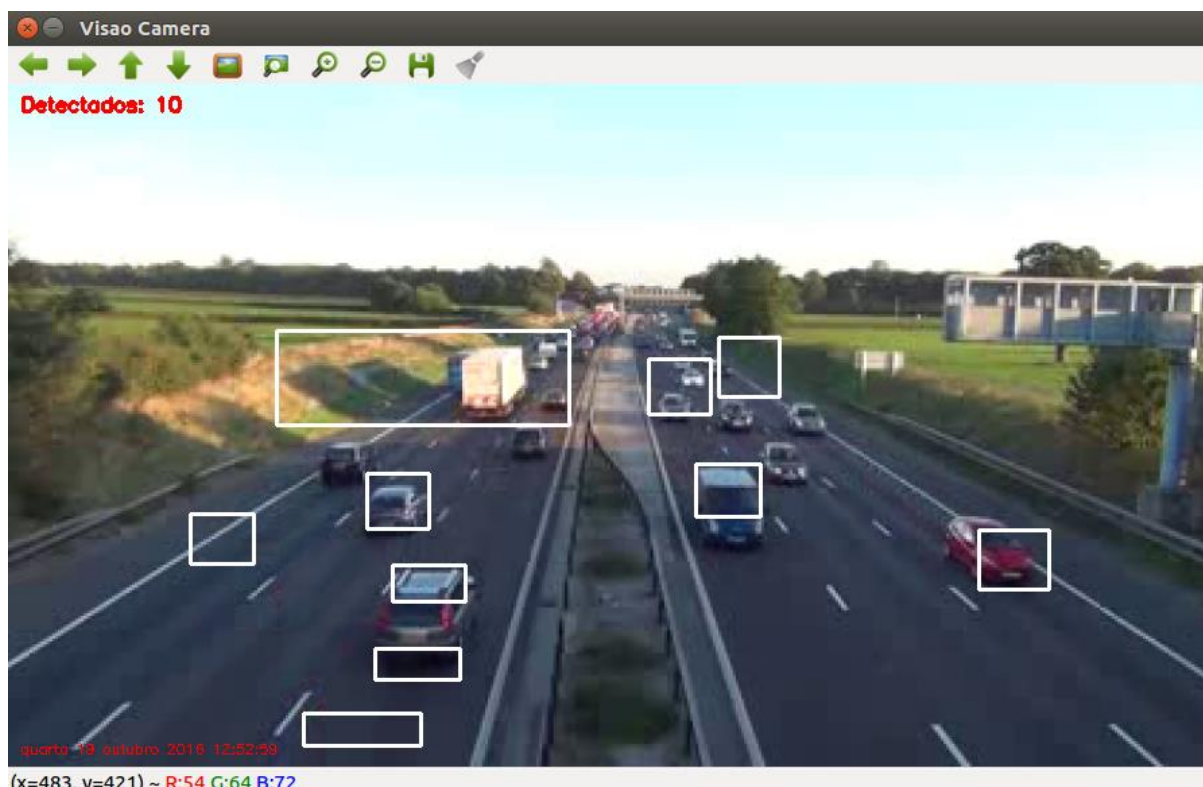
Fonte: Imagem gerada pelo autor.

Figura 4.4: Visão do sistema, após a subtração do *Background* (*frame* de referência), da imagem C, apresentada acima. Sendo destacados em branco o local onde se encontram os objetos se movendo. Com isso é possível ver os dois objetos marcados como corpos na Figura acima, destacados aqui também.



Fonte: imagem gerada pelo autor.

Figura 4.5: Resposta incorreta do sistema que considera o caminhão (lado superior esquerdo), sua sombra e a sombra de outro veículo a frente do caminhão como um único corpo.



Fonte: Imagem adaptada de vídeo da *DriveCamUK*¹¹ (2013).

Desse modo utilizando pequenas variações de iluminação é possível causar grandes danos aos resultados, desde a deformação da resposta até a apresentação de respostas incorretas. Logo nota-se a necessidade da adaptabilidade do algoritmo para que este possa evoluir e se adaptar às mudanças de iluminação de um ambiente.

É importante ainda especificar que este projeto não aborda casos nos quais não existe alguma iluminação no ambiente. Mesmo com as soluções de algoritmos propostas se baseando em adaptabilidade, e logo serem capazes de atuarem em ambientes sem iluminação, não ocorreu qualquer teste para esse tipo de situação. Em caso da câmera possuir modo noturno e coletar imagens em infravermelho, os algoritmos possuem embasamento para se adaptarem e realizarem o processo de localização, assim como será

¹¹ *DriveCamUK, M6 Motorway Traffic (2013)*. Disponível em: <https://www.youtube.com/watch?v=PNCJQkvALVc>, acessado em 28 de Outubro de 2016.

apresentado no Capítulo 4. Casos para ambientes sem qualquer tipo de iluminação, como os citados anteriormente serão um dos objetivos da Seção 6.3 deste trabalho.

3.3 Compartilhamento

Como mencionado anteriormente, na Seção 1.1, este sistema foi idealizado para ser utilizado de forma compartilhada por vizinhos. O compartilhamento tem como finalidade reduzir o custo do projeto para cada participante e aumentar a área vigiada, desse modo aumentando as chances de prevenção de crimes.

Com a redução dos custos do projeto, devido ao compartilhamento, possibilitamos uma maior inserção do sistema, além de aumentar o *range* do público alvo. Outro apelo para essa abordagem é o grande crescimento das redes sociais, que se baseia no compartilhamento de informações, desse modo partiríamos do compartilhamento de informações entre vizinhos para atingir o objetivo de aumentar as chances de prevenção do sistema.

Para uma aplicação prática, suporemos uma rua com três casas de cada lado, para realizar a vigilância em um cenário normal cada casa teria de possuir duas ou três câmeras, para vigiar uma área aceitável de modo a garantir alguma segurança. Logo seriam entre doze e dezoito câmeras na rua e cada vizinho possuiria apenas as imagens de suas câmeras, limitando assim as visões individuais. Para o mesmo cenário utilizaremos o conceito de compartilhamento proposto neste projeto, utilizando algo entre seis e nove câmeras espalhadas entre as casas, de modo que as imagens combinadas das câmeras consigam vigiar a rua inteira.

Com essa simples mudança se reduziu o custo individual que era o de duas a três câmeras para algo entre o custo de uma câmera ao custo de uma câmera e meia, ou seja, uma redução de 25% a 50% do custo individual. Além de se ter aumentado a área vigiada de alguns metros em frente ao portão de entrada, para a área da rua inteira. Vigiando a rua inteira, dificulta que criminosos possam se esconder do sistema, o que levaria a um aumento na sua eficácia.

Para possibilitar tal ato de compartilhamento o sistema necessitaria de um *Website* e/ou um *App* (Aplicativo *mobile*), de modo a fornecer o acesso simultâneo de várias pessoas localizadas em diferentes lugares. Entretanto a implementação desta parte do projeto está fora do escopo deste trabalho, sendo abordada como um dos objetivos da Seção 6.3.

Capítulo 4. Localização de Pessoas e Veículos

O principal diferencial do projeto, perante outros sistemas de vigilância, é fornecer ao usuário justamente a localização de pessoas e veículos [17], que estejam nas áreas monitoradas. E desse modo permitir ao usuário a chance de tomar uma decisão baseada em alguma análise, e aumentar suas chances de evitar ser vítima de algum crime. A fim de obter um resultado aceitável e ainda possibilitar uma fácil adaptabilidade às mudanças no ambiente tanto as resultantes de efeitos de luz e sombra, como as provenientes de agentes externos, como por exemplo objetos ou veículos adicionados ao ambiente, escolheu-se o algoritmo de *Background Subtraction*, como meio para solucionar tais necessidades e obter os resultados pretendidos.

4.1 Método Background Subtraction

Como explicado na Seção 2.4, o *Background Subtraction* consiste de uma técnica utilizada pelo computador para aprender e isolar o plano de fundo da imagem, ou seja, a parte estática da imagem e com isso o utiliza como base para identificar as futuras variações nos frames. A partir dessas variações o algoritmo então detecta o movimento das pessoas e veículos.

Neste trabalho foi utilizado uma versão do algoritmo que estima o plano de fundo (*background*) através de uma sequência de frames, ordenados cronologicamente, ou seja, o algoritmo identifica autonomamente a parte estática do cenário. Após definido uma referência, ocorre então a comparação com o frame atual e a detecção dos objetos que estejam se movimentando. Como a definição da parte estática do cenário é contínua, o algoritmo consegue se adaptar às variações de cenário. Tendo apenas dificuldade em lidar com variações bruscas, pois estas ocorrem mais rapidamente que o tempo de aprendizagem da máquina, levando assim a um pequeno erro. Entretanto caso essa grande variação persista o algoritmo então se adapta a ela e volta a sua taxa normal de assertividade. De modo simples, pode-se inferir que quanto maior o tempo de exposição do algoritmo ao cenário (tempo de gravação do ambiente) e menores forem as variações, melhor será o resultado.

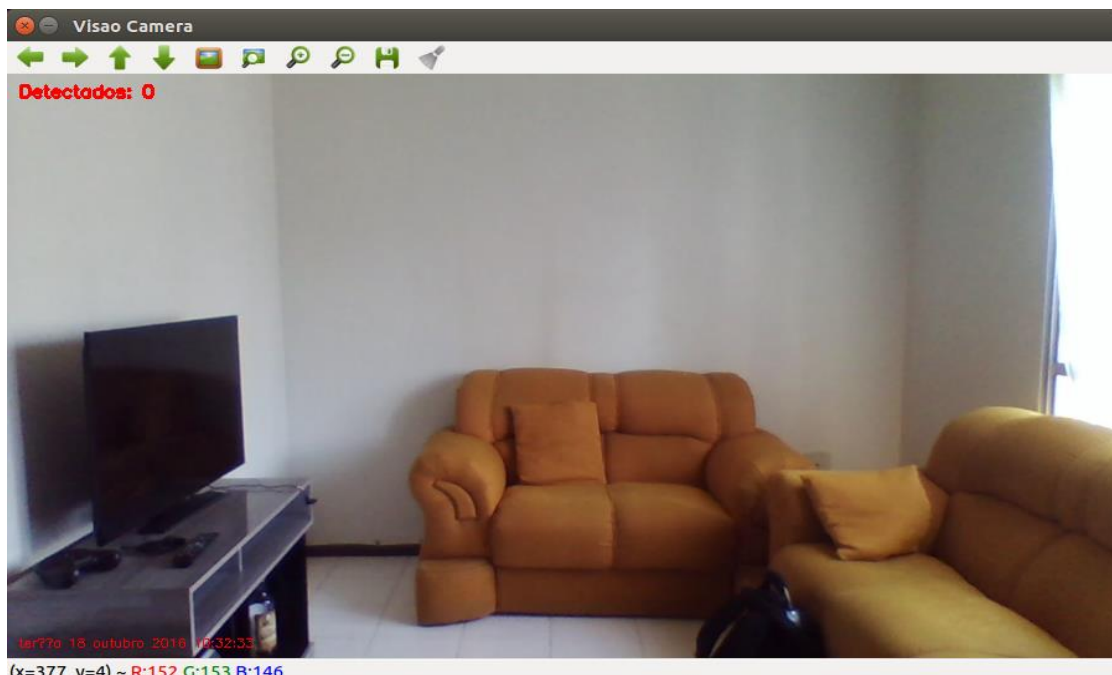
4.2 Aplicação Prática do Algoritmo

A partir do modelo do algoritmo de Background Subtraction descrito na Seção 4.1 acima e baseado nos conceitos teóricos para o método, apresentados na Seção 2.3, podemos dividir o processo para a localização de pessoas e veículos que foi utilizado neste trabalho em 4 principais partes, listadas abaixo, de modo a ilustrar o funcionamento do método e a obtenção dos resultados.

- I. Identificação do plano de fundo (*frame* de referência).
- II. Captura do frame atual.
- III. Comparação entre o frame atual e o de referência e localização dos *pixels* (pontos) diferentes nas imagens.
- IV. Uso da diferença dos frames, para localização dos objetos em movimento e a apresentação do resultado obtido com o desenho de um quadrado ao redor do objeto detectado.

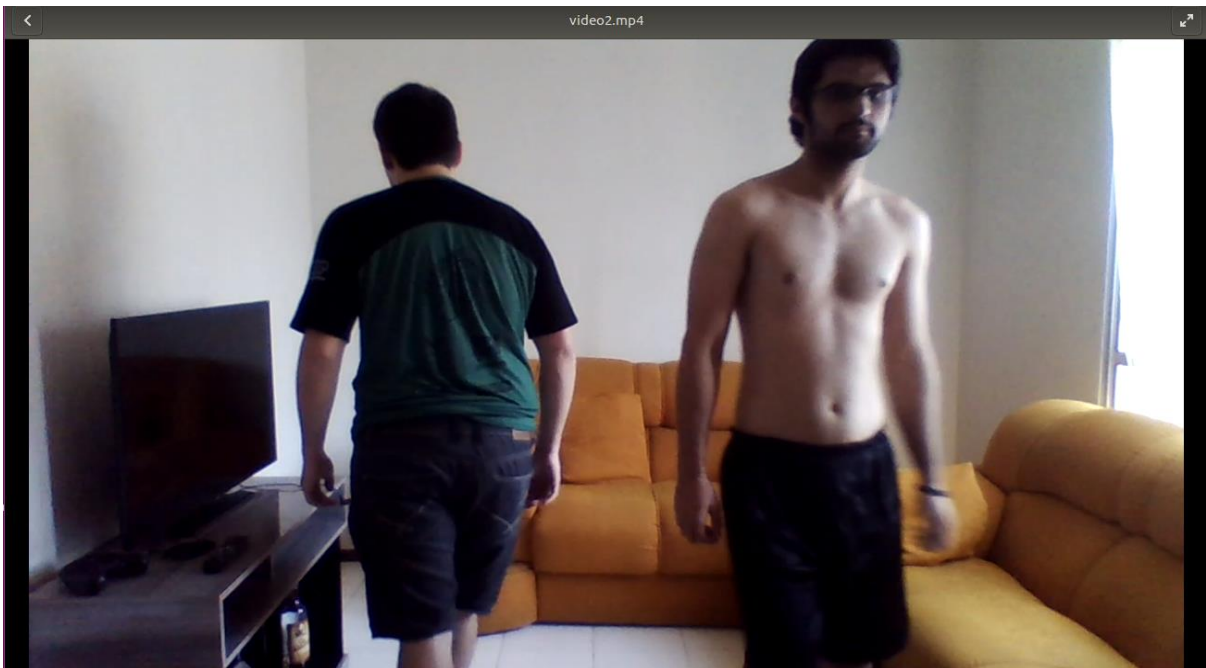
As Figuras 5.1, 5.2, 5.3 e 5.4 apresentadas abaixo, demonstram os passos citados acima durante a execução do algoritmo. A implementação desse algoritmo pode ser vista em anexos, Código do Sistema de vigilância baseado em localização de veículos e pessoas.

Figura 5.1: Definição de um plano de fundo (frame de referência), a partir da localização da parte estática (fixa) do ambiente vigiado.



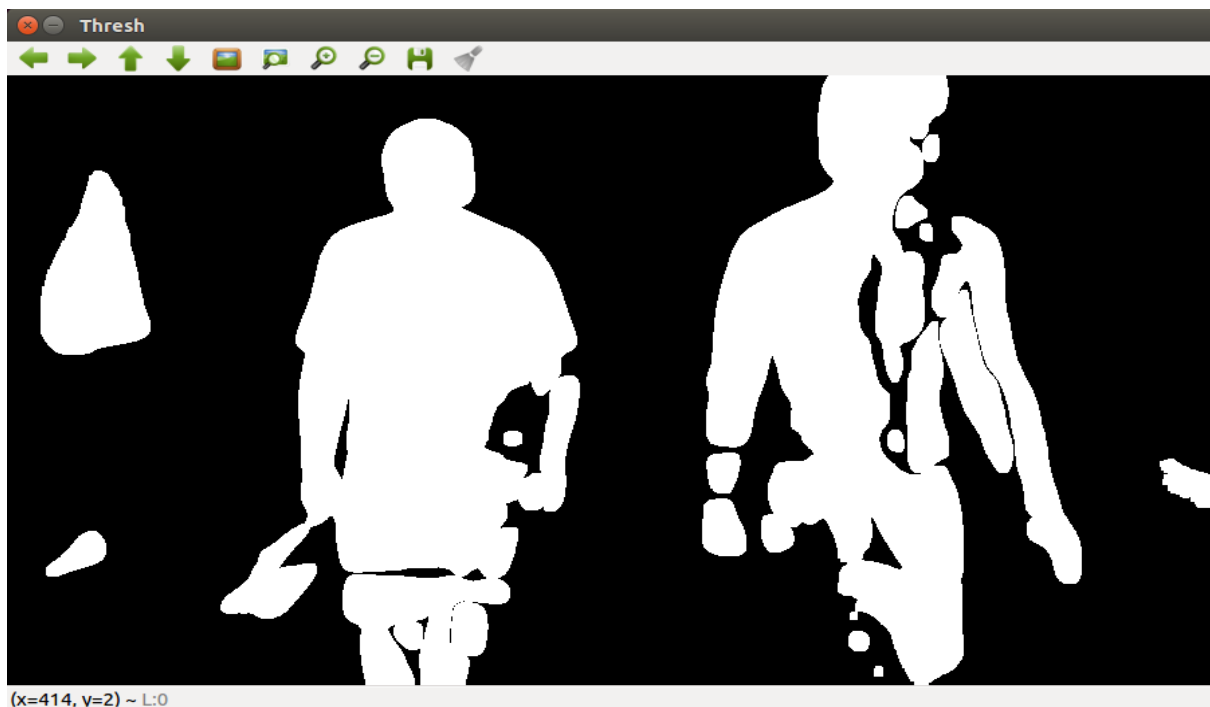
Fonte: imagem gerada pelo autor.

Figura 5.2: Frame atual capturado pela câmera, contendo duas pessoas se movendo em direções opostas.



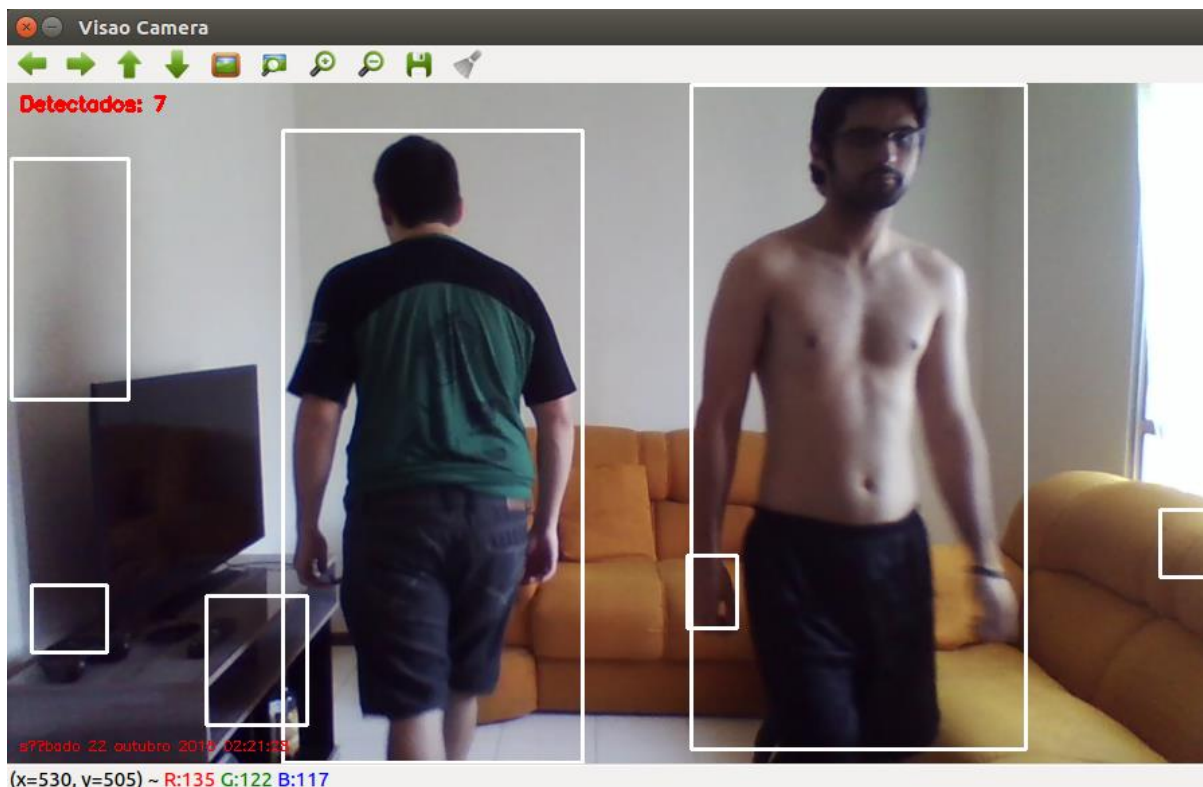
Fonte: imagem gerada pelo autor.

Figura 5.3: Resultado da diferença entre o *frame* atual e o de referência. Em branco se encontram todos os pixels do *frame* atual, que estão diferentes dos encontrados no *frame* de referência. Representando assim a localização dos objetos em movimento.



Fonte: imagem gerada pelo autor.

Figura 5.4: Resultado final apresentado pelo sistema, após a detecção dos objetos (pessoas) em movimento. Localização das duas pessoas como esperado, entretanto detecção de mais objetos devido à formação de sombras.



Fonte: imagem gerada pelo autor.

Capítulo 5. Resultados e Discussões

No Capítulo 3, descreveu-se os possíveis esquemas físicos para o sistema, partindo de uma implementação utilizando um computador (desktop ou notebook). Também foi descrito as principais situações e dificuldades que o sistema enfrenta devido às variações de iluminação.

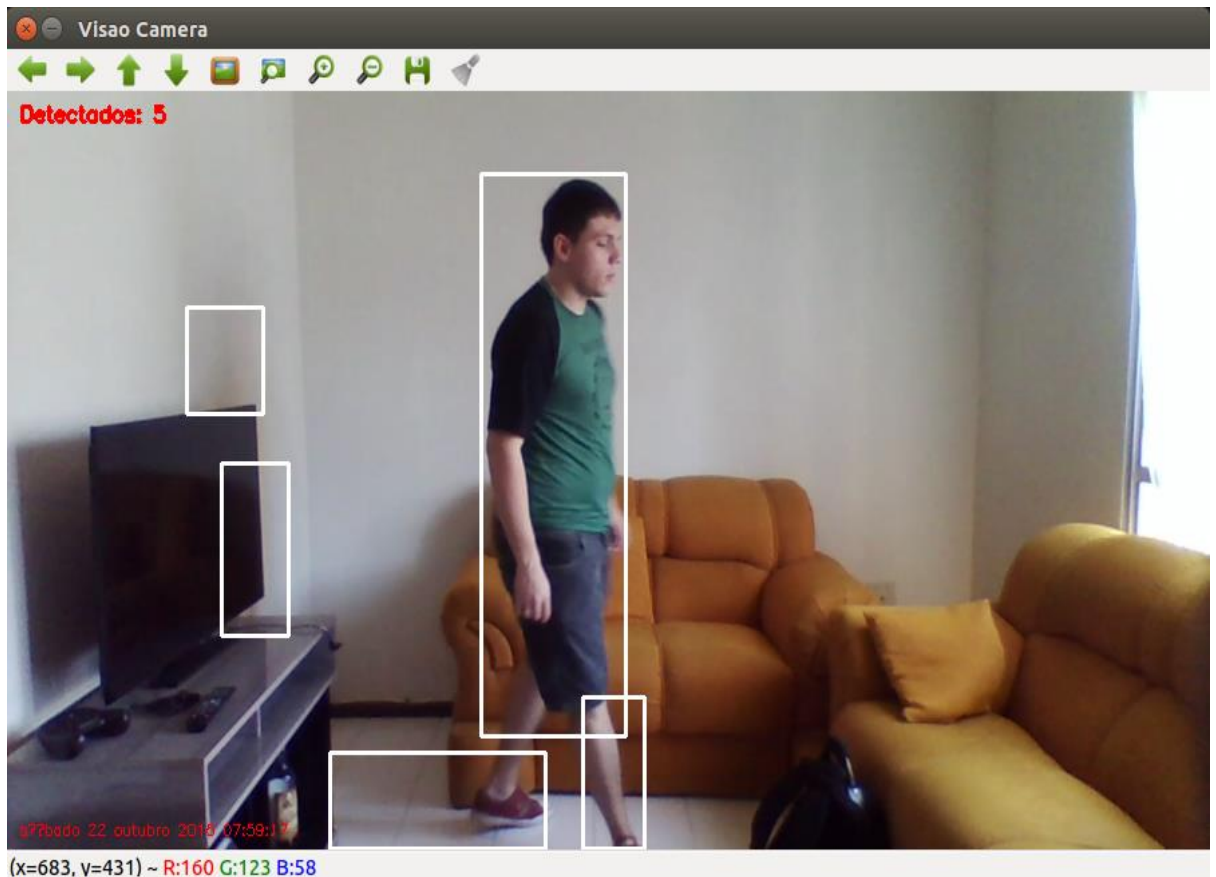
Já no Capítulo 4 apresentou-se em mais detalhes o funcionamento do algoritmo de *Background Subtraction* utilizado para a localização de pessoas e veículos nas imagens capturadas pelo sistema. E na Seção 4.2, foi demonstrada uma aplicação prática do método, com intuito de exemplificar o processo e justificar a obtenção dos resultados.

Como descrito na Seção 3.2, um dos maiores desafios enfrentados pelo sistema é a variação de luminosidade no ambiente, causando assim o surgimento de reflexos e sombras, que podem gerar distorções nos resultados.

5.1 Sombras

Qualquer ambiente iluminado que possua objetos ou pessoas nele irá possuir também as respectivas sombras, podendo ser desde sombras pequenas quase imperceptíveis até sombras maiores que o objeto ou pessoa que a produziu, tudo dependendo do ângulo de incidência da luz. A distorção do resultado apresentado pelo projeto, devido ao surgimento de sombras, pode ser visto na Figura 6.1.

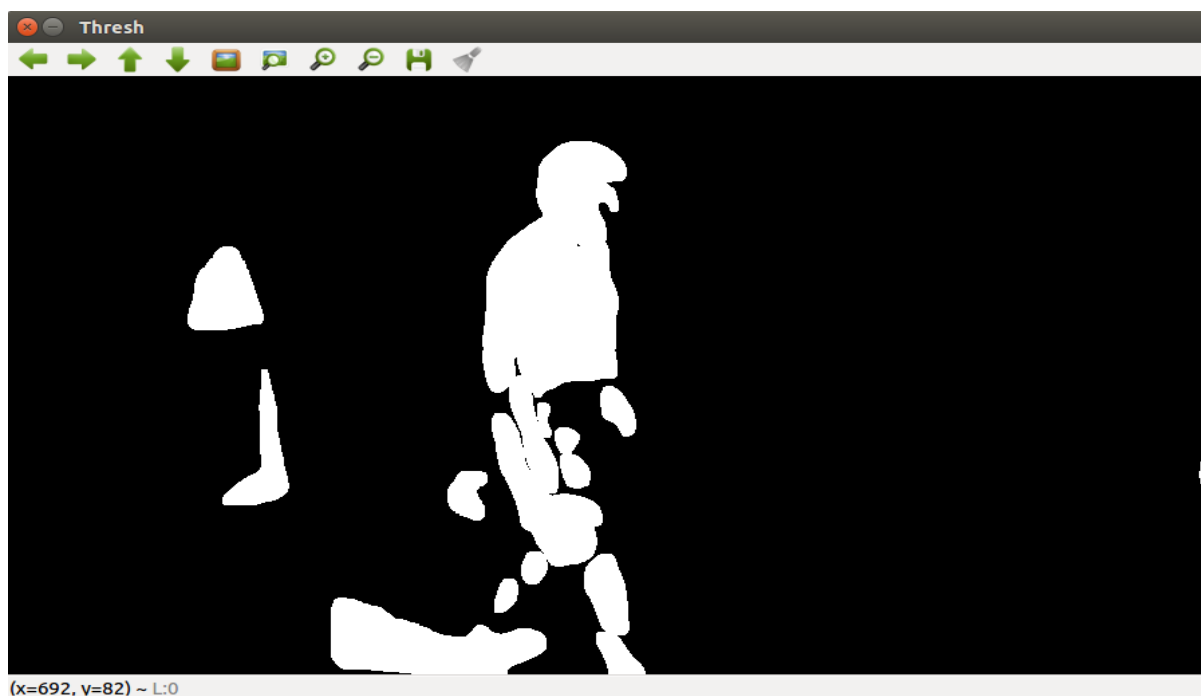
Figura 6.1: Resposta incorreta gerada pelo sistema, devido à presença de sombras na imagem. Sendo localizado a pessoa e sua perna no centro da imagem e três outras partes (a esquerda da imagem, nas partes inferior e central) da sombra dessa pessoa.



Fonte: imagem gerada pelo autor.

Na Figura 6.1, é possível notar o aparecimento das sombras, devido a existência da janela no lado direito da imagem. O sistema então acaba localizando 3 pontos de sombras, além de marcar outros dois objetos, sendo um deles a pessoa e outro a perna desta pessoa. Em um caso teórico ideal, o sistema deveria ter identificado apenas a pessoa, entretanto encontrou outros 4 objetos, no caso da perna desta pessoa, ela é identificada devido a uma descontinuidade causada pela variação de luz. Tal descontinuidade, assim como as sombras produzidas podem ser mais facilmente vistas abaixo, na Figura 6.2.

Figura 6.2: Visão após a diferença entre o frame atual e o de referência. Apresentando a localização da pessoa no centro da imagem e de suas sombras no lado esquerdo (inferior e central). Além de demonstrar as segmentações na zona onde está localizada a perna do indivíduo capturado na imagem.



Fonte: imagem gerada pelo autor.

Com o uso da Figura acima, se torna mais perceptível a descontinuidade que ocorreu na região do joelho e que acabou por confundir o sistema e fazê-lo contabilizar a perna da pessoa como outra pessoa. A confusão e a criação dessa descontinuidade ocorreu devido a qualidade da câmera usada, que nesse caso foi a webcam integrada de um notebook. Devido a baixa resolução, a incidência da luz solar e a criação de uma sombra pela bermuda nesse ponto foi então criado esse pequeno equívoco no resultado.

Como visto anteriormente na Figura 6.2 o sistema localiza as sombras, isso ocorre devido ao fato de as sombras criadas pela pessoa que aparece na foto, não estarem presentes no frame de referência, logo elas representam uma modificação no ambiente e por isso o sistema as marca com o retângulo branco.

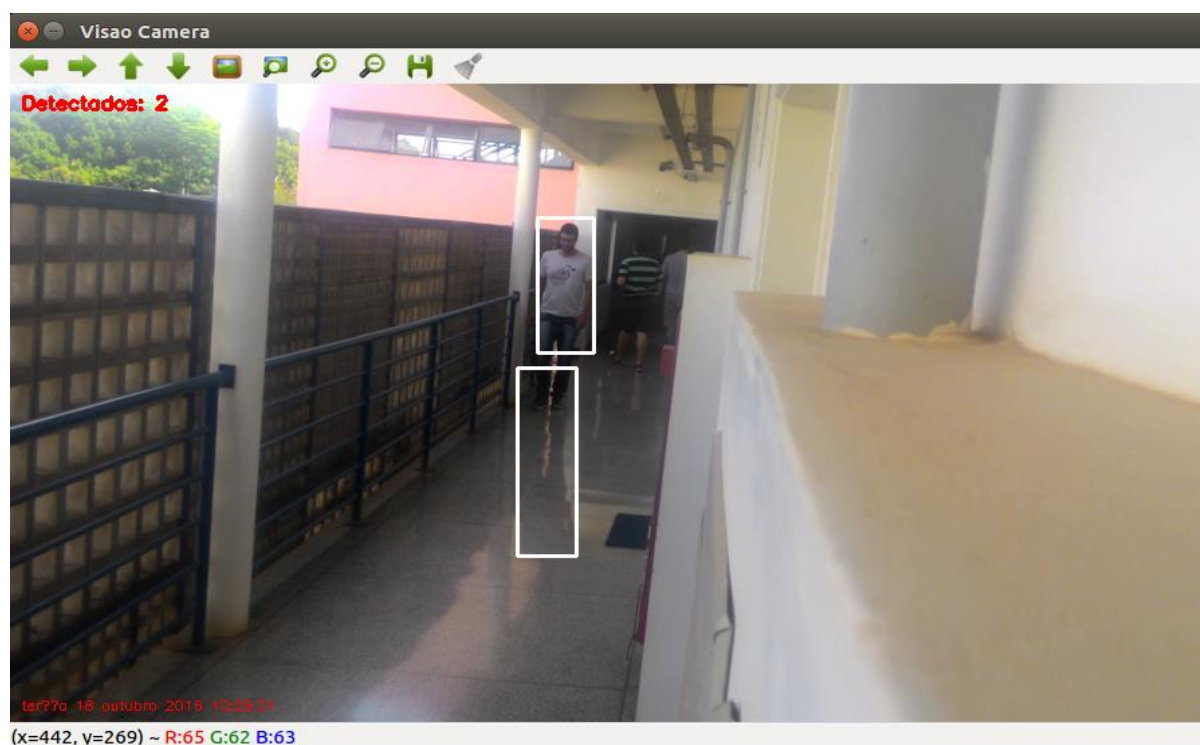
Assim como no caso das sombras essa pequena descontinuidade, está dentro do erro esperado para o sistema proposto. Por se tratar de um sistema idealizado para ser usado de maneira preventiva e com baixo requisitos de equipamento, pequenas deformações no resultado são toleráveis. Sendo que nesse caso, o uso de uma câmera de melhor resolução já seria o suficiente para resolver o problema e evitar o equívoco do sistema, prevenindo assim a formação dessas descontinuidades.

Entretanto os erros apresentados como resposta do sistema nas imagens acima, demonstram que em uma aplicação real, o sistema seria capaz de identificar a presença de um criminoso que estivesse se escondendo atrás de um carro ou muro e que deixasse a vista apenas uma pequena parte do corpo. Logo esse pequeno erro, demonstra a capacidade do sistema de detectar até mesmo as menores anomalias no ambiente e assim atingir seu macro objetivo e colaborar com o usuário na sua prevenção contra atos criminosos.

5.2 Reflexos

Um outro problema causado pela variação de luminosidade é o aparecimento de reflexos, este caso depende também da existência de uma superfície refletora, para que essa situação ocorra. Podendo ocorrer a reflexão desde partes de uma pessoa ou objeto até a reflexão completa do corpo objeto ou pessoa. A distorção gerada nos resultados, devido à ocorrência de reflexos, pode ser vista na Figura 7.1, apresentada abaixo.

Figura 7.1: Resposta incorreta do sistema, na qual ele identifica o reflexo no piso do indivíduo (de branco) cujos movimentos foram detectados, como sendo outro indivíduo se movendo.

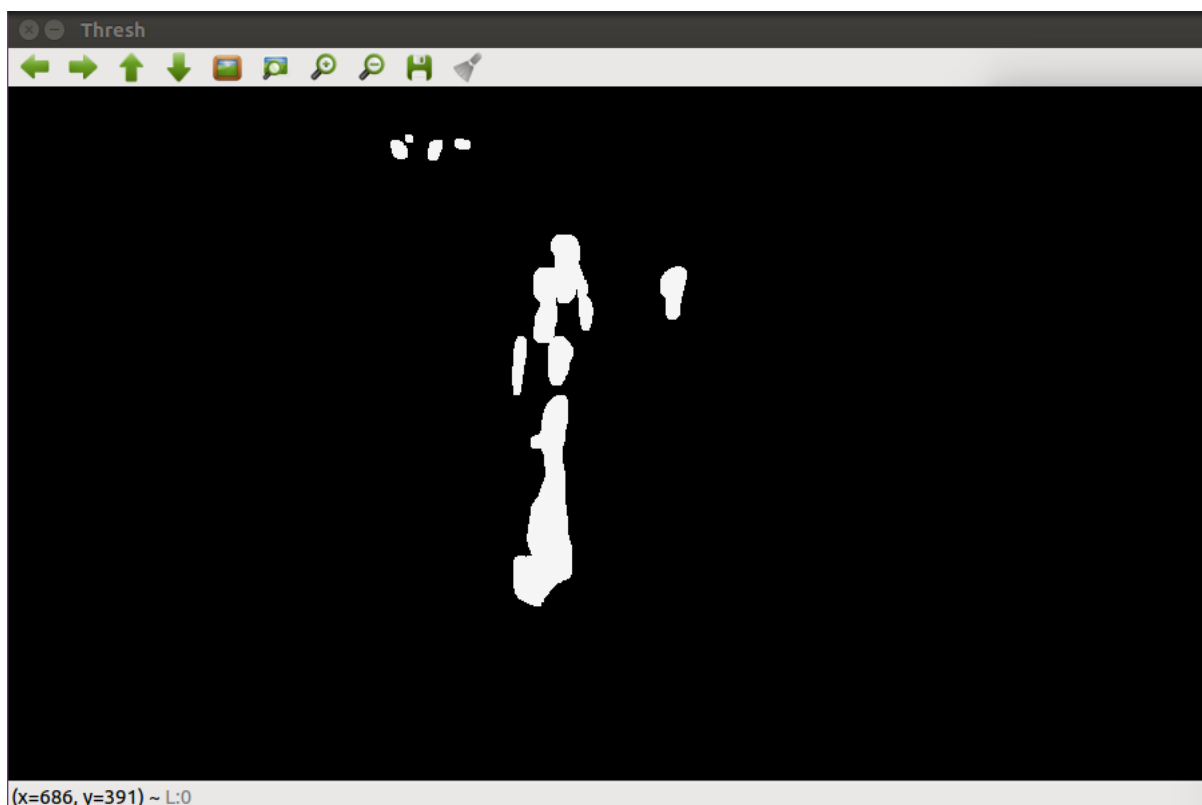


Fonte: imagem gerada pelo autor.

Para justificar a ocorrência desses erros serão utilizadas as imagens 7.2 e 7.3, que são apresentadas abaixo.

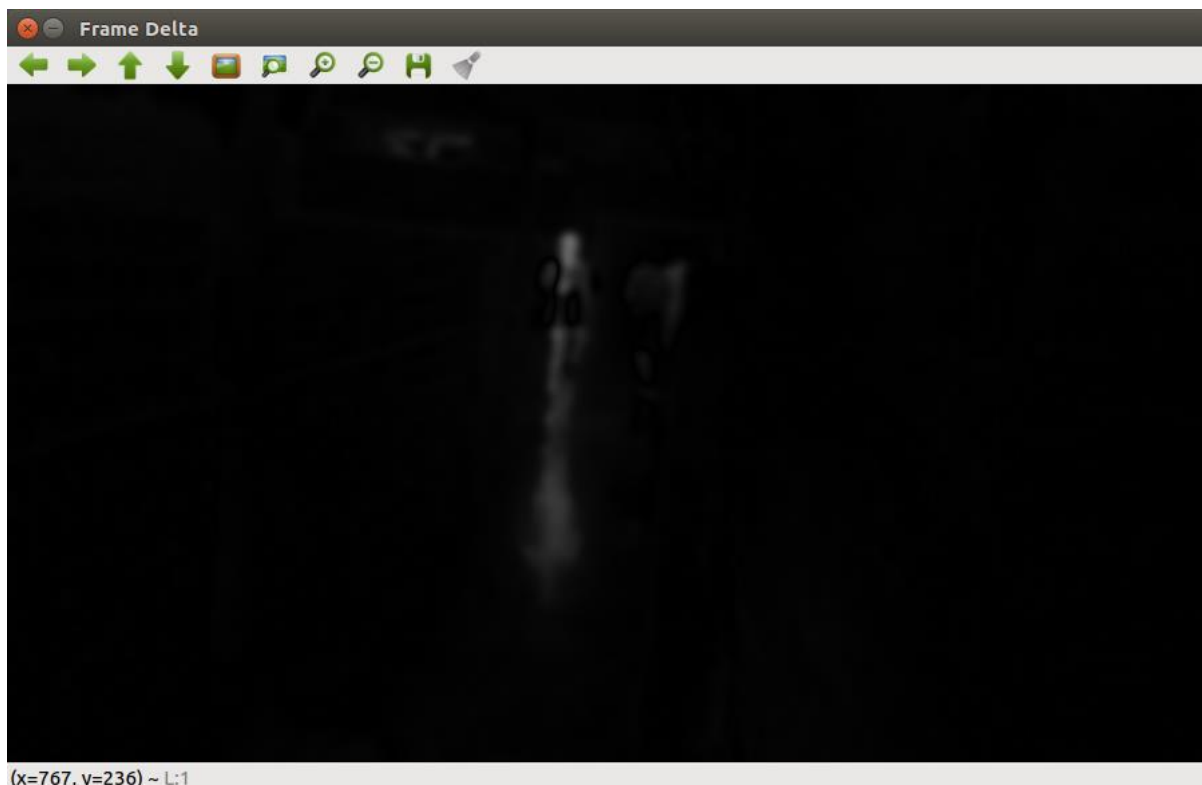
Figura 7.2: Visão após a diferença entre o frame atual e o de referência.

Demonstrando localização dos pixels em movimento, na visão do sistema.



Fonte: imagem gerada pelo autor.

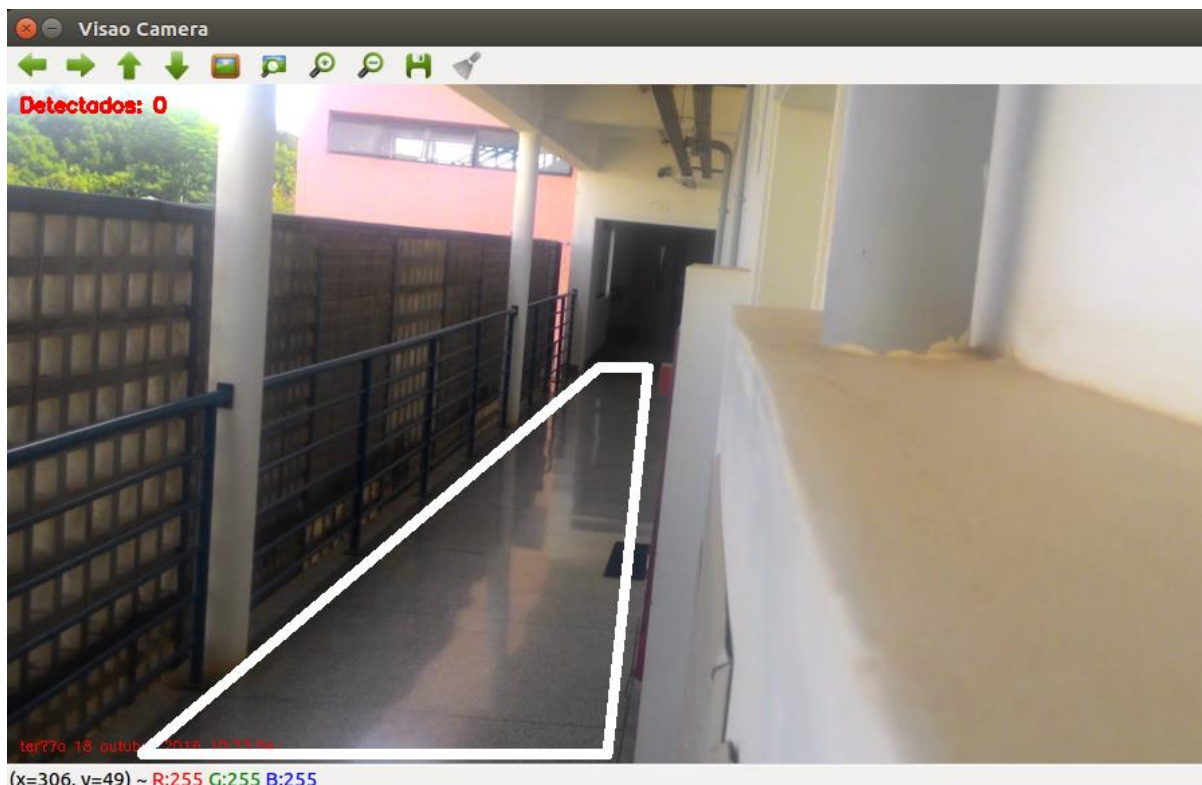
Figura 7.3: Visão em *Grayscale* (Escala de cinza) da imagem resultante após a subtração entre o frame atual e o frame de referência. ou seja, visão em *Grayscale* da Figura 7.2, apresentada acima.



Fonte: imagem gerada pelo autor.

Utilizando as figuras acima como referenciais, é possível perceber que o sistema interpreta o reflexo como se fosse outra pessoa se movendo, devido a uma confusão de projeção, ou seja o sistema é incapaz perceber que existe um chão no local e que seria impossível para alguém estar andando de ponta cabeça. Como lidar com o referencial do sistema exigiria o uso de outros métodos não abordados neste trabalho, a solução para esse caso seria calibrar o sistema de forma a ensiná-lo que existe um chão na imagem e que tudo que ocorre na região do chão deveria ser ignorado. Ou seja, o sistema passaria a ignorar a ocorrência de qualquer fenômeno que ocorresse na região do chão, o que resolveria o problema do reflexo para esse caso, a região a ser ignorada pode ser vista na Figura 7.4, apresentada na sequência.

Figura 7.4: Destaque da área (região dentro do quadrilátero branco) que deveria ser ignorada (por meio do de calibragem) pelo sistema, de modo a evitar os erros de identificação relacionados a reflexos no piso.



Fonte: imagem gerada pelo autor.

Entretanto é válido lembrar que o uso da técnica de calibragem, precisa ser feito de ambiente para ambiente, logo o uso dessa abordagem para resolver o problema, tornaria inviável o uso do mesmo algoritmo (pós-calibragem), para outro ambiente que não seja o para qual ele foi calibrado. Ou seja, para obter um melhor resultado seria necessário criar uma variação do algoritmo específica para esse ambiente, que resultaria na negligência da premissa que esse sistema deva ter um resultado satisfatório para qualquer ambiente e não apenas para um ambiente específico.

5.3 Ambiente Controlado

Enquanto nos dois itens anteriores, foram abordados os problemas mais comuns apresentados pelo sistema, nesta Seção será demonstrado os resultados obtidos em para um ambiente onde ocorre o correto posicionamento da câmera e uma baixa variação da luminosidade. Os resultados obtidos para esse ambiente de condições quase "ideais", pode ser visto abaixo, nas Figuras 8.1, 8.2 e 8.3.

Figura 8.1: Detecção de dois veículos em ambiente de condições quase ideais de luz e sombra. Além de baixas chances de aparecimento de reflexos devido ao asfalto.



Fonte: Imagem adaptada de vídeo da *BGSLibrary* (2016).

Figura 8.2: Exemplo de detecção de um veículo em ambiente de condições quase ideais de luz e sombra. Além de baixas chances de aparecimento de reflexos devido ao asfalto.



Fonte: Imagem adaptada de vídeo da *BGSLibrary* (2016).

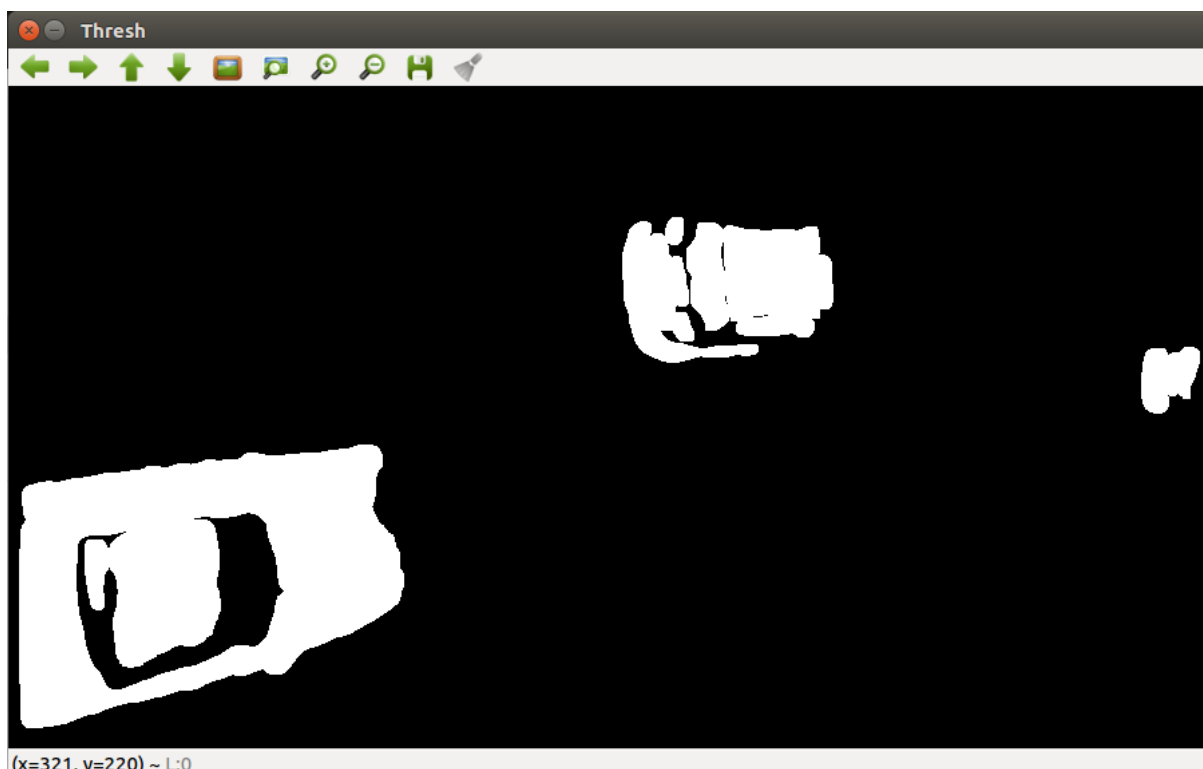
Figura 8.3: Resposta incorreta do sistema em ambiente quase ideal de luz e sombra. Erro localizado no carro central, onde existe a presença de uma sombra na sua superior.



Fonte: Imagem adaptada de vídeo da *BGSLibrary* (2016).

Como visto acima apenas a Figura 8.3 que não apresenta o resultado esperado, isso entretanto se deve ao fato de ocorrer a formação de uma pequena sombra sobre o carro naquele ponto do percurso, como pode ser observado na foto. Essa sombra faz com que o sistema assumira que existem dois objetos sobrepostos e não apenas um, e o levando a marcar 2 retângulos que se sobrepõe. Essa descontinuidade gerada pela sombra pode ser observada de modo mais preciso na Figura 8.4, apresentada abaixo.

Figura 8.4: Visão resultante da diferença entre o frame atual e o de referência, para a Figura 8.3, acima, que ilustra a distorção na resposta decorrente da segmentação no carro, devido ao aparecimento da sombra.



Fonte: imagem gerada pelo autor.

Mesmo o sistema não apresentado uma resposta igual a ideal para este caso, o resultado serve para demonstrar a capacidade do sistema de detectar dois objetos ou pessoas que estão muito próximos ou sobrepostos. Sendo estas mais uma das características necessárias para o sistema em uma situação preventiva real. Para este caso entretanto, o erro obtido demonstra que apenas o controle de iluminação e ângulo da câmera podem não ser suficientes para a obtenção de um resultado perfeito, sendo assim o uso da calibragem necessário para resolver a falha e chegar a um resultado igual ao ideal.

Capítulo 6. Conclusões

Este trabalho, como apresentado na Seção 1.1, tem como seu objetivo propor uma solução que colabore na prevenção de assaltos, sequestros relâmpagos e outros tipos de violência. Abordando os casos, nos quais estes crimes ocorrem quando as pessoas estão chegando em suas residências. O sistema proposto apresenta como alguns de seus diferenciais, a localização autônoma de pessoas e veículos e o compartilhamento de informações (imagens) entre vizinhos. Incentivando assim, a cooperação entre os mesmo com intuito de alcançarem o bem comum.

Todavia, como descrito no capítulo 2 e demonstrado no capítulo 5, este projeto apresenta alguns problemas ligados a iluminação do ambiente. Sendo estes problemas divididos em três categorias: Sombras, Reflexos e Ausência de iluminação.

As respostas dos sistema para casos com presença de reflexos e sombras, foram apresentados no capítulo 5. Baseado nestes resultados é possível perceber a necessidade do uso da técnica da calibragem, de modo a fazer com que o sistema considere apenas a região onde a probabilidade de ocorrer essas distorções (presença de reflexos e sombras) é menor. Entretanto como dito anteriormente isso aumentaria o custo do projeto e reduziria sua acessibilidade para uma parcela do público alvo.

No caso da Ausência de iluminação, este se torna um caso cuja dependência para a solução está mais ligada a câmera que ao sistema em si. Para que exista a possibilidade de abordagem nessas circunstâncias é necessário o uso de uma câmera com visão noturna, o que possibilitaria a obtenção de imagens com algum tipo de informação que não fosse simplesmente escuridão. No caso do uso da câmera adequada o sistema reagiria como descrito no capítulo 4 e se adaptaria a mudança de iluminação, resultando assim em um funcionamento correto (dentro do esperado e descrito neste trabalho).

Entretanto, uma premissa para o projeto, que foi citada no decorrer dos Capítulos 1, 3 e 5, é que o sistema seja de baixo custo, viabilizando assim o acesso a ele para uma maior faixa da população. Desse modo, tanto o objetivo quanto a premissa foram escolhidos com intuito de retribuir à sociedade com os frutos dos meus estudos, após cinco anos de curso em uma universidade pública, atacando assim um problema real enfrentado por toda a população de um modo geral. Visando cumprir esse objetivo desenvolveu-se o sistema de modo que o mesmo possa ser utilizado desde com uma simples webcam, até uma câmera ip de última geração, possibilitando seu uso por todos os grupos da população.

O curso de Engenharia de Computação do campus de São Carlos, é completo e abrange diversos temas da área da Computação e da Engenharia Elétrica, envolvendo

desde a criação e estudo de *Software* e *Hardware*, até tópicos como análises e processamento de sinais ou estudos de embarcados e robótica.

Para o desenvolvimento deste projeto em especial, as disciplinas de programação, computação gráfica, processamento digital de sinais, engenharia de *software* e programação orientada a objetos, foram fundamentais. Sem o conhecimento nestas disciplinas não seria possível superar os desafios e dificuldades encontradas durante as todas as fases do projeto, desde a sua proposição até a validação e teste do sistema de localização e a definição dos próximos passos.

6.1 Próximos passos

Como explicado na Seção 1.1, a ideia inicial abrange muito mais do que o implementado neste trabalho. Logo os próximos passos serão o desenvolvimento das demais partes, que viabilizarão o uso do sistema de modo compartilhado pelos usuários, como idealizado inicialmente.

Para que o sistema possa atingir um nível de segurança e maturidade, que possibilitem seu uso compartilhado por um determinado grupo de usuários, será necessário o desenvolvimento de uma interface *web* (*site*) e de um aplicativo *mobile*.

A interface *web* consistirá de um canal para centralização das imagens obtidas pelas câmeras espalhadas, além de servir como uma central unificada para acesso dos usuários e visualização dos resultados obtidos.

No caso do aplicativo *mobile*, será uma ferramenta para facilitar o acesso aos resultados. Além de proporcionar um acesso mais rápido, permitirá uma maior inserção do sistema na vida dos usuários, já que atualmente se cultiva a cultura de disponibilização de informações para rápido acesso via aplicativos.

Finalmente, como um passo final para o projeto seria o desenvolvimento de uma solução de computação distribuída aliada a *cloud computing* (computação em nuvem), que permitiria um gigantesco ganho de escala para o sistema além de uma significativa redução de custos. Tal redução seria devido ao compartilhamento de servidores, entre sistemas de diferentes grupos de usuários. Para garantir a privacidade e segurança dos usuários, será necessário aliar esse mecanismo a um algoritmo de encriptação, proporcionando assim um maior sigilo e proteção das informações de cada um dos grupos de usuários.

Referências Bibliográficas

1. ROSA, Edinete Maria et al. Urban violence, insecurity and fear: the need for collective strategies. **Psicologia: Ciência e Profissão**, v. 32, n. 4, p. 826-839, 2012.
 2. WORTMEYER, Charles; FREITAS, Fernando; CARDOSO, Líam. Automação Residencial: Busca de Tecnologias visando o Conforto, a Economia, a Praticidade e a Segurança do Usuário. **II Simpósio de Excelência em Gestão e Tecnologia SEGeT2005**, 2005.
 3. DEJOURS, Christophe. **Fator Humano, O**. FGV Editora, 1997.
 4. JUNIOR, Capella; GAUBERT, Alfredo. Estudo e implementação de sistema de vídeo-vigilância inteligente. 2010.
 5. SOARES, A.; FIGUEIRÓ, Thiago; SUSIN, A. A. Caracterização do desempenho de métodos de detecção de movimento aplicado a localização de pessoas através de visão computacional. **SIDEE, UFRGS**, 2004.
 6. JAVED, Omar; SHAH, Mubarak. KNIGHT M: A Multi-Camera Surveillance System. In: **ONDCP International Technology Symposium**. 2003.
 7. LUGER, George F. Inteligência Artificial-: Estruturas e estratégias para a solução de problemas complexos. Bookman, 2004.
 8. DO PROTÓTIPO, DE INTELIGÊNCIA ARTIFICIAL. Protótipo de sistema especialista para auxílio à decisão em direito ambiental: situações de desmatamentos rurais. 2009.
- MARTINS JUNIOR, P. P.; VASCONCELOS, V. V. (2008). Protótipo de sistema especialista para auxílio à decisão em direito ambiental: situações de desmatamentos rurais. Disponível em:<https://www.researchgate.net/profile/Vitor_Vasconcelos2/publications/5>. Acesso em: 04 nov. 2016.
9. VAN ROSSUM, Guido et al. Python Programming Language. In: **USENIX Annual Technical Conference**. 2007.
 10. VENNERS, Bill. The making of Python. Artima. com. <http://www.artima.com/intv/python.html> [accessed 2003-12-09], 2003.
 11. MARENGONI, Maurício; STRINGHINI, Stringhini. Tutorial: Introdução à visão computacional usando opencv. **Revista de Informática Teórica e Aplicada**, v. 16, n. 1, p. 125-160, 2009.
 12. BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV: Computer vision with the OpenCV library**. " O'Reilly Media, Inc.", 2008.

13. OSÓRIO, Fernando S.; BITTENCOURT, João Ricardo; OSÓRIO, Fernando Santos. Sistemas Inteligentes baseados em redes neurais artificiais aplicados ao processamento de imagens. In: **I WORKSHOP DE INTELIGÊNCIA ARTIFICIAL UNISC–Universidade de Santa Cruz do Sul Departamento de Informática-Junho**. 2000.
14. PICCARDI, Massimo. Background subtraction techniques: a review. In: **Systems, man and cybernetics, 2004 IEEE international conference on**. IEEE, 2004. p. 3099-3104
15. FARINES, Jean-Marie; FRAGA, Joni da Silva; OLIVEIRA, RS de. Sistemas de tempo real. **Escola de Computação**, v. 2000, p. 201, 2000.
16. SILVA, Ricardo Petri. PROCESSAMENTO DE IMAGENS PARA RESOLUÇÃO DE PROBLEMAS DE REFLEXOS E SOMBRAS.
17. RUAS, Gabriel IS et al. Rastreamento de múltiplos veículos em vias urbanas por processamento de vídeo. **XXVI Simpósio Brasileiro de Telecomunicações, Rio de Janeiro**.

Anexos

Código¹² do Sistema de vigilância baseado em localização de veículos e pessoas

```
# Bibliotecas necessarias
import argparse
import datetime
import imutils
import time
import cv2

# Construcao do argumento de entrada
# Para selecionar um video para leitura, de nome "video.avi", usaremos o seguinte argumento:
#python findback.py -v video.avi
#Sendo 'findback.py' o nome deste arquivo.
ap = argparse.ArgumentParser()
ap.add_argument("-v", "--video", help="path to the video file")
ap.add_argument("-a", "--min-area", type=int, default=900, help="minimum area size")
args = vars(ap.parse_args())

#No caso de nao haver argumento, automaticamente o programa capturara as imagens da webcam
if args.get("video", None) is None:
    camera = cv2.VideoCapture(0)
    time.sleep(0.25)

# No caso de haver um argumento, essa linha faz a leitura do video passado no argumento
else:
    camera = cv2.VideoCapture(args["video"])

#inicializa o primeiro frame do video
firstFrame = None
```

¹² Código adaptado de *Pyimagesearch, Basic motion detection and tracking with Python and OpenCV* (Adrian Rosebrock, 2015). Disponível em: <http://www.pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/>, acessado em 03 de novembro de 2016.

```

#loop de leitura de frames
while True:
    (grabbed, frame) = camera.read()
    #inicializacao de variavel para escrever na tela
    text = 0

    #No caso de existir mais frames, se atingiu o fim do video
    #if abaixo serve para reiniciar o loop do video.
    if not grabbed:
        camera = cv2.VideoCapture(args["video"])#para dar loop no video
        (grabbed, frame) = camera.read()#para dar loopno video
        #break

    # redimensiona o frame, converte para grayscale (escala de cinza) e o desfoca
    frame = imutils.resize(frame, width=900)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray, (21, 21), 0)

    # Se o primeiro frame for nulo, ele sera inicializado
    if firstFrame is None:
        firstFrame = gray
        continue

    # Computa a diferenca entre o frame de referencia e o atual
    frameDelta = cv2.absdiff(firstFrame, gray)
    thresh = cv2.threshold(frameDelta, 25, 255, cv2.THRESH_BINARY)[1]

    # Identifica as diferencas entre os frames e os preenche para localizar contornos
    thresh = cv2.dilate(thresh, None, iterations=2)
    (_, cnts, _) = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)

    # loop sobre os contornos encontrados
    for c in cnts:

```

```

# ignora conrnos muito pequenos
if cv2.contourArea(c) < args["min_area"]:
    continue

# Localiza os cotornos com tamanho significativo, desenha os retangulos
# no frame e atualiza o texto do contador
(x, y, w, h) = cv2.boundingRect(c)
cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 255, 255), 2)
text = text+1

# Escreve o texto do contador e do timestamp no frame
cv2.putText(frame, "Detectados: {}".format(str(text)), (10, 20),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)
cv2.putText(frame, datetime.datetime.now().strftime("%A %d %B %Y %l:%M:%S%p"),
            (10, frame.shape[0] - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.35, (0, 0, 255), 1)

# Apresenta na tela o frame final, com os objetos identificados e com os textos
# do contador e do time stamp
cv2.imshow("Visao Camera", frame)
#apresenta em outra janela, o resultado da subtracaoe entre o frame de referencia e o atual
cv2.imshow("Thresh", thresh)
# apresenta em outra janela a imagem desfocada em grayscale
cv2.imshow("Frame Delta", frameDelta)

# Detecta se o usuario pressionou alguma tecla
key = cv2.waitKey(1) & 0xFF

# Se a tecla pressiona for 'q', entao finalizara o loop
if key == ord("q"):
    break

# Libera a camera, limpa o buffer e fecha todas as janelas abertas.
camera.release()
cv2.destroyAllWindows()

```