



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Vinicius Ferri Pereira

**ABORDAGEM COMPUTACIONAL PARA GERENCIAMENTO DE LOTAÇÃO EM
AMBIENTES VÍDEO-MONITORADOS POR MEIO DE VISÃO COMPUTACIONAL E
SISTEMA DE INFORMAÇÕES**

Araranguá,
2022

Vinicius Ferri Pereira

**ABORDAGEM COMPUTACIONAL PARA GERENCIAMENTO DE LOTAÇÃO EM
AMBIENTES VÍDEO-MONITORADOS POR MEIO DE VISÃO COMPUTACIONAL E
SISTEMA DE INFORMAÇÕES**

Trabalho de Conclusão do Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Antonio Carlos Sobieranski, Dr.

Araranguá,
2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Pereira, Vinicius Ferri

Abordagem computacional para gerenciamento de lotação em ambientes video-monitorados por meio de visão computacional e sistema de informações / Vinicius Ferri Pereira ; orientador, Antonio Carlos Sobieranski, 2022.

31 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Araranguá,
Graduação em Engenharia de Computação, Araranguá, 2022.

Inclui referências.

1. Engenharia de Computação. 2. Engenharia de Computação.
3. Visão Computacional. 4. Contagem de pessoas. 5. Fluxo óptico. I. Sobieranski, Antonio Carlos. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Computação. III. Título.

Vinicius Ferri Pereira

**ABORDAGEM COMPUTACIONAL PARA GERENCIAMENTO DE LOTAÇÃO EM
AMBIENTES VÍDEO-MONITORADOS POR MEIO DE VISÃO COMPUTACIONAL E
SISTEMA DE INFORMAÇÕES**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação, e foi aprovado em sua forma final pelo Curso de Engenharia de Computação.

Araranguá, 01 de agosto de 2022.

Profa. Analucia Schiaffino Morales, Dr.
Coordenadora do Curso

Banca Examinadora:

Prof. Antônio Carlos Sobieranski, Dr.
Orientador
Universidade Federal de Santa Catarina

Profa. Eliane Pozzebon, Dra.
Avaliador
Universidade Federal de Santa Catarina

Prof. Anderson Luiz Fernandes Perez, Dr.
Avaliador
Universidade Federal de Santa Catarina

Prof. Martin Augusto Gagliotti Vigil, Dr.
Avaliador Suplente
Universidade Federal de Santa Catarina

Abordagem Computacional para Gerenciamento de Lotação em Ambientes Vídeo-monitorados por Meio de Visão Computacional e Sistema de Informações

A Computational Approach to Crowd Managing in Video-monitored Environments Using Computer Vision and Information Systems

Vinicius Ferri Pereira *

Antônio Carlos Sobieranski †

2022, Agosto

Resumo

O controle do fluxo de pessoas tem sido um recurso fundamental em diversas áreas como comércio, indústrias, educação e sistemas de saúde e pode ser usado para os mais variados fins, funcionando como controle de distanciamento social, ferramenta de gestão para tomar decisões estratégicas em horário de pico na área comercial e sendo até mesmo um ótimo recurso para monitorar e evitar a superlotação em ambientes fechados. Abordagens como a da visão computacional para controle do fluxo de pessoas em ambientes vídeo-monitorados têm se apresentado eficazes para contabilizar o fluxo de pessoas nos setores citados. Dessa forma, este estudo propõe um sistema utilizando cameras de luz visível na posição zenital, que passa por um processo de filtragem gaussiana, detecção de movimento por fluxo óptico utilizando o algoritmo de Lucas Kanade, detecção de centros através das bordas, rastreamento de objetos em ambientes vídeo-monitorados e envio de dados para um sistema de informação construído com Flask. Na análise final, o sistema foi capaz de contabilizar todas as pessoas detectadas nas imagens, resultando numa assertividade de 100% no estudo indicado.

Palavras-chaves: Visão computacional. Contagem de pessoas. Fluxo óptico.

*vinicius.fp@grad.ufsc.br

†a.sobieranski@ufsc.br

Abordagem Computacional para Gerenciamento de Lotação em Ambientes Vídeo-monitorados por Meio de Visão Computacional e Sistema de Informações

A Computational Approach to Crowd Managing in Video-monitored Environments Using Computer Vision and Information Systems

Vinicius Ferri Pereira * Antônio Carlos Sobieranski †

2022, Agosto

Abstract

People flow control has been a fundamental resource in several areas such as commerce, industry, education, and health systems, and can be used for a variety of purposes, such as social distance control, management tool to make strategic decisions during peak hours in the commercial area, and even being a great resource to monitor and avoid overcrowding in closed environments. Approaches such as computer vision for controlling the flow of people in video-monitored environments have proven to be effective in accounting for the flow of people in the aforementioned sectors. Thus, this study proposes a system using visible light cameras at the zenith position, which goes through a Gaussian filtering process, optical flow motion detection using Lucas Kanade's algorithm, center detection through edges, object tracking in video-monitored environments, and sending data to an information system built with Flask. In the final analysis, the system was able to count all the people detected in the images, resulting in an assertiveness of 100% in the indicated study.

Key-words: Computer vision. Counting people. Optical flow.

*vinicius.fp@grad.ufsc.br

†a.sobieranski@ufsc.br

1 Introdução

O controle de fluxo de pessoas tem sido essencial no cotidiano de estabelecimentos comerciais, hospitais e espaços escolares e acadêmicos. Dentre as aplicações possíveis para controlar a capacidade máxima de pessoas de um local, tem-se a gestão de recursos de acordo com a quantidade de pessoas por horário, a fim de estabelecer medidas de segurança em locais de acesso restrito ou para cumprir com exigências sanitárias de distanciamento social, como foi necessário durante a pandemia do novo coronavírus (COVID-19) (AQUINO, 2020).

O controle de fluxo é fundamental para evitar problemas como a superlotação de ambientes, que pode vir a causar acidentes como o ocorrido em 2013 na Boate Kiss, em Santa Maria, ou para prevenir o contágio de doenças transmitidas por meio de partículas de ar. Dessa forma, esses sistemas de controle podem monitorar o número de indivíduos que entram em um estabelecimento no mesmo período de forma a evitar casos de superlotação.

No viés corporativo, o controle de fluxo pode funcionar como uma ferramenta de gestão para que o dono do estabelecimento conheça melhor a necessidade do público, sabendo quais são os horários de pico para poder tomar decisões estratégicas para o funcionamento e desempenho do negócio. Em julho de 2020, a Organização Mundial da Saúde (OMS) apontou evidências de que o coronavírus pode se espalhar por minúsculas partículas suspensas no ar e que a transmissão aérea não podia ser descartada em ambientes lotados, fechados ou mal ventilados. Portanto, o controle do fluxo de pessoas pode ser útil para garantir a proporção em lotação de m^2 por pessoas, a fim de possibilitar o distanciamento social recomendado pelo órgão.

Soluções computacionais e industriais para o problema da contagem geralmente são baseadas na aquisição utilizando sensores mecânicos, imageamento por luz visível ou pelo uso de infravermelho. Sensores mecânicos, a exemplo das catracas, podem contar apenas uma pessoa por vez e acabar ocasionando um atraso no fluxo, gerando filas no ambiente (PÁDUA, 2002). Sensores de imageamento por luz visível, por sua vez, podem utilizar câmeras convencionais ou de segurança já existentes no ambiente para realizar o monitoramento e fazer a contagem de pessoas, porém, podem apresentar certa dificuldade em ambientes com variação de luminosidade (PÁDUA *et al.*, 2003). Tecnologias por infravermelho ou térmicas apresentam a vantagem de operação invariante à iluminação, embora apresentem um custo maior quando comparadas às câmeras de imagem por luz visível e podem não ser eficientes em identificar indivíduos separadamente em um grupo (MELO, 2011).

Com o avanço computacional obtido nas últimas décadas, principalmente as áreas baseadas em processamento digital de imagens, reconhecimento de padrões e visão computacional tiveram um incremento significativo na expressividade das suas soluções (KAMIDA, 2017). Mais recentemente, com o ressurgimento da área de inteligência artificial através dos modelos convolutivos, a solução geral para uma série de problemas foi encontrada, principalmente quando comparados aos modelos clássicos de visão computacional (RAMO, 2017). Na literatura, podem ser encontrados trabalhos utilizando a contagem de pessoas em ambientes utilizando tanto algoritmos clássicos como baseados em inteligência artificial (GONÇALVES, 2005). No entanto, os trabalhos apresentados são pouco escaláveis para situações diversas, a diversos sistemas e pontos de aquisição, e não apresentam sistema gestor demonstrando a lotação em tempo real dos ambientes (KAMIDA, 2017).

Neste trabalho, é apresentada uma abordagem de visão computacional para o

controle de fluxo e lotação em ambientes vídeo-monitorados. Tendo em vista o baixo custo e a escalabilidade do sistema, bem como as limitações do ambiente em relação à luminosidade, foi utilizada uma abordagem de algoritmo para fluxo óptico para a estimativa de direção e para a extração das características que foram utilizadas para fazer o rastreamento das pessoas e realizar a contagem. A seguir, os dados obtidos são enviados para um sistema de informações onde podem ser visualizados pelo administrador do estabelecimento. Foram realizados dois testes monitorando o fluxo de pessoas em dois ambientes distintos. Resultados experimentais demonstraram para o ambiente de testes um grau de assertividade de 100%, demonstrando a viabilidade da abordagem proposta na contagem e controle de lotação em ambientes.

O restante deste trabalho está organizado da seguinte forma: na Seção 2, são apresentados trabalhos correlatos pertinentes à pesquisa realizada na literatura sobre tecnologias de monitoramento e contagem de pessoas. Na Seção 3, são apresentadas as principais definições teóricas utilizadas no trabalho. A Seção 4 apresenta a abordagem proposta para resolução do problema tratado. Resultados experimentais obtidos nos ambientes de vídeo monitorados, assim como as taxas de precisão, são apresentados na seção Seção 5. Conclusões, discussões e direcionamentos futuros em relação ao trabalho realizado serão apresentados na Seção 6.

2 Trabalhos correlatos

A presente seção detalha a metodologia de pesquisa utilizada para a revisão da literatura acerca do tema contagem e rastreamento de pessoas. Foram buscados trabalhos publicados em eventos científicos e periódicos da área de engenharia e ciência da computação durante o período de 2009 a 2022. Materiais de referência também foram retirados dos sítios acadêmicos IEEE, Science Direct e ACM. As seguintes keywords foram utilizadas: counting people, tracking people and optical flow. Inicialmente, um total de 16 trabalhos foram analisados, no entanto, após a etapa de filtragem inicial, com a leitura do abstract/resumo, apenas 8 trabalhos foram selecionados para compor o corpus desta pesquisa.

Os trabalhos verificados são apresentados a seguir.

1. O primeiro trabalho analisado trata da contagem de pessoas utilizando imagens visíveis e infravermelhas. A pesquisa é apresentada por Filipic *et al.* (2021). No trabalho em questão, foram capturadas imagens por uma câmera de 3 canais (RGB) e de 4 canais (RGB e infravermelho) através de um drone. Nesse trabalho, redes neurais convolucionais são utilizadas para estimar o número de pessoas em ambientes abertos. As imagens captadas pelo drone foram modificadas pelos autores para incluir mais pessoas na cena. Os testes foram realizados nessas imagens semi-artificiais tiradas em diferentes distâncias e condições de luminosidade. A pesquisa comparou os resultados obtidos entre os tipos de imagem com e sem a informação do infravermelho, e, em todos os casos apresentados, o número de pessoas contadas foi mais preciso ao utilizar o infravermelho como informação adicional.
2. Trabalho baseado no algoritmo de Viola-Jones é apresentado em “People Counting at Campuses” por Cetinkaya e Akcay (2015). Nesse trabalho, câmeras são instaladas em diferentes locais, como em um corredor, na entrada de uma sala e acima de um quadro de avisos, com o objetivo de contar o número de pessoas. O algoritmo utilizado detecta faces humanas em uma sequência de frames. A assertividade do sistema foi

medida pelo cálculo do número de pessoas detectadas pelo sistema, dividido pelo real número de pessoas e multiplicado por 100 para obter o valor em porcentagem. Os testes mostraram que, para cada câmera, diferentes resultados foram obtidos, e que, na maioria das vezes, os erros aconteceram pela contagem da mesma pessoa duas vezes ou por não conseguir capturar o rosto humano. O melhor resultado obtido foi de 100% de eficácia na entrada de um laboratório de computação, enquanto o pior foi registrado em um dos quadros no corredor, obtendo 50% de acertos, totalizando uma média de 83,74% de taxa de acertos.

3. Trabalho de contagem de pessoas com rede neural Faster R-CNN e correlações de rastreamento é apresentado por Li *et al.* (2016). Nesse trabalho, é utilizada uma rede neural que detecta ombros e cabeça junto ao modelo Zeiler e o algoritmo OHEM para detectar esses padrões em diferentes perspectivas. O algoritmo de rastreamento KCF é aplicado para associar as pessoas em diferentes frames. Os testes foram aplicados em dois datasets, Vertical e Oblique, e comparados a outros estudos que utilizam as técnicas utilizadas no trabalho separadamente, mostrando que a combinação dos algoritmos Faster+OHEM+KCF proporcionou o menor índice de erro absoluto médio e menor número de erro quadrático médio dentre as comparações.
4. Pore e Momin (2016) apresentam um sistema de contagem de pessoas utilizando câmeras de segurança. Para a detecção de pessoas, é utilizado o Histograma de Gradientes Orientados (HOG) com uma janela de detecção de 4x4 pixels. A partir desse descritor, são obtidas algumas características da cena que são passadas para uma Máquina de Vetores de Suporte (SVM), que classifica objetos em humanos ou não humanos. Os autores consideram o movimento das pessoas como linear, para aplicar o filtro de Kalman para contagem e rastreamento das pessoas. Uma linha de referência no centro da imagem é utilizada para definir a direção das pessoas e realizar a contagem de entradas e saídas. O sistema apresentou uma assertividade de 91% a 100% dependendo das oclusões e variações no vídeo.
5. O trabalho de Antić *et al.* (2009) apresenta um sistema para contagem e rastreamento de pessoas através de uma câmera em posição zenital. Nesse trabalho, foi utilizado um algoritmo de subtração de background em bloco, seguido de um agrupamento pelo algoritmo K-means. Duas linhas são utilizadas como referência para a contagem das pessoas entrando e saindo da cena. O método foi aplicado em dois lugares diferentes e mostrou uma precisão de quase 95%. Os autores relatam que, apesar de algoritmos simples terem sido usados, foi possível obter uma aplicação rodando em tempo real e uma boa assertividade.
6. Myint e Sein (2021) apresentam mais um sistema de detecção e contagem de pessoas. O sistema utiliza um algoritmo de subtração de background para extrair o objeto em primeiro plano. As informações desse objeto são extraídas pelo algoritmo HOG e também por uma rede neural convolucional (CNN) pré-treinada chamada VGG-16. Os resultados dos dois extratores são passados para um classificador SVM, que separa a imagem do dataset em dois grupos, com o primeiro contendo uma pessoa e o segundo sem indivíduos detectados na imagem. A captura das imagens e a detecção após o treinamento da rede são feitas por uma câmera acoplada em um Raspberry Pi 3 fixado em posição zenital a 2,5 metros do chão para evitar oclusões. Por último, os dados de contagem são enviados a um servidor web. O trabalho mostra que os dois extratores acabam obtendo a mesma precisão de 94% para o dataset elaborado pelos autores.

7. Wang e Yang (2018) apresentam um trabalho sobre a detecção de alvos em movimento e rastreamento baseado no fluxo óptico piramidal de Lucas-Kanade. Nesse trabalho, um carro em movimento é utilizado como objeto de estudo. Sobre ele, é aplicado o algoritmo de Harris para detecção de bordas. Visando melhorar a performance do rastreamento, os autores aplicaram o método de rastreamento por sub-pixel e, sobre as informações obtidas, é aplicado o algoritmo de fluxo óptico para obter informações de movimento pela correlação entre sequências de imagens. Os autores concluíram o estudo mostrando que foi possível detectar o objeto em alta velocidade com as informações de borda por sub-pixel e com o método piramidal de Lucas-Kanade. Porém, o algoritmo de sub-pixel e o método piramidal utilizados apresentaram uma grande quantidade de recursos computacionais.
8. Trabalho para contagem de pessoas baseado na contagem de cabeças utilizando Adaboost e CNN em ambientes lotados através de câmeras de segurança é apresentado por Gao *et al.* (2016). Nesse trabalho, é proposto um framework em 3 etapas para a realização da contagem. A primeira etapa consiste na aplicação do algoritmo de Adaboost para aprender a detectar cabeças na imagem através do recurso HOG. Na segunda etapa, uma rede CNN é treinada utilizando as imagens obtidas na primeira etapa. Com os resultados da rede, um classificador SVM é treinado para classificar cabeças nas imagens. É realizado, então, um pós-processamento e, por fim, o número de cabeças é contado. Os autores relatam que a precisão varia dependendo da quantidade de pessoas em cena, tendo uma performance menor em ambientes lotados devido a efeitos de oclusão. Em média, para o dataset criado, os autores obtiveram uma precisão média de 80%.

Tabela 1 – Informações acerca de trabalhos correlatos

Trabalho	Sensor utilizado	Algoritmo de detecção	% precisão
1	Luz e infravermelho	CNN	
2	Câmera	Viola-Jones	83,74%
3	Câmera	Faster CNN+Zeiler +OHEM	
4	Câmeras de segurança	HOG + SVM + Filtro Kalman	91% - 100%
5	Câmera	K-means	95%
6	Câmera Raspberry	HOG + CNN VGG-16+SVM	94%
7	Câmera	Harris + rastreamento por sub-pixel + Piramidal Lucas-Kanade	
8	Câmera	HOG + Adaboost + CNN + SVM	80%

Fonte: Elaborado pelo autor

É possível observar na Tabela 1 que a maioria dos trabalhos apresenta uma solução utilizando o imageamento por luz visível, foi encontrado também um trabalho que utilizou a tecnologia de infravermelho. Dos oito trabalhos selecionados, cinco utilizam algum tipo de algoritmo relacionado a inteligência artificial e os outros três algoritmos de técnicas clássicas. A grande maioria dos autores utilizou como métrica de desempenho a precisão. Na tabela alguns valores de precisão não foram preenchidos pois nesses casos o valor não foi informado ou o autor adotou outra métrica de desempenho. Notou-se também que a precisão mínima foi de 80%.

3 Fundamentação teórica

Neste capítulo, serão abordados conceitos essenciais para o entendimento desta pesquisa, de forma que os termos são fundamentados por estudiosos da área como Szeliski, Forsyth, Ponce, Bradsk, Gonzalez e Woods, entre outros.

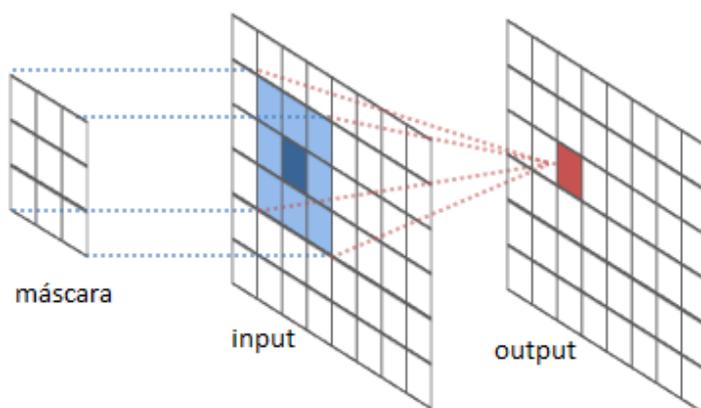
3.1 Processamento digital de imagens e visão computacional

Segundo Szeliski (2010), humanos possuem facilidade em perceber estruturas tridimensionais, sabendo diferenciar cores, formatos, luminosidade e padrões que compõem uma cena. A visão computacional nasce através da tentativa de representar essas características da visão humana através de algoritmos computacionais. Sendo assim, Bekhit (2022) define visão computacional como o campo de estudo responsável por processar, analisar e interpretar imagens digitais. Algoritmos de visão computacional já vem sendo usados em diversos casos, como no reconhecimento de caracteres, em robôs industriais, no processamento de exames médicos, vigilância, rastreamento de objetos e outros.

3.1.1 Filtragem

O processo de filtragem é uma etapa extremamente útil que permite suavizar e remover ruídos, encontrar bordas e outros padrões em imagens (FORSYTH; PONCE, 2002) Filtros no domínio espacial operam diretamente sobre os pixels da imagem. A aplicação do filtro acontece pela operação de convolução da imagem com um padrão de coeficientes de tamanho $n \times n$ também conhecido como máscara. A Figura 1 mostra como funciona a operação de convolução para uma máscara de tamanho 3×3 . Os coeficientes da máscara são multiplicados pela intensidade dos pixels da imagem de entrada e depois são somados, o novo valor será atribuído ao pixel central na imagem de saída.

Figura 1 – Demonstração da operação de convolução.



Fonte: RiverTrail

Segundo Forsyth e Ponce (2002), os pixels possuem uma característica onde cada pixel é geralmente similar aos outros ao seu redor. Desse modo, ao aplicar um filtro em uma imagem que possua ruído, os pixels que não possuem uma vizinhança bem definida

serão substituídos. Esse processo de redução do ruído é conhecido como suavização ou desfoque e é ilustrado na Figura 2.

Figura 2 – Alguns tipos de filtragem em imagem com ruído.



Fonte: Elaborado pelo autor

3.1.2 Detecção de bordas

De acordo com Bekhit (2022) a detecção de bordas é o processo de localização dos contornos dos objetos na imagem. As bordas carregam importantes informações sobre o objeto, e podem ser definidas como o limite entre objetos sobrepostos. São definidas por variações abruptas dos níveis de cinza que ocorrem em curvas acentuadas nos valores do gradiente da imagem. (SZELISKI, 2010)

3.1.2.1 Operador de Roberts

O operador de Roberts é um dos mais antigos baseado em gradiente. Segundo (GONZALEZ; WOODS, 2008) este operador utiliza a diferenciação discreta para realizar a soma dos quadrados das diferenças entre pixels nas direções x e y através das máscaras de tamanho 2x2 apresentadas na Figura 3.

Figura 3 – Máscaras de Roberts

-1	0	0	-1
0	1	1	0

Roberts

Fonte: Gonzalez e Woods (2008)

Devido ao seu tamanho de máscara é um algoritmo de execução rápida e fácil implementação porém é muito sensível a ruídos.

3.1.2.2 Operador de Sobel

O operador de Sobel também é um operador de operador de diferenciação discreta. Porém diferente do operador de Roberts, Sobel calcula o gradiente da intensidade através das duas máscaras de tamanho 3x3 apresentadas na Figura 4, que são convolucionadas com a imagem de entrada para obter estes gradientes verticais e horizontais das bordas. (GONZALEZ; WOODS, 2008)

Figura 4 – Máscaras de Sobel

0	1	2	-2	-1	0
-1	0	1	-1	0	1
-2	-1	0	0	1	2

Sobel

Fonte: Gonzalez e Woods (2008)

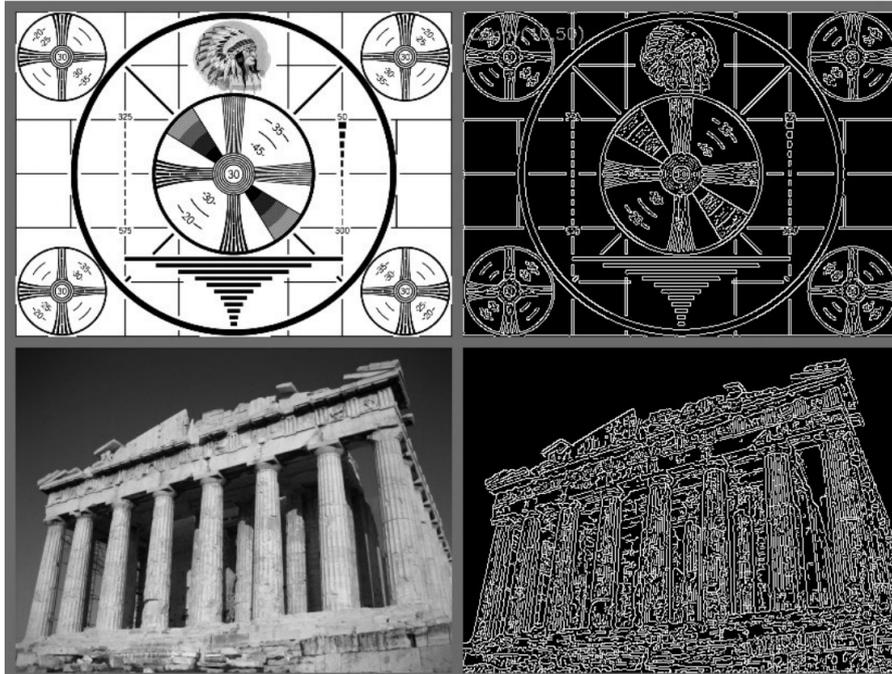
3.1.2.3 Operador de Canny

De acordo com Gonzalez e Woods (2008) o operador de Canny é baseado em uma aproximação da primeira derivada de uma função Gaussiana. Para encontrar uma função ótima afim de detectar as bordas, Canny definiu três critérios:

1. Baixa taxa de erro: Todas as bordas devem ser encontradas na imagem.
2. Pontos de borda bem localizados: As bordas encontradas pelo algoritmo devem corresponder de forma precisa as bordas reais.
3. Pontos de borda únicos: Não deve detectar mais de uma borda em regiões que só possuem uma borda.

A essência do algoritmo é achar funções matemáticas que sejam ótimas para estes três critérios. Porém por ser quase impossível, é feita a aproximação para para a primeira derivada de uma Gaussiana. A aplicação do algoritmo de Canny é mostrada na Figura 5

Figura 5 – Resultados do algoritmo de detecção de bordas utilizando Canny.



Fonte: Bradski e Kaehler (2008)

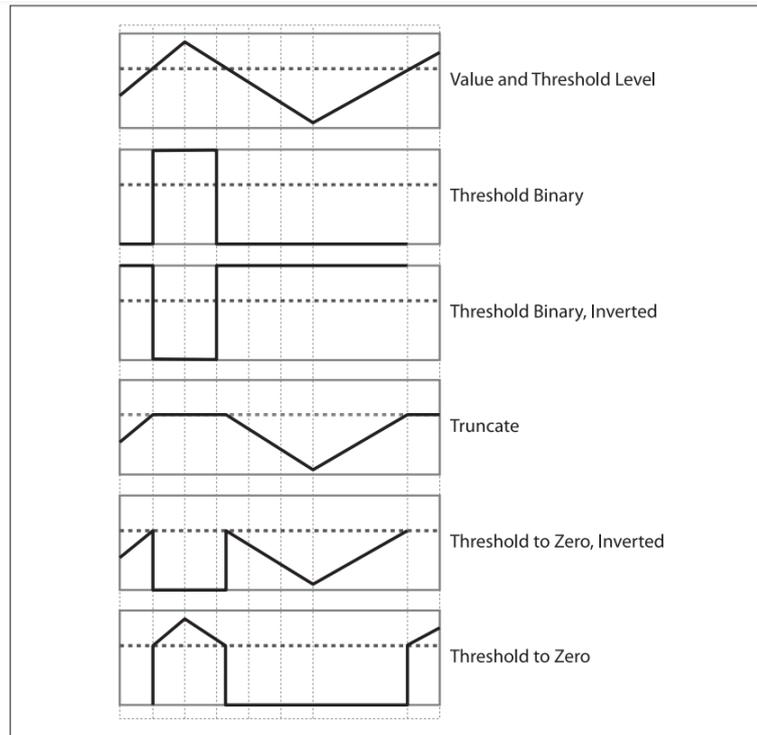
3.1.3 Segmentação de imagens

A segmentação de imagens é o processo responsável por dividir a imagem em regiões e objetos de interesse. A subdivisão está intimamente ligada ao problema a ser resolvido, devendo ser finalizada quando as regiões de interesse da imagem estiverem sendo detectadas (GONZALEZ; WOODS, 2008).

A limiarização é um processo de segmentação baseado na similaridade de níveis de cinza de cada pixel. Após realizar alguns tipos de processamento na imagem, é possível tomar uma decisão baseada em cada pixel para manter ou descartá-lo de acordo com um limiar T estabelecido (BRADSKI; KAEHLER, 2008).

Alguns tipos de limiarização são mostrados na Figura 6, em que a limiarização é representada pela linha tracejada.

Figura 6 – Tipos de limiarização.



Fonte: Bradski e Kaehler (2008)

A limiarização binária, por exemplo, assume que, quando o valor do pixel é maior que o limiar T , o pixel assume o valor lógico 1 e, quando menor que T , passa a ser 0. Assim, uma imagem limiarizada $g(x, y)$ é definida pela Equação 1.

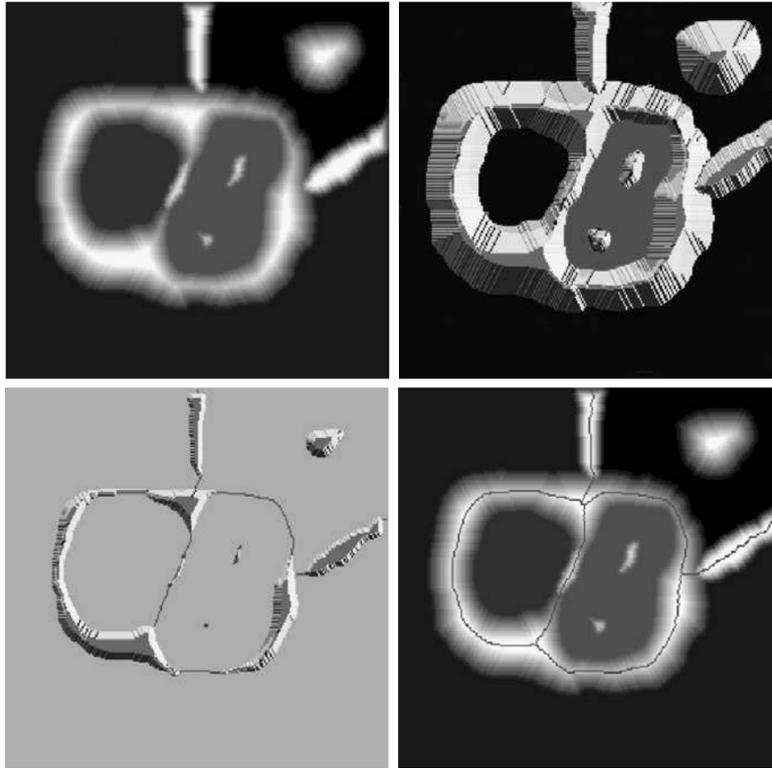
$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases} \quad (1)$$

Dessa forma, os pixels com valor igual a 1 fazem parte do objeto de interesse, enquanto os que possuem valor 0 representam o background da imagem. Em situações em que não é possível separar o background da imagem do objeto de interesse pelos métodos de limiarização, pode-se utilizar de técnicas como o algoritmo de watershed.

É possível visualizar o watershed como um problema topográfico, em que uma imagem corresponde a uma superfície e o nível de cinza em cada pixel a uma altitude. A imagem é “inundada” e, quando duas bacias hidrográficas estão prestes a se juntarem, uma barragem é construída. Ao final, tem-se apenas os topos das barragens visíveis acima da linha da água, essas linhas serão as linhas de segmentação obtidas pelo algoritmo de watershed (GONZALEZ; WOODS, 2008).

A Figura 7 ilustra o processo. Em (a), é visualizada a figura original, enquanto em (b) se observa o começo do processo de inundação, em (c), notam-se as barragens separando as bacias e é possível perceber as linhas de segmentação obtidas pelo algoritmo em (d).

Figura 7 – Processo de segmentação por watershed.



Fonte: Gonzalez e Woods (2008)

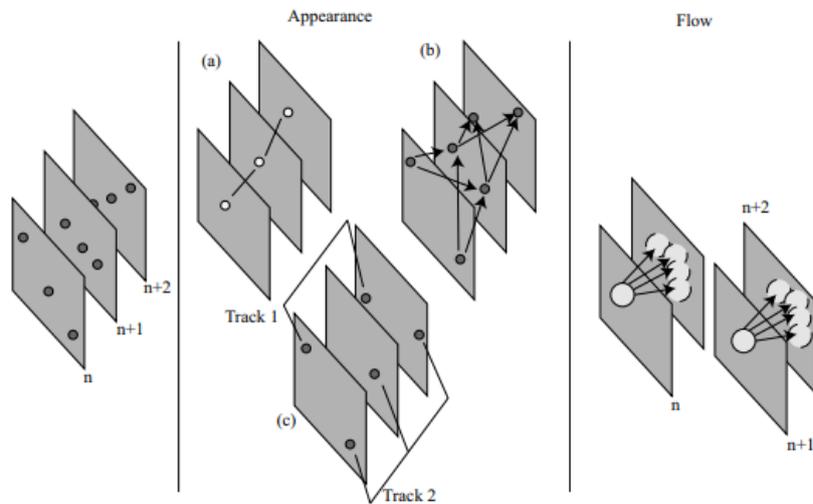
3.1.4 Detecção de movimento

De acordo com Forsyth e Ponce (2002), o rastreamento é a realização de uma inferência do movimento de um objeto em uma sequência de imagens. Há duas maneiras de realizar o rastreamento de objetos, sendo elas pela detecção, em que o modelo do objeto de interesse é bem reconhecido em cada frame, e o rastreamento de correspondência, em que existe um modelo de como o objeto se movimenta.

A detecção pelo movimento assume que apenas um objeto de interesse apareça em cada frame do vídeo, que sua posição seja conhecida e que o detector de objeto seja confiável. Assim, ao analisar as próximas imagens do vídeo, é possível encontrar em cada uma delas a posição do objeto analisado pela sua aparência. Em casos em que há mais de um objeto na mesma cena ou objetos faltantes, uma solução é implementar uma função de custo que calcule a melhor correspondência na próxima imagem para os objetos.

Outro recurso que é possível utilizar é o fluxo do objeto, em que, ao conhecer o modelo do movimento, possibilita procurar na próxima imagem qual a melhor correspondência.

Figura 8 – Processo de rastreamento.



Fonte: Forsyth e Ponce (2002)

Segundo Bekhit (2022), é possível rastrear o objeto após detectá-los no primeiro frame. Porém, caso o objeto seja sobreposto ou se mova rapidamente poderá ser perdido durante o rastreamento. Dependendo também da posição da câmera os objetos vão ficando cada vez mais distantes e é normal que erros possam ocorrer.

3.1.5 Fluxo óptico

O fluxo óptico pode ser associado ao movimento dos pixels divididos em dois tipos de algoritmos entre uma sequência de imagens, sendo classificados como densos e esparsos.

Algoritmos densos associam a movimentação ou deslocamento de cada pixel entre as imagens, não precisando do conhecimento de pontos específicos de imagens anteriores. Já os algoritmos de fluxo óptico esparsos levam em consideração algum tipo de informação sobre o objeto a ser rastreado, como por exemplo as bordas, fazendo com que obtenham um rastreamento mais robusto e confiável (BRADSKI; KAEHLER, 2008).

Um dos algoritmos de fluxo óptico mais conhecidos é o de Lucas Kanade, que depende das informações que são derivadas de uma pequena janela ao redor de cada ponto de interesse. Uma das desvantagens é que, por considerar apenas essas pequenas janelas, movimentos rápidos do objeto na imagem podem fazer com que ele deixe de ser rastreado.

Figura 9 – Detecção por fluxo óptico



Fonte: MatLab

Segundo Bradski e Kaehler (2008) o algoritmo de Lucas Kanade se baseia em 3 suposições:

1. Brilho constante: A intensidade de um pixel para imagens em tons de cinza não muda à medida que é rastreado em cada frame.
2. Pequenos movimentos: O movimento da imagem de uma superfície muda lentamente com o tempo. Ou seja, a intensidade do movimento para um pixel é pequena.
3. Coerência espacial: Os pontos vizinhos de um objeto possuem um movimento semelhante e se moverão juntos nos próximos frames.

3.2 Frameworks de visão computacional

Com o avanço do poder de processamento e interesse pela visão computacional, a biblioteca OpenCV ajuda na difusão da área, tornando tarefas acessíveis a estudantes e profissionais que antes só eram realizadas em laboratórios, segundo Bradski e Kaehler (2008).

O OpenCV é uma biblioteca de visão computacional de código-fonte aberto disponível gratuitamente. A biblioteca é escrita em C e C++, é multiplataforma e pode ser utilizada em Python, Ruby, Matlab e outras linguagens de programação.

A biblioteca OpenCV foi desenvolvida visando à otimização e utilização em aplicações em tempo real para visão computacional e aprendizado de máquina.

3.3 Frameworks WEB

Os *frameworks* são ferramentas que contribuem para a construção de outras aplicações semelhantes através da reutilização de análises, projetos e códigos. Assim, os problemas são desmembrados em partes menores, descrevendo interfaces a serem implementadas e restrições que devem ser feitas por parte do desenvolvedor e também possibilitando o uso de uma biblioteca de componentes de forma a colaborar com o desenvolvimento de uma nova aplicação (MALDONADO *et al.*, 2002).

Diversos *frameworks* têm sido desenvolvidos para oferecer soluções aos programadores a fim de contribuir com o desenvolvimento de aplicações de forma que erros não sejam repetidos com frequência, retardando o progresso de uma aplicação (VUKSANOVIC; SUDAREVIC, 2011).

Para este trabalho, foi escolhido o Flask para contribuir com o desenvolvimento da aplicação. A escolha por esse framework em especial se deu por alguns motivos detalhados a seguir.

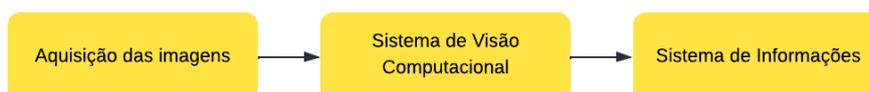
3.3.1 Flask

O Flask¹ é considerado um micro framework para aplicações web em Python que oferece as ferramentas certas para construir rapidamente uma aplicação WEB leve e sem dependências de bibliotecas de terceiros. O Flask permite que o desenvolvedor tenha liberdade quando estiver implementando as aplicações e fornece também compatibilidade para que extensões sejam importadas para adicionar novas funcionalidades à aplicação, diferentemente de outros *frameworks* no mercado.

4 Abordagem proposta

A presente seção detalha a arquitetura e desenvolvimento do sistema de visão computacional para controle do fluxo de pessoas em ambientes vídeo-monitorados. O diagrama da Figura 10 ilustra, de forma geral, as etapas para a construção desse sistema de visão computacional. Todo o processo do sistema desenvolvido começa com a captura das imagens, essas são então enviadas ao sistema de visão computacional que faz a contagem das pessoas em cena. Por fim, esses dados são recebidos pelo sistema que apresenta visualmente alguns componentes para auxiliar o usuário na gestão. Os detalhes de cada parte do sistema, serão apresentados detalhadamente nas seções seguintes.

Figura 10 – Fluxo geral da metodologia do trabalho.



Fonte: Elaborado pelo autor

¹ <https://flask.palletsprojects.com/en/2.1.x/>

4.1 Aquisição

Para realizar a aquisição dos vídeos, foi escolhida uma câmera que capture apenas a luz visível, devido ao baixo custo. Câmeras com sensores infravermelhos foram descartadas pois suas características e materiais fazem com que se tornem mais caras dependendo do comprimento de onda infravermelha que capturam.

Para este trabalho foi utilizada a webcam HP HD-4110 de 13MP com resolução de até 1920x1080. A câmera foi colocada a uma altura de aproximadamente 3 metros do chão, na posição zenital, ou seja, de forma que seja feita a captura do topo da cabeça e ombros de uma pessoa nas imagens.

Figura 11 – Exemplo de imagem obtida pela câmera.



Fonte: Elaborado pelo autor

Tem-se um desafio ao posicionar a câmera, pois é necessário achar uma angulação que diminua o nível de objetos sobrepostos na imagem e que permita capturar a movimentação das pessoas de forma que o rastreamento consiga ser realizado. Por isso, o posicionamento zenital foi escolhido, visando diminuir a sobreposição dos objetos em estudo, também conhecida como oclusão.

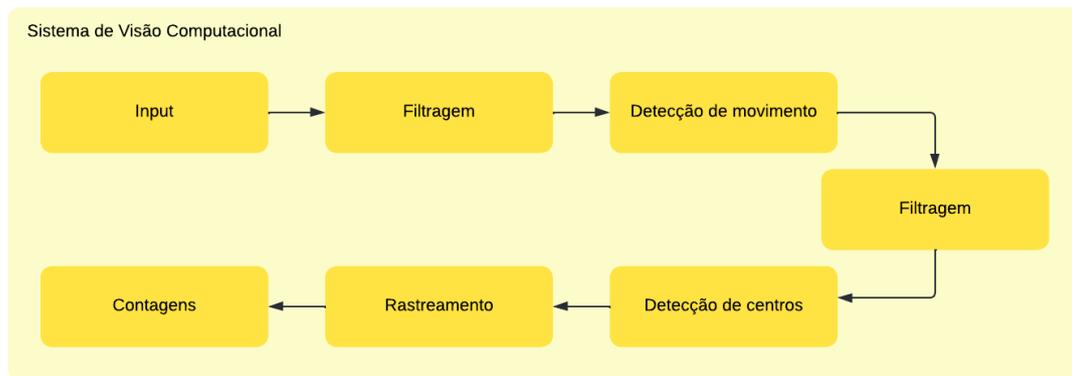
Os vídeos utilizados foram gravados em uma sala da Universidade Federal de Santa Catarina, campus Mato Alto, com resolução de captura de 800x600 pixels. No vídeo foram reproduzidos diferentes tipos de movimentações a fim de observar como o sistema se comporta em cada uma delas. Algumas movimentações reproduzidas foram: uma pessoa andando sozinha na cena, duas ou mais pessoas caminhando na mesma direção e pessoas caminhando em direções opostas.

Os vídeos gravados apresentam três indivíduos adultos. Apesar de não terem sido feitos testes com crianças, animais ou outros objetos em cena,

4.2 Sistema de visão computacional

O diagrama da Figura 12 ilustra de forma detalhada as etapas que compõem o sistema de visão computacional. Basicamente, as imagens capturadas através da câmera passam pela filtragem de ruídos, depois são enviadas para a detecção de movimento que irá aplicar o algoritmo de fluxo óptico, mais uma filtragem é realizada para eliminar ruídos adicionados. Com a imagem já binarizada, é feita a detecção dos centros de cada objeto em cena para possibilitar o rastreamento dos mesmos. Por fim, a contagem do fluxo de pessoas é realizada de acordo com os centróides localizados e informações do fluxo óptico para definição da direção. Todas as etapas do processo estão detalhadas nas próximas seções.

Figura 12 – Fluxo dos dados no sistema de visão computacional.



Fonte: Elaborado pelo autor

4.2.1 Input

O primeiro passo é realizar a captura das imagens de 800x600 pixels. Devido ao algoritmo utilizado, é necessário ter sempre os dois últimos *frame* do vídeo. Para garantir que isto ocorra é feita uma verificação antes de dar continuidade nas próximas etapas, apenas após já ter-se capturado o primeiro *frame* em uma variável o processamento do algoritmo continua. Também durante esta etapa, as imagens obtidas são convertidas da escala de cores BGR para tons de cinza. Ao final da execução do algoritmo a variável que armazena o primeiro *frame*, passa a armazenar o segundo *frame* e a variável que armazena o segundo *frame* recebe a nova imagem obtida pela câmera.

4.2.2 Filtragem

Nesta etapa é realizado o primeiro processo para reduzir os possíveis ruídos adicionados durante a captura. Para escolher o melhor filtro a ser utilizado levou-se em conta algumas considerações, como o tempo de execução da filtragem, a redução de ruídos e a preservação de bordas nas imagens filtradas.

O OpenCV disponibiliza alguns algoritmos de filtragem como o de desfoque, desfoque por mediana, desfoque gaussiano e bilateral. Dentre esses filtros o que apresentou a melhor solução para as considerações realizados foi o filtro de desfoque gaussiano.

Sendo assim, o processo de filtragem consiste na aplicação de um filtro de desfoque gaussiano com uma máscara 3x3. De forma que o filtro é aplicado uma vez na primeira

imagem após sua captura e depois será sempre aplicado a nova imagem capturada. A Figura 13 mostra a imagem antes da aplicação e após a aplicação do filtro de desfoque gaussiano. É possível perceber em algumas áreas um leve desfoque em relação à figura original.

Figura 13 – Aplicação do filtro gaussiano.



Fonte: Elaborado pelo autor

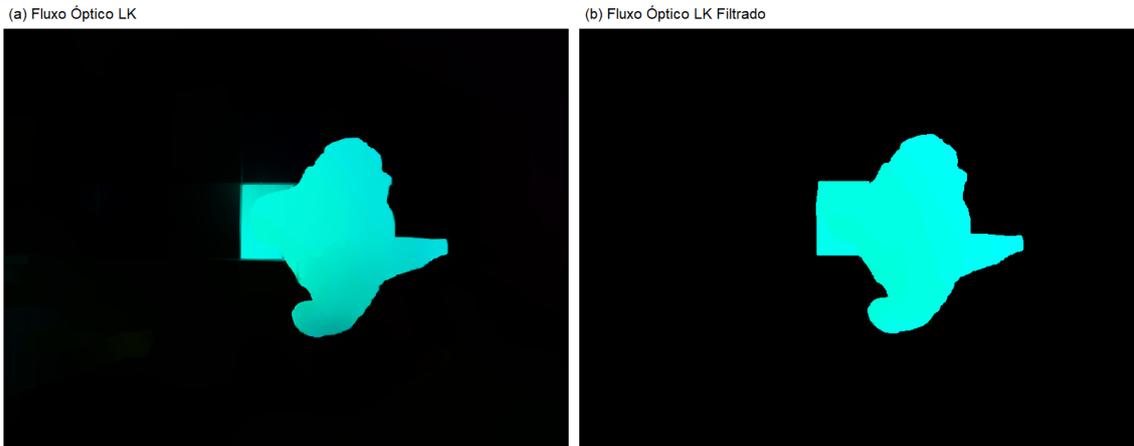
4.2.3 Detecção de movimento e filtragem

Para realizar a detecção de movimento foi utilizado o algoritmo de Lucas Kanade. Cada movimento do pixel no *frame* anterior é projetado para o *frame* seguinte, tornando possível obter um vetor de velocidade e ângulo para cada pixel na imagem.

A função da biblioteca OpenCV retorna o fluxo obtido entre as imagens, sendo necessário realizar os cálculos dos vetores de magnitude e ângulo para ter a imagem no domínio HSV. Com a imagem neste domínio fica fácil sua conversão para o sistema de cores BGR. A Figura 14-a mostra a representação do fluxo óptico obtido.

Contudo ao realizar todas essas operações e transformações, novos ruídos e objetos indesejados podem ter sido adicionados ao resultado. Com o intuito de eliminar as pequenas movimentações e ruídos, causados por conta da mudança de iluminação e sombras geradas a partir do movimento de uma pessoa, é efetuado um corte de acordo com a magnitude do vetor de velocidade gerado. Assim, regiões que não possuem uma certa quantidade de movimento passam a ser escurecidas e regiões com movimento passam a ser realçadas conforme mostra a Figura 14-b.

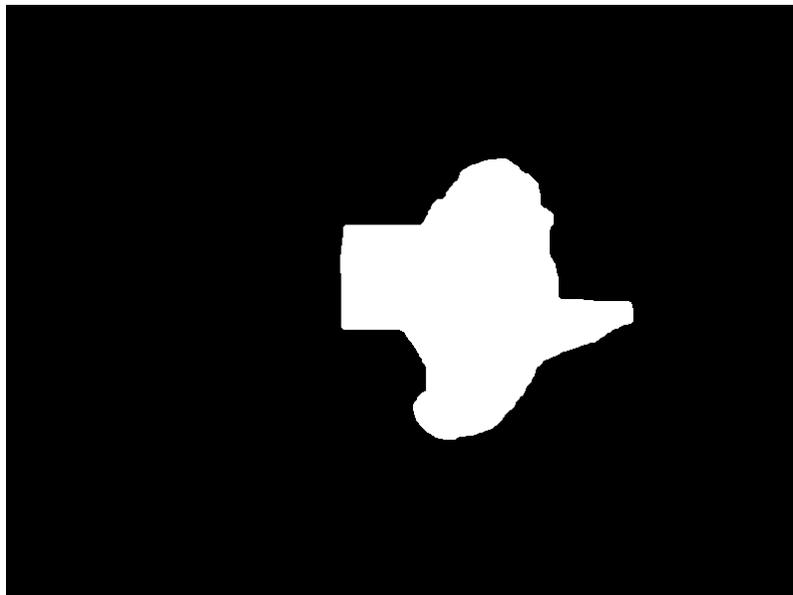
Figura 14 – Fluxo óptico obtido e após a filtragem.



Fonte: Elaborado pelo autor

Com a nova imagem, uma operação morfológica de abertura (erosão seguida de dilatação) é aplicada, para que ruídos sejam eliminados e partes segmentadas que compõem o objeto sejam conectadas. Por fim, é aplicada a limiarização binária, para que os objetos detectados em movimento na imagem sejam apresentados na cor branca e todo o restante da cena na cor preta conforme mostrado na Figura 15.

Figura 15 – Operações morfológicas de abertura e limiarização binária.

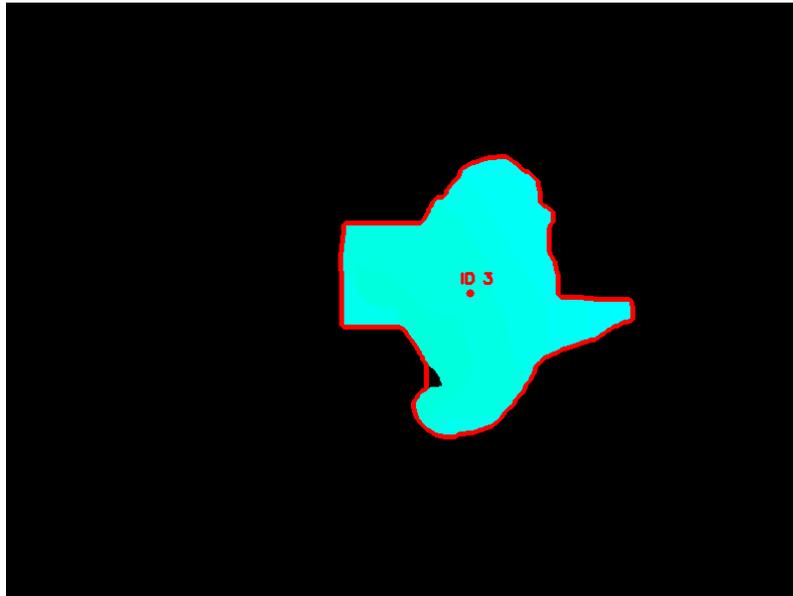


Fonte: Elaborado pelo autor

4.2.4 Detecção de centros

Com a imagem binarizada, a função *findContours* da biblioteca OpenCV é utilizada. Esta função será responsável por retornar um conjunto de pontos que representam as bordas do objeto e através desses pontos é possível calcular o centróide de cada elemento em movimento que está sendo capturado em cena.

Figura 16 – Detecção de centro do objeto.



Fonte: Elaborado pelo autor

Os centróides são os pontos centrais no plano cartesiano (x,y) de cada objeto detectado e constituem uma parte importante no sistema. Por isso a cada novo *frame* todos os centróides de todos os objetos encontrados são armazenados em um vetor e enviados ao algoritmo de rastreamento. Assim, o centróide atuará como um identificador do objeto, mantendo sua posição atualizada em cada *frame*, auxiliando no rastreamento do objeto nos *frame* subsequentes.

4.2.5 Rastreamento

O algoritmo de rastreamento funciona da seguinte maneira: ao iniciar o algoritmo, não há nenhum objeto registrado, logo, todos os centróides recebidos pelo algoritmo de detecção de centros são considerados novos objetos e passam a ter algumas propriedades, como um identificador único, um contador de desaparecimento e uma direção. No próximo *frame* do vídeo, as posições dos objetos em cena são atualizadas e, conseqüentemente, os centróides estão em novas posições. O algoritmo precisa, então, realizar algumas verificações para incluir objetos que são novos, atualizar a posição dos que já estavam em cena e desregistrar objetos que desapareceram.

Primeiramente, o algoritmo atualiza a posição dos centróides dos objetos conhecidos através do cálculo da distância euclidiana com os novos centróides passados pela etapa

de detecção de centros. Assim, a nova posição do centróide será aquela que tiver a menor distância.

Caso a distância do centróide analisado seja muito grande em relação aos novos pontos obtidos ou o número de novos centróides for menor que os já identificados, o contador de desaparecimento daquele objeto é incrementado e, ao atingir um número máximo, o algoritmo considera que o objeto saiu da cena, retirando-o do vetor de objetos a serem rastreados. Por fim, se há um novo centróide que não está no array de objetos já conhecidos, ele é registrado e passa a ser seguido nos próximos *frame*.

4.2.6 Contagens

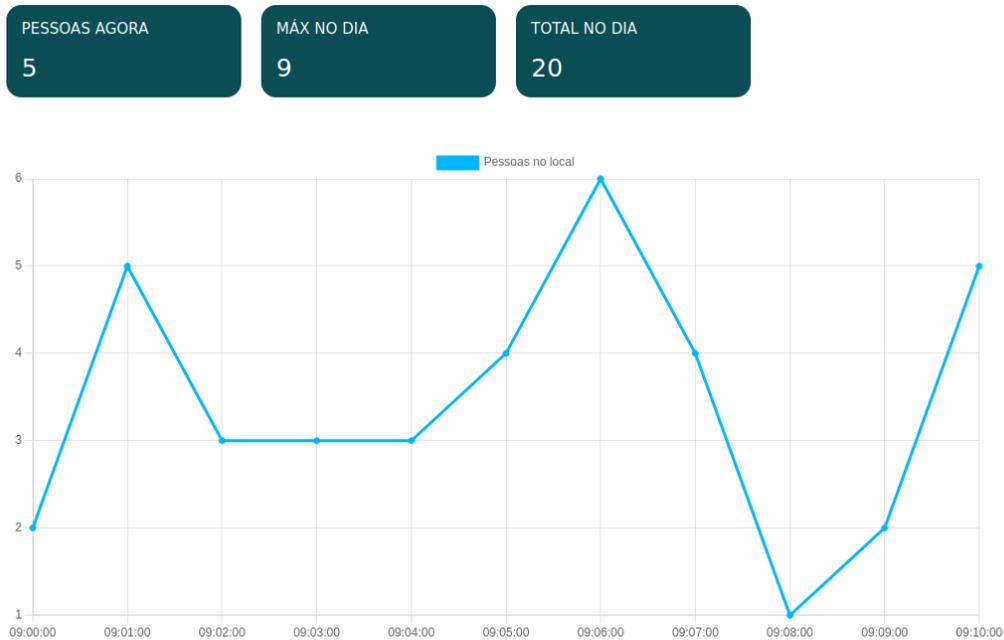
Após saber a posição de cada objeto, é necessário realizar a contagem das pessoas. Sabendo que o algoritmo de fluxo óptico atribui cores de acordo com a angulação de cada pixel, utilizou-se dessa propriedade para definir a direção das pessoas entrando e saindo da cena. Pixels com tons de vermelho incrementam o contador de pessoas entrando e pixels com tons azul incrementam o valor do contador de pessoas saindo.

No desenvolvimento observou-se que em alguns momentos devido a variação da luminosidade e criação de sombras, mesmo após a filtragem pontos de ruído eram definidos erroneamente como objetos no *frame* seguinte, recebendo um identificador e centróide para rastreamento. Sendo assim, com o intuito de contabilizar apenas ocorrências de objetos que estão em movimento em uma sequência maior de *frames* e mitigar falhas causadas pelo algoritmo de fluxo óptico e limiarização, implementou-se um controle para que os objetos só sejam contabilizados quando estiverem em cena por mais de três *frames*.

4.3 Sistema de Informações

O sistema de informações, é um servidor WEB que foi implementado na linguagem de programação Python. Para facilitar o gerenciamento das requisições, diminuir a chance de erros e aumentar a velocidade de desenvolvimento, utilizou-se o microframework Flask. Através de requisições HTTP ao servidor, o usuário do SGI obtém as informações enviadas pelo sistema de visão computacional. A Figura 17 ilustra a interface principal do SGI.

Figura 17 – Tela principal do sistema de informação.



Fonte: Elaborado pelo autor

Na Figura 17, é possível visualizar a tela do sistema no qual as principais informações são mostradas, como o número de pessoas no local, o horário do dia e o pico do fluxo, além de um gráfico com o número de pessoas em cada instante do dia para dar uma visão geral da lotação do local ao gestor, que baseado nessas informações, poderá tomar melhores decisões estratégicas afim de melhorar a performance e atendimento do seu estabelecimento.

5 Resultados experimentais

Foram analisados dois resultados para a obtenção da taxa de assertividade. O primeiro resultado consiste na classificação dos centróides obtidos após o fluxo óptico sem nenhum tipo de detecção de falhas no rastreamento. Já o segundo resultado contempla todas as etapas do algoritmo. Baseado nos dados dos trabalhos correlatos, a grande maioria dos autores mede a precisão do sistema através da equação 2, onde VP é a quantidade de objetos contados corretamente e FP a quantidade de objetos contados erroneamente.

$$P = \frac{VP}{VP + FP} * 100 \quad (2)$$

Os resultados são baseados nos dois vídeos utilizados durante os testes do projeto. As informações apenas do algoritmo de fluxo óptico estão presentes na Tabela 2, enquanto dados do algoritmo completo podem ser encontrados na Tabela 3.

Tabela 2 – Apresentação dos resultados obtidos para o modelo de fluxo óptico e limiarização para os vídeos capturados

Vídeo	Quantidade real	VP	FP	Precisão
1	20	20	8	71%
2	16	16	11	59%

Fonte: Elaborado pelo autor

Tabela 3 – Apresentação dos resultados obtidos para o sistema com tratamento de falhas para os vídeos capturados

Vídeo	Quantidade real	VP	FP	Precisão
1	20	20	0	100%
2	16	16	0	100%

Fonte: Elaborado pelo autor

É possível observar na Tabela 2 que o algoritmo de fluxo óptico acabou gerando vários falsos positivos para os dois vídeos. Ao analisar o vídeo notou-se que estes falsos positivos apareciam principalmente quando havia uma mudança brusca do movimento, como um objeto saindo de cena ou quando havia uma mudança na luminosidade do ambiente.

Porém apesar da assertividade do algoritmo de fluxo óptico não ter apresentado um ótimo resultado, os tratamentos das falhas foram suficientemente eficientes para entender quais objetos não eram reais na cena, elevando a precisão do sistema como um todo como observado na Tabela 3.

6 Conclusão e trabalhos futuros

O avanço da tecnologia em termos de velocidade de processamento em aparelhos cada vez menores e algoritmos mais eficientes tem permitido um progresso em várias áreas, como a da visão computacional, possibilitando a criação de novas soluções e o aprimoramento de outras já existentes. Neste trabalho, foi apresentado um sistema de visão computacional que possibilita a contagem de pessoas em ambientes vídeo-monitorados com o intuito de auxiliar na gestão do ambiente de acordo com as informações fornecidas por um sistema de informações.

Na abordagem utilizada foi possível visualizar falhas do algoritmo de fluxo óptico que acabaram gerando objetos inexistentes na cena em alguns *frames*. Quando não há nenhum tratamento, esses objetos inexistentes acabam recebendo centróides e são contabilizados como pessoas. Porém, através dessas observações realizou-se a implementação de uma correção para algoritmo, passando a considerar apenas objetos presentes em mais de 3 *frames*.

Em relação à precisão, um resultado significativo foi obtido considerando a aplicação da detecção de falhas durante o rastreamento, apresentando uma acurácia de 100% para os

testes realizados nos vídeos obtidos, mostrando que o sistema não ficou atrás dos trabalhos observados na Seção 2.

As possibilidades de utilização do sistema são inúmeras, tendo em vista que o módulo de captura e processamento foi implantado como um módulo independente do SIG, podendo ser escalado para coletar informações em diferentes pontos em que haja a passagem de pessoas.

Porém, o algoritmo deve ser testado em diferentes locais, com diferentes tipos de iluminação, sombras e com fluxo de pessoas mais intenso. Uma análise qualitativa do sistema implementado deve ser realizada em um trabalho futuro, mediante sua aplicação em situações reais. Nessas aplicações, poderá ser analisado como o sistema se comporta em relação a outros ambientes e como a sua utilização auxilia na tomada de decisão dos estabelecimentos.

Trabalhos futuros devem explorar ainda mais o sistema de informações, permitindo que o usuário tenha controle sobre qual algoritmo aplicar em cada módulo de processamento e admitir outras configurações como a seleção de uma janela de tempo nos gráficos ou o monitoramento em um período determinado pelo gestor.

Em resumo, o sistema apresentado neste trabalho foi capaz de alcançar elevados índices de desempenho nos vídeos aplicados e, quando incorporado em estabelecimentos, pode funcionar como um controle de lotação e uma ferramenta de gestão para ajudar nas decisões estratégicas para melhorar o desempenho do negócio.

Referências

- ANTIĆ, Borislav *et al.* K-means based segmentation for real-time zenithal people counting. In: 2009 16th IEEE International Conference on Image Processing (ICIP). [S.l.: s.n.], 2009. P. 2565–2568. DOI: <10.1109/ICIP.2009.5414001>.
- AQUINO, Estela. **Medidas de distanciamento social no controle da pandemia de COVID-19: potenciais impactos e desafios no Brasil.** *Ciência Saúde Coletiva*. [S.l.: s.n.], jun. 2020. P. 2423–2446. DOI: <org/10.1590/1413-81232020256.1.10502020>.Epub05>.
- BEKHIT, Ahmed Fathi. **Computer Vision and Augmented Reality in iOS: OpenCV and ARKit Applications.** 1. ed. [S.l.]: Appres, 2022. ISBN 978-1-4842-7461-3. DOI: <https://doi.org/10.1007/978-1-4842-7462-0>.
- BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV: Computer Vision with the OpenCV Library.** [S.l.]: O'Reilly, 2008. ISBN 0596516134.
- CETINKAYA, H. Hakan; AKCAY, Muammer. People Counting at Campuses. **Procedia - Social and Behavioral Sciences**, v. 182, p. 732–736, 2015. 4th WORLD CONFERENCE on EDUCATIONAL TECHNOLOGY RESEARCHES (WCETR-2014). ISSN 1877-0428. DOI: <https://doi.org/10.1016/j.sbspro.2015.04.821>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877042815030967>>.
- FILIPIC, Joaquín *et al.* People counting using visible and infrared images. **Neurocomputing**, v. 450, p. 25–32, 2021. ISSN 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2021.03.089>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0925231221004756>>.
- FORSYTH, David A.; PONCE, Jean. **Computer Vision: A Modern Approach.** [S.l.]: Prentice Hall Professional Technical Reference, 2002. ISBN 0130851981.
- GAO, Chenqiang *et al.* People counting based on head detection combining Adaboost and CNN in crowded surveillance environment. **Neurocomputing**, v. 208, p. 108–116, 2016. SI: BridgingSemantic. ISSN 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2016.01.097>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0925231216304660>>.
- GONÇALVES, Paulo Henrique. **Estimação do fluxo multidirecional de pedestres em ambientes abertos e não restritos, pela análise de seqüências de imagens digitais.** São Carlos: Universidade de Sao Paulo, Agencia USP de Gestao da Informacao Academica (AGUIA), 2005. DOI: <10.11606/d.18.2005.tde-02092006-132813>.
- GONZALEZ, Rafael C.; WOODS, Richard E. **Digital ImageProcessing.** 3. ed. [S.l.]: Pearson, 2008. ISBN 0-13-168728-x 978-0-13-168728-8.
- KAMIDA, Vitor Teruo. **Rastreamento de pessoas em sequência de imagens infravermelhas.** [S.l.: s.n.], mai. 2017.
- LI, Zhiqiang *et al.* Deep People Counting with Faster R-CNN and Correlation Tracking. In: PROCEEDINGS of the International Conference on Internet Multimedia Computing and Service. Xi'an, China: Association for Computing Machinery, 2016. (ICIMCS'16), p. 57–60. ISBN 9781450348508. DOI: <10.1145/3007669.3007745>. Disponível em: <<https://doi.org/10.1145/3007669.3007745>>.

- MALDONADO, José Carlos *et al.* **Padrões e Frameworks de Software**. [S.l.: s.n.], 2002.
- MELO, Victor Hugo Cunha de. **CONTAGEM DE PESSOAS POR VÍDEO USANDO CÂMERAS EM POSIÇÃO ZENITAL**. [S.l.: s.n.], 2011.
- MYINT, Ei Phyu; SEIN, Myint Myint. People Detecting and Counting System. In: 2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech). [S.l.: s.n.], 2021. P. 289–290. DOI: <10.1109/LifeTech52111.2021.9391951>.
- PÁDUA, Flávio. **Contagem de pedestres em tempo real baseada em visão**. Belo Horizonte, Brasil: [s.n.], 2002.
- PÁDUA, Flávio *et al.* Real-time pedestrian counting based on computer vision. In: PROC. of the Brazilian Symposium on Intelligent Automation. [S.l.: s.n.], 2003.
- PORE, Satish D.; MOMIN, B. F. Bidirectional people counting system in video surveillance. In: 2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT). [S.l.: s.n.], 2016. P. 724–727. DOI: <10.1109/RTEICT.2016.7807919>.
- RAMO, Nayara Cielly Freire do. **Método para contagem automática de pessoas em multidões utilizando drones**. [S.l.: s.n.], jun. 2017.
- SZELISKI, Richard. **Computer Vision Algorithms and Applications: Algorithms and Applications**. [S.l.]: Springer, 2010. ISBN 9783030343712.
- VUKSANOVIC, Irena Petrijevcin; SUDAREVIC, Bojan. Use of web application frameworks in the development of small applications. In: 2011 Proceedings of the 34th International Convention MIPRO. [S.l.: s.n.], 2011. P. 458–462.
- WANG, Zhen; YANG, Xiaojun. Moving Target Detection and Tracking Based on Pyramid Lucas-Kanade Optical Flow. In: 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC). [S.l.: s.n.], 2018. P. 66–69. DOI: <10.1109/ICIVC.2018.8492786>.