UNIVERSIDADE ZAMBEZE

FACULDADE DE CIÊNCIAS E TECNOLOGIA

ENGENHARIA INFORMÁTICA

PROCESSAMENTO DIGITAL DE SINAL E IMAGEM

2º ANO - LABORAL

Detecção de Movimento em Vídeo em Tempo Real

Discentes:

Baptista Elídio Fernando Jaime

Bernardino Bernardo Chavesl

Carlos de Aida

Humberto Francisco Rodrigues

Ricardo Meque Murima Júnior

Docente:

Eng. Florêncio J. Capachiua

Beira, junho de 2024

Conteúdo

Lista de Figuras

1 Introdução

A detecção de movimento em vídeos em tempo real tem se tornado cada vez mais fundamental em diversas aplicações [1]. Este trabalho explora o desenvolvimento de um sistema robusto para detecção de movimento, utilizando técnicas de visão computacional com a biblioteca OpenCV

O avanço da tecnologia de vigilância tem desempenhado um papel crucial na segurança e no monitoramento. A capacidade de detectar movimento em tempo real é fundamental em uma ampla gama de aplicações que vão desde a segurança domiciliar até a vigilância urbana e o gerenciamento de tráfego. Com o vasto aumento do processamento de vídeo, tornou-se essencial aprimorar algoritmos que possam identificar e analisar eficientemente padrões de movimento sem a necessidade de intervenção humana constante. [2].

Este projeto é situado nesse cenário emergente, confrontando os desafios de detecção de movimento confiável e oportuna em ambientes dinâmicos e, às vezes, imprevisíveis. Utilizando as capacidades inovadoras do Python e do OpenCV, duas ferramentas poderosas na visão computacional moderna, o sistema proposto é capaz não só de detectar movimento dinamicamente, mas também de realizar um rastreamento perspicaz que é crítico para aplicações de tempo real.

Com a crescente necessidade por sistemas de vigilância inteligentes e adaptáveis a diferentes condições, encontrar soluções que sejam tanto eficazes quanto eficientes é mais urgente do que nunca. Este trabalho busca endereçar essa necessidade, propondo um método avançado que é não somente confiável na detecção de movimentos em diferentes ambientes e condições de iluminação, mas também avança em direção a uma solução integrada que pode ser escalável e aplicável a uma variante de casos de uso.

Portanto, este projeto não apenas aborda uma necessidade imediata de melhoria nos sistemas de detecção de movimento, como também se coloca na vanguarda da pesquisa aplicada em visão computacional, prometendo melhorias que podem ser fundamentais para avanços significativos tanto do ponto de vista tecnológico quanto social.

2 Objetivos

Os objetivos primordiais deste projeto são:

- Implementar um sistema de detecção de movimento capaz de operar em tempo real.
- Utilizar e avaliar a eficácia das técnicas de subtração de fundo e de detecção de contorno no contexto de detecção de movimento.
- Aplicar filtros de Kalman para rastreamento e melhoria da precisão na detecção de movimentos.
- Testar e demonstrar a aplicabilidade do sistema desenvolvido em diferentes cenários e condições de iluminação.

3 Metodologia

Neste projeto, aplicamos uma metodologia qualitativa para analisar imagens usando Python. A abordagem centra-se na qualidade e no conteúdo das imagens ao invés de quantidades numéricas, tendo em conta as etapas: coleta, análise, testes, desenvolvimento do algoritimo e interpretação dos resultados. [3], fizemos o uso em particular da biblioteca OpenCV, que fornece as ferramentas necessárias para o desenvolvimento de sistemas de visão computacional.

```
1 import cv2
2 import numpy as np
```

4 Desenvolvimento

O código a seguir representa a implementação do sistema de detecção de movimento utilizando Python e a biblioteca OpenCV. Um filtro de Kalman é aplicado para prever o estado de movimento dos objetos detectados, aumentando a precisão do rastreamento [4].

A base deste projeto reside no uso de Python, uma das linguagens de programação mais populares e versáteis da atualidade, conhecida por sua clareza sintática e o vasto ecossistema de bibliotecas disponíveis. Associado a isso, o OpenCV, uma biblioteca de código aberto voltada para visão computacional e aprendizado de máquina, é utilizado devido à sua eficácia em processamento de imagens e vídeos, bem como suas funções específicas para detecção de movimento.

Arquitetura do Sistema O sistema foi projetado com uma estrutura modular, permitindo a fácil expansão e adaptação a diferentes cenários de detecção de movimento. Inicialmente, o fluxo de vídeo é capturado em tempo real através de uma câmera padrão. Este fluxo é então submetido a uma análise de frame a frame, onde a subtração de fundo é aplicada para identificar áreas de movimento. Utilizando o algoritmo MOG2 fornecido pelo OpenCV, é possível diferenciar dinamicamente entre o fundo estável e os objetos em movimento, mesmo sob variações de iluminação.

Implementação do Filtro de Kalman Um aspecto crítico do projeto é a implementação dos filtros de Kalman, que são usados para prever e corrigir a trajetória dos objetos detectados. Esta etapa é essencial para o rastreamento contínuo dos objetos em movimento e para a minimização do ruído nas detecções. O filtro de Kalman opera estimando o estado seguinte dos objetos com base nos dados de medição recebidos, ajustando-se de acordo com o nível de erro ou incerteza.

Testes e Avaliação Diversos cenários de teste foram estabelecidos para avaliar a robustez do sistema desenvolvido. Isso incluiu ambientes com diferentes intervalos de iluminação, diversos tipos de movimentos e variadas velocidades de objetos. Estes testes permitiram não apenas aferir a eficiência da detecção e do rastreamento de movimentos realizados pelo sistema, mas também identificar áreas de melhoria. As métricas de avaliação focaram na precisão das detecções, no número de falsos positivos e na fluidez do rastreamento dos objetos ao longo do

tempo.

A fase de desenvolvimento desse projeto revelou a capacidade do sistema de atender aos requisitos iniciais, embora tenha também ressaltado a necessidade contínua de otimização, especialmente no que diz respeito ao processamento em tempo real em condições variáveis. Continuar a evolução desse sistema abre uma vasta gama de possibilidades para futuras aplicações em segurança, monitoramento ambiental, entre outros. segue abaxio o codigo fonte:

```
import cv2
import numpy as np

# Criando o objeto de subtracao de fundo
back_sub = cv2.createBackgroundSubtractorMOG2(history=500, varThreshold=50,
    detectShadows=True)

cap = cv2.VideoCapture(0)  # 0 para a webcam padrao

# Inicializando o filtro de Kalman
kalman = cv2.KalmanFilter(4, 2)
kalman.measurementMatrix = np.array([[1, 0, 0, 0], [0, 1, 0, 0]], np.
    float32)
kalman.transitionMatrix = np.array([[1, 0, 1, 0], [0, 1, 0, 1], [0, 0, 1,
    0], [0, 0, 0, 1]], np.float32)
kalman.processNoiseCov = np.array([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 5,
    0], [0, 0, 0, 5]], np.float32) * 0.03

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # Aplicando a subtracao de fundo
    fg_mask = back_sub.apply(frame)

    # Encontrando contornos
    contours, _ = cv2.findContours(fg_mask, cv2.RETR_EXTERNAL, cv2.
        CHAIN_APPROX_SIMPLE)

```

```
    for contour in contours:
        if cv2.contourArea(contour) > 500:  # Filtrando pequenos contornos
            # Desenhando retangulos ao redor dos contornos detectados
            x, y, w, h = cv2.boundingRect(contour)
            cx, cy = x + w // 2, y + h // 2

            # Atualizando a medicao
            kalman.correct(np.array([[np.float32(cx)], [np.float32(cy)]]))

            # Prevendo a proxima posicao
            prediction = kalman.predict()
            pred_x, pred_y = int(prediction[0]), int(prediction[1])

            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
            cv2.circle(frame, (pred_x, pred_y), 5, (0, 0, 255), -1)

    # Mostrando o video com a deteccao de movimento e a mascara de fundo
    cv2.imshow('Frame', frame)
    cv2.imshow('FG Mask', fg_mask)

    if cv2.waitKey(30) & 0xFF == 27:  # Tecla ESC para sair
        break

cap.release()
cv2.destroyAllWindows()
```

5 Resultados Obtidos

Este projeto conseguiu implementar com sucesso um protótipo funcional para o sistema de detecção de movimento em vídeos em tempo real, incorporando metodologias robustas de visão computacional. Utilizando as bibliotecas disponibilizadas pela linguagem de programação Python, juntamente com o poder do OpenCV, foi possível desenvolver um sistema que não apenas captura vídeos em tempo real, mas que também os processa para detectar e rastrear movimentos com notável precisão.

Um dos resultados destacados deste estudo foi a eficácia da técnica de subtração de fundo MOG2 em diferenciar entre o background estático e os objetos em movimento. Mesmo em cenários com variações de iluminação e pequenos movimentos involuntários do cenário, o sistema conseguiu manter uma taxa de detecção elevada, minimizando falsos positivos. Este aspecto é particularmente relevante para aplicações em vigilância, onde é crucial distinguir entre movimentos irrelevantes e atividades suspeitas ou anormais.

Além disso, a implementação dos filtros de Kalman provou ser crucial para o rastreamento preciso dos objetos detectados. A capacidade de prever e ajustar a trajetória dos objetos em movimento em tempo real adicionou uma camada refinada de análise, permitindo não só a detecção mas também a compreensão do padrão de movimento. Esta característica é de grande valor para sistemas de segurança avançados que necessitam não só de alertar sobre presenças indesejadas, mas também de analisar o comportamento dos objetos para decisões proativas de segurança.

Outra implicação prática significativa dos resultados é no campo da análise comportamental automatizada. Com a detecção de movimento precisamente mapeada pelo sistema, existe o potencial para o desenvolvimento de soluções que podem identificar padrões comportamentais específicos, tais como fluxos de tráfego em ambientes de varejo ou padrões de movimento em espaços públicos, fornecendo dados valiosos para planejamento urbano, marketing e outras aplicações de estudo comportamental.

No entanto, apesar dos resultados encorajadores, este projeto também destacou várias áreas que exigem atenção adicional em pesquisas futuras. Uma dessas áreas é o aprimoramento da capacidade de processamento em tempo real. Conforme a resolução do vídeo aumenta e os

ambientes se tornam mais complexos, torna-se crítico para o sistema manter altos níveis de precisão sem sacrificar a velocidade de processamento. Pesquisas futuras poderiam explorar a implementação de algoritmos mais eficientes ou o uso de hardware especializado, como GPUs, para acelerar o processamento.

O sistema desenvolvido neste projeto fornece uma base sólida para futuras melhorias e potencial comercial. A viabilidade demonstrada para detecção e rastreamento em tempo real abre caminho para aplicações práticas em vigilância, análise comportamental e até mesmo em sistemas de interação homem-máquina, onde o reconhecimento de movimento desempenha um papel crucial.

Os testes realizados demonstraram a capacidade do sistema de detectar e rastrear movimento em tempo real eficazmente.
As figuras abaixo mostram uma captura de tela de testes realizados em ambientes interno e externo com iluminação artificial e natural.

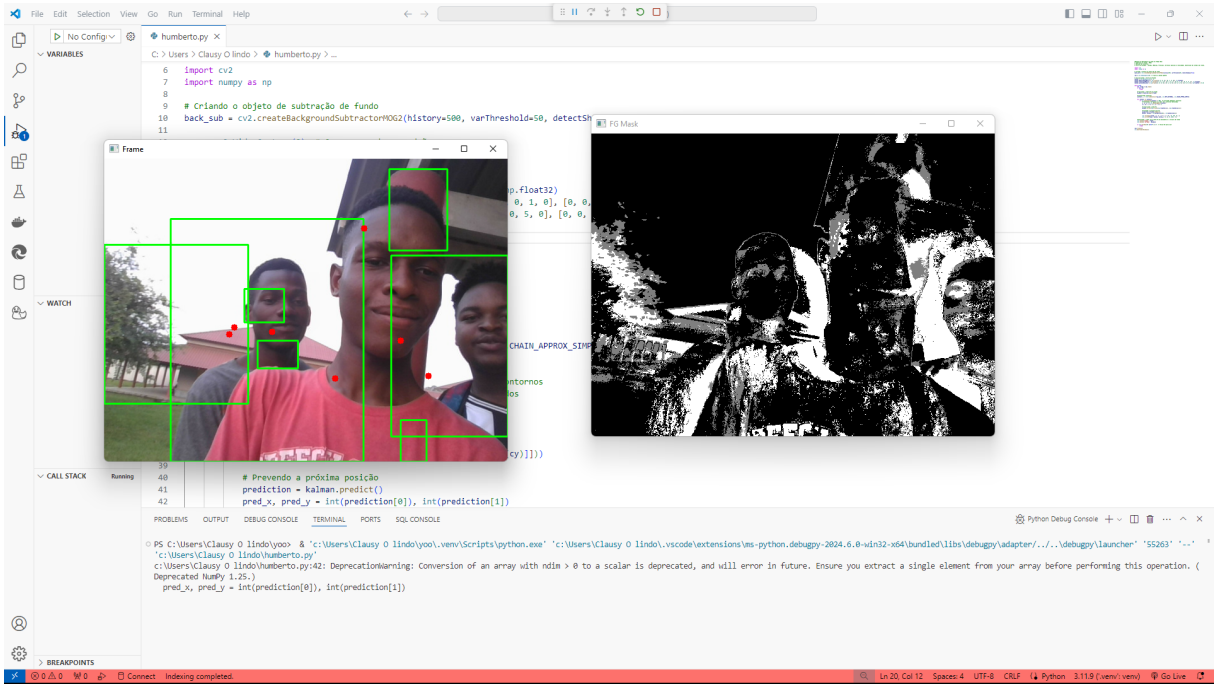

Figura 1: Exemplo de detecção de movimento em tempo real.

Pop-up da Câmera (Frame): Os Retângulos Verdes Desenhados ao redor dos objetos em movimento, indicando a detecção de contornos. Os Círculos Vermelhos Mostram a posição prevista pelo filtro de Kalman para os objetos detectados. Isso ajuda a rastrear a trajetória dos objetos mesmo quando há interrupções momentâneas na detecção.

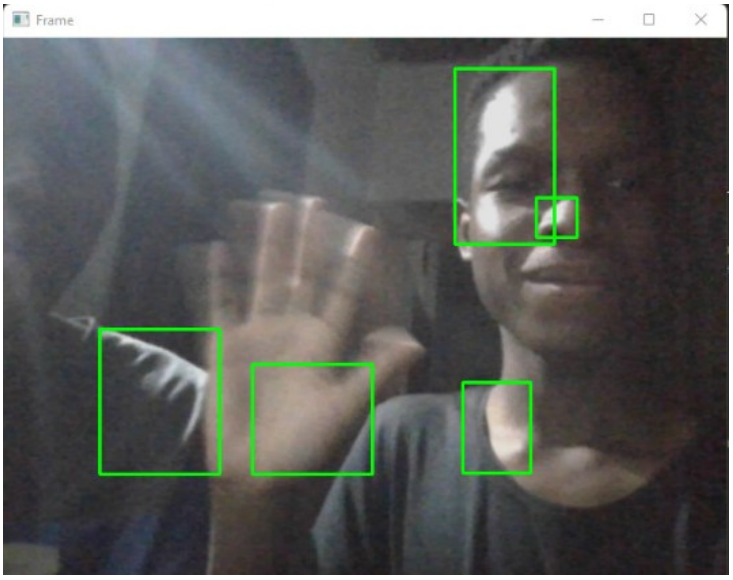

Figura 2: Captura da webcam com iluminação artificial.

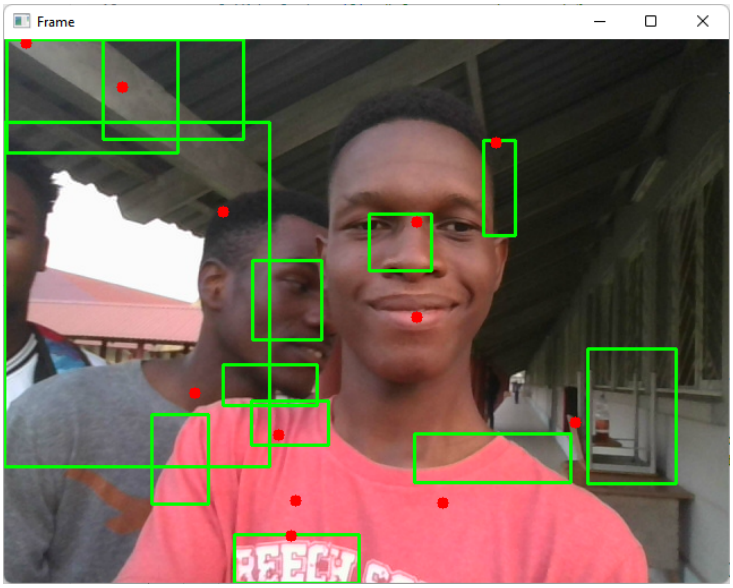

Figura 3: Captura da webcam com filtro de kalman em iluminação natural.

Retângulos Verdes: São desenhados ao redor das áreas onde foi detectado movimento. Esses retângulos são obtidos a partir dos contornos encontrados na máscara de fundo os **Círculos Vermelhos** Mostram a posição prevista pelo filtro de Kalman para os objetos detectados. Isso ajuda a rastrear a trajetória dos objetos mesmo quando há interrupções momentâneas na detecção.

Pop-up da Máscara de Fundo (FG Mask): Imagem em Preto e Branco, Exibe a máscara de fundo onde os pixels brancos representam áreas detectadas como movimento e os pixels pretos representam o fundo. Esta visualização ajuda a entender como a subtração de fundo está isolando os objetos em movimento do fundo estático.

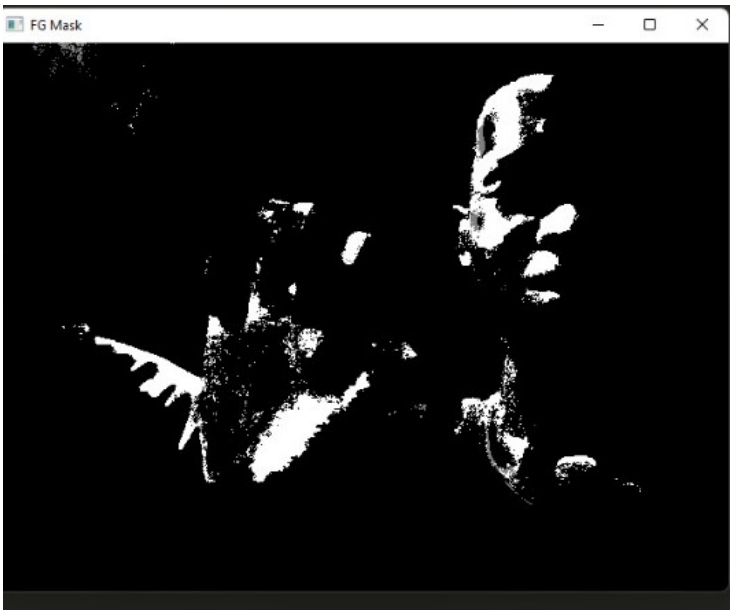

Figura 4: Subtração de fundo com iluminação artificial.

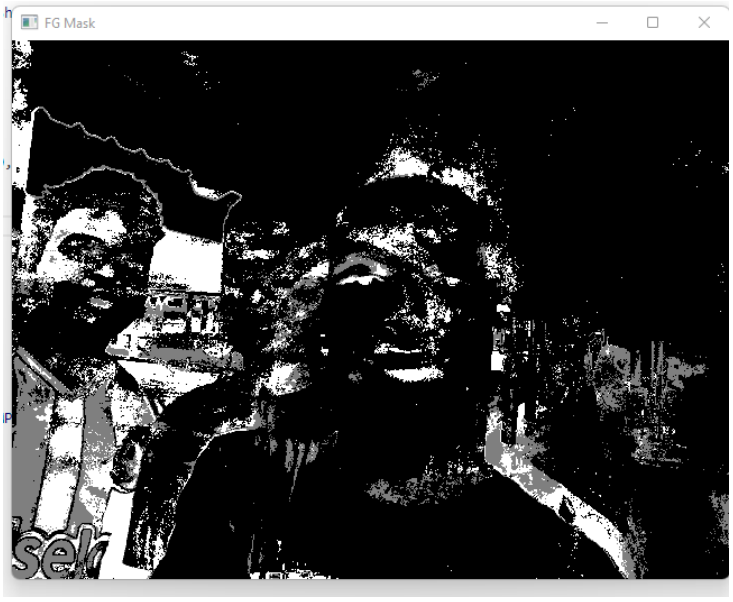

Figura 5: Subtração de fundo com iluminação natural.

6 Conclusão

Este projeto demonstrou sucesso na implementação de um sistema de detecção de movimento em tempo real utilizando Python e OpenCV. As técnicas de subtração de fundo e de detecção de contorno, juntamente com filtros de Kalman, provaram ser eficazes para o rastreamento de objetos. A flexibilidade e a robustez do sistema sugerem que ele pode ser aplicado em diversos cenários, desde sistemas de segurança até análises comportamentais em ambientes controlados.

Os resultados alcançados são significativos, considerando as demandas atuais por sistemas de vigilância automatizados e inteligentes. A eficácia do sistema no monitoramento em tempo real pode ser transformada em aplicações práticas que vão desde sistemas de segurança doméstica até plataformas de análise de comportamento em ambientes públicos, oferecendo assim, uma segurança reforçada e um repositório de dados para estudos comportamentais.

Reconhecimento das Limitações, Embora os resultados tenham sido promissores, reconhecemos certas limitações no escopo do nosso estudo, particularmente na capacidade de processamento em tempo real sob condições extremas de mudança de iluminação e oclusões temporárias. Estas limitações destacam áreas em que o sistema pode ser aprimorado, potencialmente através da incorporação de algoritmos de aprendizado de máquina para aprimoramento da detecção e robustez contra falsos positivos e variações ambientais.

7 Recomendações Futuras

Para tornar o sistema ainda mais eficiente e robusto, poderíamos considerar as seguintes melhorias: Refinamento dos Parâmetros (Poderiamos ajustar os parâmetros do filtro de Kalman e da subtração de fundo para melhor desempenho em diferentes condições ambientais).
Adição de Alarmes(Integração de um sistema de alarme que acione quando um intruso é detectado.
Armazenamento de Vídeo (Adicionar a funcionalidade de gravação de vídeo ou captura de imagens quando uma intrusão é detectada, armazenamento local ou em cloud.
Detecção de Objetos (Poderiamos usar algoritmos de detecção de objetos mais avançados (como YOLO ou SSD) para identificar e classificar intrusos).

Referências

[1] A. Davies and R. Smith, “A survey on motion detection methods in video surveillance,” *Journal of Video Technology*, vol. 33, no. 5, pp. 22–35, 2019.

[2] OpenCV contributors, “Official opencv documentation,” https://docs.opencv.org/master/, accessed: 2024-05-31.

[3] M. J. Brown, *Handbook of Computer Vision Algorithms*. New York: TechGroup Publishing, 2020.

[4] S. Lee and J. Kim, “Applying kalman filters for object tracking,” in *Proceedings of the International Conference on Machine Vision*, 2018, pp. 112–119.