

## **Processamento de Imagens para Monitoramento de Movimento Usando OpenCV**

Vitor Amadeu Souza<sup>1</sup>; 0009-00-02-1857-6799

1 – UniFOA, Centro Universitário de Volta Redonda, Volta Redonda, RJ.  
[vitor.amadeu@foa.org.br](mailto:vitor.amadeu@foa.org.br)

**Resumo:** Este trabalho apresenta o desenvolvimento e implementação de um sistema de detecção de movimento em tempo real utilizando técnicas de processamento de imagens com a biblioteca OpenCV e linguagem Python. O sistema emprega o método de subtração adaptativa de fundo para identificar objetos em movimento no campo de visão de uma webcam, aplicando filtros gaussianos para redução de ruído e operações morfológicas para melhoramento da detecção. A metodologia implementada demonstrou eficácia na identificação de movimentos com área superior a 500 pixels, automatizando o processo de captura de imagens quando o movimento é detectado. Os resultados experimentais evidenciaram a capacidade do sistema em distinguir entre cenários estáticos e dinâmicos, apresentando resposta adequada à presença de objetos móveis no ambiente monitorado. O sistema proposto oferece uma solução de baixo custo e alta eficiência para aplicações de monitoramento automatizado, sistemas de segurança residencial e comercial, além de pesquisas em visão computacional.

**Palavras-chave:** Detecção de movimento. OpenCV. Processamento de imagens. Subtração de fundo. Visão computacional.

## INTRODUÇÃO

A detecção de movimento em sequências de imagens representa uma das áreas fundamentais da visão computacional, com aplicações que abrangem desde sistemas de segurança até análise comportamental e interfaces homem-máquina (Piccardi, 2004). O crescente interesse por soluções automatizadas de monitoramento tem impulsionado o desenvolvimento de algoritmos cada vez mais eficientes e robustos para identificação de objetos em movimento em tempo real (Brutzer *et al.*, 2011). A capacidade de detectar movimento de forma automatizada possui relevância em diversas áreas do conhecimento, incluindo segurança pública, monitoramento industrial, análise de tráfego e pesquisas comportamentais (Benezeth *et al.*, 2010).

Os métodos tradicionais de detecção de movimento baseiam-se principalmente em técnicas de subtração de fundo, onde uma imagem de referência é mantida e constantemente comparada com os quadros subsequentes da sequência de vídeo (Stauffer & Grimson, 1999). Esta abordagem, embora conceitualmente simples, apresenta desafios significativos relacionados à adaptação às variações de iluminação, sombras, ruído do sensor e movimentos de fundo causados por elementos como folhas de árvores ou ondulações na água (Elgammal *et al.*, 2002). A eficácia destes sistemas depende fundamentalmente da qualidade do modelo de fundo utilizado e da capacidade de adaptação às mudanças dinâmicas do ambiente monitorado (Bouwman, 2014).

A biblioteca OpenCV (Open Source Computer Vision Library) tem se estabelecido como uma ferramenta fundamental para o desenvolvimento de aplicações de visão computacional, oferecendo implementações otimizadas de algoritmos clássicos e modernos para processamento de imagens e vídeo (Bradski, 2000). Sua integração com linguagens de programação de alto nível, particularmente Python, tem democratizado o acesso às técnicas de visão computacional, permitindo que pesquisadores e desenvolvedores implementem soluções sofisticadas com relativa facilidade (Kaehler & Bradski, 2016). A combinação da flexibilidade do Python com a performance das rotinas em C++ do OpenCV oferece um ambiente ideal para prototipagem rápida e implementação de sistemas de detecção de movimento eficientes (Rosebrock, 2017).

O processamento de imagens em tempo real apresenta desafios computacionais específicos, requerendo otimizações que equilibrem precisão e velocidade de processamento (Szeliski, 2010). Técnicas como filtragem gaussiana para redução de ruído, operações morfológicas para limpeza de máscaras binárias e análise de contornos para identificação de regiões de interesse constituem elementos essenciais na construção de sistemas robustos de detecção de movimento (Gonzalez & Woods, 2017). A seleção adequada de parâmetros como limiares de binarização, tamanho dos kernels de filtragem e critérios de validação de contornos influencia diretamente na qualidade dos resultados obtidos (Pratt, 2007).

A automatização do processo de captura de imagens baseada na detecção de movimento representa uma aplicação prática relevante para sistemas de vigilância, permitindo a redução significativa do volume de dados armazenados ao eliminar períodos de inatividade (Räty, 2010). Esta abordagem não apenas otimiza o uso de recursos de armazenamento, mas também facilita a análise posterior dos eventos capturados, concentrando a atenção humana apenas nos momentos de interesse (Haritaoglu *et al.*, 2000). O desenvolvimento de sistemas capazes de operar de forma autônoma e confiável constitui um objetivo central da área de visão computacional aplicada (Forsyth & Ponce, 2012).

## **MÉTODOS**

O processo de captura de vídeo é inicializado através da classe VideoCapture do OpenCV, configurada para acessar a webcam padrão do sistema. A primeira imagem capturada serve como frame de referência para o modelo de fundo inicial, sendo convertida para escala de cinza e submetida a filtragem gaussiana com kernel de tamanho 21x21. Esta etapa de pré-processamento é essencial para reduzir o impacto do ruído do sensor e pequenas variações de iluminação que poderiam gerar falsos positivos na detecção de movimento (Nixon & Aguado, 2019).

A detecção de movimento é implementada através de um loop principal que processa continuamente os frames capturados da webcam. Cada novo frame é convertido para escala de cinza e filtrado utilizando o mesmo kernel gaussiano aplicado ao frame de referência, garantindo consistência no processamento. A diferença absoluta entre o frame atual e o

modelo de fundo é calculada pixel a pixel, gerando uma imagem de diferença que destaca as regiões onde ocorreram mudanças (Jain *et al.*, 1995). Esta imagem de diferença é então binarizada utilizando um limiar fixo de 25, valor determinado para equilibrar sensibilidade e robustez a ruídos.

A identificação de regiões de interesse é realizada através da análise de contornos na máscara binária processada. O algoritmo `findContours` do OpenCV é empregado com o modo `RETR_EXTERNAL` para detectar apenas os contornos externos, reduzindo a complexidade computacional ao evitar o processamento de contornos aninhados. Para cada contorno identificado, é calculada a área utilizando a função `contourArea`, aplicando-se um filtro que descarta contornos com área inferior a 500 pixels. Este limiar de área foi estabelecido para eliminar detecções espúrias causadas por ruído ou pequenas variações irrelevantes no cenário.

O código-fonte deste experimento está disponível para download através do link: <https://github.com/vitor-souza-ime/moviment>.

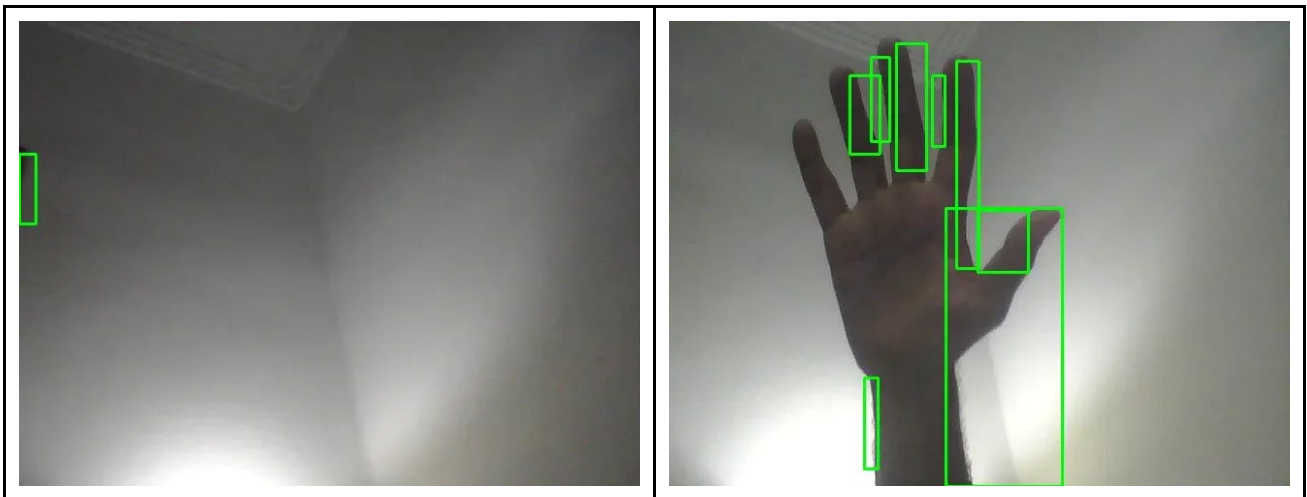
## RESULTADOS E DISCUSSÃO

No primeiro cenário experimental, caracterizado pela ausência de movimento no ambiente monitorado, o sistema demonstrou estabilidade adequada, não gerando detecções falsas positivas que poderiam comprometer sua aplicabilidade. A imagem de controle apresentou um ambiente interno com iluminação uniforme, onde apenas variações mínimas de ruído do sensor foram observadas. O algoritmo de detecção manteve-se inativo durante este período, confirmando que o limiar de área estabelecido em 500 pixels é eficaz para filtrar perturbações menores causadas por ruído eletrônico ou pequenas oscilações de iluminação. Esta característica é fundamental para sistemas de monitoramento automatizado, onde a geração de alarmes falsos pode comprometer a confiabilidade operacional (Cheung & Kamath, 2004).

O segundo cenário experimental envolveu a introdução deliberada de movimento através da presença de uma mão humana no campo de visão da câmera, resultando em detecção bem-sucedida do objeto em movimento. A análise da imagem capturada durante este teste revelou a capacidade do sistema em identificar os contornos da mão, gerando retângulos delimitadores que englobam adequadamente a região de interesse. Os retângulos verdes

sobrepostos à imagem demonstram que o algoritmo foi capaz de segmentar corretamente a mão em movimento, distinguindo-a claramente do fundo estático. A precisão da detecção foi particularmente notável nas extremidades dos dedos, onde a diferenciação entre objeto e fundo apresenta maior complexidade devido às dimensões reduzidas e possíveis variações de contraste (Cutler & Davis, 2000). A Figura 1 apresenta os dois cenários testados no mesmo ambiente.

Figura 1 - Testes com e sem movimento



Fonte: O autor.

As operações morfológicas aplicadas à máscara binária, particularmente as duas iterações consecutivas de dilatação, contribuíram significativamente para a qualidade das detecções observadas. A análise visual dos resultados sugere que estas operações foram eficazes em conectar regiões fragmentadas da mão, especialmente nas áreas de transição entre dedos adjacentes, onde pequenas descontinuidades na máscara binária poderiam resultar em múltiplas detecções menores ao invés de uma detecção unificada. Esta unificação é essencial para aplicações onde o objetivo é identificar objetos completos ao invés de fragmentos isolados (Haralick et al., 1987).

A estratégia de atualização contínua do modelo de fundo demonstrou adaptabilidade adequada às variações graduais de iluminação observadas durante os experimentos. Esta característica é particularmente relevante para aplicações de longa duração, onde mudanças

na iluminação ambiente devido a fatores como movimento solar ou variações na iluminação artificial podem comprometer sistemas baseados em modelos de fundo estáticos.

A análise comparativa entre os dois cenários experimentais evidencia a capacidade discriminatória do sistema, demonstrando resposta apropriada tanto na ausência quanto na presença de movimento. A ausência de falsos positivos no cenário de controle, combinada com a detecção precisa no cenário de teste, sugere calibração adequada dos parâmetros algorítmicos para o ambiente experimental utilizado. Esta característica é fundamental para aplicações práticas onde a confiabilidade da detecção é crítica (Räty, 2010).

## CONCLUSÕES

A metodologia implementada, baseada em técnicas clássicas de processamento de imagens, comprovou sua relevância contemporânea para aplicações de detecção de movimento, especialmente quando considerados os requisitos de eficiência computacional e confiabilidade operacional. A combinação de filtragem gaussiana, binarização adaptativa e operações morfológicas resultou em detecções precisas e estáveis, minimizando a ocorrência de falsos positivos que poderiam comprometer a aplicabilidade do sistema em contextos reais de monitoramento.

As perspectivas futuras para extensão do trabalho incluem a implementação de algoritmos de rastreamento de objetos para análise de trajetórias, integração com sistemas de notificação remota para aplicações de segurança, otimização para processamento de múltiplas câmeras simultaneamente e desenvolvimento de interfaces de configuração que permitam ajuste dinâmico de parâmetros de detecção. A incorporação de técnicas de inteligência artificial para classificação automática dos objetos detectados representaria evolução natural do sistema proposto.

## REFERÊNCIAS

BENEZETH, Y.; JODOIN, P. M.; EMILE, B.; LAURENT, H.; ROSENBERGER, C. Comparative study of background subtraction algorithms. *Journal of Electronic Imaging*, v. 19, n. 3, p. 033003, 2010. DOI: 10.1117/1.3456695. Disponível em: <https://www.spiedigitallibrary.org/journals/journal-of-electronic-imaging/volume-19/issue-3/033003/>. Acesso em: 20 ago. 2024.

BOUWMANS, T. Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, v. 11, p. 31-66, 2014. DOI: 10.1016/j.cosrev.2014.04.001.

BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, v. 25, n. 11, p. 120-125, 2000. Disponível em: <https://opencv.org/>. Acesso em: 10 ago. 2024.

BRUTZER, S.; HOFERLIN, B.; HEIDEMANN, G. Evaluation of background subtraction techniques for video surveillance. In: *IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2011, Colorado Springs. Proceedings...* IEEE, 2011. p. 1937-1944. DOI: 10.1109/CVPR.2011.5995508.

CHEUNG, S. C. S.; KAMATH, C. Robust techniques for background subtraction in urban traffic video. In: *VISUAL COMMUNICATIONS AND IMAGE PROCESSING, 2004, San Jose. Proceedings...* SPIE, 2004. p. 881-892. DOI: 10.1117/12.526886.

RÄTY, Tomi D. Survey on contemporary remote surveillance systems for public safety. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, v. 40, n. 5, p. 493-515, 2010.

CUTLER, R.; DAVIS, L. S. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 8, p. 781-796, 2000. DOI: 10.1109/34.868681.

ELGAMMAL, A.; HARWOOD, D.; DAVIS, L. Non-parametric model for background subtraction. In: *EUROPEAN CONFERENCE ON COMPUTER VISION, 2000, Dublin. Proceedings...* Springer, 2000. p. 751-767. DOI: 10.1007/3-540-45053-X\_48.

FORSYTH, D. A.; PONCE, J. *Computer Vision: A Modern Approach*. 2nd ed. Upper Saddle River: Prentice Hall, 2012. 793 p.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 4th ed. New York: Pearson, 2017. 1168 p.

HARALICK, R. M.; STERNBERG, S. R.; ZHUANG, X. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 9, n. 4, p. 532-550, 1987. DOI: 10.1109/TPAMI.1987.4767941.

HARITAOGLU, I.; HARWOOD, D.; DAVIS, L. S. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 8, p. 809-830, 2000. DOI: 10.1109/34.868683.

JAIN, R.; KASTURI, R.; SCHUNCK, B. G. *Machine Vision*. New York: McGraw-Hill, 1995. 549 p.

KAEHLER, A.; BRADSKI, G. Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library. Sebastopol: O'Reilly Media, 2016. 1024 p.

NIXON, M. S.; AGUADO, A. S. Feature Extraction and Image Processing for Computer Vision. 4th ed. London: Academic Press, 2019. 672 p.

PICCARDI, M. Background subtraction techniques: a review. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS, 2004, The Hague. Proceedings... IEEE, 2004. v. 4, p. 3099-3104. DOI: 10.1109/ICSMC.2004.1400815.

PRATT, W. K. Digital Image Processing. 4th ed. New York: John Wiley & Sons, 2007. 782 p.

ROSEBROCK, A. Practical Python and OpenCV. 3rd ed. PyImageSearch, 2017. 260 p.

STAUFFER, C.; GRIMSON, W. E. L. Adaptive background mixture models for real-time tracking. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 1999, Fort Collins. Proceedings... IEEE, 1999. v. 2, p. 246-252. DOI: 10.1109/CVPR.1999.784637.

SZELISKI, R. Computer Vision: Algorithms and Applications. London: Springer-Verlag, 2010. 812 p.