

People tracking utilizando uma câmera acoplada no gimbal e suas aplicações

ALEXANDRE COSTA FERRO FILHO
alexandre_ferro@discente.ufg.br
MARCELO HENRIQUE LOPES FERREIRA
marcelomarcelo2@discente.ufg.br

MURILO DE OLIVEIRA GUIMARÃES
muriloguimaraes@discente.ufg.br
Werisson Ernesto Silva Pereira
werissonws@discente.ufg.br

UFG – Universidade Federal de Goiás
INF – Instituto de Informática
Cx. Postal 131 – CEP 74690-900 Goiânia (GO)

Resumo:

O rastreamento de pessoas é uma tarefa essencial na área de visão computacional e robótica, com aplicações práticas em segurança, vigilância e navegação autônoma. Neste artigo, apresentamos uma proposta de solução que combina técnicas de aprendizado profundo, detecção de objetos e rastreamento por correspondência para alcançar um rastreamento robusto. Além disso, integramos o sistema em uma câmera equipada com um gimbal, permitindo o acompanhamento do alvo. A proposta visa superar desafios como variações de iluminação, fundos complexos e presença de múltiplas pessoas. Além disso, é suposto que o artigo traga aplicações da tarefa proposta e comente sobre os resultados obtidos.

Palavras Chave:

Tracking, YOLO, Visão computacional, Gimbal.

Seção I. Introdução

O rastreamento robusto de pessoas tem sido objeto de interesse crescente devido à sua importância em diversas aplicações práticas. Este artigo propõe uma abordagem baseada em inteligência artificial (IA), que utiliza técnicas de aprendizado profundo para detecção de objetos e rastreamento por correspondência para acompanhar com precisão a trajetória de pessoas em sequências de vídeo ou em tempo real. Além disso, a integração com um gimbal possibilita o movimento automático da câmera, mantendo o alvo centralizado no quadro, ampliando ainda mais a utilidade do sistema.

O problema de rastreamento robusto de pessoas abrange diversas áreas de aplicação, como segurança e vigilância [1], análise de multidões, navegação autônoma, interatividade em jogos e realidade aumentada, e monitoramento de saúde e esportes[2]. Em cenários de segurança, o rastreamento auxilia na detecção de comportamentos suspeitos e monitoramento de espaços públicos. Já na análise de multidões, é utilizado para entender o fluxo de pessoas em eventos ou locais movimentados. Em veículos autônomos ou drones, o rastreamento permite seguir alvos em movimento e evitar colisões. Nos campos de jogos e realidade aumentada, é fundamental para a interação com elementos virtuais no mundo real, e em aplicações de monitoramento de saúde ou esportes, possibilita a análise de postura e movimentos específicos.

Seção II. Fundamentação teórica

Nossa proposta de solução baseia-se em uma combinação de técnicas de aprendizado profundo, detecção de objetos e rastreamento por correspondência. Inicialmente, utilizamos algoritmos de detecção de objetos, como YOLO (You Only Look Once) [3] ou Faster R-CNN, para identificar e delimitar as pessoas em cada frame do vídeo. Em seguida, aplicamos métodos de rastreamento por correspondência, considerando informações de posição, tamanho e características dos objetos para estabelecer correspondências entre frames consecutivos.

Uma forma utilizada de melhorar a robustez e suavidade do rastreamento, é empregar técnicas de filtragem de dados, como o Filtro de Kalman ou o Filtro de Partículas. Esses métodos reduzem ruídos e incertezas, resultando em um rastreamento mais preciso e confiável.

Para viabilizar o rastreamento de objetos no âmbito do hardware, foi utilizado um gimbal, que é um dispositivo mecânico que permite o movimento controlado da câmera em duas ou três direções, geralmente pan (movimento horizontal), tilt (movimento vertical) e roll (rotação). Além disso, é fundamental para person tracking implementar um controle de posicionamento que receba as informações de rastreamento do objeto e movimente o gimbal de forma a manter a pessoa rastreada centralizada no quadro. Essa centralização pode ser feita de algumas maneiras, uma possível seria a extração do centróide da Bounding box da pessoa e centralizar esse ponto no quadro da câmera.

Uma das abordagens mais promissoras para a tarefa de rastreamento robusto de pessoas é a utilização do YOLOv8, um modelo de última geração que se baseia no sucesso das versões anteriores do YOLO (You Only Look Once). O YOLOv8 apresenta novos recursos e melhorias significativas, visando aumentar ainda mais o desempenho e a flexibilidade do modelo.

O YOLOv8 foi projetado com foco em três principais atributos: velocidade, precisão e facilidade de uso. Essa combinação torna-o uma excelente escolha para uma ampla gama de tarefas, incluindo detecção e rastreamento de objetos, segmentação de instâncias, classificação de imagens e estimativa de pose.

A integração da câmera com os servos do gimbal oferece uma solução robusta e versátil para o problema de person tracking, possibilitando a captura de dados precisos em tempo real e abrindo caminho para aplicações em áreas como segurança, monitoramento e interação humano-máquina. Ao longo deste artigo, serão abordadas as metodologias utilizadas para processamento das informações provenientes dessas tecnologias, bem como sua aplicação prática na tarefa de rastreamento personalizado.

Seção III. Metodologia

Para a utilização do nosso gimbal juntamente com o YOLOv8, optamos por criar um ambiente de desenvolvimento usando o Robot Operating System (ROS) em conjunto com containers Docker. Essa combinação nos proporciona um ambiente de testes ideal para experimentar nosso robô e aprimorar o rastreamento por meio de simulações e ajustes nas configurações.

Dentro do ambiente Docker, temos a flexibilidade de criar e executar os nós ROS necessários, simplificando a configuração das interações entre o

gimbal e o YOLOv8. Isso nos permite realizar testes controlados e iterações eficientes no desenvolvimento.

Para iniciar, é necessário instalar o Docker em seu sistema. O Docker é uma plataforma que permite criar e gerenciar contêineres, proporcionando um ambiente isolado para nossos desenvolvimentos. A instalação pode ser feita através do site oficial do Docker [4] e configure para que esse tenha acesso a GPUs dentro de containers.

Uma vez que o Docker esteja instalado, o próximo passo é configurar o ambiente ROS [5]. O ROS oferece uma estrutura robusta para o desenvolvimento de sistemas robóticos, permitindo a criação de nós que podem se comunicar e executar tarefas específicas.

Ao utilizar essa configuração, conseguimos uma abordagem mais eficaz para a colaboração entre o gimbal e o YOLOv8. A combinação do ROS e dos containers Docker nos proporciona um ambiente ágil e flexível, permitindo a otimização de nossos algoritmos de rastreamento e a melhoria contínua de nosso sistema robótico.

Após a preparação dos nossos ambientes, chegou a hora de instalar o YOLOv8. Para simplificar esse processo, aproveitamos o Ultralytics [3], uma ferramenta que nos permite instalar facilmente os pacotes necessários para acessar o YOLOv8. Tudo o que precisamos fazer é executar o comando 'pip install ultralytics'. Com nossa biblioteca pronta, podemos usar os comandos listados na documentação para começar a detecção de objetos.

O ambiente ROS se mostra especialmente valioso para conduzir essas simulações. Isso se deve ao fato de que não precisamos usar nossos próprios vídeos ou fotos de teste. Em vez disso, podemos explorar os objetos e pessoas presentes no próprio ambiente do ROS, como simulações no gazebo. Isso nos proporciona um ambiente de teste realista e controlado para aprimorar nossa solução de detecção e rastreamento.

Dentre as configurações do YOLOv8 foi utilizado o tracking bytetrack, sendo o que mais se adequou às nossas necessidades. O bytracking é um algoritmo que rastreia objetos ao longo do tempo, usando uma série de previsões feitas pelo modelo. Ele é capaz de identificar objetos que estão se movendo rapidamente ou que estão sendo obscurecidos por outros objetos. O bytracking também é capaz de rastrear objetos que estão entrando e saindo da área de visão da câmera.

A partir desse ponto, para chegarmos em um detector de quedas, tivemos que utilizar um modelo pré treinado, e o escolhido foi o YOLO-NAS[6], onde temos um modelo que otimiza a relação precisão-latência, que será utilizado para detecção de quedas. Para treinamento, inicializamos o modelo de acordo com seu tutorial [7] e com ele já pré-treinado com o dataset coco. A partir disso, utilizamos o nosso dataset de quedas [8] para treinar nosso modelo e conseguir classificar as quedas. O treino foi de 10 épocas, feito em uma GPU Tesla 4, totalizando aproximadamente 3 horas e 30 minutos. Os parâmetros utilizados estão disponibilizados no github[9].

Seção IV. Resultados e Conclusões

Após todas as simulações, gravamos vídeos documentando o resultado satisfatório do nosso sistema realizando o person tracking: [vídeos](#). Como podemos observar no vídeo, tivemos sucesso na nossa integração do detector e tracker, onde nossa câmera consegue seguir o alvo designado e acompanhá-lo sem se perder.

Possibilitamos assim resolver algumas das tasks da RoboCup@Home com sucesso. Temos agora nossa câmera RealSense acoplada ao gimbal que consegue fazer o person tracker no alvo designado. Além de concluir essas tasks conseguimos também implementar junto a isso nosso detector de quedas: [vídeos](#).

Obtivemos métricas não muito boas porém satisfatórias nos testes práticos realizados. Sendo 61% de acurácia. Porém esse número foi devido principalmente ao baixo poder computacional disponível e já termos obtido o resultado esperado para realizar a aplicação.

Observamos que a aplicação implementada em nosso projeto destacou-se de tremenda importância, uma vez que a detecção precoce de quedas não só serve como um meio de prevenção, mas também como um suporte essencial para garantir a segurança e o bem-estar desses indivíduos mais vulneráveis.

No entanto, também poderíamos aplicar nossa metodologia em uma variedade de outros contextos:

Em cenários de segurança, o rastreamento auxilia na detecção de comportamentos suspeitos e monitoramento de espaços públicos.

Já na análise de multidões, é utilizado para entender o fluxo de pessoas em eventos ou locais movimentados.

Em veículos autônomos ou drones, o rastreamento permite seguir alvos em movimento e evitar colisões.

Nos campos de jogos e realidade aumentada, é fundamental a interação com elementos virtuais no mundo real, e em aplicações de monitoramento de saúde ou esportes, possibilita a análise de postura e movimentos específicos.

3 - Referências bibliográficas:

- [1] ROOKSBY, John et al. Personal tracking as lived informatics. In: **Proceedings of the SIGCHI conference on human factors in computing systems**. 2014. p. 1163-1172.
- [2] FUNG, Ng Ming et al. Elderly fall detection and location tracking system using heterogeneous wireless networks. In: **2019 IEEE 9th Symposium on Computer Applications & Industrial Electronics (ISCAIE)**. IEEE, 2019. p. 44-49.
- [3] ULTRALYTICS. *Documentação Oficial*. Disponível em: <https://docs.ultralytics.com/>. Acesso em: 03 ago. 2023, às 22:15.
- [4] DOCKER. Site oficial. Disponível em: <https://www.docker.com/>. Acesso em: 17 ago. 2023, às 00:05.
- [5] ROS. Site oficial. Disponível em: <https://www.ros.org/>. Acesso em: 17 ago. 2023, às 00:06.
- [6] YOLO-NAS. Site oficial. Disponível em: <https://docs.ultralytics.com/models/yolo-nas/>. Acesso em: 18 ago. 2023, às 16:36
- [7] YOLO-NAS. Github. Disponível em: <https://github.com/AarohiSingla/YOLO-NAS>. Acesso em: 18 de ago. 2023, às 18:25.
- [8] Bogdan Kwolek, Michal Kepski, Human fall detection on embedded platform using depth maps and wireless accelerometer, Computer Methods and Programs in Biomedicine, Volume 117, Issue 3, December 2014, Pages 489-501, ISSN 0169-2607

[9]Ferro Filho, Alexandre. Integração de Docker e ROS 2 Foxy para Realização de Person Tracking com Câmera RealSense Acoplada em Gimbal. Disponível

em:

https://github.com/alexandreacff/a03-visao-computacional/tree/main/fall_detection. Acesso em: 18 de agosto de 2023.