

Controlador de Smart Home

Bernardo Freitas – 79295 – a79295@ualg.pt

Tomás Anastácio – 79301 – a79301@ualg.pt

Relatório Técnico – LESTI – Computação Visual

Resumo

Este relatório documenta o desenvolvimento de uma aplicação de deteção de gestos e objetos baseada em Python, integrada ao Blender para controle remoto de animações, elementos gráficos e dispositivos simulados. A solução proposta utiliza MediaPipe para rastreamento de mãos e dedos, técnicas de deteção de objetos para identificar elementos no ambiente e sockets para comunicação entre sistemas. A programação orientada a objetos foi aplicada para garantir modularidade e manutenção. Entre os resultados obtidos, destacam-se uma interface funcional para captura de gestos e objetos, bem como controle interativo no Blender. As diferenças entre o planeamento inicial e a execução real estão detalhadas no documento.

Introdução

O controle de interfaces gráficas através de gestos e a deteção automática de objetos representam áreas crescentes de pesquisa em computação visual, com aplicações em design, animações e experiências interativas. Este projeto teve como objetivo principal desenvolver uma aplicação que permite controlar elementos gráficos no Blender utilizando gestos capturados por câmara e deteção de objetos no ambiente.

A solução combina tecnologias como MediaPipe, para rastreamento de mãos, e algoritmos de deteção de objetos, como parte de uma abordagem integrada para controle mais robusto e versátil de dispositivos.

Os objetivos específicos incluíram:

- Implementar a deteção de gestos com MediaPipe.
- Integrar deteção de objetos para complementar o controle.
- Estabelecer comunicação bidirecional entre o sistema principal e o Blender.
- Modularizar o código para facilitar a manutenção e extensibilidade.

Materiais e Métodos

Tecnologias Utilizadas

- Python: Linguagem principal para desenvolvimento.
- MediaPipe: Biblioteca para rastreamento de mãos.
- OpenCV: Biblioteca de visão computacional para processamento de imagens.
- ObjectDetection: Classe personalizada para deteção de objetos.
- Sockets: Para comunicação em rede.
- Blender: Para visualização e controle de animações.

Arquitetura do Sistema

O projeto foi dividido em quatro componentes principais:

1. Detecção de Gestos: Implementada na classe `GestureDetector`, que utiliza `MediaPipe` para identificar gestos de mãos e dedos.
2. Detecção de Objetos: A classe `ObjectDetection` emprega técnicas de visão computacional para identificar objetos, como "controle remoto" ou "copo".
3. Controle de Dispositivos: A classe `DeviceControlMenu` gerencia o estado dos dispositivos controlados por gestos e objetos detectados.
4. Servidor Socket: A classe `SocketServer` implementa a comunicação entre o sistema principal e o Blender.

Resultados e Discussão

A solução final permitiu controlar os seguintes aspectos no Blender e dispositivos simulados:

- Luzes: Ativar e desativar luzes com um dedo levantado e reconhecimento de objetos como "controle remoto".
- Animações: Iniciar animações através de comandos enviados via socket.
- Dispositivos específicos: Comandos ativados apenas na presença de objetos reconhecidos, como "copo" ou "garrafa".

Diferenças entre Planeado e Desenvolvido

Originalmente, o planejamento previa apenas a detecção de gestos. A introdução da detecção de objetos ampliou o escopo do projeto, permitindo comandos condicionais baseados na combinação de gestos e objetos presentes no ambiente. Além disso, novos gestos foram introduzidos durante o desenvolvimento para aumentar a funcionalidade.

Conclusão

O projeto demonstrou a viabilidade do controle remoto de aplicações gráficas usando gestos e detecção de objetos, com resultados positivos em termos de precisão e responsividade. A integração entre diferentes tecnologias, como `MediaPipe` e detecção de objetos, mostrou-se eficiente para criar uma interface natural e intuitiva.

Como trabalho futuro, sugere-se:

- Melhorar a precisão da detecção de objetos em ambientes com maior complexidade.
- Expandir o suporte a novos dispositivos e gestos personalizados.
- Implementar feedback háptico ou visual para auxiliar o usuário na interação.

Referências

- **Mediapipe Documentation:** <https://google.github.io/mediapipe/>
- **Blender Python API:** <https://docs.blender.org/api/current/>
- **Python Socket Programming Documentation:** <https://docs.python.org/3/library/socket.html>
- **OpenCV Documentation:** <https://docs.opencv.org>