

Controlo do Cursor do Rato

Aplicação desenvolvida em Python
utilizando a biblioteca OpenCV

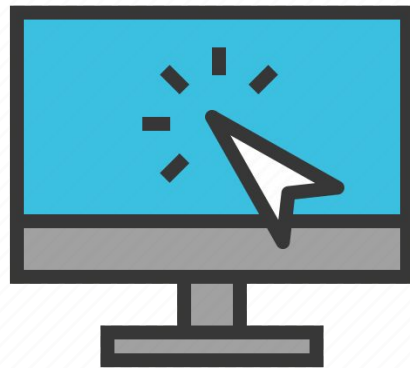


Márcia Pires (88747)
Tomás Martins (89286)

Computação Visual
Janeiro de 2021

Introdução

Aplicação para **detetar e seguir**, em tempo real ou através de um vídeo pré-gravado, um dado **objeto**, controlando assim os **deslocamentos do cursor no ecrã do computador**.
Objetos com **cores diferentes** permitem realizar **diferentes ações** sobre ficheiros, pastas ou aplicações.



Funcionalidades

Principais funcionalidades que podem ser encontradas na aplicação:

```
----- APPLICATION MENU -----  
1 - Livestream your image  
2 - Record your own video  
3 - Show video  
4 - Calibrate your colors  
5 - Exit program
```

Fig. 1: Opções do menu inicial

```
Do you want to blur your image?  
1 - Yes  
2 - No  
> 1
```

Fig. 2: Opção de adicionar *blur*

Funcionalidades

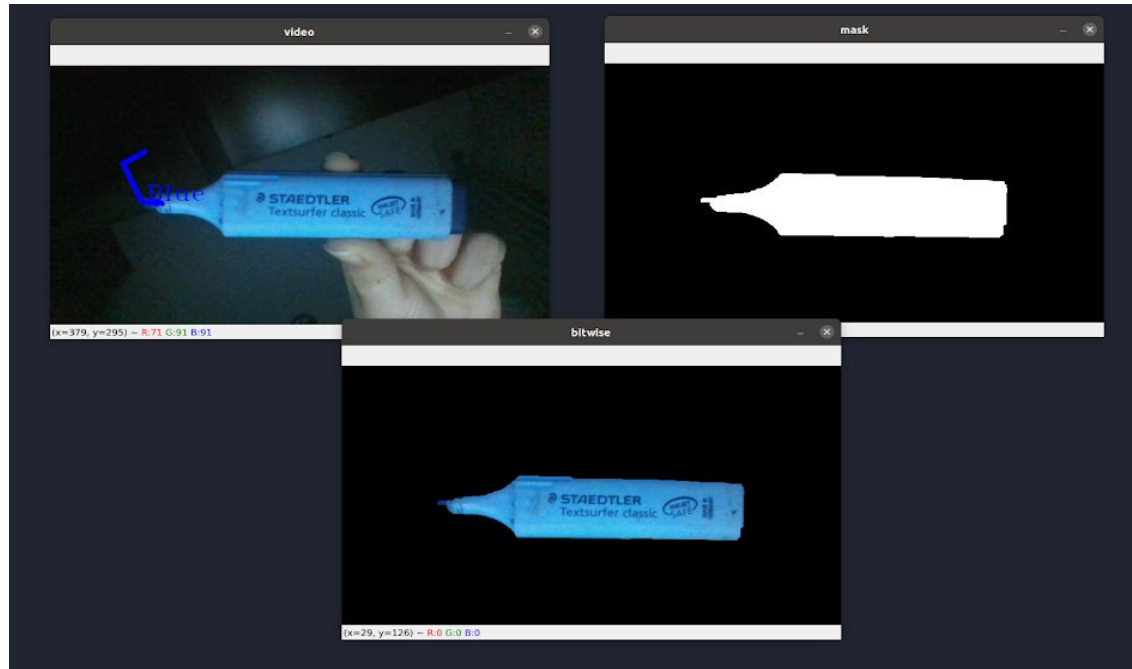


Fig. 3: Exemplo de detecção do objeto e respetiva cor

Funcionalidades

Ação associada a cada **cor do objeto**:

- **Azul** - Controlo da posição do cursor
- **Amarelo** - Scroll para cima
- **Laranja** - Scroll para baixo
- **Verde** - Clique com o lado esquerdo do cursor
- **Verde + Amarelo** - Clique com o lado direito do cursor
- **Verde + Laranja** - Mantém premido o lado esquerdo do cursor



Fig. 4: Cores utilizadas para manipular o sistema

Implementação

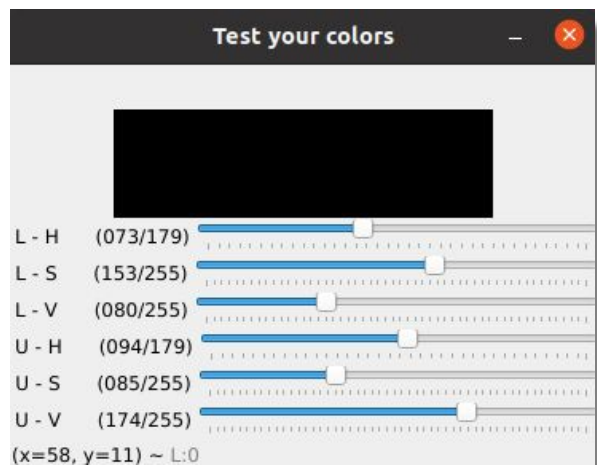


Fig. 5: Trackbar para calibração das cores

- Inicialmente foi realizada uma **calibração das cores**.
- Para implementar **blur** foi utilizado o método **Gaussian Blur**.
- A **máscara** é calculada através da função **cv2.inRange()** e em seguida, através das **operações morfológicas Open e Close**, são corrigidas falhas e ruído.
- Para verificar se determinada cor existe, é efetuada a **soma** de todos os elementos da matriz resultante da aplicação da máscara respetiva.

Implementação

- Cada cor é submetida a um processo de **busca de contornos**, que a identifica com um **rectângulo** e com o **nome da cor**.
- O **movimento do rato** é calculado ao encontrar o centro do rectângulo azul e utilizando uma regra de 3 simples entre o tamanho da janela e do ecrã do computador.
- Array que guarda as consecutivas posições do cursor, utilizadas para desenhar o rasto do movimento através da função **cv2.line()**.



Fig. 6: Rasto do movimento do rato.

Conclusão



Ao longo do projeto fomos nos deparando com **algumas dificuldades**, entre elas:

- Conseguir encontrar o **valor das cores** correto para os nossos objetos, evitando interferências do ambiente de trabalho (cenário e iluminação);
- **Falta de objetos** de diferentes cores para cada operação.

Ainda assim, apesar de haver espaço para melhorar, acreditamos que o resultado final é **bastante positivo** face aos objetivos que tínhamos estabelecido, baseado numa **boa aplicação dos conhecimentos obtidos** nas aulas de Computação Visual.