

Change Detector

Francisco Petronilho 89241

Rodrigo Santos 89180

Turma P2 – Computação Visual

Resumo - O presente relatório apresenta o desenvolvimento de uma aplicação que permita visualizar e detetar mudanças significativas numa sequência de vídeo obtido de uma câmara, em tempo real labirinto utilizando a biblioteca *OpenCV* de *Python*.

O relatório começa por introduzir os objetivos da aplicação, de seguida, são apresentados os módulos que a constituem e a sua organização. Depois, são descritas as funcionalidades para familiarizar o leitor com a aplicação e detalhes da sua implementação e do seu funcionamento. Finalmente, são apresentadas limitações, dificuldades encontradas e conclusões.

Abstract - This report presents the development of an application that allows to visualize and detect significant changes in video sequence obtained through a camera, in real time utilizing *Python* library *OpenCV*.

The report starts by introducing the objectives of the application, following, the modules that constitute it and their organization. Then, the features are described to familiarize the reader with the application and details of its implementation. Finally, conclusions are presented and some detected limitations.

I. INTRODUÇÃO

Neste relatório iremos explicar as principais funcionalidades do projeto “Change Detector” desenvolvido no âmbito da cadeira de Computação Visual do 4º ano de Mestrado Integrado em Engenharia de Computadores e Telemática.

A ideia foi apresentada pelos professores da cadeira e consistia no desenvolvimento de um algoritmo de deteção de alterações significativas de uma sequência de vídeo obtido através de uma câmara fixa, este devia permitir ao utilizador visualizar tais alterações, sinalizando as em tempo real no vídeo.

A partir desta ideia mais algumas operações foram adicionadas tais como a de aplicar funções para fazer a deteção de mudanças significativas para níveis de iluminação diferentes num mesmo *background*.

II. DESCRIÇÃO DA ORGANIZAÇÃO E CONTEÚDO DOS FICHEIROS

O projeto é composto por apenas um ficheiro de código necessário para o correto funcionamento do mesmo, nesta secção apresentamos brevemente o conteúdo deste ficheiro.

- **changeDetector.py:** Ficheiro único contendo o código, *Python* utilizando a biblioteca *OpenCV*, necessário para o correto funcionamento do projeto e como tal detetar e apresentar em tempo real as mudanças significativas numa sequência de vídeo.

III. FUNCIONALIDADES

Nesta secção são apresentados todas as funcionalidades que o utilizador tem á disposição durante a execução do projeto.

A. Visualização de sequência de vídeo em vários formatos

Como foi proposto, foi implementada a funcionalidade de visualizar em tempo real uma sequência de vídeo sendo que esta é apresentada de 4 modos diferentes cada um apresentado numa janela diferente.

- **Main Frame:** Janela principal, onde são apresentadas e sinalizadas as mudanças mais significativas na sequência de vídeo.
- **Delta Frame:** Nesta janela é possível verificar a diferença entre o *frame* atual da sequência de vídeo e a imagem calculada do *background*.
- **Threshold Frame:** Nesta janela, aparecem as diferenças representadas de um modo mais fortemente sinalizado, em que qualquer diferença encontrada é representada com um pixel branco puro num fundo preto.

- **Main Frame Gray Scale:** Nesta janela a sequência de vídeo encontra-se a preto e branco, com base nestes *frames* é tornada mais fácil a verificação de alterações significativas tendo em conta a alteração de luminosidade.

B. Visualização das mudanças mais significativas

De encontro aos objetivos do projeto, foi implementada uma funcionalidade, que desenha em tempo real na janela **Main Frame**, sequência de vídeo retângulos que sinalizam os objetos novos que aparecem no vídeo. Tal como é ilustrado na imagem seguinte.



Figura 1. Sinalização de mudanças

C. Visualização do Background definido em tempo real

Para facilitar ao utilizador verificar qual o background atual de comparação com a sequência de vídeo na janela **Background Gray Scale** é apresentado o background atual em Gray Scale, sendo este atualizado periodicamente. Na imagem a seguir é mostrado, um exemplo de background.

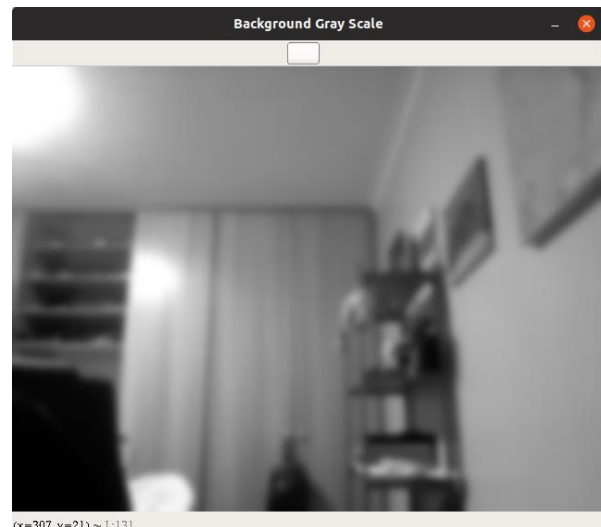


Figura 2. Janela Background Gray Scale.

D. Reset do background definido em tempo real

Para que o utilizador possa testar e definir um *background* correto para o programa, foi implementada uma funcionalidade que define o *background* de comparação, ao premir a tecla “R” do teclado do computador o utilizador define como *background* o *frame* atual da sequência de vídeo. Estas alterações são refletidas imediatamente nas janelas de análise de vídeo (**Delta Frame** e **Threshold Frame**).

E. Saída do programa

Após terminada verificação, para que o utilizador pare corretamente a execução do programa, foi criado o atalho de fecho e interrupção da captura de vídeo, que o utilizador pode executar premindo a tecla “Q” do teclado do computador.

IV. DETALHES DE IMPLEMENTAÇÃO E FUNCIONAMENTO DA APLICAÇÃO

Nesta secção são apresentados alguns detalhes importantes da implementação efetuada e do funcionamento da aplicação.

A. Construção e atualização do background

A construção do background é feita no início do programa, nos primeiros segundos de execução, este é gerado a partir do cálculo da mediana de vários *frames* consecutivos. *Frames* são colocados em grupos de 5 em 3 filas (*Queues*) diferentes. Quando uma das filas enche é feito o cálculo da sua mediana, esta mediana é depois introduzida na fila seguinte. E o processo é repetido até as

3 filas estarem cheias. Após as 3 filas estarem cheias é feito o ultimo cálculo da mediana, sendo o resultado definido como background da sequência de vídeo. Para manter o background atualizado, este processo vai se repetindo ao longo da execução do programa, tornando assim mais fácil a detecção de diferenças mais significativas. O background é também convertido para tons de cinzento o que facilita a exclusão de mudanças derivadas a iluminação.

Para tornar mais fácil a mudança do background para um background mais preciso de acordo com o que o utilizador pretende, foi implementada a funcionalidade de *reset* do background na qual o background é recalculado imediatamente com o *frame* atual da sequência de vídeo.

B. Detecção de mudanças significativas

Para detetar as mudanças mais significativas na sequência de vídeo, o *frame* atual é recebido e é transformado numa imagem em tons de cinzento ao resultado desta operação é de seguida aplicado um filtro de *Blur* Gaussiano fazendo uso da função **GaussianBlur** de OpenCV, depois é feito o cálculo da diferença absoluta através da função de OpenCV **absdiff** desse *frame* modificado com o background já calculado anteriormente, esta diferença encontra se expressa na janela **Delta Frame**.

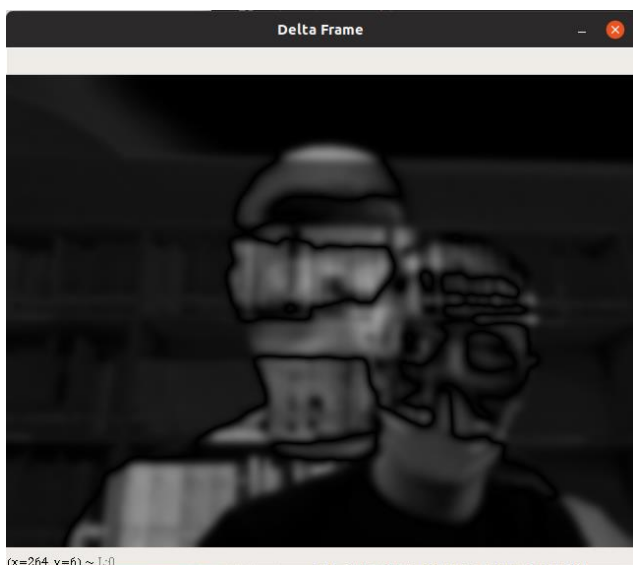


Figura 3. Delta Frame

Uma vez calculada a diferença absoluta esta é transformada numa imagem de puro preto e branco, através da função de OpenCV **threshold**, para facilitar a detecção de contornos na sequência de vídeo. Esta transformação é visível na janela **Threshold Frame**.

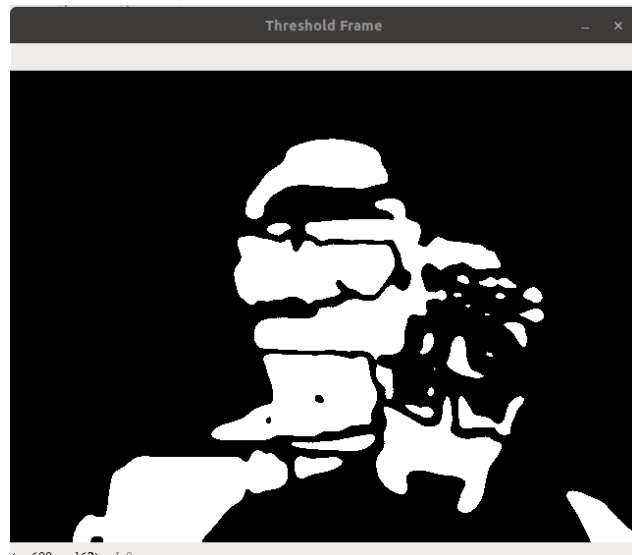


Figura 4. Threshold Frame

Utilizando a função **findContours** de OpenCV são detetados todos os contornos (espaços brancos) de diferença na sequência de vídeo. De seguida para todos os contornos encontrados é feito o cálculo da sua área, e caso a área seja superior a 5000 é considerado como mudança significativa e é desenhado na janela **Main Frame** um retângulo contendo a mudança e sinalizando-a, tal como é apresentado na **Figura 1**. Também é alterado o estado de detecção presente no canto superior esquerdo da janela **Main Frame** para “Status: Changes detected” visível também como exemplo na **Figura 1**.

V. LIMITAÇÕES E DIFICULDADES ENCONTRADAS

Nesta secção serão expostas algumas observações acerca da implementação e utilização da aplicação bem como algumas dificuldades encontradas.

Na utilização da aplicação é possível verificar que algumas mudanças na cena não são detetados isto deve-se ao valor de área escolhida como valor de comparação, uma vez que se for escolhido um valor muito baixo, qualquer mudança na cena seria sinalizada como uma mudança significativa, mesmo se apenas se tratasse de uma mudança derivada, por exemplo, á mudança de iluminação. Por outro lado caso fosse escolhido um valor demasiada grande, poderíamos ter alterações já significativas que não seriam sinalizadas como tal, foi com estas limitações em mente que escolhemos um valor de área que achamos adequado, no entanto é possível ainda assim observar algumas irregularidades. Outro fator limitativo na implementação do nosso projeto é numero de *frames* necessários para construir um background, uma vez que caso seja escolhido um valor muito pequeno de *frames* pode resultar numa representação incorreta do background, no entanto se o numero for demasiado elevado será mais difícil manter o background atualizado com as mudanças menos significativas que possam ocorrer na cena.

VI. CONTRIBUIÇÃO DE CADA MEMBRO

A contribuição de cada membro do grupo, em percentagem, foi a seguinte:

- Francisco Petronilho 89241: 50 %
- Rodrigo Santos 89180: 50 %

Todo o trabalho necessário para a conclusão do projeto foi desenvolvido por ambos os membros do grupo com participação repartida de igual forma entre os dois membros.

VII. CÓDIGO EXTERNO UTILIZADO

O desenvolvimento deste simulador teve por base o conhecimento adquirido nas aulas, por isso foi utilizado código fornecido através dos guiões práticos. Houve pequenas funções retiradas ou derivadas da internet, particularmente no cálculo da área e na deteção de contornos numa imagem pura preto e branco[1].

VIII. CONCLUSÕES

A realização deste projeto permitiu utilizar conhecimentos adquiridos durante as aulas práticas e teóricas e exigiu explorar mais acerca de outras componentes que permitissem atingir os objetivos do projeto.

Os conteúdos presentes neste projeto, foram alguns dos que foram lecionados nas aulas: a leitura e carregamento de imagens através de um vídeo, aplicação de filtros; nomeadamente a aplicação de GaussianBlur; transformações de thresholding, a criação e adição de figuras, utilizadas neste caso para o contorno de mudanças significativas na cena, assim com a utilização do teclado para execução de certas funcionalidades.

Para futuro trabalho seria interessante desenvolver mais o aspeto de deteção de mudanças melhorando o conceito de mudança significativa. Também seria interessante desenvolver o aspeto de segurança no qual estas mudanças seriam contadas e analisadas para fazer, por exemplo, contagem de pessoas.

Tendo em conta o trabalho realizado e o proposto, achamos que atingimos fortemente o objetivo proposto, o de criar um detetor de mudanças significativas numa sequência de vídeo.

REFERÊNCIAS

[1] 2020 Area_de_objetos - [2020 Area de objetos.ipynb](#)
- [Colaboratory \(google.com\)](#)