

# Live Video Paint

Diogo Marques 83631 Marco Silva 84770

**Abstract** - Within the subject of Visual Computation (CV), our group developed a self proposed project (Live Video Paint), using the OpenCV library. The main feature of the project is allowing the user the ability to draw on a computer window through an object of a specific color detected by the webcam. This report first approaches the implementation and functionalities of our theme. In the final we discuss some future work we would like to do and a final conclusion.

## I. INTRODUÇÃO

Neste relatório consta as funcionalidades, forma de implementação e informações relevantes do projeto em OpenCV desenvolvido na unidade curricular Computação Visual, no 4º ano do Mestrado Integrado de Engenharia de Computadores e Telemática.

A ideia do nosso projeto surgiu da vontade de criar uma aplicação de desenho onde não fosse necessário a interação com nenhum dispositivo físico de entrada como o rato ou teclado. Assim, utilizamos a câmara para obter o vídeo em tempo real.

O objetivo principal da nossa aplicação é fornecer ao utilizador a possibilidade de desenhar numa janela do computador como se de uma folha se tratasse. O utilizador ao ter um objeto de cor amarela consegue interagir com a aplicação como se este fosse um lápis/pincel tendo assim total liberdade criativa. Para além da possibilidade de desenho a aplicação proporciona ao utilizador funcionalidades de forma a tornar a experiência mais intuitiva e simples.

## II. IMPLEMENTAÇÃO

- Numa primeira etapa foi necessário capturar e mostrar o vídeo da webcam do computador do utilizador. Para tal, num ciclo infinito, um frame é guardado numa matriz e mostrado, sendo que o próximo frame é guardado por cima do anterior descartando o mesmo. Optámos ainda por espelhar os frames na vertical de modo a que o vídeo possa replicar os movimentos do utilizador exatamente como este os está a fazer.
- Para que pudesse existir uma interação simples e eficaz com o utilizador optamos por criar um layout que é aplicado por cima dos frames referidos no ponto anterior. Foi então necessário obter a máscara, isto é, uma imagem a preto e branco com a forma do layout, para o poder aplicar

de acordo com a mesma. Na Figura 1 podemos ver a imagem (em formato png e sem fundo) correspondente ao layout antes de o aplicar ao frame.

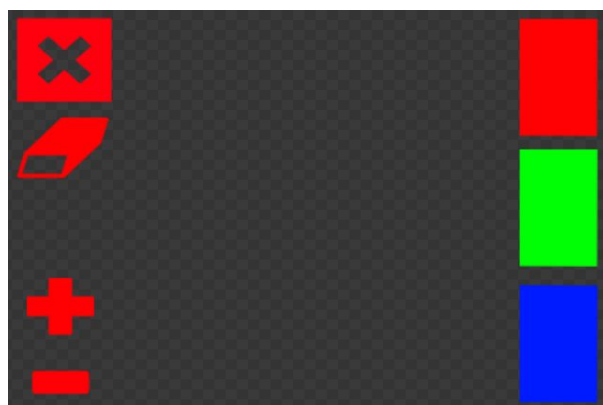


Fig. 1 - Layout.

- Após algum tempo a trabalhar com o vídeo notámos algum ruído no fundo deste. Para resolver este problema optámos então por aplicar um filtro, no caso *medianBlur*, para atenuar este. Desta forma conseguimos também melhorar a deteção de objetos que vai ser falada mais a frente.
- O próximo passo foi então detetar um objeto amarelo na imagem. Escolhemos esta cor por ser uma cor que se destaca bem das restantes. Para o fazer foi necessário converter a matriz para HSV (Hue, Saturation, Value), que inicialmente estava em BGR (Blue, Green, Red), e de seguida aplicar a função *inRange* do OpenCV. Esta devolve uma matriz *Threshold* que, tal como a máscara, é preta e branca no entanto os pontos brancos aqui são apenas os objetos em amarelo (os limites da cor foram estabelecidos por nós) presentes na imagem. Na Figura 2 podemos ver um exemplo.

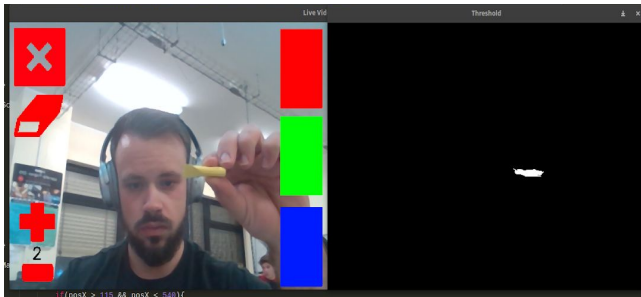


Fig. 2 - Detecção de um objeto amarelo.

- Além de detetar o objeto foi também necessário obter as coordenadas. Como é possível utilizar um objeto de qualquer formato e dimensão optamos por, primeiramente, obter todos os pontos brancos da matriz *threshold* e calcular o ponto médio dos mesmos, utilizando este como a localização do objeto. De notar que com este método se forem detetados vários objetos é calculado o ponto médio entre ambos e não de cada um isoladamente, logo a aplicação não tem capacidade de desenhar em simultâneo.
- Após termos conhecimento da localização do objeto é então necessário verificar a sua posição em relação ao layout apresentado e caso este se encontre sobreposto a uma das áreas de cor, isto é, vermelho, verde ou azul, a cor associada ao objeto é alterada. Da mesma forma, se este se encontrar num dos outros “botões” também a função associada aos mesmos é acionada. Estas funções serão abordadas na secção III - Interação com o Utilizador.
- Para desenhar uma linha em OpenCV são necessários dois pontos diferentes. Uma vez que as coordenadas do objeto são detetadas em cada um dos frames é necessário guardar as coordenadas do frame anterior para possa desenhar a linha.
- Todos os desenhos produzidos pelo utilizador são feitos num painel, inicialmente branco, que aparece lado a lado juntamente com o vídeo da webcam, como é possível ver na Figura 3.

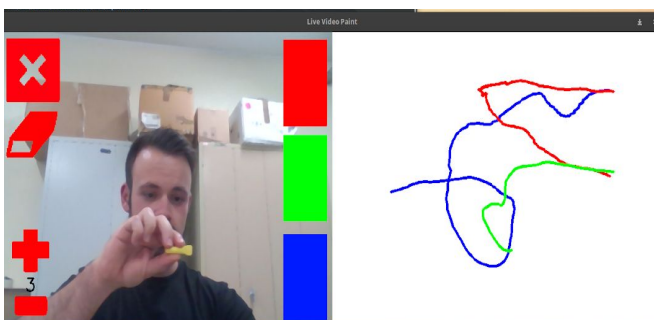


Fig. 3 - Demonstração de desenho.

### III. INTERAÇÃO COM O UTILIZADOR

Sendo o principal objetivo desta aplicação a simplicidade na interação com o utilizador optámos por incluir poucas funcionalidades, que no caso são:

- Trocar a cor da “caneta” para vermelho, verde ou azul;
- Aumentar e diminuir a grossura da linha;
- Limpar o canvas de desenho;
- Fechar a janela.

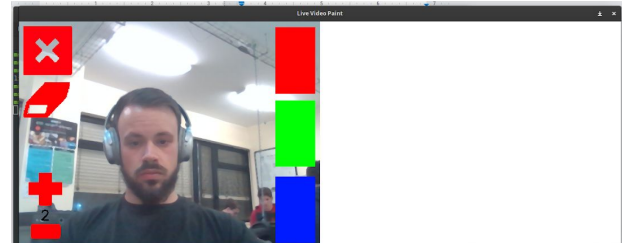


Fig. 4 - Janelas da aplicação.

Começando pela troca de cores: o utilizador tem que colocar o objeto por baixo do retângulo pintado com a cor que deseja e quando o retirar de lá este já vai pintar com a mesma. Nas Figuras 5, 6 e 7 podemos ver a seleção das 3 cores disponíveis.

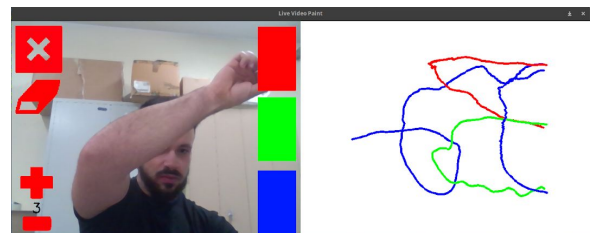


Fig. 5 - Seleção da Cor Vermelha.

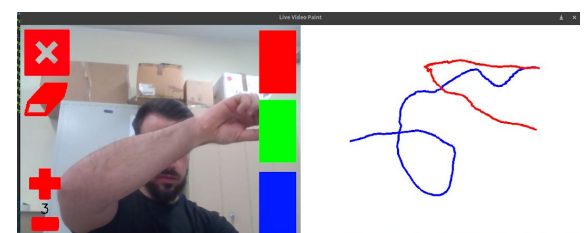


Fig. 6 - Seleção da Cor Verde.

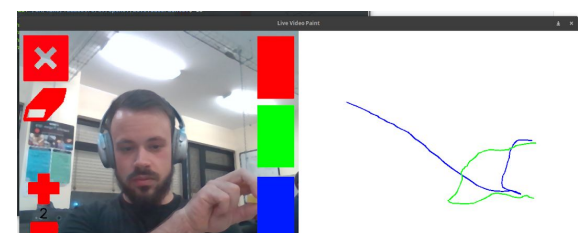


Fig. 7 - Seleção da Cor Azul.

Como referido nos pontos iniciais desta secção o utilizador pode também aumentar e diminuir a grossura da linha, sendo que a grossura inicial é 2. Nas Figuras 8, 9 e 10 é possível ver o utilizador aumentar e diminuir a grossura da linha e desenhar com uma grossura diferente, no caso, 25.

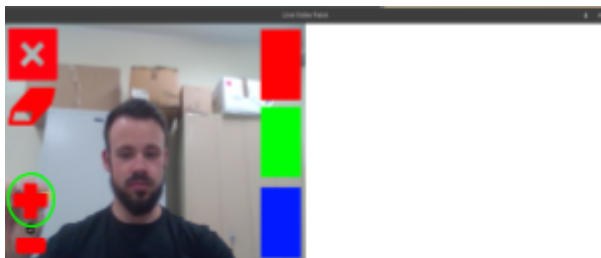


Fig. 8 - Aumentar grossura da linha.

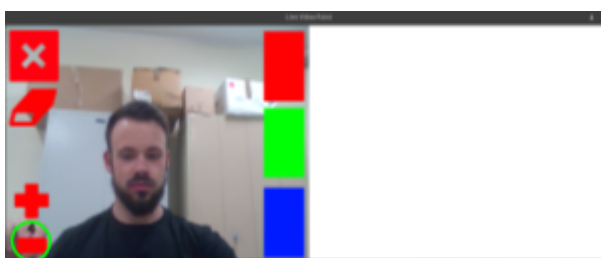


Fig. 9 - Diminuir grossura da linha.

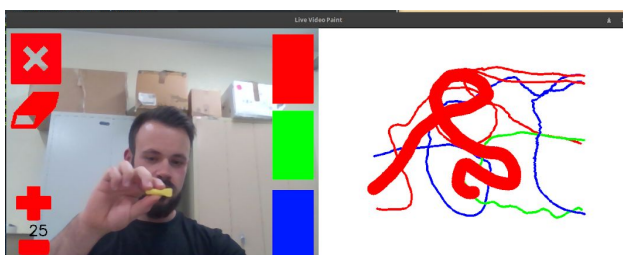


Fig. 10 - Desenhar com grossura 25

Para limpar o canvas de desenho o utilizador deve manter o objeto durante 3 ou mais segundos na área onde se encontra a borracha, como na Figura 11, tendo o resultado na Figura 12. Caso este mova o objeto para fora desta área a contagem é reiniciada.

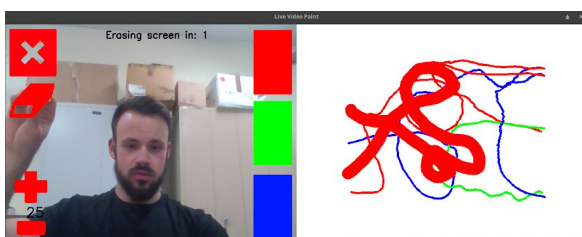


Fig. 11 - Apagar canvas

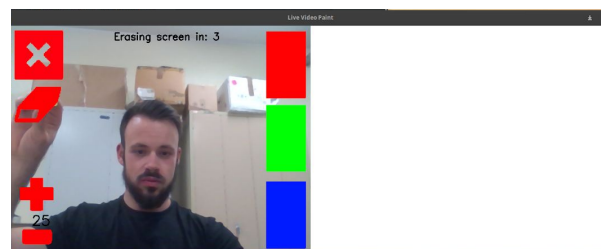


Fig. 12 - Após 3 segundos na borracha

Para que o utilizador não tenha necessidade de interagir com o nenhum dispositivo de entrada no momento de fechar a aplicação, este pode também fechá-la mantendo o objeto na área do “X” durante 3 segundos também, sendo que, da mesma forma que na borracha, esta contagem também recomeça caso o objeto saia desta área. Caso o utilizador abra a aplicação e não tenha nenhum objeto amarelo pode também fechar a mesma pressionando a tecla “ESC”.

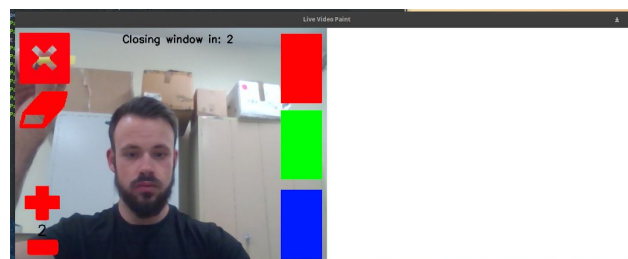


Fig. 13 - Fechar a janela

#### IV. MELHORIAS E TRABALHO FUTURO

No futuro, tencionamos adicionar mais algumas opções à aplicação de forma a tornar o trabalho mais completo:

- Permitir que o utilizador escolha a cor do objeto que está a utilizar como lápis/pincel. Desta forma o utilizador tem uma maior flexibilidade na utilização da aplicação pois não está dependente de um objeto de cor específica. Ter um objeto de cor específica pode restringir ao ambiente onde o utilizador necessita estar.
- Permitir que o utilizador consiga exportar o desenho que realizou para que possa apagar a tela e voltar a desenhar pois o tamanho da tela é limitado.

#### V. CONCLUSÃO

A realização deste trabalho permitiu a integração de alguns conhecimentos obtidos nas aulas, assim como a necessidade de exploração de conceitos de OpenCV. Sendo que durante as aulas abordamos mais o processamento de imagem foi necessário um estudo e trabalho de investigação na área de processamento de vídeo.

A fraca qualidade da imagem que as web-câmeras dos nossos computadores portáteis capturam foi um obstáculo no desenvolvimento do projeto. Apesar desta dificuldade conseguimos proporcionar um bom resultado final que, achamos, corresponder ao que foi pedido inicialmente pelo professor.

## VI. CONTRIBUIÇÃO

A contribuição de cada membro do grupo foi equivalente, ou seja, cada elemento contribui com 50%.

## REFERÊNCIAS

- [1] Filtro Median Blur  
<https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html#medianblur>
- [2] Color detection in OpenCV  
<https://stackoverflow.com/questions/19189482/color-detection-in-opencv>
- [3] Thresholding Operations using inRange  
[https://docs.opencv.org/3.4/da/d97/tutorial\\_threshold\\_inRange.html](https://docs.opencv.org/3.4/da/d97/tutorial_threshold_inRange.html)
- [4] Unidade curricular - Computação Visual  
<http://sweet.ua.pt/jmadeira/OpenCV/>