

LAB 1 - OPERAÇÕES DIÁDICAS E HOMOGRAFIA PLANAR

Observação: Entregar (via Moodle) relatório técnico e códigos produzidos até as 23h59min do dia 09/09.

1 Atividade 1

Na primeira atividade deste laboratório cada aluno deverá implementar o efeito de *Chromakey* para combinar os quadros de dois vídeos de entrada, produzindo um vídeo resultante (conforme ilustrado na Figura 1). Os vídeos de entrada são disponibilizados no Moodle (veja arquivos `Chromakey.mp4` e `Clouds.mp4`). Um exemplo de código que realiza a leitura e gravação de vídeo no Matlab é apresentado na Seção 4 deste roteiro.

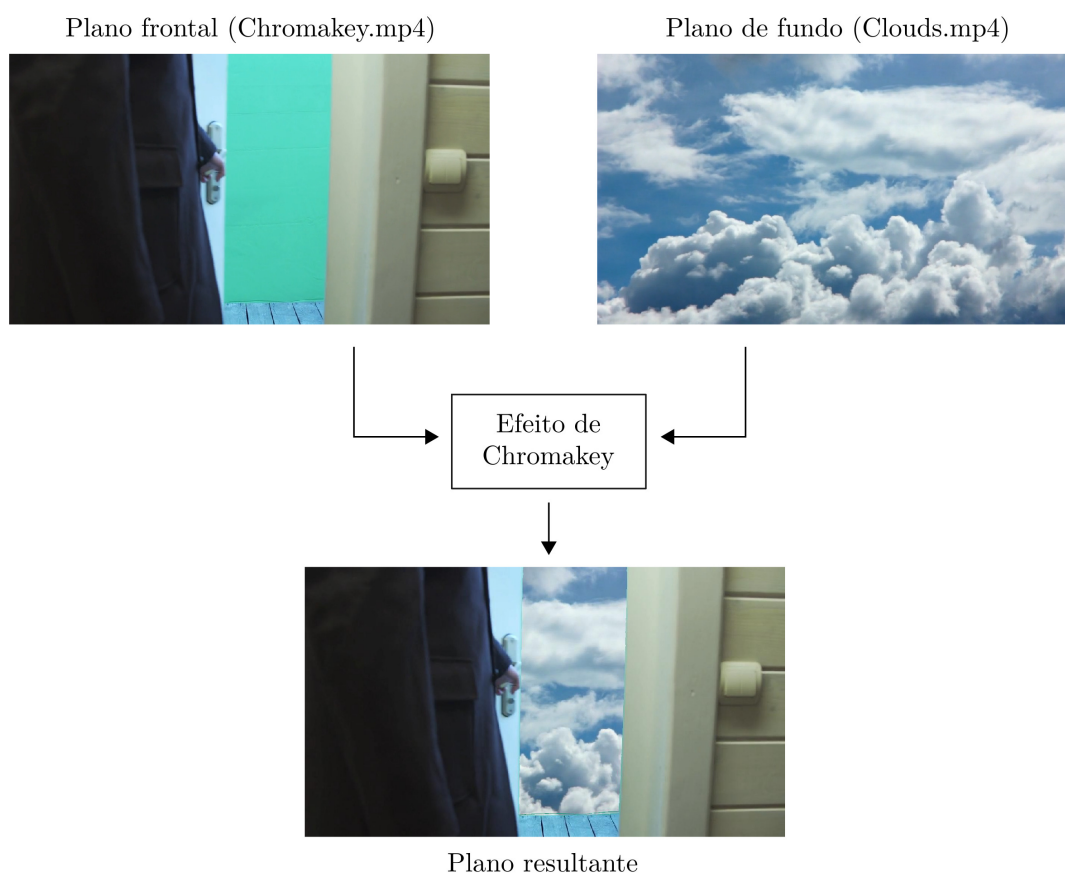


Figura 1: Ilustração do efeito de Chromakey que deve ser implementado neste laboratório.

A implementação do efeito de Chromakey em cada quadro do vídeo deve seguir o procedimento ilustrado na Figura 2 (discutido em aula).

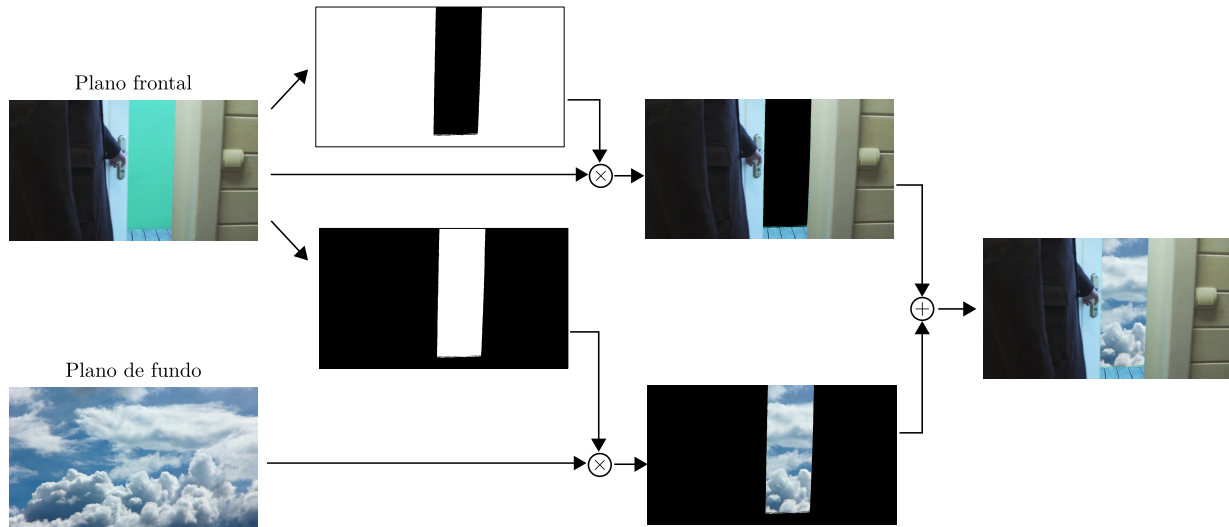


Figura 2: Etapas de processamento para geração do efeito de Chromakey.

É importante ressaltar que nas etapas iniciais de processamento, deve-se aplicar (em cada quadro do plano frontal) um algoritmo de segmentação de imagem baseado em cor. As etapas de processamento de tal algoritmo são apresentadas na Figura 3.

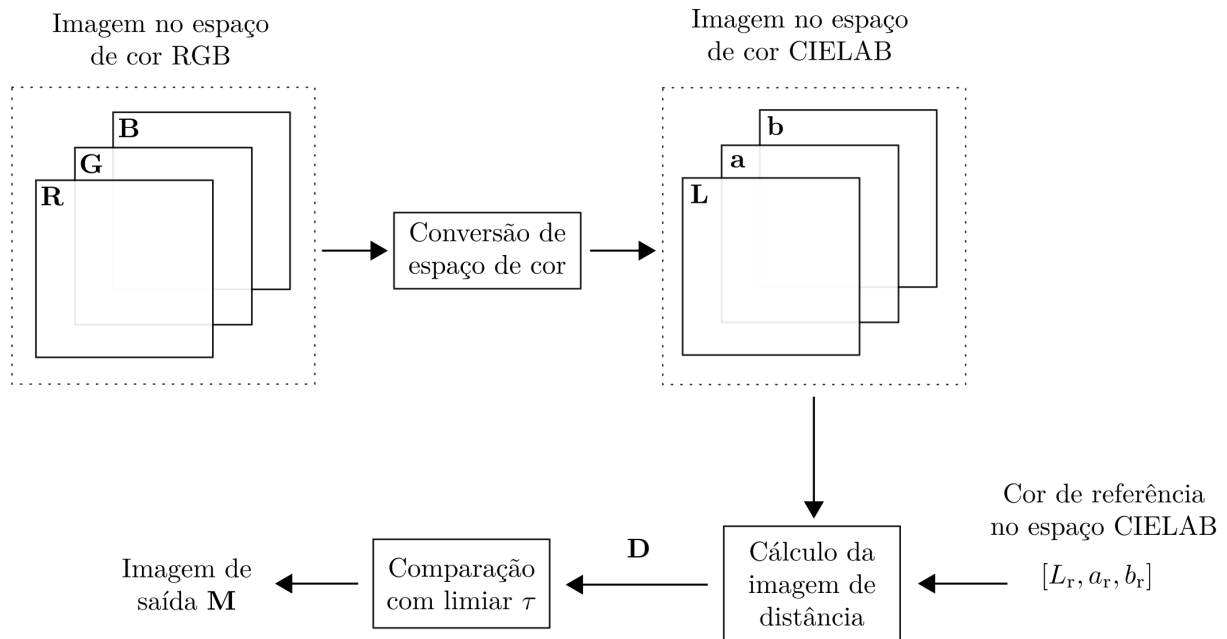


Figura 3: Etapas de processamento do algoritmo de segmentação de imagem baseado em cor.

Observe que na primeira etapa do algoritmo de segmentação de imagem baseado em cor é realizada a conversão da imagem do espaço de cor RGB para o CIELAB (ou ainda, $L^* a^* b^*$).

b^*). No espaço CIELAB, cada cor é representada por três valores, a saber: L^* caracterizando a variação de preto para branco (ou seja, o brilho da imagem), a^* denotando a variação de verde para vermelho e b^* representando a variação de azul para amarelo. O espaço de cor CIELAB foi projetado para que uma mesma quantidade de alteração numérica desses valores corresponderem (aproximadamente) a uma mesma variação na percepção visual da cor. Após a conversão para o espaço de cor CIELAB e tendo em mãos uma cor de referência $[L_r, a_r, b_r]$, obtém-se na sequência a imagem de distância \mathbf{D} . Em particular, cada pixel com coordenada (u, v) da imagem \mathbf{D} é calculada como a distância Euclidiana entre o pixel da imagem no espaço CIELAB e a cor de referência. Ou seja,

$$\mathbf{D}(u, v) = \sqrt{[\mathbf{L}(u, v) - L_r]^2 + [\mathbf{a}(u, v) - a_r]^2 + [\mathbf{b}(u, v) - b_r]^2}. \quad (1)$$

Por fim, para obtenção da imagem binária \mathbf{M} (com regiões em branco denotando as regiões com cores próximas a cor de referência), deve-se realizar a comparação de cada pixel da imagem \mathbf{D} com um limiar τ (com valor definido pelo projetista). Assim,

$$\mathbf{M}(u, v) = \begin{cases} 1, & \mathbf{D}(u, v) \leq \tau \\ 0, & \mathbf{D}(u, v) > \tau. \end{cases} \quad (2)$$

Onde a imagem binária \mathbf{M} deve ser utilizada nas etapas seguintes para obtenção do efeito de Chromakey.

2 Atividade 2

Na segunda atividade deste laboratório, deve-se realizar a projeção de cada quadro do vídeo produzido na atividade anterior no espaço de um outdoor (utilizando a operação de homografia planar), gerando um outro vídeo de saída (conforme ilustrado na Figura 4). A imagem de fundo (com o Outdoor em branco) está disponível na pasta dos arquivos deste laboratório no Moodle.

Imagem de fundo



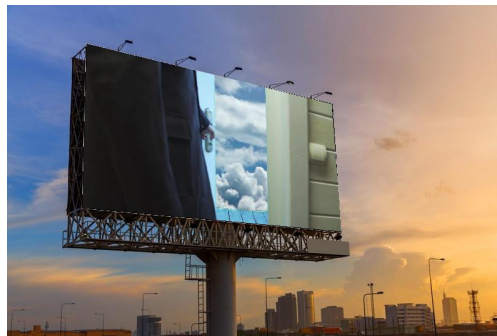
(a)

Vídeo de entrada



(b)

Vídeo de saída



(c)

Figura 4: (a) Imagem de fundo com o outdoor em branco. (b) Quadro do vídeo de entrada. (c) Quadro do vídeo de saída.

3 Orientações para elaboração do relatório técnico

A elaboração do relatório visa formalizar de forma textual as atividades realizadas, para tal é importante apresentar uma descrição do algoritmo desenvolvido e uma discussão acerca dos resultados obtidos. Para este laboratório, o relatório técnico deve conter as seguintes seções e informações:

1. **Identificação:** identificação do aluno e da atividade realizada.
2. **Descrição do problema:** nesta seção deve-se descrever o problema a ser solucionado. Para essa seção, é possível utilizar como base o enunciado do roteiro. Porém é importante não apresentar simplesmente uma cópia do roteiro, ou seja, descreva o problema com as suas palavras.
3. **Descrição do algoritmo proposto:** nesta seção, os principais pontos do algoritmo proposto devem ser descritos de forma detalhada e clara. Sugere-se que a descrição seja

apresentada com o auxílio de diagramas e, também, através da apresentação de resultados parciais. Em outras palavras, evite descrever o algoritmo proposto apenas com texto. Em relação aos códigos, apresente apenas as seções/porções que considerar relevante. Ou seja, não é necessário apresentar uma cópia de todo código no relatório (lembre-se que o código deve ser enviado em arquivos separados).

4. **Resultados:** apresentar e discutir os resultados obtidos.

4 Código exemplo: leitura e criação de arquivo de vídeo

Para fins de referência, no Quadro 1 é apresentado um código exemplo de leitura e gravação de vídeo no Matlab.

Source Code 1: Código para leitura e gravação de arquivos de vídeo no Matlab.

```
1  % carrega arquivo do vídeo de entrada
2  video_entrada = VideoReader('Clouds.mp4');
3
4  % cria arquivo do vídeo de saída
5  video_saida = VideoWriter('video_saida.avi');
6  video_saida.FrameRate = video_entrada.FrameRate;
7  open(video_saida);
8
9  % Processamento
10 while hasFrame(video_entrada)
11
12     % apresenta na tela a informação do instante de tempo
13     % do quadro atual
14     fprintf('time: %f\n', video_entrada.CurrentTime);
15
16     % Obtém quadro (imagem) atual do video de entrada
17     frame_entrada = readFrame(video_entrada, 'native');
18
19     % Processa a imagem (conversão RGB para nível de cinza)
20     frame_saida = rgb2gray(frame_entrada);
21
22     % Escreve imagem processada no vídeo de saída
```

```
23 writeVideo(video_saida,frame_saida);
24 end
25
26 % Finaliza e salva o vídeo de saída
27 close(video_saida);
```