

LAB 1 - ROTEIRO

1 Atividade 1

Na primeira atividade deste laboratório cada dupla de alunos deve implementar um sistema de visão computacional simplificado de contagem de objetos. Especificamente, deve ser desenvolvido um sistema que seja capaz de processar os quadros do vídeo `sequencial.mp4` (fornecido juntamente com este roteiro) e contabilizar quantas bolas amarelas passam no vídeo. Na Figura 1 é apresentado o diagrama contendo as etapas de processamento que devem ser executadas para cada quadro do vídeo. Observe que o sistema é composto de 3 etapas de processamento, a saber: *i*) segmentação baseada em cor, *ii*) extração da linha central da imagem binária e *iii*) detecção de novo objeto.

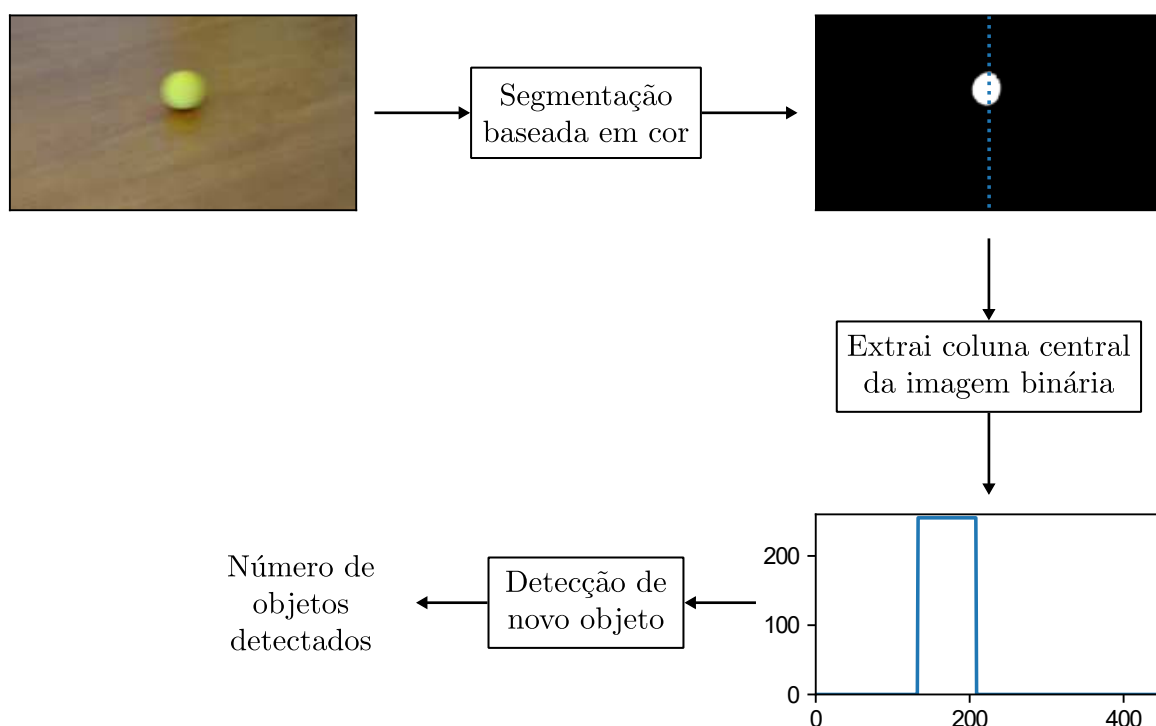


Figura 1: Diagrama com as etapas de processamento do sistema de contagem de objetos.

Na etapa de segmentação baseada em cor, a imagem colorida no espaço de cor BGR é convertida para uma imagem binária, com o objetivo de isolar a região correspondente à bola

amarela. Após a conversão, espera-se que a região da bola seja representada em branco na imagem binária. Após a etapa de segmentação, deve-se extrair a coluna central da imagem binária (conforme ilustrado no diagrama da Figura 1) e analisar a curva obtida visando detectar a passagem de novos objetos.

Especificamente, para a etapa de conversão da imagem colorida para binária, os alunos devem implementar o algoritmo de segmentação descrito no diagrama da Figura 2.

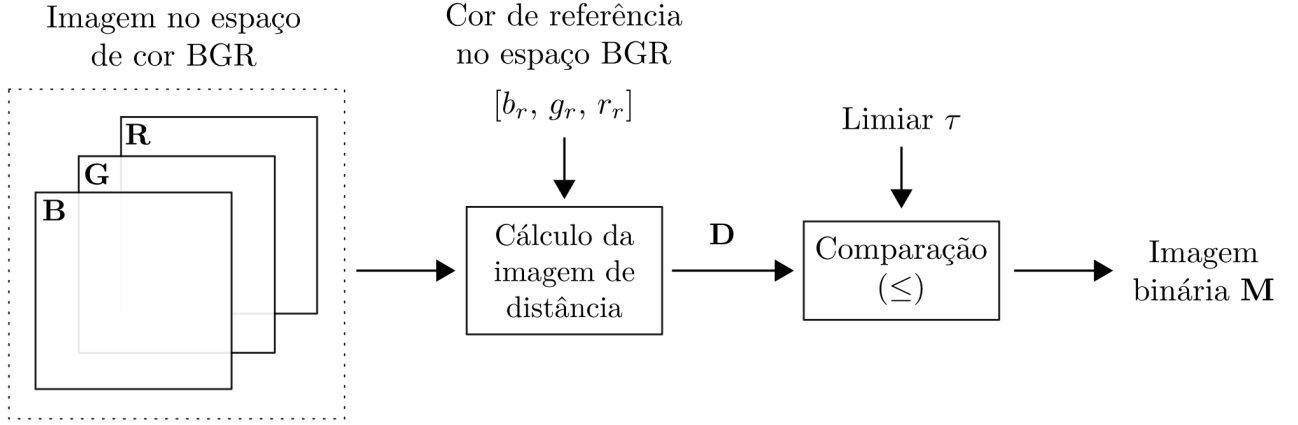


Figura 2: Algoritmo de segmentação baseada em cor e em medida de distância.

No algoritmo de segmentação que deve ser implementado é necessário que o projetista defina de ante mão uma cor de referência (no espaço de cor BGR) com componentes $[b_r, g_r, r_r]$. A partir da imagem de entrada e da cor de referência, obtém-se a imagem de distância \mathbf{D} , onde cada pixel $\mathbf{D}[y, x]$ é dado pela distância euclidiana entre o pixel $\mathbf{I}[y, x]$ (da imagem de entrada) e a cor de referência. Especificamente, tem-se

$$\mathbf{D}[y, x] = \sqrt{(\mathbf{B}[y, x] - b_r)^2 + (\mathbf{G}[y, x] - g_r)^2 + (\mathbf{R}[y, x] - r_r)^2}. \quad (1)$$

onde $\mathbf{B}[y, x]$, $\mathbf{G}[y, x]$ e $\mathbf{R}[y, x]$ denotam, respectivamente, as componentes azul, verde e vermelho do pixel $\mathbf{I}[y, x]$ (da imagem de entrada). Por fim, para obtenção da imagem binária \mathbf{M} , deve-se realizar a comparação de cada pixel da imagem \mathbf{D} com um limiar τ (definido pelo projetista).

Assim,

$$\mathbf{M}[y, x] = \begin{cases} 255, & \text{se } \mathbf{D}[y, x] \leq \tau \\ 0, & \text{caso contrário.} \end{cases} \quad (2)$$

Observe que no algoritmo da Figura 2, os pixels da imagem de entrada \mathbf{I} que tiverem cor

próxima da cor de referência $[b_r, g_r, r_r]$ aparecerão na imagem binária resultante como brancos.

Observação: Para a realização desta atividade utilize o arquivo `lab1_atividade1.py` que acompanha este roteiro. Além disso, produza um vídeo similar ao fornecido pelo professor (utilizando objetos de cor e/ou formas diferentes) e aplique o sistema desenvolvido para contagem de objetos.

2 Atividade 2

Na segunda atividade deste laboratório cada aluno deverá implementar o efeito de *Chromakey* para combinar os quadros de dois vídeos de entrada, produzindo um vídeo resultante. Os vídeos de entrada são disponibilizados no Moodle (veja arquivos `Chromakey.mp4` e `Clouds.mp4`). As etapas de processamento que devem ser implementadas para geração do efeito de chromakey são ilustradas na Figura 3.

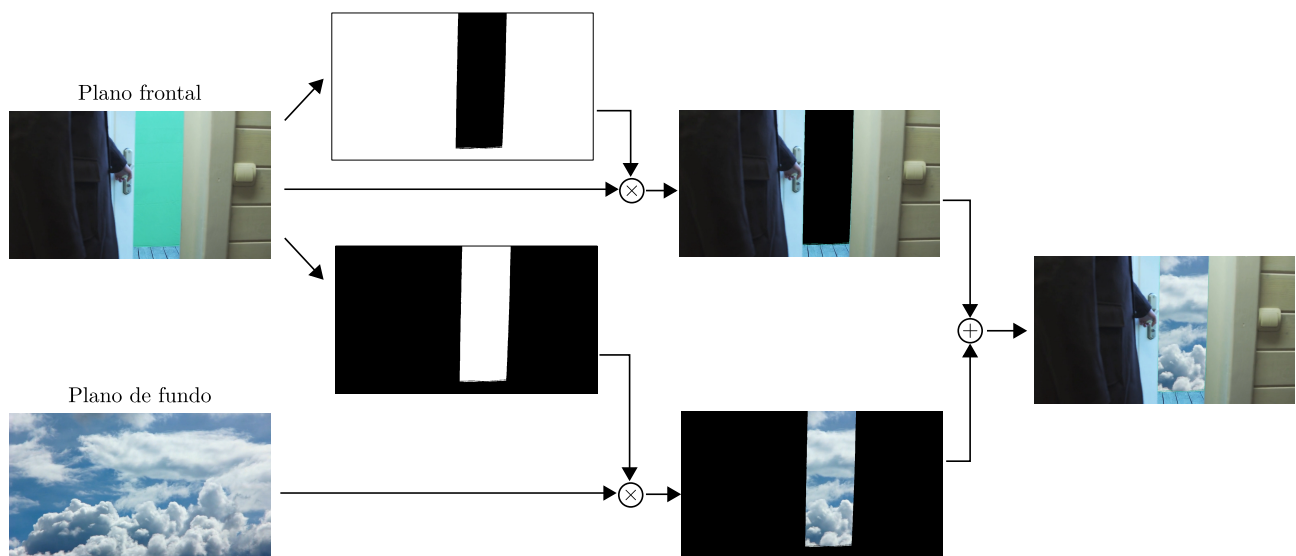


Figura 3: Etapas de processamento para implementação do efeito de Chromakey.

Para geração das máscaras binárias, necessárias para realização do efeito de chromakey, utilize o algoritmo de segmentação baseado em cor da Atividade 1.

Observação: Para a realização desta atividade utilize o arquivo `lab1_atividade2.py` que acompanha este roteiro.

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 clara e formal do algoritmo desenvolvido e uma discussão acerca dos resultados obtidos. É essencial que o relatório seja bem estruturado, com uma sequência lógica e fluida, garantindo que o leitor compreenda tanto a visão geral quanto os detalhes técnicos do trabalho. Para este laboratório, o relatório técnico deve conter as seguintes seções e informações:

1. **Seção 1: Identificação.** Apresentar as informações de identificação dos alunos e da atividade realizada (por exemplo, Laboratório 1 da disciplina CAC3040 - Visão Computacional em Robótica).
2. **Seção 2: Descrição da Atividade 1.** Nesta seção os alunos devem descrever com suas próprias palavras o problema proposto, evitando copiar o enunciado do roteiro.
3. **Seção 2.1: Solução proposta para a Atividade 1.** A seguir, devem apresentar o algoritmo desenvolvido, utilizando uma combinação de texto, diagramas e/ou equações. Recomenda-se começar com uma visão geral do sistema, destacando de forma concisa as principais etapas de processamento, antes de entrar em uma descrição mais detalhada de cada fase. É importante que cada etapa seja bem explicada, e, sempre que possível, resultados intermediários devem ser incluídos para facilitar o entendimento do leitor. Se for utilizado um diagrama para representar o fluxo das etapas de processamento, a figura correspondente deve ser citada logo no início da explicação. Dessa forma, o texto pode se referir ao diagrama de forma integrada, proporcionando ao leitor uma visão mais clara e visual do sistema implementado.
4. **Seção 2.2: Resultados obtidos para a Atividade 1.** Nesta seção, devem ser apresentados, descritos e analisados os resultados obtidos durante o experimento. É importante incluir figuras que ilustrem claramente o desempenho do sistema, como: um quadro do vídeo de entrada, a imagem binária correspondente, a curva referente à coluna central da imagem binária, e os resultados de contagem em diferentes momentos do vídeo.

Cada figura deve ser acompanhada de uma descrição clara, explicando o que está sendo mostrado e como esses resultados se relacionam com o funcionamento do sistema. Ao longo da análise, comente se o sistema desenvolvido foi capaz de realizar a contagem dos objetos de maneira precisa e sem erros.

Se houver resultados intermediários ou dados que ajudem a esclarecer o comportamento do sistema em situações específicas, como em momentos de sobreposição de objetos ou mudanças nas condições do vídeo, estes também devem ser discutidos. Dessa forma, o leitor terá uma compreensão mais completa da eficácia do sistema e das possíveis limitações.

Outra informação importante que deve ser apresentada é o tempo médio de processamento de cada quadro do vídeo. Para isso utilize as funções `getTickCount()` e `getTickFrequency()` do OpenCV. O código implementado é capaz de operar em tempo real? Isto é, o processamento de cada quadro do vídeo é finalizado no intervalo de $1/FPS$ segundo, onde FPS é a taxa de quadros por segundo do vídeo. **ATENÇÃO:** contabilize apenas o tempo de processamento no intervalo que inicia após a aquisição do quadro do vídeo até a contagem do objeto, isto é, exclua da medição de tempo os códigos que são utilizados para apresentação de imagens na tela do computador.

5. **Seção 3: Descrição da Atividade 2.** Similar a Seção 2.
6. **Seção 3.1: Solução proposta para a Atividade 2.** Similar a Seção 2.1.
7. **Seção 3.2: Resultados obtidos para a Atividade 2.** Similar a Seção 2.2, com exceção de que para a Atividade 2 não é necessário computar o tempo médio de processamento de cada quadro do vídeo, tendo em vista que em uma aplicação real o efeito de chromakey não precisa ser executado em tempo real.