UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
Fundamentos de Processamento de Imagens
Prof. Manuel Menezes De Oliveira Neto

Relatório - Trabalho Prático III
Eduardo Veiga Ferreira (578144)

---

1) (20 pontos) Configure o seu ambiente de programação (e.g., Visual Studio, etc.) compile e execute o programa básico disponibilizado no Apêndice A deste documento. - Completo Satisfatoriamente

A parte inicial demanda mais tempo em instalar e configurar a biblioteca OpenCV. Baixada e instalada do link anexado a especificação do trabalho prático, é integrada ao código por meio de um arquivo CMakeList.txt.

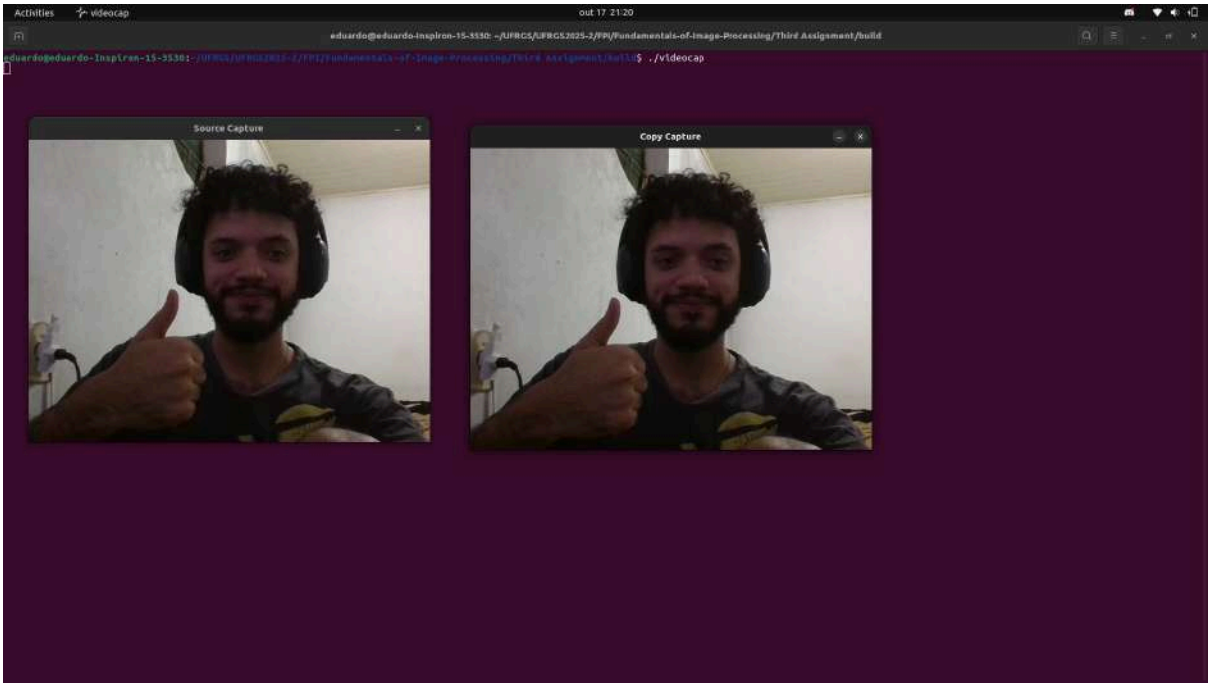


Ilustração 1: Visualização dos vídeos gerados pelo programa inicial, modificado para mostrar duas capturas.

2) (20 pontos) Utilize o comando GaussianBlur para aplicar borramento ao vídeo. Utilize um Trackbar para definir o tamanho do kernel Gaussiano. - Completo Satisfatoriamente

Foi utilizado as funções `createTrackbar` e `GaussianBlur` para realizar o blur em um nível de 0 a 50, modificando o tamanho do kernel pelos valores associados ímpares e positivos. Aqui foi criada a função `int operations(Mat src, Mat cpy, int operation, int size)` que retorna 0 para qualquer tecla que não seja ESC e 1 para qualquer outra tecla que venha representar uma operação, neste caso o switch realiza

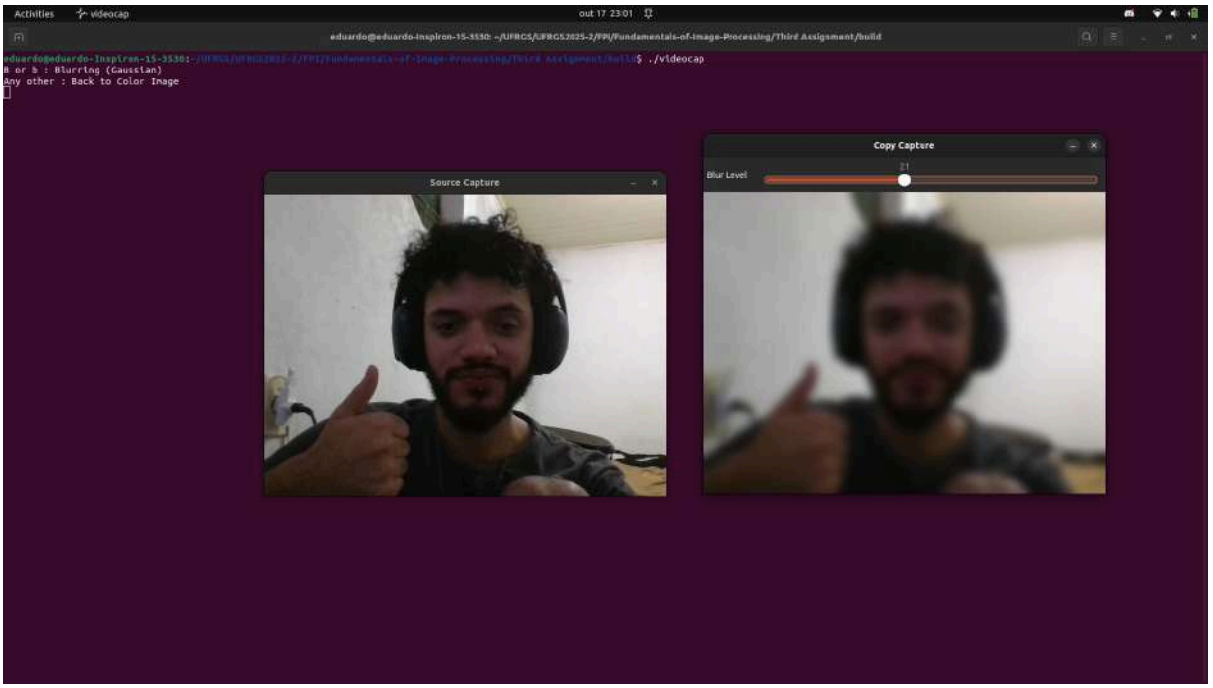


Ilustração 2: Visualização da operação de blur controlada por uma trackbar ao lado do vídeo original.

3) (20 pontos) Utilize o comando Canny para detectar as arestas no vídeo. - Completo Satisfatoriamente

Foi utilizada a função `cvtColor` para fazer as conversões de cor que a operação canny necessita, isto é, transformar os frames para escala de cinza, aplicar a operação Canny, e transformar a imagem novamente em RGB para exibição. Além disso, foram aplicadas algumas lógicas para que as operações sejam cumulativas, e apenas sejam capturadas aquelas teclas que representam alguma operação no programa.

repositório: https://github.com/drdgfv/Fundamentals-of-Image-Processing.git
demonstração: https://youtu.be/d2MIG9FC224

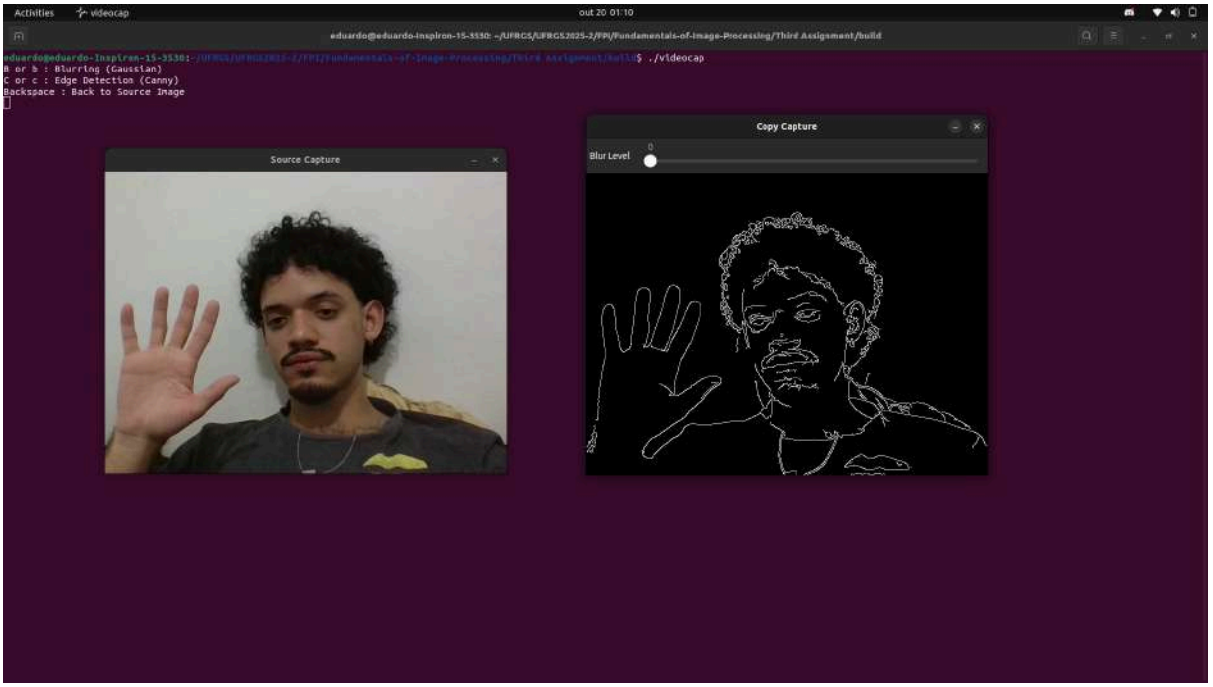
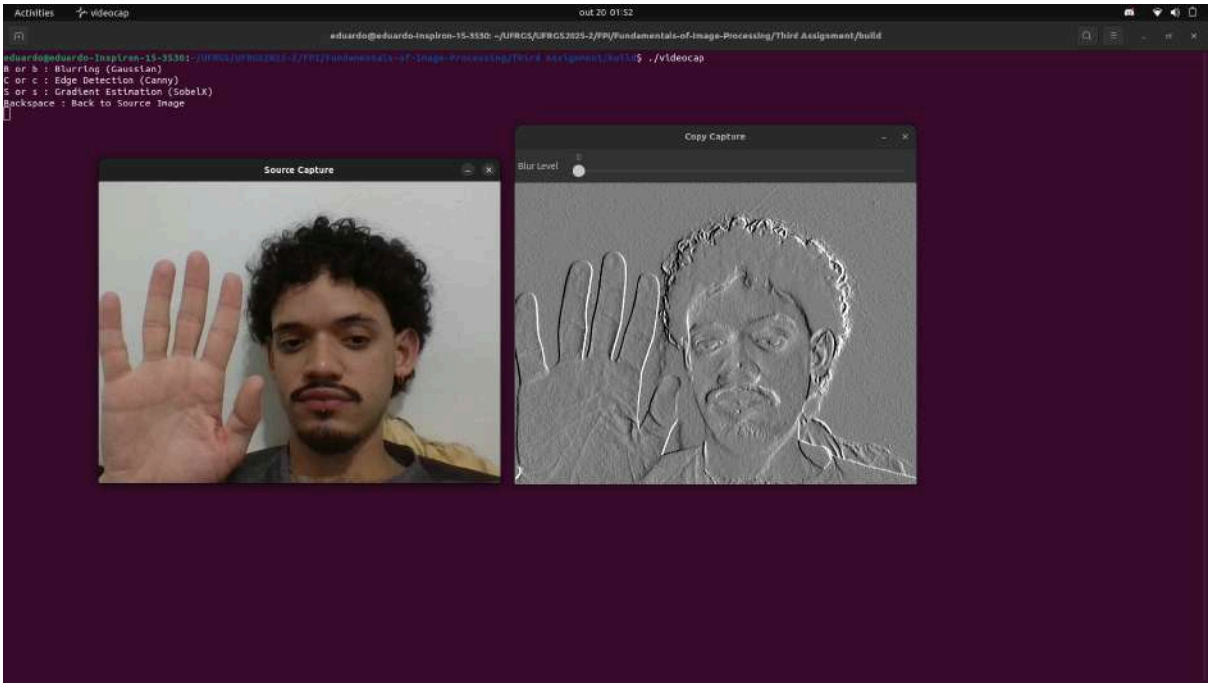


Ilustração 3: Visualização da operação de detecção de arestas ao lado do vídeo original.

4) (20 pontos) Utilize o comando Sobel para obter uma estimativa do gradiente do vídeo. - Completo Satisfatoriamente

Foi utilizada a função `cvtColor` e `convertTo` para fazer as conversões de cor e clamping que a operação Sobel necessita. É realizada a conversão para escala de cinza, e o retorno da luminância para 3 canais, para manter a consistência das operações, em seguida, é realizada a operação sobel e o clamping com `convertTo`. Dessa forma, obtemos os frames com o aspecto de Sobel usual utilizado na disciplina.



repositório: https://github.com/drdgfv/Fundamentals-of-Image-Processing.git
demonstração: https://youtu.be/d2MIG9FC224

Ilustração 4: Visualização da operação de gradientes em tons de cinza e com clamping ao lado do vídeo original.

5) (20 pontos) Utilize o comando convertTo para realizar ajuste de brilho, ajuste de contraste, e obter o negativo do vídeo. - Completo Satisfatoriamente

Aqui foi realizado um ajuste inicial para criar mais de uma trackbar na janela da cópia, de possa controlar a intensidade não só do blur, mas de outras operações que podem ser niveladas como brilho e contraste. Além disso, foi implementado as operações utilizando `convertTo` como base, pois esta função já utiliza a fórmula padrão de operações pontuais lineares vista na disciplina g(x,y) = alpha * f(x,y) + beta.

OBS: Os ajustes de brilho e contraste podem ser nas duas direções, diminuir ou aumentar. Parte-se sempre de um valor mínimo, que diminui.

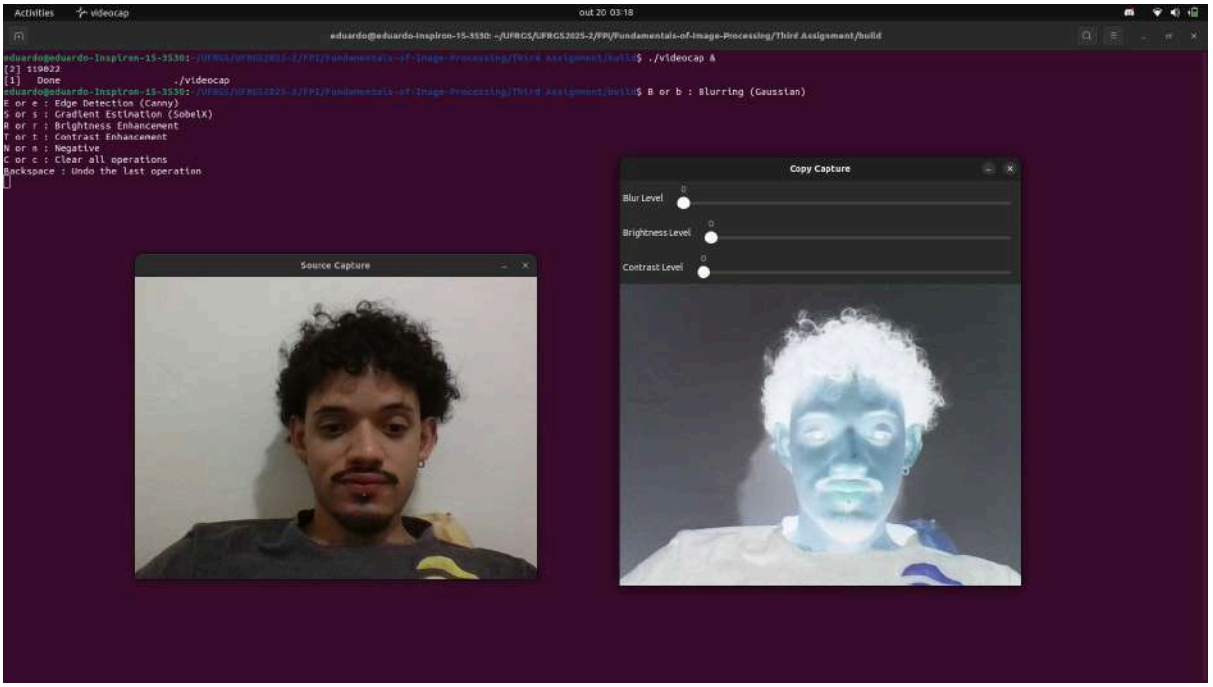


Ilustração 5.1: Visualização da operação de negativo ao lado do vídeo original.

repositório: https://github.com/drdgfv/Fundamentals-of-Image-Processing.git
demonstração: https://youtu.be/d2MIG9FC224

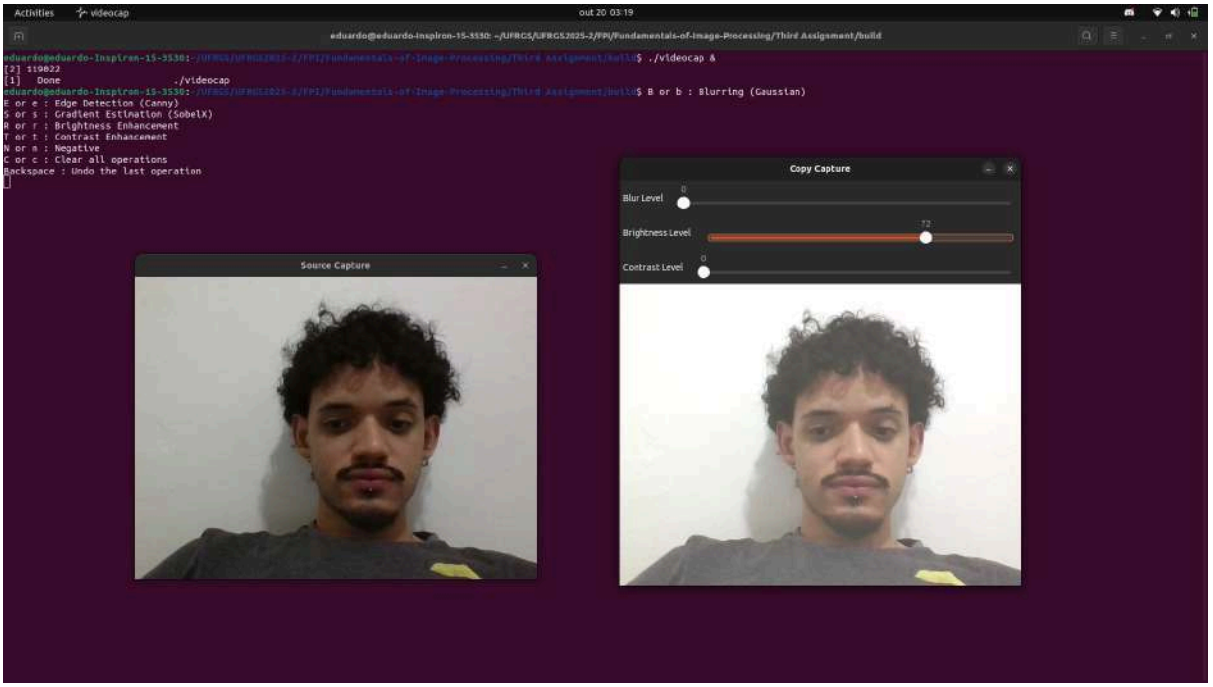


Ilustração 5.2: Visualização da operação de ajuste de brilho ao lado do vídeo original.

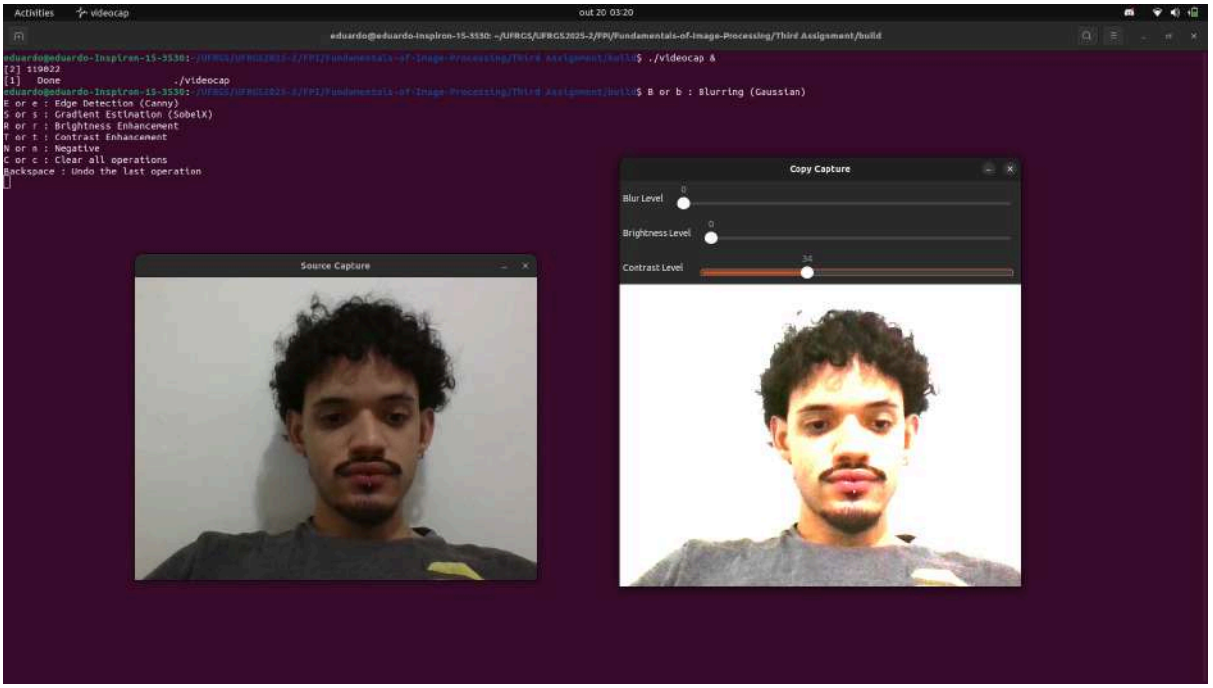


Ilustração 5.3: Visualização da operação de ajuste de contraste ao lado do vídeo original.

6) (10 pontos) Conversão de cores (RGB) para tons de cinza (grayscale). - Completo Satisfatoriamente

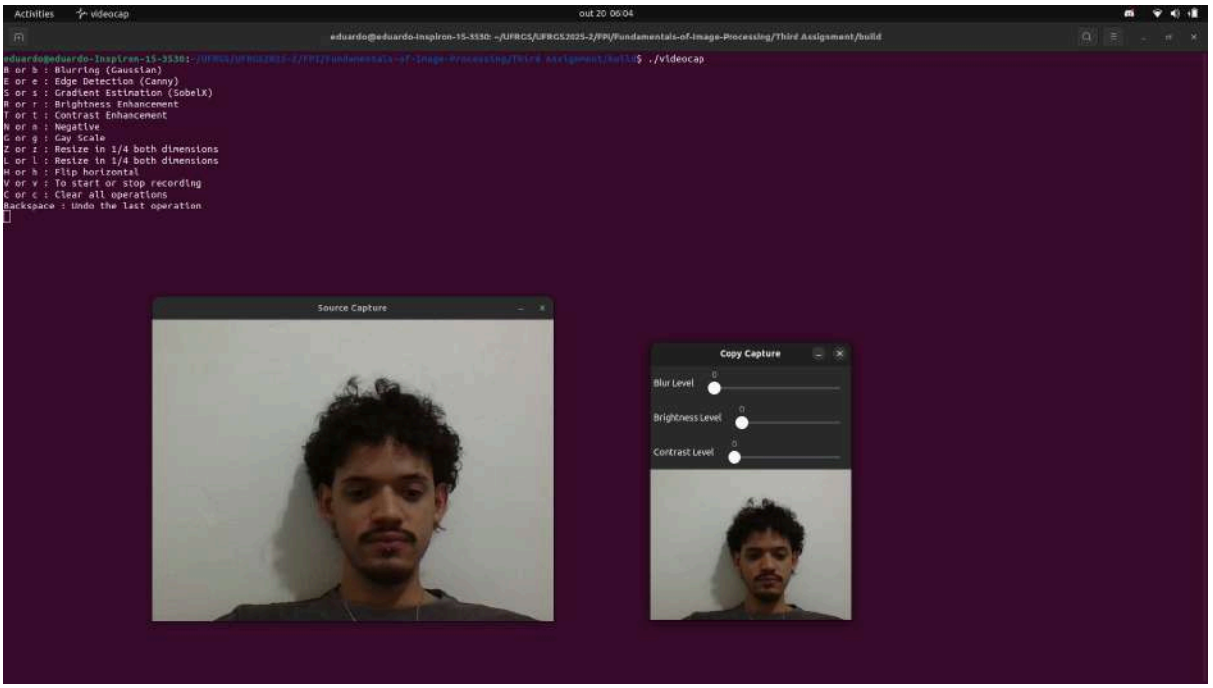
Foi utilizada a função `cvtColor` para transformar os canais da forma apropriada.

repositório: https://github.com/drdgfv/Fundamentals-of-Image-Processing.git
demonstração: https://youtu.be/d2MIG9FC224

Ilustração 6 : Visualização da operação de luminância ao lado do vídeo original.

7) (20 pontos) Redimensionamento do vídeo para a metade do número de píxels em cada uma de suas dimensões. - Completo Satisfatoriamente

Utilizada a função `resize` para redimensionamento de metade das dimensões do vídeo cópia. Os redimensionamentos podem tomar números arbitrários



repositório: https://github.com/drdgfv/Fundamentals-of-Image-Processing.git
demonstração: https://youtu.be/d2MIG9FC224

Ilustração 7: Visualização da operação de redimensionamento por 0,5 do vídeo cópia ao lado do vídeo original.

8) (20 pontos) Rotação do vídeo de 90 graus.

Utilizada a função `rotate` para rotação em 90 graus do vídeo cópia. As rotações podem tomar números arbitrários

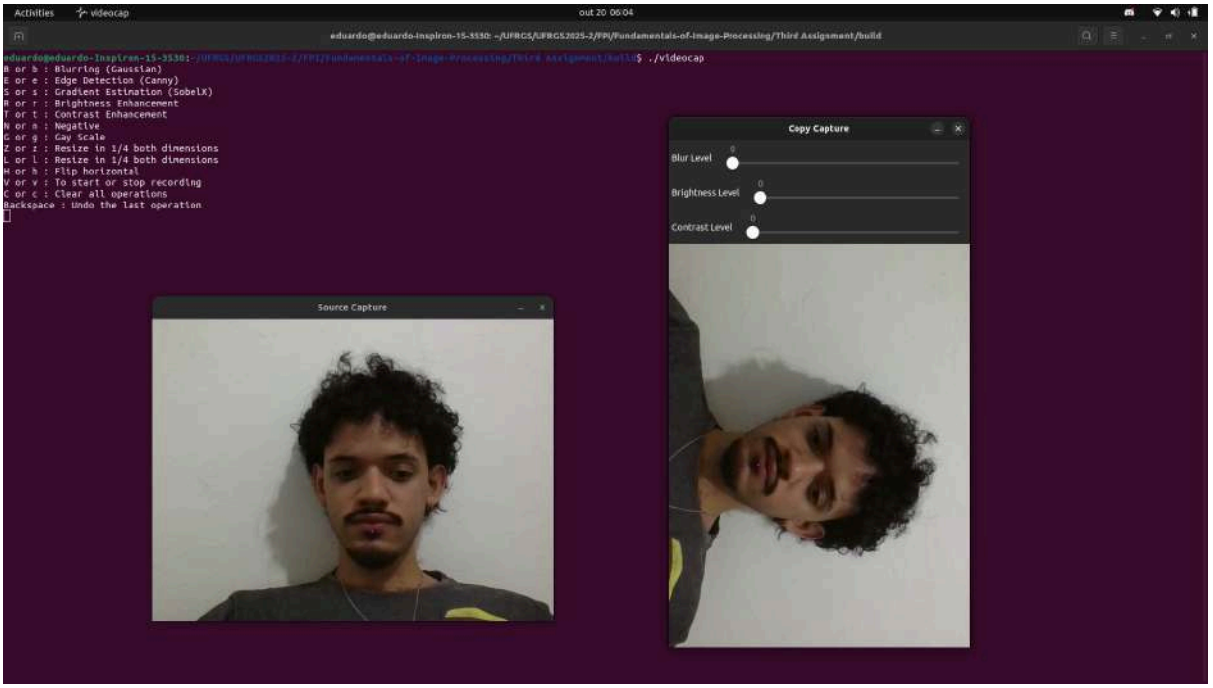


Ilustração 8: Visualização da operação de rotação em 90 graus do vídeo cópia ao lado do vídeo original.

9) (20 pontos) Espelhamento do vídeo (horizontal e vertical).

Utilizada a função `flip` para espelhamento horizontal e vertical do vídeo cópia. Os espelhamentos podem tomar números arbitrários.

repositório: https://github.com/drdgfv/Fundamentals-of-Image-Processing.git
demonstração: https://youtu.be/d2MIG9FC224

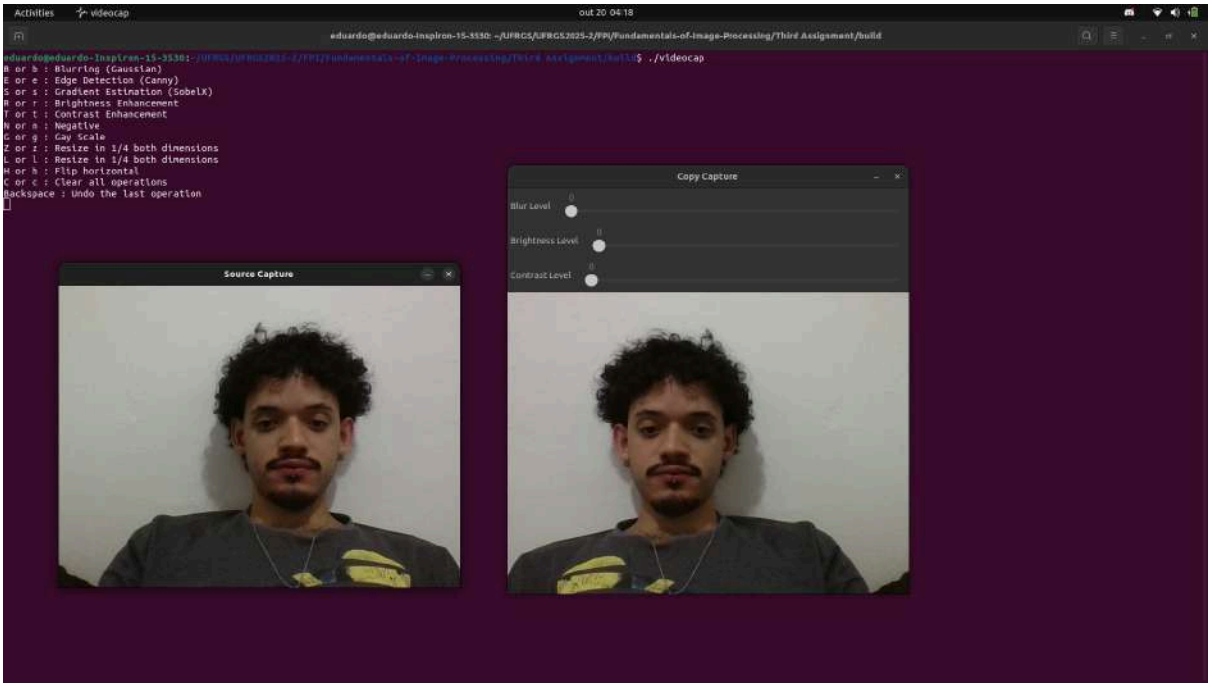


Ilustração 9.1: Visualização da operação de espelhamento horizontal do vídeo cópia ao lado do vídeo original.

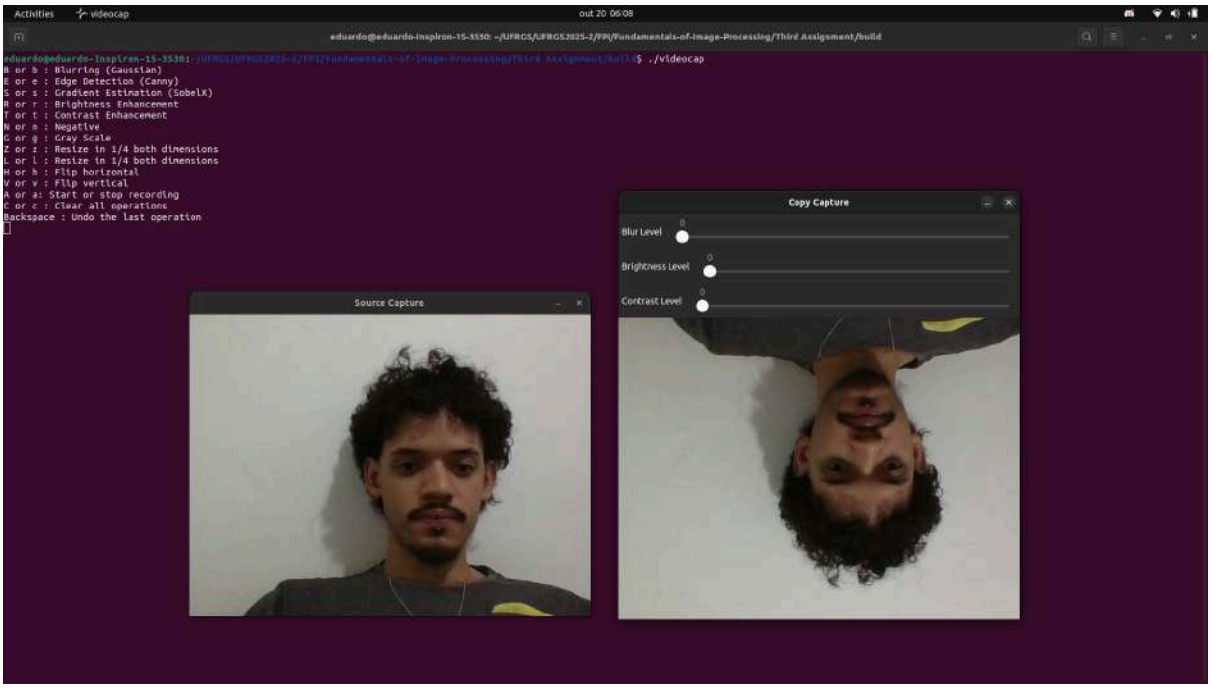


Ilustração 9.2: Visualização da operação de espelhamento vertical do vídeo cópia ao lado do vídeo original.

10) (30 pontos) Gravação de vídeo, levando em conta todos os efeitos acima, exceto Rotação e Redimensionamento.

Aqui foi necessário entender um pouco melhor as estruturas em openCV responsáveis pela gravação/ escrita de um vídeo, quais os parâmetros estava recebendo e como manipular essas informações. Em um primeiro momento foi um pouco difícil, entender qual codec mais

apropriado, a captura que eu devia enviar, e por isso tive alguns problemas com essas especificações, mas logo foram resolvidas. Além dessas dificuldades, a sincronização do fps com a gravação foi outro desafio, por algum motivo que ainda não resolvi completamente, o vídeo gravado fica acelerado em relação a captura, mesmo que a captura dos frames esteja atrasada em relação ao frames por segundo sendo gravados. Mas então, por fim, o vídeo é gravado com o codec motion-jpg e salvo em um .avi na pasta principal da atividade. ***Link do video gravado: https://youtu.be/d2MIG9FC224***

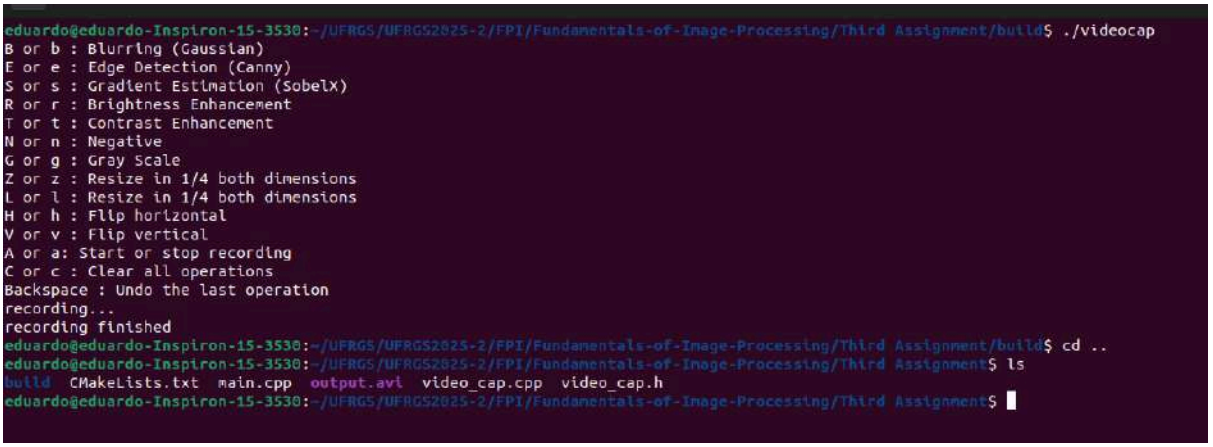


Ilustração 10: Visualização da gravação do vídeo via terminal.