INF01046 – Fundamentos de Processamento de Imagens
Prof. Manuel M. Oliveira
Aluno Gabriel de Souza Seibel

3º Trabalho de Implementação

Manipulação de Vídeos com OpenCV

Introdução

O trabalho cujo desenvolvimento é relatado neste documento teve como objetivo a familiarização com processamento de vídeos em tempo real utilizando a biblioteca OpenCV. Para tanto, foram implementadas funcionalidades dos trabalhos de implementação anteriores com a API simples e prática da biblioteca, que é amplamente utilizada tanto no meio comercial quanto no acadêmico.

O programa desenvolvido é baseado na captura, manipulação e gravação de quadros de imagens de *webcam*. O projeto, apelidado de *OVP - OpenCV Video Processing*, está disponível em *https://github.com/gabrielseibel1/OVP*, e foi implementado na linguagem C++, utilizando OpenCV versão 3.4.3.

Estrutura Básica do Programa

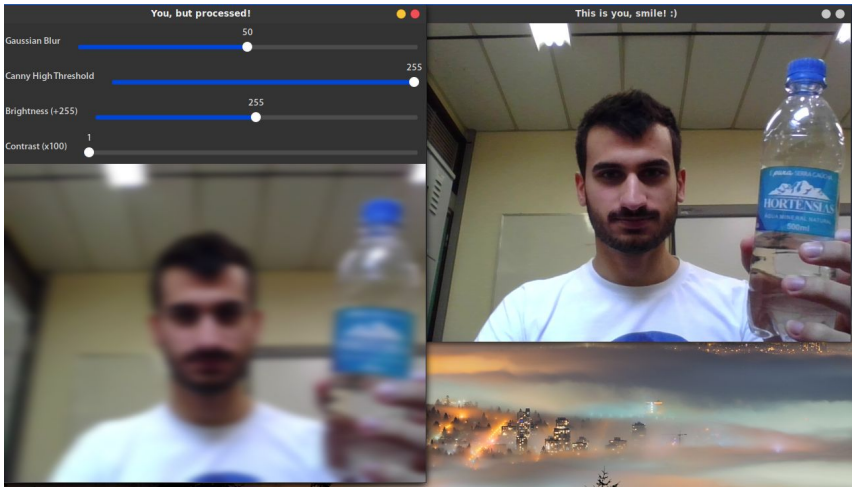
Para controlar quais processamentos de imagem são aplicados a cada quadro, determinados pelo desejo do usuário, foram criadas duas estruturas de controle. A primeira representa os *toggles* (ou ativações) do programa, enquanto a segunda representa parâmetros de processamento (tamanho do filtro gaussiano, *threshold* do Canny etc).

Usando essas estruturas, há uma função que observa a entrada do usuário, por teclas pressionadas, mudando as ativações. Também há barras seletoras (*trackbars)* que determinam os valores dos parâmetros recém citados.

Finalmente, para o processamento de imagens em si: como já foi indicado, a cada quadro do vídeo obtido da *webcam,* é chamada uma função que aplica os processamentos habilitados naquele momento, de acordo com os parâmetros configurados.

Borramento - Filtro Gaussiano

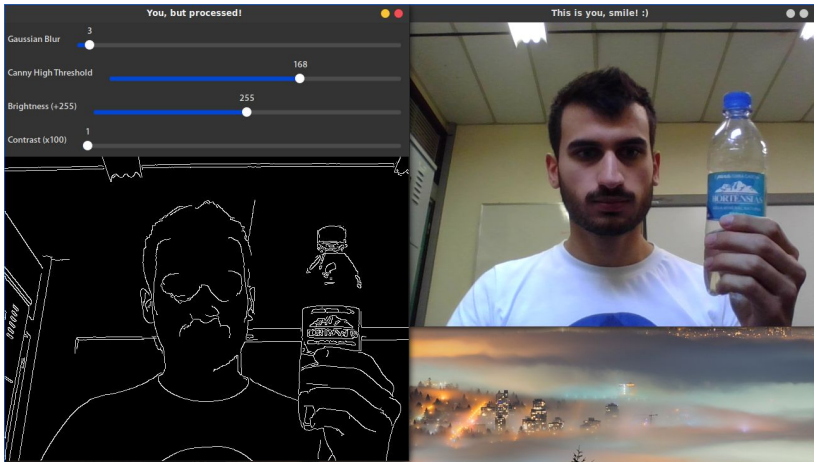
Sob demanda do usuário, pelo pressionamento de uma tecla definida, o programa aplica borramento ao vídeo, utilizando um filtro gaussiano. O tamanho do *kernel* do filtro é definido pela posição de uma *Trackbar*, aumentando a intensidade do borramento de acordo com a barra.



Filtro gaussiano de tamanho 50.

Detecção de Arestas - Canny

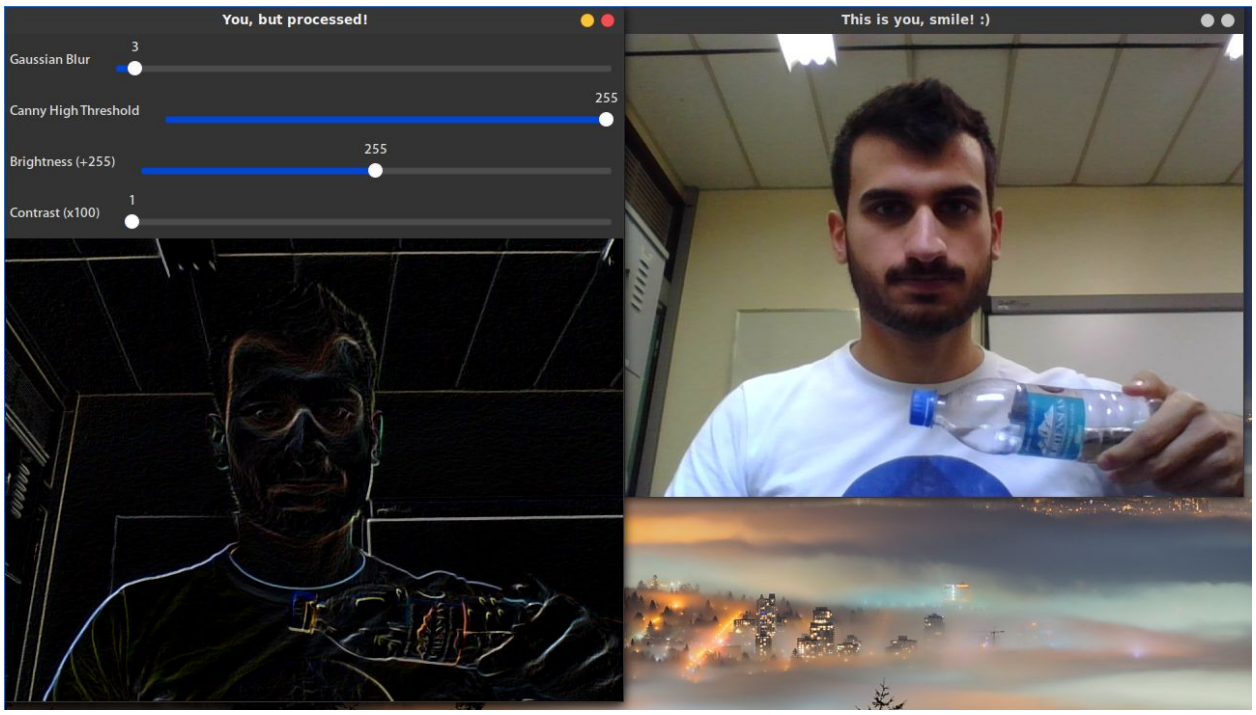
O programa também é capaz de realizar detecção de arestas, utilizando o algoritmo de Canny. O parâmetro *high threshold* do algoritmo é definido pelo usuário, através de uma trackbar de 0 a 255, e o valor *low threshold* é a terça parte de *high threshold* (*ratio* = 3).



Detecção de arestas pelo algoritmo Canny.

Gradiente - Sobel

É possível estimar o gradiente dos quadros do vídeo somando as aplicações de filtros sobel horizontais e verticais. Ou seja, grad = 0.5 * sobel_x + 0.5 * sobel_y.



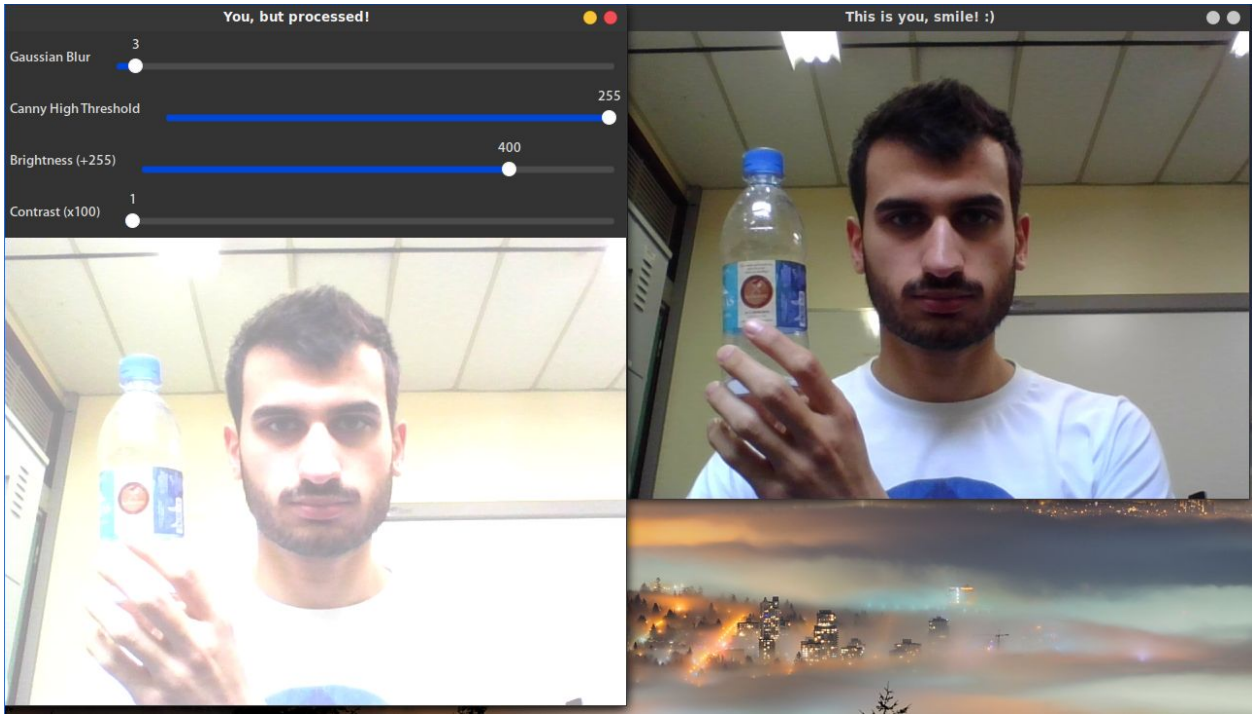
Cálculo de gradiente pelos filtros Sobel.

Negativo e Ajustes de Brilho e de Contraste

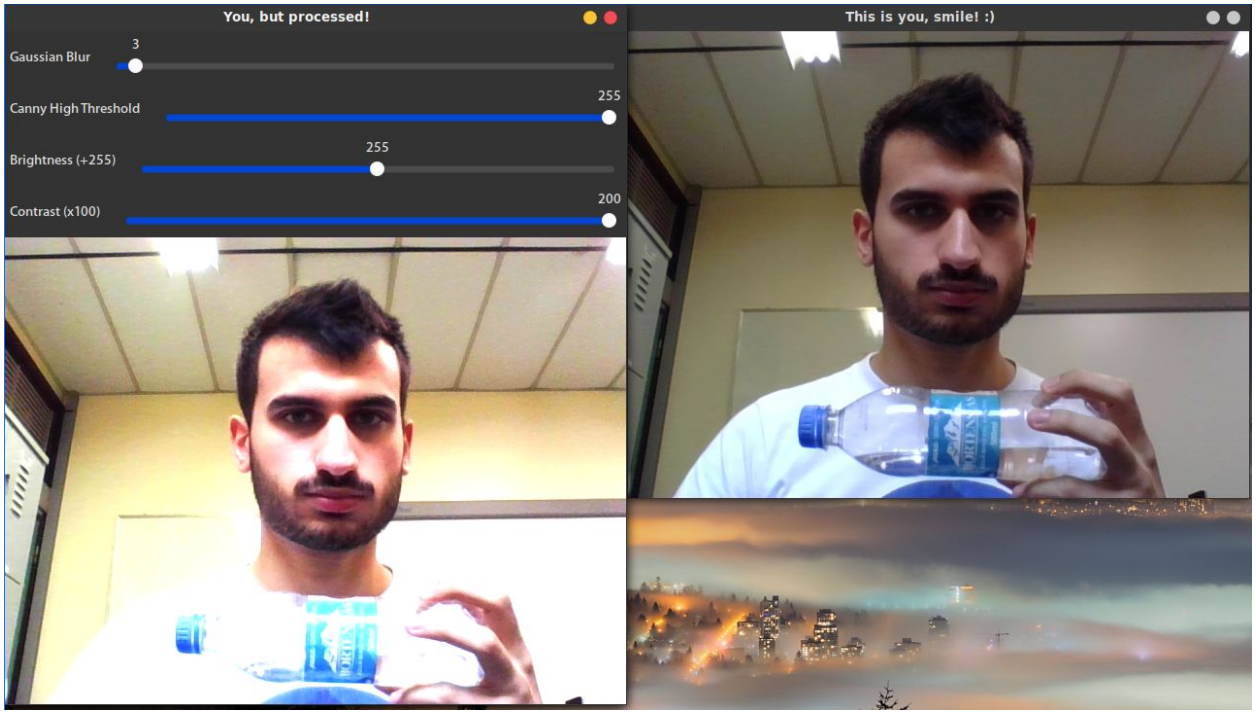
Em seguida, a função *convertTo( )* foi utilizada com os seguintes objetivos:

i. Ajuste de brilho pela soma de um termo de *bias* em cada pixel;
ii. Ajuste de contraste pela multiplicação de um termo de *gain* em cada pixel;
iii. Obtenção do negativo da imagem, com atribuições *pixel = 255 - pixel*.

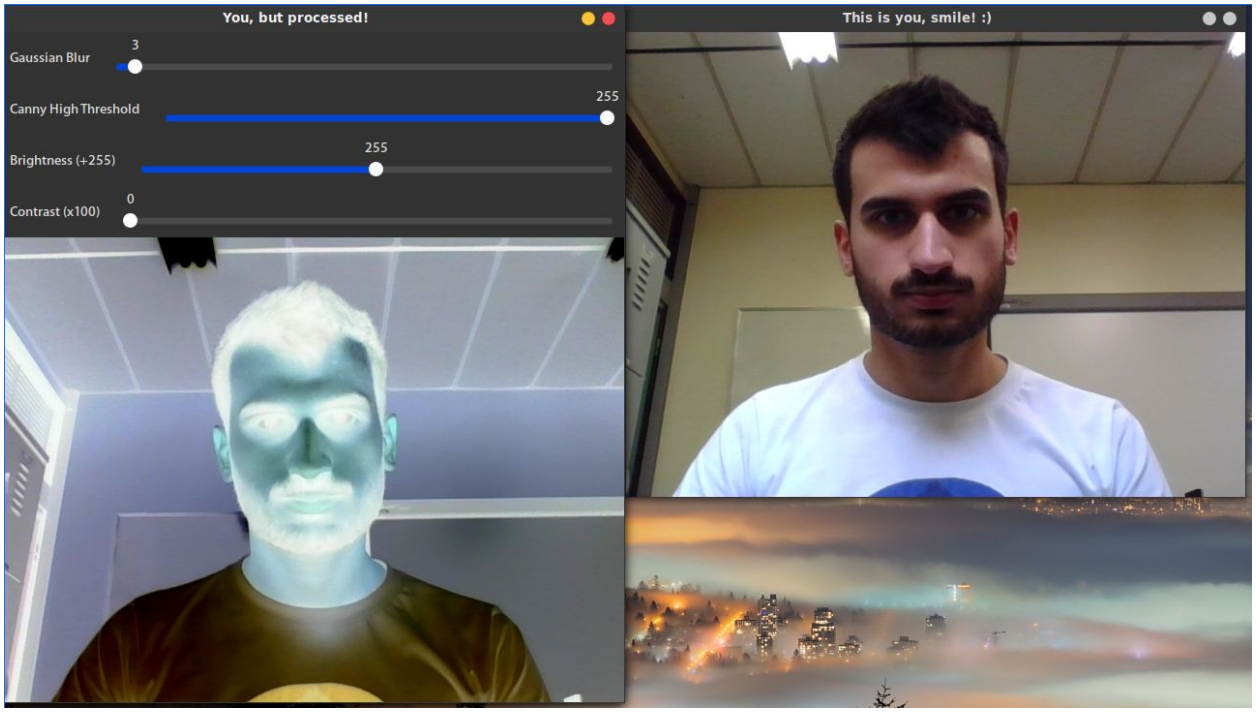
Para o ajuste do brilho, foi usada uma *Trackbar* de 0 a 510, representando valores de -255 a 255. Já para o contraste, há uma barra que permite selecionar valores de 0 a 200, que são divididos por 100 para obter valores em ponto flutuante de 0 a 2.



Aumento de brilho, somando valor 145 (400-255).



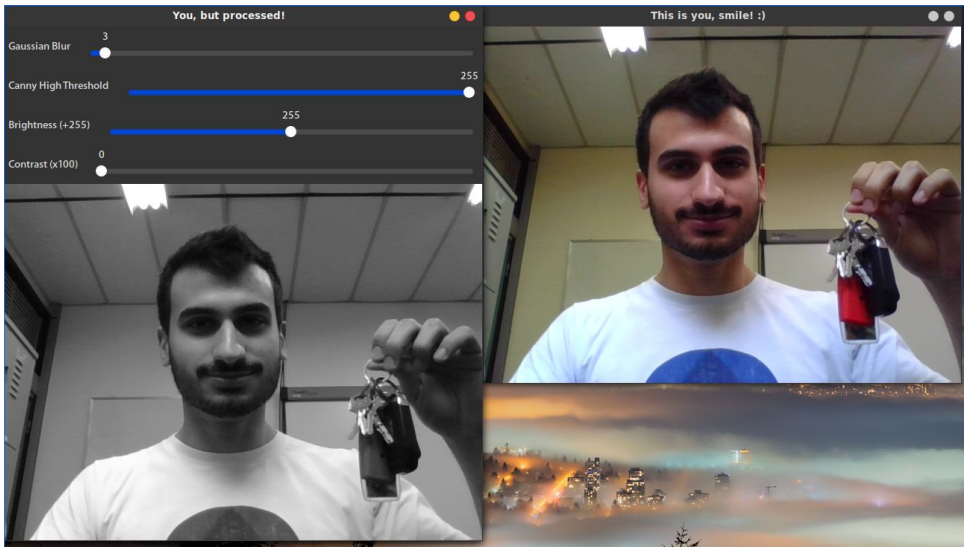
Aumento de contraste, multiplicando valor 2 (200/100).



Obtenção do negativo da imagem.

Conversão para Escala de Cinza

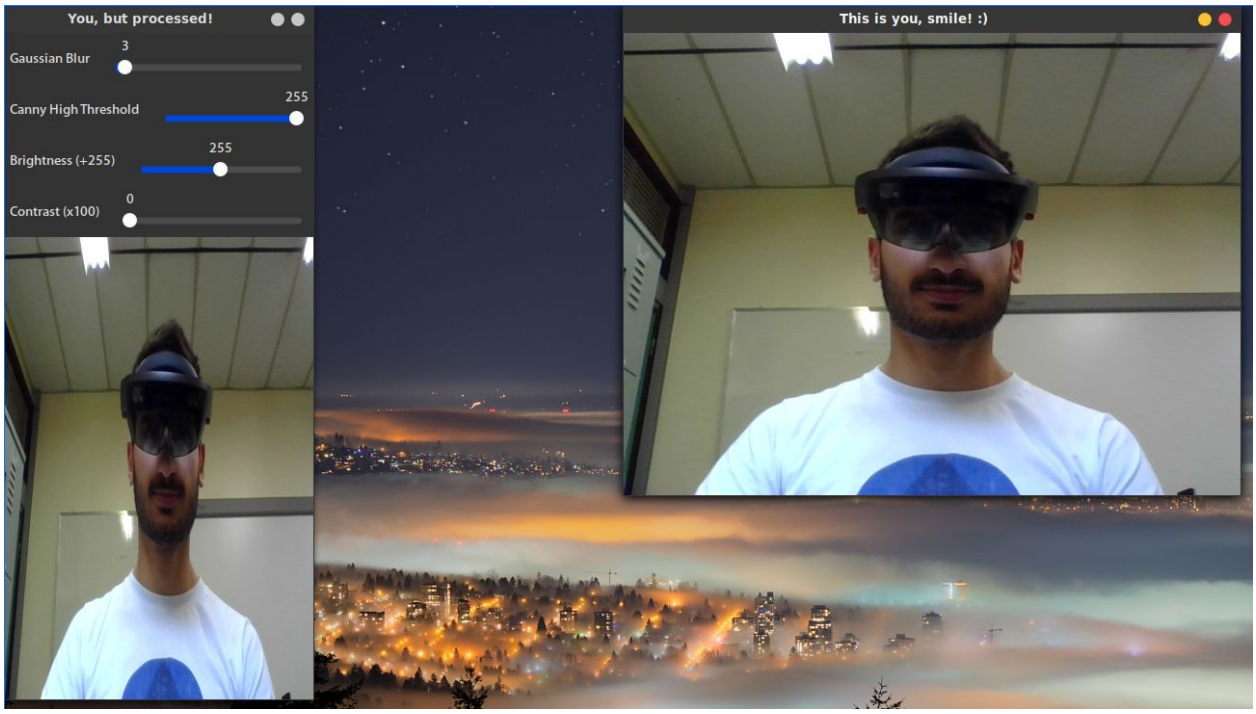
O programa, sob demanda do usuário, pode converter os quadros do vídeo para escala de cinza, ficando com apenas um canal (luminância) em vez de 3 (BGR).



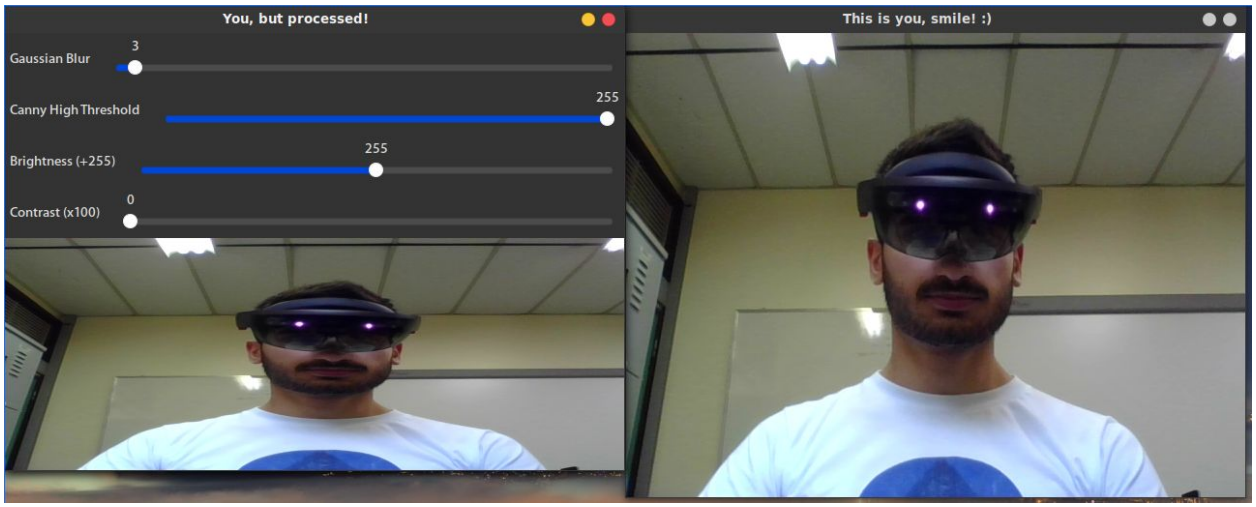
Conversão para grayscale.

Redimensionamento

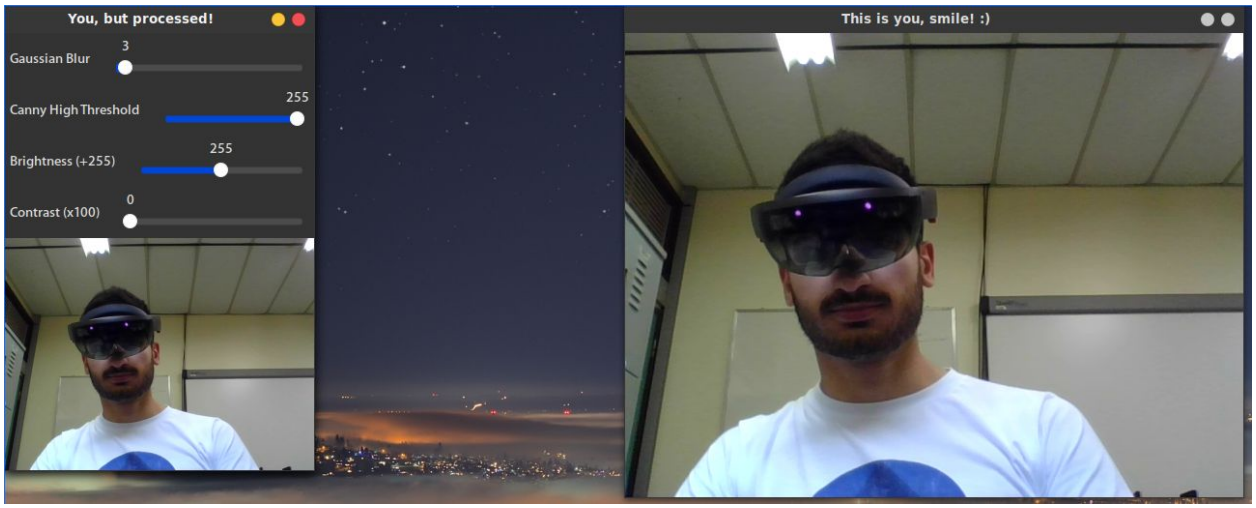
A próxima *feature* implementada foi a de afastamento (*zoom out*) pelo redimensionamento dos quadros do vídeo. O afastamento pode ser tanto na horizontal quanto na vertical (ou ambos), sempre resultando em uma dimensão metade do tamanho da original.



Redimensionamento horizontal.



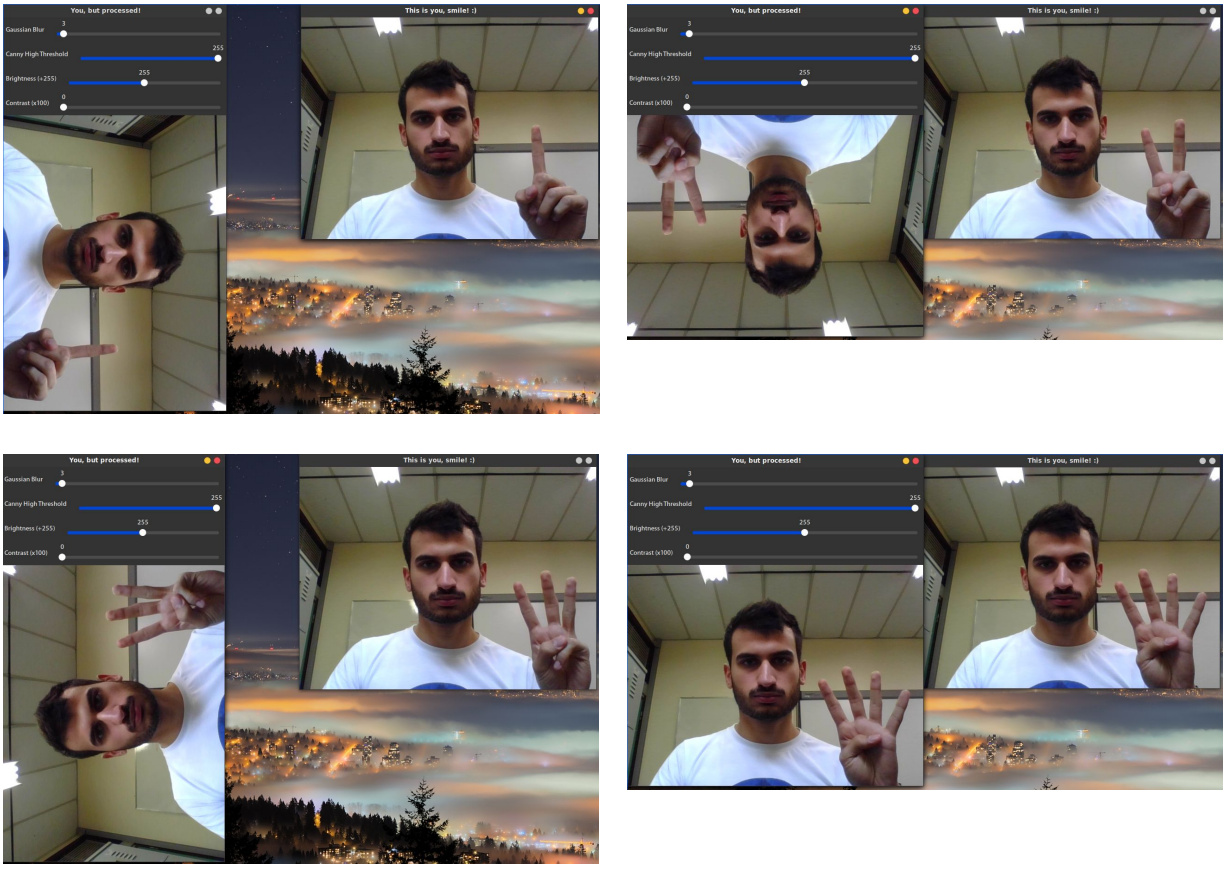
Redimensionamento vertical.



Redimensionamento horizontal e vertical.

Rotação de 90 Graus

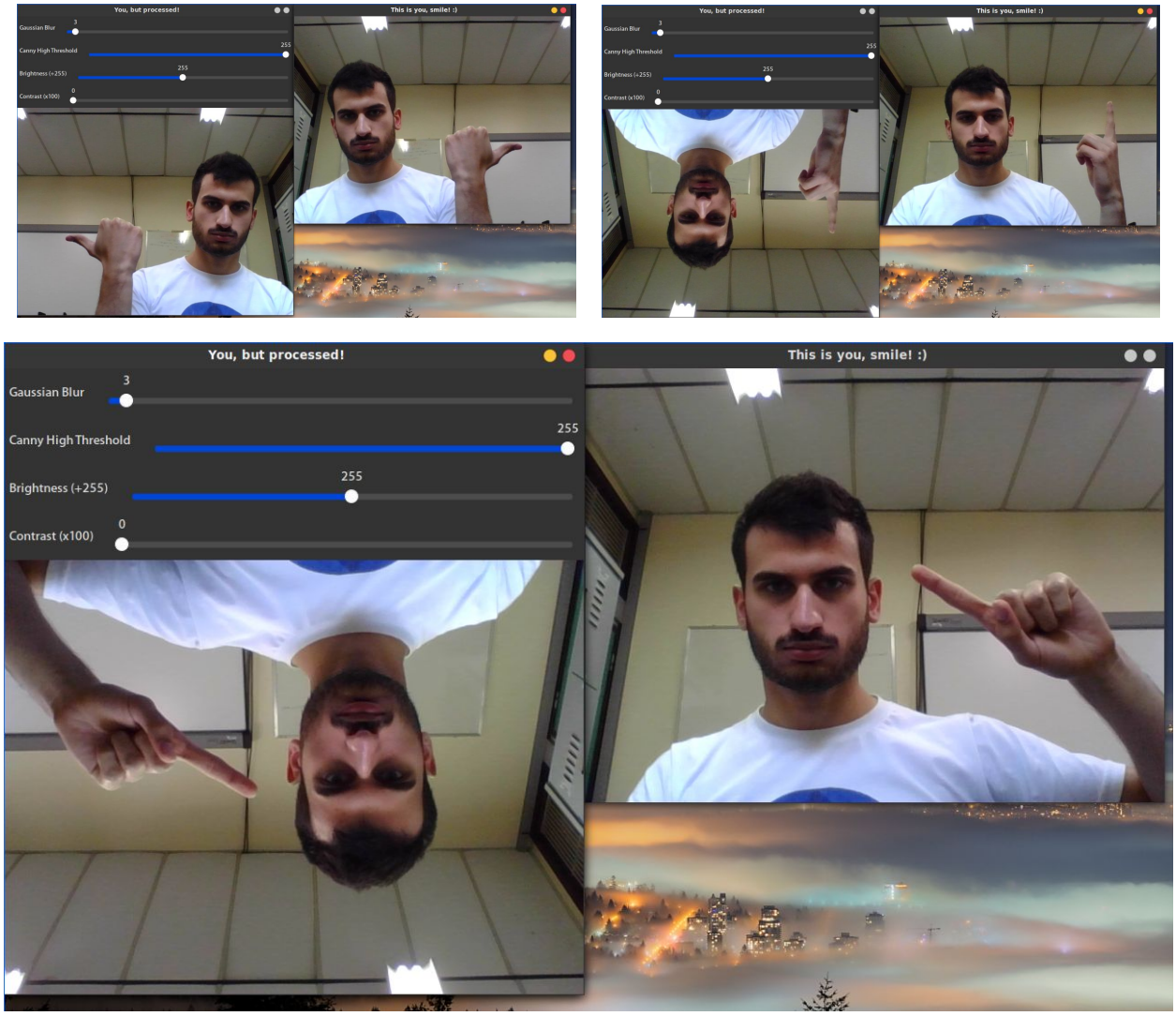
Em seguida, foi implementada a rotação de 90 graus da imagem. Aplicações repetidas resultam em diferentes combinações de ângulos retos: 90, 180 e 270 e 360 (0).



Rotações de 90 (1a), 180 (2a), 270 (3a) e 360 (4a) graus.

Espelhamento Horizontal e Vertical

A última manipulação de vídeo implementada foi a de espelhamento da matriz de pixels. Ela funciona nos eixos horizontal e vertical, separadamente ou em conjunto.



Espelhamento horizontal, vertical e em ambas as direções.

Gravação de Vídeo

O OVP é capaz de gravar a captura de vídeo do usuário em formato avi, utilizando codec XVID, em resolução 640x480, a 30 quadros por segundo. Foi feito o *upload* de um vídeo gravado pelo programa no Youtube, em https://youtu.be/KX-wrBpGUho, demonstrando a maior parte das funcionalidades implementadas.

Conclusões

Como relatado no documento, foi estudado o uso de diversas funções de manipulação de imagem do OpenCV, aplicadas ao contexto de captura de vídeo, quadro a quadro. Como consequência, foi adquirido domínio técnico sobre diferentes métodos de manipulação e processamento de imagens aritméticos e espectrais usando a biblioteca.

A maioria das funcionalidades estudadas estão presentes em produtos comerciais do ramo, portanto a prática do uso de APIs que as implementam é fundamental para futuros e eventuais projetos a serem desenvolvidos na área. Além disso, vale ressaltar a praticidade do OpenCV, que permitiu que a realização deste projeto fosse rápida e prazerosa.