Real Time Background Blur

Orion Crocker

June 8th 2020

1 Introduction

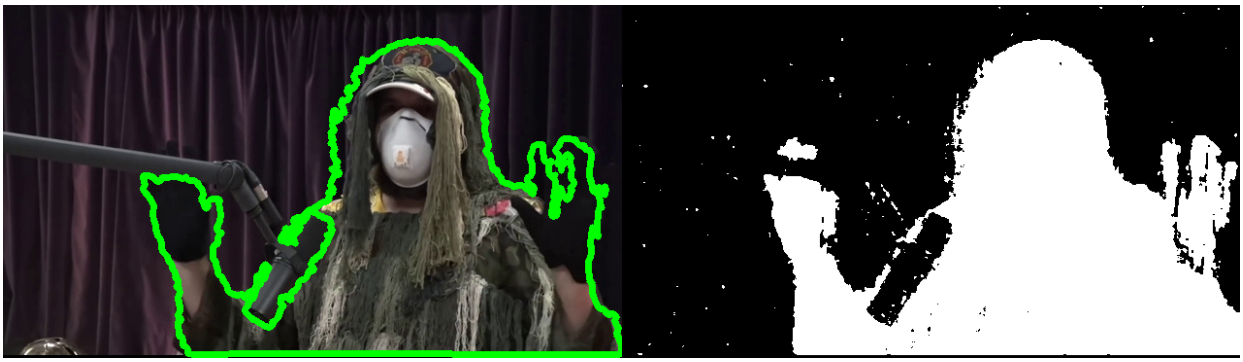
During the Covid-19 pandemic, we have all become intimately familiar with video conferencing software. Because most of us are stuck at home, chances are you have gotten an unwarranted look into your friends, classmates, or colleagues homes. Currently there are a number of effects that are available to protect the privacy of users on video conferencing services, such as Zooms background replacement tool or Skypes background blur tool. This project represents a personal intrigue into how these tools function.

2 Functionality

The KNN (K nearest neighbors) algorithm is the underlying method of which all other following methodology of the program's algorithm is built. In this regard, it is the backbone of the entire program. OpenCV's KNN algorithm is utilized as a background subtractor, meaning that it is used to differentiate between objects in the background of a frame from objects to the foreground of a frame.

Primarily, OpenCVs KNN algorithm detects 'movement' within a set of frames by comparing the pixel color and intensity change between those frames. The smaller the neighbor threshold, the more the algorithm detects 'movement' within the video.

From the set of 'movement,' shapes can be inferred and interpretted by OpenCV. From these shapes, outlines, or contours can be drawn. In the case of this program, it is assumed that the largest moving object in the frame is the object which will not be blurred. Because of this assumption, only the contour with the most surface area is kept, all others are discarded.



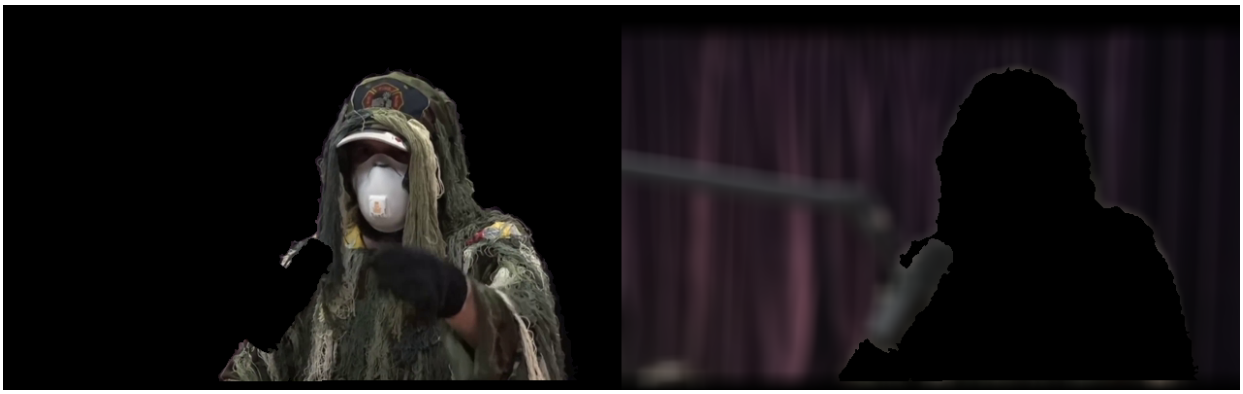
Now that the primary object (assumedly the user) in a video or webcam feed has been established, we can proceed to blur the rest of the frame. This is accomplished by first transforming

the original image to the RGBA color space. RGBA functions exactly the same as the RGB color space with the exception of an added fourth dimension named 'Alpha.' This dimension determines the transparency of the image, or how visible the image is. While transparency is not strictly used in layering images for the purposes of this algorithm, the fourth dimension 'Alpha' is integral to the success of the result. When transforming an image from RGB to the RGBA color space, all pixels undergo a simple transformation of it's pixel matrix from [R,G,B] to [R,G,B,A], where A is the highest value available or 255.

After being transformed to the RGBA color space, the frame is copied. This second copy will be referred to as the 'background,' and the original referred to as the 'foreground.' Next, the background frame quickly blurred using OpenCV's built in box blur algorithm. After doing many experiments on multiple live webcam feeds and test videos, a kernel size of 21 was chosen as the best looking outcome.

Once the background is blurred, the previously selected largest contour is applied on the now blurred background image. For every pixel within the largest contour, the background frame's corresponding pixel is set to a pixel value of [0,0,0,0]. Next, the foreground is inferred using the now blurred and contour drawn background image. For every pixel in frame 'background' that is not [0,0,0,0], set corresponding 'foreground' pixel to value [0,0,0,0].

$$\sum_{p=1}^{n} \text{if bg(p) != [0,0,0,0], fg(p) = [0,0,0,0]}$$



In the final step, the two images 'foreground' and 'background' simply have their total pixel values combined

3 Advantages

This is a tool that may be used to provide an extra layer of personal privacy when broadcasting from your home. The objects in the background of the frame, books on your shelf, etc will be obscured to curious onlookers.

This algorithm was made with the intention of being used in 'real time,' meaning that it employs quick and dirty solutions with the goal of keeping the video it is processing running as smoothly as possible.

4 Disadvantages

OpenCVs KNN algorithm is heavily dependent on a video capture device that is steady. Any motion of the device is interpreted as motion in the frame. This can pose a problem to users utilizing this software on laptops or unstable desks, as each keystroke could create a micromovement of the machine itself which translates to the software as movement. Because any movement of the camera is interpretted by KNN as movement, this software will best be used in situations where the camera is fixed to a solid desk or a wall.

Below is an example of a video that does not work well with this software. While the shot looks relatively stable, the camera is a great distance away from the target and is zoomed in. This ensures that the smallest movements of the camera itself create a lot of noise for KNN to have to deal with. Additionally, the flag in the background creates movement which complicate's KNN's task of background and foreground seperation. The results of the software given these circumstances do not produce satisfactory results.

1

[1]Trump mobilizing 'heavily armed' military to stop protests

5 Results

All videos used to test this software were downloaded from YouTube.com.

2 3 4

[2]Joe Rogan Experience 1444 - Duncan Trussell
[3]Ozgur Baba - Dertli Dolap
[4]TASH SULTANA - JUNGLE (LIVE BEDROOM RECORDING)