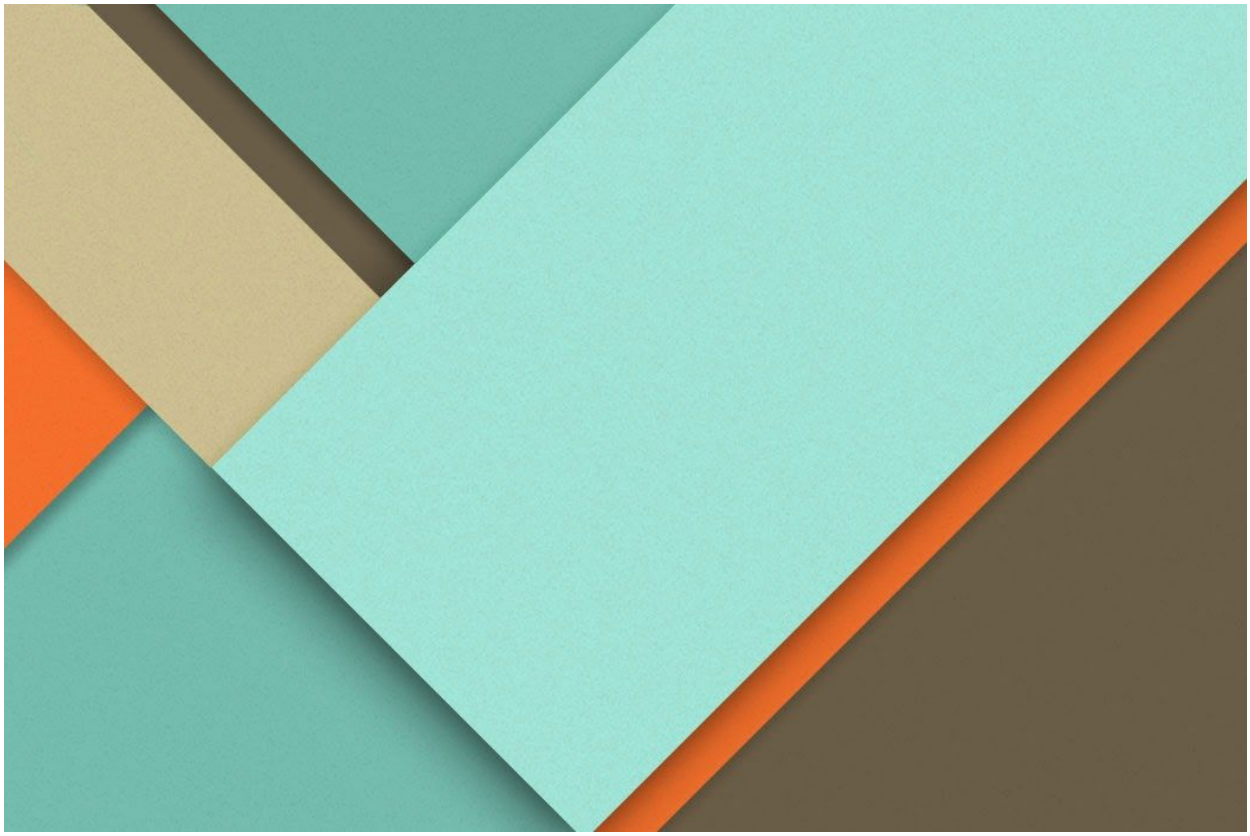


Computer Vision

ASSIGNMENT 0 - CHROMA KEYING WITH OPENCV

AJAY SHRIHARI
20171097

TASK 1

Video <-> Images.

For this task, two functions were written vid_to_img() and img_to_vid(). Both functions take in the video and image paths.

vid_to_img()

This function uses a flag exist to check if any video is present in the path, and whether any image can be read from the cv2.VideoCapture() object, which reads the video frame by frame This then writes the image frame to a folder, with each frame assigned a different name.

In the code, 'Bird.mov' is the video written to the 'img/' folder as an experiment.

img_to_vid()

This function uses glob to run through through the files with the given pattern, which are the frames of images which need to be combined to a video. We use glob() method in glob for this, and sort the files based on the regular expression. Once the images are read, we use the cv2.VideoWriter with fourcc codec and use the fps we require (30 used). These frames are then written to the file as a video.

In the code, the images from 'img/' are written to 'video.avi' as an experiment.

TASK 2

Capturing Images

This task requires us to capture the video from the webcam and convert it to frames. This process is identical to the above problem, except we subscribe to the webcam using cv2.VideoCapture(0). We also include a flag to close the webcam on pressing the 's' key when the webcam is to be stopped.

The required frames are stored in the 'webcam/' folder as an intermediate experiment.

TASK 3

Chroma keying

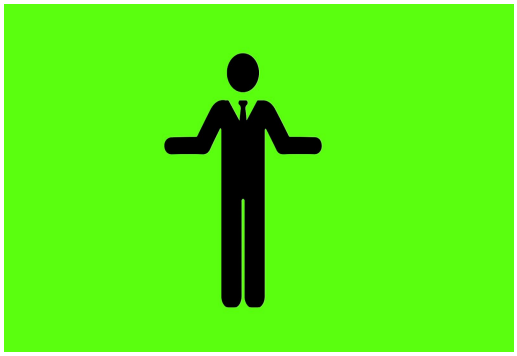
This task requires us to have a background video, and include our given image video with the changed background. For this, we deconstruct both videos to their frames, then apply chroma keying for the corresponding frames. For this, chroma_key() function was written.

chroma_key()

This function takes in the image and background image, and outputs with the image, with the given background. For this, an upper and lower bound of green is created. This is turned into a mask using the cv2.inRange() function. When the image contains the texture of green within the mask, the corresponding pixels are set to 0. After this, the image and background are concatenated to get the required chroma-keyed output.

Here, the frames are taken from 'img/', and the background is 'bue_sky' (which is deconstructed to frames and kept in 'sky/'). The output concatenated is sent to 'chroma/', and reconstructed into an image using img_to_vid() with the name 'chroma_video.avi'.

An example of the chroma keying, with archetypal **input** frames used:



Output frame:

Input and output contained in the following drive link:

https://drive.google.com/drive/folders/1nikx5IFJQgdZRDbLHEeMY6UXmtWm-wI0?usp=sharing

Inputs: 'Bird.mov"

FInal output video: 'chroma_video.avi'

Intermediate outputs: 'img', 'video.avi', 'sky/', 'webcam/', 'chroma/'

Requirements (all pre installed):

OpenCV

Python3

Glob

Re

Learning from the assignment:

A good recap of OpenCV video and file handling functions. Apart from this, I got a good understanding of chroma-keying. The assignment helped me in exploring OpenCV functions.