



Invisibility Cloak using Color Detection and Segmentation with Open CV

Sapna Malik, Simran Kaur, Sejal Rana

Abstract: Have you ever thought of making visible things invisible, just like the Harry Potter? Have you ever thought how does one supersede backgrounds and add effects in a movie?

The cloak was magical and invisible in Harry Potter, the movie. As we know there is no magic and no invisible cloak which exists in the world. It's all about the graphics tricks.

The concept of an invisibility cloak is a mixture of science, fantasy, and the collective imagination. This paper helps to create one's own 'Invisibility Cloak'. It will make use of Python and OpenCV module specifically targeting Image Processing and Image Segmentation to create a false sense of invisibility in the frame. It will explore how an object of a specific color or texture can be manipulated using the OpenCV library of python.

To achieve this, initially we'll be capturing and storing the backdrop frame. Thereafter we'll be identifying the red coloured fabric by making use of the above mentioned algorithms. Then we'll segment out the red colored fabric by generating a mask and then finally, we'll generate the final augmented(magical) output to create Invisibility cloak. These steps are discussed deeper in the paper.

Keywords: Invisibility Cloak, Image processing, OpenCV, Python

I. INTRODUCTION

With the advancement of Artificial Intelligence, [1] Computer vision came into the picture in late 1960s. Its whole purpose was to increase the intellect of the artificial mechanism available by installing the cameras into them and describe whatever they saw just like humans' visual system.[2] Thus, Computer vision should be able to detect actual daily-life 3D objects through 2D pictures. Every picture tells us a story, something present now or what has been going on at a particular time. OpenCV is Open Computer Vision Library of python. It was launched by Intel in 1999. With more updates, it has been modified since then to aim for the real-time computer vision. [3] This library has been written under programming languages like C and C+. It can be easily run on operating systems Windows and Linux. This library can be easily interfaced with programming languages like Python, MATLAB, Ruby and others as well. Along with Numpy and [4]Python image processing (shape & color detection) can be performed at ease.

Revised Manuscript Received on April 12, 2020.

* Correspondence Author

Dr. Sapna Malik*, Assistant Professor, Department of Computer Science & Engineering, Maharaja Surajmal Institute of technology, GGSIPU, New Delhi, India.

Simran Kaur, Student, B.Tech, Computer Science & Engineering, Maharaja Surajmal Institute of technology, GGSIPU, New Delhi, India.

Sejal Rana, Student, B.Tech, Computer Science & Engineering, Maharaja Surajmal Institute of technology, GGSIPU, New Delhi, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

II. METHODOLOGY

The working is opposite to the concept of green screening. As in the case of green screening background is removed but in this application we remove the foreground. Red colored cloth is used as a cloak for this application. We can use any colour with a little bit of changes in the code.

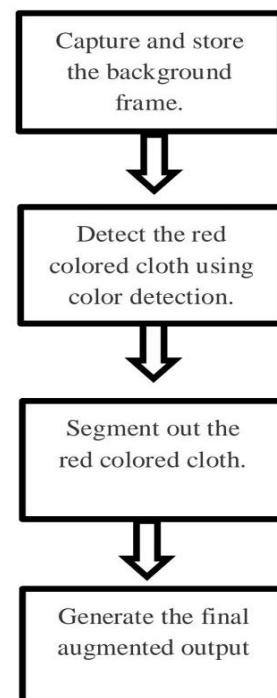


Fig 1: Flow Chart

This fig shows the steps to be followed in order generate the augmented output.

1. "Capture and store the background frame [This will be done for some seconds]"

The main concept is interchanging the current frame picture element equivalent to the fabric with the backdrop pixels, so that we obtain the magical effect of invisibility. Thus, we're required to save a frame of the backdrop.

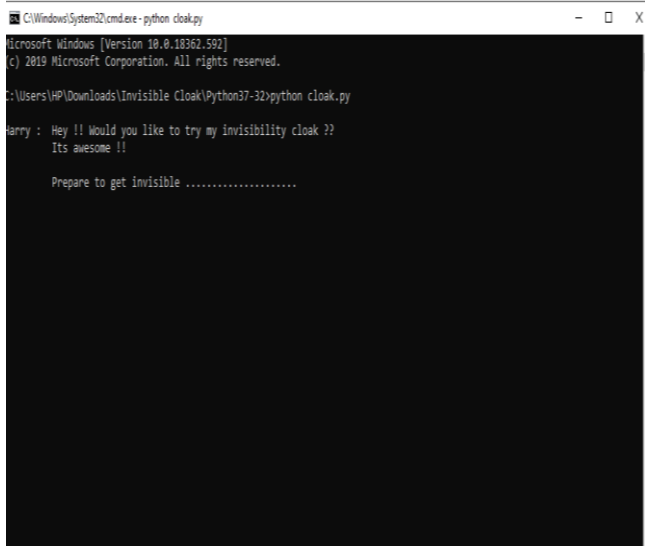


Fig 2: Application Launch

This is the initial stage where the application is launched.



Fig 3: Background Capture

Capturing and storing the backdrop would take a few seconds

2. “Identify the red coloured cloth (cloak) by using the colour detection & segmentation algorithm.”

The correct concept is to change the colorspace of the picture from ‘R.G.B’ (Red.Green.Blue) value to ‘H.S.V’ (Hue . Saturation . Value).

About ‘H.S.V’ color space:

- ‘Hue’ : It gives the information about color . Hue can be defined as a number from [7] 0 to 360 degrees in which 0 degree refers to the red , 120 degree refers to the green & 240 degrees defines the blue color.
- ‘Saturation’ : It explains about the intensity of the color.
- ‘Value’ : It tells about the luminosity of color. Shading component of an pictures appears here.

For our cloak, the most-important merit of using the “H.S.V” colorspace is that the color or wavelength is presented by the ‘Hue’.

In this, we’ll initially capture a live frame, then convert the image by using ‘R.G.B’ to ‘H.S.V’ color space conversion and later define an appropriate range of ‘H.S.V’ values [12]. Here we used the red color cloth to change into invisibility cloak. [13]The main concentration was put in founding the red color in the background.

[5]Now, we have a “R.G.B” valued picture and it is enticing to simply threshold the R value channel and get the mask. Hence, we get to know , that this will not work properly since, the “R.G.B values” are highly - ‘sensitive to illumination’ [6].

Thus , [8] the proper way, is transforming the color space of our picture from **R.G.B** to **H.S.V** (Hue .Saturation .Value). [“mask1 = mask1 + mask2”]

Using this, we join the masks generated for both of the red color range.

3. “Segmenting out the red colored fabric by generating a mask.”

We refine the mask & then it is further used for segmenting out the fabric from the frame.

4. “Generate the final augmented(magical) output to create Invisibility cloak.”

Finally, we’ll be replacing ‘the pixels value of the detected red colored area of the cloth with corresponding pixel value of the background’ & ultimately generating an augmented (magical) output.

III. RELATED WORK

1. (Puri, Gupta, & Sikri, 2018) studied “Contour, Shapes, and Color Detection with Open Cv library of Python” and found that the contours, shapes and colors of several geometrical shapes in the sample binary-images through Python 2.7, Open Source Computer Vision Library and Numpy. The very significant functions were used for processing of the images, that involved- loading them, recognizing various shapes and colors inside the example images.

2. (Naveenkumar, 2016) studied “OpenCV for Computer Vision Applications” and found that image processing aimed to help the computer to understand the content of an image.OpenCV was a library of programming functions essentially used for image processing. It provides a de-facto standard API for computer vision applications. We can solve many real-time problems using image processing applications.

3. RAGHAV PURI, ARCHIT GUPTA, MANAS SIKRI [2018] analysed that the computer C/C++, allows to write computational intensive code in C or C++ and create Python wrappers . All the basics of the detection technique and various ways of achieving it were discussed in depth. Python and MATLAB can be used for computer vision during the development process, but Python was chosen because it takes less time to simulate than MATLAB.

IV. LIMITATIONS

1. We use the range 0 to 100 & 170 to 180 to avoid the detection of skin as red. This is because generally in winters, the skin gets red.[9] The large range of 120 to 255 for saturation is used because ‘our cloth should be of a highly saturated Red-color’.
2. We require to use the lower-range value as 70 so that ‘we could detect red color in the creases of the cloth as well [11]’.

3. Since the RGB values are enormously responsive to illumination, [10]. It seems that this may no longer work expeditiously. Despite of the cloak's color being red there might be some areas where, due-to shadow, the red channel values of the corresponding-pixels are a bit low.

V. RESULT

Our main task is 'to substitute the current frame pixels corresponding to the fabric with the backdrop pixels to create the effect of an invisibility-cloak' ; to achieve this , we require to save a frame of the backdrop. Since w're using a red color fabric to transform it into an invisibility-cloak we would focus on the discovery of red colour in the frame. Firstly, we would capture a live frame, change the picture from R.G.B to H.S.V colour space & then specify an appropriate range of 'H.S.V' values in order recognize the red colour.

VI. CONCLUSION

Computer vision can be used to solve the most intriguing problems with utmost sophistication. All the basics regarding the colour detection technique along with different ways to achieve it have been profoundly discussed in the paper. During the course of programming, we can use both Python and MATLAB for Computer Vision, but we prefer Python because it takes less simulation time than MATLAB. Someone having prior coding experience finds it easy to implement.

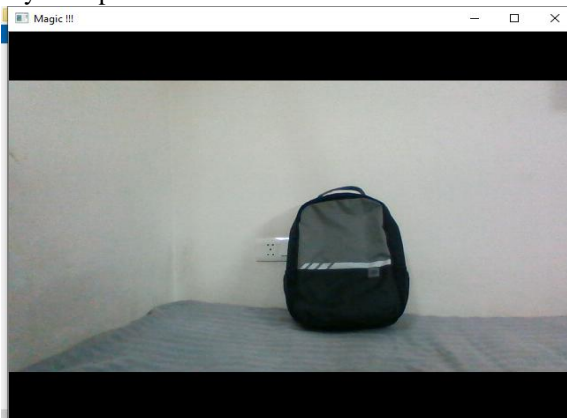


Fig 4: Before Implementation

Figure represents the image when the red coloured fabric wasn't placed in front of the object.

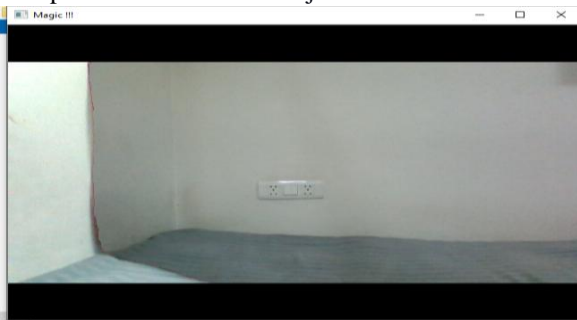


Fig 5: After Implementation

After placing the red coloured fabric in front of the object , the final augmented (magical) output was generated.

FUTURE SCOPE

Computer vision has still not achieved a level wherein it can be directly put into use to solve life problems, as it is still in its developmental phase. Also, it can be widely used in the applications of Augmented Reality.

Some of the applications are

- Video Editing & media in order to create stunning visuals.
- "Infinity Tower" is an invisible skyscraper in South Korea (under construction).



Fig 6: Video Editing



Fig 7: Infinity Tower (under construction in South Korea)

REFERENCES

1. (PDF) Facial Recognition using OpenCV. (2012, March 1). Retrieved from https://www.researchgate.net/publication/267426877_Facial_Recognition_using_OpenCV
2. AUTOMATIC IMAGE PROCESSING ENGINE ORIENTED ON QUALITY CONTROL OF ELECTRONIC BOARDS. (n.d.). Retrieved from https://www.academia.edu/36556406/AUTOMATIC_IMAGE_PROCESSING_ENGINE_ORIENTED_ON_QUALITY_CONTROL_OF_ELECTRONIC_BOARDS

3. (PDF) Image Processing using OpenCV and Python in Ubuntu. (2014, June 15). Retrieved from https://www.researchgate.net/publication/274634881_Image_Processing_using_OpenCV_and_Python_in_Ubuntu
4. (PDF) Introduction to Computer Vision in Python. (2009, January 1). Retrieved from https://www.researchgate.net/publication/228378315_Introduction_to_Computer_Vision_in_Python
5. (PDF) Face Detection & Face Recognition Using Open Computer Vision Classifies. (2017, August 4). Retrieved from https://www.researchgate.net/publication/318900718_Face_Detection_and_Face_Recognition_Using_Open_Computer_Vision_Classifies
6. Attendance Marking System Based on Face Recognition Using OpenCv and Python. (2017, November 7). Retrieved from https://www.researchgate.net/publication/320716769_Attendance_Marking_System_Based_on_Face_Recognition_Using_OpenCv_and_Python
7. Invisible Cloak using OpenCV | Python Project. (2019, June 7). Retrieved from <https://www.geeksforgeeks.org/invisible-cloak-using-opencv-python-project/>
8. Color Detection and Segmentation with OpenCV | Learn OpenCV. (2019, February 11). Retrieved from <https://www.learnopencv.com/invisibility-cloak-using-color-detection-and-segmentation-with-opencv/>
9. A Gentle Introduction to Computer Vision. (2019, July 5). Retrieved from <https://machinelearningmastery.com/what-is-computer-vision/>
10. Introduction to computer vision: what it is and how it works. (2018, April 2). Retrieved from <https://algorithmia.com/blog/introduction-to-computer-vision>
11. Computer vision. (2001, October 25). Retrieved from https://en.wikipedia.org/wiki/Computer_vision
12. (PDF) Experimental study on invisibility cloaks. (2016, May 1). Retrieved from https://www.researchgate.net/publication/305027092_Experimental_study_on_invisibility_cloaks
13. Invisibility cloak with image projection capability. (n.d.). Retrieved from <https://www.nature.com/articles/srep38965>

AUTHOR PROFILE

Dr. Sapna Malik, is an Assistant professor in department of Computer Science & Engineering at Maharaja Surajmal Institute of technology, GGSIPU, New Delhi, India.

Simran Kaur, is a final year student pursuing B.Tech in Computer Science & Engineering from Maharaja Surajmal Institute of technology, GGSIPU, New Delhi, India.

Sejal Rana, is a final year student pursuing B.Tech in Computer Science & Engineering from Maharaja Surajmal Institute of technology, GGSIPU, New Delhi, India.