World Journal of Advanced Engineering Technology and Sciences

eISSN: 2582-8266

Cross Ref DOI: 10.30574/wjaets

Journal homepage: https://wjaets.com/

(Research Article)

Check for updates

Invisible cloak and skin-tone masking

Kunal Roy [1,*], Dishannita Bhattacharjee [2], Shubhendu Nath Biswas [2], Biswajit Mondal [2], Asha Laha [2] and Akash Ghosh [2]

[1] CSE, Indian Institute of Technology (Indian School of Mines), Dhanbad, India.
[2] Computer Science Engineering, Durgapur Institute of Advanced Technology and Management, Durgapur, India.

World Journal of Advanced Engineering Technology and Sciences, 2025, 15(01), 2039-2047

Publication history: Received on 14 March 2025; revised on 21 April 2025; accepted on 23 April 2025

Article DOI: https://doi.org/10.30574/wjaets.2025.15.1.0407

Abstract

This study is based on the development of "Invisible Cloak" and "Skin-tone Masking" systems using Python. Advanced Computer Vision techniques the background removal is manipulated, creating an effect where the person who will be wearing a cloak of "Green" color will seamlessly blend into their environment. Real-time segmentation and background subtraction are the two important methods used here to achieve the aim. The second part, that is, the Skin-tone Masking is basically modifying or counterbalancing the skin-tones in images or video streams. Computer vision and Machine learning techniques are used in the project for the enhancement of privacy, which in turn can be used in surveillance, security and others.

Keywords: Computer Vision; Invisible Cloak; Python; Real-Time Segmentation; Skin-Tone Masking

1. Introduction

Day by day advancements in the fields of computer vision and image processing technologies is surprising everyone. The numerous possibilities for manipulating and enhancing the visual media is seen in a remarkable way. One of those applications is our project, that is on development of techniques for creating "invisible cloaks" and "skin tone masking" gaining significant attention for their potential use in fields like entertainment, privacy and aesthetic enhancement.

The first part of the project ,that is, the "invisible cloak" involves creating an illusion of invisibility well of an object, basically a cloth piece or something which is "green" to make it invisible or blend seamlessly with the background. In the world of science fiction, invisibility has been widely popularized but these days the modern technologies in image processing and computer vision brings it closer to reality. Different methods of background subtraction, object segmentation and pixel manipulation makes it possible to remove or replace the subject from a scene creating an invisibility effect [1].

On the other hand "skin tone masking" leverages skin detection algorithms for identifying and modifying regions of an image that represents the humans can tones skin tones it can be applied for various purposes like enhancing visual esthetics in photographs or videos implementing beauty filters ensuring privacy or even as part of machine learning applications to anonymize faces.

This paper explores the area of developing and implementing both the concepts using Python and open source computer vision libraries such as OpenCV. The application of advanced image processing techniques aims to demonstrate how an "invisible cloak" effect can be achieved while removing or blending objects with the background and how "skin tone masking" can provide numerous modifications to image while respecting privacy concerns. Our core

* Corresponding author: Kunal Roy.

goal was to showcase the potential for python in development of innovative visual solutions in both creative and practical domains.[2].

1.1. Objective

The main objective is to create an illusion of invisibility in the real time scenario. This work comprises the use of computer vision techniques to manipulate images or videos. The invisibility is created by using background subtraction techniques where after identifying the background it is subtracted leaving only the subject visible. The goal of the skin tone masking is identification and masking the specific regions of an image based on the skin tones basically for skin detection. The python library such as OpenCV detecting regions that match humans skin tones and applying certain effects for the visual modifications. The project is a mix of image processing and color recognition. Thus. a seamless blending and a more natural output can be gained, enhancing the overall realism of the effect.[3].

1.2. Population and Sample

Population refers to the whole group from which data will be collected. Defining the population as:

- All people in a video image as potential subjects of the invisibility effect.
- Various types of cloaks of “Green” color and materials bear relevance because color and texture of the cloak can bring about the success of an invisibility effect.
- Skin shades differ from individual to individual, thus, skin color masking has to cater to a range of skin complexes.
- Environmental factors like various kinds of lighting conditions, camera angles, and types of backgrounds increase the susceptibility of the invisibility effect.

Sample refers to a small part of the population that is chosen for testing, representing a smaller, manageable group that we can analyze to make conclusions about the larger population. Out of the full data set in the CSV file we have chosen a smaller subset to work with the context of our experiment. The sample includes different shades of “Green” for the invisible cloak [4].

1.3. Data and Sources of Data

Data for the invisible cloak and skin-tone masking is basically visual, captured using video recordings and images. The frames have individuals wearing the specially colored cloak in different environments. In addition, it includes diverse skin tones and different background settings to ensure proper testing.

1.3.1. Data Sources

- Real-time video using a webcam or camera for live testing.
- Images of simple backgrounds and intricate scenarios for testing the blending effect.
- Data capture in indoor, outdoor, and low-light scenarios to reflect as much as possible real-world use cases [5].

1.4. Theoretical framework

The invisible cloak and skin-tone masking project is based on the underlying theories of computer vision, image processing, and manipulation in color space. The theoretical model is designed as a backbone based on major techniques, the purpose of our project is to explore how these techniques can be combined for real-time invisibility effect production that will be able to selectively mask certain elements, especially those are human skin-tones, while retaining the overall integrity of the surrounding scene.

- **Background Subtraction:** This is the fundamental technique in processing the video, often used to detect moving objects by comparing each new frame to a reference background frame. A steady background frame is the reference for Invisible Cloak. It shows areas that are different for the cloak or subject in each video frame [6]**.** The accuracy of this technique is crucial, as it determines the precision of the regions that will be masked in real-time, ensuring that the invisibility effect is seamless.
- **Color-Based Segmentation using the HSV color space:** HSV color space is used for finding the color of the cloak, which is superior to RGB for varied lighting conditions [7]. The standard RGB color model, that is Red, Green and Blue, can sometimes struggle with varied lighting conditions, making it difficult for the consistent identification and tracking of the cloak or the objects that are needed to be concealed. HSV is often preferred for tasks such as object tracking or color segmentation because it is more robust to changes in lighting and

shading. The channel hue in the HSV is particularly useful for identifying the specific color of the cloak, which can then be isolated from the rest of the scene.

- **Skin Tone Masking:** the invisibility function is supposed to apply to skin tones if and only if, the user gets objects from the invisibility filter in those particular skin tones. On using the advanced skin detection techniques such as the color thresholding or nay machine learning models trained on skin tone data-the system can identify areas that correspond to human skin and apply the invisibility effect only to those regions, ensuring that the user's face, hands, or any other exposed skin are masks in real time without affecting other objects in the scene.
- **Morphological Computations:** Masks can be refined more perfectly by erosion and dilation. Erosion reduces the size of the mask, helping to remove small unwanted regions or noise, while dilation on the other hand expands the mask slightly, ensuring that it fully covers the intended object or skin regions, thus giving more valid object regions and not interfering with noise.
- **Real-time video processing:** One of the biggest challenges in implementation of an invisible cloak effect is performing all the necessary operations in real-time. Video processing, especially when it is dealing with complex tasks like background subtraction, color segmentation and morphological refinement, requires significant computational power. The CV toolkit comes with numeral computation and a library to perform operations at lightning speed, with no jitter at all on real-time video blanking out perfectly [8].

2. Research methodology

This section outlines the materials and methods used to develop and evaluate real-time background capture and background masking techniques. It includes the hardware and software setup, data acquisition, algorithm implementation, and evaluation metrics. The program's methodology can break down into distinct phases, focusing on capturing a clean background, detecting specific regions (clock and skin), and replacing the cloak region with the background to create the Invisibility Effect [9].

2.1. Materials

2.1.1. Capturing The Real-World Background

The first and foremost pivotal step is creating the invisibility cloak effect. A uniform clear background is captured in order to place it in the region of the cloak. Capturing a clean stable real-world background is a crucial step in this program. Because any discrepancies between the original background and the generated replacement can break the illusion of invisibility.

2.1.2. Steps

The first task is capturing the real world with a camera, here we implemented with the webcam, as it can be done using any standard webcam or even higher-end cameras depending on the details desired. The position of the camera should be in such a way that it will not hinder the view of the background that will be used to replace the cloak region.

After the background is captured, the system uses a technique called median frame computation for stable background generation, reducing noise caused by subtle variations [10]. When a single frame is taken it can contain noise or small variations due to motion, lighting changes, or slight camera shifts. To mitigate this, the median frame is computed over a series of frames, essentially reducing the noise and creating a more consistent and stable representation of the background. This process is to ensure that any temporary fluctuations in the background will not affect the background image, thereby offering what we need for an invisibility effect.

2.1.3. Video Frame Processing

The core process where the actual working happens. Each frame is processed in the live video feed to identify cloak and non-cloak regions. Video frame processing is the core of the invisibility cloak program. It involves analyzing each video frame to identify and separate specific regions, such as the cloak and skin, and replacing the cloak region with the previously captured background [11].

2.1.4. Capture Frame

Here, the capturing and processing of each frame from the video stream is done. Ensuring that there is uniform lighting across the video, it is important that the captured frame undergoes brightness and contrast adjustments. To not make the cloak difficult to detect the brightness and the contrast is adjusted and by this the system ensures that the lighting conditions are consistent, hereby the cloak is easily distinguished from the rest of the environment.

2.1.5. Converts to HSV Color Space

The conversion from the BGR (Blue-Green-Red) color space to the HSV (Hue, Saturation, Value) color space is done for efficient segmentation of different parts of the image [12]. Decoupling of the chromatic content (hue) from the intensity (saturation and value) is necessary, so HSV color space is more suitable for color segmentation. This decoupling helps in better color detection, for the objects like “cloak”, which may have a specific hue, such as green in this case of our project.

2.1.6. Mask Creation

The cloak and skin tones in the frames correspond to identifying. Program success depends on Mask Creation because it selects areas of interest which include the targeted cloak along with skin areas in each frame. The created masks serve to isolate specific regions of the image frame before the background replacement takes place for achieving invisibility effects [13].

2.1.7. Cloak Mask

The cloak here, typically green in color, is detected by defining a range of acceptable hues in the HSV color space corresponding to the green color cloak. The binary mask is created by OpenCV's color segmentation function, where the pixels matching the specified color range are marked as background. The result is highlighting the cloak regions of the video frame that will later be replaced with the captured real-world background.

2.1.8. Mask Filtering

A suitable filtering approach must be applied to the combined mask since it contains unwanted noise together with artificial jagged edges that compromise invisibility. Outstanding segmentation results demand both filtering operations and mask smoothing methods to be applied in sequence. Morphological operations of dilation and erosion can remove small noise particles from the combined mask alongside edge smoothing functions.

Gaussian blurring and similar smoothing approaches should be used to blur the mask edges which creates a natural transition from the invisible areas to the frame. The likelihood of a visible outline around the cloak is reduced, ensuring a seamless invisibility effect.

The given code snippet uses the popular computer vision library in Python, OpenCV, to give an effect of invisibility or a change of background in a video or image [14].

As a consequence, this operation leaves only those parts of the frame where the inverse_mask is white, hence discarding those areas that correspond to the background.

invisibility = cv2.bitwise_and (background, background, mask=mask)

Here, it is creating an image called invisibility by applying a bitwise AND operation on the background image with a mask. It is a binary image where the white areas are to be retained in the background as 255, and the black areas are discarded as 0. Only parts of the background that have white parts of the mask will be shown in the output invisibility image. The areas in the mask are black and therefore black in the output.

Current background = cv2.bitwise_and (frame, frame, mask=inverse mask)

This line creates an image called current_background by performing a bitwise AND operation on the frame (which is usually the current frame from a video or a live camera feed) using an inverse_mask. The inverse_mask is the reverse of the original mask, so that areas that were white in the original mask are now black, and vice versa. As a consequence, this operation leaves only those parts of the frame where the inverse_mask is white, hence discarding those areas that correspond to the background.

final_output = cv2.add(invisibility, current_background)

This line combines the two images created in the previous steps: invisibility and current_background. The cv2.add function adds the pixel values of the two images together. Since invisibility contains the background where the mask was applied, and current_background contains the current frame where the mask was inverted, the result is an image that shows the current frame with the background replaced by the original background in the masked areas.

The final output will show the current frame with the background replaced, thus making it appear invisible or substituting the background.

2.2. Region Isolation and Replacement

Isolate cloak and non-cloak regions and blend to achieve invisibility. Region isolation and replacement are the final steps in the invisibility cloak program. These steps ensure the clock region is seamlessly replaced with the captured background while the rest of the scene remains unaffected.

Inverse mask: Create an inverse mask to isolate non-clock regions of the current frame.

2.2.1. Apply mask

- Use the original mask to extract corresponding region from the background frame
- Use the inverse mask to extract the remaining regions from the current frame.
- Combining the back region and current frame region using OpenCV add to produce the final output.[15].

2.2.2. Display and Interaction

The output in real time is displayed and the user control is allowed the invisibility effect is applied in the processed frame. · Show the processed frame with the "Invisibility Effect" applied.

- At regular intervals the frame rate is statically displayed.
- To allow the user to exit the program the "q" key is used.

3. Results and discussion

In the course of our project we implemented two primary components: the invisible croak and skin tone masking using the libraries. Each was evaluated based on functionality, accuracy and performance. It involves reading the script and processing a live video stream from the webcam and then detecting specific color features using the HSV color space. In terms of the invisible cloak it can be discussed as:

Implementation: utilization of the background subtraction techniques and object segmentation isolates the subject from the background allowing us to either replace the subject with the surrounding background or create an invisible effect by filling the area with the corresponding pixels from the background.

The "invisible effect" walked effectively in terms of a stable background. Here, the system successfully detects the subject and blends it with the background into the region making the cloth appear invisible. In complex conditions the varying lighting, moving objects in the background or busy scenes can struggle with the system with accuracy.

In terms of skin tone masking it can be described as:

Implementation: For the detection of the skin tone we used a combination of color space transformations like converting it to HSV and thresholding techniques to identify regions that matched human's skin tones. Once detection is done various modification is applied to these regions

Outcome: The masking technique showed promising results in images with a clearly defiant skin tone and the system accurately detected and masked the areas that match typical skin colors. However, challenges that limited the process arose with varied lighting conditions, shadows and non-human skin tones causing some areas to be incorrectly detected as scheme skin.

Both of these concepts performance was generally good and in terms of computational efficiency the skin tone masking And invisibility can be seen in real time performance by using efficient background subtraction methods hence, reducing the computational load.

3.1. Visual Results

- **Live Video Stream:** It shows the stream of video straight from the camera.
- **Cloak "Disappearance":** It is the detection and changing of regions with skin and the cloak to that of the recorded background, where the person can see the invisibility.

- **Refined Masks:** By applying morphological operations like opening, closing, and dilation it smoothened and improved the masks making them more real and cleaner.

3.2. Key Takeaways:

3.2.1. Detection of Skin Tones

The program uses two different HSV ranges to identify skin tones. The benefit of the approach used herein is that the HSV color space is less sensitive to changes in lighting condition than the conventional RGB space. In addition, skin tone detection by the program can be influenced by lighting conditions and it sometimes fails to segment the skin. Then, the HSV ranges would have to be adjusted to correct the issue.

3.2.2. Cloak Detection

Here, in our project we used only the one color which will be identified by the predefined HSV range for the color 'green'. Therefore, only that color cloak is made invisible. Another limitation of the cloak is it has to be green and if it is some other color (blue, red, and others) it may ruin the invisibility effect. The constraint is very evident from this method. The constraint is very evident that there is a requirement of a particular color for the cloak to be made effective **[16].**

3.2.3. Background Capture

At the beginning of the video session it is essential to first capture a stable background to successfully achieve the invisibility effect which is stored using the median of 30 frames. In dynamic backgrounds, the environment does not capture well due to the movement of objects and people. It creates a problem for the idealistic nature of the illusion.

3.2.4. Adjusting Brightness and Contrast

To enhance segmentation, the script adjusts the brightness and contrast of each frame so that the color features are more distinguishable. If the adjustments are too extreme, it may lead to overexposure or underexposure, which could degrade the quality of the invisibility effect, making it less accurate.

3.2.5. Morphological Refinement

Morphological operations, such as dilation and opening and closing operations, are used to widen the regions of interest to get a sharper and cleaner mask. However, it is very critical not to let dilation grow the masked areas over much because it inadvertently increases the size of the masked areas. So this has to be balanced. This needs to be optimized [17].

3.2.6. FPS

The frame rate is tracked every 30 frames to evaluate the processing efficiency. This makes mask processing and background extraction processes more complex and slower, making FPS go choppy while the video runs [18].

3.3. Challenges and Limitations

3.3.1. Lighting Sensitivity

One of the major challenges encountered in this project was dealing with the unpredictability of real world environments. Variations in lighting may greatly reduce the extent of accuracy in detection both for skin tone and cloaks. Since distinct lighting may lead to the failure of the system in recognizing specific skin tones or cloaks, the system may miss particular areas. Dynamic lighting adjustments or adaptive background subtraction could be used to make the system handle diverse lighting conditions. These can affect the accuracy of both the invisible clock and skin tone masking systems.

3.3.2. Color of the Cloak Limitation

In the present method, it is assumed that the cloak is of a certain color green for the invisibility effect to occur. The efficiency is low since only the cloaks colored green will effectively make the user invisible. This is something that can be solved by using dynamic color calibration tools where the users can, in real-time, adjust the range of color to suit their cloak [**19].**

Lighting variations, camera stability, and accurate color detection were addressed during the project to ensure that the invisibility effect is consistent. Overall, this work illustrates the power of computer vision techniques in achieving practical solutions while also providing a glimpse into how far we can push the boundaries of digital imaging in the future.

Here in this section we present the visual results obtained from implementing the invisible cloak and skin tone masking techniques illustrated with sample images highlighting the effectiveness and limitations of the methods used. The primary goal was to make the person in the image blend into the background achieving the illusion of invisibility. For the skin tone masking, the goal was to detect and mask the skin regions only to the respective areas.

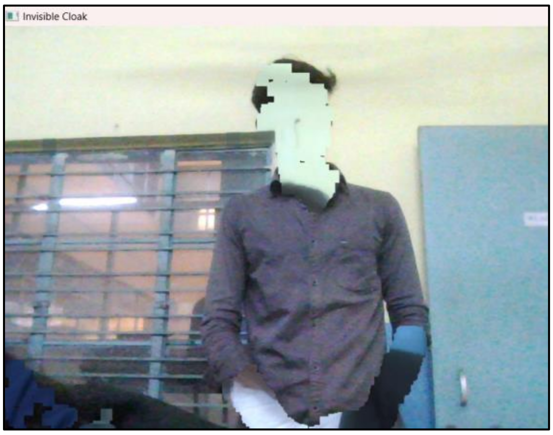

Figure 1 Successful skin tone masking on a simple background.

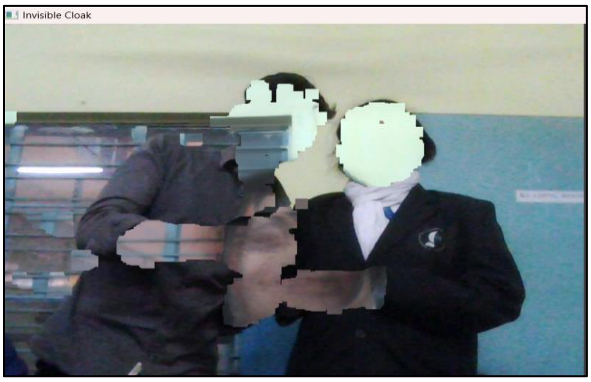

Figure 2 Successful skin tone masking on a simple background (two people).

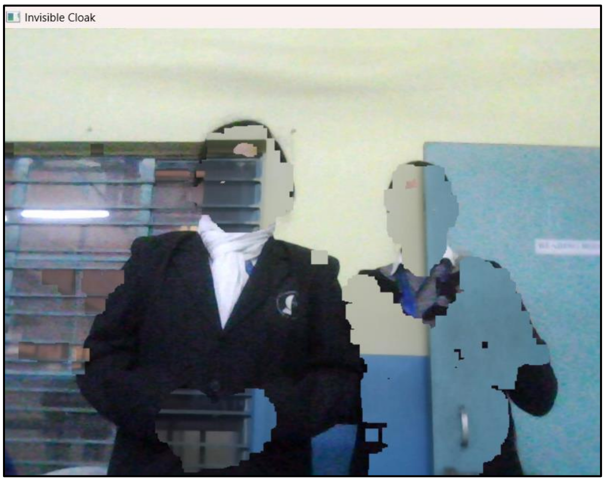

Figure 3 "Skin tone masking" and "invisibility effect" on two people (Here, the first person's skin is masked and the second person is holding a Green Cloak which has got invisible)

Future enhancements

Here in this project we basically create an invisibility effect for the green colored cloaks or cloth or any object and masking of the skin tones in the real time scenario. First off modifications remain in this aspect such as adding more features like blurring, mosaic, applying filters and enhancing the working more. In future journeys, there will be improvement in performance, by focusing on refining the algorithms integrating deep learning models as a result enhancing accuracy and generalization. In real-time video processing and optimization for resource constraints devices could broaden the applicability of these techniques [20].

4. Conclusion

Through this research scientists proved the successful development of an invisible cloak technology which integrates image processing methods to provide both visual invisibility and privacy protection capabilities. These outcomes show how these technologies can boost privacy levels as well as communication security and enjoyment-based capabilities. The research brings societal advantages through new privacy development for virtual and physical spaces that ultimately creates advanced and accessible surveillance measures in augmented reality systems.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Szeliski, R. (2022). Computer vision: algorithms and applications. *Springer Nature*.

[2] Burger, W. &. (2022). Digital image processing: An algorithmic introduction. *Springer Nature.*

[3] Piccardi, M. (2004). "Background subtraction techniques: a review,". *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), The Hague, Netherlands,*, pp. 3099-3104 vol.4, doi: 10.1109/ICSMC.2004.1400815.

[4] Vågnes, Ø. (2021). Images and Invisibility. *The Palgrave Handbook of Image Studies*, 533-545.

[5] Howse, J. J. (2016). Opencv: computer vision projects with python. *Packt Publishing Ltd.*

[6] Zivkovic, Z. (2004, August). Improved adaptive Gaussian mixture model for background subtraction. *In Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*, (Vol. 2, pp. 28-31). IEEE.

[7] Pitas, I. (2000). Digital Image Processing Algorithms and Applications. *John Wiley & Sons Inc google scholar, 2,* 133-138.

[8] Fernandes, A. F. (2020). Image analysis and compute

[9] Wiley, V. &. (2018). Computer vision and image processing: a paper review. *International Journal of Artificial Intelligence Research, 2(1)*, 29-36.

[10] Burger, W. &. (2022). Digital image processing: An algorithmic introduction. *Springer Nature.*

[11] Kulkarni, N. (2012). Color thresholding method for image segmentation of natural images. *International Journal of Image, Graphics and Signal Processing, 4(1)*, 28.

[12] Soille, P. &. (2004). Erosion and dilation. Morphological Image Analysis: *Principles and Applications*, 63-103.

[13] Burger, W. B. (2009). Principles of digital image processing (Vol. 111). *London: Springer.*

[14] Cheng, H. D. (2001). Color image segmentation: advances and prospects. *Pattern recognition, 34(12)*, 2259-2281.

[15] I. Culjak, D. A. (2012). "A brief introduction to OpenCV,". *2012 Proceedings of the 35th International Convention MIPRO, Opatija, Croatia*, pp. 1725-1730.

[16] Saravanan, G. Y. (2016, April). Real time implementation of RGB to HSV/HSI/HSL and its reverse color space models. *In 2016 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0462-0466. IEEE

[17] P. Salembier, P. B. (June 1996). "Morphological operators for image and video compression,". *IEEE Transactions on Image Processing, vol. 5, no. 6*, 881-898.

[18] Kindberg, L. (2020). A frame differencing algorithm that allows for small camera movementsr vision applications in animal sciences: an overview. *Frontiers in Veterinary Science, 7*, 551269.

[19] Yilmaz, A. J. (2006). Object tracking: A survey. *Acm computing surveys (CSUR), 38(4)*, 13-es.

[20] Said, K. A. (2021, October). Analysis of image processing using morphological erosion and dilation. *In Journal of Physics: Conference Series*, Vol. 2071, No. 1, p. 012033