IV. Experimental results and Discussion

We experimented our proposed method in two way. First one is with tracking and second one is without tracking. With tracking can recognize the hand in any region and without tracking needs to be place the hand in a specific area. The IDE we used to build the program is visual studio enterprise 2017 edition. The program was running on a Lenovo laptop consisting of corei5 7th generation, 2.70 GHz processor and the operating system was Windows 10. A pre-installed webcam is used to recognize the hand. The program was developed using EmguCV. EmguCV is a cross-platform image-processing library [10]. It is closely related to OpenCV because EmguCV is a .NET wrapper to OpenCV. We can say EmguCV is OpenCV in .NET. In without tracking the program was written using Python with NumPy along with OpenCV OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library [11]. NumPy is the fundamental package for scientific computing with Python [12].

As we mentioned we were working with two types of program, we used the webcam to recognize the video. Four people were organized to perform the task that they will show their finger from one to one and some other posture. They sat before the camera and perform the task as instructed. As it is a real time application, so there is no still image dataset. We used our own self-made video dataset to recognize the hand gesture. Each were instructed to perform the task for five minutes. They raise their finger from one to five in both hand and keep the hand steady for 5 seconds each. They done the process five times.

The experiment was occurred in home environment. We experimented on various background. Firstly, we experimented on simple plain background. Than we choose a background that’s match the skin color to examine that how perfect the process can segment the hand. Than we experimented on a very complex background consists of many things. No external light is used to focus the hand. Typical room light was the light source.

Our research is basically subdivided into three phases. They are recognition of hand gesture; recognition of AR marker and final step is to generate the geometrical shape with interaction process. As previously mentioned, we only worked with the numbers of fingers to testify that whether the system can recognize hand or not. So, we have conducted experiments on how accurately the program is recognizing enlisted postures. To further extend the results we have tried experiments with two types of program-

*I. With Tracking:* This program is able to track the hand from any position and extract it from other parts of body as the skin types are quite similar for hands and other parts of the body. It then puts AR markers around the hand. The approach shown is based on skin detection and convex hull and defects computation. So, for checking its accuracy we run this program in various backgrounds. The base of the program is implemented by Luca Del Tongo. In its base the program was able to take specified videos of Hand gestures as input. When we first collect the code, it only takes input from a pre-defined video dataset. We have modified it so that it would take input from webcam. There was a grabber function which takes input from video, we made it global to take input from any kind of desktop or laptop facilitated with webcam.

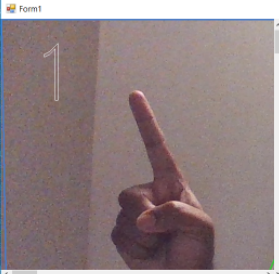
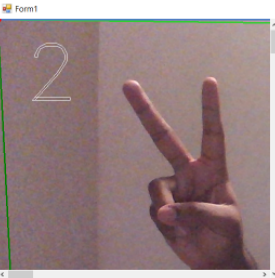


Fig. 4. Recognition in Simple Background

In Fig. 4 Regarding Simple (Non-Complex) background the program is able to track and recognize the gesture. As the color pallet of the background is different from skin color pallet it shows no problem in tracking the hand posture. So, in terms of simple/non-complex background this program holds potential.

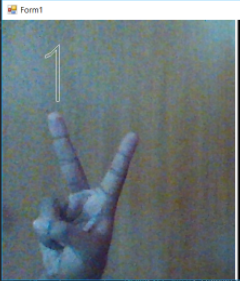
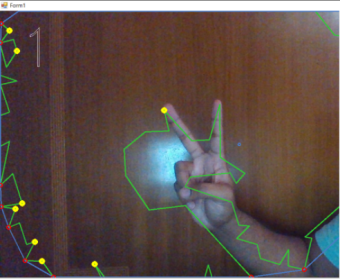


Fig. 5. Recognition in Complex Background

In Fig. 5 Regarding Complex background, it was observed that the program had great difficulties in tracking the hand. Rather most of the times it would track the background as it may have similar color pallet that of skin. Hence it rarely gave the correct response according to the posture. Moreover, it overlaps the hand skin with the same color background and gives falls recognition.

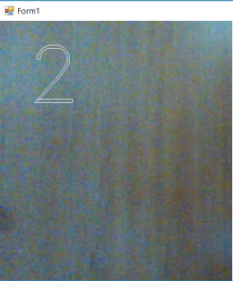
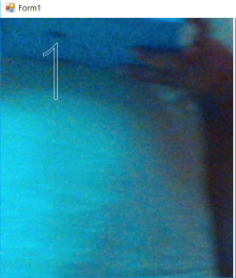


Fig. 6. Incorrect Recognition in Complex Background

In Fig. 6 Despite having no hand displayed on camera the program still responded in complex background because of matching color pallet with skin.

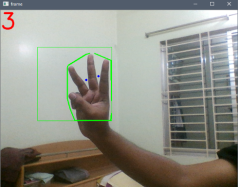
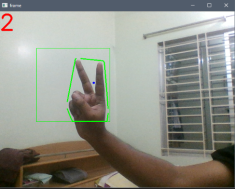
According to the discussion above it is apparent that it has a high accuracy rate in recognizing and tracking hand gestures when the background color pallet didn’t match with the skin color pallet. But it was mostly unable to recognize the hand when the background color pallet matched with the skin color pallet.

Table 2. Performance comparison (EmguCV)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Background Type | Accuracy = Successful recognition/Total no of attempts | Error  = Unsuccessful recognition/ Total no of attempts | Tracking Rate = Assumed based on the marked area | System |
| Simple (Non-Complex) | 90% | 10% | 65% | Core i5 CPU with 2.70 GHz processor |
| Complex | 20% | 80% | 10% |

In Table 2 we illustrated what we have observed from the performance of the program. The program was run in Core i5 CPU with 2.70 GHz processor system. We operated the program in two different types of background. One was Simple background which did not have any large mix of color pallet and also did not match with skin color pallet. In this case we used white wall as a background as it is the simplest and the hand is in a great contrast with the wall to recognize. Another was Complex background which had large mixture of color pallets and/or matched with skin color pallet like doors or windows. For each background we have tried gestures 20 times. And as mentioned before we raised our fingers from one to five in both hand and keep the hand steady for 5 seconds each. It was observed that 65% of the time the program was able to track the hand in Simple background. But it had a hard time tracking the hand in Complex background which was 10% of the time. For Simple background the program was able to recognize the gestures successfully 18 times but for Complex background it only successfully recognized 4 times.

*II. Without Tracking:* This program is able to track the hand from a specific position where only the hand will be shown. The program will recognize the hand from a specified area hence it doesn’t need to track. So, for checking its accuracy we run this program in various backgrounds. The base of the program is implemented using Python 2 which is backdated. So, we have converted the program from Python 2 to Python 3 as it is more updated. This also opens more capacity for future improvement of the program.



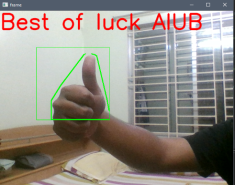


Fig. 7. Recognition in Simple Background

In Fig.7 Regarding Simple (Non-Complex) background the program is able to recognize the gesture when the hand is displayed in the specified area. As the color pallet of the background is different from skin color pallet it shows no problem in responding according to the hand postures. So, in terms of simple/non-complex background this program shows great results.

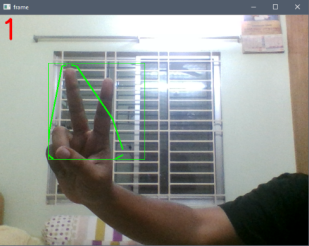
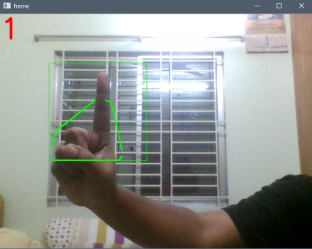


Fig. 8. Incorrect Recognition in Complex Background

In Fig. 8 Regarding Complex background, it was observed that the program had few difficulties in tracking the hand. Though with low accuracy rate the program was able to recognize the hand postures and responded accordingly.

The experiment shows promising result in both simple and complex background. Though it showed lower accuracy rate in complex backgrounds then the simple backgrounds it was still higher than the program with tracking based on complex back ground.

Table 3. Performance comparison (Python with NumPy)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Background Type | Accuracy | Error | Tracking Rate | System |
| Simple (Non-Complex) | 95% | 5% | None | Core i5 CPU with 2.2 GHz processor |
| Complex | 70% | 30% |

In Table 3 we illustrated what we have observed from the performance of the program. The program was run in Core i5 CPU with 2.2 GHz processor system. We operated the program in two different types of background. One was Simple background which did not have any large mix of color pallet and also did not match with skin color pallet. In this case we used white wall. Another was Complex background which had large mixture of color pallets and/or matched with skin color pallet like doors or windows. For each background we have tried gestures 20 times. And as mentioned before we raised our fingers from one to five in both hand and keep the hand steady for 5 seconds each. For Simple background the program was able to recognize the gestures successfully 19 times and for Complex background it successfully recognized 14 times. But unlike the other program this program did not have the tracking feature. So, there is no tracking rate for this program.

Our first way to recognize the hand gesture is a tracking-based method. In this method wherever the hand is it can be extracted from the background. In Table 2, the method got 90% accuracy and 10% of error in simple background. Complex background reduces the accuracy to 20% and as produce an error rate of 80%. On the other hand, from Table 3, another method that is without tracking produces a 95% accuracy in simple background with 5% of error and in complex background produces 70% of accuracy including 30% of error rate. It can be seen that both the method works well in simple background but when it comes to complex background, in both case of with and without tracking a dramatical change arise. In non-tracking method the outcome is quite desirable but the main problem of the method is the hand must be in a pre-defined position to be tracked which is quite unexpected during teaching session. So, if the hand is not in the expected position the recognition will fall as well as the whole system. Now come to the tracking method, though the method is not producing a good result but its main advantage is that the hand can be recognize in any position. So, what is necessary to change factors like light, color range and others to be modified to gain optimum result with the tracking method.

**Comparison of Experimental Results with Previous Research Results**

Table 4. Performance comparison between modified method and previous method

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | Accuracy / Recognition Rate | Error Rate | Processing Time (In Second) | Name of Background | Processing Speed in Frame Per Second |
| The “extrusion” technique [16] | 80% | 20% | The AR application took 53.8 s  GeoGebra took 71.95 s  Cabri3D took 47.67 s | Not mentioned | (depends on processor) |
| Modified extrusion technique (Tracking) | 90% | 10% | The AR application took 9 s to recognize the hand gesture. | Simple | The process used 25% of processor  in intel corei5 7th generation laptop. |
| 20% | 80% | Complex |
| Modified extrusion technique (Non-Tracking) | 95% | 5% | Simple |
| 70% | 30% | Complex |

Fig. 9. Accuracy among the modified techniques and previous technique

Fig. 10. Error among the modified techniques and previous technique

From the above-mentioned Table. 4 and the fig. 9 and fig. 10, it is clearly noticeable that the extrusion technique has got the highest position with the accuracy rate of 80%. The background was not disclosed in the research. Our proposed modified extrusion technique without tracking got the second place with 95% of recognition of accuracy rate when it was simple background. The recognition rate become 70% when the background become complex. Dramatically the accuracy rate falls to 20% when we applied modified extrusion technique with tracking. On the contrary it was 90% when the background was simple. Opposite scenario was seen in terms of error rate. As the recognition rate is higher in the extrusion technique, the error rate is lower. Non-tracking modified extrusion technique got 5% and 30% error rate in simple and complex background respectively. 10% error occurred with modified extrusion technique with tracking, but the amount rises when the background became complex and the rate was 80%. The extrusion technique took 58.3 s to process the whole system. But we only deal with the recognition of hand, not the whole system. Until now our system took 9s to complete its recognition task. Extrusion technique can be called a process dependent technique as the processing speed depends on the processor.

References:

[10]."Emgu CV". (2019, January 23). Retrieved from <http://www.emgu.com/wiki/index.php/Main_Page>.

[11]."OpenCV". (2019, January 23). Retrieved from https://opencv.org.

[12]."NumPy". (2019, January 23). Retrieved from<http://www.numpy.org>.