MATCHCUT ASSIST: A MOBILE SYSTEM TO AUTOMATE MATCHCUT PROCESS USING COMPUTER VISION

[1]Harrison Zhou, [2]Bhargav Parekh and [2]Yu Sun

[1]Irvine High School, Irvine, CA 92604
[2]Department of Computer Science
California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

In film industry, camera angle and smooth transition between shots are very important and time-consuming manual process. Match Cut is most frequently used technique in this industry to achieve seamless transition. This paper proposes an app that offers solution to assist match cut technique using edge detection algorithm on mobile device. Users can use the app to take two different shots which can be used to match cut in future. It is helpful to take different shots without manual efforts and heavy equipment.

KEYWORDS

Match Cut, Android Application, OpenCV, Photography, Photo Edit, Canny Edge Detection

1. INTRODUCTION

As of the recent boom in people’s interest in movies, the film industry has never been more popular. To make a good film with the least amount of money spent is ideally what every company wants, but the process of making the work makes their goal difficult; just as simple as setting up and talking to actors takes a lot of time, and time is the factor that determines how much one has to spend to pay for its crew, so the more time is spent, the more money the company has to take from its budget. But to make a good film requires a lot of time for there are multiple factors link to it. In this case we will be looking at positioning the camera; one way to make the film more interesting is to have more angles and use techniques so that there won’t be always the same shots, which would make the film look dull and bore the audience. The one example we would be looking at is the match cut techniques, where a cut from one shot to another where the two shots are matched by the action or subject and subject matter, but the issue is that it requires a lot of time to set up the camera to match the subject from the second shot exactly with the first. Some of the early solution to that is to have first after having taken the first picture of your subject, you would have to select exactly parts that you wanted, which then the subjects would be cropped and overlaid on the next camera frame with lower opacity. But the Problem with that is that it is too bulky, hard to use, and inaccurate, which led to the idea of making it automatic rather than having the users operating it manually. In this paper, we used the same techniques but with the replacement of just overlaying the images after putting it through edge detection, which made the image overall easier to see, locate, and more efficient as it eliminates the selection phase.

Natarajan Meghanathan et al. (Eds) : NLP, ARIA, JSE, DMS, ITCS - 2019
pp. 75-80, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.91209

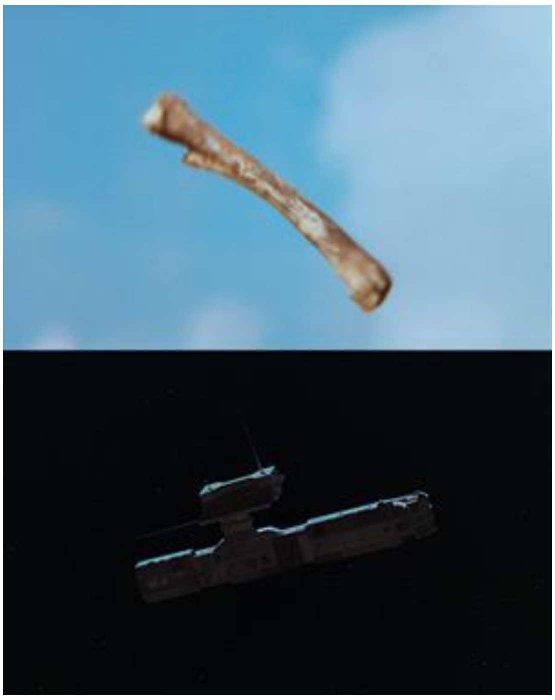


Figure 1: Match Cut application example from movie “2001: A Space Odyssey”

Match Cut often involves a graphics match, a smooth transition between scenes and comparison between elements in both shots. Figure 1 is a good example of Match Cut technique from “2001: A Space Odyssey” (Sci-Fi movie).

So as one of the solutions would be an assistance to help make setting up and changing angles more efficiently, and in this case, an app that marks out the outlines of the subjects for you with one click of a button.

2. MOTIVATION AND CHALLENGES

As a video editor and an After-Effect user, we often use plug-ins when operating. The plug-ins are essentially premade codes that once installed could save hours by presenting and providing the effects or results one needed instead of spending the extra time to make it himself. With the process eliminated extra time is created for more available changes and improvements in other areas and essentially making the entire film better by improving its quality.

One challenge is to decide how to display or overlay the first image to when taking the second picture, of what way to be able to “select” and show the wanted subject to the user, but just lowering the opacity and overlaying the first image would blend with the second image and making it difficult to set apart the wanted and unwanted parts of the shot, and it would be difficult and inconvenient for the user to manually “edit” the parts he/she wanted (for example, not enough space to work with)

3. SOLUTION

The solution we came up with is to use edge detection. It would just look at the shapes of the first image taken, and draw out shapes then to overlay the outlines of the subjects, so then the pictures won’t collide with each other as there won’t be a blend of colors, and with adjustment of the detections won't be showing all the unnecessary details in the pictures, making it easier to read and be matched on.

First is the loading screen, it takes about 4 seconds for the app to launch, and once it is first launched, it would ask the user for access for the camera, as it needs it later to take pictures, and access for it to create its own folder in the picture album, as well as access to take the pictures taken and import it from the album to the app.

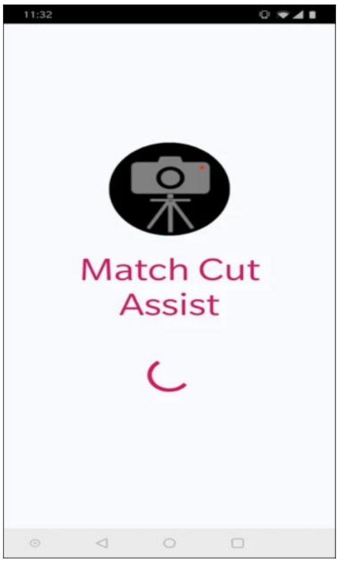


Figure 2: Screen 1 - Loading page with permission

The second screen is the start screen of the app, it is there to start the entire process of taking the first image, and it would also be the page the user will return to once they are done with their matches and wants to take another.

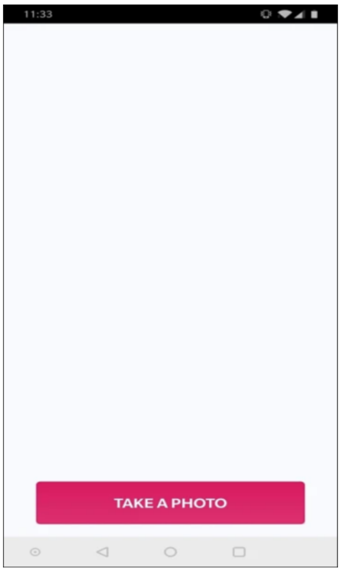


Figure 3: Screen 2 - Home and starting page of application

The third screen is the camera, with permission to access it uses the phone's camera and will take the first image then transfer it to the folder it created in the album.

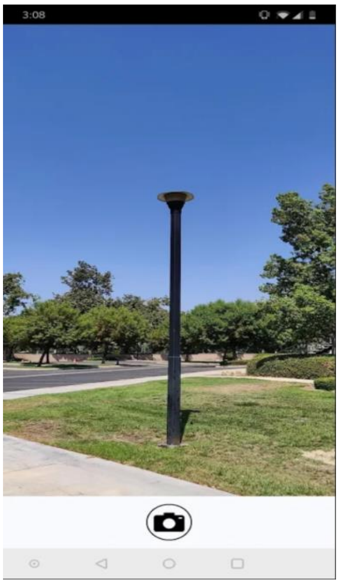


Figure 4: Screen 3 – Camera to capture first shot

The fourth screen is a confirm page, it re-inputs the image from the phone album after putting through the process of Edge detection using OpenCV library, it shows the outlines of the subjects from the image first taken, and is given options of either confirming "yes" as if the image is displaying satisfies the user and moves onto the next page, or "no", then it takes the user back to Camera 1 for the picture to be re-taken.

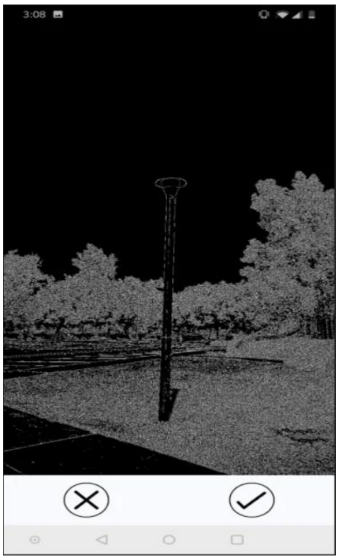


Figure 5: Screen 4 – Confirmation page with edge detection

The fifth screen is again the phone camera but with the overlay of the outlines of the subjects from the first picture taken so that the subject in the subject in the second shot could be placed in the correct spot.

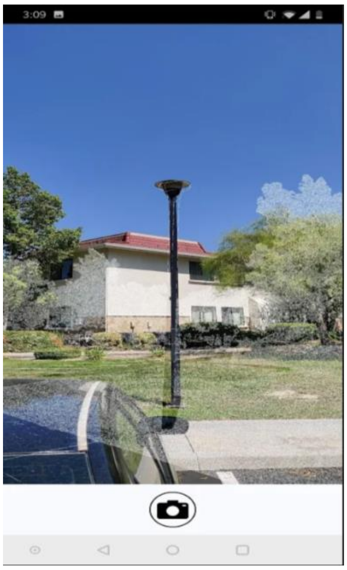
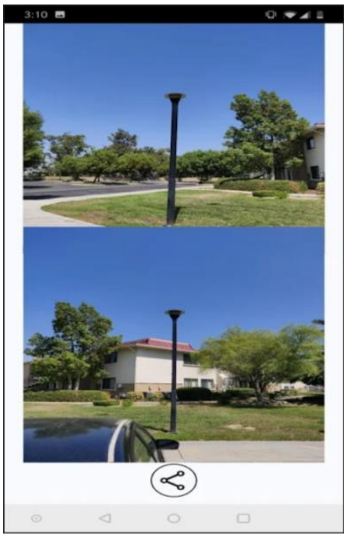


Figure 6: Screen 5 – Camera with edges to capture second shot

The last screen shows the comparison of the two pictures that you took, it would give you a choice of whether or not to share them over different channels, and right afterwards goes back to home screen (screen 1) to be ready for another set of match cuts.



Screen 6 – Result page with both images and share button

4. RELATED WORK

In the field of movie creation and editing, match cut is highly used technique. In one article “Scene completion using millions photographs”, the algorithm patches up holes in images by finding similar image regions in the database that are seamless.[11] So this kind of process requires heavy processing to match pixel by pixel and requires manual intervention too.

Another paper based on scene creation with smooth transition “Testing the developmental foundations of cinematic continuity”, heavily uses match cut to create continuous video with various background scene.[12]

5. CONCLUSIONS AND FUTURE WORKS

In this development, we programmed an app that could help save time and money for film companies as it uses our app to achieve their goal. The app could read and detect the edges of the image taken, put it into lines and shapes so that it would be easier and more accurate for the user to successfully get the match cuts they need.

In the future, instead of capturing first shot app can incorporate option to import image from storage. Also feature of merging (transition) between shots can be achieved using openCV library in the same app.

REFERENCES

[1] Wikipedia contributors. (2019, July 31). Match cut. In *Wikipedia, The Free Encyclopedia*. Retrieved 06:34, August 15, 2019, from https://en.wikipedia.org/w/index.php?title=Match_cut&oldid=908680395

[2] Bao, P., Zhang, L., & Wu, X. (2005). “Canny edge detection enhancement by scale multiplication”. *IEEE transactions on pattern analysis and machine intelligence*, *27*(9), 1485-1490.

[3] May, J. (2000, December). Perceptual principles and computer graphics. In *Computer Graphics Forum* (Vol. 19, No. 4, pp. 271-279). Oxford, UK and Boston, USA: Blackwell Publishers.

[4] Baggio, D. L. (2012). *Mastering OpenCV with practical computer vision projects*. Packt Publishing Ltd.

[5] Howse, J. (2015). *Android application programming with OpenCV 3*. Packt Publishing Ltd.

[6] Topal, C., Akinlar, C., & Genc, Y. (2010, August). Edge drawing: A heuristic approach to robust real-time edge detection. In *2010 20th International Conference on Pattern Recognition* (pp. 2424-2427). IEEE.

[7] Canny Edge Detector. Retrieved 08:11, August 08, 2019, from https://docs.opencv.org/master/da/d5c/tutorial_canny_detector.html

[8] Muhammad, A. (2015). *OpenCV android programming by example*. Packt Publishing Ltd.

[9] Camera API Documentation: Android Developers. Retrieved 11:45, August 06, 2019, from https://developer.android.com/guide/topics/media/camera

[10] OpenCV in advanced Android development: Edge detection - Zaven: Blog, (2017, July 28). Retrieved from https://blog.zaven.co/opencv-advanced-android-development-edge-detection/

[11] Hays, J., & Efros, A. A. (2007). Scene completion using millions of photographs. *ACM Transactions on Graphics (TOG)*, *26*(3), 4.

[12] Smith, T. (2018). Testing the developmental foundations of cinematic continuity.