

# **CSE578: Assignment #1**

Due on Tuesday, January 12, 2016

*Dr. Anoop Namboodiri*

**Abhinav Moudgil**

**20331039**

## Problem 1

*Video  $\leftrightarrow$  Images:* Write a program to convert a given video to its constituent images. Your output should be in a specified folder. Write another program that will merge a set of images in a folder into a single video. You should be able to control the frame rate in the video that is created.

### Video $\rightarrow$ Images:

Following is the code that captures each frame of video (path to video file is passed as an argument) and saves it as an image.

```

1 //
2 //  main.cpp
3 //  Ass1
4 //
5 //  Created by Abhinav Moudgil on 10/01/16.
6 //  Copyright 2016 Abhinav Moudgil. All rights reserved.
7 //
8
9 #include <iostream>
10 #include <opencv2/opencv.hpp>
11 #include <opencv2/highgui/highgui.hpp>
12 #include <stdio.h>
13
14 using namespace std;
15
16 int main(int argc, const char * argv[]) {
17
18     if (argc !=2)
19     {
20         std::cout << "USE: " << argv[0] << " <video-filename>" << std::endl;
21         return 1;
22     }
23
24     //Open the video that you pass from the command line
25     cv::VideoCapture cap(argv[1]);
26     if (!cap.isOpened())
27     {
28         std::cerr << "ERROR: Could not open video " << argv[1] << std::endl;
29         return 1;
30     }
31     cout << argv[0] << endl;
32     int frame_count = 0;
33     bool should_stop = false;
34     while(!should_stop)
35     {
36         cv::Mat frame;
37         cap >> frame;
38         if (frame.empty())
39         {
40             if (frame_count != 0)
41                 should_stop = true;
42             continue;
43         }
44         char filename[128];
45         sprintf(filename, "frame_%06d.jpg", frame_count);
46         cv::imwrite(filename, frame);
47         frame_count++;
48         cout << "frame - " << frame_count << endl;
49     }
50
51     return 0;

```

52 }

**Video ← Images:**

Below code takes images (specified in particular folder) and converts it back to video. By tweaking with arguments passed to VideoWriter object, we can control the various parameters of video like frame rate, video format etc.

```
1 //
2 //  main.cpp
3 //  videoCreator
4 //
5 //  Created by Abhinav Moudgil on 10/01/16.
6 //  Copyright 2016 Abhinav Moudgil. All rights reserved.
7 //
8
9 #include <iostream>
10 #include <opencv2/opencv.hpp>
11 #include <stdio.h>
12
13 using namespace std;
14 using namespace cv;
15
16 int main()
17 {
18     Mat frame;
19     VideoCapture cap("/Users/abhinavmoudgil95/Library/Developer/Xcode/DerivedData/Ass1-
20     daefxswifrukddqlgakasnftizy/Build/Products/Debug/frame_%06d.jpg");
21
22     VideoWriter writer("/Users/abhinavmoudgil95/Library/Developer/Xcode/DerivedData/Ass1-
23     daefxswifrukddqlgakasnftizy/Build/Products/Debug/created.avi", CV_FOURCC( 'M', 'J', 'P', 'G'
24     ), 30, Size(640, 360));
25     int i = 0;
26     while(1)
27     {
28         cout << i << endl;
29         i++;
30         cap >> frame;
31         if (frame.empty())
32             break;
33         writer.write(frame);
34     }
35     return 0;
36 }
```

## Problem 2

*Capturing Images: Learn how to capture frames from a webcam connected to your computer and save them as images in a folder. You may use either the built-in camera of your laptop or an external one connected through USB. You should also be able to display the frames (the video) on the screen while capturing.*

Each frame captured by camera is saved as an image and displayed on the screen side by side. Code runs upto 100 frames and terminates. Highgui functions in OpenCV (for example, imshow() here) takes some time to process its event loop and hence need a call of waitKey() function. Hence the output displayed on the screen is a bit slow than the actual streaming of the camera.

```
1 //
2 //  main.cpp
3 //  inbuiltCamera
```

```
4 //
5 // Created by Abhinav Moudgil on 10/01/16.
6 // Copyright 2016 Abhinav Moudgil. All rights reserved.
7 //
8
9 #include <iostream>
10 #include <opencv2/opencv.hpp>
11 #include <opencv2/highgui/highgui.hpp>
12 #include <stdio.h>
13
14 using namespace cv;
15 using namespace std;
16
17 int main(int, char**)
18 {
19     VideoCapture cap(0);
20     if(!cap.isOpened())
21         return -1;
22
23     Mat edges;
24     int frame_count = 0;
25     for(; frame_count <= 100;)
26     {
27         Mat frame;
28         cap >> frame; // get a new frame from camera
29         imshow("Live Camera", frame);
30         char filename[128];
31         sprintf(filename, "cameraframe-%06d.jpg", frame_count);
32         imwrite(filename, frame);
33         frame_count++;
34         cout << frame_count << endl;
35         if(waitKey(30) >= 0) break;
36     }
37
38     return 0;
39 }
```

## Problem 3

*Create an interesting composite of two videos using this technique, possibly with one video including yourselves.*

Path of video files are passed as arguments (any number of videos can be passed) whose backgrounds should be replaced by an image (specified in particular folder whose path should be mentioned in the code). For each video:

- Frame is processed one by one.
- Each image frame is converted to HSI color space.
- Green background is segmented by selecting Hue within appropriate range and we get segmented image that has binary values corresponding to foreground and background.
- For each pixel in the background image (that should be replaced with), if corresponding pixel value in the segmented image is 0, then we replace the color of pixel with the image frame color.

**Results:**

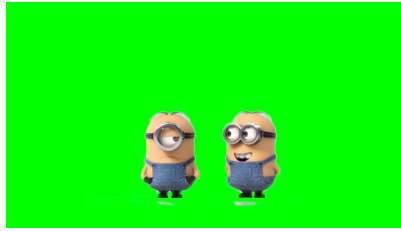


Figure 1: Original frame



Figure 2: Segmented image



Figure 3: Output

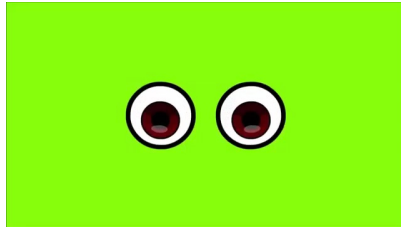


Figure 4: Original frame



Figure 5: Segmented image



Figure 6: Output

```

1 //
2 //  main.cpp
3 //  chromaKeying
4 //
5 //  Created by Abhinav Moudgil on 10/01/16.
6 //  Copyright 2016 Abhinav Moudgil. All rights reserved.
7 //
8 #include <iostream>
9 #include <opencv2/opencv.hpp>
10 #include <opencv2/highgui/highgui.hpp>
11 #include <stdio.h>
12
13 using namespace std;
14 using namespace cv;
15
16 void makeVideo(int i, const char * argv[])
17 {
18     VideoCapture cap(argv[i]);
19     if (!cap.isOpened())
20     {
21         std::cerr << "ERROR: Could not open video " << argv[1] << std::endl;
22         return;
23     }
24     char filename[500];
25     sprintf(filename, "/Users/abhinavmoudgil95/Library/Developer/Xcode/DerivedData/Ass1-
26     daefxswifrukcdqlgaksnfizy/Build/Products/Debug/chromakeyed%d.avi", i);
27     VideoWriter writer(filename, CV_FOURCC('M','J','P','G'), 30, Size(640, 360));
28     int frame_count = 0;
29     bool should_stop = false;
30     Mat im, backhsv;
31     Mat background = imread("/Users/abhinavmoudgil95/Documents/OpenCV/Ass1/Ass1/chromaKeying
32     /background.jpg");
33     cvtColor(background, backhsv, CV_BGR2HSV);
34     vector<Mat> channels(3);
35     vector<Mat> segmented(3);
36     while(!should_stop)
37     {
38         cv::Mat frame, img_hsv, nseg, seg;

```

```
37     cap >> frame;
38     if (frame.empty())
39     {
40         if (frame_count != 0)
41             should_stop = true; // End of the video
42         continue;
43     }
44     cvtColor(frame, img_hsv, CV_BGR2HSV);
45     if (frame_count == 0)
46     {
47         Size s = frame.size();
48         resize(backhsv, im, s);
49     }
50
51     split(img_hsv, channels);
52     double g = img_hsv.at<Vec3b>(1,1)[0];
53     cout << g << endl;
54     inRange(channels[0], double(g) - 4, double(g) + 4, segmented[0]);
55     segmented[1] = segmented[0];
56     segmented[2] = segmented[0];
57     merge(segmented, seg);
58     bitwise_not(seg, nseg);
59     Mat x1, x2;
60     Mat res, result;
61     bitwise_and(seg, im, x1);
62     bitwise_and(nseg, img_hsv, x2);
63     bitwise_or(x1, x2, result);
64     cvtColor(result, res, CV_HSV2BGR);
65     writer.write(res);
66     char filename[128];
67     sprintf(filename, "segmented%d_%06d.jpg", i, frame_count);
68     imwrite(filename, seg);
69     sprintf(filename, "frame%d_%06d.jpg", i, frame_count);
70     imwrite(filename, frame);
71     sprintf(filename, "output%d_%06d.jpg", i, frame_count);
72     imwrite(filename, res);
73     frame_count++;
74     cout << frame_count << endl;
75 }
76 }
77
78 int main(int argc, const char * argv[]) {
79     if (argc < 2)
80     {
81         std::cout << "USE: " << argv[0] << " <video-filename>" << std::endl;
82         return 1;
83     }
84     int i;
85     for (i = 1; i < argc; i++)
86         makeVideo(i, argv);
87     return 0;
88 }
```