Progress Report 2
Group-13 project-4

Jack Carlson - JAC5982

Stephen Arnett - SJA696

Chris Renard

Project Topic:

Interactive video overlaying. We want to extend the tracking functionality developed in project 3 to allow a user to overlay a video seamlessly with image resources from a toolbar. As an extension, we will hopefully be able to track the added object as well to make sure it render properly as well within the viewing screen of the video and possible track more than one face with these objects at the same time.

Planned Meeting Schedule:

We plan on meeting after every class from now until December 2nd. We will also meet on weekends as needed and if possible. Also, a shared repo has been created so that if we cannot meet we can still share what we have done with the rest of the group

Objective 1 (Half-Finished) : be able to accurately track a face in video in real time

Result (updated): have the app be able to track at least a 100 points in 3D space on the face in real time. (this is implicitly completed in the method we chose to do the face tracking with I believe)

Result (updated) : be able to track a face with 60% accuracy in an offline situation, such a high rate of accuracy for videos over a long period of time, and in real time is very difficult given the current knowledge of tracking, also we added tracking more than one feature, which complicates the accuratly depending on the speed of the movement in the video

Sample code for tracking:

```
def track_face(video):
    """As track_ball_1, but for face.mov."""
     face_cascade = cv2.CascadeClassifier('haarcascades/haarcascade_frontalface_default.xml')
     eyes_cascade = cv2.CascadeClassifier('haarcascades/haarcascade_eye.xml')
     mouth_cascade = cv2.CascadeClassifier('haarcascades/haarcascade_mcs_mouth.xml')
    nose_cascade = cv2.CascadeClassifier('haarcascades/haarcascade_mcs_nose.xml')
    cap = video
    toList = []
    t = 0
    while(1):
        ret, image = video.read()
        if ret is True:
            gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
            faces = face_cascade.detectMultiScale(gray, 1.3, 5)

            if len(faces) == 0:
                toList.append(toList[len(toList) - 1])

            for (x, y, width, height) in faces:

                if faces.size == 8:
                    a, b, c, d = faces[0]
                    toList.append((a, b, a + c, a + d))
```

```
                break
            toList.append((x, y, x + width, y + height))
        else:
            break
    cv2.destroyAllWindows()
    video.release()
    return toList
```

Estimated effort: minimal

Estimated time: 1 day, just in case we have to go back and track more points, it would not take long to do so in the code.

Objective 2: Have transparent images overlaid in the video and move with relation to it.

Result: Identify the points that are being tracked and map them onto the image of choice so that wherever those points go the image also moves. (Update: we are still currently working on this)

Estimated effort: moderate

Time Estimate: about 5 days, once we get the starter code running it should be relatively simple to get the mappings of to

Objective 3: Extend the functionality to a mobile app that allows users to modify a video for vine or instagram.

Result: Have a working Android or iOS app that allows interactive video editing and implements the same functionality in Objectives 1 and 2 on the mobile platform.
Result: The user will be able to record a video and drag and drop images onto the screen during playback, which will follow the tracked object in the video.

Estimated effort: Large

Estimated time for completion: 1 weeks, this seems to be the hardest part of the project as we have no real starting code for the app to work, we may need to change this as we draw closer 1 the deadline

Written Report (update): the report has yet to be started and will be one of the last things we do in this project as to use our project is pretty code intensive. However, the data we using to make the code is being kept in a log so that we can later make it easy on ourselves.

Overall estimated time: 2 weeks, until end of semester. We got started on this project rather late and so we are pretty far behind the other groups as a result. Estimated date of completion Dec 2 (due date)

Schedule:

11/24 Hope to have objective 2 done
12/1 Hope to have objective 3 done