COMP3419

GRAPHICS AND MULTIMEDIA

ASSIGNMENT PROJECT

INTELLIGENT ANIMATION

October 21, 2018

Francisco Manuel Colmenar Lamas

University of Sydney

Contents

INTRODUCTION

The objective of this Project is developing a program which is able to create a short video using different digital video processing techniques. The output video is build by the program using the source video provided in the project folder, in the *"media"* folder. In the next section the distribution of the folders will be explained in further detail.

According to the programming language chosen in order to developed this assignment, I have chosen *Python* . The main reason why I have chosen this programming language instead of other one, such as Processing, is due to the fact that I have more experience with Python as well as there are more resources to find information in the case that any problem happen.

The output of the video is located in the folder *"final"*. Moreover, in order to check the correctness of each section in a easier and faster way, there are folders which contains the video of partial solution of the assignment, such as the video with the background replaced or the monkey labeled with the motion capture.

Regarding to the option chosen, I have chose the first option which is **Intelligent Animation**. This option is divided into for different subsections: *Motion Capture*, *Replace Background and Marionette*, *Intelligent Objects* and *Sound Track*. Each of these sections will have three different subsections in this report explaining the implementation chosen for them, the problems encountered while developing them and the final results.

FOLDER DISTRIBUTION

According to the distribution of all the files of this project, the organization selected is supposed to make easier the access to the files of each of the different parts of the assignment. Mainly, it is divided into *media*, where the input video is stored; *src*, where the actual code is located; and *output*, where the result frames and videos of the different subsections are stored. Below we are going to make a more detailed explanation of the folder distribution.

1. **output** : In this folder the result of each of the parts of the assignment are stored. The different sub-folders are the following:
   (a) *background*: Stores the frames of video used to replaced the background of the original video.
   (b) *background_replaced*: Stores the frames of the original video with the background replaced by another video.
   (c) *frames*: Frames of the original video.
   (d) *motion_capture*: Have the frames of the result of the first part of the assignment. In this part the monkey is substituted by red pixels.
   (e) *motion_capture_video*: Result video of the motion capture.
2. **src** : Contains the source code. It is divided in the following sub-folders:
   (a) *constants*:
      i. variables.py: Stores constant variables used in the execution of the program
   (b) *data_struct*:
      i. frame_data: It is the data structure used for represent the frames of the video. It has these fields:
         A. *name*: Name of the frame.
         B. *bmap*: The image divided into blocks with the size of the grid size. It is used fro calculating the SSD.

C. *r_img*: The image itself.

D. *disp*: Displacement of each of the blocks of the image.

E. *stickman*: Stickman used for replacing the monkey.

F. *arrow*: Intelligent object used in the third section.

G. *mini_stickman*: Intelligent object used in the third section.

ii. stickman.py: Class containing the object of the Stickman. It has the coordinates of the different parts of it as well as a method to draw the whole stick-man.

iii. arrow.py: It is the file containing the class for the object Arrow. It is composed of the coordinates and a draw method.

(c) *functions*: Set of files containing auxiliary functions. Mainly, there are two files which are:

i. func.py

ii. auxiliary.py

(d) *sections*: It contains a file for each section if the assignment. These files are:

i. motion_capt.py

ii. replace.py

iii. intelligent_obj.py

iv. sound.py

3. **src**: It contains the input videos for the background and for the original video.

MOTION CAPTURE

The objective of this section is performing a *Motion Capture* to the Monkey in the original video. In order to show this tracking in a graphical way, the Monkey is labeled with red pixels.

Implementation

In order to implement this section of the Assignment, the approach which I have chosen is using the *gray scale* and the *Binary Threshold* in order to identify the Monkey.

The first thing to be done is calculating the *size of the grid* for the image. This is used later in order to be able of calculating the SSD and the displacement of the Monkey. Once that we have calculated the grid size, we create all the *FrameData* objects. Then, we create the folders needed for storing the output of the Motion Capture.

After creating all the FrameData objects, we are going call to the method *monkey_to_red* with each of the frames. In this method the input image is turned into *gray scale* and then the gray scale image is converted to a *binary threshold*. After performing these two operations, an *erosion* is done to the background in order to have the Monkey figure clearer in the image.

The last thing that we have to perform is just changing the black pixels of the image, the Monkey, to read. Then, we save the image in the correct folder.

Finally, after performing the change of the monkey to red pixels, the displacement is calculated using the SSD formula. The displacement of all the blocks of each frame is stored in the field *disp* in the FrameData object of the corresponding frame.

Result

The result frames of this section are stored in the folder *motion*. Moreover, a video with the resulted frames is stored in *motion_capture_video*.

Problem

The main problem found was selecting only the Monkey and not some parts of the Background when setting the Money to red. This was solved performing an adjusting of the parameters of the Binary Threshold Operation as well as performing an erosion to the background.

Moreover, the main problem of this part was calculating the SSD in a fast way. This was not achieved and the calculation of the SSD, although is done correctly, is done in a slow way, Because of this, the output videos from now to the last section is going to be obtained using a reduced version of the input video in order to speed up the generation of the video.

REPLACE BACKGROUND AND MARIONETTE

The aim of this part of the Project is implementing a replacement for both, the background and for the Monkey. Moreover, the background is replaced by another background video chose by me and the Monkey is replaced by a Stickman.

Implementation

According to the approach used in the implementation of the replacement of the background, the first thing to be done is converting the background video to frames. These frames are stored in the folder *background*. Once that this is done, we used the *binary threshold* of the Motion Capture section to replace the old background with the new one. This is saved in the folder *background replaced*

Moreover, the next action to be performed is the replacement of the red pixels of the Monkey with the Stickman. In order to perform the replacement in the correct place, first the center of the Monkey is calculated. However, due to the fact that the Monkey is not an uniform shape we have to apply some fixed offsets in order to calculate the center.

Once that the center of the Monkey is correctly calculated, the coordinates of each of the parts of the Stickman are calculated. Then, a new Stickman object is created, added to the FrameData object and drawn in the image.

Finally, the red pixels of the Monkey are replaced by the rest of the background and the video with the output of this section is saved in the folder *background_replaced_video*

Result

Regarding to the result of the replacement of the Background and the Marionette, the frames are saved into the folder *background_replaced* and the video created with the frames is stored

in *background_replaced_video*.

Problem

The main problem of this section was calculating the movement of the different parts of the Money and apply that movement to the new Stickman. This problem was not fully solved. The movement of the Monkey is tracked and passed to the Stickman but it is not correctly translated into the correct movement of the Stickman.

INTELLIGENT OBJECTS

The purpose of this part is creating at least two randomly moving objects which are going to interact with the moving marionette. The ways that they are going to interact with it are by collision and by mimicking its movement. Moreover, when the collision happens some special interactions occur in the frame.

Implementation

According to our implementation, two Intelligent Objects are going to be added. These two objects are: an *arrow* and a *mini stickman*.

The *Arrow* is going to implement the *collision*. It is going to go to collide with the body of the Stickman. Once that the collision happens, the Arrow disappeared and the size of the head of the mini stickman changes it size.

In order to be able of implement this feature, the new Data Structure has been created, the Arrow. Moreover, for being able of knowing when the collision happens an implementation of the intersection of two lines has been done. It takes as the two lines: the body of the Stickman and the Stick of the Arrow.

Regarding the second Intelligent Object, the Mini Stickman try to mimic the movement of the original stickman. Moreover, when the collision of the Arrow and the Stickman happens, the size of its head changes.

Result

The frames with the result of this part are stored in the folder *intelligent_objects* and the video created with these frames is saved in the folder *intelligent_objects_video*

Problem

According to the problems faced to in this third section, the main problem was getting the movement of the main Stickman and adapted to the Mini Stickman in order to implement the tracking. As I was not able of implemented correctly, only the changing of the size of its head is done.

Moreover, at the beginning I had several problems in order to calculate the collision of the Arrow with the body of the Stickman. Hopefully, this problem was solved.

SOUND TRACK

In the last section of this Project at least two sound tracks should be added to the video. These soundtracks should be related to the interactions of the Intelligent Objects added in the former section.

Implementation

The implementation of this part was done using the library *ffpyplayer.player*. Using this library I can display the previous video created with the frames of the Intelligent Objects and add the soundtracks that I want in the moment that I want.

According to the soundtracks added, there is one which is the background sound that is playing all the time until the Arrow hits the Stickman. Once that this collision happens, the soundtrack stops and an additional soundtrack is played. After it, the background is payed again.

Result

In order to see the result the whole execution of the program has to be run previously. Moreover, in order to be able to make a fast demo of it, we can changed the variable *demo_sound* to True. If we do this, only the sound section is run and the input taken for the video is the video created in a previously execution of the program.

Problem

The biggest problem in this section was finding the correct library to manage video and audio at the same time. Before finding the solution i tried with several libraries, such as pygame,

but due to the fact that most of them were deprecated or they did not have all the featured needed for this section I chose *ffpyplayer.player*.

CONCLUSION

This project covers all the main topics taught in the different lab sessions. Because of this, in order to be able of developing correctly all the different sections, a high skill and understanding of computer graphics is needed.

However, the fact that the project is so big and cover so many techniques, while you are solving the errors that you find during the development you realized that your understanding more and more the main concepts of computer graphics and its techniques.

Because of this, although that in my opinion this project is very big and a bit difficult to implement all the different sections, it is worth it to spend a lot of time trying to finish it in order to learn as much as possible of computer graphics.