

Assignment

470176001

October 2018

1 Introduction

The main purpose of assignment portion 1 is about motion capture, background and marionette replacement, interaction between objects and sound tracking. Involving theories as dividing video frames, motion estimation, animation techniques, ect. Instead of complementing each of them, in this Assignment we seems to make different processing to one video, and try to make these new added applications work harmoniously together. For implementing the Assignment, The most importance thing is learn how to control pixels in image, for example, tracking object motion by compare the RGB differences of pixels in same position between adjacent frames. the analysis of RBG is frequently as well, each pixel contain a color, and each color is produced by the combination of three colors (red, green, blue) RGB, this theory can be used to identify boundary.

2 Outcome description

At the beginning of video, the marionette is marked by red mark with original blue background. Later, the original background will automatically changed to a dynamic video found online, then the arms, legs, heart and head of marionette are labelled by black points while the other parts of it's body are replaced by background pixels. After marionette changes back to it's original color, the Bubble launched (junk food object, randomly moving objects), whenever a Bubble hits marionette, Bubble returns back with a more fast speed, a sound for celebrating displayed, the color of marionette change to yellow, a string shown on left top corner is " Lose 1 kg!". Then last, there is a little bee follow marionette until the end of video.

3 Methods

3.1 setup()

Method for preparation, loading images, sound and videos for variables, setting size of window, play rate of video ect.

Movie m → *Given monkey video*

Movie bg → *Self chosen background video*

PImage refer → *final version of image shown onto window*

PImage remove → *image with extracted object*

PImage pre → *store previous frame*

PImage[] food → *image pool for randomly moving objects*

Bubble b1, b2, b3, b4 → *created bubbles*

int framenummer → *current number of frames*

float threshold → *set for compare RGB difference between two points*

3.2 draw()

Fistly, implementing PImage remove (setting blue background to color(-1)),the method is try to compare the blue and green value of each pixel to thresholds then identify the background.

Secondly, motion estimation,loadpixels of PImage pre (previous frame), refer (current frame), and remove (current frame with extracted object). comparing RGB values between each pair of points at same position on remove (current frame) and pre (previous frame) frame, the points only showing moving object. For algorithm calculating, call disqt(float x1, float y1, float z1, float x2, float y2, float z2) method and return the difference value, compare the value with specific threshold, if the difference is big enough then save the x, y coordinates for red mark drawing.

Thirdly, do marionette by calling function Marionette(PImage remove, PImage refer, PImage bg), this method can be explained later. Generally, using RGB analysis to identify arms, legs, haert and head of marionette.

Fourthly, loading PImage refer to window

Fifthly, launch different junk food (different pictures of randomly moving objects) during different frame range, each Bubble object drawing a junk food object and begin to move, whenever it hits marionette the object return with a more rapid speed, and the color of marionette change to yellow, a string shown on left top corner " Lose 1 kg.". Then a small bee tracking the movement of marionette until the end of video.

3.3 Bubble class

Each Bubble is a junk food object, with it's own attributes, such as X, y coordinates, diameter and moving speed. And it's own methods, Bubble construct method for creating new Bubbles, display() for drawing image into window and accsece() method for controlling the moving of Bubbles. Since a new Bubble appear, it keeps moving if it has not achieves the boundary of window nor hits marionette. However, if Bubble hits marionette, it will return with a more rapid speed, a sound displayed with a string "Lose 1 kg!" shown at the left top corner of window.

Bubble(int temp_x, int temp_y, int temp_d, int sp) → *construct method for Bubble*
void display(PImage food) → *to draw images in window*
int accsece(PImage remove, PImage food) → *controlling the movement of Bubbles*

3.4 void movieEvent (Movie m, Movie bg)

Function used to read videos and divide frames, current frame is loaded and previous frame is stored in variable, the stored previous frame is used to identify object motion by comparing with current frame.

Movie m → *Given monkey video*
Movie bg → *Self choose background video*

3.5 float disqt(float x1, float y1, float z1, float x2, float y2, float z2)

Method for find the RGB differences between position A and position B. The algorithm used is $\text{float } d = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2$, seems like distance algorithm.

float x1, y1, z1 → *RGB value of position A*
float x2, y2, z2 → *RGB value of position B*

3.6 void redMark(List<Integer> coordListX, List<Integer> coordListY, int framenum)

The red mark for identifying moving object is drawn in this method. During period of frame 0 to 300, following the input coordListX and coordListY list, change RGB value to red for each pixel. The input lists are generated from setup() function, which stands for the position of moving object.

List<Integer> coordListX → *a list of x value*

List < Integer > coordListY → a list of y value
intframenumber → the current number of frames

3.7 void Marionette(PImage remove, PImage refer, PImage bg)

Method for replacing marionette. As we can see the marionette has the most red arms, legs, heart and yellow face, all them contain stronger red than any other part. Therefore, extracting marionette from PImage remove, then using RGB analysis to identify arms, legs, heart and face of marionette, label them by black points, and replace other parts of body by background. Lastl, load new pixels to refer image

PImage remove → image with extracted object
PImage refer → final version of image shown onto window
PImage bg → background image

3.8 void hitReflection(PImage remove, PImage refer)

Function for implementing reflection of marionette when hits food, color of marionette changed to yellow then change back. Extracting object from remove image, loading final version to refer image.

PImage remove → image with extracted object
PImage refer → final version of image shown onto window

4 Conclusion

During the Image and Video processing, every operation is based on pixels, dividing video into frames then dividing images into pixels. The RGB analysis of pixels can be used to identify obvious object, such as the green object in red background. The motion estimation, by comparing the RGB differences of pair of pixels. The separate object class (such as Bubble class), which contains its own attributes and methods can work as a object, which is much easier and clean. Animation techniques might be one of the applications of object oriented method, by keeping changing coordinates of objects when it has not achieved boundaries.