186

# Activity 11: Basic Video Processing

Mary Chris Go / 2014 11122

# Goal

Mary Chris Go / 2014 11122

To be able to measure kinematic constants using image processing in frames

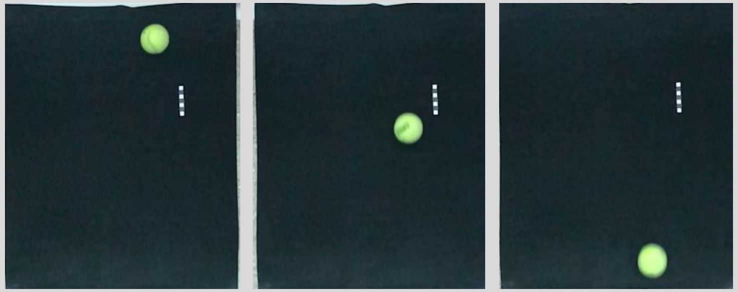
## Video Processing

There are existing freeware to achieve the goal of this activity. But here, we try to manipulate the video such as we use image processing to the extracted frames from the video. The camera had 240 fps feature. This helped a better quality of frames. (Thanks to ip7+ of Sam)

## Frame Rate

Fps = frames per second. The inverse of this gives us the time interval between each frames.

# Extracting frames

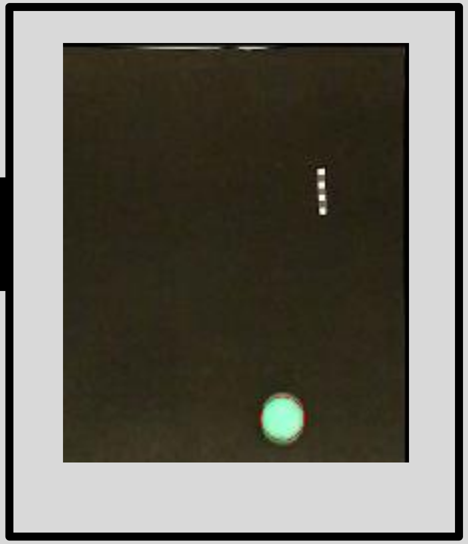


With the use of python, we were able to extract frames from the videos. Around 500 frames were extracted but we only use the time the ball was in the black cartolina. What you see are some chosen frames. A snippet of the code is shown below.

```python
#Extracting Frame
import cv2
vidcap = cv2.VideoCapture('C://Users//MaryChrisGo//Documents//1st Sem AY 2019-2020//App Physics 186//Act_11//greenball.MOV')
success,image = vidcap.read()
count = 0
while success:
  cv2.imwrite("frame%d.jpg" % count, image)     # save frame as JPEG file
  success,image = vidcap.read()
  print('Read a new frame: ', success)
  count += 1
```

# Setting up and testing blob detector

Mary Chris Go / 2014 11122



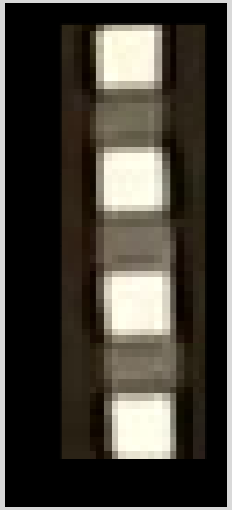
```
# Setup SimpleBlobDetector parameters.
params = cv2.SimpleBlobDetector_Params()

# # Filter by Area.
params.filterByArea = True
params.minArea = 200

# Filter by Circularity
params.filterByCircularity = True
params.minCircularity = 0.5

params.filterByColor =  False
params.blobColor = 200

# Filter by Inertia
params.filterByInertia = False
params.minInertiaRatio = 0.1
```

```
params.filterByConvexity = False

# Set up the detector with default parameters.
detector = cv2.SimpleBlobDetector_create(params)


image = cv2.imread('pics/frame213.jpg')
image = image[385:960,120:580]
keypoints = detector.detect(image)
im_with_keypoints = cv2.drawKeypoints(image, keypoints

# plt.figure(figsize=(20,10))
plt.imshow(im_with_keypoints)
plt.show()
```

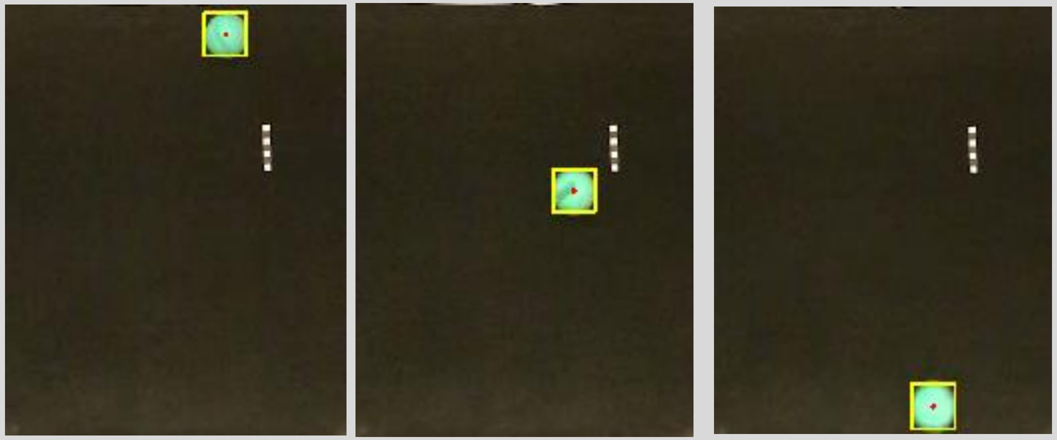
This part, we try to detect the ball using one image first. We set up the parameters of the blob detector that we will use for the many frames extracted. Using the past code for blob analysis, we were able to detect the ball as seen in the sample picture above.

# Setting up and testing blob detector

Mary Chris Go / 2014 11122



Scale used.



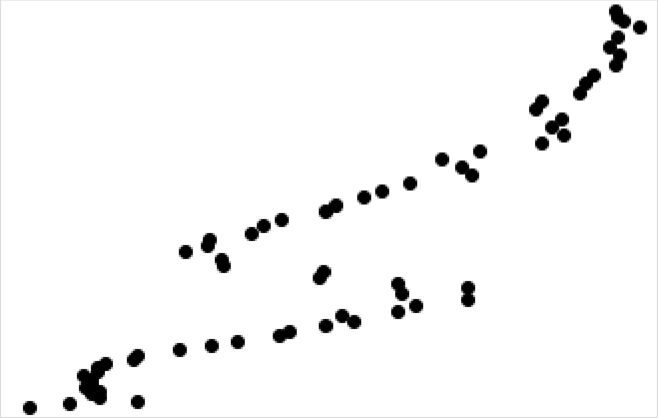
Using **for** loop, we can detect the ball on the many frames extracted on the first part using blob analysis. The snippet of the code is shown below. This is under a for loop.

```
#Draw center and square
pt = keypoints[0].pt
d = keypoints[0].size
im_with_sq = cv2.rectangle(image, (int(pt[0] - d/2), int(pt[1] - d/2)), (int(pt[0] + d/2), int(pt[1] + d/2)), (255,255,0
im_with_sq = cv2.rectangle(image, (int(pt[0]),int(pt[1])), (int(pt[0]+1),int(pt[1]+1)), (255,0,0), 5)

#Record position
centroids.append(keypoints[0])
```

# Finding g

```python
print("g = " + str(g_fit*((7/60)*0.01)))
```

```
g = 9.471877341253874
```



```python
#Getting x and y values
for keypoint in centroids:
    x,y = keypoint.pt
    traj_x.append(x)
    traj_y.append(y)
```

Now we get the trajectory of the ball by getting the x and y values per frame. The picture on the left is the plotted x and y.

We convert the frames to time using an equation. Right after that, we fit the data through a defined function of acceleration.

The theoretical value of the acceleration due to gravity is 9.8 m/s^2. From this experiment, we calculated 9.5 m/s^2 yielding a 3% error.

Mary Chris Go / 2014 11122

# Summary

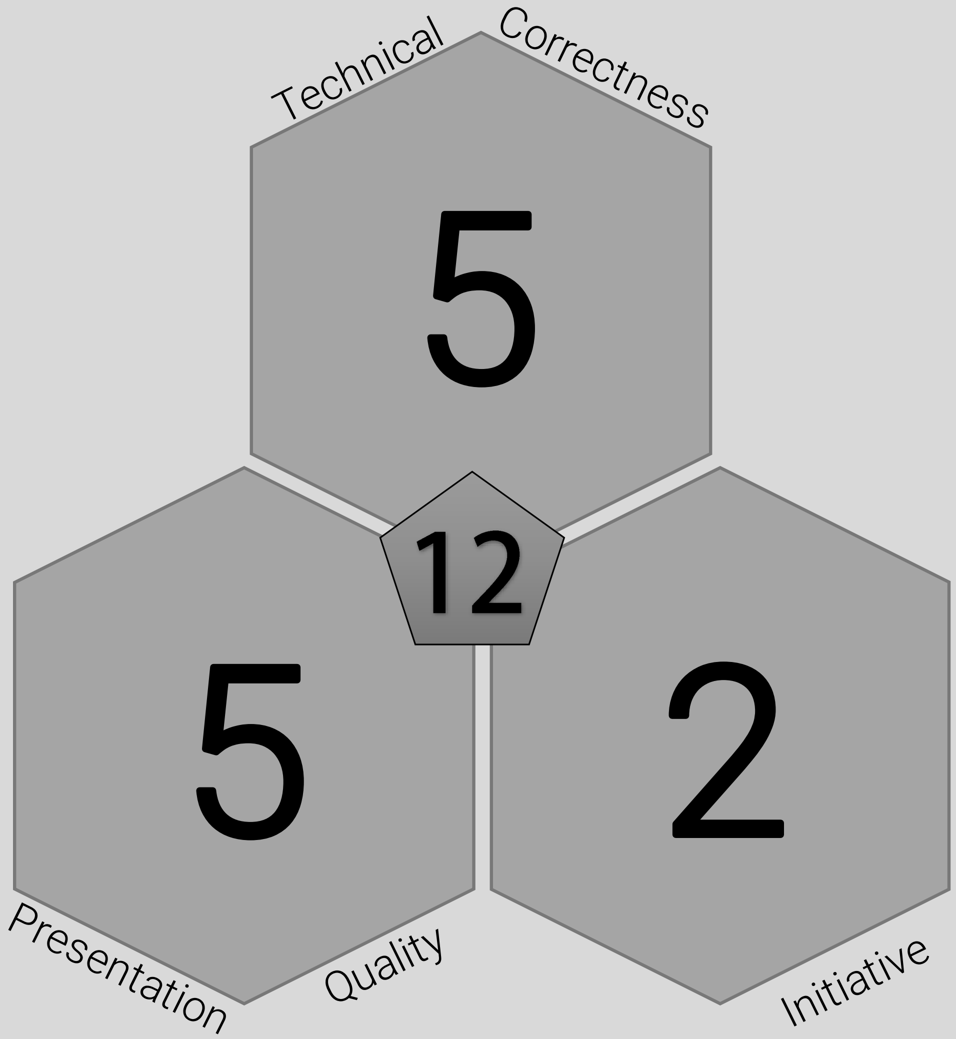
We were able to measure the acceleration due to gravity with a very small percent error. This shows the effectivity of image processing in such applications. The percent error can still be improved by having higher fps. The limitation of the technology used affected the accuracy of the method.

Side note: Python makes life easier. Just by simple code, you can batch rename files and extract frames from the videos. No need for third party freeware.

The activity was so fun especially doing it with friends. Although we know there were more efficient ways to know kinematic measurements, this kind of method taught us the power of image processing and its many applications. Now I get to appreciate the past activities we had.

Groupmates: Delima, Ancheta, Lahoz, Rigonan

# Self-Evaluation




Mary Chris Go / 2014 11122

# References


- Soriano, M., “Blob Analysis”. 2019