**Objective:**

We would like to devise a solution to track the center of action in sports footage. That is, the region of interest that the viewer would want to follow when watching some particular sport. The solution should be able to be generalized across a variety of sports.

**Heatmap:**

Our initial idea was to utilize a heatmap to detect "hot" regions in a sequence of frames. This is achieved via background subtraction and frame differencing.

There are a variety of existing algorithms for background subtraction. In general, most background subtraction algorithms attempt to create a binary segmentation of a frame to identify the regions of non-stationary objects.

Since we expect to analyze footage in real-time, we use fast algorithms such as CNT and MOG from OpenCV. The former performs better without Nvidia CUDA and on cheaper hardware. The latter utilizes Gaussian distributions to represent pixels to identify whether they are in the foreground/background. This is also fast, especially on GPU boosted architectures.

See:

<https://github.com/sagi-z/BackgroundSubtractorCNT>

<https://link.springer.com/chapter/10.1007/978-1-4615-0913-4_11>

In our initial implementation, we chose to use MOG to create an initial filter for each frame. Then, we combine the filters for the past N frames and apply a color map to produce the output heatmap. N is a tunable parameter that may be adjusted depending on how long we want a hot region to stay relevant. Also, note that the combined frames from the background subtraction produces a grayscale image making where we can identify hot regions based on lighter regions using the ColorMap operations in OpenCV.

Overall, this method is very effective on stationary cameras as the difference between frames are the moving objects. However when using a non-stationary camera, every time the camera moves the background subtraction identifies most of the pixels on the screen as the foreground. As a result, we are unable to identify actual moving objects from the noise identified as moving from the camera movement. We may define camera movement in 3 forms: Zooming, Panning, or Tilting. Any camera movement may be one or any combination of these movements.

There have been various attempts to produce an effective background subtraction on video from moving cameras. We would like to test some of these results in the heatmap context.

See:

<https://www.cs.cmu.edu/~yaser/SheikhJavedKanade_ICCV_2009.pdf>

**Object Detection/Tracking:**

A problem with using background subtraction is that we are also identifying moving objects that are not of interest. We define object of interest as objects relevant to the center of action (i.e. players in the game). Thus, a context-sensitive analysis should be done to identify such objects. As a result, we look towards combining the heatmap approach with an object detection/tracking solution to find objects of interest prior to applying the heatmap. From these objects of interest, we can create a general region of interest to apply the heatmap.

First, we distinguish object detection from object tracking:

* object detection identifies objects in a frames without considering past frames or potential future frames
* object tracking follows an object's trajectory over time

In essence, object tracking is not responsible for deciding regions of interest, but rather tracking them after they have been identified.

For our purpose, we would like to achieve automatic object detection to differentiate objects of interests from objects not of interest. This may be done implicitly by identifying unique (identifying) features of our objects of interests and only detecting objects that have these characteristics. For instance, we will likely be interested in the players in a sports game, but not the audience. Thus, common human detection algorithms may not be sufficient. Instead, we may rely on assumptions such as the size and shape of an object of interest.

Furthermore, object tracking may not serve our purpose well because our objective with this supplementary algorithm is to obtain a rough estimate of a region of interest; thus, we do not need to keep track of individual items. Object tracking may obtain a more accurate interest region since it utilizes temporal data, but the extra computation is significant and may not be necessary.

Again, there are various object detection schemes. Thus far we have experimented with using Gaussian background subtraction with OpenCV's MOG2 algorithm to binarize each frame and generate relevant contours. Then, we are able to generate bounding boxes for contour regions with an area larger than some minArea and smaller than some maxArea threshold. minArea and maxArea are tunable parameters that may be tuned to reflect a range for the area of any object of interest. These parameters may eventually be learned for various cases, but this is not considered at this point. After generating the x coordinate, y coordinate, width, and height of the contour's bounding box, we also consider the ratio of the width to height of these bounding boxes. For example, we may typically prefer objects that are tall (like a human). This ratio is also a tunable parameter and we may consider multiple ratios.

After determining the bounding boxes, we maintain the top X largest bounding boxes in a priority queue that will be used to detect our general region of interest. X is a tunable parameter, where the larger X will generally produce a larger . To create this region of interest, we merge the bounding boxes by taking the most extreme coordinates to create a single large bounding box to represent our final region of interest. This approach has proven to be unreliable due to the inconsistency of the object detection procedure. Thus far, any attempt to combine our object detection and heatmap has not proven effective. However, we suspect this is largely due to the object detection procedure.

Since we are only using object detection, we perform this feature extraction on each frame independent of other frames. Thus, this solution works on moving cameras (with varying degrees of success). However, if we are able to find an efficient object detection algorithm that accounts for the effects of a moving camera. There have been attempts to implement object detection with moving cameras that should be investigated. A common approach is to use points in the background of a frame to serve as a trajectory basis for the camera. This is then used to compute the movement of the camera so that

See:

<https://arxiv.org/abs/2001.05238>

<https://www.sciencedirect.com/science/article/abs/pii/S0167865517300260>