Object tracking is 360 camera footage

lkbgift

Jun 22

Hey!

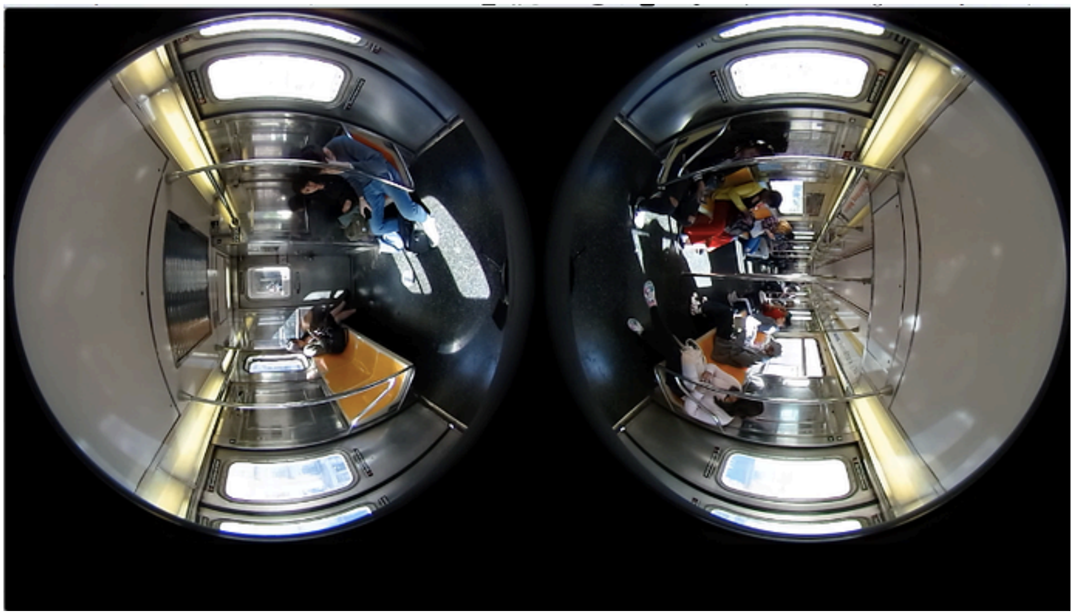
I recently purchased a https://theta360.com/en/ to take 360 photos. The camera has a streaming video option, which streams content in duel photosphere format.

I want to know how I go about turning this duel photosphere into a equirectangular panorama photo, then to track an object across the image.

Im sure once the image is converted, the tracking is like any other photo, but since there is some distortion, I was wondering if there is anything I need to watch out for.

Image examples below.

1. exported video format.
2. desired equirectangular panorama, with distortion



Adrian Chief PyImageSearcher **Jun 23**

Very cool, I haven't seen this camera before, it looks quite neat!

Although, as far as my understand goes, I don't think OpenCV has any built-in functionality for **equirectangular projections** -- this is also not something that I have any previous experience with either.

The biggest issue you're going to run into is that the "undistorted image" will still have distortion due to the fish eye effect. Depending on the objects you want to track, this may cause *a lot* of problems. For example, the HOG-based tracking method is a structural tracker -- if the gradients and objects become started, you won't be able to detect them or track them as easily. However, color-based tracking methods should still work without a problem.

lkbgift **Jun 23**

What do you think of the SIFT or SURF method for object tracking? Will that make a difference in regards to identifying object features?

I did some basic tests using a SIFT implementation, and the object was being tracked well through a video.

Adrian Chief PyImageSearcher **Jun 25**

Keypoint detectors and local invariant descriptors are good for tracking objects *provided* that there are enough unique, distinguishing keypoints that can be detected on both your *reference image* and your input *image/frame*.

For example, tracking the cover of video game with lots and lots of graphics, edges, corners, etc. is *much easier* than say, tracking a paper towel -- which other than the corners of the towel itself, it has no distinguishing characteristics.

As long as you keep that in mind when using keypoint methods for tracking, you'll likely save yourself a lot of headaches. If an object isn't being tracked properly, your first step should be to see how many keypoints were detected in both objects -- and then how many were matched. If not many were detected, then SIFT/SURF isn't a good choice for the problem. If keypoints are detected on both objects, then it's a matching problem. You might need to try a better a different local invariant descriptor or adjust your matching algorithm.