Template-Matching Based Target Tracking

For this project we worked on implementing the basic ideas of a template-matching based target tracking to experiment with visual target tracking. We used a 500 frame video to test various different image matching metrics, a sample of this video is shown below:

This video will be referred to as girl.mov for simplicity.

The Algorithm

The algorithm we used to solve this problem was a basic exhaustive search around a search window, the algorithm can be created as follows:

1. Create the initial bounding box for the object you want to track, known as a template. An example of this cropped image is shown below (the head of the woman in girl.mov):

2. Create a search window around the image template, this will be the area in which we will search for our best matching bounding box. The search window is the green rectangle in the image below labeled ‘search window’:

3. For every potential bounding box in the search window (must be the same size as the template), use a metric (SSD, CC, or NCC) and find the best matching bounding box using the metric. In the above image, that is the blue box labeled ‘object’.
4. Repeat from step (2) for the next image, setting our template to the best matching image we just found in (3), until we run out of images in the video
5. Once we are out of images, turn these series of images into a video

Implementation Results

The video results of this exercise are not really possible to put in this document, but they can be found in the submission as part of the files. I will show some screenshots of the head from girl.mov being tracked for various types of movement.

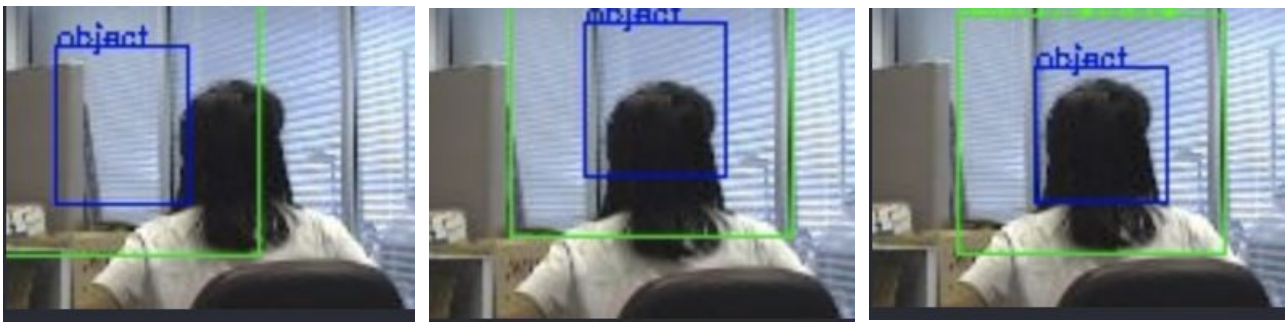
- X-Y Translation

Here are the object following results from a X-Y translation (the girl has moved to the left), the metrics used are SSD, CC, and NCC from left to right respectively. As we can see both SSD and NCC seem to handle this task very well, keeping a good track of the head. CC does alright, still following the shape of the head, but there has been some offset in the y direction that does not perfectly align with the face.

- Z-direction translation

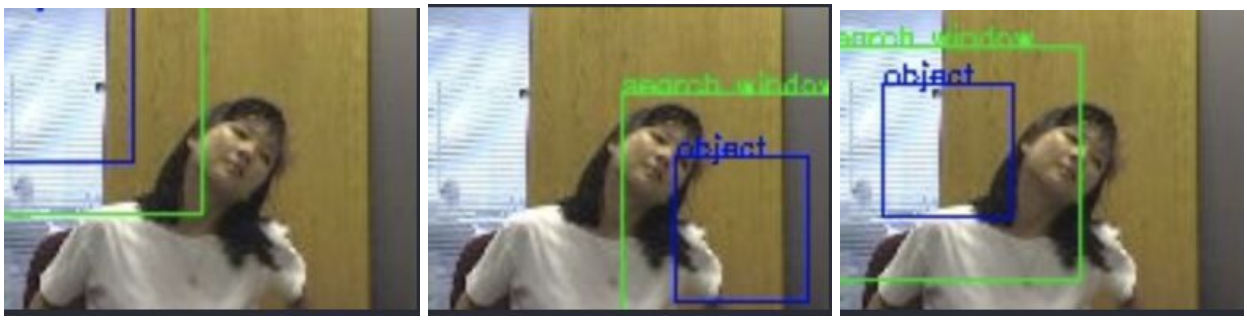
Here are the object following results from a Z-direction translation (the girl has moved backwards away from the camera), the metrics used are SSD, CC, and NCC from left to right respectively. For this type of translation, NCC appears to do the best, still tracking the head with fairly good accuracy. SSD is also good, but the object's boundary box has shifted slightly and the head is no longer centered in the box. CC is also alright, as it seems to be following the x direction fairly well, but there is still some offset in the y direction that does not perfectly align with the face.

- Y axis rotation



Here are the object following results from a rotation around the Y axis (the girl has spun in her chair), the metrics used are SSD, CC, and NCC from left to right respectively. For this rotation, we can see that NCC and CC have much better results than SSD. NCC has the best results, tracking the head with good accuracy, while CC is close but still has some slight offset in the y direction. SSD is not completely off and doesn't seem to be on the head at all.

- Z axis rotation



Here are the object following results from a rotation around the Z axis (the girl has tilted her head), the metrics used are SSD, CC, and NCC from left to right respectively. For this rotation, we can see that none of our metrics actually produces accurate results.

Analysis

SSD and NCC perform well on X-Y translation and Z translation, which is expected for simple exhaustive search. The only problem is CC, which is still slightly offset even on X-Y translation. The reason CC has problems performing on these transformations is because of its sensitivity to noise and lighting. This causes CC to change extremely rapidly from accurate to inaccurate. NCC helps to remove this problem through its normalization process.

Y axis rotation appears doable for NCC and CC but not for SSD, while Z axis rotation is not doable for any metric. This makes sense as we have no way to account for rotations, as our metrics all assume pure translation transformations. Y axis rotation appears okay for CC and NCC as they are slightly more robust than SSD.

NCC appears to be the most robust metric of the three. This is because, by normalizing CC we remove sensitivity to lighting, and by subtracting the means to the image and template each time we help to remove sensitivity to noise.