Report of MP7

1. Template-matching based Target Tracking

It is used to locate a template image within a larger image or sequence of images. In technological, slide the template across the search area and computing a similarity metric(SSD, CC,NCC) at each position. In our project, we need track a girl's head in 500 images.

2. Algorithm Implementation

The project consist of 5 part

1. Select_bounding_box: in the first frame (image), we manual select a box around the target, it just the template.
2. Track_object: for each subsequent frame (image), use exhaustive search compare every region with the template in the previous frame.in Here, call three different metrics: Squared Differences (SSD), Cross-Correlation (CC), and Normalized Cross-Correlation (NCC). The region with highest(lowest) score are drawed with an new bounding box, take it as the next template. Continue annotate each frame until the whole 500 frames are went through.
3. Create_vedio: compile the annotated frame into a video, used VedioWriter in cv2 package.

3. Result analysis

After we use SSD, CC, NCC 3 different metrics, we get 3 .mp4 videos. Each one has 25s.

Target movement modes can be classified into three types: horizontal translation, vertical translation, and rotation(tilt). Analysis of the performance of the three modes can be carried out from the overall accuracy of the entire 500 frames(overall accuracy), the maximum offset, and the above three movement modes.

3.1 Overall Accuracy

In the entire 25s video and 500frame picture, NCC performed the best. Regardless of the motion mode, the bounding box fit the target head. I think the normalization process allowed NCC to reduce the influence of noise and lighting in the picture.

(NCC before rotation) (NCC rotating) (NCC after roatation)

The second is SSD. The accuracy of SSD is very good before the target rotates. However, after the rotation, the bounding box is obviously offset in the vertical direction, and the horizontal direction is still synchronized with the head.

(SSD before rotation) (SSD rotating) (SSD after roatation)

The worst performer is CC. CC captures target changes very quickly, but too fast even causes the two bounding boxes between previous frame and next frame far away. I think CC is the most sensitive to noise and lighting.

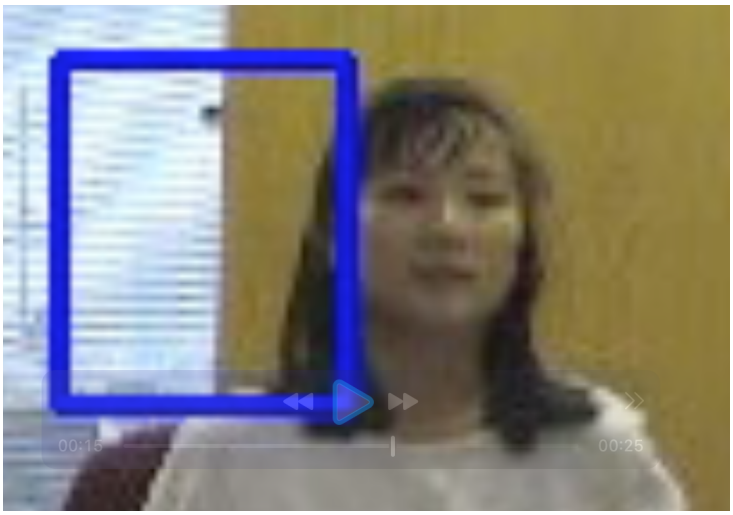

(in 14 frame) (in 15 frame)

3.2 Maximum offset

In SSD, the maximum offset happened in the frame the target just completed rotation,:

In CC, the box multiple completely separate from target head, the maximum offset happened due to it sensitive to lighting

In NCC, the maximum happened after one rotation, the box generates obvious offset in horizontal direction, but still but of box connect to the target head.



From 3 metrics, The offset is due to the noise and lighting in the frame. From the maximum offset, it can be seen that NCC performs the best, but CC is most affected by noise and lighting.

3.3 Rotation and tilt

In the 5s of videos, the target starts to rotate, all three metrics have a certain offset. The SSD moves in the horizontal direction and still chases the target in the vertical direction. CC has a certain offset in both the horizontal and vertical directions. The overall sending box moves like the upper left side of the object. NCC basically maintained that the target is still the center of the box

(SSD rotation) (CC rotation) (NCC rotation)

In the 16s of the videos, the target tilts its head to the left, but the SSD, CC, and NCC are not tracked. The offsets of the three in the horizontal and vertical directions are obvious. It can be seen that the SSD is directly affected by the light in the frame. cc is due to noise and NCC. Because of the influence of the box of the previous frame, although part of it is captured, it is not in the center of the box.

(SSD tilt) (CC tilt) (NCC tilt)

4. Option

The aim it to overcome occlusion, we need modifier the track_object function.

In function we need implement some policy:

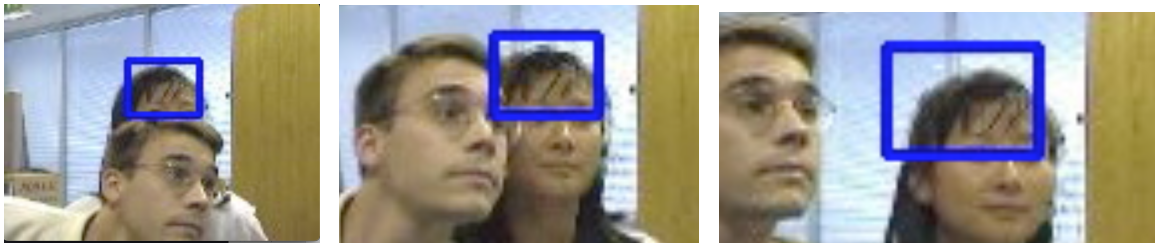
Tracking Failure Detection: Detect when the tracking score falls below a threshold *match_quality*, which indicates a potential tracking failure, possibly due to occlusion.

Template Update Strategy: Update the template only when program confident that the tracking is correct, to prevent updating the template with occluded images.

Re-detection: In case of tracking failure, use a detection algorithm to find the target again in the subsequent frames. This could involve a manual re-selection by the user or an automated process that searches for the target in the entire frame.

In this updated function, track_object now includes logic to handle tracking failure. When the match quality falls below or rises above a the threshold, it indicates that tracking may have been lost. At this point, prompt the user to reselect the target using the select_bounding_box function.

The three screenshot show the overcome occlusion performance under NCC metrics.



(boy coming) (boy moving) (boy leaving)