Assignment 2: Template Tracking

We present here the details for each of the methods we implemented for Template Tracking. We also provide the details on results obtained.

1. Appearance-based Techniques

Methods Used: We have implemented two variants of multi-scale Block-matching techniques. First is the SSD variant which uses the sum of squared distance between the templates and image as a metric to track the object. It determines the location of the object by choosing the window with the minimum SSD. Second is the NCC variant which uses normalised cross-correlation between image and templates to track the object. Opposite to the case of SSD tracker, it determines the location of the object by choosing the window with the maximum NCC. We have used the OpenCV function 'matchTemplate' to perform block matching.

Problems associated with these methods: Multiscaling of templates makes the tracker perform well even when the size of the object changes. But this at times, introduces spurious matches, especially when down-scaling the template. Appearance-based methods fail when multiple objects similar to the one we are tracking are present in the frame. The model confuses these similar objects to be the object to be tracked. Block matching methods also fail when the template undergoes deformation or affine/projective transformations.

Failure Examples: Appearance-based trackers fail when the template gets significantly distorted. One such case is shown below, here Bolt's figure looks significantly different from the initial template.

These trackers also fail when more objects similar to the one being tracked are introduced into the frame. They fail to identify the individuality of objects in these cases. Examples are shown below, instead of choosing the original object, the tracker mistakes another similar looking object for the match and ends up failing.

Multiscaling of templates introduces the problem of spurious matches. Such failure examples are shown below, here the tracker finds a better match for a downscaled version of the template than the original one, making it fail.

Mean IOU scores:

BlurCar2 with NCC: 0.8979

BlurCar2 with SSD: 0.4205

Liquor with NCC: 0.6727

Liquor with SSD: 0.6057

Bolt with NCC: 0.0305

Bolt with SSD: 0.0060

2. Lucas Kanade Tracking

Methods Used: We have implemented 3 different models based on Affine, Projective and Translation transformations to perform template tracking using Lucas Kanade Method. We have followed the same algorithms as discussed in the lectures.

Comparison of Result: This comparison does not provide us with a winner as both the algorithms/techniques outperform each other in some or the other cases. Block matching technique out-performs LK tracking whenever there is a large change in the template position (i.e. it moves swiftly in a short span of time). LK tracking fails to keep track of the template whenever a large change in template position occurs. Following figures show one such example:

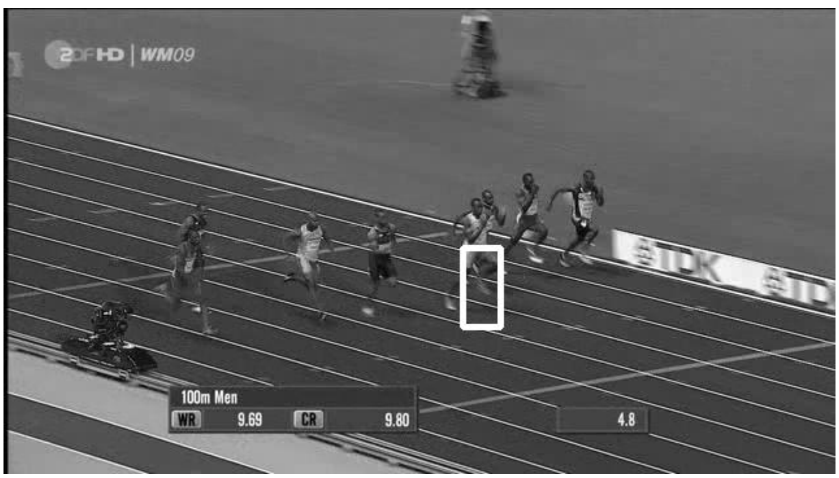
The output of LK Tracking after a large change in car position:

The output of NCC Block matching technique for the same frame:

LK Tracking out-performs block matching techniques whenever there occurs template deformation or when it undergoes affine/projective transformations. In these cases, block matching techniques fail to recognize the template whereas LK Tracking works under the assumption that the template does not move by large amounts swiftly and hence it searches for the best possible template match in the neighbourhood of its previous position, leading to better results. One such example is shown below:

The output of NCC block matching with significant changes in template:

The output of LK Tracking on the same frame:



LK Tracking could not completely track the template, but still performed significantly well compared to Block Matching techniques.

Mean IOU scores:

Liquor with Affine model: 0.3183

Liquor with Projective model: 0.3391

Liquor with Translation model: 0.3245

BlurCar2 with Affine model: 0.3111

BlurCar2 with Projective model: 0.3013

BlurCar2 with Translation model: 0.3252

Bolt with Affine model: 0.02957

Bolt with Projective model: 0.03668

Bolt with Translation model: 0.03218

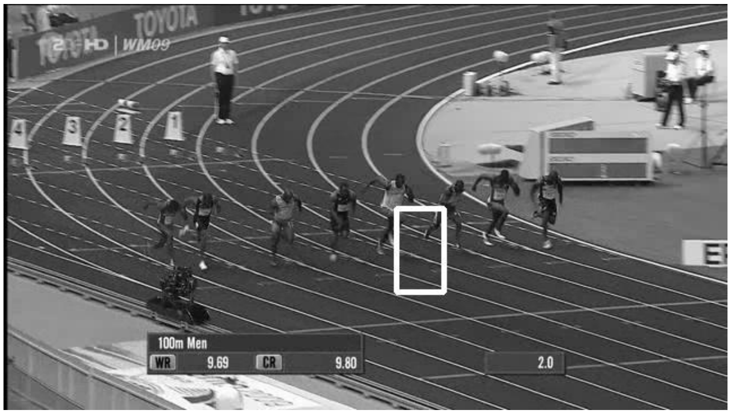
3. Pyramid based Lucas Kanade Tracking

Methods Used: In pyramid based Lucas Kanade tracking, we applied the Lucas Kanade algorithm of question 2 at the top of the image pyramid and integrated the resultant template to the next inferior level of the image pyramid for again applying the Lucas Kanade algorithm of question 2 and repeated this until the original image size is reached. We have implemented this using two different approaches. In the first approach, we have scaled down the image using pyrDown, then resized it to the original size of frame and then applied the Lucas Kanade algorithm of question 2. The template size of the downscaled image is same as the original image and therefore there is no need to change the resultant template size for next iteration of our algorithm. In the second approach, we didn't resize the image initially after scaling down and therefore, for the next iteration of the algorithm, we have resized the resultant template.

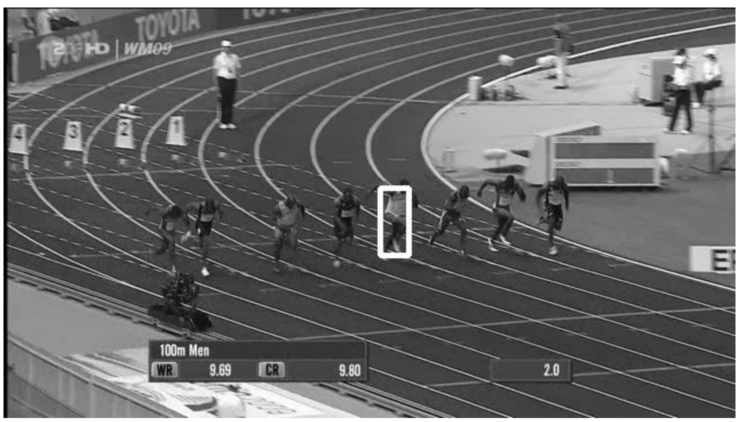
Parameter Estimation: The optimal number of pyramid levels for the BlurCar2 and Liquor is 5 levels and for Bolt is 2 levels. The optimal threshold(ε) for comparing ΔP is 0.01. The optimal kernel size to calculate gradient is a 3×1 or 1×3 central kernel.

Comparison of Result: Pyramid based Lucas Kanade Tracking performs better than simple Lucas Kanade Tracking, sometimes with a large margin while at other times with a smaller margin. We present here, some examples showing the same:

The output of Lucas Kanade Tracking without Pyramid (Part 2) :



The output of Pyramid Based Lucas Kanade Tracking (Part 3) for the same frame:



The following example shows that even with a not-so-accurate initial estimate, Pyramid based LK Tracking manages to have a better matching than simple LK tracking in similar situation. The output of Lucas Kanade Tracking (Part 2) for Liquor frame:

The output of Pyramid Based Lucas Kanade Tracking (Part 3) for the same frame:

Mean IOU scores:

Liquor with Affine model: 0.4340

Liquor with Projective model: 0.3990

Liquor with Translation model: 0.4509

BlurCar2 with Affine model: 0.3242

BlurCar2 with Projective model: 0.3019

BlurCar2 with Translation model: 0.3308

Bolt with Affine model: 0.1827

Bolt with Projective model: 0.100355

Bolt with Translation model: 0.1072

4. Live Demo

Method Used: We use our best performing method, multi-scale block matching with NCC to track the provided template live using a webcam. Please refer to file readme.md for instructions on how to perform live tracking.

Submitted by:

Rakshita Choudhary, 2019CS10389

Deepak, 2019MT10685