Computer Vision 1: Harris Corner Detector and Optical Flow

Gabriele Bani
11640758
gabriele.bani@student.uva.nl

Andrii Skliar
11636785
andrii.skliar@student.uva.nl

Introduction

In this homework we explore various algorithms for corner detection and optical flow estimation.

1 Harris Corner Detector

You can observe obtained results in fig. 1 and fig. 2. We have tried different values of sigma and threshold and, as we found out, they influence the final result quite heavily. You can see results with different parameters in fig. 4.

Lowering the threshold makes more points to be considered as corners and in many cases we would like to avoid it, because this makes results noisy. We have found out, that with larger σ algorithm tends to detect more points and we need larger threshold to filter them out.

After searching for perfect threshold, we have concluded, that for $pingpong$ images threshold should be fairly low - 10, while for $person\_toy$ images it should be larger - 400 worked out pretty well for us. In both cases, we were using value of $\sigma = 1$, which works out pretty well for us.

Also, as you can see in fig. 3, Harris algorithm is indeed rotation-invariant (it will also be discussed in the next question).

1. Is the algorithm rotation-invariant?
   Yes, it is rotation-invariant, because we are not looking at the whole image, but detecting corners using eigenvalue-decomposition of the patch, which means, that when image is rotated, patches will change their orientation, and this will change the shape of the neighbourhood, but eigenvalues will stay the same and this makes algorithm rotation invariant. If we would look at the whole image at one to detect corners, that would make it sensitive to rotation, but as long as we don't, it is invariant.
2. How do Shi and Tomasi define cornerness?
   $H = min(\lambda_1, \lambda_2)$, which according to their experiments, is supposed to be a better criteria for cornerness.
3. Do we need to calculate the eigen decomposition of the image or the patches?
   We need to calculate the eigen decomposition of the patches, because to do so, we need to calculate convolution between Gaussian filter and window centered at the current pixel. Also, we could do it with the whole image as a window, but this way we would only be able to calculate cornerness for one pixel right in the center of the image and that wouldn't make sense, because we need to find pixels, for which cornerness is largest in a certain window.
4. In the following scenarios, what could be the relative cornerness values assigned by Shi and Tommasi?
   - (a) Both eigenvalues are near 0.
     Cornerness value will be 0. This pixel is not corner.

- (b) One eigenvalue is big and the other is near zero.
  Cornerness value will be near 0. This pixel is not corner.
- (c) Both eigenvalues are big.
  Cornerness value will be large. This pixel has a high chance of being corner.

2 Optical Flow with Lucas-Kanade Algorithm

You can observe obtained results in fig. 5.

- At what scale those algorithms operate; i.e local or global?
  Lukas-Kanade operates on a local scale, because it is based on the asumption regarding that pixel being examined has approximately constant optical flow in a neighbourhood and that allows for optical flow equation to hold in that neighbourhood.
  Horn-Schunck operates on a global scale, because it uses the assumption about that flow over the whole image is smooth and minimizes flow as a global energy functional.
- How do they behave on at regions?
  Lukas-Kanade will perform worse due to the fact, that it is operating in a local neighborhood, which is flat and therefore has small gradient. However, Horn-Schunck should perform much better, because it operates on a global scale and can therefore overcome the problem of a region being flat by taking information about changes in the whole image into account.

3 Feature Tracking

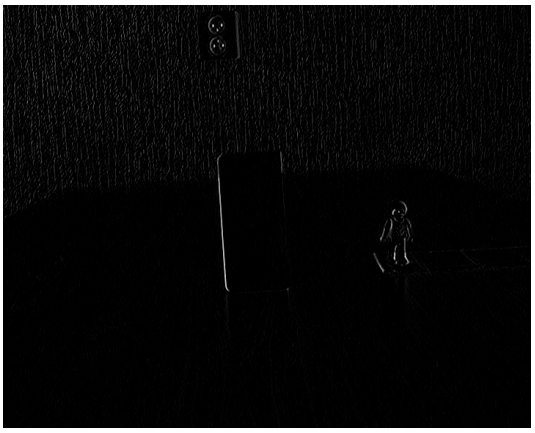
1. Why do we need feature tracking while we can detect features for each and every frame?
   In most of the cases, feature detection is much more expensive, than tracking, which can be done in real-time, and that might be the first reason. The other reason is that, for example, for object detection, tracking is much more robust to changes in viewpoint and therefore might perform better. Also, tracking can be more robust to overlapping objects and can predict the movement of the points in the image.

Conclusion

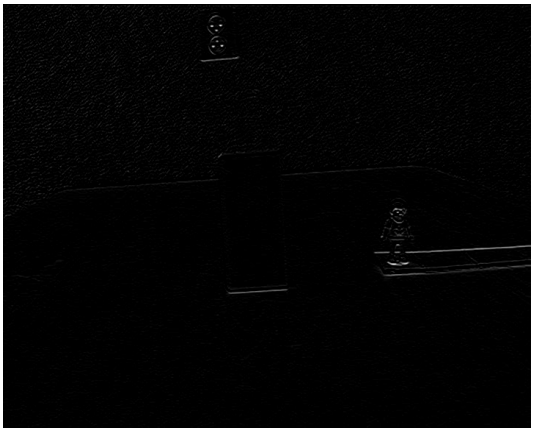
In this homework we have explored various algorithms for corner detection and optical flow estimation.

Figure 1: Results for Harris Corner detection

(a) Image derivatives along x direction I_x



(b) Image derivatives along y direction I_y



(c) Detected corners (marked by red points)

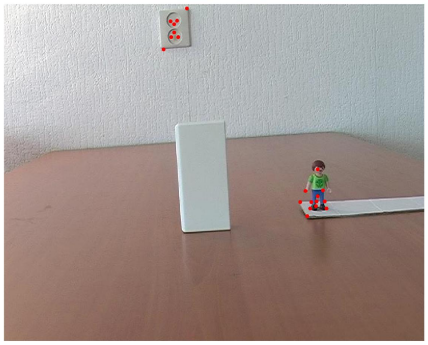
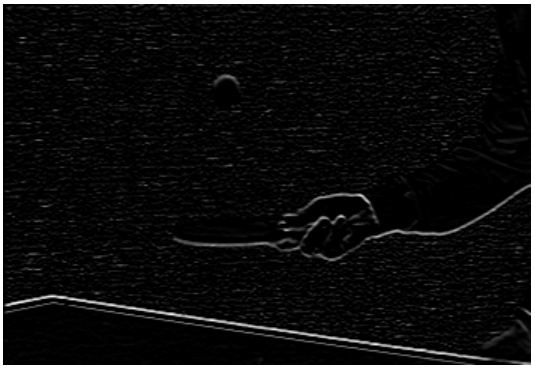


Figure 2: Results for Harris Corner detection

(a) Image derivatives along x direction I_x

(b) Image derivatives along y direction I_y



(c) Detected corners (marked by red points)

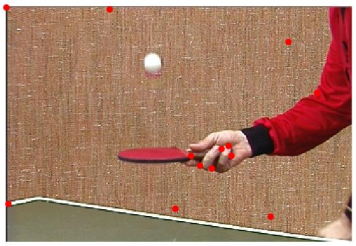
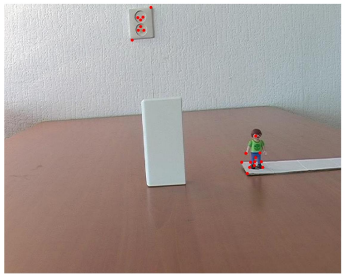
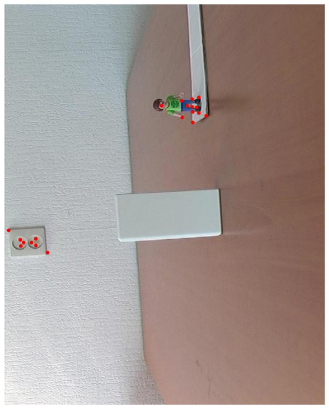


Figure 3: Results for Harris Corner detection rotation

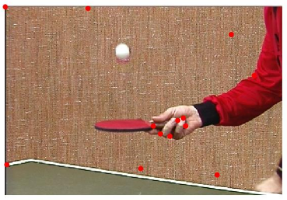
(a) No rotation



(b) 90 degree rotation



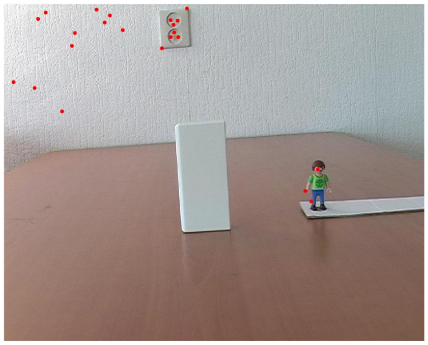
(c) No rotation



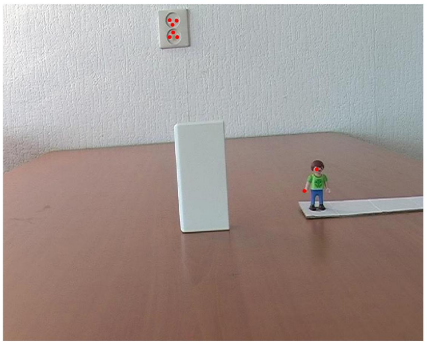
(d) 90 degree rotation

Figure 4: Harris corners with varying sigma and threshold

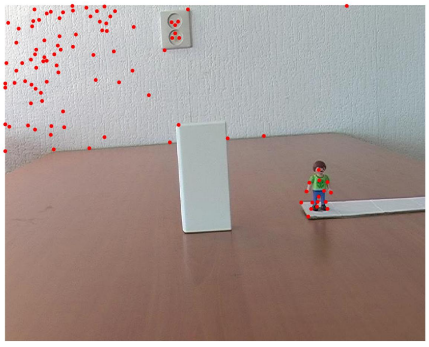
(a) σ 0.5, 100



(b) σ 0.5, 400



(c) σ 1, 100



(d) σ 1, 400

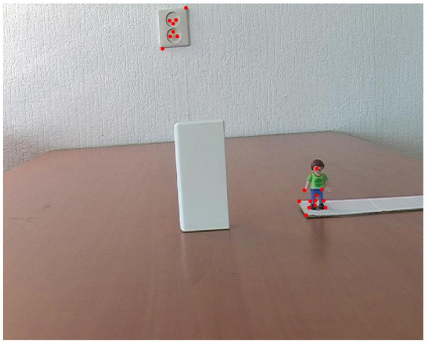


Figure 5: Optical flow in sphere and synthetic image