CS 415: Computer Vision

Mini Project 4

Author:
Gaetano Coppoletta
Email:
gcoppo2@uic.edu

November 8, 2022

1 Question 1

In order to understand the difference between invariance and equivariance, first we define those concepts.

- **Invariance**: image is transformed and corner locations do not change
- **Equivariance**: if we have two transformed versions of the same image, features should be detected in corresponding locations

We want corner locations to be invariant to photometric transformations and equivariant to geometric transformations. Now we will discuss invariance and equivariance of the Harris corner detector for the image transformations below listed below. To give some examples and to justify the answers some images representing the transformation applied to lena.png are provided.

- **Traslation**: corner location is equivariant with respect to traslation because it is a geometric transformation. The image below shows an example.

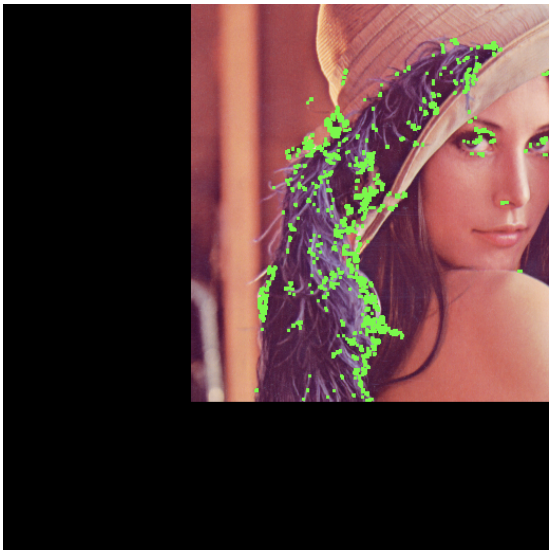


Figure 1: lena.png traslated

- **Rotation**: corner location is equivariant with respect to rotation because it is a geometric transformation as we can see from the image below.

Figure 2: lena.png rotated

- **Horizontal flipping**: corner location is equivariant with respect to horizontal flipping because it is a geometric transformation as we can see from the image below.

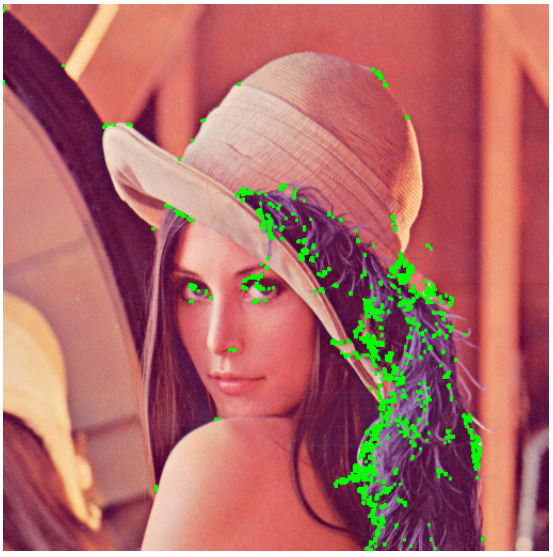


Figure 3: lena.png flipped

- **Scaling**: corner location is neither invariant nor equivariant with respect to scaling. We should use a larger patch for larger scale but we do not know the scale in advance. Below an example is reported.

Figure 4: lena.png scaled

- **Adding a constant to every pixel intensity**: corner location is invariant with respect to adding a constant to every pixel intensity but it is not invariant with respect to multiplying a constant to every pixel intensity.

2 Question 2

The benefit of using the histogram is that the feature descriptor will be invariant to local deformation. The benefit of using the gradient is that the feature descriptor will be invariant to brightness change. Using the cells makes the SIFT descriptor variant to the spatial layout, changing the order of the cells has an impact on the descriptor.

3 Question 3

For each of the desired transformations below, we want to provide the 3x3 transformation matrix that could be used to transform an arbitrary point in the homogeneous coordinate. We suppose that the top-left corner of an image is the origin, and the x and y axes point to the right and bottom, respectively.

- **Shift to the right by 100 pixels**: the transformation matrix is M = $\begin{bmatrix} 1 & 0 & 100 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- **Rotate around the origin in the clockwise direction by 45 degrees**: the transformation matrix is M= $\begin{bmatrix} cos(45) & -sin(45) & 0 \\ sin(45) & cos(45) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- **Rotate around the point(20,20) in the counterclockwise direction by 90 de-**

grees: the transformation matrix is $M = M3 * M2 * M1$ where M1= $\begin{bmatrix} 1 & 0 & -20 \\ 0 & 1 & -20 \\ 0 & 0 & 1 \end{bmatrix}$ M2 = $\begin{bmatrix} cos(90) & -sin(90) & 0 \\ sin(90) & cos(90) & 0 \\ 0 & 0 & 1 \end{bmatrix}$ M3 = $\begin{bmatrix} 1 & 0 & 20 \\ 0 & 1 & 20 \\ 0 & 0 & 1 \end{bmatrix}$.

Doing the calculus the transformation matrix is M = $\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 40 \\ 0 & 0 & 1 \end{bmatrix}$

4 P1

In this section we create an all white RGB image of resolution 500x500 and we apply some geometric transformation to pixel coordinates. Using the cv2.circle() method we draw a solid red circle with a radius 10 at a=(100,40). The result is listed below.

Figure 5: Red circle

Now we rotate a around the origin in the clockwise direction by 60 degrees to get a new point b. We draw a solid green circle at b, the result is listed below.

Figure 6: Green circle

After that we draw a solid black circle at c=(100,100), the result is listed below.

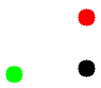


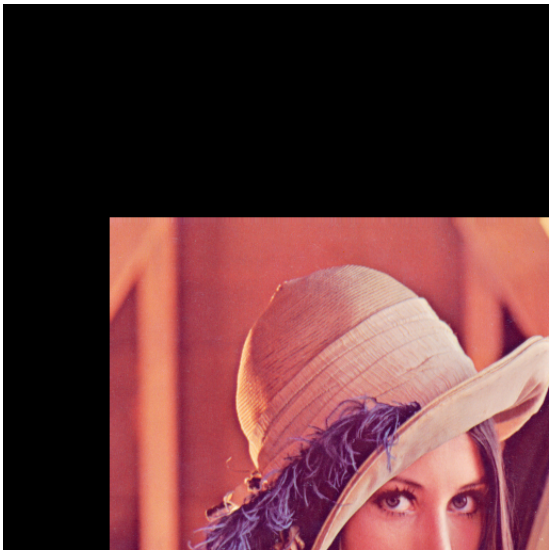
Figure 7: Black circle

Now we rotate a around c in the clockwise direction by 60 degrees to get a new point d and we draw a solid blue circle at d.

Figure 8: Blue circle

5 P2

In this section we use the function cv2.warpAffine() to apply some transformation to lena.png. Move the image to the right by 100 pixels and to the bottom by 200 pixels. The result is listed below.



Flip the image horizontally with respect to the image center. The result is listed below.

Rotate the image around the origin in the clockwise direction by 45 degrees. The result is listed below.

Rotate the image around the image center in the clockwise direction by 45 degrees. The result is listed below.