Computer Vision I

Heiko Neumann, Christian Jarvers
Institute of Neural Information Processing
Ulm University

Assignment 4: Submission date (Moodle): July 6, 2022, 12:00.

Both the Moravec operator and the structure tensor can be used to identify image locations with 2-dimensional structure, such as corners. Apply both methods to the image `shapes1.png` (see Figure 1). Load it and convert it to grayscale.

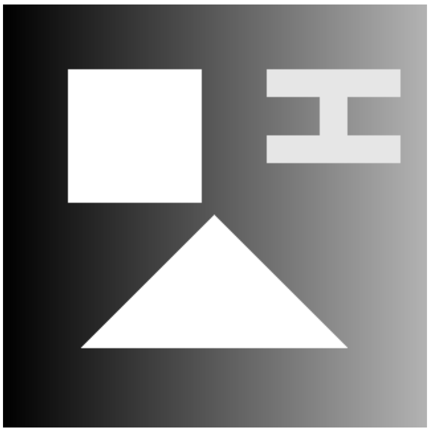

Figure 1: Image `shapes1.png` with simple geometric shapes

1 Moravec Operator (4 points)

The Moravec-Operator detects corners by comparing directional differences in a local neighborhood. Use the differences in four directions $(0°, 45°, 90°, 135°)$. The decision criterion is the (squared and normalized) minimum difference.

1. For each of the given directions, calculate the directional differences for a 1-pixel shift in that direction. Square the resulting differences to achieve polarity invariance.
2. In each orientation channel, aggregate the differences over a local 5×5 neighborhood (e.g., convolve with a box filter).
3. In order to achieve contrast invariance, normalize the aggregated coefficients at each

location using an appropriate scheme (c.f. part VI, slide 22). Note that the normalization should not lead to errouneous responses in homogenous areas.

4. At each location, determine whether a corner is present by comparing the minimum coefficient to a threshold θ. Identify an appropriate threshold value.
5. Plot your results.

Apply the same procedure to the image `shapes1_noisy.png`. How does the outcome differ from the first image? Explain the underlying reasons and comment on how a better processing result could be achieved.

2 Receiver Operator Characteristics (2 points)

The performance of the Moravec operator on `shapes1_noisy.png` varies a lot with the chosen value for threshold θ. A receiver operator characteristic (ROC) curve is a useful tool to understand how a classifier or detector (in this case, a detector for keypoints) behaves for different threshold settings. It plots the true positive rate against the false positive rate for a range of threshold values (see part V, slide 70).

- Load the image `keypoints.png`. In this image, the keypoints of `shapes1_noisy.png` are encoded as white pixels, the non-keypoint locations as black pixels. Convert the image to an array of boolean values.
- Write a function `roc` that takes an image and a ground truth array as inputs. It should compute the true positive rate (TPR), i.e., the number of correctly detected keypoints divided by all keypoints and the false positive rate (FPR), i.e., the number of non-keypoints locations that were falsely detected as keypoints divided by the number of all non-keypoint locations.
- Use your implementation of the Moravec operator from task 1 to compute the TPR and FPR on the images `shapes1.png` and `shapes1_noisy.png`. The values of threshold θ used in the Moravec operator should range from 0 to 1 in steps of 0.01.
- Plot the two ROC curves.

3 Structure Tensor (4 points)

The structure tensor characterizes the distribution of orientations in a local patch and can be used to identify 2-dimensional structure.

1. Use the Sobel operator to calculate the discrete image gradient $\nabla I = (I_x, I_y)^T$. Calculate and plot the magnitude $\|\nabla I\|$.
2. The structure tensor characterizes the local image structure. It is defined as

$$\boldsymbol{S} = G_\sigma * \nabla I \cdot (\nabla I)^T = \begin{pmatrix} G_\sigma * I_x^2 & G_\sigma * I_x I_y \\ G_\sigma * I_y I_x & G_\sigma * I_y^2 \end{pmatrix}.$$

where G_σ is a Gaussian weighting function. Use the function `gaussian()` from module `ImageFiltering.Kernel` to construct a Gaussian filter G_σ with $\sigma = 3$.

3. The eigenvalues of the structure tensor (which are found by principal component analysis)

$$\mathrm{PCA}(\boldsymbol{S}) = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

characterize the type of local 2D structure in the image. Calculate the eigenvalues and the corresponding eigenvectors of $\boldsymbol{S}$ at each position, and save the eigenvalues in descending order (i.e. $\lambda_1 \geq \lambda_2$). You can use the function `eigen()` from the `LinearAlgebra` package.

4. Plot the eigenvector corresponding to the first eigenvalue at the different image positions (`quiver`). Discard vectors where the corresponding eigenvalue is too small. For clarity, it may help to subsample the plot to reduce the number of arrows drawn. In extra plots, show also the first and second eigenvalues.

5. Detect the different structure types (homogeneous area, edge, corner) in the image. Compute three binary images for each structure type denoting if the type is present or not. Use the eigenvalues and define appropriate thresholds (c.f. part VI, p. 42). Additionally, display all three structure types in a single image, using an appropriate color scheme.

Submission procedure

- Submission exclusively in Moodle.
- Please name all scripts or notebooks with the current assignment and exercise number, e.g. `sh01ex02.jl` for the second assignment of the first assignment sheet.
- Please present your results in a pdf-document or Jupyter notebook with
  - your name and matriculation number
  - and a brief description of your results and images.
- Please submit all files as a zip-document.

Have fun!