Multiple View Geometry: Exercise Sheet 5

Prof. Dr. Daniel Cremers, Robert Maier, Rui Wang, TU Munich
`http://vision.in.tum.de/teaching/ss2016/mvg2016`

Exercise: June 9th, 2016

Part II: Practical Exercises

This exercise is to be solved **during the tutorial**.

Harris Corner Detector
In this exercise you will implement the Harris corner detector.
Hint: Again, it is possible to implement this exercise without loops. However, we recommend to start with a loop-based version, and only afterwards replace the loops by single function calls. For quick testing, we provide a small image `small.png`.

1. Download `ex5.zip` from the website.

2. Compute the structure tensor as introduced in the lecture for every pixel of the image. Implement the missing parts in `getM.m`.

   (a) Compute the image gradients I_x and I_y (in x- and y-direction) using central differences.

   (b) As weighting function use a two-dimensional Gaussian Kernel with a standard deviation of $\sigma_w = 2$. Compute the kernel size k (and hence the integration window size) as $k = 2 * (2 * \sigma_w) + 1$.
   Hint: you can use `fspecial` *(see Exercise Sheet 1).*

   (c) The structure tensor M for pixel x, y is a 2×2 matrix, which summarizes the structure of the image in the vicinity (determined by σ_w) of the pixel x, y. It can e.g. be interpreted as (weighted) covariance matrix of the image gradient around the pixel, and is computed as

$$M(x, y) := \sum_{(x', y')} w(x' - x, y' - y) \begin{pmatrix} I_x^2(x', y') & I_x(x', y')I_y(x', y') \\ I_x(x', y')I_y(x', y') & I_y^2(x', y') \end{pmatrix} \quad (1)$$

   where I_x and I_y are the x and y image gradients, and w is the weighting function.
   Hint: use `conv2`.

3. Compute the scoring function $C(x, y) := \det(M(x, y)) - \kappa \, \text{trace}^2(M(x, y))$ using $\kappa = 0.05$. Therefore, complete the missing parts in `getHarrisCorners.m`. Visualize the scoring function using `imagesc`. *Hint: if you cannot see much, try to display a non-linearly transformed scoring function, e.g. sign(C)* $\cdot |C|^{\frac{1}{4}}$.

4. Find all pixels x for which $C(x) > \theta$, and which are a local maximum of the scoring function, i.e., all four adjacent pixel have a lower score (non-maximum suppression). Use $\theta = 10^{-7}$. Display the found Harris Corners using the provided function `drawPts`.

5. Try different values for σ_w - what do you observe?

Patch Tracking

Compute the local velocity of each pixel from `img1` to `img2` in file `getFlow.m`.

1. Again, compute the image gradients I_x and I_y as well as the time derivative I_t.
2. Compute the structure tensor (re-use the first exercise), considering also the Gaussian weighting function.
3. Compute the local velocity of each pixel using the formula from the slides.
4. Visualize your result using `imagesc`.