Computer Vision I, SS13 - Homework Assignment 4

July 05, 2013

Prof. Stefan Roth
Qi Gao
Thorsten Franzel
Stephan Richter

This assignment is due on July 19th, 2013 at 23:55.

Please see the previous assignments for general instructions and follow the handin instructions from there.

Problem 1 - Optical Flow (8 points)

For estimating the optical flow, the method of Lucas and Kanade minimises the local energy

$$E\big(\mathbf{u}(x_0,y_0)\big) = \sum_{(x,y)\in\mathcal{N}_\rho(x_0,y_0)} \Big(f_x(x,y,t)u + f_y(x,y,t)v + f_t(x,y,t)\Big)^2,$$

where u and v are horizontal and vertical components of optical flow vector $\mathbf{u}$, $\mathcal{N}_\rho(x_0,y_0)$ is the spatial neighbourhood window of size ρ around $(x_0,y_0)^\mathrm{T}$.

By setting the derivatives w.r.t. u and v to be zero, we can get following systems of equations:

$$\begin{bmatrix} \sum_{\mathcal{N}_\rho} f_x^2 & \sum_{\mathcal{N}_\rho} f_x f_y \\ \sum_{\mathcal{N}_\rho} f_x f_y & \sum_{\mathcal{N}_\rho} f_y^2 \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum_{\mathcal{N}_\rho} f_x f_t \\ -\sum_{\mathcal{N}_\rho} f_y f_t \end{bmatrix}.$$

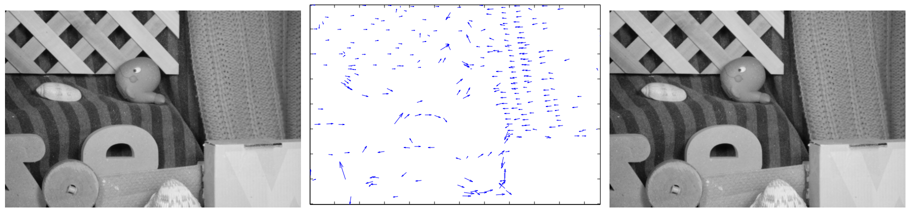
For simplicity, we can replace the hard window $\mathcal{N}_\rho$ by convolution with a Gaussian smoothing filter G_ρ and get

$$\underbrace{\begin{bmatrix} G_\rho * (f_x^2) & G_\rho * (f_x f_y) \\ G_\rho * (f_x f_y) & G_\rho * (f_y^2) \end{bmatrix}}_{A} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -G_\rho * (f_x f_t) \\ -G_\rho * (f_y f_t) \end{bmatrix}.$$

Note the matrix A is just the structure tensor, which we have known for Harris points detection.

Tasks:

Compute the optical flow from the 9th frame (`frame09.png`) to the 10th frame (`frame10.png`) of the rubber-whale sequence and show the estimated flow vectors as in the following figure.



- Start by loading and presmoothing the images with a Gaussian filter ($\sigma = 2$ and size 25×25).

- Now calculate the spatial and temporal derivatives from the smoothed images. Use central differences for estimating the spatial derivatives f_x and f_y and forward differences for the temporal derivative f_t.
- As full flow can only be estimated at the points for which A has full rank (i.e. corner points), you should use the Harris detector to find all feature (corner) points in the 9th frame (with the parameters $\sigma = 1$, filter size 15×15, threshold 500). Note that you do this in the original image (before presmoothing).
- Compute the coefficients of the linear system of equations. For the Gaussian window G_ρ, use $\sigma = 2$ and window size 11×11.
- Calculate the optical flow by solving for u and v.
- Show the flow vectors by using the Matlab built-in `quiver()`.

Problem 2 - Fundamental Matrix (10 points)

In this problem, you will use the 8-point algorithm to compute the fundamental matrix between two images `a4p2a.png` and `a4p2b.png`, which are shown as following:

Tasks:

Load the coordinates of 8 point correspondences from the file `impts.m`. Plot these points into the images to verify that they correspond. Then write a function `fundamental_from_8.m`, which takes the 8 correspondences between two images as arguments, and outputs the fundamental matrix. The function performs the following steps:

- Condition the image coordinates numerically, by moving their mean to $[0,0]^{\mathrm{T}}$ and scaling them such that coordinates are included in the interval between 1 and -1. The transformation should be done via a matrix multiplication, since you need the two transformation matrices later to undo the conditioning.
- Using the conditioned image points, build the homogeneous linear equation system for the elements of the fundamental matrix, and solve it using singular value decomposition.
- The preliminary fundamental matrix is not yet exactly of rank 2, due to noise and numerical errors. Enforce the rank constraint using SVD.
- So far you still have the fundamental matrix for the conditioned image points. Transform it to obtain the one for the original pixel coordinates. This is done by matrix multiplication.

Finally, check the correctness by verifying that the epipolar constraints are satisfied (up to a small residual) for all pairs of corresponding points.

Problem 3 - Image Stitching (27 points)

In this problem, you will use the RANSAC principle to robustly estimate the homography between two images. Then you can register two partially overlapping images and build a panorama image, as shown in the following figure.

A: Homography estimation and RANSAC

From the lecture we know that the homography between two images can be computed using at least four point correspondences. In practice, however, it is hard to directly get four correct correspondences. By matching the feature vectors obtained using interest point detectors and descriptors, we can find putative correspondences which also contain a lot outliers. To robustly estimate the homography from all putative matches, we can apply the RANSAC principle.

Tasks:

1. Load the images `a4p3a.png` and `a4p3b.png` taken by a rotating camera. Apply the Harris detector and SIFT descriptor to obtain the feature vectors from the overlapping regions of these two images (i.e. right half of the first image and left half of the second). Please use the parameters $\sigma = 1.4$ (for both Harris and SIFT), filter size 15×15, threshold 1500 and ignore the feature points within a 10-pixel wide boundary.

2. Write a function `find_matches.m`, which takes two sets of descriptors D_1 and D_2 as the input and find the matching descriptor pairs using the χ^2 distance. Note that the two feature points in a pair should be the nearest neighbours to each other.

3. Use your above function and computed SIFT feature vectors to find the putative matches. Display the these matches with connecting lines.

4. Write a function `compute_homography.m`, which takes the 4 correspondences between two images as arguments, and outputs the homography matrix. Do not forget to do conditioning due to numerical stability, as in problem 2.

5. Estimate the amount of iterations needed to draw an uncontaminated sample of 4 corresponding point pairs with a probability of 99%. Conservatively guess a probability of the proportion of inliers in your set of point pairs.

6. Randomly draw a sample of four corresponding point pairs and estimate the corresponding homography using your homography function from above task.

7. Test the homography: for each putative correspondence, compute the (squared) distance of one point to the transformed other point
$$d^2(H, \mathbf{x}_1, \mathbf{x}_2) = \|H\mathbf{x}_1 - \mathbf{x}_2\|^2 + \|\mathbf{x}_1 - H^{-1}\mathbf{x}_2\|^2,$$
where $\|\cdot\|$ means the $\mathscr{L}^2$ norm of a vector. Count the number of inliers (consistent correspondences), for which this distance is smaller than a threshold.

8. Repeat above two steps (6–7) for an appropriate number of times and remember the homography that has the highest number of inliers.

9. Finally, show the 4 point correspondences wrt the obtained homography using RANSAC as well as all corresponding inlier matches on the two images.

B: Panorama stitching

With the estimated homography, you can now stitch the two images into a panorama image. In case that you can not successfully estimate the homography from above tasks, use the provided homography matrix stored in `H.mat`.

Tasks:

- Transform the image `a4p3b.png` into the image plane of `a4p3a.png` using the homography. You should use bi-linear interpolation to get the pixel values. (Matlab build-in function `interp2` might be useful.)
- Initialize one larger image with 700-pixel width and the same height as the given images. Fill the left 300 pixel columns with pixels from image `a4p3a.png`. The rest part of the panorama image should be reconstructed from transformed `a4p3b.png`. Finally, display your panorama image.