Industrial & Medical Image Processing Lab

Session 02

Document the results of the exercises in a protocol. Upload both, protocol and source code in a zip-file.

Exercise 2.1

Harris Corner Detector

Theory: Use a local window and determine differences in intensity when the local window encounters small shifts in all directions. The weighted SSD (sum of squared differences) is then defined by

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2$$

$w(x,y)$	Window function (Gaussian smoothing window)
$I(x,y)$	Intensity at position (x,y)
$I(x+u, y+v)$	Intensity in the shifting window $(x+u, y+v)$

$I(x+u, y+v)$ can be approximated via a Taylor expansion, i.e.

$$I(x+u, y+v) \approx I(x,y) + I_x(x,y)u + I_y(x,y)v$$

Hence,

$$E(u,v) \approx \sum_{x,y} w(x,y)\left[I_x(x,y)u + I_y(x,y)v\right]^2$$

Further, $E(u,v)$ can be expressed in matrix form, therefore

$$E(u,v) \approx (u \quad v)\, S \begin{pmatrix} u \\ v \end{pmatrix}$$

whereby S represents the so called structure tensor (see lecture). Instead of the expensive computation of many Eigen values for the score, one can use the original Harris measure:

$$R_{Harris} = \left(S_x S_y - S_{xy}^2\right) - k\left(S_x + S_y\right)^2 \text{ with } k = 0.04$$

Using a Non-Maximum-Suppression and thresholding onto R_{Harris} corners in the image can be detected.

(a) Realize a Harris corner detector by using the structure tensor and detect all corners in the test image (you are warmly welcome to use checkerboard() with different parameter): 4P

```
Img = checkerboard(80,3,3);
```

- Visualize the Harris score and show the detected corners as an overlay in the original image.
- Use a normalized derivative filter after Gauss for computation of the structure tensor.
- For Non-Maximum-Suppression use the function *nonmaxsuppts()* available in Moodle
- Don't forget to threshold!

(b) Use the image "*gantrycrane.png*" (available in Matlab) with the algorithm implemented in (a). Evaluate the results with different thresholds. What is the influence of the Gaussian smoothing window (size, parameter) on the detection result? Therefore, vary kernel size and σ for the smoothing window. 1P

(c) Can you achieve a similar result with the Matlab function „*corner()*"? 1P

(d) Understand and explain Non-Maximum-Suppression 1P

(e+) Modify (a) in such way, so only 5, 15 or 25 of the most prominent edges are detected in the image „*gantrycrane.png"*. 2P

Exercise 2.2

CT Reconstruction

In this exercises a CT projection/backprojection according to the theory given in the lecture should be performed on an image and subsequently its content reconstructed. Use '*square.jpg*' during the implementation. For the final test image use *img=phantom(256);*

(a) Load the image and in a first step compute the projections in the interval [0, 360] degrees in steps of 1°. Use Matlab's *imrotate* and simply sum up the columns. Implement an animation showing the rotated image, its projected 1D signal and the evolving sinogram. Hint: Take care that the output image from *imrotate* takes the same size as the input image 2P

(b) Implement a simple backprojection process. Therefore, loop over each projection: use *repmat* to replicate a single projection over the entire size of the output image (same size of input image) and the rotate the image according to the angle the projection has been acquired. This equals the backprojection of one single projection. Sum up each backprojection in a loop and don't forget to normalise the output image according to the numbers of images summed up. Display the resulting backprojected image after each iteration in order to see how the original image becomes reconstructed 2P

(c) Use Matlab's *radon* and *iradon* to perform a filtered backprojection with a Ram-Lak, Shepp-Logan, Cosine, Hamming and Hann filter kernel (you can use the projection from a) as input as well, but you need to zero-pad the image in order to not lose some information at the border). Which filter method delivers the best image quality? Analyze visually and additionally use the Peak Signal to Noise Ratio (PSNR) and the Mean Squared Error (MSE) to quantify the results. How are PSNR and MSE computed and what do they express? Any weaknesses with the two measures? 3P

(d+) Given the projections in *projections.mat*: what's in the image? 1P