Title: **Image processing and recognition**			
Credits: 4 **Semester: 1** **Compulsory: Yes**			
Format	Lectures 30	Guided project 15 Laboratory Tutorials	Private study 30
Lecturer(s): Robert Sitnik			

Objectives:

Become familiar with techniques and algorithms of digital image acquisition, processing and recognition. To learn about processing and analysis of image sequences. To know modern applications of imaging techniques. Become familiar with techniques and algorithms of digital image improvement, enhancement, template localization and correlation. To practice of development and implementation of selected algorithms.

Contents:

1. Introduction. Basic definitions: image acquisition, processing and recognition, computer graphics and animation. Digital vs. analog representation. Sampling and quantization. CCD/CMOS detector vs. human eye. Electromagnetic spectrum. Colour spaces.
2. Geometrical operations. Image correction. Camera calibration.
3. Arithmetical operations. LUT. Histogram and its modifications. Thresholding.
4. Fourier transform. Numerical implementation of DFT and FFT. Gabor transform. Wavelet transform. Hough transform. Practical applications.
5. Image filtering: low and high band, edge detectors and image improvement. Image and frequency space filtering. Convolution and correlation.
6. Morphological operations. Skeletonization. Image segmentation.
7. Image recognition. Feature vector and feature space. Features: geometrical, topological, moments and statistical. Exemplary problem discussion.
8. Correction of opto-electronic imaging errors (thermal noise, quantization, non uniform illumination, nonlinear intensity transfer function). Methods of opto-electronics errors correction.
9. Image enhancement techniques (contrast modification, linear and nonlinear histogram transformations and filtering, statistical analysis of local intensity distribution).
10. Image reconstruction (nonlinear point operations, linear and nonlinear statistical analysis, Wiener filtration, local polynomial fitting, pseudo-inverse SVD).
11. Fitting of geometrical primitives (line, circle, ellipse, polynomial, 2D polynomials). Algorithms: Monte Carlo, least square error minimization and iterative. Non uniform weights of pixels.
12. Template matching. Methods of template matching translation/rotation/scale invariant (correlation, Monte Carlo, iterative).

13. Analysis of image sequences. Variants of spatio-temporal image analysis techniques (arithmetic, filtering and morphology). Tracking of objects as a rigid body.

Abilities:
Knowledge of techniques and algorithms of digital image acquisition, processing and recognition. Ability to process and analyse of image sequences. Understanding of algorithms of digital image improvement, enhancement, template localization and correlation. Practice in development and implementation of digital image processing algorithms.

Assessment: 40% points from two tests, 60% project grade

Practical work:

Individual C++ programming project:
Individual image processing and recognition C++ programming task. Goal to achieve is the localization and recognition of objects in image set. Student starts from scene arrangement (illumination, background, detector, lens). Capturing of learning image sequence. Development and implementation of processing path. Establishing of optimal feature space for recognition of requested objects. Definition of features values for each class of objects. The project has four phases:

- UML modelling where student creates model of his processing including use case, interaction and class diagrams,
- implementation phase where whole algorithm is developed and implemented,
- testing and optimization phase where application is tested and optimized,
- documentation phase.

Literature:
A.R. Weeks, *Fundamentals of Electronic Image Processing,* IEEE/SPIE Press, New York, 1996
W.K. Pratt, *Digital Image Processing*, 3rd edition, John Willey & Sons, New York, 2001.
J.C. Russ, *The Image Processing Handbook*, 3rd edition, CRC Press, London, 1998.

Prerequisites: Engineering course of mathematics. Fundamentals of C programming.