# Open Informatics – International Computer Science Program

## Master program on artificial intelligence and computer vision

Michal Pěchouček, Jiří Matas, Filip Železný, Radim Šára, ...*

document generated at: 9:22, February 19, 2013

## Contents



*document preparation Tomáš Svoboda

## 1 Study plans

### 1.1 Master program Open Informatics, AI major/Vision minor

| **Term** | Mandatory courses | Mandatory courses ai/vision | facultative courses | | |
|---|---|---|---|---|---|
| 1 (winter) | AE4M33PAL<br>2p+2c<br>Advanced algorithms | AE4M33RZN<br>2p+2c<br>Advanced Methods for Knowledge Representation | AE4M33DZO<br>2p+2c<br>Digital image | AE4B33RPZ<br>2p+2c<br>Pattern Recognition and Machine Learning | |
| 2 (summer) | AE4M01TAL<br>3p+1s<br>Theory of Algorithms | AE4M35KO<br>3p+2c<br>Combinatorial Optimization | AE4M36PAH<br>2p+2c<br>Planning and game playing | AE4M33BIA<br>2p+2c<br>Bio Inspired Algorithms | AE4M33MPV<br>2p+2c<br>Computer Vision Methods |
| 3 (winter) | AE4M36MAS<br>2p+2c<br>Multiagent Systems | AE4M33SAD<br>2p+2c<br>Machine Learning and Data Analysis | AE4M33TDV<br>2p+2c<br>3D Computer Vision | AE4M39VG<br>2p+2c<br>Computational Geometry | AE4M99SVP<br>TBD<br>Software or Research Project |
| 4 (summer) | | | | AE4M33AU<br>2p+2c<br>Automatic Reasoning | AE4M99DIP<br>TBD<br>Master Thesis |

### 1.2 Master program Open Informatics, Vision major/AI minor

| Term | Mandatory courses | Mandatory courses ai/vision | facultative courses | | |
|---|---|---|---|---|---|
| 1 (winter) | AE4M33PAL<br>2p+2c<br>Advanced algorithms | AE4M33RZN<br>2p+2c<br>Advanced Methods for Knowledge Representation | AE4M33DZO<br>2p+2c<br>Digital image | AE4B33RPZ<br>2p+2c<br>Pattern Recognition and Machine Learning | |
| 2 (summer) | AE4M01TAL<br>3p+1s<br>Theory of Algorithms | AE4M35KO<br>3p+2c<br>Combinatorial Optimization | AE4M36PAH<br>2p+2c<br>Planning and game playing | AE4M33MPV<br>2p+2c<br>Computer Vision Methods | AE4M33GVG<br>2p+2c<br>Theoretical foundations of computer vision, graphics, and interaction |
| 3 (winter) | AE4M36MAS<br>2p+2c<br>Multiagent Systems | AE4M33SAD<br>2p+2c<br>Machine Learning and Data Analysis | AE4M33TDV<br>2p+2c<br>3D Computer Vision | AE4M39VG<br>2p+2c<br>Computational Geometry | AE4M99SVP<br>TBD<br>Software or Research Project |
| 4 (summer) | | | | AE4M33AU<br>2p+2c<br>Automatic Reasoning<br>**or:**<br>AE4M33BIA<br>2p+2c<br>Bio Inspired Algorithms | AE4M99DIP<br>TBD<br>Master Thesis |

# 2 Mandatory courses

## 2.1 AE4M33PAL: Advanced algorithms

**Title:** Advanced algorithms
**Lecturer:** RNDr. Genyk-Berezovskyj Marko
**Term:** winter
**Lectures + seminars or labs:** 2p+2c
**ECTS credits:** 6
**FEL www:** AE4M33PAL
**Anotation:** The advanced course of algorithms construction and analysis is dedicated to the students which have an interest to be able to evaluate in a experienced way effective and complex algorithms. The aim of the course is to acquaint with advanced algorithms such as advanced search and sorting algorithms, hash tables, tree structures used in searching, text searching, syntax analysis, Internet search algorithms principles (page-ranking), parallel algorithms.

**Syllabus:**

1. Abstract data types, signature, axioms.
2. The methods of specification and implementations of abstract data types.
3. Optimization of the basic sorting algorithms.
4. Hash tables, implementation, complexity.
5. Advanced search trees, AVL, B, Red-Black, operations on the search trees.
6. Balancing the trees implementation and the complexity of the operations.
7. Trees for multidimensional searching.
8. Structure and implementation of a lexical analyzer.
9. LL(1) grammars and transformations to LL(1) grammar.
10. Syntax tree, syntax analysis implemented via recursive descent.
11. Text search algorithms, K-M-P, B-M, Karp-Rabin, finite automaton.
12. Algorithm Page-rank, Internet search algorithms.
13. Parallel and distributed algorithms, motivation and overview.
14. Reserve.

**Labs/Seminars:**

1. Signature, axioms of basic ADT: queue, stack, list, array.
2. Implementation variants of ADT queue, stack, list, array.
3. Comparison of effectivity of different enhancements of QuickSort, MergeSort, HeapSort.
4. Comparison of effectivity of open and chained hashing, different hash functions.
5. Iterative and recursive processing of the AVL, B, Red-Black trees.
6. Balancing trees - practical implementations.
7. Trees for multidimensional searching.
8. Grammar of the statements and expressions of a simple language .
9. Decomposition tables, transformations to LL(1) grammars.
10. Implementation of the recursive descent.
11. Comparison of the text search algoritms on large datasets.
12. Efectivity of the data structures used in Page-rank algorithm.
13. Parallel number addition, measures of the complexity of the parallel algorithms.
14. Reserve.

**Literature:**


1. Sara Baase, A. van Gelder, Computer Algoritms, Addison Wesley, 2000


**Notes from Faculty Information System:**

### 2.2 AE4M01TAL: Theory of Algorithms

**Title:** Theory of Algorithms
**Lecturer:** TBD
**Term:** summer
**Lectures + seminars or labs:** 3p+1s
**ECTS credits:** 6
**FEL www:** AE4M01TAL
**Anotation:** The course brings several algorithms from the theory of graphs and cryptography. Stress is put on the analysis of time complexity of the algorithms presented. Further, basics of the theory of complexity are given. Next an example of randomized algorithms is given, it is the Miller-Rabin?s algorithm. When dealing with time complexity of specific algorithms suitable data structures will be given.

**Syllabus:**

1. Analyzing algorithms and problems, classifying functions by their growth rates, time complexity.
2. Basic graph algorithms, minimal spanning tree, Prim?s and Kruskal?s algorithms.
3. Algorithm for strongly connected components.
4. Matching in bipartite graphs.
5. Hungarian Algorithm.
6. Isomorphism of graphs.
7. Algorithms in cryptography, Eucleid?s Algorithm, RSA.
8. The classes of P and NP, polynomial reduction of problems.
9. NP-complete problems, examples of NP-complete problems.
10. Cook?s Theorem, reductions of NP-complete problems.
11. Heuristics for NP-complete problems, colouring.
12. Randomized algorithms, Miller-Rabin algorithm for primality testing.
13. Undecidable problems.

**Labs/Seminars:**

1. Basic graph algorithms, minimal spanning tree, Prim?s and Kruskal?s algorithms.
2. Algorithm for strongly connected components.
3. Hungarian Algorithm.
4. Eucleid?s Algorithm, RSA.
5. NP-complete problems, polynomial reduction of problems.
6. Vertex colouring of graphs.
7. Miller-Rabin algorithm.

**Literature:**


1. Kozen, D. C.: The design and Analysis of Algorithms, Springer-Vrelag, 1991
2. Harel, D: Algorithmics: The Spirit of Computing, Addison-Wesleyt Inc., Reading MA 1002
3. Talbot, J., Welsh, D.: Complexity and Cryptography, Cambridge University Press, 2006

### 2.3 AE4M35KO: **Combinatorial Optimization**

**Title:** Combinatorial Optimization
**Lecturer:** TBD
**Term:** summer
**Lectures + seminars or labs:** 3p+2c
**ECTS credits:** 6
**FEL www:** AE4M35KO
**Anotation:** The goal is to show the problems and algorithms of combinatorial optimization (often called discrete optimization; there is a strong overlap with the term operations research). Following the courses on linear algebra, graph theory, and basics of optimization, we show optimization techniques based on graphs, integer linear programming, heuristics, approximation algorithms and state space search methods.
We focus on application of optimization in stores, ground transport, flight transport, logistics, planning of human resources, scheduling in production lines, message routing, scheduling in parallel computers.

**Syllabus:**

1. Introduction of Basic Terms of Combinatorial Optimization, Example Applications and Test on Preliminary Knowledge
2. Integer Linear Programming - Algorithms
3. Problem Formulation by Integer Linear Programming
4. Heuristics, Test
5. Shortest Paths
6. Network Flows and Cuts
7. Multicommodity network flows
8. Dynamic Programming, Test
9. Knapsack Problem and Pseudo-polynomial Algorithms
10. Traveling Salesman Problem and Approximation Algorithms
11. Monoprocessor Scheduling
12. Scheduling on Parallel Processors
13. Project Scheduling with Time Windows
14. Constraint Programming

**Labs/Seminars:**

1. Introduction to the Experimental Environment and Optimisation library
2. Integer Linear Programming
3. Applications of Integer Linear Programming
4. Assignment of Term Projects
5. Branch and Bound Technique
6. Shortest Paths
7. Applications of Network Flows and Cuts
8. Presentation of Term Projects
9. Monoprocessor Scheduling - Earliest Deadline First
10. Approximation Algorithms - List Scheduling
11. Reserved
12. Test
13. Giving over Term Projects
14. Credits

**Literature:**

1. B. H. Korte and J. Vygen, Combinatorial Optimization: Theory and Algorithms. Springer, third ed., 2006.
2. J. Blazevicz, Scheduling Computer and Manufacturing Processes. Springer, second ed., 2001.
3. J. Demel, Grafy a jejich aplikace. Academia, second ed., 2002. TORSCHE http://rtime.felk.cvut.cz/scheduling-toolbox/

**Notes from Faculty Information System:**

### 2.4 AE4M33RZN: Advanced Methods for Knowledge Representation

**Title:** Advanced Methods for Knowledge Representation
**Lecturer:** Ing. Kléma Jiří Ph.D.
**Term:** winter
**Lectures + seminars or labs:** 2p+2c
**ECTS credits:** 6
**FEL www:** AE4M33RZN
**Anotation:** This course aims to deepen understanding of knowledge representation principles beyond the predicate logic formalism. Firstly, the course presents ontologies and description logic, the principle elements of semantic web. Then, attention will be paid to statements whose validity varies in time. Uncertainty makes the next issue to be discussed. Modal logic extends the classical logic with additional modalities, namely, possibility, probability, and necessity. Probabilistic graphical models associate the classical probabilistic theory with the graph theory. Fuzzy sets allow to represent vagueness.

**Syllabus:**

1. Introduction frames and ontologies.
2. Description logic language and its expressivity, interactions with rule-based systems.
3. Description logic inference, tableuax method.
4. Description queries forming and evaluation. Inconsistency in ontologies.
5. Tractable fragments of description logic. Present and future of semantic web.
6. Modal logic definitions and applications.
7. Temporal logic definitions and applications.
8. Uncertainty in knowledge-based systems role and representation.
9. Uncertainty and conditional independence introduction to probabilistic networks.
10. Probabilistic graphical models introduction, inference.
11. Dynamic models applications of probabilistic networks.
12. Fuzzy logic vagueness.
13. Fuzzy logic operations.
14. Fuzzy logic inference.

**Labs/Seminars:**

1. Introduction, ontological editor Protege.
2. OWL language modeling, examples.
3. Inference engine Pellet.
4. Query language SPARQL.
5. The first assignment OWL ontology for a selected domain, difference between OWA and CWA.
6. The first assignment autonomous working.
7. The first assignment autonomous working.
8. Conjunctive queries working with the ontology.
9. Expert and knowledge-based systems with uncertainty.
10. SW probabilistic modeling tools (Bayes Net Toolbox for Matlab, Bayesian Networks in Java).
11. The second assignment implementation of a probabilistic model.
12. The second assignment autonomous working.
13. Fuzzy sets.
14. Spare slot finishing, credits.

**Literature:**

1. Franz Baader , Diego Calvanese , Deborah L. McGuinness , Daniele Nardi , Peter F. Patel-Schneider, The Description Logic Handbook, Cambridge University Press, New York, NY, 2007.
2. Baader, F., Sattler U.: An overview of tableau algorithms for description logics ; Studia Logica, 69:5-40, 2001.
3. Charniak, E.: Bayesian Networks without Tears. AI Magazine 12(4): 50-63, 1991.
4. Pearl , J.: Causality: Models, Reasoning and Inference. Cambridge University Press, 2001.

**Notes from Faculty Information System:**

### 2.5 AE4M33DZO: Digital image

**Title:** Digital image
**Lecturer:** Prof.Ing. Hlaváč Václav CSc.
**Term:** winter
**Lectures + seminars or labs:** 2p+2c
**ECTS credits:** 6
**FEL www:** AE4M33DZO
**Anotation:** The subject teaches how to process two-dimensional image as a signal without interpretaion. Image acquisition, linear and nonlinear preprocessing methods and image compression will be studied. Studied topics will be practised on practical examples in order to obtain also practical skills.

**Syllabus:**

1. Image, its acquisition and physical foundation. Color. Projection through the lens.
2. Digital image. Contiguous pixels. Distance. Intensity histogram.
3. Brightness and geometric transformations, interpolation.
4. Processing in the spatial domain. Convolution. Corelation. Noise filtration.
5. Fourier transformation I.
6. Fourier transformation II. Sampling theorem.
7. Use of the Fourier transform for images. Filtration in the frequency domain.
8. Splines, B-splines. Advanced interpolation.
9. Image reconstruction. Edge detection.
10. Multiple scale image rocessing. Wavelet transformation.
11. Image compression.
12. Texture.
13. Mathematical morphology.
14. Image processing as a solution of partial dif. equations. Inpainting.

**Labs/Seminars:**

1. MATLAB. Homework 1 assignment (image acquisition).
2. Constultations. Solving the homework.
3. Constultations. Solving the homework.
4. Constultations. Solving the homework.
5. Homework 1 handover. Homework 2 assignment (Fourier transformation).
6. Constultations. Solving the homework.
7. Constultations. Solving the homework.
8. Constultations. Solving the homework.
9. Homework 2 handover. Homework 3 assignment (local preprocessing).
10. Constultations. Solving the homework.
11. Constultations. Solving the homework.
12. Constultations. Solving the homework.
13. Consultations. Homework 3 handover.
14. Written test. Presentation of several best student homeworks.

**Literature:**


1. Šonka, M., Hlaváč, V., Boyle, R.D.: Image processing, analysis and machine vision. 3. vydání, Thomson Learning, Toronto, Canada, 2007.
2. Svoboda, T., Kybic, J., Hlaváč, V.: Image processing, analysis and machine vision. The MATLAB companion, Thomson Learning, Toronto, Canada, 2007.


**Notes from Faculty Information System:**

### 2.6 AE4M36PAH: Planning and game playing

**Title:** Planning and game playing
**Lecturer:** Prof.Dr. Pěchouček Michal MSc.
**Term:** summer
**Lectures + seminars or labs:** 2p+2c
**ECTS credits:** 6
**FEL www:** AE4M36PAH
**Anotation:** This course provides an introduction to classical AI planning (linear, nonliner planning, graph-plan planning, heuristic planning, SAT-based planning) and game-tree representation and methods of adversarial search (such as minimax and alpha/beta pruning).

**Syllabus:**

1. planning problem representation and planning problem komplexity
2. linear planning, TOPLAN algorithm,
3. nonlineární planning, causal links thread resolution
4. Graf-oriented planning
5. planning by means of SAT
6. Introduction to game playing
7. Minimax, alfa-beta prunning
8. Advenced methods of adversarial planning
9. Hierarchical HTN planning
10. Heuristic planning
11. Contingency planning, temporal planning
12. Planning a probability
13. Planning in game playing

**Labs/Seminars:**

1. Planning problems
2. Semestral project specification: design and development of a general planner
3. - 5. Laboratories
4. Game playing algorithms
5. Semestral project specification: design and development of a game playing algorithm
6. - 12. Laboratories
7. Competition

**Literature:**


1. Nau, D., Ghallab, M., and Traverso, P. 2004 Automated Planning: Theory and Practice. Morgan Kaufmann Publishers Inc.
2. Russell, S. J. and Norvig, P. 2003 Artificial Intelligence: a Modern Approach. 2. Pearson Education.


**Notes from Faculty Information System:**

### 2.7 AE4M33BIA: Bio Inspired Algorithms

**Title:** Bio Inspired Algorithms
**Lecturer:** Ing. Kubalík Jiří Ph.D.
**Term:** summer
**Lectures + seminars or labs:** 2p+2c
**ECTS credits:** 6
**FEL www:** AE4M33BIA
**Anotation:** The students will learn some of the uncoventional methods of computational intelligence aimed at solving complex tasks of classification, modeling, clustering, search and optimization. Bio-inspired algorithms take advantage of analogies to various phenomena in the nature and society. The main topics of the subject are artificial neural networks and evolutionary algorithms.

**Syllabus:**

1. Introduction – relations to conventional optimization methods, black-box optimization, randomized search methods.
2. Introduction to artificial neural networks, history, typical tasks and their solutions, types of neural networks learning. Perceptron.
3. Supervised learning – approximation and classification, local and global units in neural networks. Multilayered perceptron, RBF networks, GMDH networks.
4. Unsupervised learning – clustering with neural networks, self-organization, Hebb's rule, Hopfield network, associative memory, ART networks.
5. Kohonen's self-organizing map (SOM), competitive learning, reinforcement learning.
6. Error back-propagation algorithm, universal approximation, Kolmogorov theorem.
7. Temporal sequences processing, recurrent neural networks, Elman network, back-propagation through time.
8. Simple genetic algorithm (SGA) – history, basic cycle, genetic operators, schema theorem.
9. Evolutionary algorithms with real representation – evolutionary strategy, crossover operators. Differential evolution (DE).
10. Neuroevolution – evolutionary techniques for neural network structure learning and parameter tunning. NEAT system.
11. Multiobjective optimization – dominance principle, Pareto-optimal solutions, multiobjective evolutionary algorithms (NSGA-II, SPEA2).
12. Genetic programming (GP) – tree representation, initialization, operators, strongly-typed GP, automatically defined functions (ADF).
13. Reserved.

**Labs/Seminars:**

1. Seminar organization. Black box neural network (MLP), approximation, classification, local search examples in Matlab.
2. Neural network software, Mathematica, Weka.
3. First assignment introduction (introduction to data).
4. Elaboration of the first assignment.
5. Elaboration of the first assignment.
6. First assignment presentation and evaluation.
7. Second assignment introduction (evolutionary algorithms).
8. Simple genetic algorithm (SGA). Influence of SGA parameters on its behaviour. Examples of evolutionary algorithms in Matlab.
9. Elaboration of the second assignment.

10. Successful applications of evolutionary algorithms.
11. Test.
12. Second assignment hand-in and evaluation.
13. Assignments.

**Literature:**


1. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edition, Prentice Hall, 1998
2. Rojas, R.: Neural Networks: A Systematic Introduction, Springer, 1996
3. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, Springer, 1998
4. Michalewicz, Z.: How to solve it? Modern heuristics. 2nd ed. Springer, 2004.

## 2.8 AE4M33MPV: Computer Vision Methods

**Title:** Computer Vision Methods
**Lecturer:** Prof.Ing. Matas Jiří Ph.D.
**Term:** summer
**Lectures + seminars or labs:** 2p+2c
**ECTS credits:** 6
**FEL www:** AE4M33MPV
**Anotation:** The course covers selected computer vision problems: search for correspondences between images via interest point detection, description and matching, image stitching, detection, recognition and segmentation of objects in images and videos, image retrieval from large databases and tracking of objects in video sequences.

**Syllabus:**

1. Introduction. Course map. Overview of covered problems and application areas.
2. Detectors of interest points and distinguished regions. Harris interest point (corner) detector, Laplace detector and its fast approximation as Difference of Gaussians, maximally stable extremal regions (MSER).Descriptions of algorithms, analysis of their robustness to geometric and photometric transformations of the image.
3. Descriptors of interest regions. The local reference frame method for geometrically invariant description. The SIFT (scale invariant feature transform) descriptor, local binary patterns (LBP).
4. Detection of geometric primitives, Hough transfrom. RANSAC (Random Sample and Consensus).
5. Segmentation I. Image as a Markov random field (MRF). Algorithms formulating segmentation as a min-cut problem in a graph.
6. Segmentation II. Level set methods.
7. Inpainting. Semi-automatic simple replacement of a content of an image region without any visible artifacts.
8. Object detection by the "scanning window" method, the Viola-Jones approach.
9. Using local invariant description for object recognition and correspondence search.
10. Tracking I. KLT tracker, Harris and correlation.
11. Tracking II. Mean-shift, condensation.
12. Image Retrieval I. Image descriptors for large databases.
13. Image Retrieval II: Search in large databases, idexation, geometric verification
14. Reserve

**Labs/Seminars:**

1. - 5. Image stitching. Given a set of images with some overlap, automatically find corresponding points and estimate the geometric transformation between images. Create a single panoramic image by adjusting intensities of individual images and by stitching them into a single frame.
2. - 9. Segmentation and impainting. Implement a simple impainting method, i.e. a method allowing semi-automatic simple replacement of a content of an image region without any visible artifacts.
3. -12. Detection of a instance of a class of objects (faces, cars, etc.) using the scanning window approach (Viola-Jones type detector).
4. -14. Submission and review of reports.

**Literature:**


1. M. Sonka, V. Hlavac, R. Boyle. Image Processing, Analysis and Machine Vision. Thomson 2007
2. D. A. Forsyth, J. Ponce. Computer Vision: A Modern Approach. Prentice Hall 2003


**Notes from Faculty Information System:**

### 2.9 AE4M33GVG: Theoretical foundations of computer vision, graphics, and interaction

**Title:** Theoretical foundations of computer vision, graphics, and interaction
**Lecturer:** Prof.Ing. Hlaváč Václav CSc.
**Term:** summer
**Lectures + seminars or labs:** 2p+2c
**ECTS credits:** 6
**FEL www:** AE4M33GVG
**Anotation:** We will explain fundamentals of image and space geometry including Euclidean, affine and projective geometry, the model of a perspective camera, image transformations induced by camera motion, and image normalization for object recognition. Then we will study methods of calculations with geometrical objects in images and space, estimating geometrical models from observed data, and for calculating geometric and physical properties of observed objects. The theory will be demonstrated on practical task of creating mosaics from images and determining camera positions in space. We will build on linear algebra, probability theory and numerical mathematics and lay down foundation for other subjects such as computational geometry, computer vision, computer graphics, digital image processing and recognition of objects in images.

**Syllabus:**

1. Computer vision, graphics, and interaction - the discipline and the subject.
2. Modeling world geometry in the affine space.
3. The mathematical model of the perspective camera.
4. Relationship between images of the world observed by a moving camera.
5. Estimation of geometrical models from image data.
6. Optimal approximation using points and lines in L2 and minimax metric.
7. The projective plane.
8. The projective, affine and Euclidean space.
9. Transformation of the projective space. Invariance and covariance.
10. Random numbers and their generating.
11. Randomized estimation of models from data.
12. Construction of geometric objects from points and planes using linear programming.
13. Calculation of spatial object properties using Monte Carlo simulation.
14. Review.

**Labs/Seminars:**

1. Introduction, a-test 2-4 Linear algebra and optimization tools for computing with geometrical objects 5-6 Cameras in affine space - assignment I 7-8 Geometry of objects and cameras in projective space - assignment II 9-10 Principles of randomized algorithms - assignment III. 11-14 Randomized algorithms for computing scene geometry - assignment IV.

**Literature:**


1. P. Ptak. Introduction to Linear Algebra. Vydavatelstvi CVUT, Praha, 2007.
2. E. Krajnik. Maticovy pocet. Skriptum. Vydavatelstvi CVUT, Praha, 2000.
3. R. Hartley, A.Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2000.
4. M. Mortenson. Mathematics for Computer Graphics Applications. Industrial Press. 1999


**Notes from Faculty Information System:**

### 2.10 AE4M36MAS: Multiagent Systems

**Title:** Multiagent Systems
**Lecturer:** Prof.Dr. Pěchouček Michal MSc.
**Term:** winter
**Lectures + seminars or labs:** 2p+2c
**ECTS credits:** 6
**FEL www:** AE4M36MAS
**Anotation:** This course provides foundations of multi-agent systems and agent technologies. It provides a formal model of an agent, the concept of reactive, deliberative and deductive agent, BDI architecture,basics of inter agent communication and coordination. Introduction to distributed decision making and game theory will be also provided.

**Syllabus:**

1. Introduction to multi-agent system, formal model of an agent
2. Reactive agents
3. Deliberative and deductive agents
4. BDI architectura
5. Modal logic based formal model of multi-agent system
6. Inter-agent comunication
7. Cooperation and coordination
8. Negotiation
9. Distributed decision making
10. Coalition formation
11. Introduction to game theory
12. Development of scalable multi-agent systems
13. Multi-agent applications
14. Wrapups

**Labs/Seminars:**

1. Multi-agentni programming in 3APL
2. Multi-agentni programming in 3APL
3. - 6. Semestral project - design and development of deliberative agents
4. Multi-agentni programming in AGLOBE
5. Multi-agentni programming in AGLOBE
6. - 13. Semestral project - design and development of multi-agent system

**Literature:**


1. An Introduction to Multiagent Systems by Michael Wooldridge. Published in February 2002 by John Wiley and Sons (Chichester, England). ISBN 0 47149691X 340pp


**Notes from Faculty Information System:**

### 2.11 AE4M33SAD: Machine Learning and Data Analysis

**Title:** Machine Learning and Data Analysis
**Lecturer:** Doc.Ing. Železný Filip Ph.D.
**Term:** winter
**Lectures + seminars or labs:** 2p+2c
**ECTS credits:** 6
**FEL www:** AE4M33SAD
**Anotation:** The course explains advanced methods of machine learning helpful for getting insight into data by automatically discovering interpretable data models such as graph- and rule-based. Emphasis is given to the modeling of multi-relational data. The course will also address a theoretical framework explaining why/when the explained algorithms can in principle be expected to work.

**Syllabus:**

1. Cluster analysis, k-means algorithm, hierarchical clustering
2. Principal and independent component analysis.
3. Graphical probabilistic models
4. Grammar and Markov model learning
5. Association rules, the Apriori algorithm
6. Frequent subgraph search
7. Computational learning theory, concept space, PAC learning
8. PAC learning of logic forms
9. Classification rule learning. Algorithms AQ, CN2.
10. Inductive logic programming, least generalization, inverse entailment
11. Learning from logic interpretations, relational decision trees, relational features
12. Statistical relational learning: probablistic relational models
13. Statistical relational learning: Markov logic
14. Learning from texts

**Labs/Seminars:**

1. Entry test (prerequisite course RPZ). SW tools for machine learning (RapidMiner, WEKA)
2. Data preprocessing, missing and outlying values, clustering
3. Hierarchical clustering, principal component analysis
4. Graphical probabilistic model parameterization
5. Association rule and frequent subgraph search
6. Classification. Learning and ROC curves.
7. Bias vs. variance, ensemble classification
8. Individual task assignment
9. Individual work
10. Individual work
11. Submission of completed assignments
12. Inductive logic programming: the Aleph system
13. Statistical relational learning: the Alchemy system
14. Credits

**Literature:**


1. T. Mitchell: Machine Learning, McGraw Hill, 1997
2. P. Langley: Elements of Machine Learning, Morgan Kaufman 1996

3. T. Hastie et al: The elements of Statistical Learning, Springer 2001
4. S. Džeroski, N. Lavrač: Relational Data Mining, Springer 2001
5. L. Getoor, B. Taskar (eds): Introduction to Statistical Relational Learning, MIT Press 2007
6. V. Mařík et al. (eds): Umělá inteligence II, III, IV (Czech)

**Notes from Faculty Information System:**

### 2.12 AE4M33TDV: **3D Computer Vision**

**Title:** 3D Computer Vision
**Lecturer:** Doc.Dr.Ing. Šára Radim
**Term:** winter
**Lectures + seminars or labs:** 2p+2c
**ECTS credits:** 6
**FEL www:** AE4M33TDV
**Anotation:** This course introduces methods and algorithms for 3D geometric scene reconstruction from images. The student will understand these methods and their essence well enough to be able to build variants of simple systems for reconstruction of 3D objects from a set of images or video, for inserting virtual objects to video-signal source, or for computing ego-motion trajectory from a sequence of images. The labs will be hands-on, the student will be gradually building a small functional 3D scene reconstruction system.

**Syllabus:**

1. 3D computer vision, goals and applications, the course overview
2. Real perspective camera
3. Calibration of real perspective camera
4. Epipolar geometry
5. Computing camera matrices and 3D points from sparse correspondences
6. Autocalibration
7. Consistent multi-camera reconstruction
8. Optimal scene reconstruction
9. Epipolar image rectification
10. Stereoscopic vision
11. Algorithms for binocular stereoscopic matching, multi-camera algorithms, carving
12. Shape from shading and contour
13. Shape from texture, defocus, and color
14. Surface reconstruction

**Labs/Seminars:**

1. Labs introduction and overview, experimental data, entrance test
2. Camera calibration without radial distortion from a known scene
3. Camera calibration with radial distortion from a known scene
4. Computing epipolar geometry from 8 points
5. Computing epipolar geometry from 7 points, RANSAC
6. Constructing projection matrices from epipolar geometry, computing camera motion and scene structure
7. Autocalibration of intrinsic camera parameters
8. Consistent reconstruction of a many-camera system
9. Accuracy improvement by bundle adjustment
10. Time slot to finish all pending assignments
11. Epipolar rectification for stereoscopic vision
12. Stereoscopic matching by dynamic programming
13. 3D point cloud reconstruction
14. 3D sketch reconstruction

**Literature:**

1. R. Hartley and A. Zisserman. Multiple View Geometry. 2nd ed. Cambridge University Press 2003.
2. Y. Ma, S. Soatto, J. Kosecka, S.S. Sastry. An Invitation to 3D Vision. Springer 2004.

### 2.13 AE4M39VG: Computational Geometry

**Title:** Computational Geometry
**Lecturer:** TBD
**Term:** winter
**Lectures + seminars or labs:** 2p+2c
**ECTS credits:** 6
**FEL www:** AE4M39VG
**Anotation:** The goal of computational geometry is analysis and design of efficient algorithms for determining properties and relations of geometric entities. The lecture focuses on geometric search, point location, convex hull construction for sets of points in d-dimensional space, searching nearest neighbor points, computing intersection of polygonal areas, geometry of parallelograms. New directions in algorithmic design. Computational geometry is applied not only in geometric applications, but also in common database searching problems.

**Syllabus:**

1. Computational geometry (CG), typical applications, effective algorithm design techniques
2. Fundamentals of geometry, data structures for CG.
3. Geometric searching
4. Geometric searching 2
5. Planar convex hull
6. Convex hull in 3D
7. Proximity problems, Voronoi diagram.
8. Voronoi diagram applications.
9. 2D a 3D triangulations
10. Intersections of line segments
11. Intersections of polygonal areas and halfspaces
12. Geometry of parallelopids.
13. Dual algorithms.
14. New directions in algorithmic design

**Labs/Seminars:**

1. Introduction to the form of the seminars, Selection of topics for assignment.
2. Individual preparation of the presentations
3. Presentations of the topic assigned, discussion. Evaluation of the presentation materials and evaluation of the speech by classmate students. Ideas for improvements.
4. Presentation of the topic assigned,?
5. Presentation of the topic assigned,?
6. Presentation of the topic assigned,?
7. Presentation of the topic assigned,?
8. Presentation of the topic assigned,?
9. Presentation of the topic assigned,?
10. Presentation of the topic assigned,?
11. Presentation of the topic assigned,?
12. Presentation of the topic assigned,?
13. Presentation of the topic assigned,?
14. Assignment

**Literature:**

1. Preperata F.P.- M.I.Shamos: Computational Geometry An Introduction. Berlin, Springer-Verlag,1985.
2. Edelsbrunner H.: Algorithms in Combinatorial Geometry. Berlin, Springer - Verlag, 1987.
3. de Berg, M.,van Kreveld, M., Overmars, M., Schvarzkopf, O.: Computational Geometry, Berlin, Springer, 1997.
4. O' Rourke, Joseph: Computational Geometry in C, Cambridge University Press, 1st ed, 1994 or 2nd ed, 2000

**Notes from Faculty Information System:**

### 2.14 AE4M33AU: Automatic Reasoning

**Title:** Automatic Reasoning
**Lecturer:** Prof.RNDr. Štěpánková Olga CSc.
**Term:** summer
**Lectures + seminars or labs:** 2p+2c
**ECTS credits:** 6
**FEL www:** AE4M33AU
**Anotation:** Theorem proving is no more restricted to mathematics, but it is ever more often used in situations, when one needs to make sure that the suggested procedure meets the initial requirements it is used in deductive databases as well as for verification of SW or HW components. The process of proof construction has to be automated for that purpose. The course reviews current systems of 1st order theorem proving and their practical applications. There are explained underlying theoretical principles (model checking, resolution, tableaux) together with their practical and theoretical constraints. Special attention is devoted to gaining experience in choosing the best tool to solve a specific problem, in identification of mistakes in input or in strengthening the obtained results.

**Syllabus:**

1. History of automated reasoning in the context of artificial intelligence, review of its historical and up-to-date applications.
2. Problems in Boolean domains: from their specification and representation, to the formal solution. Properties of logical theorem proving - correctness and completness.
3. DPLL method, its existing implementations and their practical applications.
4. Model checking as a tool for verification, applications for finite automata.
5. Model checking - existing systems and how they are used in practice.
6. Automated theorem proving in general domains, language of 1st order logic.
7. Resolution and theorem provers based on this paradigma.
8. Workflow in resolution theorem provers and its basic steps: transformation to clausal form, ANL loop.
9. Non-resolution theorem provers: ”tableaux”, equality, transformation to DPLL.
10. Methods and systems for model construction in general domains.
11. Practical and theoretical boundaries for existing methods and systems.
12. Review of current automated theorem provers, their efficiency and practical utilization.
13. Algorithmic complexity of automated theorem provers and choice of the language for knowledge representation.
14. Spare slot.

**Labs/Seminars:**

1. Examples of typical problems for automated reasoning from various domains..
2. Specification of some simple problems in the formal language.
3. Transformation of the specification into the form requested by the theorem prover.
4. The tools for automated theorem proving - hands-on exercise.
5. Specification of more complex problems in the formal language and their automated solution I.
6. Specification of more complex problems in the formal language and their automated solution II.
7. Choice of the tool to be used in a specific problem.
8. Assignment of an individual project its goal is to implement a simple theorem prover applying specific clearly defined strategy or constraint on the used language in order to check impact of this.
9. Transformations of input and output for various theorem provers, interpretation of conclusions.
10. How to find an error in the problem statement.

11. Autonomous work on the assigned project the second part.
12. Simplification and strengthening of the obtained results.
13. Autonomous work on the assigned project the third part.
14. Credits, spare slot.

**Literature:**


1. Bundy, A.: The Computational Modelling of Mathematical Reasoning, Academic Press 1983 (Bundy). http://www.inf.ed.ac.uk/teaching/courses/ar/book/book-postcript/
2. Clarke, E.M. Jr., Grumberg, O. and Peled, D. A.: Model Checking, The MIT Press, 1999, Fourth Printing 2002. http://mitpress.mit.edu/catalog/item/default.asp?ttype=2andtid=3730
3. McCune, W.: Otter 3.3 Reference Manual (http://www-nix.mcs.anl.gov/AR/otter/otter33.pdf)
4. Newborn, M.: Automated Theorem Proving: Theory and Practice
5. Robinson, J.A., Voronkov, A. (Eds.): Handbook of Automated Reasoning (in 2 volumes). Elsevier and MIT Press 2001
6. Weidenbach, Ch.: SPASS: Combining Superposition, Sorts and Splitting (1999)
7. Wos, L. and Pieper, G.W.: A Fascinating Country in the World of Computing: Your Guide to Automated Reasoning


**Notes from Faculty Information System:**

## 3 Facultative courses

There will be many more courses. The following one is strongly recommended.

### 3.1 AE4B33RPZ: Pattern Recognition and Machine Learning

**Title:** Pattern Recognition and Machine Learning
**Lecturer:** Prof.Ing. Matas Jiří Ph.D.
**Term:** winter
**Lectures + seminars or labs:** 2p+2c
**ECTS credits:** 6
**FEL www:** AE4B33RPZ
**Anotation:** The basic formulations of the statistical decision problem are presented. The necessary knowledge about the (statistical) relationship between observations and classes of objects is acquired by learning on the raining set. The course covers both well-established and advanced classifier learning methods, as Perceptron, AdaBoost, Support Vector Machines, and Neural Nets.

**Syllabus:**

1. The pattern recognition problem. Overview of the Course. Basic notions.
2. The Bayesian decision-making problem, i.e. minimization of expected loss.
3. Non-bayesian decision problems.
4. Parameter estimation. The maximum likelihood method.
5. The nearest neighbour classifier.
6. Linear classifiers. Perceptron learning.
7. The Adaboost method.
8. Learning as a quadratic optimization problem. SVM classifiers.
9. Feed-forward neural nets. The backpropagation algorithm.
10. Decision trees.
11. Logistic regression.
12. The EM (Expectation Maximization) algorithm.
13. Sequential decision-making (Wald´s sequential test).
14. Recap.

**Labs/Seminars:**

1. Students solve four or five pattern recognition problems, for instance a simplified version of OCR (optical character recognition), face detection or spam detection using either classical methods or trained classifiers.
2. Introduction to MATLAB and the STPR toolbox, a simple recognition experiment
3. The Bayes recognition problem
4. Non-bayesian problems I: the Neyman-Pearson problem.
5. Non-bayesian problems II: The minimax problem.
6. Maximum likelihood estimates.
7. Non-parametric estimates, Parzen windows.
8. Linear classifiers, the perceptron algorithm
9. Adaboost
10. Support Vector Machines I
11. Support Vector Machines II
12. EM algoritmus I

13. EM algoritmus II
14. Submission of reports. Discussion of results.
15. Submission of reports. Discussion of results.

**Literature:**


1. Duda, Hart, Stork: Pattern Classification, 2001.
2. Bishop: Pattern Recognition and Machine Learning, 2006.
3. Schlesinger, Hlavac: Ten Lectures on Statistical and Structural Pattern Recognition, 2002.


**Notes from Faculty Information System:** http://cw.felk.cvut.cz/doku.php/courses/a4b33rpz/en_labs

# 4 Project and Master Thesis

## 4.1 AE4M99SVP: Software or Research Project

**Title:** Software or Research Project
**Lecturer:** TBD
**Term:** Z,L
**Lectures + seminars or labs:** TBD
**ECTS credits:** 6
**FEL www:** AE4M99SVP
**Anotation:** TBD

### 4.2 AE4M99DIP: Master Thesis

**Title:** Master Thesis
**Lecturer:** TBD
**Term:** summer
**Lectures + seminars or labs:** TBD
**ECTS credits:** 25
**FEL www:** AE4M99DIP
**Anotation:** TBD

# 5 Index

# Index