MASTER OF SCIENCE IN COMPUTER SCIENCE

Course title: ADVANCED GRAPHICS ALGORITHMS

Credits: 4

Course outline:

Geometrical description and modeling; Transforms; Color, Shading and Lighting; Texturing;

Rasterizing and Fragment processing; Blending and Transparency;

Lighting Techniques; Collision Detection; Realistic Scene; Natural Details

Literature

1. Tom McReynolds – David Blythe: Advanced Graphics Programing Using OpenGL, Elsevier, Morgan Kaufmann Publisher, 2005

2. Tomas Akenine-Möler, Eric Haines: Real Time Rendering, Second Edition A K Peters Wellesley, Massachusetts, 2002

3. Randima Fernando and Mark J. Kilgard: Cg he Cg Tutorial, The Definitive Guide to programmable Real-Time Graphics, Addison-Wesley, 2005

Course title: ADVANCED IMAGE PROCESSING

Credits: 4

Aim of the course:

The course is to provide mathematical foundations and practical techniques for some important topics in digital image processing and review some advanced techniques for some fundamental areas. Education Aims: to develop a foundation that can be used as the basis for further study and research in the basis for further study and research in image processing.

Course outline:

Color image processing. Advanced techniques for image segmentation. Wavelets. Morphological image processing. Binary image processing. Shape representation, skeletonization. Texture description. Mapping (morphing, warping). Image registration. Motion analysis and motion tracking. Pattern recognition

Literature

1. R.C. Gonzales, R.E. Woods: Digital Image Processing, 3rd edition, Prentice-Hall, Inc., 2008

2. M. Sonka, V. Hlavac, R. Boyle: Image Processing, Analysis, and Machine Vision, 3rd edition, Thomson Learning, 2007.

Course title: ADVANCED NUMERIC AND SYMBOLIC COMPUTING

Credits: 4

Aim of the course:

The course enables the students to deeply understand the numerical methods, involving their complexity, precision, and memory requirements. The graduated students will be capable to design such computational algorithms that can achieve preset precision requirements, while running on rounding error prone hardware and software components.

Course outline:

Orthogonalization algorithms, the Gram-Schmidt process, Householder transformations and Givens rotations. Numerical solution of the eigenvalue problem, Jacobi rotations, the QR algorithm, perturbation and separation theorems.

Approximations in linear spaces, generalized interpolation, Haar spaces, rational and spline interpolation, best approximations, orthogonal polinomials, Chebyshev approximations.

Solving systems of equations by iterative methods, fixed point theorems and fixed point iteration, relaxation methods, generalized Newton's methods.

Numerical integration, quadrature rules based on interpolating functions, Gaussian and Romberg quadrature, error estimates and convergence.

Differential equations, initial value problems, linear multistep methods for the numerical solution of IVPs, consistency, stability and convergence.

Boundary value problems for second-order ordinary differential equations.

Literature:

1. G. Dahlquist and A. Björck: Numerical Methods in Scientific Computing Volume I. SIAM, 2008
2. A. Neumaier: Introduction to Numerical Analysis. Cambridge University Press, 2001.

3. R. Plato, Concise numerical mathematics, AMS, 2003

4. J. Stoer and R. Bulirsch, Introduction to Numerical Analysis, Springer-Verlag, New York, 1992.

Course title: ADVANCED PROGRAMMING

Credits: 5

Aim of the course:

Make the students learn the generic programming paradigm and the practical usage of the C++ language and the Standard Template Library.

Course outline:

Object oriented programming in C++ (repetition). Classes - creating new types, fields, methods, overloading. Implementation hiding, namespaces. Reusage - composition, aggregation, inheritance. Overriding, polymorphism, late binding. Abstract and interface classes, multiple inheritance, virtual inheritance. Error handling with exceptions. Generic programming. Templates. Generic programming idioms (traits, policy, curiously recurring template pattern). Metaprogramming. Expression templates. The implementation and usage of the Standard Template Library (STL). STL foundations. Strings, data flow. Manipulators, effectors. Generic algorithms, predicates. Function objects, function object and pointer adapters. Iterators, ordering, searching, modifying. Generic containers and adapters

Literature

1. Bruce Eckel: Thinking in C++: Introduction to Standard C++, Volume One (2nd Edition), Prentice Hall; 2 edition (March 25, 2000), ISBN: 0139798099

2. Bruce Eckel: Thinking in C++, Volume 2: Practical Programming, Prentice Hall; 1 edition (December 27, 2003), ISBN: 0130353132

3. Matthew H. Austern: Generic Programming and the STL: Using and Extending the C++ Standard Template Library, Addison-Wesley Professional; 1 edition (October 23, 1999), ISBN: 0201309564

4. Bjarne Stroustrup: The C++ Programming Language: Special Edition, Addison-Wesley Professional; 3 edition (February 11, 2000), ISBN: 0201700735

Course title: ANALYSIS

Credits: 5

Course outline:

Measure, measure spaces, extension of a measure, outer measure. Measurable functions, integrable functions. The integral and its properties. Lebesgue measure, Lebesgue integral. Connection between Riemann- and Lebesgue-integrals. Product measures, Fubini's theorem, product stochastic measures. Function spaces, inequalities of Hölder and Minkowski, Riesz—Fischer theorem. Banach spaces, Hilbert spaces.

Functions with complex variable, Cauchy's integral theorem and Cauchy's integral formula. Analytic functions and their properties: representation by power series, set of zeros. Laurent series, classification of isolated singular points. Residue theorem and its application to evaluate contour integrals and real-valued integrals. Function series, Fourier series. Fourier transformation, Laplace transformation and their applications.

Literature:

1. A. N. Kolmogorov, S.V. Fomin, Elements of the Theory of Functions and Functional Analysis, Dover, 1999.

2. W. Rudin, Principles of Mathematical Analysis, McGraw-Hill, 1976.

3. B. Sz-Nagy, Introduction to real functions and orthogonal expansions, Oxford Univ. Press, 1965.

Course title: APPLICATIONS OF LINEAR PROGRAMMING

Credits: 4

Aim of the course:

The course enables the students to understand the functioning of linear programming algorithms, their application to such optimization programs that are built on the simplex method. Those who have completed this course will be capable to apply linear programming tools for general real life problems.

Course outline:

Duality, Integer programming. The assignment problem and the Hungarian method. Transportation problem. Hyperbolic programming. Convex programming. Gradient method.

Literature:

1. Chvátal, V., Linear Programming, Freeman, New York, 1983.

2. Dantzig, G. B., Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey, 1963.

3. Martos, B., Nonlinear Programming: Theory and Methods, American Elsevier, New York, 1975.

4. Salkin, H. M., K. Mathur, Foundations of Integer Programming, John Wiley and Sons, North-Holland, 1989.

Course title: AUTOMATA AND FORMAL LANGUAGES

Credits: 6

Aim of the course:

The course provides students with deeper understanding of the mathematical background of formal and programming languages, formal models in computer science, and foundations of computer science.

Course outline:

Characterization of regular langauges by congruences of finite index. Nerode and Myhill theorems. Minimization of automata. Syntactic monoid. Characterization of regular languages by monadic second-order formulas. Mealy automata and Moore automata. Finite automata mappings. Analysis and synthesis of Mealy automata. Generalized sequential machines. Chomsky and Greibach normal forms of context-free languages. Parikh's theorem and its consequences. The Chomsky-Schützenberger theorem. Composition of automata. Cascade product and the Krohn-Rhodes theorem.

Literature:

1. D. C. Kozen, Automata and Computability, Springer Publishing Company, 1997.

2. S. Eilenberg, Automata, Languages, and Machines, Vol. A. Academic Press, New York, 1974.

3. J. Berstel, Transductions and Context-Free Languages, B. G. Teubner, 1979.

4. J. E. Hopcroft, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison Wesley, Reading, 1979.

5. H. Straubing, Finite Automata, Formal Logic, and Circuit Complexity, Birkhauser, 1994.

6. Gécseg Ferenc: Products of automata, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1986.

Course title: COMPUTER VISION

Credits: 5

Aim of the course.

Students will learn the fundamentals of 3D reconstruction from a stereo image pair as well as the basics of motion analysis.. They will solve a project assignment in small teams, which develops their ability to work in a team, find creative solutions to real problems, analyse various aspects of their algorithms and to present their ideas in written and oral form.

Course outline:

Introduction - Human and Computer Vision. Vision models (Marr, Gestalt laws).
Camera geometry, parameters of 3D -> 2D projection.
Surface reconstruction from a single image 1.: shape from shading.
Surface reconstruction from a single image 2.: shape from texture.
Motion estimation, Optical Flow. Parametric motion models. Tracking. Video mosaicing.
Stereo Vision, epipolar geometry, Essential Matrix, Fundamental Matrix.
3D reconstruction from a pair of images.
3D reconstruction from multiple views.
Photometric stereo, motion-based reconstruction.
3D reconstruction and virtual view generation.

Literature

1. R. I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge University Press, 2004.

Course title: DATA MINING

Credits: 5

Aim of the course:

In recent years, data mining has been used widely in several areas of science and it is also important in business areas like customer relationship management, market basket analysis. Therefore the solid knowledge on data mining can be important for both the students learning towards PhD studies and students moving to industry.

Course outline**:**

Introduction to data mining (feature representation, main tasks),
Data visualization, Multidimensional scaling, Locally linear embedding,

Regression,

Sequence similarity measures, dynamic time warping,

Vector space model, Latent semantic indexing,

Clustering (sequential and hierarchical agglomerative methods),

Probabilistic and soft clustering.

Literature:

1. D. Hand, H. Mannila, P. Smyth, Principles of Data Mining, MIT Press, 2001 (546p).

2. P.N. Tan, M. Steinbach, V. Kumar: Introduction to Data Mining, Pearson Addison Wesley, 2006 (769p)

3. J. Han, M. Kamber: Data mining: Concepts and Techniques, 2nd ed., Morgan Kaufman, 2006 (800p)

Course title: DISTRIBUTED APPLICATION DEVELOPMENT

Credits: 5

Aim of the course:

The course is an introduction to elementary Windows programming in C#. By accomplishing the course students will be able to write basic Windows Forms applications utilizing the standard Windows graphical user interface. They are given examples on how to build applications in CLR (CLR is a development class library) and how to connect to databases via ODBC/OLE DB interface. The course includes a lecture on mixed programming language development (C++/VB/J#/C#), and a short overview about managed C++ CLR programming.

Course outline:

Main features of the C# programming language, the basics of the development environment. The language syntax, changes, differences. The memory usage paradigm of the .NET C#. The structure of C# programs.

How the .NET Framework does work? Access and usage of most important CLR library classes for the development of Windows Forms applications.

The most common Forms components and their properties. Customizing Forms, making them alive by processing different messages and responding to them. SDI, MDI and MultiSDI style user interface. How to write MDI (Multiple Document Interface) applications.

Introduction to resources. Writing international applications by the help of the CultureInfo and the ResourceManager helper classes.

Processing information stored in the Settings database. Binding component properties to user Settings.

Writing threaded applications, the timer component. Introduction to graphics. What is managed C++?

Writing database client applications using ADO.NET. Connecting to ODBC and OLE DB data sources like Oracle, MySQL, MS SQL Server.

Literature:

1. Chris Sells, Michael Weinhardt: Windows Forms 2.0 Programming (second edition), Addison-Wesley Professional, ISBN: 978-0321267962 (2006)

2. Christian Nagel, Bill Evjen, Jay Glynn et al: Professional C# 2005, ISBN: 07645-75341, Wrox Inc., (2005)

3. Solid Quality Learning: Microsoft SQL Server(TM) 2005: Database Essentials Step by Step, Microsoft Press, ISBN: 978-0735622074 (2006)

4. Jeff Ferguson, Brian Patterson, Jason Beres, Pierre Boutquin, Meeta Gupta: C# Bible, Wiley Publishing Inc, Indianapolis, USA, ISBN: 0-7645-4834-4 (2002)

Course title: EMBEDDED SYSTEMS

Credits: 5

Aim of the course:

To develop low-level, close-to-hardware programming competences in the students.

Course outline:

Introduction to embedded systems (SW/HW architectures). Cross-platform software development. Embedded operating systems (embedded Linux, VxWorks, software without operating system). Special tools for embedded software development (On-chip debugging support, JTAG, ICD, ICE). Hardware interfaces (UART, ICC, USB, network interface). Real-time systems, multitasking, inter-process communication, timing and memory handling. Debugging methodologies

Literature

1. Qing Li and Caroline Yao: Real-Time Concepts for Embedded Systems. CMP Books, 2003. ISBN: 1-57820-124-1

2. Robert Love: Linux Kernel Development. Addison-Wesley, 2010. ISBN: 0-672-32946-8

3. Karim Yaghmour: Building Embedded Linux Systems. O'Reilly, 2003. ISBN: 059600222X

Course title: GAME THEORY

Credits: 5

Aim of the course:

The course provides the necessary competences in game theory: the students will be enabled to classify the game theoretical models, learn basic theoretical background to basic algorithms and will be capable to recognize game theoretical problems, build models, and solve related practical problems.

Course outline:

Basic definitions, Neumann-Zermelo theorem. Gale-Stewart game. The connection between LP and matrix games. Minimax theorem. Domination and saddle points.

General sum games and their applications. Nash equilibria. Bimatrix games, Lemke-Howson algorithm. Subgame perfect and correlated equilibria.

n-person games, imputations, core and stable sets. Simple games. The LP characterization of the core. Shapley theorem and computation of Shapley value. Stable matchings and the kernel of directed games. Nash Bargaining Solution, Nash program.

Groups decision making, Arrow theorem. Choquet measure.

The elements of the Conway theory. Combinatorial games. Cop-Robber and Geography games. The Erdős-Selfridge theorem and its generalization.

Literature:

1. Vašek Chvátal: Linear Programming, Freeman, New York, 1983.

2. G. Fudenberg, J. Tirole: Game Theory, MIT Press, 1991.

3. M. J. Osborne, A. Rubinstein, A course in game theory, MIT Press, 1994.

4. Notes: www.inf.u-szeged.hu/~pluhar/oktatas/games.pdf

Course title: GRAPH THEORY

Credits: 5

Aim of the course:

Introducing the major graph theoretical notations and basic graph algorithms and algorithmic techniques. The students successfully completing the course must be able to understand graph theory problems, to solve efficiently the basic graph problems, distinguish tractable and intractable problems and be acquainted with the major graph theoretical techniques.

Course outline:

Basic notions and data structures related to graphs. Matrices and graph algorithms. Trees, connectivity. Higher connectivity of graphs, Menger theorem. Flows, Ford-Fulkerson algorithm, MFMC theorem. Coloring of graphs, coloring algorithms, coloring planar graphs, relation of the clique number and chromatic number, edge coloring of graphs, Vizing theorem. Matchings, Hungarian method, Edmonds algorithm, minimax theorems for matchings, randomized matching algorithm. Greedy algorithm for finding a big clique, Turan theorem, extremal graph theory. Ramsey theorems and its applications. Planar graphs, planarity testing, graph drawing problems.

Literature:

1. R. Diestel, Graph theory. Fourth edition edition. Graduate Texts in Mathematics, 173. Springer-Verlag, Berlin, 2010.
2. B. Bollobas, Modern graph theory. Graduate Texts in Mathematics, 184. Springer-Verlag, New York, 1998.
3. J.A.Bondy, U.S.R, Murty, Graph theory, Graduate Texts in Mathematics vol. 244. Springer, New York, 2008.

Course title: IMAGE PROCESSING SYSTEMS

Credits: 5

Aim of the course:

Knowledge on various image processing and visualization software libraries and systems.

Course outline:

Lecture and hands-on exercises with several image processing libraries and systems:

MATLAB + Image Processing Toolbox.

ImageJ: image processing routines, plugin and macro development.

VTK (Visualization Toolkit): graphical and visualization model, objects for image processing.

ITK (Insight Segmentation and Registration Toolkit): data visualization, access, and processing objects, filters, transformations.

Slicer: basic operations, loading and displaying image volumes, developing new modules

IPL; OpenCV; IDL (Interactive Data Language): IDL basics, graphical and image processing routines

Literature

1. R.C. Gonzalez, R.E. Woods, S.L. Eddins: Digital Image Processing Using MATLAB, Pearson Prentice Hall, 2003

2. ImageJ: http://rsbweb.nih.gov/ij/

3. VTK: http://www.vtk.org

4. ITK: http://www.itk.org

5. Slicer: http://www.slicer.org

6. IPL: Intel Image Processing Library, Reference Manual, Intel Corp. 2000.

7. OpenCV: http://opencv.willowgarage.com

8. IDL: http://www.ittvis.com/language/en-US/ProductsServices/IDL.aspx

Course title: LEGAL, ETHICAL AND INFORMATICS ISSUES OF PERSONAL DATA PROTECTION

Credits: 3

Aim of the course:

The course gives introductory legal knowledge about personal data processing. By accomplishing the course students will have sufficient background knowledge about European laws and conventions governing the Hungarian data protection legislation, and about the Hungarian law itself. The course discusses ethical questions of utilizing personal data for business and research purposes. The topics include works and achievements of top European data protection experts, and Hungarian data protection commissioners.

Course outline:

Historical overview. The immunity of home and private life and the confidentiality of personal correspondence and communication; Basic concepts concerning to the course, and international legal instruments controlling jurisdiction on personal data protection; Overview of most important international treaties, and Hungarian laws, decisions of the Constitutional Court; The recommendations of the Council of Europe and Working documents of the European Commission; Elements of the special legal rulings concerning to personal health data; Outstanding data protection commissioners in the EU member states, and their activities;
Ethics of database research in the mirror of international publications; Special biometric personal data: fingerprints, iris image, lip print, blood, DNA, proteomic profile; Questions of physical data protection, encryption. Crimes against electronic systems. The task of the DPOs (Data Protection Officers). Methods for anonymisation of personal data and the most important ethical questions of anonymisation; The transparent state and the opaque citizen, and the freedom of information based on László Majtényi's work. The Hungarian Data Protection Commissioners, annual reports, and their most important statements.

Literature:

1. Belgacem Raggad: Information Security Management – Concepts and Practice, CRC Press, Taylor & Francis Group, ISBN: 978-1-4200-7854-1, 2010

2. Eckstein, S.: Manual for Research Ethics Committees, King's College, ISBN: 0521810043, 2003

3. Matti Häyry, Ruth Chadwick, Vilhjálmur Árnason, Gardar Árnason: The Ethics and Governance of Human Genetic Databases, Cambridge University Press, ISBN: 978-0521-85662-1, 2007

4. Serge Gutwirth, Yves Poullet, Paul De Hert: Data Protection in a profiled World, Springer Verlag, ISBN: 978-90-481-8864-2, 2010.

5. Serge Gutwirth, Yves Poullet, Paul De Hert, Cécile de Terwangne, Sjaak Nouwt: Reinventing Data Protection?, Spriger Verlag, ISBN: 978-1-4020-9497-2, 2009.

Course title: MACHINE LEARNING ALGORITHMS

Credits: 6

Aim of the course.

During the course the students learn the foundations of machine learning. This gives the skill of using such techniques in developing algorithms for applied problems.

Course outline:

Introduction to machine learning,
Theory of machine learning (Vapnik-Chervonenski dimensions, PAC learning),

Perceptron and neural nets,

The regression task,

Generative and discriminative approaches,

Training by Maximum Likelihood.

Literature:

1. M.J. Kearns, U.V. Vazirani, An Introduction to Computational Learning Theory, MIT Press, Cambridge, Massachusetts, 1994 (221p)

2. T. Mitchell, Machine Learning, McGraw Hill, 1997. (414p)

3. R.O. Duda, P..E. Hart, D.G. Stork, Pattern Classification, Wiley and Sons, 2001.(p 654)

4. C. M. Bishop, Pattern recognition and machine learning, 2nd edition, Springer, 2007 (738p)

Course title: NONLINEAR OPTIMIZATION

Credits: 4

Aim of the course:

The competencies that will be provided by the course: ability to recognize the advantageous nonlinear optimization models, capability to select the proper solution algorithms, and to be able to interpret the obtained result in a professional way. The students will learn how to transform the original problem setting to obtain a more favorable form in terms of computational complexity and result precision.

Course outline:

Introduction, unconstrained optimization, convex sets and convex functions in optimization.

Optimization algorithms, direct search, gradient method, conjugate gradient method, Newton-method, quasi-Newton-methods.

Least Squares Method, Levenberg-Marquardt method, Gauss-Newton method, Convex programming.

Karush-Kuhn-Tucker optimization conditions, Penalty function method, Optimization with equation constraints, Lagrange multipliers.

Case studies.

Literature:

1. Bazaraa, M.S., H.D. Sherali, C.M. Shetty: Nonlinear Programming, Theory and Algorithms, Wiley, New York, 1993.

2. Gill, P.E., W. Murray, M.H. Wright: Practical Optimization, Academic Press, London, 1981.

3. E.M.T. Hendrix and B. G.-Tóth. Introduction to Nonlinear and Global Optimization. Springer, New York, 2010.

Course title: ON-LINE ALGORITHMS

Credits: 4

Aim of the course:

During the course the students learn the basic methods to handle the algorithmic problems where there is lack of information. This knowledge is useful in managing IT systems. Moreover the course also improves the skill of using mathematical methods to analyze algorithms.

Course outline:

Ski rental problem. Paging problem. Randomized algorithms, Yao principle. List update problem. Scheduling problems. Bin packing problems. k-server problem. Data acknowledgement. Routing. File caching. Generalizations of competitive analysis

Literature:

1. A. Borodin, R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, 1998. (414 p)

2. A. Fiat, G. J. Woeginger (eds), Online algorithms: The State of the Art, Vol. 1442 of Lecture Notes in Computer Science, Springer-Verlag Berlin, Heidelberg, 1998**.** (436p)

3. Cs. Imreh, Competitive analysis, in Algorithms of Informatics, (Volume I, eds Antal Iványi), 395-428

Course title: PARALLEL PROGRAMMING

Credits: 5

Aim of the course:

The main aim of the course is that the students get a firm understanding of the parallel programming paradigm. By studying the details of the Occam language and the components of the Java programming language that deal with parallelism and concurrency, the students get the opportunity to acquire a deep understanding of how parallel programs are constructed and what kind of data structures can be used in solving problems.

Course outline:

Parallel computers, parallel hardware systems. Parallel programming, parallel software systems. Processes, process interactions, interprocess communication, efficiency of parallel computations. Channels, messages, synchronous and asynchronous communication. The Occam language and the PVM system architecture. Multiplexers, pipelines, structure clash. Load balancing, processor farm. Parallelism in the Java language. Semaphores, mutual exclusion, synchronization. Produces-consumer, readers-writers problem. Monitors, condition variables, synchronization in monitors

Literature:

1. Course syllabus: /pub/Parhuzamos/ParallelProgramming.pdf
2. Wilkinson, Allen: Parallel Programming, Prentice Hall, 1999
3. Inmos Ltd: OCCAM Reference Manual, 1985
4. PVM: Users Guide and Tutorial, MIT Press, 1994

Course title: PROGRAM SYSTEMS DEVELOPMENT

Credits: 5

Aim of the course:

The goal of the course is to provide a basic understanding of the issues a system architect could face during the design and development of a robust and scalable software system. The course focuses on the JEE ecosystem, but it shows the issues solved on a more abstract level in order to enable the students the understanding of other possible ecosystems (.NET, PHP, etc).

Course outline:

Distributed systems (cloud vs. traditional enterprise applications, CAP). Crosscutting issues (security, transactions, etc). Middleware: the goal of the middleware in enterprise world, categories, services of the middleware. Implementing the business logic: 4^{th} and 5^{th} generation languages. Implementing the domain model: relational vs OO modeling. Ontologies. Handling the end user interaction (Ajax, JSF, SEAM). Defining the business logic (EJB)
Persistence: ORM. Service oriented architectures, ESB. Orchestration: BPEL. Crosscutting issues: security

Literature:

1. Wolfgang Emmerich: Engineering Distributed Objects, ISBN-10: 0471986577 | ISBN-13: 978-0471986577 | Publication Date: June 9, 2000

2. Martin L. Shooman: Reliability of computer systems and networks. ISBN-10: 0471293423 | ISBN-13: 978-0471293422 | Publication Date: December 15, 2001

Course title: SOFTWARE DEVELOPMENT

Credits: 5

Aim of the course:

Provide introductory information in basic Windows programming. By accomplishing the course students will be able to write applications utilizing the standard Windows graphical user interface in C and C++. Students are given examples on how to build applications in MFC (MFC is a C++ development class library) and how to connect to databases via ODBC interface. The course provides basic knowledge about X-Window graphical system as a comparison.

Course outline:

Windows Software development by command line tools; the usage of the make utility;

Development environments for applications running in graphical windowing operating systems: Windows API, Cygwin, X-Window, and OpenMotif;

Overview of X-Window system, presentation of several demonstrating software source codes;

X emulators in Windows, using the X font server;

Software development in Win32 SDK; Frequently used file types by their extensions: (.def, .rc, .res, .resx, .c, .cpp).

The Win32 resource files, their content, and structure;

How to write Win32 API programs, native window handling in C/C++;

Dialog windows, message processing.

Main features of the window components (menus, buttons, textboxes, listboxes, …) in Windows. How to handle dialog windows and their components.

Writing Windows applications in MFC. MDI and SDI software development in MFC; the document/view, architecture and serialization.

Using ODBC interface for connecting and querying different database systems like MS Access, dBase, Oracle, MySQL.

Literature:

1. Mastering MFC Development Using Microsoft C++, Microsoft Press, (book + CD-ROM edition), ISBN: 073560925X, (2000)

2. Richard M. Jones: Introduction to MFC Programming with Visual C++, Prentice Hall, PTR; (book + CD-ROM edition) ISBN: 0130166294 (1999)

3. Microsoft Visual C++ 6.0 Reference Library, Microsoft Press, fifth edition, ISBN: 1572318651, (1998)

4. Robert W. Scheifler, James Gettys et al: X Window System – The Complete Reference to XLib, X Protocol, ICCCM, XLFD, second edition, X Version 11, Release 4, ISBN: 1-55558-050-5, Digital Press, (1990)

Course title: SPEECH RECOGNITION AND NATURAL LANGUAGE PROCESSING

Credits: 5

Aim of the course:

The course shows the applications of artificial intelligence methods in the area of speech recognition and natural language processing. Besides teaching the most important results of the area it helps the students to learn to build artificial intelligence based solutions on applied problems.

Course outline.

The interdisciplinarity and subtasks of speech recognition. The hierarchy of linguistic sources of information. The current state-of-the-art and applications of speech recognition. The conventional source/filter model of human speech production. The very basics of human hearing. Speech recognition approaches and architectures. Preprocessing methods for feature extraction. Dynamic Time Warping. The mathematics of the hidden Markov model (HMM)
Application of HMM to speech recognition.
Recognition of continuous speech with HMMs. Methods of reducing the number of parameters. The main problems of natural language processing. Main statistical properties of natural languages.
Morphological modelling and analysis. Part-of-speech tagging. Estimating the quality and complexity of stochastic language models. N-gram language models. Probabilistic context-free grammars (P-CFG) and treebank-based language models. Approaches to semantic modeling. Dialogue models

Literature:

1. Huang, X., Acero, A., Hon, H-W.: Spoken Language Processing, Prentice Hall, 2001 (1008p)

2. Rabiner, L., Juang, B-H.: Fundamentals of Speech Recognition, Prentice Hall, 1993. (507p)

3. Young, S. et. al.: The Hidden Markov Model Toolkit (HTK) manual, Cambridge University, 2005 (210p)

Course title: TREE AUTOMATA

Credits: 5

Aim of the course:

Tree automata deal with tree structures, rather than the strings of more conventional state machines. Tree automata can be thought of, among others, as formal models for parsing of context free languages. The course gives an introduction to the most important concepts and constructions concerning tree automata.

Course outline:

Algebraic concepts, trees. Different types of tree automata. Regular tree gramamars. Operations on tree languages. Regular expressions, Kleene theorem for tree languages. Tree languages definable in monadic second-order logic. Büchi's theorem. Minimization of tree automata. Pumping lemma and decidability questions. Local tree languages. Tree automata and context-free languages. Weighted tree automata.

Literature:

1. F. Gécseg, M. Steinby, Tree Automata, Akadémiai kiadó, Budapest, 1984.

2. F. Gécseg, M. Steinby, Tree Languages, In. G. Rozenberg and A. Salomaa eds., Handbook of Formal Languages, Vol. 3, Chapter 1, pp. 1-68. Springer-Verlag, 1997.

3. H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi, Tree automata techniques and applications, http://www.grappa.univ-lille3.ft/tata, 1997.

4. Z. Fülöp, H. Vogler, Weighted Tree Automata and Tree Transducers, in: Handbook of Weighted Automata (Szerk.: M. Droste, W. Kuich és H. Vogler), Springer-Verlag, 2009, Chapter 9, 313-403