3D Photography

Rationale

This covers a topic of great research and commercial interest: the acquisition and processing of 3D models of real environments. There is a large volume of research work and industrial need in this field.

Course Description

Recent advances in computer hardware have made possible the efficient rendering of realistic 3D models in inexpensive PCs, something that was possible with high end visualization workstations only a few years ago. This class will cover the field of 3D Photography, the process of automatically creating 3D texture mapped models of objects, in detail. We will concentrate on the topics at the intersection of Computer Vision and Computer Graphics that are relevant to acquiring, creating, and representing 3D models of small objects or large urban areas. Many very interesting research questions need to be answered. For example: how do we acquire real shapes? how do we represent geometry? can we detect similarities between shapes? can we detect symmetries within shapes? how do we register 3D geometry with color images? etc. Applications that benefit by this technology include: historical preservation, urban planning, google-type maps, architecture, navigation, virtual reality, e-commerce, digital cinematography, computer games, just to name a few.

Topic List

Topics may include but are not limited to:

- 3D laser sensing and 2D image sensing.
- Alignment between 3D point sets (registration).
- Alignment between 3D point clouds and 2D images.
- 3D Segmentation.

- 3D Modeling (3D meshes and volumes).
- Mesh simplification and compression.
- Classification in 3D point clouds.
- Repeated patterns and symmetry detection.
- Texture Mapping.
- Online Classification for Large-Scale Datasets.

Learning Goals

- A general understanding of the importance of 3D Photography applications in various fields.
- Understaning of 3D geometry, rigid transformations, projective geometry, classification and segmentation algorithms.
- Ability to formulate research questions and to write research reports.
- Ability to present technical talks.
- Understanding of selected Computer Vision & Graphics algorithms.
- Skills to apply programming tools for solving 3D photography problems.
- Understanding the various sensor and acquisition technologies.
- Ability to provide solutions in problems involving large-scale 3D and 2D datasets.

Assessment

The grade will be based upon the following: 50% for group or individual projects, 30% for presentation(s) and 20% for class participation. The course does not require any exams. Each student will complete two introductory homeworks (one theoretical and one programming). Each student will prepare a research report that surveys a specific area of 3D Photography and solve a final project. The report will also be supported by a student presentation in class. Grading will be based on the attendance, student presentation, homework completion and the final research report and project. Students can work in groups if they desire so for the final project, upon the consent of the instructor. A list of possible topics that would be appropriate for the final project and report can be provided. Students can pick a topic from this list or can also work on any 3D Photography related topic approved by the instructor.

Advanced Data Structures

Rationale

Efficient computing involves the use and maintenance of advanced data structures in a wide variety of algorithms used in Data Sciences. This course covers the theory and algorithms for these advanced data structures.

Course Description

Data structures are a building block for algorithms. A data structure models some abstract structure with a specified set of operations. These include

- maintaining a set under insertions, deletions, and find operations, or
- maintaining a linear order under insertions, deletions, and comparisons, or
- maintaining a graph under insertion of edges and queries whether two points are in the same connected component, or
- answering orthogonal range queries for a point set, or
- answering substring queries for a string

In each of these situations, we have a well-specified behavior of what the structure should do, but it is not immediately clear how we should achieve that behavior. This is an algorithmic question: to design and analyze algorithms that realize the required operations, and answer questions like

How fast can we perform the operations? Can we do better if we allow amortized complexity instead of worst-case complexity? How much space does the structure need? Does the structure support changes, or only queries? How can we access past versions of the structure?

Once we know good methods to realize these structures, they are available as components to higher-level algorithms, like the heap in Dijkstra's shortest-path algorithm, or the set-union strukture in Kruskal's minimum spanning tree algorithm; whenever an algorithm performs such operations often, one should look for the most efficient realization of that operation.

In this course we will study many data structures, their analysis and implementation.

Topic List

The topic list may include but is not limited to:

- Search and update sets of numbers, intervals, or strings by various data structures such as
  - Search trees
  - Structures for sets of intervals
  - Structures for piece-wise constant functions
  - Structures for orthogonal range searches
  - Heaps
  - Union-find structures
- Dynamization and persistence of structures
- Structures for strings
- Structures for hash tables.

Learning Goals

Be able to determine computational complexity of and program algorithms for

- set maintenance: insertions, deletions, and find operations
- maintaining linear under under insertions, deletions and comparisons
- inserting or deleting edges in a graph
- determining whether two graph nodes are in the same connected component
- determining whether one string is a substring of another string

Assessment

There will be four implementation homeworks of structures; test code will be provided, and homework submissions are not accepted until they pass this test. Programs are to be written in C or C++. and will be tested in a linux environment using gcc/g++.

Algorithms For Big Data Analysis

Rationale

Traditional analysis of algorithms generally assumes full storage of data and considers running times polynomial in input size to be efficient. Operating on massive-scale data sets such as those of tech companies such as Google, Facebook, etc., or on indefinitely large data streams, such as those generated by sensor networks and security applications, leads to fundamentally different algorithmic models. MapReduce/Hadoop in particular has seen widespread adoption in industry.

Course Description

This course addresses algorithmic problems in a world of big data, i.e., problems in settings where the algorithm's input [the data] is too large to fit within a single computer's memory. Traditional analysis of algorithms generally assumes full storage of data and considers running times polynomial in input size to be efficient. Operating on massive-scale data sets such as those of tech companies such as Google, Facebook, etc., or on indefinitely large data streams, such as those generated by sensor networks and security applications, leads to fundamentally different algorithmic models. In previous decades, DBMS settings where the data sets on a machine's disk but not in memory motivated the external memory or I/O model (e.g. ex- ternal mergesort and B-trees). More recently, models such as MapReduce/Hadoop have appeared for computing on data distributed across many machines (e.g. PageRank computation or matrix multiplication).

Topic List

The topics may include but are not limited to:

- MapReduce/Hadoop
- Streaming and Sketching
- Finding Matchings

- Counting Triangles
- Deciding Bipartiteness and Connectivity
- The Secretary problem
- Matrix Multiplication
- Compressed Sensing

Learning goals

The student will learn what the modern models for massive-scale data algorithms are, how to analyze algorithms in these models, and when to use them. In particular, the student will gain experience with MapReduce/Hadoop and with a number of streaming/sketching algorithms.

Assessment

Several bi-weekly problem sets and a course project. The project can take the form of either an expository paper, a nontrivial implementation project, or performing original research on a problem.

Artificial Intelligence

Course Rationale

Artificial intelligence (AI) develops programmed agents (systems) that match or outperform people's abilities to make decisions, to learn, and to plan. To do so, AI develops algorithms and methodologies that sense a system's environment, decide what to do given that data, and effect its chosen actions in its environment.

Course Description

This is a graduate-level course on artificial intelligence. It emphasizes fast and clever search heuristics, thoughtful ways to represent knowledge, and incisive techniques that support rational decision making. Application areas will include game playing, natural language processing, and robotics.

Students are expected to have a solid background in the analysis of algorithms, proofs in propositional and first-order logic, discrete mathematics, and elementary probability.

Topic List

Topics include but are not limited to:

- Introduction: foundation definitions, classic AI problems, and their solutions, knowledge representation
- Agents and Problems
- State-space search: uninformed search, informed search, local search
- Constraint satisfaction: principles and practices
- Logics for Agents: probabilistic and logical reasoning
- Knowledge Representation
- Constraint satisfaction: principles and practices

- Machine learning: foundation definitions, computational learning theory, major paradigms
- Planning
- Markov Models
- Introduction to more advanced topics (e.g., embodied cognition, cognitive architectures, autonomy)

Learning Objectives

Students who successfully complete this course will be able to:

- Discuss the agent paradigm as the goal of an intelligent machine
- Describe state space search as a mechanism for problem solving, including optimal solutions and their complexity
- Explain the role of caching, reactivity, heuristics, and planning in state space search.
- Define machine learning and describe the specifics of several prominent machine-learning methods (e.g., SVMs, decision trees, Bayes nets, artificial neural networks, genetic algorithms)
- Evaluate the complexity of an approach to a specific problem and its realistic impact.
- Describe and illustrate the role of constraint satisfaction in AI, with appropriate examples.
- Discuss the role of probabilistic reasoning and mechanisms that employ it
- Discuss the role of logical reasoning and mechanisms that support it

The course has a general goal providing a capability for Empirical AI research that addresses a real-world problem with appropriate knowledge representations and a reasoning methodology for it, identifies or constructs algorithms to address it, and implements, tests, and evaluates alternative solution(s) to it.

Assessment

- Class participation 10% will asses both applied and theoretical topics
- Assignments 60% will assess the theoretical topics
- Term project (presentation and report) 30% will assess the applied topics

Big Data Analytics

Rationale

In addition to constantly growing volumes of proprietary transaction, product, inventory, customer, competitor, and industry data collected from enterprise systems, organizations are also faced with overwhelming amounts data from the Web, social media, mobile sources, and sensor networks that do not fit into traditional databases in terms of volume, velocity and variety (the three Vs of Big Data). This Big Data flood poses challenges as well as opportunities, if managed and analyzed properly, to derive new actionable knowledge and intelligence in a timely manner. This course will explore existing and emerging methods to manage, integrate, analyze and visualize domain-specific Big Data, to identify and provide domain specific solutions.

Description

This course covers the research issues and practical methods of managing and analyzing Big Data to gain and discover insights, patterns, and knowledge nuggets that can support decision makers.

Topic List

Topics may include but are not limited to:

- Introduction
  - Environment, Challenges, and Opportunities
  - Analytics Platform: architecture, process, and analytic tools
  - Multiple data source management and data integration
- Structured Data Analytics
  - Structured Big Data: Issues and Approaches
  - Transportation Data Analytics
  - Financial, Banking, Web-based Transaction Data Analytics

- Semi/Unstructured Data Analytics
  - Textual Data Analytics
  - Social Media Data Analytics
  - Short Text Classification/Clustering
  - Real-time Big Data Processing
- Media Data Analytics
  - Fundamentals of Image/Video Data Analytics
  - Cultural Analytics and Visualization
  - Statistical Inference and Real-time Classification
- Network and Graph Data Analytics
  - Social Network/Graph Data Analytics
  - Semantic Web and Linked Data Analytics
- Societal Impacts on Big Data Analytics
  - Security and Privance Issues
  - Accountability Issues: Open Government Data

Learning Goals

- To expose students to Big Data as a scientific or engineering problem. Students will be guided to focus on a particular domain specific area, identify research challenges or application utilities, and present existing and/or innovative methods and algorithms to design a solution. Students are expected to submit a conference paper and/or a demonstration paper to a conference related to Big Data Analytics by the end of this seminar, in collaboration with the faculty member(s). A series of student presentations are expected at the end of the semester.

Assessment

Each student will present a critical review, summarizing the problems and solutions given in a selected research paper. Students will select one application domain area and collect a repository of data sets. Collectively, the data sets will serve a wider community for Big Data Analytics experiments and tests.

Students will identify a Big Data Analytics research problem related to their domain application and dataset, and write a research paper discussing the existing solutions and design/propose a potential new applied solution that can be used by the domain area decision makers.

With the given dataset, each student can analyze and design a specific use-case related to her/his research problem, and design (and possibly implement) his/her proposed solution as a tool.

The final presentation of the research paper and a demo will be given in the form of a workshop/poster presentation at the end of the semester with an audience of invited faculty, students and industry leaders. Top paper awards will be given and students will have the chance to work with a faculty or industry mentor on a conference paper and/or journal publication.

Big Spatial Data

Rationale

Recent advances in computer hardware have made possible the efficient rendering of realistic 3D models in inexpensive PCs, something that was possible with high end visualization workstations only a few years ago. This class will cover the field of 3D Photography, the process of automatically creating 3D texture mapped models of objects, in detail. We will concentrate on the topics at the intersection of Computer Vision and Computer Graphics that are relevant to acquiring, creating, and representing 3D models of small objects or large urban areas. Many very interesting research questions need to be answered. For example: how do we acquire real shapes? how do we represent geometry? can we detect similarities between shapes? can we detect symmetries within shapes? how do we register 3D geometry with color images? etc. Applications that benefit by this technology include: historical preservation, urban planning, google-type maps, architecture, navigation, virtual reality, e-commerce, digital cinematography, computer games, just to name a few.

All of the above issues must be solved in a parallel processing environment. For example, the Nvidia GTX Titan GPUs with 2,688 cores that support 15*2048 concurrent threads, 6 GB memory and 1.3 and 4.5 Teraflops computing power (double and single precision, respectively) currently available from the market for around $1,000.

Course Description

The increasingly larger data volumes and more complex semantics of spatial information never cease to request more computing power to turn such data and information into knowledge to facilitate decision making in many applications, ranging from location based services to intelligent transportation systems. Current generation of spatial databases and moving object database technologies based on aged hardware architectures is incapable of processing data with reasonable effort and there are Spatial Big- Data (SBD) challenges. In particular, although locating and navigation devices (e.g. GPS, cellular/wifi network-based and their combinations) embedded in smartphones (nearly 500 million sold in 2011) have already generated large volumes of location and trajectory data, the next generation of consumer electronics, such as Google Glasses, are likely to generate even larger volumes of location-dependent multimedia data where spatial and trajectory data management techniques will play critical roles in understanding the data. Graphics Processing Units (GPUs) are massively data parallel devices featuring a much larger number of processing cores and concurrent threads which make them significantly different from CPUs that currently support much fewer processing cores and concurrent threads. In addition, the current GPU memory bandwidth is more than an order of magnitude higher than that of CPUs and three orders higher than that of disks. Different from high-performance computing resources in the past that are typically only available to highly selective research groups, GPUs nowadays are quite affordable to virtually all research groups and individuals.

On the other hand, the Intel Xeon Phi accelerators based on its Many-Integrated-Core (MIC) architecture represent a hybridization of classic multi-core CPUs and GPUs and are suitable for speeding up a variety of applications.

1 Topic List

- Commodity Parallel Hardware
- Research Practices of Large-Scale Data Management
- Relational and Non-Relational Data
- OpenMP
- Nvidia CUDA
- Intel TBB based parallel programming techniques
- Parallel Indexing and Query Processing on Multidimensional Spatial and Trajectory Data
  - Grid- and tree-based indexing
  - Selectivity estimation
  - Various types of spatial joins and their optimization

- Identifying inherent parallelisms in processing large-scale multi-dimensional data
- High-level parallel primitives
- Multi-core CPUs, GPUs, and Intel MICs

2 Learning Goals

- Be able to understand the variety of kinds of parallel processing
- Be able to understand the different kinds of programming for parallel processing
- Understand how to identify parallelisms in a large multi-dimensional data set
- Be able to solve spatial data problems in a parallel processing environment

3 Assessment

Grading will be based on the attendance, student presentation, homework completion and the final research report and project. Students can work in groups if they desire so for the final project, upon the consent of the instructor. A list of possible topics that would be appropriate for the final project and report can be provided. Students can pick a topic from this list or can also work on any 3D Photography related topic approved by the instructor.

- 50% for group or individual projects
- 30% for presentation(s)
- 20% for class participation

Combinatorial Algorithms

Rationale

Combinatorial algorithms is a core part of algorithms, which is a core part of computer science, Many of the optimization problems that are most fundamental to computer science and have had the greatest "broader impact" outside of computer science and indeed within the wider world – shortest paths for travel, network flow for business and transportation, maximum matching for resource allocation, linear programming for myriad operations research problems – are among the topics covered in this course.

Course Description

This is a course on combinatorial algorithms covering topics (far) beyond the scope of the first-year algorithms class. More precisely, this is an advanced course in algorithms for optimization problems concerning discrete objects, principally graphs. In such problems, we search a finite but typically exponentially large set of valid solutions—e.g., all matchings in a graph for maximizing or minimizing some objective function. Nonetheless, most of the problems we study in this course are optimally solvable in polynomial time. The fundamental topics here are matchings, flows and cuts, shortest paths and spanning trees, and matroids. An overarching theme is that many such problems have traditionally been studied both a) by computer scientists, using discrete, combinatorial algorithms (greedy, dynamic programming, etc.), and b) in the operations research optimization community, where they are treated as continuous optimization problems (solved by linear programming, etc.). We will often compare the two approaches, and we will find that it can be fruitful to combine them. In particular, we will repeatedly use linear programming throughout the course.

Topic List

Topics can include but are not limited to:

- shortest paths and spanning trees

    - single-source (Dijkstra, Bellman-Ford)
    - all-pairs (Floyd-Warshall)
    - spanning trees (Prim, Kruskal)
    - arborescences (Edmonds)
    - NP-Complete extensions (CDS, Steiner, etc.)

- linear programming
    - convex hulls and polyhedra
    - simplex,
    - LP-rounding for approximation
    - duality
    - packing and covering LPs
    - primal-dual method
    - totally unimodular matrices

- convex optimization
    - convex programming
    - Lagrangian duality
    - semi-definite programming

- matchings
    - unweighted bipartite (augmenting paths)
    - weighted bipartite (Hungarian)
    - unweighted non-bipartite (Edmonds)
    - weighted non-bipartite (more Edmonds)
    - NP-Complete extensions (GAP, 3DM, etc.)

- network flows
    - disjoint paths (Menger)
    - max flow / min cut

- augmenting path algorithms
- push-relabel algorithms
- min-cost max flow
- multi-commodity flow and multicut

Learning goals

Learn some of the canonical algorithms (and algorithm schemata) for solving fundamental matching, flow, and path problems; become able to apply and extend these techniques to new problem variations; come to see how many of these problems are mutually reducible to one another; gain an appreciation of the conceptual foundations of duality and matroid theory, and of polyhedral combinatorics as mathematical technology.

Assessment

The theoretical concepts and proofs will be assessed by bi-weekly homeworks. The algorithms will be assessed by a final project.

Computer Vision and Image Processing

Rationale

Computer vision and image processing are important and fast evolving areas of computer science, and have been applied in many disciplines. This course will introduce students to the fascinating fields. Student will gain familiarity with both established and emergent methods, algorithms and architectures. This course will enable students to apply computer vision and image processing techniques to solve various real-world problems, and develop skills for research in the fields.

Course Description

This course introduces fundamental concepts and techniques for image processing and computer vision. Topics to be covered include image formation, image filtering, edge detection and segmentation, morphological processing, registration, object recognition, object detection and tracking, 3D vision, and etc.

Topic List

The topics may include but are not limited to:

- Image formation and perception, image representation
- Image filtering: space- and frequency- domain filtering, linear and non-linear filters
- Morphological image processing.
- Image geometric transformations, image registration.
- Edge detection, image segmentation, active contours, level set methods
- Object recognition, template matching, classification

- Object detection and tracking: background modeling, kernel-based tracking, pChanghe filters
- Camera models, stereo vision

Learning Goals

The learning goals include:

- Understand the major concepts and techniques in computer vision and image processing
- Demonstrate computer vision and image processing knowledge by designing and implementing algorithms to solve practical problems
- Understand current research in the fields
- Prepare for research in computer vision and image processing

Assessment

The course assessments include homework and programming assignments (30%), one middle term exam (30%), and a final project (40%). The programming projects require students to implement algorithms to solve real computer vision problems to demonstrate their understanding on concepts of computer vision and image processing. A final project on a research topic requires a proposal, a final project report, and a presentation to prepare students for research experience.

Data Mining

Rationale

Datasets consist of observations sampled from a population. They can be as large as terabytes with many variables and many records. The population may consist of subpopulations with each subpopulation having different sets of dependencies among the variables. Data Mining has tools and techniques to identify the structure that enable making valid predictions.

Course Description:

Data mining is the name given to a variety of new analytical and statistical techniques that are already widely used in business, and are starting to spread into social science research. Other closely-related terms are machine learning 'pattern recognition' and predictive analytics. Data mining methods can be applied to visual and to textual data, but the focus of this class is on the application of data mining to symbolic or numerical data. In this area, data mining offers interesting alternatives to conventional statistical modeling methods such as regression and its offshoots.

Each student will undertake a data mining analysis project as a final paper, typically analyzing a dataset chosen by the student.

Topic List

The topic list may include but is not limited to:

- Exploratory Data Analysis
- Association Rules
- Distance and Similarity Measures
- Clustering
  - K-means
  - Hierarchical Clustering: Agglomerative and Divisive

- Subspace Clustering
  - Linear Manifold Clustering
  - Graph Theoretic Clustering
  - Spectral Clustering
  - Mixture Models
  - Biclustering
  - Density-based Clustering

- Prediction and Classification with K-Nearest Neighbors
- Discriminant Analysis
- Classification and Regression Trees
- Random Forests
- Logistic Regression
- Validation Techniques
  - Training and Test Sets
  - Permutation Tests
  - Bootstrap Resampling

Learning Goals

- Understand the mathematical and statistics foundations of the methodology and algorithms of data mining techniques
- Become proficient with data mining software such as WEKA and R
- Given a dataset, be able to discover patterns and relationships in the data that may be used for descriptive modeling or to make valid predictions

Assessment

Assessment of understanding the mathematical and statistic foundations will be done through a midterm (40%) and homeworks (20%). Assessment of proficiency in using data mining software and discovery of patterns and relationships in a data set will be done by a project (40%).

Data Visualization

Course Rationale

Today quantitative and symbolic data are easily collected in computer format, from databases, websites, smartdevices, and anything that has interconnect capabilities. When such large amounts of data are put in spreadsheets or tabular reports, it becomes difficult to see the patterns, structure, trends, or relationships inherent in the data. Effective data visualization exposes these inherent relationships, consolidating and illustrating them in graphics.

Course Description

A visualization organizes data in a way that the structure and relationships in the data that may not be so easily understood becomes easily understood and interpreted with the visualization. Visualizations of a data set give the reader a narrative that tells the story of the data.

The purpose of data visualization is to convey information contained in data to clearly and efficiently communicate an accurate picture of what the data says through understandable and context appropriate visualizations.

To do a visualization can be just exploratory or entails using Machine Learning techniques that determine the structure of the data. The visualizations are then matched to the data structure.

The course will explore how principles of information graphics and design and how principles of visual perception, can be used with machine learning techniques to make effective data visualizations.

Each student will make a presentation of some principles of data visualizations or do a visualization project.

The course is open to PhD students in all programs. Non-computer science students will be paired with computer science students for the visualization project.

Topic List

The topic list may include but is not limited to:

- Visualization Techniques
  - Pie and Donut Charts
  - Histograms
  - Scatter Plots
  - Heat Maps
  - Matrix Diagrams
  - Candlestick Charts
  - Bubble Charts
  - Graphs and Networks
  - Alluvial Diagrams
  - Dendrograms
  - Ring Charts
  - Tree Diagrams
  - Treemaps
  - Polar Area Diagram
  - Parallel Coordinate Displays
  - Time Series
  - Line Charts
  - Cartograms and Choropleths
  - Dot Distribution Maps
- Visualization Issues
- Visualization Tools
  - Profuse
  - Protoviz
  - R

Learning Goals

- Be able to describe the key design guidelines and techniques used for the visual display of information
- Understand how to best use the capabilities of visual perception in a graphic display
- Understand the principles of interactive visualizations
- Understand how Machine Learning techniques can determine data structure and pattern
- Explore and critically evaluate a wide range of visualization techniques and applications

Assessment

Every student will do a project involving a presentation of the project at the end of the course.

Database Management Systems

Rationale

Database Management Systems (DBMS) are vital components of modern information systems. Database applications are pervasive and range in size from small in-memory databases to terra bytes or even larger in various applications domains. The course focuses on the fundamentals of knowledgebase and relational database management systems, and the current developments in database theory and their practice.

Course Description

The course reviews topics such as conceptual data modelling, relational data model, relational query languages, relational database design and transaction processing and current technologies such as semantic web, parallel and noSQL databases. It exposes the student to the fundamental concepts and techniques in database use and development as well provides a foundation for research in databases.

The course assumes prior exposure to databases, specifically to the relational data model and it builds new technologies on this foundation. In the first half of the course the relational data model, relational query languages, relational database design and conceptual data modeling are reviewed. It then focuses on XML, RD, OWL, parallel, and noSQL databases. It also bridges databases and knowledgebases which is the current trend.

The course requires a term project in which the student implements a database application or explores a database issue.

We will use PostgresSQL as the database platform for doing the assignments.

Topic List

Topics can include but are not limited to:

- Database concepts
- Relational model

- Relational query languages
  - Relational Algebra and Calculus
  - Datalog
  - SQL
  - QBE
- Triggers
- Embedded SQL
- Recursion
- Web database programming
- Conceptual data modeling
- E/R data model
- OO data model
- Relational database design
  - Normal Forms (NF)
  - 1-4NF
  - Lossless join decomposition
- XML, XPath and XQuery
- Ontology and Data Model
- Semantic Web
- RDF, RDF Schema, and OWL
- Storage and indexing
- Query processing and optimization
- Parallel and distributed databases
- NoSQL databases

- Transaction processing and database recovery
- Database security
- Current developments in knowledgebase
- Big data and Hadoop.

Learning Goals

The course content is balanced on theory and practice, the course aims at achieving the following learning outcomes:

- An appreciation of pervasive use of Knowledgebase and DBMS in different application domains
- Skill for developing database applications
- Skills for devising data models and query languages
- Skills for developing web database applications
- Learning storage and indexing of data
- Learning transaction processing and database recovery
- Learning knowledge representation and semantic web technologies
- Skills to integrate knowledge to databases

Assessment

Class participations, discussions and attendance are a critical component of the course and accounts 10

Written assignments will provide the students the opportunity to appreciate the theoretical underpinnings of the databases systems and comprise 20% of the term grade. These are on data modeling (Entity/Relationship data model), query languages (Relational Algebra and Relational Calculus), database design, and Resource Definition Framework or Ontology Web Language for knowledge representation.

Two programming assignments and a term project will provide the opportunity for the students to develop technical skills and comprise 20% of the term grade. The first project is on Structured Query Language and the second one is on the web database programming. The database project involves a complete database application development from design to implementation or a deeper investigation of a topic in databases and knowledgebases.

A mid-term and a final exam, each is 25%. Individual Assignments or Quizzes 20% Term Projects 15% Midterm Exams 30% Final Exam 35%

Graph And Social Network Analysis

Rationale

A graph has nodes and edges which connect some pairs of nodes. The edges can be directed or undirected. Graph theory has broad application to areas of physics, chemistry, communication science, biology, electrical engineering, operations research, psychology, linguistics, and social networks.

4 Course Description

The course first studies fundamental concepts in graph theory including data structures that can represent graphs. Concepts include flows and connectivity (e.g., Mengers theorem), planarity (coloring), Eulerian and Hamiltonian graphs. Then the course studies fundamental concepts, metrics, and algorithms associated with Social Networks.

Topic Lists

- Fundamentals
  - Graphs and subgraphs
  - Connected graphs
  - Trees
- Flows and Connectivity
  - Non-separable graphs
  - Flows in networks
  - Edge and Vertex Connectivity
- Graph Representation and Algorithms.
  - Adjacency matrix and adjacency-linked lists
  - Dijkstra's Shortest Path Algorithm

- Planarity and the Four-Colour
- Independent Sets
- Cliques and Quasi Cliques
- Matching
- Eulerian and Hamiltonian Cycles
- Vertex and Edge Covers
- Dominating Sets
- Random Network Models
- Social Network Analysis
  - Types of Social Networks
  - Homophily
  - Multiplexity
  - Mutuality/Reciprocity
  - Propinquity
  - Bridges
  - Degree Centrality
  - Betweenness Centrality
  - Closeness Centrality
  - Network Reach
  - Network Integration
  - Boundary Spanners
  - Peripheral Players
  - Density
  - Structural Holes
  - Tie Strength
  - Communities

Learning Objectives

Students will be able to

- Formally apply graph-theoretic terminology and notation
- Apply theoretical knowledge acquired to solve practical graph problems
- Understand and apply social network analysis techniques

Assessment

There will be two exams to assess understanding of the theoretical concepts of Graph Theory: a Midterm (30%) and a Final (30%). There will be Homework and Programming Projects (30%) to assess knowledge of algorithms.

Graphical Models

Rationale

Probabilistic graphical models have been applied to various domains for modeling and reasoning uncertain information. This course will provide students with essential backgrounds on these methods.

Course Description

Probabilistic graphical models, especially Bayesian networks, offer a compact, intuitive, and efficient graphical representation of uncertain relationships among the variables in a domain and have proven their value in many disciplines, including machine or medical diagnosis, prognosis, bioinformatics, planning, user modeling, natural language processing, vision, robotics, data mining, fraud detection, and many others. This course will familiarize you with the basics of graphical models and provide a foundation for applying graphical models to complex problems. Topics include basic representations, exact inference, approximate inference, parameter learning, structure learning, and applications.

Topic List

The topic list may include but is not limited to: Bayesian network

Examples (HMM, diagnostic system, etc.) Separation and independence Markov properties and minimalism

Markov network

Examples (Boltzmann machine, Markov random field, etc.) Cliques and potentials Markov properties

Exact inference

Complexity Bucket elimination Junction tree Belief propagation (message passing) Application to HMM Sum- and Max-product algorithms

Parameter learning

Exponential family Bayesian learning Expectation-Maximization (EM)

- Bayesian networks

- Examples (HMM, diagnostic system, etc.)
- Separation and independence
- Markov properties and minimalism

• Markov networks

• Undirected Graphical Models
  - Decomposable Graphs
  - Separation and Conditional Independence
  - Junction Trees
  - Examples (Boltzmann machine, Markov random field, etc.)
  - Cliques and potentials
  - Markov properties

• Inference
  - Complexity
  - Bucket elimination
  - Belief propagation (message passing)
  - Application to HMM
  - Sum- and Max-product algorithms

• Parameter learning
  - Exponential family
  - Bayesian learning
  - Expectation-Maximization (EM)

• Model Estimation

Learning Goals

The Learning Goals are to acquaint students with the major topics of probabilistic graphical models in order for them to gain an appreciation of the techniques that are available and the problems that are yet to be solved. Students will be able to

- read and understand technical Changhes describing work in the field of uncertainty artificial intelligence.
- apply knowledge of probability theory to understand the principles behind graphical models and uncertainty reasoning.
- evaluate the applicability of different probabilistic inference methods and determine which is most likely to be most applicable and effective to a specific problem.
- understand different Bayesian methods for learning graphical models from data and conduct experiments to assess their performance.
- identify, formulate, and solve a real-world problem using uncertainty artificial intelligence techniques by collaborating in an interdisciplinary team.

Assessment

There will be a midterm exam (40to assess understanding of the probability theory underlying graphical models and Bayesian Networks. There will be a term project to assess ability to identify, formulate and solve real-world problems using graphical models.

Machine Learning In Quantitative Finance

Rationale

Because of their greater power than classical statistical methodologies, machine learning techniques for classifying and evaluating risk, have an important place in the world of quantitative finance. This course combines relevant material from quantitative finance and sets up its interface to machine learning methodologies.

Course Description

This course studies the application of machine learning techniques in various quantitative finance problems. Contents include basics of financial instruments, basics of quantitative trading, machine learning and their applications in quantitative finance.

Topic List

Topics may include but are not limited to:

- Financial Instruments
  - Equities
  - Bonds
  - Futures
  - Options
  - Derivatives
- Trading Algorithms
  - TWAP
  - VWAP

  - Percent of Volume
  - Minimal Impact
  - Implementation Shortfall
  - Adaptive Shortfall
  - Market On Close
  - Pairs Trading Algorithms
- Machine Learning
  - Time Series Prediction
  - Neural Networks
  - Genetic Algorithms
  - Reinforcement Learning

Learning Goals

The specific learning goals of this course are:

- To become familiar with the the different kinds of financial investments
- To understand basics of market and trading, including different types of trade executions, orders, and financial markets
- To study basic trading algorithms as well as portfolio and multiasset trading
- To learn machine learning methods and their applications to various financial market prediction problems

Assessment

Class participation (5%) and homework assignments (45%) will assess the theoretical part of the learning goals. Each student will do a project and that will satisfy the algorithm and programming aspects of the learning goals.(50%)

Machine Learning

Rationale

The rapid growth of computer power and the needs for information technology have made Machine Learning an essential part of systems that must interpret data by classifying or clustering. This course gives a a thorough grounding in the methodologies, technologies, mathematics and algorithms currently needed by people who do research in machine learning.

Description

Machine learning is a branch of artificial intelligence, concerned with the construction and study of systems that can learn from data. Data may be numeric or symbolic and typically has the form of an N-tuple. The anthropomorphic term learning in the machine learning context means being able to predict some unobserved components of an N-tuple given some observed components of the N-tuple. This course provides a detailed explanation of many of the techniques used in machine learning and statistical pattern recognition.

Topic List

Topics may include but are not limited to:

- Bayesian Classification
  - Class conditional probabilities
  - Prior Probabilities
  - Gain Matrix
  - Maximizing Expected Gain
  - Minimax Classification
- Parametric Probability Models
- Non-Parametric Probability Models

- Making Decisions in Context
  - Conditional Independence
  - Hidden Markov Models
  - Forward Backward Algorithm
- Graphical Models
  - Semi-graphoids
  - Graphoids
  - Bayesian Nets
- Decision Trees
- Nearest Neighbor
- Linear Regression
- Logistic Regression
- Principal Component Analysis
- Neural Networks
  - The Perceptron Algorithm
  - The Back Propagation Algorithm
  - Deep Learning
- Linear Decision Rules
  - Fisher Linear Decision Rule
  - Support Vector Machines
  - Kernel Methods
- Ensemble Learning
- Evolutionary Learning
- Clustering

- K-Means Clustering
- Expectation Maximization
- Linear Manifold Clustering
- Gaussian Mixture Models
- Clustering Evaluation Measures

- Experimental Protocols
  - Training Sets
  - Test Sets
  - Cross-Validation
  - Performance Characterization

Learning Goals

The student must be able to demonstrate a working knowledge of the theoretical foundations and software of machine learning represented by the topics of

- Bayesian Classification
- Non-parametric Probability Models
- Clustering
- Dimensionality Reduction
- Performance Characterization

Assessment

Written exams and course projects will be assigned to make sure students are capable of identifying suitable algorithms for making certain types of predictions, designing experimental protocols to evaluate the performance of those proposed algorithms, and implement experiments on the algorithms and evaluations. 40% Important machine learning knowledge to be assessed by a final project includes but not limited to: Classification, Regression, Clustering, Dimensionality Reduction, and Performance Characterization. 60%

Modeling and Simulation

Rationale

Systems have become so complex that it is often the case that understanding them cannot be done analytically. Therefore, their behavior can be observed by modeling them and simulating them. This course will introduce the theories and applications of computer modeling and simulation, focusing on discrete event system modeling and simulation.

Course Description

Basic concepts of systems modeling, in-depth discussions of modeling elements, simulation protocols, and their relationships are covered. The modeling and simulation techniques will be illustrated by examples in DEVSJAVA, which is a Java implementation of the systematic and generic DEVS (Discrete Event System Specification) approach to modeling and simulation. Related application domains of this course include communication, manufacturing, social/biological systems, and business. Some advanced concepts and practices will be presented to attract students' interests in a seminar format.

Topic List

Topics may include but are not limited to:

- Computer Modeling
- Simulation Protocols
- Discrete Event Models
- Tools For Simulating Dynamic Models
- Tools For Simulating Complex Systems

Learning Goals

Students are expected to learn concepts of computer modeling and simulation applicable to a wide variety of technological, natural, and social systems, provide hands-on experience with modeling and simulation and specifically object-oriented simulation of discrete event models. After the class, students will establish a sound foundation of computer modeling and simulation and learn a set of computer-based tools for constructing, simulating and analyzing dynamic models of complex systems.

Assessment

The course includes three homework assignments and a term project (report and demonstration). The total grade is broken down as follows (subject to change): homework 1 20%, homework 2 20%, homework 3 25%, term project 35%.

Natural Language Processing

Rationale

Natural Language Processing (NLP) is one of the most important areas within Artificial Intelligence. It is deeply connected with Algorithms, Machine Learning, Programming Languages and Compiler Theory, and Automata and Formal Language Theory.

Course Description

Computers process massive quantities of information every day in the form of human language, yet machine understanding of human language remains one of the great challenges of computer science. How can advances in computing technology enable more intelligent processing of all this language data? Will computers ever be able to use this data to learn language like humans do? This course provides a systematic introduction to statistical models of human language, with particular attention to the structures of human language that inform them and the structured learning and inference algorithms that drive them. This is a lecture course, not a seminar course, but aims to cover both fundamental and cutting-edge research issues.

Topic List

Topics can include but are not limited to:

- Natural Language Understanding
- Hidden Markov Models
- Structured learning
- Inference Algorithms
- Coreference Resolution
- Keyphrase Extraction

- Extraction Based Summarization
- Text Mining

Learning Goals

- be able to write simple programs that understand natural language text by implementing classical NLP algorithms such as Viterbi and CKY
- be able to understand the mathematical theory of noisy-channel model
- be able to understand the formal machineries of describing natural language, such as finite automata and context-free grammars
- be able to understand current NLP research

Assessment

- Python Programming Exercises (50%)
- Quizes (25%)
- Final Project (25%)

Parallel Scientific Computing

Rationale

Computationally complex problems cannot be solved on a single computer. They need to be run in an environment of 100 to 1000 processors or more. Designing algorithms to efficiently execute in such a parallel computation environment requires a different thinking and mindset than designing algorithms for single processor computers. This course is designed to give the students the parallel computation perspective using the MPI framework.

Description

Computationally complex problems cannot be solved in a single computer either because they are combinatorially complex (NP-Hard) or because they are large involving much data such as very large matrices or much computation. The framework we use to solve these kinds of problems in parallel is called MPI, short for Message Passing Interface. We examine combinatorial problems such as Boolean Satisfiablility, Set Partitioning, Traveling Salesman and large problems such as might be in matrix multiplication or simulated annealing.

Topic List

- MPI Tutorial
- Amdahl's and Gustfason's Laws
- Matrix Multiplication
- Boolean Satisfiability
- Set Partitioning
- Simulated Annealing
- Graph Coloring

- Graph Betweenness
- Large Optimization Problems
- Student Presentations of Papers and their Programming Results

Learning Goals

- Learn how to design algorithms in parallel environments
- Learn how to use MPI in a parallel program
- Learn how to use MPI in solving
  - Clustering Problems
  - The Traveling Salesman Problem
  - The Set Partitioning Problem
  - Matrix Multiplication
  - Simulated Annealing
  - Optimization Problems
  - Graph Coloring
  - Graph Betweenness

Assessment

Every student will work on two different MPI programs to solve computationally complex problems of their own choosing. In the second half of the course they will report on their algorithms and the results of their programs. Grades will be based entirely on their programs and presentations.

Pattern Matching

Rationale

The advent of the worldwide web, next generation sequencing, and increased use of satellite imaging have all contributed to the current information explosion. One of the most basic tasks common to many applications is the discovery of patterns in the available data. To render the searching of big-data feasible, it is imperative that the underlying algorithms be efficient, both in terms of time and space. Pattern Matching is a branch of theoretical computer science whose ideas are used in practice daily in many different data-driven areas, including (but not limited to) word processors, web search engines, biological sequence alignments, intrusion detection systems, data compression, database retrieval, and music analysis. This course gives a student training in the process of developing and analyzing efficient algorithms through the study of pattern matching algorithms that are used for searching and indexing large textual data.

Description

Pattern Matching is one of the fundamental problems in Computer Science. In its classical form, the problem consists of 1-dimensional string matching. Given a string (or text) T and a shorter string (or pattern) P, find all occurrences of P in T. Over the last four decades, research in Pattern Matching has developed the field into a rich area of algorithmics. This course covers several variants of the pattern matching problem. Emphasis is placed on the algorithmic techniques used to speed up naive solutions, and on the time complexity analysis of the algorithms.

Topic List

Topics may include but are not limited to:

- Exact String Matching
  - Knuth-Morris-Pratt

  - Boyer-Moore

- Suffix Trees and Applications
  - Wiener
  - Ukkonen
  - Practical Implementation Issues
- Multiple Pattern Matching
  - Aho-Corasick
  - Generalized Suffix Tree
- Approximate Pattern Matching
  - Hamming Distance
  - Edit Distance (dynamic programming)
  - Don't Cares (convolutions)
- Lowest Common Ancestor
  - Range Minimum Query
  - Complete binary trees
- Periodicity
  - Squares
  - Repetitions
  - Approximate tandem repeats
- Palindromes
  - Manacher
  - Suffix trees
  - Palindromes with errors
- Naming
  - Karp-Miller-Rosenberg

  - Lyndon Word Naming

- 2-Dimensional Matching
  - Bird-Baker
  - Dueling for Alphabet Independent Matching (Amir-Benson-Farach)
  - Small Space 2d Matching
  - 2d Dictionary Matching

- Suffix Arrays
  - Definition and Construction
  - String Matching with suffix array
  - Burrows-Wheeler Transform

Learning Goals

The student must be able to demonstrate knowledge of how to apply and analyze the following algorithmic tools:

- Finite Automata
- Dynamic Programming
- Suffix Trees
- Naming
- Convolutions
- Dueling

Assessment

Two written exams, one midterm and one final will be used to assess students' knowledge of the subject. The questions on the exams will address problems related to those that were discussed in class, and the student will have to demonstrate the ability to apply techniques learned to new problems. For example, after learning dynamic programming for edit distance with unit cost, a student will have to answer a question related to weighted edit distance.

Programming Massively Parallel Systems

5 Rationale

Computationally complex problems such as graphical representations of movement cannot be processed in a reasonable amount of time on a single CPU. Currently, most graphical computations and many scientific calculations involving large datasets and complex systems are run in a massively parallel environment. Designing algorithms to efficiently execute in both time and memory usage in such environments requires an understanding of concurrency and the hardware requirements of massively parallel systems, for example, Graphical Processing Units (GPUs). This course is designed to give the students an introduction to the concepts and usage of GPUs and the CUDA extensions to the C/C++ languages.

6 Description

A survey of the approaches to massively parallel computer applications with emphasis on using graphical programming units (GPUs) and the CUDA extensions to the C/C++ programming languages. Comparisons between multicore CPUs and multi-processor GPUs will be given. Issues such as organization of large data sets, memory usage, and communication concerns will be addressed. Different levels of concurrency will also be discussed with most the focus on thread level-concurrency. Also multiple data streams on a single GPU and multiple GPUs will be covered with quick reviews of OpenMP and OpenMPI usage. Standard problems will be discussed.

7 Topic List

Topics may include but are not limited to:

- The Graphical Processing (GPU) Capabilities
- CUDA
- Threading Concurrency

- Open MP
- Open MPI

8 Learning Goals

1. The student will understand the concept of concurrency in an environment involving many parallel processors. 2. The student will understand the relationship between programming using a traditional multicore-CPU versus using massively parallel GPUs. 3. The student will acquire an understanding of the importance of the memory model needed for massively parallel programming. 4. The student will gain experience programming with CUDA on a GPU. 5. The student will be introduced to how to use multiple GPUs connected to a single CPU and using multiple GPUs over a network.

9 Assessment

- At least five weekly or biweekly assignments 25%
- Each student will be expected to carry out a project transforming an existing serial code or writing a new parallel code to use CUDA on a GPU. The student will write a paper in a research format describing their project 25%
- One mid term exam 25%
- A final presentation of the project in class 25%

Quickest Detection and Applications

10 Rationale

This is a comprehensive course on the topic of quickest detection that has been offered at the Graduate Center since the Fall of 2008. It covers the fundamental theory of quickest detection, the algorithms associated with it and the applications of quickest detection in a variety of fields, namely finance, signal processing and 3D Computer Vision. The basic financial notions of asset pricing and risk are also discussed. There is a great commitment in research and a huge commercial interest in all of the application areas on this field. A number of graduate students in the PhD program are currently using such algorithms in their research.

11 Course Description

The problem of detecting abrupt changes in the statistical behavior of observation arises in a variety of fields including signal processing, computer vision and finance. Using the mathematical methods of statistical sequential techniques and stochastic optimization, this course describes the fundamentals underpinning the field providing the background necessary to design analyze and understand quickest detection algorithms and stopping times. In this course we will provide a unified treatment of several different approaches to the quickest detection problem and draw examples from the field of signal processing, finance and computer vision. The course also covers models used in finance and signal processing, brownian motion, Ito calculus, markov processes and the fundamental theory of asset pricing. The notion of stopping time and its association with detection algorithms is further examined. Moreover, connections between detection algorithms and drawdown measures are drawn. The course finally examines the use of detection algorithms in online trading and the detection and classification of objects in point clouds of urban scenes.

12 Topic List

Topics may include but are not limited to:

- Statistical and sequential hypothesis testing
- The sequential probability ratio test and the cumulative sum algorithms as stopping times
- Applications to computer vision, algorithmic trading and signal processing
- Modeling in finance and signal processing, brownian motion
- Ito calculus, martingales, markov processes and the fundamental theory of asset pricing
- Drawdowns measures of risk and connections to detection

13 Learning Goals

A general understanding of the importance of quickest detection in various fields.

- Understanding of the notion of stopping time and online detection
- Ability to formulate research questions and to write research reports
- Ability to present technical talks
- Understanding of selected detection algorithms and how they can be applied in various fields
- Knowledge of basic models and stochastic processes used in signal processing and finance
- Knowledge of the fundamental theory of asset pricing, the notion of risk and how it relates to drawdowns and detection

14 Assessment

The course requires a midterm exam, a project and a final. Each student will prepare a research report either related to the theoretical study of detection algorithms or to adjusting and applying detection algorithms in a topic of their choice. The report will also be supported by a student presentation in class. Grading will be based on the attendance, student presentation, midterm, final and the final research report and project. Students can work in groups if they desire so for the final project, upon the consent of the instructor. I will provide a list of possible topics that would be appropriate for the final project and report. Student can pick a topic from this list or can also work on any other related topic of their choice subject to instructor approval.

Social and Cultural Computing

Course description

The joint availability of massive social and cultural data sets (including social media and digitized cultural artifacts) make possible fundamentally new paradigms for the study of social and cultural activities and histories. While the recently emerged field of social computing started to explore some of these possibilities, we are only at the very beginning.

Image analysis will be used to illustrate basic concepts of exploratory data analysis. We will also examine both papers from computational social science science and data analysis/visualization projects by designers and artists

Topic List

Topics may include but are not limited to:

- Basic concepts and methods of data analysis (using R)
- visualization techniques (using R / Mondrian / Tableu / other software)
- use of visualization for explorative data analysis
- elements of graphic design as they relate to visualization and project web site design
- strategies for presenting projects online
- how to write effective project descriptions for the web presentation;
- promoting projects through social media and getting media coverage

Learning Goals

- Understand current research directions in social and cultural computing
- Identify not yet explored possibilities in working with social media data
- Learn how to prepare data for analysis

- Learn basic techniques for data exploration
- Becom proficient with data visualization techniques
- Understand the structure and organization of web projects that present results of social and computing projects, or visualizations of cultural data
- Learn how to clearly and effectively write project summaries for general audiences
- Learn how to contact members of the press / getting project publicity and promote projects using social media;

Assessment

Students will complete 3 practical assignments which involve organizing data set, analysing them and creating effective visualizations. The goals of these assignments is to meet learning goals 3-5. Students will work in groups on final projects which address goals 6-8. They will be also responsible to completing and discussing readings and sample projects (goals 1-2).

Text Mining and Classification

Rationale

With the explosion of textual data on the world-wide web, text mining has become an important area of research. Text sources such as blogs, literature, social media, web pages and news articles can be analyzed to learn patterns, opinions, trends, and ideas. Text mining is a sub-area of data mining that deals with unstructured text. Algorithms have been developed for learning from unstructured text, and these often have practical applications in areas such as health care, advertising and homeland security.

Description

Text mining can be defined as the process of finding or learning patterns from textual data to aid in decision making. This course will include the study of different representations of textual data and the algorithms used to glean new information from the data. It encompasses ideas from many other areas in computer science including artificial intelligence, machine learning, databases, information retrieval, and natural language processing.This class will primarily focus on the statistical methods for text mining, including machine learning techniques that are used to facilitate decision making.

Topic List

The topic lists may include but is not limited to:

- Text Data Representation
  - Bag of Words
  - Named Entities
  - Relationships
- Text Categorization
  - Rule-based classifiers

  - Decision trees
  - Nearest neighbor
  - Maximum margin classifiers
  - Probabilistic classifiers
  - Semi-supervised Learning using EM
- Text Clustering
  - Hierarchical clustering
  - K-means clustering
  - Dimensionality Reduction
  - Latent semantic indexing
- Topic Modeling
  - pLSI
  - LDA
- Information Retrieval and Text Mining
  - Key Word Search
  - Indices
  - Link Analysis
  - Text Mining from Social Media
- Sentiment Analysis

Learning Goals

Students should be able to:

- Demonstrate an understanding of the algorithms that were taught in class.
- Use current text mining software with practical, real-world data sets in a way that aids decision making.

Assessment

Homework sets and a final exam with questions that target the learning goals will be used to assess student knowledge. In addition, a semester project will be used to assess student ability to use text mining packages, and to choose appropriate representations, algorithms, and testing methodology.

Vision, Brain and Assistive Technologies

Rationale

Based on the World Health Organization 2012 Report, there are more than 285 million visually impaired people, of which 39 million are blind. About 65% of all people who are visually impaired are aged 50 and older, while this age group comprises about 20% of the world's population. With an increasing elderly population in many countries, more people will be at risk of age-related visual impairment. Research on multimodal and alternative perception will have a long term impact on the health and wellness of society, not only for the visually challenged, but for people who often work in dangerous environments, such as firefighters, drivers and soldiers.

Course Description

This course will discuss modern vision science and explore how the brain sees the world, thus including introductory on computational neuroscience, motion, color and several other topics. Then the basics of computer vision will be introduced, for both preprocessing and 3D computer vision. Finally, we will discuss the needs and state-of-art in sensing, processing and stimulation for assisting visually challenged people (blind and visually impaired people) using advanced technologies in machine vision, robotics, displays, materials, portable devices and infrastructure.

The course will be offered as an interdisciplinary seminar course, in which a few lectures will be provided from the recommended textbook on human vision as well as the lecture notes of the instructor on computer vision, and then students from mathematics, physics, electrical engineering, computer science and psychology and other social sciences will be assigned to read, present and discuss materials in vision, brain, computing and devices for assisting the visually impaired. The major reading materials will include the papers and talks from the references below. Finally students will team up to do course projects.

Topic List

Topics may include but are not limited to:

- Introduction to Human and Computer Vision
- Human Eyes and Visual Brain
- Depth and Color
- Image Formation: Digital Image Basics
- Image Enhancement
- Camera Models
- Stereo Vision and Visual Motion
- Assistive Technologies for the Blind and Visually Impaired
- Visual Prosthetics
- Vision Algorithms for the Blind and Visually Impaired

Learning Objectives

Through the course, the students should be able to:

- Demonstrate basic knowledge of human brain and vision, visual impairment, and computer vision
- Identify need of visually impaired people and related assistive technologies
- Apply computer vision algorithms/techniques and assistive technologies to assisting visually impaired

Assessment

Grading policy:

- an in-class exam of the basics theory (30%)
- student reading reports and presentations (30%)
- project reports and presentations for applications(40%).