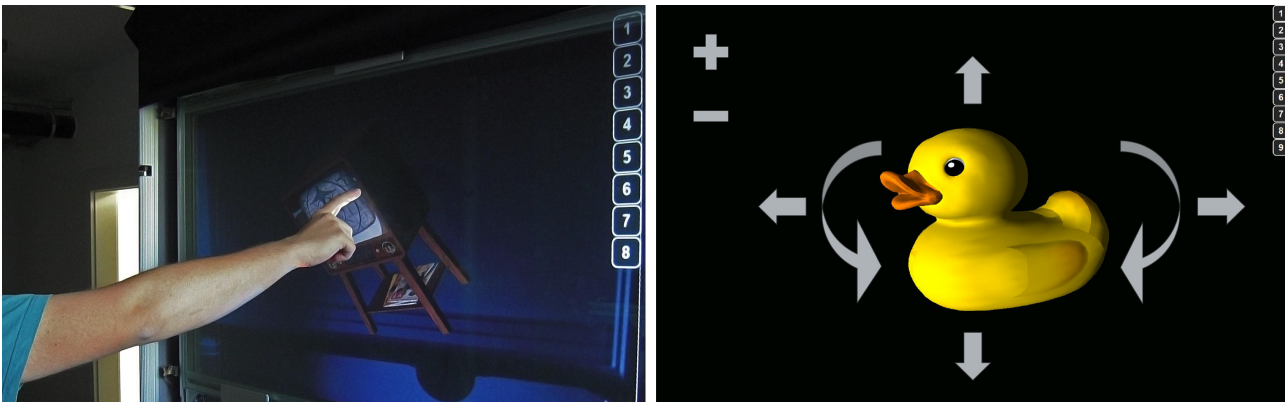

3D-Videokonferenz mit Toucheingabe (Face-2-Face)

Arbeitsbereiche Technische Informatik / Eingebettete Systeme

Dennis Hospach Philipp Mock

Projektbeschreibung

Face-2-Face ist ein Multitouch-Aufbau für kollaboratives Arbeiten über Netzwerk. Im Rahmen dieses Projekts soll ein bestehender 3D-Modellbetrachter weiterentwickelt werden. Die Software kann simultan durch Multi-Touch-Eingaben an zwei Installationen bedient werden. Als Teilnehmer lernt ihr Toucheingaben zu verarbeiten und in einem graphischen 3D-Interface zu nutzen. Hierzu verwendet ihr aktuelle Technologien aus der Computergrafik und der Mensch-Maschine-Interaktion.



Ziele

- Erweiterung der Gesteninteraktion
- Verbesserung der graphischen Darstellungsqualität
- Hinzufügen nicht-interaktiver Teilnehmer (z.B. Tablet)
- 3D-Stereo Darstellung (optional)
- Integration optischer Marker (optional)

Inhalte

- Programmierung von Multi-Touch-Interfaces
- 3D-Darstellung mit OpenGL
- Videoverarbeitung in verteilten Anwendungen

Technische Grundlagen

- C++
 - OpenGL (Version 4)
 - Qt 5
-

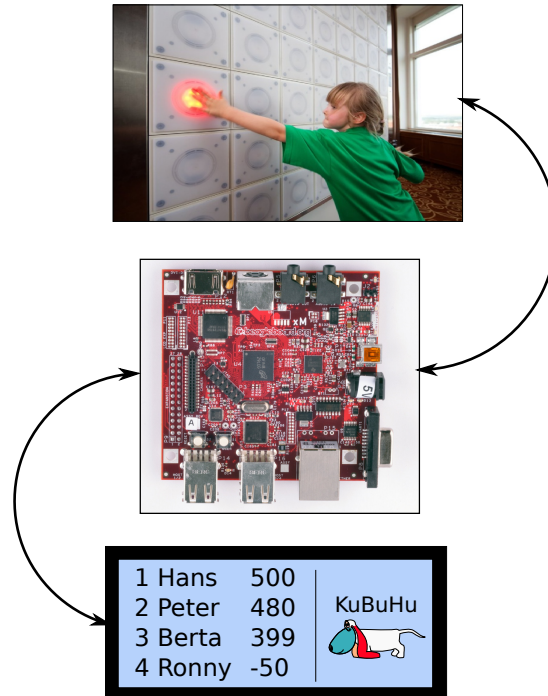
Kunterbunter Hund 3.0

Arbeitsbereich Eingebettete Systeme

Sebastian Burg Dustin Peterson Philipp Schlicker

Projektbeschreibung

Beim **Kunterbunten Hund** handelt es sich um ein System, welches u.a. aus einem Mikrocontroller und einer Lichtschalterwand besteht, mit Hilfe dessen Spieleklassiker wie Memory oder Simon Says umgesetzt und gespielt werden können. Das System wurde von Studenten in zwei Programmierprojekten entwickelt. In den letzten Semestern wurde u.a. die Entwicklung einer zugehörigen Android-App und einer Serverapplikation fortgesetzt. In diesem Semester soll die Entwicklung einer komplett entkoppelten Version im Fokus stehen. Neben einem zusätzlichen Display für Highscore-Anzeige und Debugging-Zwecke soll ein Client/Server-System entwickelt werden, welches es ermöglicht, Spiele via Ethernet von einem beliebigen PC aus auf das Beagle Board zu laden. Zudem soll dieses Client/Server-System eine erweiterte Möglichkeit zum Debugging der Applikation bei Fehlern ermöglichen.



Ziele

- Entwicklung einer Standalone-Version des Kunterbunten Hunds
- Anschluss eines Displays zur Highscore-Anzeige
- Entwicklung einer TCP/IP-basierten Plattform zur Administration der BeagleBoard-Plattform

Inhalte

- Programmierung eingebetteter Systeme
- Programmierung verteilter Systeme
- Konfiguration von Linux-Systemen
- Kommunikationsnetze

Technische Grundlagen

- C/C++
 - Linux
 - BeagleBoard
 - TCP/IP
-

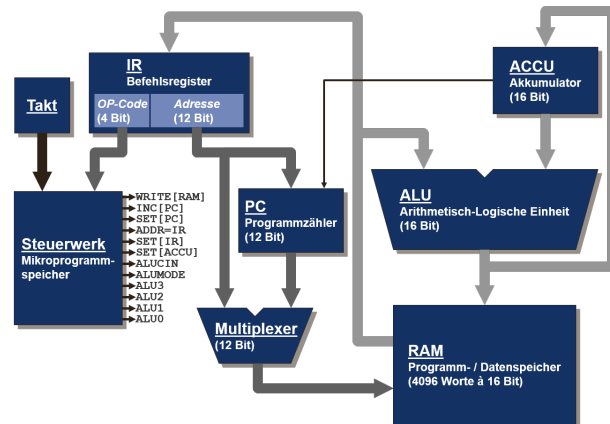
Toy Processor - eine Mikroprozessorsimulation

Arbeitsbereich Eingebettete Systeme

Fiete Botschen

Projektbeschreibung

Falls Du dich schon immer gefragt hast, wie ein Computer auf unterster Ebene funktioniert, bist Du hier richtig. In diesem Projekt soll für die Vorlesung "Einführung in die Technische Informatik" eine Mikroprozessorsimulation für den Toy-Prozessor entwickelt werden. Die Simulation soll den Aufbau des abgebildeten Toy-Prozessor eins zu eins abbilden und den Code nicht nur ausführen, sondern die Ausführung auch detailliert grafisch darstellen. Im Single-Step Modus können dadurch die Abläufe innerhalb des Prozessors während einer Programmausführung einfacher nachvollzogen werden. Die grafische Benutzeroberfläche sollte Zustandsänderungen des Prozessors grafisch hervorheben und gleichzeitig die Manipulation des Prozessors ermöglichen, indem zum Beispiel Registerinhalte jederzeit vom Benutzer geändert werden können. Außerdem soll der Simulator die Ausführung selbstgeschriebener Programme unterstützen. Hierfür muss ein Parser entwickelt werden, der die verschiedenen Codeformate (Bitcode, Assembler) verarbeiten kann.



Blockschaltbild des Toy-Prozessors

Ziele

- Entwicklung eines grafischen Simulators für den Toy-Prozessor

Inhalte

- Programmierung einer Mikroprozessorsimulation
- GUI Design
- Programmierung eines Parsers

Technische Grundlagen

- C++
 - GUI Framework Qt
-

Kognitive Steuerung von selbstständigen Agenten / Graph-Interaktions App

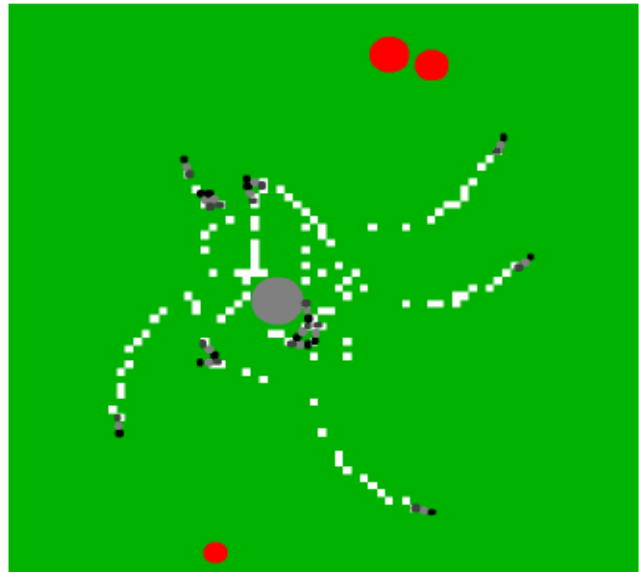
Arbeitsbereich Kognitive Modellierung

Prof. Dr. Martin Butz, Dr. Jan Kneissler

Termin für Treffen: Mo 10-12 bzw. Fr 10-12 (bitte nur anmelden wenn mindestens einer der beiden Termine geht). Es ist bei der Anmeldung noch nicht nötig, sich für eines der beiden Themen zu entscheiden.

Projekt 1

Es wird darum gehen eine kognitive Steuerung eines Agenten in einer simulierten Umgebung zu erlernen. Der Agent nimmt die Umgebung über Sensoren wahr und hat die Möglichkeit, über einfache Aktoren Einfluss auf die Umgebung auszuüben (Arm/Hand bewegen, Fortbewegung, Objektinteraktion). Über Feedbacksignale (Energiegewinn, Schmerz) soll er erlernen, sich sinnvoll in der Umgebung zu verhalten und so lange wie möglich zu überleben.

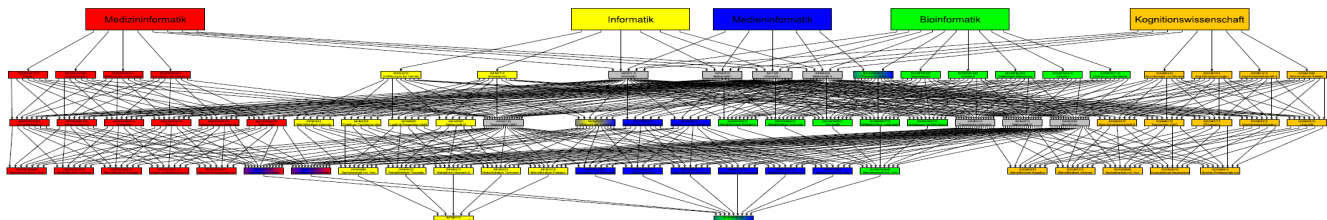


Technische Grundlagen

- Java
- Eclipse
- GIT

Projekt 2

Dieses Projekt wird bei ausreichend hoher Teilnehmerzahl/Interesse alternativ angeboten.



Es soll eine Anwendung zum interaktiven Umgang mit umfangreichen, hierarchisch strukturierte Graphen erstellt werden. Beispielsweise könnten die Knoten angebotene Vorlesungen in den diversen Informatikstudiengängen darstellen und der Benutzer kann durch anklicken seinen individuellen Studienplan erstellen. Dabei reorganisiert sich die Darstellung des Graphen dynamisch: je nach bisheriger Auswahl werden sinnvolle/mögliche Fortsetzungsvorlesungen an erster Stelle vorgeschlagen.

Technische Grundlagen

- Java+Swing / JavaScript
 - Eclipse
 - GIT
-

Mario lernt seine Gegner kennen

Arbeitsbereich Kognitive Modellierung

Prof. Dr. Martin Butz M. Sc. Fabian Schrodtt

Projektbeschreibung

Das zur Verfügung gestellte "Mario Lives"-Framework basiert auf einem Super Mario Klon (Mario AI), in dem die Spielfigur von einem intelligenten Agenten gesteuert wird. In den vergangenen Semestern wurde das Framework für eine beliebige Anzahl von Agenten angepasst, die Sprachfähigkeiten (per Mikrophon und Sprachsynthese) beherrschen, ihre Umwelt durch Objektinteraktionen kennenlernen und Motivationen haben, anhand derer sie sich selbst Ziele setzen. Im Rahmen des Praktikums "Mario lernt seine Gegner kennen" sollen diese selbst-motivierten Agenten nun mit der Fähigkeit ausgestattet werden, Einzel- und Teamstrategien zum Bewältigen bestimmter Aufgaben zu erlernen – und dabei ihre Gegner zu besiegen. Dazu werden die Agenten mit unterschiedlichen Fähigkeiten und Motivationen ausgestattet. Es sind Algorithmen zur Wegplanung und zum Wissenserwerb durch Beobachtung anderer Spieler in einer simulierten Umgebung mit beweglichen Zielen auszuarbeiten bzw. zu anzupassen.



Beginn und Ort

Zeitraum für erstes Treffen zur Projekteinführung: 22.4.-28.4., Terminabsprache erfolgt per Doodle
Nach Ende der Vorlesung: Mittwochs, 14 Uhr in C412 oder C421

Ziele

- Ergänzung eines Multiplayer-Team-Szenarios
- Implementierung von spielerbezogenem Wissen
- Ergänzung von Freund- und Gegner-Interaktionsstrategien

Inhalte

- Einarbeitung in bestehendes Projekt
- Erwerb von situativem Effekt-Wissen in einer simulierten Umwelt
- Wegplanung durch Vorwärtssimulation
- Planung von Handlungssträngen durch Vorwärtssimulation
- Interaktions- und Manipulationsstrategien intelligenter Agenten
- Ausarbeitung und Modellierung kognitiver Konzepte

Technische Grundlagen

- Java-Programmierung
 - Umgang mit Eclipse, SVN und Trac
 - SCUM-Softwareentwicklungsmodell
-

The Construction of a SASL Compiler (Compilerbau)

Lehrstuhl für Datenbanksysteme

Tobias Müller Benjamin Dietrich

Projektbeschreibung

Was passiert, wenn ihr den Compiler für eure Liebessprache auf Euren Quellcode loslasst? Hier findet Ihr darauf eine Antwort für die einfache funktionale Programmiersprache SASL.

Also **back to the roots!**

Basierend auf einem ausführlichen Tutorial werdet ihr einen *vollständigen Compiler* (mit Lexer, Parser und Optimierer) selbst bauen und somit ein Gefühl dafür entwickeln, was alles passieren muss, bevor eine Sprache ausgeführt werden kann.



Der Code, den Euer Compiler erzeugen wird ist auf einer simplen virtuellen Maschine (SK-Reduction Machine) ausführbar, die auch Teil eurer Anstrengungen sein wird.

Am Ende werdet ihr eine einfache aber vollständige Implementierung eines Compilers und einer virtuellen Maschine für SASL “in euren Händen halten”.

Ziele

- Vollständige Implementierung eines Compilers und einer virtuellen Maschine für SASL
 - Lexer, Parser, Optimierer
 - Einfache Ausführungsmaschine (SK-Reduction Machine)

Inhalte

- Die funktionale Programmiersprache SASL
- Compilerbau
- Design-Patterns für Compilerbau

Technische Grundlagen

- Java
 - Nicht nötig, aber von Vorteil ist die Kenntnis einer funktionalen Programmiersprache
-

MinION simulator (Bioinformatics)

Algorithms in Bioinformatics Group

Anna Górska Sina Beier

Project Description

DNA sequencing is one of the fastest growing fields - one of the newest advances is the MinION technology, a sequencer of the size of an USB-stick.

It is also different than previous technologies - MinION produces sequences *online* which means that they appear in real-time on your computer ready to be analyzed. Unfortunately currently MinION data are affected by a big error rate. For these reasons current algorithms are not sufficient to analyze MinION-produced sequencing data.

We aim to produce a user-friendly Java application for simulation of MinION data.

The main project language will be English, as is common for projects in bioinformatics.



Goals

- Creation of a well-designed and user-friendly application for sequencing data simulation.
- Working on a real life problems in the context of the newest technology.

Topics

- Sequence-based bioinformatics
- MinION sequencing technology
- Research driven programming

Technical Background

- Java GUI programming
- Analysis of the existing data-sets to generate a real error distribution
- Code sharing using GitHub, diving-in with GitHub community

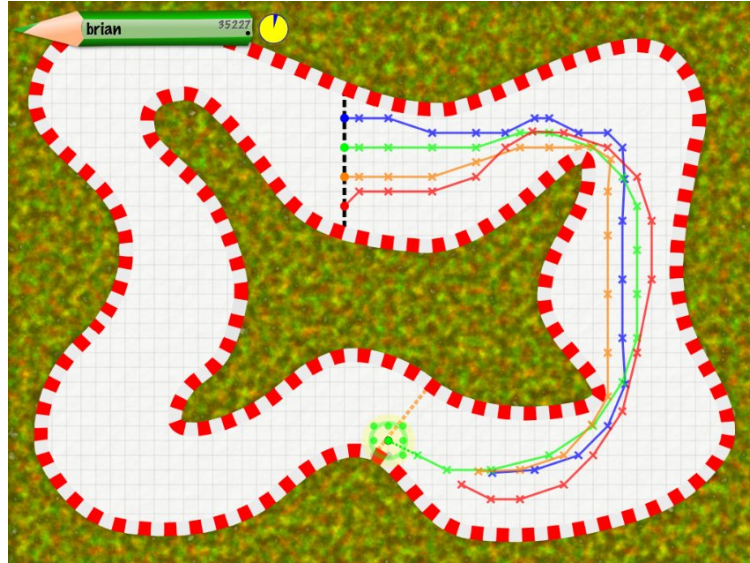
Netzwerk Racetrack

Arbeitsbereich Algorithmik

Michalis A. Bekos Till Bruckdorfer

Projektbeschreibung

“Racetrack” ist ein Strategiespiel mit Papier und Bleistift für mindestens zwei Spieler, welches ein Autorennen simuliert, bei dem Autos eine gewisse Trägheit haben. Auf einem karierten Blatt Papier wird eine Rennstrecke gezeichnet. Die Autos der Spieler stehen zu Beginn an einer Startlinie auf einem Gitterpunkt mit Geschwindigkeit 0 in x- und y-Richtung. Reihum bewegen die Spieler ihre Autos mit einer Geschwindigkeit, die sich bezüglich des vorherigen Zuges in jeweils x- und y-Richtung nur um +1, 0 oder -1 ändern darf. Wer die Begrenzung berührt verliert das Spiel. Gewinner ist der Spieler, dessen Auto zuerst die Ziellinie überquert.



Ziele

- Analyse und Modellierung des Spiels mittels objektorientierten Programmierens
- Entwicklung einer Client-Server-Struktur für Netzwerkanwendung
- Erstellung eines Spielverwaltungssystems und Realisierung einer GUI
- Erlernen der Verwendung externer Bibliotheken und die Zusammenarbeit im Team
- Optional: Implementierung verschiedener KI-Heuristiken

Inhalte

- Java-Programmierung
- Netzkommunikation
- Nebenläufige Programmierung
- Versionsverwaltung
- Graphentheorie

Technische Grundlagen

- Java
 - Umgebung: Eclipse
-

Xiangqi

Videogames

Luis de la Garza Timo Sachsenberg

Applied Bioinformatics Group

Description

Xiangqi is also known as *Chinese Chess*. It is a strategy board game for two players and is one of the most popular board games in China.

As with other board games in the *chess family*, the game represents a battle between two enemy armies, the purpose being to capture the enemy's general.

The rules of Xiangqi are quite simple, but a game can evolve into complex scenarios challenging your strategic thinking skills.



Objectives

- Program a Xiangqi videogame in which two people could play across the internet
- Program a simple artificial intelligence (AI) and let it play against another human or computer!
- Work in small groups and simulate a professional development environment

Contents

- GUI programming
- Multithreaded programming
- Network programming
- Object-oriented programming

Technical aspects

- Java (Java I/O, Java Threads, Java Swing, etc.)
 - Eclipse or any other Java IDE of your preference
-

Verarbeitung und Visualisierung formaler Sprachen (Desktop-Anwendung)

Arbeitsgruppe Theoretische Informatik

Silke Czarnetzki Michael Ludwig

Projektbeschreibung

Formale Sprachen bilden die Grundlage der theoretischen Informatik und haben innerhalb der Informatik ein breites Anwendungsspektrum. Den Teilnehmern sollten die Grundzüge der Theorie formaler Sprachen aus der entsprechenden Vorlesung bekannt sein. Besondere Aufmerksamkeit wurde dort den kontextfreien und den regulären Sprachen geschenkt. Reguläre Sprachen haben eine vergleichsweise kleine Ausdrucksstärke, sind jedoch sehr gut handhabbar. Probleme reguläre Sprachen betreffend sind entscheidbar, außerdem haben sie gute Abschlusseigenschaften und es gibt es viele Beschreibungsmöglichkeiten, wie endliche Automaten, Grammatiken, reguläre Ausdrücke, endliche Monoide etc.

In diesem Programmierprojekt soll nun eine Software zur Verarbeitung und Visualisierung formaler Sprachen erstellt werden, wobei der Fokus auf den regulären Sprachen liegen soll.

Ziele

Die Verarbeitung betrifft das Eingeben und Speichern von Sprachbeschreibungen (z.B. als Automat), das Überführen von einer Beschreibung in eine andere (z.B. Automat hin zu regulärem Ausdruck), sowie das Anwenden elementarer Operationen, wie das Schneiden zweier Sprachen.

Für die verschiedenen Beschreibungsarten sollen geeignete Visualisierungen gefunden werden. Ebenso wünschenswert wäre das Darstellen verschiedener Vorgänge und Algorithmen, wie z.B. das Erkennen eines Wortes in einem Automaten oder das Minimieren eines deterministischen Automaten.

Die modulare Natur der Aufgabenstellung erlaubt es den Umfang des Projekts an den Bearbeitungszeitraum anzupassen.

Das Endprodukt könnte Verwendung in der Lehre finden, z.B. im Rahmen der Veranstaltung Informatik III. Hinreichende Funktionalität und Qualität vorausgesetzt ist ebenso ein forschungsbezogener Einsatz denkbar.

Inhalte

Bei der Bearbeitung der Aufgabe wird man sich mit folgenden Inhalten auseinander setzen.

- Implementierung grundlegender formalsprachlicher Konzepte, Algorithmen
- GUI und Visualisierung
- Datenhaltung
- Modulare und skalierbare Softwarearchitektur

Technische Grundlagen

Als Programmiersprache schlagen wir Java vor. Weitere Werkzeuge können sein: Swing, XML, sowie weitere Bibliotheken, etwa zum Graphenzeichnen oder zur Datenhaltung und Verarbeitung.

Automatic Colorization of gray-scale images using Deep Learning

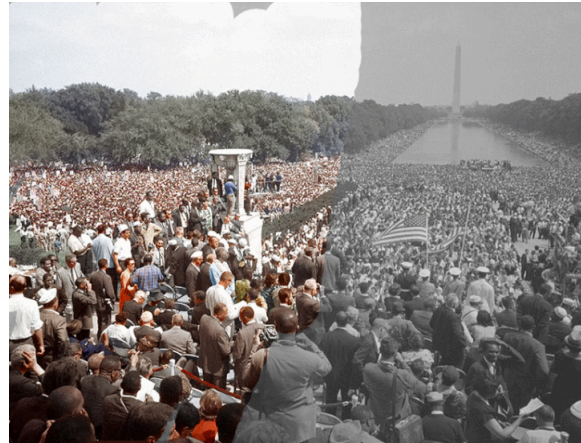
Computer Graphics (Prof. Dr.-Ing. Hendrik Lensch)

Patrick Wieschollek

project description

Assigning color to gray-scaled images is an ill-posed problem relying heavily on human interaction. For example, a red car is mostly indistinguishable from a blue car when photographed in black-and-white. You will use a state-of-the-art method in machine learning to build a system that automatically learns how to colorize gray-scale images by predicting aesthetically-believable colors.

Your task is to build an efficient system consisting of a responsive web frontend running on a NodeJS server and background worker, which will predict the colorized version of a requested gray-scaled image using a deep convolutional neural network.



goals

- create and train a deep convolutional neural network
- create a user-friendly application for automatic colorization of images

content

- train a deep neural network on the GPU¹ including writing a new C++-layer for this framework
- write a background worker which uses the Caffe-implementation to predict the colorized version
- create a responsive web-frontend which handles the user request and returns the result
- create a test interface to capture the performance of the convolutional neural network

technical foundations

- C++, JavaScript
- GIT
- GoogleTestingFramework, NodeJS or equivalent (at a pinch PHP)

¹using Caffe <http://caffe.berkeleyvision.org/>

2D and 3D Sclow Plots

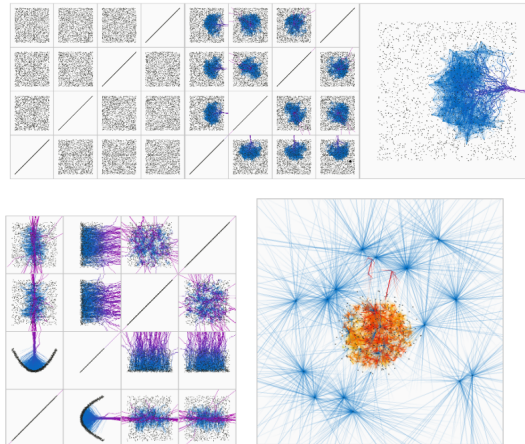
Computer Graphics (Prof. Dr.-Ing. Hendrik Lensch)

Patrick Wieschollek

project description

Correlation analysis in big data demands powerful data visualization techniques. A useful tool for such a correlation analysis is an *augmented scatter plot* for exploring high-dimensional data. For example, the illustrated scatter plot on the right (top row) contains a 4D hypercube, where a cone has been cut out. It is impossible to discover this empty region using a classical visualization algorithms.

You will write a cross-platform desktop application in Qt5 implementing a new visualization algorithm. Ideally this application should provide an interactive "fly through".



goals

- a cross-platform desktop application in Qt which visualizes high dimensional data from different data sources
- a fast and user friendly representation¹ of these visualization including rich export functions²
- a virtual fly-through on a 3D projection of this visualization

content

- implement a recent published visualization algorithm (Sclow Plots)
- write an cross-platform application³ in Qt5 (C++)
- augmented scatter plot on orthographic projection of the data in 2D
- use OpenGL or another 3D engine for visualization in 3D

technical foundations

- C++, Qt5
- GIT
- GoogleTestingFramework, OpenGL
- ideal: knowledge in GPU programming or at least OpenMPI and Eigen3 library⁴

¹<http://colorbrewer2.org/>

²png, postscript, pgf, pdf

³<http://qt-project.org/qt5>

⁴<http://eigen.tuxfamily.org/>

Entwicklung eines Video-Streaming-Dienstes (Netzwerke)

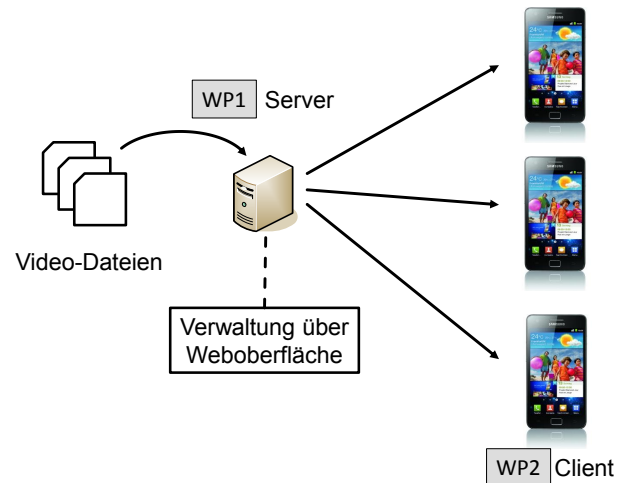
Lehrstuhl Kommunikationsnetze

Florian Heimgärtner Michael Höfling Andreas Stockmayer

Projektbeschreibung

Videoplattformen wie Youtube sind aus unserem täglichen Leben nicht mehr weg zu denken. Wir können auf beliebige Videos zugreifen, die im Internet gespeichert sind, und diese auf beliebigen Geräten abspielen lassen.

Ziel dieses Projektes ist die Entwicklung eines einfachen Video-Streaming-Dienstes. Neben der Entwicklung des Streaming-Servers soll eine leicht zu bedienende Weboberfläche zur Verwaltung des Video-Streaming-Dienstes entworfen und implementiert werden. Abgerundet wird das Projekt durch einen Streaming-Client für Android-Smartphones und -Tablets.



Ziele

Das Projekt ist in zwei Arbeitspakete (WPs) aufgeteilt:

- WP1: Entwurf und Implementierung der Serverkomponente (*maximal 4 Studierende, 1 Betreuer*)
- WP2: Entwurf und Implementierung der Clientkomponente (*maximal 4 Studierende, 1 Betreuer*)

Wir stellen die notwendigen *Schnittstellendefinitionen*, das *Datenbankmodell*, sowie entsprechende *Demo-Videos* bereit, damit in den Gruppen möglichst unabhängig voneinander gearbeitet werden kann. Bei entsprechend hohem Interesse ist eine zweite Clientgruppe möglich.

Inhalte

- Client-Server-Kommunikation mit JSON (*JavaScript Object Notation*)
- Video-Streaming über HTTP
- Android-Programmierung (GUI und Backend)
- Web-Programmierung (HTML, PHP, JavaScript)
- Datenbank-Programmierung (SQL)

Technische Grundlagen

- Programmiersprachen: Java, PHP, SQL, HTML, JavaScript
 - Kollaboration: SVN, Trac/Redmine
 - Entwicklungsumgebung: Eclipse, Android Studio
-

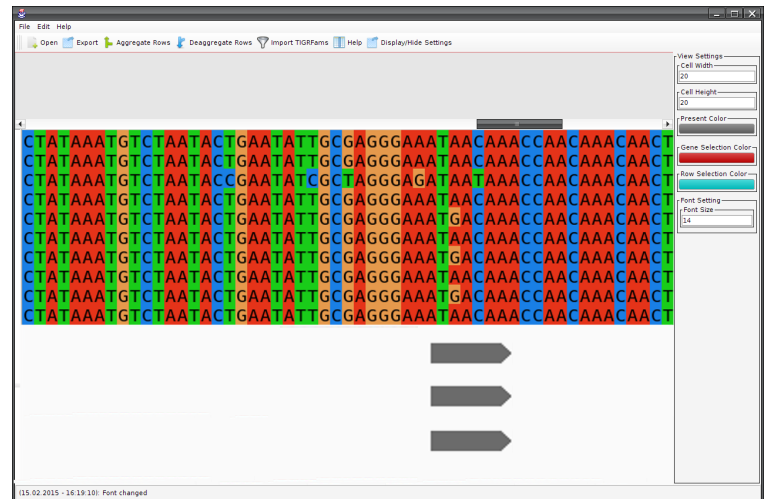
SuperGenome-Browser (Bioinformatik)

Arbeitsgruppe: Integrative Transkriptomik

André Hennig Alexander Peltzer

Projektbeschreibung

Multiple Genom-Alignments spielen in der Bioinformatik eine zentrale Rolle um homologe Regionen zwischen verschiedenen Genomen zu finden. Die Visualisierung des Alignments mit der Integration von Gen-Annotationen der individuellen Genome hilft bei der Detektion homologer Gene. Ziel des Projektes ist die Entwicklung eines Visualisierungs-Tools für multiple Genom-Alignments, welches darüber hinaus die Integration von Gen- und Meta-Annotationen erlaubt. Dabei soll von dem in unserer Gruppe entwickelten Konzeptes des SuperGenomes, welches ein einheitliches Koordinatensystem für alle Genome darstellt, Gebrauch gemacht werden.



Ziele

- Implementierung geeigneter Datenstrukturen
- Realisierung einer effizienten Visualisierung eines multiplen Genom-Alignments
- Schaffung einer interaktiven grafischen Oberfläche
- Integration von Annotationen

Inhalte

- Optimierung von Datenstrukturen zur Speicherung von und Interaktion mit großen Daten
- Java Swing Programmierung/ GUI Entwicklung
- MVC Pattern

Technische Grundlagen

- Java, Swing
 - Eclipse / NetBeans o.ä.
 - Git
 - Scrum
-

Drahtloses Informations- und Aktionssystem

Technische Informatik

Prof. Dr. Wolfgang Rosenstiel Wolfgang Fuhl

Projektbeschreibung

Unser Projekt soll praktische Erfahrungen beim Programmieren von mobilen Eingebetteten Systemen vermitteln. Das Informations-System ist ein kleines Kommunikations-Netzwerk auf der Basis der drahtlosen Bluetooth-Technologie (BT) und könnte beispielsweise ortsfremden Besuchern in öffentlichen Gebäuden den Weg zu ihrem Ziel weisen, z. B. zum Lift, WC, ins Sekretariat, oder behinderten Menschen, z.B. Rollstuhlfahrern den Alltag erleichtern, indem diesen eine Fernsteuerung für die Beleuchtung, die Jalousien etc. in die Hand gegeben wird. Es wäre möglich, in öffentlichen Gebäuden, z. B. im Rathaus, im Museum, in der Uni am Eingang ein Smartphone zu verleihen, oder die entsprechende Software auf dem Smartphone des Besuchers zu installieren.



Ziele

- Einarbeitung in Bluetoothkommunikation.
- Bereitsstellung von Basisfunktionalität:
 - Mikrocontroller: Notfallschalter, Akku-Füllstand Überwachung und Beleuchtungssteuerung.
 - Mobiltelefon: Suche und Auswahl von Kommunikationspartnern, Abfrage der Dienstliste und Ausführung der Dienste mit Visualisierung.
 - PC: Telefonnummernverwaltung, Gebäudeplan und Quiz.
- Spezifizieren eines einheitlichen Kommunikationsprotokolls.
- Entwicklung eines Testprogramms.
- Abschluss des Projekts am 15.07.2015.

Inhalte

- Programmierung für Mikrocontroller (BTNode).
- Programmierung für Android.
- Java Programmierung.
- C Programmierung.
- Kommunikationsprotokoll spezifizieren.
- Bluetooth Kommunikation.
- Softwareentwicklungsmethode Scrum.

Technische Grundlagen

- Mikrocontroller (Sensor/Aktor Knoten):
 - Zielbetriebssystem: NUT/OS
 - Programmierumgebung: Beliebig
 - Programmiersprache: C
 - Sonstiges: Zum Flashen des Mikrocontrollers wird Windows benötigt.
 - Mobiltelefon (Mobiler Knoten):
 - Zielbetriebssystem: Android
 - Programmierumgebung: Beliebig
 - Programmiersprache: Java
 - PC (Server):
 - Zielbetriebssystem: Beliebig
 - Programmierumgebung: Beliebig
 - Programmiersprache: Java
-

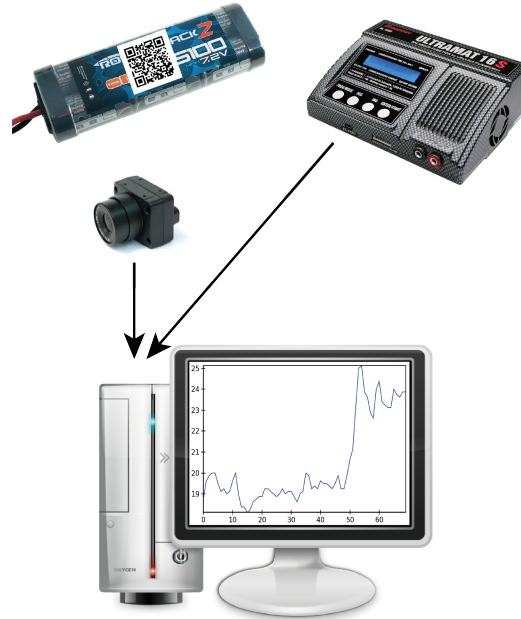
Automatisiertes Powermanagement für Roboterakkus

Arbeitsbereich Kognitive Systeme

Richard Hanten

Projektbeschreibung

Für den Einsatz von mobilen Robotern ist eine zuverlässige Energieversorgung durch Akkus extrem wichtig. Damit die Akkus eine lange Lebensdauer haben, sollten sie stetig überwacht werden. Um die Überwachung nicht aufwendig von Hand protokollieren zu müssen, soll in diesem Projekt eine Anwendung erstellt werden, die automatisch Akkus beim Laden überwacht und die Akkuladezustände übersichtlich visualisiert. Die verwendeten Ladegeräte sind hierfür über USB an einem Hostsystem angeschlossen. Mittels QR-Codes und einer Kamera sollen die Akkus identifiziert werden. Zur Verwaltung der ermittelten Daten wird eine Datenbank verwendet. Die Akkuzustände aus der Datenbank sollen in einer grafischen Oberfläche übersichtlich visualisiert werden.



Ziele

- Softwareseitige Anbindung der Ladegeräte
- Implementierung eines QR-Codesscanners auf Basis einer Kamera
- Erstellung einer automatischen Protokollierung
- Datenbankbasierte Verwaltung
- Grafische Visualisierung ermittelter Daten

Inhalte

- Verwendung externer Hardware
- Treiberprogrammierung
- Anbindung einer Datenbank
- Entwicklung einer intuitiven, grafischen Oberfläche
- Übersichtliche Visualisierung von Messdaten

Technische Grundlagen

- Java und JFreeChart o.ä.
 - MySQL o.ä.
 - Umgebung: Eclipse Luna oder IntelliJ
-

Visualisierung und 3D Rekonstruktion mit Stereo und RGB-D Kameras

Arbeitsbereich Kognitive Systeme

Radouane Ait Jellal, Lixing Jiang

Projektbeschreibung

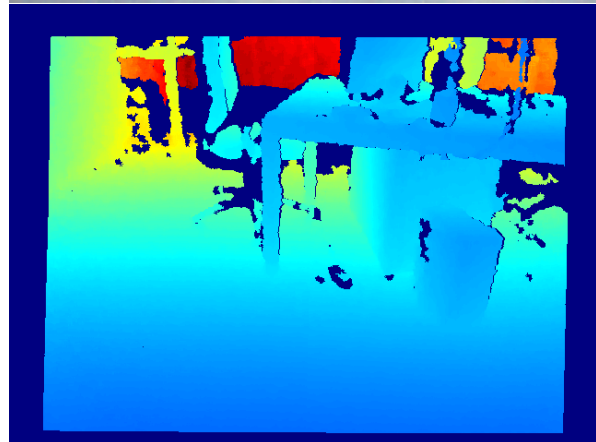
In diesem Programmierprojekt sollen 3D Modelle aus Stereo/RGB-D Bildern erzeugt werden. Dazu sollte das System zwischen Stereo und RGB-D wechseln können. Des Weiteren muss aus dem Stereobildpaar das Tiefenbild erst noch berechnet werden. Am Schluß sollen einige Bilder zur Erzeugung des 3D-Modell ausgewählt werden. Punktwolken sollen für diese Bilder berechnet werden und durch Registrierung dieser Punktwolken soll ein 3D Modell erzeugt werden.

Ziele

- Entwurf einer GUI: Auswahl von Stereo/RGB-D, Einstellung von Parametern usw.
- Integration eines Stereo-Algorithmus.
- Integration einer 3D Rekonstruktion aus Punktwolken.
- Visualisierung (Rohdaten, Tiefe, Punktwolken, 3D Modell).
- Vergleich der beiden Tiefenschätzungs-Methoden: RGB-D und Stereo.

Technische Grundlagen

- C/C++
- API: OpenCV, Point Cloud Library (PCL)
- Ubuntu/Linux



Ein Farbbild (oben) mit dem zugehörigen Tiefenbild (unten). Diese Bilder wurden mit der Microsoft Kinect aufgenommen.