

Rohan Prinja

CONTACT

- **Address:** Room A706, Hostel 13, IIT Bombay, Powai 400076, Mumbai, India
- **Email:** rohanp@cse.iitb.ac.in
- **Homepage:** www.cse.iitb.ac.in/~rohanp
- **Github:** github.com/wenderen

ABOUT

- Pursuing B. Tech with Honors in Computer Science and Engineering
- **Research Interests:** Graphics, Vision and Image Processing, Graph Theory, Combinatorics
- **Other Interests:** Game Development, Web development

SCHOLASTIC ACHIEVEMENTS

- **All India Rank 53** in IIT-JEE 2011 among 500,000 candidates
- **All India Rank 51** in AIEEE 2011 among over 1,200,000 candidates
- Qualified for the **Indian National Physics Olympiad, 2010**
- Qualified for the **Indian National Mathematics Olympiad, 2011**
- Scored in the top 1% in the state for the **Indian National Chemistry Olympiad, 2011**
- **All India Rank 1** in the 5th **National Cyber Olympiad (2006)**
- **All India Rank 28** in the 6th **National Cyber Olympiad (2007)**

RESEARCH EXPERIENCE

• **Augmented MPM for Physically Simulating Burning**

Undergraduate Dissertation

Ongoing

Guide: Prof Parag Chaudhuri

We consider the problem of modifying the material-point method to physically simulate the burning of arbitrary solid objects. The solids are represented by tetrahedral meshes which are generated by DeLaunay mesh tetrahedralization. The evolution of the flame is done by the Navier-Stokes equations for fluid simulation within the Mantaflow solver framework. The deformation of the solid, smoke emission and residue formation are done separately. Each step of the physical simulation is output to a scene file and rendered by a raytracing program like PRMan or POV-Ray.

• **Bounding error-correcting codes**

Research Project

Ongoing

Guide: Prof Ankur Kulkarni

We consider the problem of estimating an asymptotic upper bound on the cardinality of an error-correcting code for strings sent over a deletion channel. To obtain such a bound we restate the problem as one of finding a maximum independent set in a graph in which the vertices represent fixed-length strings over some finite alphabet, and edges connect two vertices if they have a common subsequence after deleting a fixed number of characters from each string. We use a theorem that relates the number of triangles in the graph to the maximum independent set size, so our problem is reduced to counting triangles in this graph given the length of each string and the deletion channel parameter.

SEMINARS

• **GPU Accelerated Compositing in Google Chrome**

Web Platform Group, Samsung Electronics

Summer 2014

Guide: Hyunki Baik, Senior Engineer

Traditionally, browser engines render webpages entirely on the CPU. By using the GPU, we can decrease webpage rendering time by parallelizing rendering and other tasks like Javascript execution. Beginning with an overview of the Chrome graphics stack, this seminar talks about the various ways in which Chrome reduces webpage rendering time by offloading part of the rendering pipeline onto the GPU, as well as the challenges in doing so. These challenges include non-uniform GPU support across devices, GPU-CPU syn-

chronization issues and deciding if certain webpage content is more suitable for software rasterization or hardware-accelerated rasterization. This seminar was presented as part of an internship with the **Web Platform Group** at Samsung Electronics.

- **Augmenting Hand-drawn Animation with 3D Physical Effects**

Spring 2014

Advanced Computer Graphics course

Guides: Prof Parag Chaudhuri, Prof Sharat Chandran

Given a hand-drawn animation of a single human character, can we augment the animation by adding scene elements that interact realistically with the character? In this paper, by matching the animation to a member of a pre-existing motion-capture database, the 2D animation is reconstructed in 3D. Then, the interaction of the scene elements and the character is physically simulated in an animator like Maya or Blender. Finally, the entire frameset projected back into 2D. This paper was presented in a team of 2 as part of an **Advanced Computer Graphics** course.

- **Computational Humour**

Spring 2014

Artificial Intelligence course

Guide: Prof Pushpak Bhattacharya

Computational humour is the problem of generating, identifying and responding to humorous situations. It lies in the AI-complete category of problems, meaning that if it is solved, then computers can be said to be as intelligent as humans. In the seminar we analyse existing theories of humour and their associated mathematical models. We then survey a paper that uses machine learning to classify text fragments as humorous or not, a paper that algorithmically generates knock-knock jokes, and a paper that constructs pairs of graphs in order to generate conversational jokes. This seminar was presented in a team of 3 as part of an **Artificial Intelligence** course

WORK EXPERIENCE

- **Web Platform Group, Mobile Division, Samsung Electronics**

Summer 2014

Software Development Intern

Suwon, Korea

- Understood the Chromium graphics stack and presented a seminar to teammates
- Filed bugs; submitted patches and bugfixes for Chromium's 2D drawing library **Skia**
- Developed a **WebGL** browser game using **three.js** with a hand-written simple game engine that runs at a nearly steady 60 FPS on mobile. The game runs as a webapp on SBrowser and Chrome, and is controlled via the Samsung Wireless Gamepad, and in general any gamepad that supports the W3C-standardized **Gamepad API**
- Awarded a **pre-placement job offer**

- **Chronus Software**

Summer 2013

Software Development Intern

Chennai, India

- **SSH Key Management:** Developed a CLI app in **Ruby** for privileged users (admins, ops team) to efficiently provision, rotate and revoke SSH key permissions for **AWS EC2** instances, using **AWS S3** as the keystore
- **RTalk:** Group chat app in **Rails** with emoticons, clickable links and an option to export chat history as a text file. Users create chat rooms and invite friends by sharing a randomized room URL. Room and chat logs are destroyed when everyone leaves

POSITIONS OF RESPONSIBILITY

- **Speaker, Web and Coding Club, IIT Bombay**

- Conducted two **Python programming workshops** for beginners
- Conducted a **Haskell programming workshop** for beginners

- **Teaching Assistant, Computer Graphics course**

- Developing teaching material for the course, guiding students, grading papers

COURSE PROJECTS

- **Depixelizing Pixel Art**

Fall 2014

- Implementing the research paper 'Depixelizing Pixel Art' in a team of 2 for the Digital Image Processing course project

- The paper describes an algorithm for outputting a vector representation of a pixel art raster image
- **Rendering with Photorealistic Renderman** *Spring 2014*
- Wrote shaders and rendered raytraced scenes using Pixar's rendering software **PRMan**
- **Musicbox Animation** *Autumn 2013*
- Modeled, textured, posed and animated a dancing man in a music box from scratch with **OpenGL**
- Implemented Bezier curves for camera movement and keyframing for the animation
- **MS Paint clone** *Autumn 2013*
- Basic MS Paint clone made with OpenGL with support for floodfill tool and checkered patterns
- Implemented Bresenham's Line Drawing algorithm and other graphics algorithms
- **Other Projects**
- Wrote a **terminal-based file manager** in C++ for Linux, supporting standard file management tasks
- Built an Amazon/eBay/craigslist **clone** in Rails tailored towards services
- Simulated a Rube Goldberg Machine in C++ using Box2D physics engine and OpenGL for rendering
- Implemented **Tonelli-Shanks** algorithm for **elliptic curve cryptography** in MIPS Assembly
- Wrote an **automated theorem prover** to search for proofs of arbitrary formulae using Propositional Calculus
- Implemented 8-point radix-2 Decimation in Frequency **Fast Fourier Transform** on Atlys Spartan 6 FPGA using VHDL
- Used socket programming to design and implement a **minimal Dropbox clone** (distributed file storage)
- Built an interpreter and compiler that worked on **control flow graph** dumps of GCC
- Built a GUI and implemented heuristic game AIs (Minimax and Alpha-beta pruning) for **Chinese Checkers** in Racket

PROGRAMMING AND TECHNICAL SKILLS

- **Knowledgeable about** C, C++, OpenGL, WebGL, JavaScript, Ruby, Rails, Python
- **Basic familiarity with** Rust, Lua, PRMan, SQL, Amazon AWS (S3, IAM)

EXTRA-CURRICULAR ACTIVITIES

- **Contributing to Servo**: Writing patches and bugfixes for Servo, Mozilla's experimental open-source parallelizing browser engine. Servo is written in **Rust**, a memory-safe and fast low-level language
- **Scrobbet**: Wrote a command-line app to extract and send song metadata to **Last.fm**, a artist-database / social-network for music listeners
- **Time Glider**: Webapp that displays an interactive timeline of news articles related to a user's search term. The backend uses **NLTK** for news article summarization and **Apache Solr** for search indexing. Winner of the **second prize** at the **Yahoo! HackU 2013 IIT Bombay** event
- **Train Tracker**: A webapp for train travellers that uses PNR number to return real-time data about the upcoming train station (distance, time left, weather forecast, emergency numbers). Winner of the **Best Technical Hack** award at the **Yahoo! Hack India 2013 Hyderabad** event
- Foreign languages: studied **French** as a second language for 7 years at the school level), and completed one-year courses in **German** (certified by **DAAD**) and **Mandarin** (certified by **Beijing Jiaotong University**)
- Selected to be part of the **Windows Internet of Things** developer program
- Participated in MozBoot India - a program conducted by Mozilla to introduce programmers to working on bugs in large open-source projects
- Built a self-controlled wall-following robot in a team of 3 in **Techfest 2012**, which was selected among the top-32 teams, as well as a remote-controlled **racing robot** in a team of 3 for **Trackmania 2012**, which navigated an obstacle course