CCSC Consortium for Computing Sciences in Colleges

Southeastern Region

29th Annual Southeastern Conference

Student Research Contest

Extended Abstracts

November 6 and 7, 2015
Roanoke College
Salem, Virginia

Table of Contents

GPU Acceleration of SQL Queries on Distributed Systems

Linh Van Nguyen
Hampden-Sydney College, VA
nguyenl16@hsc.edu
Faculty Advisor: Dr. Paul Hemler
Department of Mathematics and Computer Science

Introduction:

Structured Query Language (SQL) is the industry standard language for managing and querying relational databases. In the relational database model, data is organized into related tables. Each table contains rows of data that span the table's columns. In recent years, significant efforts have been put into improving SQL databases' performance due to the explosion of data. As SQL is traditionally executed on the Central Processing Unit (CPU), novel massively parallel frameworks offer an alternative to accelerate SQL operations.

Graphics Processing Units (GPUs), originally built for rendering graphics, have become a major computing resources in recent years, acting as a coprocessors to the CPUs. NVIDIA released the Compute Unified Device Architecture (CUDA) platform in 2006, enabling programmers to harness the massively parallel structure of the GPUs and apply them to general purpose applications. The CUDA platform communicates with the CPU via the "kernel" and has a rich memory hierarchy. Each CUDA kernel is launched in a grid of thread blocks, each has its own shared memory space. Each block contains up to 1024 threads; each thread also has its private memory (registers). All threads are executed in a Single Instruction Multiple Thread (SIMT) fashion, and can access DRAM global memory, read-only constant memory, and texture memory.

Drawing from previous works, we aim to accelerate SQL Select and Join queries using the GPU. The Blue Waters (BW) supercomputer provides the hardware used in this project. Each BW's GPU-enabled node contains one Kepler GK110 (K20X) GPU and a multi-core CPU. Our implementation utilizes specific features of the Kepler architecture. Furthermore, we seek to test the implementation in a cluster setting.

Related Works:

SQL queries have been shown to benefit largely from GPUs acceleration, using parallel primitives such *as map*, *scatter*, or *reduce*. The Virginian database [1][2] approaches the problem differently by using an opcode model, which has certain advantages over using primitives. These advantages are fully discussed in [2]. Each opcode is the identifier for an instruction of a customized virtual machine (VM), similar to SQLite's VM. This customized VM is generated by parsing the abstract syntax tree of a SQL query in several passes [2]. The Virginian framework uses a custom data structure with column-major organization of data to support GPU coalescing. Originally this implementation only supports one-table Select queries. The authors of [3] extend this project to accelerate natural Joins queries. To compute the Cartesian product of the two involved columns, this extension linearizes the nested loop structure to adapt to the three-dimension nature of a CUDA grid. Each thread computes a data point. For this reason, this approach can only join at most three tables. Besides, the Virginian database and this extension only support single GPU systems.

Implementation:
Our main contributions to this project include: expanding the Select query to run on multiple GPUs, support the On clause in the implementation, and using Dynamic Parallelism (DP) to program the Joins query directly into the VM. To run Select query on multiple GPUs, we query the data in a Multiple Program Multiple Data (MPMD) fashion: each node will process its own partition of data on a table and send the result back to the master node. The On Clause is supported similarly to the Where clause as the parser accepts the syntax of the On Clause to generate the corresponding VM.

A Join predicate consists of multiple conditionals that specify the data in the joined tables. Each conditional relates to the data in a particular table. Therefore, the data could be kept independently after evaluating each conditional. We use DP, a feature introduced in Kepler architecture that enables threads to launch kernels, to utilize this characteristic of a Join predicate. All threads that satisfy the first conditional in the predicate will launch a second kernel into the consecutive table and evaluate this table's conditional. The threads processing the last table write the result back to main memory. Currently, we are only supporting two tables.

Future Works:
We intend to further extend this implementation by launching the kernel recursively to support more tables and support other kinds of joins, such as *inner*, *outer*, and *natural*. After that, we intend to compare the performance of this Join implementation to that of [3]. Since the Cartesian product is straightforward to divide across multiple nodes, we also want to test the scalability of both implementations.

References:
1. Bakkum, P. The Virginian Database. Date accessed: September 12, 2015
2. Bakkum, P., and Chakradhar, S. Efficient data management for GPU databases. Tech. rep., NEC Laboratories America, Princeton, NJ.
3. Angstadt, K and HartCourt, E. A Virtual Machine Model for Accelerating Relational Database Joins using a General Purpose GPU. In *Proceedings of the High Performance Computing Symposium*, HPC '15, Alexandria, VA, 2015.

Constructing and Performance Benchmarking of a Parallella cluster

Samuel Sheffield
Faculty Advisor: Dr. Paul Hemler
Department of Mathematics and Computer Science, Hampden-Sydney College, VA

Parallel computing is the required infrastructure for today's High Performance Computing (HPC) needs, where computing requirements are more demanding than any current single-threaded processor can deliver. To improve the overall performance, multiple processors are utilized to simultaneously compute different parts of an overall solution to a computation problem. The improved performance is a function of the number of processors used in the parallel solution, but it is a sublinear improvement.

The two prominent models of parallel computing are shared memory and distributed memory systems. Message Passing Interface (MPI) is the standard for executing programs in the distributed memory model. In this model, each processor has its own private memory resources. The advantages of the distributed memory model is that each processor has the entire memory bandwidth of its local memory without contention from the other processors. The distributed memory model is also scalable, since there is not a limit on the number of processors allowed in the cluster. Processors can add be added or removed for the best performance.

A Beowulf cluster is a distributed system where computers are privately networked to allow shared processing and memory resources. Recently, a company named Adapteva has introduced a single board parallel computer called the Parallella board. These boards provide an inexpensive way to build an educational parallel cluster using MPI. These boards provide an eighteen-core computer, including two ARM Central Processing Units (CPUs) and sixteen-Epiphany core coprocessor. We purchased eight of these boards for the system we built and benchmarked. Our system included one Parallella board referred to as the Desktop model, which served as the master node for the cluster. The other seven boards were the Micro-Server models, which acted a slave nodes.

We benchmarked the Parallella cluster by implementing a parallel program that estimated the value of PI using Monte Carlo method. By increasing the number of iterations, the computed value of PI becomes closer to the actual value of PI found in the MPI library. The runtimes for the parallel version of the algorithm were not very different from a serial implementation of the method provided there were less than 1000 iteration. As the number of iterations grew, discrepancies in performances began to emerge. For example, at one million iterations the serial version ran in about 437 seconds, while the parallel version only ran in about ten seconds.

We will be using this cluster in our new parallel computing course to demonstrate the methods of programming a distributed system. Since parallel computing has become pervasive, we hope that future computer science majors can benefit from this system. The course will introduce a new approach to solving problems by parallelization.

Using PHP and JavaScript in CISE Website Content Management

Phu Quach - Christopher Newport University - Newport News, VA 23606
Faculty Sponsor: Dr. Lynn Lambert

Background Information

Website content management of websites is a dynamic area [1, 2, 3]. Less attention, however, has been given to creating low overhead single web pages that handle quickly changing content (e.g., a website for upcoming STEM summer camps).

This summer, Christopher Newport University's department of Physics, Computer Science and Engineering wanted to improve its outdated CISE website in order to make it more attractive and add more contents for CNU students and computer science's community. The Center for Innovation in Science Education (CISE) uses this website to provide general information about STEM education. The website is not only an effective communication tool to connect and encourage the CNU's STEM education, but it also provides the connection between CNU's STEM and Hampton Road's STEM education. This poster will address this last part, using the website as a way that CNU can connect with the regional STEM community. We employed website development tools to ease how the CISE directors manage the website content and enhance the effectiveness of communication tool.

Problem statement

The main problem that we focused on is figuring out a new way to allow STEM community to post their activities on CNU's CISE website. In the past, when community partners wanted to post activities on our website, they needed to email the CISE director. The director then had to spend time editing and approving each activity's content, manually posting each on CNU's CISE website and manually deleting each after the event had finished. This complicated process makes communication inefficient.

Problem approach

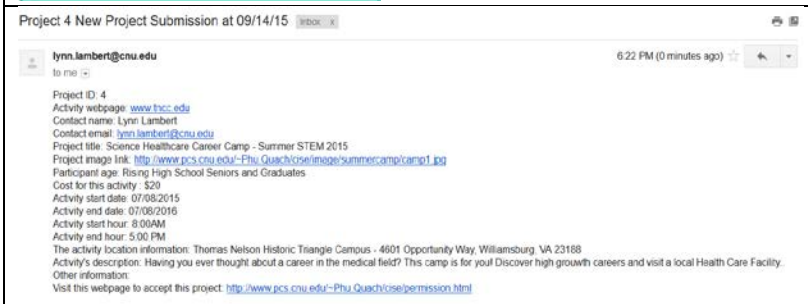
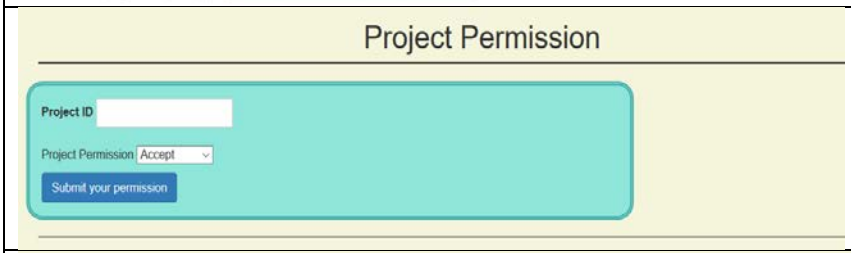
I streamlined the process using PHP and JavaScript. Community STEM users can fill out a short online form. The CISE website then automatically generates the activity's content and sends an email to the director which includes the activity's content. The director clicks a link that has a button to approve, which then automatically posts the activity on CISE website. After this activity ends, it will automatically remove the activity's content from the CISE website.

Result

I used an HTML5 form and PHP to receive data from users, save this activity's information in an XML file, and send email to the CISE director. After the director approves the information with a simple click, the website uses JavaScript to post this information on the CISE website (the Result Reference figure on the next page shows what each of these looks like).

Conclusion

By using PHP, JavaScript, and XML files for storing data, we improve the efficiency of CISE's communication channel, reduce the content management time for the director, and promote the Hampton Road area's STEM education activities.

Result Reference	
	User Interface for project submission
	Sample Email for new project submission
	Permission Page Project ID : to identify the project Project Permission: Accept (information will disappear after activity end date)/Permanent/Not Accept
	Sample Display after approval

References

1. Ana Iglesias et al. "Evaluating the accessibility of three open-source learning content management systems: A comparative study", Computer Applications in Engineering Education, Volume 22, Issue 2, pages 320–328, June 2014.
2. “How Should You Handle Expired Content”, by Stephanie Chang, April 11, 2012, https://moz.com/blog/how-should-you-handle-expired-content.
3. Shah, Rima. "Building a Web Content Management" (2012):14-22.San Diego State University Web. <https://sdsu-dspace.calstate.edu/bitstream/handle/10211.10/3228/Shah_Rima.pdf?sequence=1>.

A Framework for Social Networks Sentiment Analysis

Rob Schultz, Leilei li

Faculty Sponsor: Dr. Gilliean Lee

Lander University

This research is to develop a framework for sentiment analysis posted on social networks, including Twitter. The purpose is to be able to connect to these social networks and to collect public/private data from them to store for analysis. To do this we have created libraries to connect, store, and retrieve the data from social networks. We store the data into a No-SQL database which can then be used to analyze. We use the data to compute sentiments based on certain criteria, such as keywords, location and time. With the sentiment we are able to see how people feel about certain things, such as products, the economy, politics, etc.

The framework is built as two libraries; the first library created is the data. The problems we ran into are where we wanted to get our data from and how. Some sites have different ways of being able to retrieve their data and what we wanted was one with public availability and popularity, and we chose Twitter. The information we retrieved was in JSON format which we stored into a document database called MongoDB, a No-SQL database. Two different APIs were used in the downloading, storing and retrieval of tweets which were, Twitter4j and MongoDB driver.

For the actual sentiment analysis a combination of three different libraries of English words were used. They were Affective Words for English Words (ANEW), SentiWordNet, and another list created by Alex Davies. Each individual list of words has their own rating system and along with those rating systems we created each had their own individual algorithm on calculating a Tweets sentiment.

For presentation of the data we were able to retrieve we used a visual representation with graphs and plot points that would show in relation to the sentiment score on the chart. Depending on the library used for the score that is represented each chart would look separate from the other with each being color coded and plotted according to their rank on the scale of the library used. The demo made could make a handful of queries to show the potential that the libraries could achieve some examples of the queries are: looking up specific words, looking up phrases, the ability to look at states of the United States or the ability to look into some other countries using their full name or country code, and by looking up certain hashtags that could be trending.

Team Development of a Model View Controller Software in the Unity 3D Engine

Nicholas Blum, James Morrow, Sean Stamm
Computer Science Department
University of North Carolina at Asheville
One University Heights
Asheville, NC 28804
nblum@unca.edu, jmorrow1@unca.edu, sstamm@unca.edu
Faculty mentor: Dr. Adam Whitley (awhitley@unca.edu)

ABSTRACT

This project implements a board game, King of Tokyo, using object oriented abstractions and MVC (Model View Controller) architecture. Group software development is often lacking in undergraduate computer science curriculums, and this project strives to address that through teaching software design skills, team skills, and good coding practices. The game is designed and constructed with the software platform Unity, self-described as "a flexible and powerful development platform for creating multi-platform 3D and 2D games and interactive experiences" [1] . Unity3D is not designed with object oriented data structures and MVC in mind, which makes this project not only more challenging to implement, but more unique in its concept and execution.

SOFTWARE DESIGN

The software's MVC architecture provides a separation between the game state (model) and the graphical representation of the game (view), which allows us to work on both modules simultaneously and easily make design changes. The MVC architecture also gives us the freedom to change the view at any point without sacrificing the work we did on the model.

In terms of object oriented design, the source code makes use of classes, structs, and interfaces. The key classes, paralleling the MVC idea, are the GameState (model), the vUI (view), and the Controller. An interface, called IGameState, provides a contract between the Controller and GameState classes. Additionally, we have a class for each discrete game mechanic, as well as classes for their visual representations in the view.

UNITY AND MVC

The Unity game engine is powerful, allows for quick testing, and is well documented with resources for learning and working with it. The language chosen for the project, C#, is one of the major languages used for Unity development, and has an extensive library of supporting functions. In addition, there is a much larger community which can answer questions relevant to game development with Unity using C#, than for Unity's other scripting languages, Javascript and Unityscript [2].

Unity follows the scheme of a game object as a pure aggregation [3], in that the objects or entities in the game world are collections of different components, such as Renderers, Scripts, and Controllers. Combining those components into discrete objects allow the developer to build complex interworking objects out of simple reusable pieces. Thus, Unity programs lack the kind of hierarchical class structure present in our model. Using a controller with Unity creates a powerful bridge between Unity and the game model: the model can function with a traditional

class hierarchy, inheritance, and data structures while Unity can graphically represent the game model. The controller allows the aggregate Unity objects to query and update the model. Both the model and the view systems can operate independently of each other, while the controller negotiates the interface between them.

TEAM DEVELOPMENT

A large software development project like this one requires best practices in documentation, team communication, and decoupled design, to allow multiple team members to work simultaneously and cohesively. These skills, so often missing in the context of smaller assignments, are invaluable in large-scale, real-world software development. This project develops these skills via well-documented code, strict version control, and a well-communicated, iterative design.

CONCLUSION

This collaborative project enhances the undergraduate computer science learning experience by introducing the concept of large, team-based software development. It also combines Unity with traditional object oriented programming through MVC, which allows for decoupled design, making it possible for team members to build separate portions of the software that interface with the work of others on the team.

REFERENCES:

[1]Unity Technologies, *Unity* , 2015, https://unity3d.com/unity, retrieved Sept. 24th 2015
[2] La Voie, M., "Is there a performance difference between Unity's Javascript and C#?", *Unity* , 2009, http://answers.unity3d.com/questions/7567/is-there-a-performance-difference-between-unitys-j.html , retrieved Sept. 21st 2015
[3] West, M., "Evolve Your Hierarchy", *Cowboy Programming* , 2007, http://cowboyprogramming.com/2007/01/05/evolve-your-heirachy/, retrieved Sept 21st, 2015.

Optical Musical Recognition using Hidden Markov Models

Natalie Wilkinson
Roanoke College
Faculty Sponsor: Mr. Scotty Smith, Computer Science

1 Introduction

Optical Character Recognition software (OCR) is a commonplace software that converts images of text into a form that can be interacted with, such as a text file or a pdf. A similar software, Optical Music Recognition software (OMR) is a software that has the ability to read sheet music, but it has yet to become as widespread. OMRs have the ability to create a sound file from a scanned sheet of music, without requiring a trained musician to play the piece. Thus this type of software can be incredibly useful for students learning to read sheet music. This type of software can also be very useful as it can convert scanned music into a digital form that directly corresponds to music instead of having to save scanned music as an image. This can help with the libraries of sheet music that currently exist.

Current OMRs already exist, however there are many different ways that they can be implemented, and each implementation has different strengths and weaknesses [4]. The goal of this research was to investigate a particular algorithm that is used for the identification process of OMRs called Hidden Markov Models (HMM). Previous implementations of HMMs in OCRs for handwriting have been shown to be very successful [1]. Since sheet music seems to be of the same difficulty as handwritten text, it was assumed that HMMs in OMRs would be just as successful. However, previous implementations of OMRs using HMMs have been unsatisfactory compared to the other algorithms currently in use. We attempt to investigate if an increase in training could improve the accuracy of the algorithm and hopefully provide a more successful algorithm for OMRs.

2 Research Methods

To begin the research, it was necessary to create a simplistic OMR in which to implement the HMM. The two major components of an OMR are pre-processing and segmentation. Once the simplistic OMR was built it was then used to create the HMMs and the training data for the HMMs.

The pre-processing stage is where an image file is converted into the proper form in which to segment the sheet music. This stage is not always needed, but since it eases the complexity of the segmentation we implemented it [3]. This involved removing the staff lines of a piece of sheet music and then aligning the various musical staves into one long image.

The second stage of an OMR is the segmentation stage. This involved creating various histograms that represented the number of black pixels in the rows and columns of the image, and then splicing the image accordingly. Single symbols could be extracted upon a single pass of the histogram. Beams required multiple passes to segment into the various individual symbols [2].

Once the pre-processing and segmentation stages of the OMR was built, we created training data using the software. This involved acquiring images of sheet music that had a sufficient diversity of musical symbols. Once the pages of sheet music were converted into musical symbols they were then sorted by unique musical symbol, such as flats and quarter notes.

Training the HMM required creating a model for each unique musical symbol. To do so we created feature vectors for each symbol image [3] and then used the Baum-Welch algorithm to create the matrices that make up the HMM. This was done by running the Baum-Welch algorithm repeatedly for each unique musical symbol until one model was created for the given musical symbol.

3 On Going Research

The goal of the research was to accurately identify a symbol that was extracted from a piece of sheet music. In this regard we were successful. However, a full OMR software not only identifies symbols, but is also able to correctly interpret an entire piece of music. Thus future work will include implementing this feature into the current software that has been written. To do so requires creating a digital language that can accurately represent any given piece of sheet music, and using that digital language to convert the sheet music into a sound file, effectively playing the given piece of music.

We would also like the program to be able to be given a greater variety of sheet music as well. Currently it is assumed that the staff lines are completely horizontal with no deviation, and it is assumed that the given symbols on the are not blurred in any way. Thus future work will include making the software more robust, in order to be able to handle scanned sheet music that may be titled or blurred, as normally results from scanning.

Currently the number of symbols that the program can accurately identify is also limited. The current number of symbols that can be identified is sufficient for most pieces of sheet music but with older pieces or more complex pieces of music, the software will fail since they may contain symbols that the software does not know exist. Thus more hidden markov models should be made to encompass all possible symbols that could be encountered.

References

[1] Mou-Yen Chen, A. Kundu, and Jian Zhou. Off-line handwritten word recognition using a hidden markov model type stochastic network. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 16(5):481{496, May 1994.

[2] S. Marinai and P. Nesi. Projection based segmentation of musical sheets. In Document Analysis and Recognition, 1999. ICDAR '99. Proceedings of the Fifth International Conference on, pages 515{518, Sep 1999.

[3] Laurent Pugin. Optical music recognition of early typographic prints using hidden markov models. In Proceedings of the 7th International Conference on Music Information Retrieval, Victoria (BC), Canada, October 8-12 2006. http://ismir2006.ismir.net/PAPERS/ISMIR06152_Paper.pdf.

[4] A. Rebelo, G. Capela, and JaimeS. Cardoso. Optical recognition of music symbols. International Journal on Document Analysis and Recognition (IJDAR), 13(1):19{31, 2010.

Humanitarian Open Source Software

Sean Workman
Christopher Newport University
Newport News, VA
Sean.workman.14@cnu.edu
Faculty Sponsor: Dr. Lynn Lambert

The goal of this project is to create an environment where students can contribute to humanitarian open source applications, inside or outside class, without doing excessive research. We created a website, http://www.pcs.cnu.edu/~sean.workman/Open-Source/opensource.html, which contains a step by step guide for programmers to contribute to Open Source.

Background Open source software is software where the source code is available to everyone. Open source is not a new concept; for example Linux and Apache have been used for years. Recently, however, some colleges and universities are beginning to add it to their curriculum. Open Source software is helpful on an individual level because it allows people to change the source code to make the software better suit their needs. On a larger scale, such as non-profit organizations and other companies, the organization is able to post bugs online for developers to fix making maintaining the software little to no cost for the organization. This is especially helpful for non-profit humanitarian organizations because they do not have to spend money on maintaining the software that could be going to helping people. Computer Science programs in colleges and universities would like to use open source, both to allow students to work with large, real projects and to allow students to combine their major with an ever increasing service focus at universities. For students, the reasons for contributing to humanitarian open source can be both selfish and selfless, it can be selfish for personal gain because each contribution is tracked showing exactly what was done which can be seen by anyone allowing employers to see exactly what your skill set is, it can be selfless by helping others because the software is used to help the less fortunate around the world.

Rationale One problem with contributing to open source is the overwhelming hurdles for entry. Several researchers [1, 2] have addressed this in the classroom. Our goal was to make a general "how to" document for students who know how to program and want to contribute code in or out of a classroom. To achieve this, I actually contributed to three open source projects, including one humanitarian one, Mifos X.

Approach Our initial attempt at open source contribution was Sahana Eden, a humanitarian open source application that provides solutions for disaster management, development, and environmental management, that was used, among many disasters, by Red Cross during the Haiti 2010 Earthquake Response [3]. Although I found bugs to fix, I could not find a contact to answer questions despite following their IRC and email list (IRC: http://eden.sahanafoundation.org/wiki/Chat, Email list: http://eden.sahanafoundation.org/wiki/MailingList). An evaluation rubric for open source [4] indicates that a higher response level was critical, so we examined OpenHatch next. OpenHatch (openhatch.org) proved to be a good point of entry for finding bugs. Eventually, through OpenHatch and the HFOSS community, we settled on Mifos X, a humanitarian open source software that provides financial inclusion for the world's poor and unbanked.

Along the way, I documented what was successful and what tools were needed, made videos and instructions about the tools, and spoke to open source contributors.

Lessons Learned Upon starting this project, there were a few clear issues that most students would encounter; the size of the projects is much larger than any project that they may have worked on in a class; the code is written by hundreds of other people whereas in class students are usually working with their own code; and there is not always enough documentation to figure out what is going on. Throughout this project more

issues arouse such as the amount of research that is required to figure out which outside libraries are used and the code methods and syntax that goes along with them. As the project progressed it became even more apparent how necessary an environment would be for students to effectively contribute to humanitarian open source because almost every college student has very little free time between classes, clubs, sports, and other activities making researching the contribution processes nearly impossible.

Results The result of this project was the creation of a step-by-step document that provides almost everything that is needed to get started contributing to humanitarian open source projects (http://www.pcs.cnu.edu/~sean.workman/Open-Source/opensource.html). A student should be able to sit down, follow the instructions, start contributing, and submit the contribution provided that they are willing to put some time into it, but not nearly as much time as it would be if they started from scratch.

Future Work The next step is to ask students to work with the document with me watching so that I can modify the document where necessary. This semester, two students are doing that. The research and improvement of the document will continue in efforts to make humanitarian open source easier to approach and contribute to.

References

[1] "Revitalizing computing education through free and open source software for humanity" Morelli, Ralph et.al. Communications of the ACM Volume 52 Issue 8, August 2009 Pages 67-75.

[2] "Free and open source software in computing education" Jacobs, Stephen; Kussmaul, Clif; Sabin, Mihaela. Proceeding SIGITE '11 Proceedings of the 2011 conference on Information technology education Pages 41-42.

[3] Tressel, Pat; Boon, Fran; Kohli, Shikhar; Koenig, Dominic; Lopez, Belinda; Lev, Eli; Howden, Michael; Goldenberg, Anne. "Who Uses Sahana Eden?" Sahana Eden. Accessed September 24, 2015.

[4] Ellis, Heidi; Purcell, Michelle; Hislop, Gregory. “An Approach for evaluating FOSS projects for student participation, SIGCSE ’12 Proceedings of the 43rd ACM technical symposium on Computer Science Education, pages 415-420.

Automating Database Creation through the use of Information Extraction

Andrew Safigan

Faculty Advisor: Dr. Chris Healy

Computer Science Department, Furman University

Usually, databases are designed and initialized by an expert and/or a team of professionals. This is a long and resource intensive process, which is a burden on smaller projects such as research conducted by small teams or individuals. In some cases, the data that needs to be analyzed is an existing resource like the 2000 US census. There are a large amount of existing resources like old physical almanacs, which can be digitized, and reports that have been published in plain text, Word documents, PDFs, or HTML documents. The information found in these resources could be used to make a database. One could manually design a database that fits the form of the resource and then copy the information in the resource to the database by hand, but it would be easier if this could be done automatically by a computer. This paper proposes a method for a single user to create a database from an existing resource in a relatively short amount of time by automating large parts of the Database Creation process. The focus will be on turning existing resources of hierarchically organized human-readable text into relational databases through the use of Information Extraction algorithms. This paper will consider the problem of Database Creation, the issued involved, and how they might be resolved automatically.

The problem of Database Creation is simplified by only considering Hierarchically Organized Human-Readable Text (HOHRT) resources. HOHRT covers a large range of resources from large reports like the US census to websites like Dictionary.com. In fact, a dictionary is a great example of a HOHRT. The basic structure of a dictionary is a word followed by a part of speech followed by a definitions followed by the next word. This example illustrates the structure of HOHRT's; a head followed by information related to the head followed by a new head. The information related to the head doesn't have to be simple, it can also take the form of a HOHRT. This structure is much like a dictionary where there isn't just one part of speech and one definition for a word, but multiple parts of speech and multiple definitions for each part of speech. The resources used to test the software system proposed in this paper include: Golden Gate Weather's report on 1981-2010 United State Monthly Climate Normals[1]; the MLA's 2006 report on foreign language enrollments in US colleges and universities[2]; and hockey play-by-plays published by the NHL in the form of HTML web pages, for example [3].

Some of the issues involved when creating a relational database include: identifying fields, organizing fields into tables, and establishing relationships between tables. Fields are different bits of information, for instance three different fields in a dictionary are word, part of speech, and definition. To identify all the individual instances of fields in the HOHRT, first, the user provides a list of field names and then labels some instances of each field in a sample from the text resource, for instance the first page or two. An Information Extraction algorithm is then used to learn the form of each field from the sample and identify all instances of the field in the full resource. The Information Extraction algorithm can identify individual instances by looking at the formatting of the document and the text surrounding the field. For example, it could identify all the defined words in a dictionary by finding words that start on a new line and have an colon following them. The Information Extraction algorithm used should have high precision and recall on structured text. Precision is the percentage of the identified field instances that are correct. Recall is the percentage of field instances present in the resource that were identified. High precision and recall are need to insure both correctness and completeness in the data put into the final database. In this project, the Information Extraction algorithm called STALKER[4] was used. The fields are then organized into tables. To organize the fields into tables, a table is created for every field. Relationships are then created between these tables based on the hierarchical structure found in the resource. The database is then populated with the information that was extracted from the resource. At this point, the database is in Third Normal form, which is the standard for a quality database, but it has too many unnecessary

tables. To simplify the database, related tables that could be a single table are identified and combined. Once it has been simplified, the database should still be in Third Normal form.

With the software system described, a user was able to create a database from the climate and MLA resources within a 4-5 hours each. This included the time it took to gather the resource from the websites they were published on. To check for inaccuracies in databases generated by this system, the MLA resource will be used. This resource has been published in three different forms: by state, language, institution; by state, institution, language; and by language, state, institution. The system will then be used to create three different databases starting from each of these three publications. We can then compare the information in the databases automatically to find any irregularities within the databases. A successful system would be a system in which irregularities only come from those found in the original publications.

References:

[1] *United States Climate Normals*. URL: http://ggweather.com/normals/monthly.htm

[2] (2008, March 27). *Enrollments in Languages Other Than English in United States Institutions of Higher Eduation, Fall 2006*. URL: http://www.mla.org/2006_flenrollmentsurvey

[3] URL: http://www.nhl.com/scores/htmlreports/20132014/PL020001.HTM

[4] I. Muslea, S. Minton and C. Knoblock, "STALKER: Learning Extraction Rules for Semistructured, Web-based Information Sources," Univ. of Southern California, Los Angeles, CA, AAAI Tech. Report WS-98-14, 1998.

URL: http://www.aaai.org/Papers/Workshops/1998/WS-98-14/WS98-14-011.pdf

Using EEG Data to Command Technology

Brent Hall

University of North Georgia

Faculty Supervisor: Dr. Bryson Payne

Abstract:

Interaction with machines and artificial intelligence has expanded greatly in recent years. Researchers have shown that people can interact with machines through interfaces that include EEG scans and facial recognition. The focus of this paper is to further the existing research on EEGs by demonstrating the possibilities and advantages of using an Emotiv EPOC EEG headset to receive cognitive signals in the form of raw sensor data. In order to do this, a subject must learn the proper techniques to train their thoughts and brainwave/EEG-emission patterns by eliminating distractions. With effective training, a researcher or research subject is able to give a command by simply thinking about it and sending the correct signal via the EPOC headset. Once received, our program converts these signals into readable commands. This program is in Java and recognizes and interprets the EPOC headset's signals; we are extending the program to translate these signals into commands that can be displayed on the screen or sent directly to devices such as a robotic toy or small UAS (unmanned aerial system) drone. With this research, scientists have the potential to control and operate machines with their minds.

Rapid and Interactive Statistical Analysis of English Words Using a Computer Cluster

Jim Mouer

Faculty Advisor: Chris Healy

Furman University

This project was inspired by the letters game section of the British game show *Countdown,* in which contestants compete to find a longer word than their opponent using only the nine-letter combination provided to them. The goal of this project was to find possible winning solutions and to determine which of these solutions appeared more frequently than others. In order to accomplish this goal, we had two major components to create: all possible letter combinations and a program to determine which words could be formed by these combinations.

For the combinations, we needed a way to create all combinations of nine letters containing three, four, or five vowels, which totaled 13,116,026 unique combinations[1]. To generate these combinations, we individually generated the possible combinations for three, four, and five vowels and six, five, and four consonants, and took the Cartesian product of each respective pair.

The creation of the program to determine words that could be formed was much more difficult than creating the combinations. We built a cluster computer using 32 Raspberry Pis and building the MPICH2 library on them from source for use in a parallel C program. We broke the list of combinations into 32 equal portions of approximately 400,000 combinations each to distribute to each node in the cluster. The parallel program compared each combination to each word in a custom-built dictionary[2] built from GNU Aspell and word data from actual rounds of *Countdown*[3] which contained approximately 128,000 words, and determined if the combination had enough of each letter to form the word. The program output included the number of matches, the number of matches of the longest possible length for that combination, and all words of the longest possible length for that combination.

For analysis of the output data, we calculated the probabilities of creating each combination based on the frequency of the letters appearing in actual rounds of Countdown, adjusting for the number of permutations for each combination. The average combination had a probability of appearing of 7.62×10^{-8}. The most common was "AEIODNRST" with a probability of 4.00×10^{-5}, and the least common was "UUUQQQQQ" with 7.80×10^{-16}. We also compiled a list of all the words that were included in the best solution, along with the number of times each word appeared and the sum of permutation-adjusted probabilities of the combinations for which

[1] http://cs.furman.edu/~jmouer/master.txt
[2] http://cs.furman.edu/~chealy/words2.txt
[3] http://wiki.apterous.org/Category:Episodes

that word was a solution. As would be expected, each nine-letter word only appeared once, since each combination was unique. For 28,776 different nine-letter words were the best solutions for 26,817 combinations. The shorter solutions yielded more interesting results, however. Overall, the word “iiwi” (correctly written “‘i’iwi”) topped the chart as the solution for 62,088 combinations, followed by “juju”, “wuzu”, and “wudu”. These words were solutions for large blocks of combinations with several letters that do not usually appear together in words. Out of all combinations, 3,714 could not be used to form even a two-letter word. For further application, we filtered out combinations that could not feasibly appear in a round of Countdown by finding the maximum number of each letter that ever appeared in a single combination on the show. Using this data, we removed combinations from the list that had too many of any single letter. This filtering removed less than 10% of all combinations, most of which had small solutions.

SELF-MONITORING INNOVATIONS (SMI) USABILITY STUDY

Samuel Hill
Furman University, Greenville SC, 29613
shill2@furman.edu
Faculty Advisor: Dr. Andrea Tartaro

Obesity and eating disorders are major health concerns whose effects have been shown to diminish with weight management interventions ([1], [2]). Of current intervention techniques, self-monitoring (recording consumption, weight, and activity) has been identified as a key feature in the treatment of obesity and eating disorders ([3], [4]). However despite the effectiveness of this technique – and that increased consistency of self-monitoring increases likelihood of weight loss – individuals demonstrate low levels of adherence to self-monitoring ([5]). A possible reason for low adherence is self-monitoring's most common form, the food log, which involves recording the amount of foods consumed in standard measuring units including cups, grams, ounces, servings, or points (e.g., Weight Watchers). These traditional self-monitoring measurement units are inconsistent with the recently revised USDA nutrition guidelines, which are meant to facilitate adherence to the guidelines and to simplify diet ([6]). The new plate-based diet recommends that each meal (or "plate") include various amounts of different food groups based on proportions or percentages of the total "plate". However, although the measurements aren't consistent with the new guidelines one success of the current food logging systems is their use of technology which puts them literally in the palm of the users hand and makes them much more accessible ([7]). Given that self-monitoring is key to implementing behavioural change and that these new guidelines are easier to follow, a plate-based self-monitoring system could address the problems of adherence to both self-monitoring and nutrition guidelines.

Although the new USDA guidelines are simplified and some research suggests people can provide reasonable estimates for portion sizes ([8], [9]), there are not any studies on estimating proportion each food group on a plate. There is also little research on whether this plate-based diet improves adherence self-monitoring. Though there has been some preliminary research on the use of electronic self-monitoring, further research on the role of internet-based self-monitoring adherence and effectiveness is needed ([10]).

Due to the potential of an improved self-monitoring system, we have developed 3 web application systems and begun testing to verify that they are all usable. The first system is a traditional food log method wherein users have to break down their meal into food and give an estimated caloric content and fat. This method uses a table where each row of the table is a food item, and the food description input can also be used to search a list of about 5000 common household foods from the USDA (Figure 1). The second two systems are based off of the USDA MyPlate method of monitoring which recommends that each meal (or plate) include the consumption of certain amounts of different food groups based on proportions or percentages of a circular plate (i.e., 50% vegetables/fruits, 25% whole grains, and 25% lean protein). The first plate based monitoring method uses a pie chart (taken care of by an API called highcharts) to let the user show what portion of the meal is taken up by each of the 5 food groups – dairy, protein, grain, vegetable, and fruit (Figure 2). The 'quick' method uses the same pie-chart method as the regular plate method, but only to records the portion of fruit and vegetables (together) on a plate. As well as each method, to have uniformity for a long term study there will be little to no variation in the usability of the various systems – every system asks for some basic information; date and time, setting, designation of whether this is a snack or meal, and the number of glasses of water consumed. The layout of these forms is all the same amongst the three systems and they utilize a familiar minimalist theme from Bootstrap 3 – a widely used framework for designing websites ([11]).

To test these systems we took pairs of participants out to lunch or dinner to use the systems and let us know how usable each is. Each participant met with a friend and at least one research assistant. This is to avoid uncomfortable situations with one person eating while another observes, however we still want the meals to be as independent as possible while having two participants at a time. This means it is important that the participants do not discuss what they are going to order or hear each other order and that they do not share food, so as to not be influenced by each other. To do this, only counter service restaurants were used and the research assistant went up to the counter with one of the participants at a time to order while the other waited for the pair to get back. Once the food arrived, the research assistant took a picture of and weighed each meal. When they were done eating the research assistant took a picture of and weighed any leftovers. The research assistant also made note of anything removed from or added to each meal. All of this information was then compiled into composite pictures, which were used later for judging the accuracy of each meal input. After finishing eating, each of the participants used the website to record what they had eaten individually. During the use of the system the interactions on screen were recorded – the recording was used for timing the use of each system – and after using the system the 'post-use' interview audio was recorded. The questions in this interview asked what was their overall impression, if there was anything they liked or disliked, what they found difficult or easy to use, what if any changes would they make to the system, and whether or not they would use this system regularly or recommend it to a friend. After collecting this data, two researchers graded the composites of each meal by entering each into each system as if they were the participants. This data was used to compare the accuracy of each participants input.

The timing analysis showed that traditional is the least time efficient system taking about 3 times longer than plate and 6 times longer than quick. The interviews revealed that about 25% of participants found it challenging to use the traditional method in comparison to 0% of participants using both the plate and quick systems. The only major complaints about the systems were directed at the traditional method, mostly because despite having about 5000 food items the participants wanted – and were accustomed to – more options. In terms of accuracy, there were some obvious discrepancies

in what to enter especially with “stacked” food (sandwich, soup…), mistaken groups (potato – grain or veggie/misclassification of plant items), mistaken food items (not knowing specific types of meat etc.). These discrepancies made us think that the meals need to be broken down and analyzed thoroughly to determine what the individual parts are. Overall, it seems that the plate-based systems we developed were the most usable but we still have one major question to answer and that is how accurate our systems are.

Our system of grading meals using composites should be a viable way of estimating the actual contents of each meal ([8], [9]). However, breaking down each meal into its parts and doing some research to figure out what the true contents of the pictured meal is a lot of work. To properly estimate the contents of each meal you must know what the food items are as well as the associated calories, fat, and food group of each. Thankfully, services such as Mechanical Turk allow us to send out a survey to users so that they can grade the meals using our guidelines (what is what food group, etc.) and we can establish an appropriate baseline for each meal. These surveys will allow us to see if the systems bias a users output and if they facilitate nutrition guideline adherence.

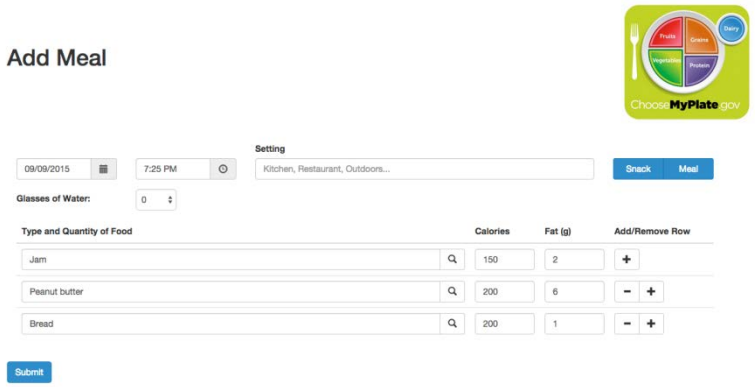


Figure 1: Traditional System

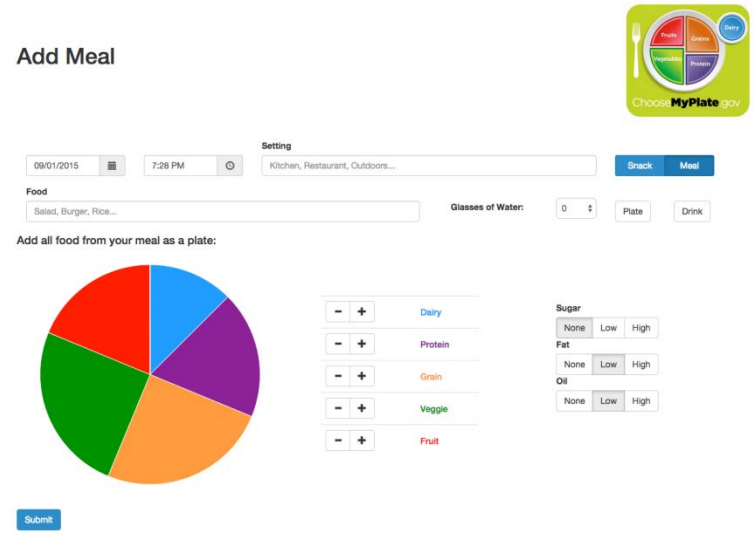


Figure 2: Plate-based System

References:

[1] Ogden, C.L., Carroll, M.D., Curtin, L.R., McDowell, M.A., Tabak, C.J., & Flegal, K.M. (2006). Prevalence of overweight and obesity in the United States, 1999-2004. Journal of American Medical Association, 295(13), 1549-1555.
[2] Haines, J. & Neumark-Sztainer, D. (2006). Prevention of obesity and eating disorders: A consideration of shared risk factors. Health Education Research: Theory and Practice, 21(6), 770-782.
[3] Raymond C. Baker, Daniel S. Kirschenbaum, Self-monitoring may be necessary for successful weight control, Behavior Therapy, Volume 24, Issue 3, Summer 1993, Pages 377-394, ISSN 0005-7894, http://dx.doi.org/10.1016/S0005-7894(05)80212-6.
(http://www.sciencedirect.com/science/article/pii/S0005789405802126)
[4] Wilson, G.T., & Vituosek, K.M. (1999). Self-monitoring in the assessment of eating disorders. Psychological Assessment, 11(4), 480-489.

[5] Boutelle, K. N. and Kirschenbaum, D. S. (1998), Further Support for Consistent Self-Monitoring as a Vital Component of Successful Weight Control. Obesity Research, 6: 219–224. doi: 10.1002/j.1550-8528.1998.tb00340.x (http://onlinelibrary.wiley.com/doi/10.1002/j.1550-8528.1998.tb00340.x/abstract)
[6] First Lady Michelle Obama at the announcement in Washington, D.C. on June 2, 2011
[7] Cushing, C.C., Jensen, C.D., & Steele, R.G. (2010). An evaluation of a personal electronic device to enhance self-monitoring adherence in a pedriatric weight management program using a multiple baseline design. Journal of Pediatric Psychology, 36(3), 301-307.
[8] Validation of a Method for the Estimation of Food Portion Size. Fabrizio Faggiano, Paolo Vineis, Daniela Cravanzola, Paola Pisani, Graziella Xompero, Elio Riboli and Rudolf Kaaks. Epidemiology, Vol. 3, No. 4 (Jul., 1992), pp. 379-382. Published by: Lippincott Williams & Wilkins. Article Stable URL: http://www.jstor.org/stable/3702742
[9] Donald A. Williamson, H.Raymond Allen, Pamela Davis Martin, Anthony J. Alfonso, Bonnie Gerald, Alice Hunt, Comparison of digital photography to weighed and visual estimation of portion sizes, Journal of the American Dietetic Association, Volume 103, Issue 9, September 2003, Pages 1139-1145, ISSN 0002-8223.
[10] Burke, L.E., Wang, J., & Sevick, M.A. (2011). Self-monitoring in weight loss: A systemic review of the literature. Journal of American Dietetic Association, 111, 92-102.
[11] "Bootstrap · The World's Most Popular Mobile-first and Responsive Front-end Framework." Bootstrap · The World's Most Popular Mobile-first and Responsive Front-end Framework. N.p., n.d. Web. 02 June 2014. http://getbootstrap.com/.

Measuring Attention and Fatigue in Exergames

Derek LaFever
Faculty Advisor: Dr. Durell Bouchard
Computer Science Department, Roanoke College

Imagine a video game that adapts to your play style and decision making. A game that not only looks at in-game responses, but also takes into account physical demeanor to determine the appropriate level of adaptation. We call this an agile game. In game responses can be tracked through decisions from the player involving paths they take in a level, what items they decide to use, etc. While the in-game responses are interesting, our goal is to determine the physical demeanor of the player at any given time. In particular, if we knew when the player was engaged during the game, we could better decide when to change gameplay to increase or decrease the level of engagement for the player. In order to determine engagement, we use the definition from Brown and Cairns [1] where engagement is a step towards total immersion in a game and to become engaged, time, effort and attention must be invested into the game.

The first two requirements are not hard to achieve: design a game that requires some amount of time to play and is challenging. Making a game challenging can be achieved with dynamic difficulty and make the game take some finite amount of time. The third requirement, attention, is the more difficult element to control. Of course, even if we can capture the attention of the player, we needed some way of measuring attention. So we created an exergame to have a controlled environment where we can more easily capture and measure attention.

Our exergame is largely modeled after the popular smart phone game Fruit Ninja because it is easy to learn for a wide skill range of players and requires constant focus to play. Fruit Ninja is a game where fruit spawn in waves from the bottom of the screen periodically and the objective is to 'cut' as many fruit as possible by swiping the screen. Some waves of fruit also contain bombs that the player wants to avoid so he or she does not lose points or the game. Our experiment is based on motion, however, so we adapted the game to players having to move their hands in a swiping motion to 'cut' the fruit instead of a touch screen. This is motion-control aspect of the game is achieved using the Microsoft Kinect sensor to track the body.

The main factor to consider when designing the exergame was figuring out a way to keep the player focused on the game throughout the experiment. This is achieved by having a scoring system and dynamic difficulty. The scoring system is meant to promote players to do their best by trying to reach some target score, promoting focus on the game as opposed to other external events. The dynamic difficulty is designed to keep every player at the highest skill level they can handle for the game. We theorize that the amount of attention required to play the game is dependent on the number of entities (objects pertaining to the game) that are on the screen. The idea is the more entities, the more the player is visually and mentally taxed to perform well, so they have less capacity to notice other non-related entities that appear in or out of the exergame. It is important to note that the dynamic difficulty changes at smaller increments once it has reached a plateau, a state of which the player is preforming about the same for each wave.

Once the game design was in place, we needed a way to distract the player while playing to measure for attention. If the player gets distracted from what we call an unexpected event in or out of game, they were most likely not paying enough attention to the game, and thus a possible indication of their level of attention to the game. Each participant was exposed to five unexpected events, both in and out of the virtual environment. These events, by design, were out of context from the objective of the game so as to test the level of attention for each player. Examples of the unexpected events include a gorilla image which fades in and out to test attention, emulating Simons and Chabris [3] study on 'inattentional blindness'. The next two tests focused on attempting to invoke a physical reaction out

of participant. The first one is a projectile that is fired at the participant in virtual space. The second is tilting the virtual world to again try to provoke a reaction from the participant. In the real world, the instructor bumped the participant's chair and play a sound. Each participant filled out a survey at the end of the game on whether they saw each of the unexpected events during the game. Finally, with the experiment setup for measuring attention, we also looked at measuring fatigue as it is a good indicator of the player's wiliness to play an exergame.

To measure fatigue, we use the endurance calculation from the work done by Hincapié-Ramos et al[2] which uses shoulder torque as a measure of exertion in calculating gestural fatigue. Shoulder torque is calculated assuming the arm is a pendulum from the shoulder joint to the arm's center of mass. The should torque is calculated as the torque due to acceleration from movement minus the torque due to gravity and inertia. Torque relates to exertion because it takes into account the angle of your arms and the forces applied, which can then be measured every frame of the Kinect.

We hypothesized two ideas for attention and fatigue. First we believed players who are playing at a higher difficulty will pay more attention to the game and its objectives and as a result, will miss more external events. Essentially, players who are more visually and mentally taxed should have less capacity to notice external stimuli. Second, we believed that players who are engaged will not succumb to fatigue and instead will push through it. The idea here is the more focused you are on on the game, the less likely the player is going to notice how tired they are.

In looking at the data, we compare all the unexpected events based on how many people noticed or did not notice the events during the experiment. This was done to remove unexpected events that may have been implemented incorrectly or simply not useful measures of attention. We are looking for the event that was closest to 50% notice rate, which is the projectile that comes toward the player's view. We then used this the projectile as our means of attention and compared it to difficulty average over the duration of the game for each player or the average exertion over the duration of the game. We found that players who play at a higher difficulty average are more attentive to the game (missed the projectile) and its objectives than external stimuli. We also found that these same people who are more focused on the game push through fatigue (exert as much in the beginning as the end) and exert themselves more physically than the other players. These observations not only help in validating our measurements of both attention and fatigue, but they also serve as step towards measuring engagement in exergames. Measuring engagement can ultimately give us a better understanding of creating an agile game.

References

[1] E. Brown and P. Cairns. A grounded investigation of game immersion. In CHI'04 extended abstracts
on Human factors in computing systems, pages 1297{1300. ACM, 2004.
[2] J. D. Hincapié-Ramos, X. Guo, P. Moghadasian, and P. Irani. Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In Proceedings of the 32nd annual ACM conference on Human factors in computing systems, pages 1063{1072. ACM, 2014.
[3] D. J. Simons and C. F. Chabris. Gorillas in our midst: Sustained inattentional blindness for dynamic events. Perception-London, 28(9):1059{1074, 1999.