CDMTCS
Research
Report
Series

University of Auckland Computer Science Graduate Workshop 2006

N. Hay, Al. Shorin, J. Wang (eds.)
University of Auckland, NZ

CDMTCS-297
January 2007

Centre for Discrete Mathematics and
Theoretical Computer Science

University of Auckland Computer Science Graduate Workshop 2006

Nick Hay, Al Shorin, and James Wang (Editors)

8th September 2006

Introduction

This booklet contains the proceedings of the First University of Auckland Computer Science Graduate Workshop (UACSGSW'06) held on Friday, 8 September 2006.

The Workshop, the first of its kind in our department, offers graduate students a chance to present their research to an audience including computer science and information technology graduate students, academics and industry representatives. The participants are MSc, Honours, PGDipSci, PhD, MEng and stage-4 BE project students in computer science and related areas. All submissions have been refereed.

The 2006 keynote speaker, Dr. Santokh Singh, a recent PhD graduate in our department, will kick off the event with a presentation on how to survive academic research.

The Workshop was sponsored by the Research Committee of the department (with PBRF money). It was organised by the following graduate students:

- Nick Hay, chair, programme committee
- Al Shorin, chair, publicity & sponsoring committee
- Cong Wang, chair, organisation committee.

Three prizes will awarded at the Workshop: the Microsoft prize for best paper, the Microsoft prize for second best paper, and the Computer Science Department award for best presentation.

We hope that the first edition of UACSGSW will be a productive and enjoyable experience.

Cristian S. Calude, Clark Thomborson

Papers

Surviving Academic Research

Santokh Singh

KEYNOTE: Surviving Academic Research

Santokh Singh PhD

Research Programme Manager, Centre for Software Innovation
Honorary Research Fellow, Dept of Computer Science
The University of Auckland

"To know the road ahead, ask those coming back."
- ancient Chinese proverb

Contents

- Introduction
- Stages & Timeline of Research
- Motivating yourself
- Research
- Predicting the Future and Writing the Research Proposal
- Literature Review
- Carrying out the research
- Implementation and Testing
- Writing the thesis/report
- Pitfalls
- Finally, if all goes well…
- Conclusion

Introduction

- to the Computer Science Graduate Student Workshop
- In Computer Science, Post Graduate research should involve some form of original IT related research, ideas, implementations, presentations etc.
- What are the difficulties?
- How to overcome the difficulties?
- Ultimately, how to excel?

Stages

- Deciding to do Post Grad research: opportunities, work, time, family, money needed, existing loan…
- Finding a supervisor(s).
- Determining a research topic.
- Writing the research proposal.
- Doing the literature review and research.
- Writing up the thesis/report.
- Submitting.
- Getting thesis marked.
- Graduation and celebrations (if all goes well ☺).

Timeline of Research and Thesis

- Workout a proper plan and timeline, illustrate the project schedule, e.g. by using a Gantt Chart.
- Keep track of work done and work that needs to be done.
- Do the research, don't postpone.
- Alert! Try to be ahead of the timeline & milestones– but this is not practicable or realistic, most times (To err is human…).
- Publish conference papers, journal papers, posters.
- Give seminars, presentations.
- Discuss outcomes, issues, research direction. Collaborate.
- Write parts of report, thesis along the way & collate.

"Failing to plan is planning to fail."

Motivating yourself

- Be honest, the research is important (maybe, more so is the degree).
- Be prepared to spend time (a lot of time) reading.
- Research on the relevant material first, and keep to it as closely as possible and you'll be more enthusiastic about your research.
- Talk to those who have made it – I've spoken to people (for advise) even moments before my final PhD Oral exam.
- Attend conferences, workshops, present whenever possible, network with postgraduates, you'll realise that you are not the only (troubled?) one in this world…
- Be in high spirits, but not high on spirits!

".. the Twentieth Century, the innovative century that brought you WW I, WW II, and WWW."   — Bill Higgins

Research

- a means of ensuring that we keep up to date and further the boundaries of innovation and knowledge
- promotes a tolerance towards uncertainty
- instils a questioning and inquiring attitude
- develops specific skills
- equip students for life-long learning
- *See, its more than just cheap labour for academics…*

Research

- Research areas – AI, algorithms and data complexity, graphics, software tools, computer architecture, data communications, distributed computing etc.
- Research area should match your research interest.
- **Discuss and choose a (suitable) topic with the supervisor(s) – though this topic may 'evolve' over time.**

Predicting the future?

- "Computers in the future may weigh no more than 1.5 tons."
  - Popular Mechanics, 1949
- Heavier than air machines are not possible
  - Lord Kelvin
- "I think there is a world market for maybe five computers."
  - Thomas Watson, chairman of IBM, 1943.
- "640K (of memory) ought to be enough for anybody."
  - Bill Gates, 1981.
- "Everything that can be invented, has been invented."
  - Charles H. Duell, Commissioner, U.S. Office of Patents, 1899.

Writing Research Proposal

- Hard to know and predict what you are going to learn or produce without carrying out the activity.
- Seek professional advise
  - Communicate with supervisors
  - Discuss with colleagues
  - Attend workshops
  - Forums (but chatrooms, texting, instant messaging have a tendency to be counter productive if trying to carry out serious research)

Literature Review

- Do literature review
  - To determine the current status, background knowledge
  - So that we do not reinvent the wheel, i.e. do not duplicate work already done
  - Constructively critique work done
  - Increase knowledge (including in other related areas to become more aware of happenings around us)
- If not sure about the existing literature, ask – "Ask and you shall (may) be answered".
  - do not sweep it under the carpet as otherwise you may be asked about it at a time you least expect, e.g. during the examination.
- Be critical - don't believe everything you read or are told. But do believe that the fundamentals are (may be) true.

Literature Review

- Efficient Reading
  - Read smartly and selectively – e.g. start from title, abstract, then introduction and conclusion, lastly the content of the journal paper/chapter if not bored yet.
  - Summarise the literature review – write down remarkable ideas, interesting problems, and possible solutions etc.
  - Take down useful references, quotes.
- Write the related works/background chapter.

Carry out the research

- This is the most important phase
- Plan the execution of your research
- Break it up to smaller manageable units
- Do the research and record all results and observations
- Do the research all the way to the finish – it can be lonely, but you are never alone.

The Feynman Problem Solving Algorithm:
1) Write down the problem.
2) Think very hard.
3) Write down the solution.

Implementation

- Usually inescapable, e.g. implementing software tools, systems etc.
- Do Requirement Engineering, analysis, design, implementation, testing, refactoring
- Can get reusable code from web, friends etc but must acknowledge.
- Do testing – prototype? Seldom for commercial release.
- User feedback

Testing and evaluation

- Apply for ethics approval to the Ethics Committee of the University of Auckland
- Get volunteers – be polite.
- Get the results, record, interpret, analyse, and conclude.
- Make changes according to the suggestions/comments/feedback from the tests
- Good research demands the ability to evaluate - intelligently and correctly.

"I can calculate the motions of the heavenly bodies, but not the madness of people."
- Isaac Newton

"Present to inform, not to impress; if you inform, you will impress. "
- Fred Brooks

Writing the thesis/report

- Collate all information/data and relevant material – should have mastery over the subject matter already.
- Write chapter by chapter if possible.
- "Proof-read carefully to see if you any words out";-)
- Submit the chapters to the supervisor, get feedback & …

"Remember the Golden Rule:
Those who have the gold make the rules."

"If it wasn't backed-up, then it wasn't important."

Pitfalls

- Trying to solve the world's problems.
- Solving all other problems except the thesis.
- Misunderstood but claims genius capabilities.
- Absolute Love for making things more complex.
- Lost in abstraction.
- Planning to win Nobel prize before even deciding on the topic.
- Just starting to do the evaluation but ready to submit…
- Still no proper literature review even after several years – possibly very learned in other things due to reviewing all other literature except the relevant ones.
- No motivation to continue, nor any inclination to end it.

Pitfalls continued - Procrastination

- A bus station is where a bus stops. A train station is where a train stops. On my desk, I have a work station....
- Since "my future depends on my dreams", I'm learning to sleep more.
- Never put off until tomorrow what you can put off today.

EDGAR DOWSE, AGED 93 (LITERALLY AGED), THE OLDEST PERSON IN THE WORLD TO GET A PHD

"A PhD is about finding out more and more about less and less until one eventually knows everything about nothing"

Finally, if all goes well…

- Award of the degree!!!
- "The degree represents the beginning -- the start of a never-ending journey of discipline, work, and the pursuit of an ever-higher standard for the master"
- “If you understand this, you are ready to receive your degree and ***BEGIN*** your work."

Conclusion

- Post-grad research has its peaks and troughs, ups and downs, and flat-outs.
- It can become 1% inspiration, 99% perspiration
- No matter what, don’t give up the journey…

> ***..to sail beyond the sunset….***
>
> ***One equal temper of heroic hearts***
> ***Made weak by time and fate***
> ***But strong in will***
> ***To strive, to seek, to find...***
> ***And not to yield.***
> **- Lord Tennyson**

- Thank you and good luck

Topological Analysis of Admissable Heuristics in IDA*

Santiago Franco

Extended Abstract: Topological Analysis of Admissible Heuristics in IDA*

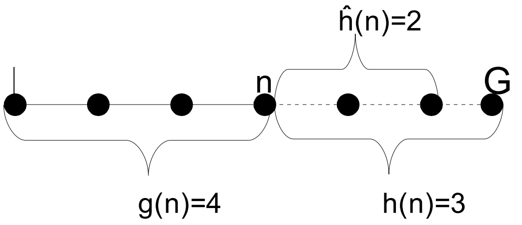
I. Goal

Given an admissible heuristic, a problem instance and an informed search based problem solver we want to predict accurately the size of the search tree generated to find an optimum solution.

II.Introduction

Heuristics are created to reduce the time it takes to solve problems. Admissible heuristics are used to perform informed searches(A*,IDA*,etc) to find optimal solutions to problems.

Figure 1: Informed Search Diagram



n is the current state.
g(n) is the Optimal Distance(*OD*) from I to n
h(n) is the OD from n to G
$\hat{h}$ is the heuristic estimate OD from n to G.
F(n)=g(n)+h(n)=OD.

$\hat{F}(n)=g(n)+\hat{h}(n)=estimated\ optimal\ distance$

Each dot represents a problem state in the optimal path from initial state I to goal state G. Informed search on planning expands all nodes from I to G whose $\hat{F}(n)\leq optimal\ distance$. When no heuristic is used all the nodes whose distance to G is less or equal to the OD would be expanded. This tree is called Brute Force Search Tree (BFST).

$\hat{h}$ is admissible iff: $\forall n \rightarrow \hat{h}(n)\leq h(n)$ Eq. 1

$\hat{h}$ is consistent for any choice of problem states n, m iff:

$\forall n, m \rightarrow \hat{h}(n)\leq distance_{min}(n,m)+\hat{h}(m)$ Eq 2.

Informed search algorithms which use admissible heuristics are guaranteed to find the optimal solution (eventually). Consistency guarantees that when node n is expanded it has already found an optimal path to n. All admissible heuristics can be made consistent[1]. Admissible heuristics are lower bounds on the optimal distance.

The text book standard for characterizing the effect of admissible,consistent heuristics on search performance is to model the search for an optimal solution as the expansion of a Heuristic Search Tree(HST), from initial state to goal state, on which each node represents a list of successive actions taken from the initial state. HST is a sub-tree of the BFST and thus smaller[2][3]. The quality of the heuristic is defined by how small the HST is.

Among the informed search algorithms IDA* is of particular interest to us. IDA* is a linear-space version of A*. It performs a series of depth first searches, pruning a path and backtracking when the cost $\hat{F}(n)$ of a node n on the path exceeds a $\hat{F}$ bound *C* for that iteration. The initial $\hat{F}$ bound C_o is set to the heuristic estimate of the initial state, and increases in each iteration to the lowest cost of all the nodes pruned on the last iteration, until a goal node is expanded. IDA* guarantees an optimal solution if the heuristic function is admissible.[4]

III.Problem Description

The goal stated in section I is impossible. In the best case scenario we would know how the HST expands as it grows but we would still need to know how far the HST will need to be expanded to find an optimal solution to the problem instance. Only solving the problem instance tells us the optimal distance.

The best next goal would be to predict the size of the HST given a $\hat{F}$ bound. The main difference between IDA* and other informed search algorithms is IDA*'s iterative nature, which is $\hat{F}$ bounded.

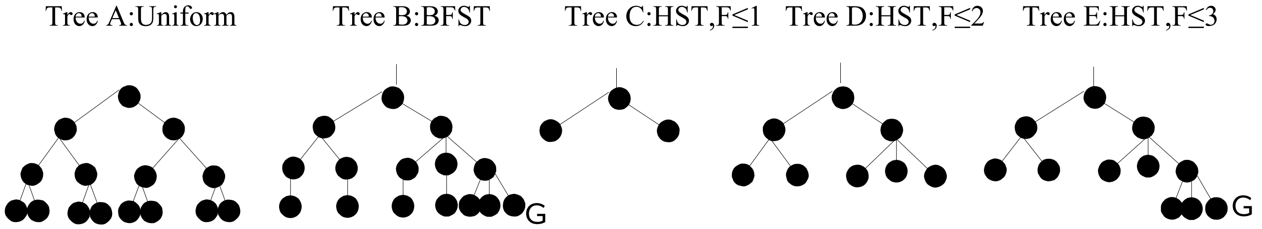
Each iteration of IDA* generates a $\hat{F}$ bounded HST which is a sub-tree of the next IDA* iteration's HST. The effect of each successive iteration is to raise the $\hat{F}$ -bound, adding nodes to the HST's until the final iteration. Earlier IDA* iterations can be used to predict the size of future iterations HSTs.

Existing approaches perform statistics on a problem domain by expanding HSTs for a significant number of problem instances. Individual instances are assumed to behave similarly to the average case. No approach has addressed whether it is possible to predict the HST size for future iterations with the data gathered from earlier IDA* iterations for the problem instance being solved.

Our main goal is to develop a new domain-independent model which will use the data gathered on earlier IDA* iterations to predict the $\hat{F}$ -bounded size of the HST on later iterations

IV.Models for predicting the size of a Search Tree

Figure 2: Example Search Trees



A *uniform tree* is a tree on which all expanded nodes have the same branching factor and the depth of all its search paths is constant[2]. A uniform search tree is defined by its depth and branching factor. Tree A on Figure 2 is a uniform search tree.

$$N_T = \frac{B^{D+1} - 1}{B - 1}$$

Eq. 3; N_T=nodes created; B=branching factor;D=depth

The informed search for an optimal solution in planning is currently modeled as the expansion of a HST whose size is smaller or equal to that of a BFST. Even though neither the BFST nor the HST are uniform search trees Eq3 has been used to model the size of BFST and HSTs for significant number of problem instances across a domain[2][3]. We will use this formula as the basis of the formula to predict the BFST and HST size as we increase the $\hat{F}$ -bound iterations of IDA* for individual problem instances. Its two features are B and D. Tree A has B=2,D=3 so N_T=15.

The BFST does not expand with one uniform branching factor. Each node in the BFST represent a state in the domain, and each of its children represents the result of an action applied to the parent node. Consequently each node in the BFST has a varying branching factor depending on how many actions are available from the current state. Tree B is an example of BFST.

In order to model the BFST as a uniform search trees the effective branching factor(EBF) is used in the text book model[2][3]. EBF is a simplification and represents the mean branching factor of the BFST. The BFST B has an EBF=2,D=3 so NT=15.

Korf also proposed the Heuristic Branching Factor(HBF) as an alternative branching factor. HBF is the rate of growth of the HST between two IDA* iterations. It is stable in the limit of large iterations but not on the initial IDA* iterations. Korf also claims that heuristic choice does not alter the HBF[4].

If we define depth as the length of the path from I to the tree leaves then a uniform depth is not necessarily accurate for all paths of the BFST and specially not for the HST. Depending on the heuristic used some nodes will not be expanded in the HST and thus some search paths terminate earlier than the optimal depth. Also the BFST and consequently the HST may have nodes which do not expand because no action is possible from that state. Effective Depth(ED) has been suggested as an alternative to uniform depth [5]. ED is a simplification and represents the mean depth factor of the search tree.

The final HST E has a varying branching factor and a varying depth, so how do we apply Eq 3? How do we go from the quasi uniform BFST B whose size, EBF and depth we can model easily to its subtree HST E? There are infinite solutions for eq. 3 with 2 unknowns variables (EBF, ED) and only one known variable(NT).

The textbook approach would model HST E size as a uniform search tree whose EBF [2][3] has been reduced relative to the BFST EBF and whose depth is fixed to the optimal solution depth. Korf's alternative approach would be to model the HST E as a uniform tree whose EBF is the same as the corresponding BFST but whose ED has been reduced relative to the BFST[5].

Both approaches are arbitrarily fixing the depth or the branching factor as invariants. If our goal was to use Eq3 to model only the final HST E for each problem instance then it would not matter whether EBF or ED is the variant, they are equally valid.

But our goal is to use the data gathered on earlier IDA* iterations to predict the $\hat{F}$ -bounded size of the HST on later iterations. HST E is the final iteration of IDA* search. The two previous iterations generated the HSTs C & D. Using a modified version of eq 3 we can create a system of equations with 2 formulas describing C & D with two unknowns (depth and branching factor). Once depth and branching factor variables have been solved we can predict the size of the HST for any $\hat{F}$ bound.

V.Proposed Approach & Current status

Given data gathered for earlier IDA* iterations, on a problem instance, our goal is to predict the size of the HST for future IDA* iterations up to the OD. We need a formula which models the growth of the HST up to the OD. The formula should be able to predict the size of the HST given a $\hat{F}$ bound. We need to define the variables that, given a $\hat{F}$ increase, can be used in the formula to predict the size of the HST.

The HST for any iteration is a subtree of the corresponding BFST. So we can describe the size of the $\hat{F}$ bounded HST as a function of the pruning of the BFST.

$$N_T = \frac{(EBF_{BFST} - EBFR)^{\hat{F} - EDR + 1} - 1}{EBF_{BFST} - EBFR - 1}$$

Eq. 4. EBFR=EBF Reduction;EDR=ED Reduction

Only two iterations are needed to numerically solve Eq. 4 for EBFR and EDR. After running the first two iterations the only unknown variables are the EBFR and the EDR. Once they are calculated, the size of future iterations is a function of $\hat{F}$ only.

Solving Eq 4 with two iterations does not guarantee a perfect prediction. It assumes that both the EBFR and EDR will remain constant as $\hat{F}$ increases.

We have run thousands of instances of the Eight Puzzle for different OD with three different admissible heuristics (Out of Place, Manhattan, Relaxed Adjacency). We have come across some interesting results. If we fix EDR to 0 as the textbook model would suggest the predictions have a very low quality. If we instead fix the EBFR to zero, as Korf model would suggest, the predictions are much better. But when we do not fix EBFR or EDR and instead calculate them independently with 2 iterations we get the better predictions for all heuristics.

Our preliminary results support Korfs claim that EDR is a better predictor than EBFR. when used as the sole variant feature mapping growth of IDA* iterations for the same problem instance, as Korf claimed. However our results support that the EBF vary depending on which heuristic we use, as in the textbook model, contrary to Korf claims. Our preliminary results validate our approach to account both for a EBFR and a EDR when predicting the size of a $\hat{F}$ bounded HST.

Domain statistical approaches like Korfs', Russell's or Nilson's are not specific to the problem instance, thus the differing effects of the heuristic can be averaged out across different problem instances. Since our approach is specific to the individual problem instance it does not have this problem.

Eq 4 depends on the EBF_{BFST} to be calculated for each iteration. On our experiments on the eight puzzle domain, EBF_{BFST} is very stable and does not change after two iterations, saving us the computational effort of expanding BFST for large depths. This is not necessarily the case for other less regular domains. More experiments are needed in other domains to test and improve our approach.

Bibliography

[1] L. Mero. A heuristic Search algorithm with modifiable estimate. *Artificial Intelligence* (1984) **23** : p. pp 13-27..

[2] Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach. . Prentice Hall, 2003.

[3] Nils J Nilsson. Artificial intelligence: a new synthesis. . Morgan Kaufmann Publishers Inc, 1998.

[4] Richard E. Korf. Recent Progress in the Design and Analysis of Admissible Heuristic Functions. In *AAAI/IAAI:1165-1170*. 2000.

[5] Richard E Korf, Michael Reid & Stefan Edelkamp. Time complexity of iterative-deepening-A*. *Artif. Intell* (2001) **129**: pp. 199-218.

Semantic XML query optimization

Ke Geng

Semantic XML query optimization

Ke Geng
Supervisor: Gill Dobbie

The objective of my research is to carry out query optimization of XML queries based on the content of the documents.

Now eXtensible Markup Language (XML) [1] is widely accepted by website constructors and programmers because of its capability and its flexibility in storing and transferring semistructured data. Query transformation is the method used to improve the query execution time of the database system. Some research has been done with XML query transformation. To date, these works concentrate on the structure of XML documents and little work has been done that takes into account the content of XML documents. With XML documents there is a need to concentrate more on the content because XML is more flexible than other databases and XML documents do not typically have a schema. One of the disadvantages of the flexibility is that the structure of XML documents is inconsistent and unpredictable.

Web Ontology Language (OWL) [4] is an ontology language that can formally describe the meaning of terminology used in XML documents. In contrast to other ontology languages, such as RDF (Resource Description Framework) [6], OWL provides more meaningful terms to describe the relationships of elements of the XML document. With these description methods, the elements in XML document can be grouped and described based on the features of their contents.

In this research, we will undertake work in the content analysis of XML documents and how the content analysis can help XML query transformation. All the elements in an XML document will be analyzed and classified with their content features using an On Line Analytical Processing (OLAP) [10] system. We choose the OLAP system as the analysis tool because it has two important features: a multidimensional cube can help us to analyze the XML documents from different dimensions and the hierarchy provides a method to classify the elements with their content. The generated groups will be described with their features in an OWL document. With the generated OWL documents, the input XML query can be transformed to a more efficient one, which has the same function as the original query.

Currently little work on building OLAP systems for XML databases has been done because of XML's unpredictable structure. The existing methods, which can present XML content in an OLAP system, can not store all the information of XML documents, especially the hierarchy relationship between elements.

The optimization method is described in the following steps:

1. Extract and analyze the content of an XML document using OLAP technology.
2. Transfer the result of the analysis and represent it in OWL.
3. Carry out query optimization using the content of the OWL document.

Some of the challenges of this research include:

- What is the best way to describe elements that leads to best result in XML query transformation?

  OWL provides several patterns to describe the elements of the XML document. For example: The element “dot” can be defined as a fundamental class of two subclasses “commercialDot” and “residentialDot”. It can also be described as the union of “commercialDot” and “residentialDot”, which looks like:

  ```
  <owl:Class rdf:ID="dot">
     <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="commercialDot" />
        <owl:Class rdf:about="residentialDot" />
     </owl:unionOf>
  </owl:Class>
  ```

  These two definitions have their own features. When the “dot” is defined as the fundamental class of two subclasses, we do not restrict that only these two subclasses are built based on class “dot”. The element “dot” may have other subclasses. This definition is flexible and easy to update when the original XML document is changed.

  When the “dot” is defined as the union of “commercialDot” and “residentialDot”, it has been restricted such that there are only two kinds of Dots in the composition of element “dot”. The relationship of elements defined with “unionOf” is more restricted than the definition of “subclass”, which may reduce the complexity of the OWL document analyzing procedure and help the database system to derive knowledge efficiently. But it is not as flexible as “subclass” to be maintained.

- How can the size of the OWL document be controlled?

  There is a trade-off between the amount of data that should be represented and the time it takes to deal with that data.

  Deriving knowledge from XML to generate OWL documents is carried out through the implementation of analysis with OLAP. Deriving knowledge from OWL documents for the query transformation is part of the algorithm of the implementation using OWL. The relation between the two implementations is very interesting. The more knowledge can be derived from the XML documents, the simpler the OWL implementation is. But this will increase the difficulty of the implementation of OALP analysis procedure. If the implementation of BI is not powerful enough, we should do more work to design a powerful algorithm to derive enough knowledge for the query transformation from the generated OWL document.

- What are the best data and tools for testing the system adequately?

  A number of benchmarks for XML have been developed recently. These benchmarks can be classified into two classes: application benchmarks and micro benchmarks. Application benchmarks are used to evaluate the overall performance of a database system by testing as many query language features as possible. Currently, most benchmarks of XML are application benchmarks. These benchmarks include: XMark [8], XBench [2], XOO7 [9] and XMach-1[5]. Micro benchmarks are used to test individual performance critical features of the language. There is only one micro benchmark for XML: Michigan Benchmark [7]. In my research, the content structure is the key point for the optimization. All the existing benchmarks can not satisfy the experiment because the benchmarks are designed for general purposes and the content is built with random algorithm. So seldom can content features be abstracted from the generated XML documents. A tool, which can adapt the existing benchmarks and build some content information for the experiment, is necessary in generating data (XML documents) for the experiments. Also there is no Native XML Database (NXD) [3] system that supports query transformation with information presented in OWL. So an assistant system, which can traverse OWL document and find the related information, must be designed. An algorithm, which can traverse OWL documents effectively without the support of some existing method, such as index, also needs to be designed.

In this paper, a semantic query optimization method is introduced. The challenges of the project are also discussed in detail. Now a tool that can adapt the existing XML documents has been designed, which can modify not only the content but also the structure of the XML document for the experiment. An assistant system, which can traverse OWL document and carry out query optimization with the information stored in the OWL document is being designed. More functions, such as choosing useful description from several descriptions of one element, will be added to the assistant system in the future research.

Reference

[1] Extensible Markup Language (XML). (2006). Retrieved 24/06, 2006, from http://www.w3.org/XML/

[2] Benjamin Bin Yao, M. T. Ö. (2002). XBench - A Family of Benchmarks for XML DBMSs Retrieved 15/09, 2005, from http://se.uwaterloo.ca/~ddbms/projects/xbench/People.html

[3] Bourret, R. (2005). Going native: Use cases for native XML databases. Retrieved 03/05, 2005, from http://www.rpbourret.com/xml/UseCases.htm

[4] Deborah L. McGuinness, F. v. H. (10/02/2004). OWL Web Ontology Language Overview. 20/08/2005, from http://www.w3.org/TR/owl-features/

[5] Erhard Rahm, T. B. (2002). XMach-1: A Multi-User Benchmark for XML Data Management. Paper presented at the Conference Name|. Retrieved Access Date|. from URL|.

[6] Frank Manola, E. M. (2004). RDF Primer. Retrieved 27/08, 2005, from http://www.w3.org/TR/rdf-primer/

[7] Kanda Runapongsa, J. M. P., H.V. Jagadish, Yun Chen, Shurug Al-Khalifa (2001). The Michigan Benchmark: Towards XML Query Performance Diagnostics. from http://www.eecs.umich.edu/db/mbench/mbench.pdf

[8] Ralph Busse, M. C., Daniela Florescu,Martin Kersten,Ioana Manolescu,Albrecht Schmidt, Florian Waas. (2003). XMark — An XML Benchmark Project. Retrieved 18/09, 2005, from http://monetdb.cwi.nl/xml/index.html

[9] Stéphane Bressan, M. L. L., Ying Guang Li, Bimlesh Wadhwa, Zoé Lacroix, Ullas Nambiar, Gillian Dobbie The XOO7 Benchmark Retrieved 17/09, 2005, from http://www.comp.nus.edu.sg/~ebh/XOO7.html

[10] Youness, S. (2000). Professional Data Warehousing with SQL Server 7.0 and OLAP Services. Birmingham: Wrox Press Ltd.

Real-time Traffic Flow Patterns and Behaviours Analysis

DongJin Lee

Real-time Network Traffic Flow Patterns and Behaviours Analysis

Extended Abstract

DongJin Lee and Nevil Brownlee

Computer Science Graduate Workshop 2006,
Department of Computer Science
The University of Auckland, New Zealand

Today's network traffic is enormous and complex. According to Moore's Law [1, 2] in network traffic, bandwidth is doubling every 12~18 months. As the technology grows up, communication has become a vital part of the IT industry resulting in an exponential increase in network usage and has become a critical environment for multibillion dollar businesses. As well, network traffic is an important research area for large ISPs, Universities and Research Network Operators. The Internet is a giant mesh of hosts and devices (e.g. clients, servers and intermediates) all interconnected with their neighbours, enabling users to communicate with anyone using Internet Protocols (IP). Intermediate devices are what we normally call routers; the main function of these devices is to forward IP packets to their neighbouring devices, so as to ultimately deliver messages across to receiving hosts.

There are many types of applications and each of them behaves differently to others, for example, some can be distinguished by packet counts, sizes, protocol, lifetimes, but others can be hard to be distinguish in that way due to various behaviour changes.
There are two widely known types of traffic measurement: active and passive. Active measurement analysis is done by sending an actual IP packet across the network so as to find latency and delay times (e.g. ping, traceroute). On the other hand, passive measurement analysis is done by collecting series of packets passing one or more points in the network (e.g. NeTraMet [3]). No packet is sent using this method.
Both types have trade-offs depending on the particular measurement. Passive measurement is especially suitable for monitoring overall traffic usage, host behaviours and also for security monitoring, for example, as part of Intrusion Detection System (IDS).

Traditional analysis consists of identifying packet payload to accurately determine a host's applications and activities. However, matching the content of each packet is not only inefficient, but it may also violate privacy within many organizations. Also, payload itself is not sufficient to discover host's behaviours. Recently, several different approaches to analyzing the network traffic were invented and are now well known among researchers.
Flow Analysis is about using 5-tuple[1] from a packet header to uniquely identify and build a table of flow information. Packet and byte counts would be updated for each packet in

[1] A 5-tuple (source address, destination address, source port, destination port, protocol) identifies a traffic flow.

the flow. Analysts have found flows useful because they are more efficient and they avoid privacy issues since the payload itself is not analyzed. Recent studies in [4-7] demonstrate notable results.
Clustering is about grouping hosts into the same group based on their connections, ports, addresses and so on. Its end result often displays the differences between two or more separate groups using various graphs. Examples in [8, 9] shows that clustering can achieve various host classification groups.

Some of the previous methodologies were heavily based on *trace files*[2]. This approach can perform detailed analysis of the traffic, but in practice, analysis lacks responsiveness and it is often too slow to discover misbehaving hosts. What we need is a system that can capture and process packets from the live network interfaces in real-time to generate views so as to see the changes in hosts' behaviours and patterns. By doing this, a user can receive feedback and also be notified how/when the behaviours and patterns change.

Our analysis is based on passive measurement: traffic data is collected at the UoA's Internet gateway and processed in real time. Initially, we focus on popular hosts such as DNS Servers, Proxy/Gateway Servers, Web Mail and typical Web Servers. We build properties from each individual host traffic flow. For each host, we define a set of metrics $M = [hf, vsf, sf, lrf, uf]$ where all the metric values are ratios. For each measurement interval we gather two sets of metric values for each host $C = counts(M)$ and $S = sizes(M)$ where C and S are host flow metrics of counts and sizes respectively.

```
a)--------------------------------
[Host X]
-C(M)=[hf, vsf, sf, lrf, uf]
-S(M)=[hf, vsf, sf, lrf, uf]
----------------------------------
hf – host flow ratio
vsf – very short flow ratio
sf – short flow ratio
lrf – long running flow ratio
uf – utilization flow ratio
```

```
b)----------------------------------
[Host DNS_130.216.1.1]
-C(M)=[97.2, 81.3, 18.5, 0.3, 75.4]
-S(M)=[2.1, 57.9, 39.1, 2.8, 68.8]
```

```
c)----------------------------------
[Host EZProxy_130.216.191.84]
-C(M)=[5.1, 90.2, 9.8, 0.0, 79.2]
-S(M)=[0.5, 36.5, 63.4, 0.1, 60.4]
```

Figure 1: a) sets of metrics. b,c) example of hosts with sets of metrics

Figure 1(a) illustrates basic sets of metrics in percentage ratio. Each set represents a host's flow patterns; these values are differs among various types of hosts.
Figure 1(b,c) are examples of DNS and Proxy hosts with metrics; we see different metrics for the two hosts' sets of metric values.

[2] A single or several files that contains packets to be analyzed.

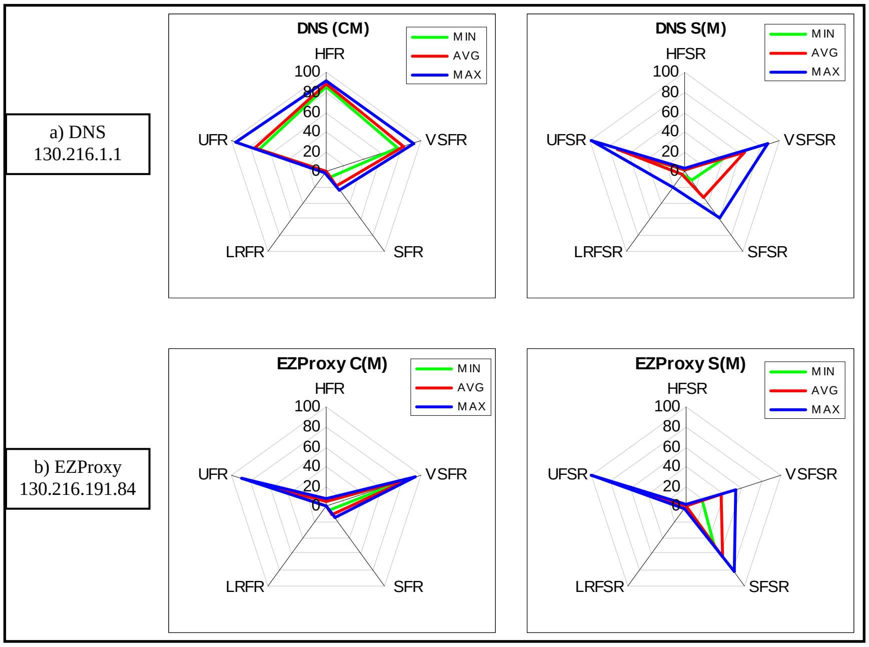


Figure 2: Two hosts with radar graph of sets of metrics

Figure 2 shows the radar graphs of DNS and EZProxy which have been measured continuously over 30 minutes. We see shape-like radar graphs depending on each host's sets of metrics. What is interesting is that the ratio values not only determine uniqueness of the hosts, but also catches a host's patterns; similar types of host behaviours have similar metrics values. We often want to find out a host's likely behaviours in relation to the other hosts so that when it misbehaves, we would see the changes in its metrics. For example, if a typical web server is being attacked by Denial of Services (DOS), then we see the metrics change.

We include a reset threshold on sets for each host so that in every reasonable period (e.g. 30 minutes), sets are reset and built again so as to compare with previous ones for the changes. Metrics are updated if the changes are gradual, if not, then we regard these as a misbehaving host. Sets are retained within memory throughout the measurement so that we can also match to other unknown hosts' patterns. In this we found that the majority of similarly behaving hosts fall into the similar sets of metrics we created initially.

We also desire to find hosts that are mixed with several patterns so as to classify them into different host groups.

One of the problems in analysing patterns is to discover reliable sets of metrics. Host behaviours can be highly dynamic and varied depending on the traffic congestions, popularities, application versions and so on. Because of uncertainties, we need to have a detailed analysis of individual host's behaviour so as to discover their patterns.

Again, finding out the acceptable threshold values in collection time, resetting time, etc to accurately and consistently produce sets is another challenge.

Furthermore, we would like to analyze traffic data from other sources (e.g. data from other countries, etc) and run our methodologies so as to compare and understand the different host behaviours and patterns.

Reference

[1] K. G. Coffman and A. M. Odlyzko, "Internet growth: is there a "Moore's law" for data traffic?," in *Handbook of massive data sets*: Kluwer Academic Publishers, 2002, pp. 47-93.

[2] C. A. Eldering, M. L. Sylla, and J. A. Eisenach, "Is there a Moore's law for bandwidth?," *Communications Magazine, IEEE*, vol. 37, pp. 117-121, 1999.

[3] N. Brownlee, "NeTraMet - a Network Traffic Flow Measurement Tool, http://www.caida.org/tools/measurement/netramet/," vol. 2006, N. Brownlee, Ed., 2006.

[4] L. Anukool, P. Konstantina, C. Mark, D. Christophe, D. K. Eric, and T. Nina, "Structural analysis of network traffic flows," in *Proceedings of the joint international conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM Press, 2004.

[5] B. Paul and P. David, "Characteristics of network traffic flow anomalies," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. San Francisco, California, USA: ACM Press, 2001.

[6] K. P. Thomas Karagiannis, Michalis Faloutsos, "BLINC: multilevel traffic classification in the dark," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*. Philadelphia, Pennsylvania, USA: ACM Press, 2005.

[7] K. Myung-Sup, K. Hun-Jeong, H. Seong-Cheol, C. Seung-Hwa, and J. W. Hong, "A flow-based method for abnormal network traffic detection," presented at IEEE/IFIP Network Operations and Management Symposium, Seoul, 2004.

[8] X. Kuai, Z. Zhi-Li, and B. Supratik, "Profiling internet backbone traffic: behavior models and applications," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*. Philadelphia, Pennsylvania, USA: ACM Press, 2005.

[9] K. Balachander, W. Jia, and X. Yinglian, "Early measurements of a cluster-based architecture for P2P systems," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. San Francisco, California, USA: ACM Press, 2001.

Mathematical Foundation for Semistructured Data

Scott Uk-Jin Lee

Mathematical Foundation for Semistructured Data

Scott Uk-Jin Lee
Department of Computer Science
The University of Auckland
Auckland, New Zealand
scott@cs.auckland.ac.nz

1. Introduction

The rapid growth of the World Wide Web and its technologies has resulted in enormous amounts of data being used over the Internet by Web Services and Web-based applications. The increase in semistructured data usage is not limited to Web applications but expands into various other applications such as digital libraries, biological databases and multimedia data management systems. This expansion of semistructured data usage creates the need for effective and efficient utilization of semistructured data [15].

With such a rapid increase in its usage semistructured data needs to be stored, manipulated, and queried to be utilized properly by various applications and tools. For these purposes, many researchers have proposed to design and develop adequate database systems for semistructured data. As a result, several database systems have already been developed for eXtensible Markup Language (XML) [6], which is a common representation for semistructured data, while traditional database companies, such as Oracle, have provided XML support for their existing database systems.

As with widely used database systems, various operations that transform the schema have been adopted by the database systems developed for semistructured data to provide effective and efficient data storage and utilization. These schema transforming operations are often performed using algorithms developed specifically for semistructured data storage. The schema transforming operation guided by the algorithms must perform correctly to ensure the consistency of the data and ensure no information is lost. Although the algorithms claim to maintain the lossless and dependency preserving properties, the database systems developed for semistructured data lack verification support to prove the correctness of the transformations.

In widely adopted database systems, one of the features that is used to prove the correctness of the operations and algorithms is the mathematical foundation. For example, in relational database systems, a mathematical foundation has been extraordinarily useful in the definition of normalization, to prove that lossless and dependency proving algorithms can be defined. Also a mathematical foundation has been defined to capture object oriented concepts, and used to reason about the correctness of query results in object oriented database systems. Such verification support for operations and the algorithms of database systems ensures the correctness of data manipulations, making the mathematical foundation essential.

However, current developments of database systems that store semistructured data lack a mathematical foundation. When there is no general formal way of distinguishing between correct and incorrect transformations, incorrect data transformation can be introduced resulting in unreliable or even corrupt data. Without establishing a well defined mathematical foundation, many limitations will be imposed on the functionality of the database systems for semistructured data making it not as effective and reliable as it should be.

Therefore this research proposes to establish a well defined mathematical foundation for semistructured data in order to address this problem. The derived mathematical foundation will verify whether operations and algorithms that transform the schema of semistructured data maintain the lossless and dependency preserving properties.

2. Proposed Research

The main objective of this research is to establish a well defined mathematical foundation for semistructured data to verify the operations and algorithms that transform schemas of semistructured data. The mathematical foundation for semistructured data should contain formally specified semantics of a data modeling language with various database operations and algorithms represented accordingly. Also the mathematical foundation must be supported with adequate formal verification tools for verification of various database operations and algorithms. Hence, to define a mathematical foundation for semistructured data, a standard representation of schemas and the schema transformation operators for semistructured data must be formally specified and verified using adequate data modeling languages and formal languages.

There has been other research that provides a formal semantics for semistructured data. For example, the formalization of DTD (Document Type Definition) and XML declarative description documents using expressive description logic has been presented by Calvanese et al. [3]. Anutariya et al. presented the same formalization using a theoretical framework developed using declarative description theory [1]. Also spatial tree logics have been used to formalize semistructured data by Conforti and Ghelli [4]. More recently, hybrid multimodal logic was used to formalize semistructured data by Bidoit et al. [2]. While these works have helped us develop a better understanding of the semantics of semistructured data, none of them have applied adequate and automated verification. Furthermore, none of these researchers have considered providing specification and verification for operations and algorithms that transforms semistructured data schema.

As a result of examining related work and intensive background research, Object Relationship Attribute model for semistructured data (ORA-SS) data modeling language [5, 9] will be adopted as a data modeling language. The ORA-SS data modeling language is used because it not only captures the constraints that are represented in textual languages such as XML Schema [12] but also it is a diagrammatic notation which can be used for conceptual modeling.

With ORA-SS, we also applied a similar approach to formalize semistructured data using Z/EVEs [8] and Alloy [14]. But the approach using Z/EVEs had problems with complicated and time consuming verification and the approach using Alloy had a scalability problem. Considering these problems, the research will use Prototype Verification System (PVS) [10] and its theorem prover as the formal specification language and verification tool. Also PVS has proven its effectiveness by providing precise formal definitions and powerful automated verification support in various other research projects [11, 7, 13].

With ORA-SS data modeling language and PVS we will conduct the following tasks to complete the mathematical foundation for semistructured data.

- Specifying formal semantics of ORA-SS data model using PVS formal specification language with automatic verification support
- Verifying the defined ORA-SS formal semantics
- Specifying basic transformation operators according to the specification of ORA-SS formal semantics
- Representing existing database operations and corresponding algorithms for semistructured data such as normalization and view
- Verifying the correctness of the operations and algorithms (whether they maintain lossless and dependency preserving property)

When the formal definitions and verification described above are successfully completed, the research will result in the establishment of a well defined mathematical foundation for semistructured data. Additionally, the research can compare different algorithms derived for each database concept based on its performance to find the best algorithms using the verified representations of the database concepts.

This part of the research extends the defined mathematical foundation even further by enabling its practical applications on utilization of semistructured data in various applications and in its database systems. Also by representing these algorithms and verifying their correctness, the research will demonstrate the correctness and applicability of the mathematical foundation.

At the completion of all these tasks, the research will have defined a mathematical foundation and demonstrated its applications. The defined mathematical foundation of the research, that consists of verified formal specification of ORA-SS data model semantics, incorporated schema transformation operators and the verified representation of the best algorithms for each database operations, will be powerful enough to support effective and efficient use of semistructured data in various applications as well as its database systems. In addition, the research will also help the ORA-SS data model to evolve and provide a possibility for real Web-based applications to be developed from the ORA-SS data model.

3. Conclusions

This proposed research will establish a mathematical foundation for semistructured data to verify the schema and data transforming operations and algorithms for semistructured data. The advantages of having such a mathematical foundation includes providing formal semantics for semistructured data design, enhancing the discovery of inconsistencies in the data, providing verification for correctness of database operations such as normalization and view definitions, and maintaining lossless and dependency preserving properties of algorithms for database systems. Also the generic nature of the defined mathematical foundation allows it to be applied to any applications or database systems that use semistructured data.

Currently, the formal semantics of ORA-SS data model language has been specified and verified using PVS and its verification support. Using this formally specified and verified ORA-SS semantics, basic transformation operators will be defined and verified. Furthermore, based on the defined semantics of the ORA-SS data model and basic transformation operators various database concepts and its algorithms will be defined and verified completing the mathematical foundation for semistructured data. Then the defined mathematical foundation will be evaluated through several case studies of conducting verification for some essential database operations such as normalization and view definitions and their algorithms.

According to the related work and preliminary research, the proposed research is evaluated to be unique and very valuable. The outcome of the research will surely eliminate the current shortcomings of database systems for semistructured data and provide tremendous benefits to the database systems for semistructured data.

Therefore it is believed that the contribution of this research is essential for the dramatically expanding semistructured data to be as powerful and versatile as existing structured data providing a solid basis for less structured information to be used in various applications.

References

[1] C. Anutariya, V. Wuwongse, E. Nantajeewarawat, and K. Akama. Towards a Foundation for XML Document Databases. In *EC-Web*, pages 324–333, 2000.

[2] N. Bidoit, S. Cerrito, and V. Thion. A First Step towards Modeling Semistructured Data in Hybrid Multimodal Logic. *Journal of Applied Non-Classical Logics*, 14(4):447–475, 2004.

[3] D. Calvanese, G. D. Giacomo, and M. Lenzerini. Representing and Reasoning on XML Documents: A Description Logic Approach. *Journal of Logic and Computation*, 9(3):295–318, 1999.

[4] G. Conforti and G. Ghelli. Spatial Tree Logics to reason about Semistructured Data. In *SEBD*, pages 37–48, 2003.

[5] G. Dobbie, X. Wu, T. Ling, and M. Lee. ORA-SS: Object-Relationship-Attribute Model for Semistructured Data. Technical Report TR 21/00, School of Computing, National University of Singapore, 2001.

[6] E. R. Harold and W. S. Means. *XML in a Nutshell*. O'Reilly, Sebastopol, 3rd edition, 2004.

[7] M. Lawford and H. Wu. Verification of real-time control software using PVS. In P. Ramadge and S. Verdu, editor, *Proceedings of the 2000 Conference on Information Sciences and Systems*, volume 2, pages TP1–13–TP1–17, Princeton, NJ, mar 2000. Dept. of Electrical Engineering, Princeton University.

[8] S. U. Lee, J. Sun, G. Dobbie, and Y. F. Li. A Z Approach in Validating ORA-SS Data Models. In *3rd International Workshop on Software Verification and Validation*, Manchester, United Kingdom, 2005.

[9] T. W. Ling, M. L. Lee, and G. Dobbie. *Semistructured Database Design*, volume 1. Springer-Verlag, 2005.

[10] S. Owre and J. M. Rushby and and N. Shankar. PVS: A Prototype Verification System. In D. Kapur, editor, *11th International Conference on Automated Deduction (CADE)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752, Saratoga, NY, jun 1992. Springer-Verlag.

[11] M. Srivas, H. Rueß, and D. Cyrluk. Hardware Verification Using PVS. In T. Kropf, editor, *Formal Hardware Verification: Methods and Systems in Comparison*, volume 1287 of *Lecture Notes in Computer Science*, pages 156–205. Springer-Verlag, 1997.

[12] H. S. Thompson, C. Sperberg-McQueen, N. Mendelsohn, D. Beech, and M. Maloney. XML Schema 1.1 Part 1: Structures. http://www.w3.org/TR/xmlschema11-1/.

[13] J. Vitt and J. Hooman. Assertional Specification and Verification Using PVS of the Steam Boiler Control System. In J.-R. Abrial, E. Boerger, and H. Langmaack, editors, *Formal Methods for Industrial Applications: Specifying and Programming the Steam Boiler Control*, volume 1165, pages 453–472. Springer-Verlag, 1996.

[14] L. Wang, G. Dobbie, J. Sun, and L. Groves. Validating ORA-SS Data Models using Alloy. In *17th Australian Software Engineering Conference (ASWEC 2006)*, Sydney, Australia, 2006.

[15] X. Wu, T. W. Ling, M. L. Lee, and G. Dobbie. Designing Semistructured Databases Using the ORA-SS Model. In *WISE '01: Proceedings of 2nd International Conference on Web Information Systems Engineering*, Kyoto, Japan, 2001. IEEE Computer Society.

Simulation Models of Prebiotic Evolution of Genetic Coding

Sidney Markowitz

Simulation Models of Prebiotic Evolution of Genetic Coding
Extended Abstract

Sidney Markowitz
Department of Computer Science, University of Auckland, PB 92019, Auckland, New Zealand
sidney@sidney.com

Abstract

Common to all life on Earth are the mechanisms of genetic encoding, in which specific trinucleotide sequences in DNA and RNA map to specific amino acids in synthesized proteins. This thesis project investigates feasible models of the evolution of genetic encoding from an initially random population of genes and proteins. Discrete simulations on the order of 10^{10} event steps demonstrate self-organisation to apparent attractor states. This paper presents new results using a small gene coding model that appears to find an attractor state using purely stochastic processes beginning from a random state with no preferred genetic coding system. The focus of the research is on developing an abstract framework that does not depend on explicit molecular details while demonstrating a plausible mechanism of self-organisation and persistence.

Introduction

All known life uses proteins made of sequences of amino acids for structural and chemical purposes, and replicating genes of DNA or RNA as templates for the synthesis of the proteins. All life uses the same small set of building blocks of nucleotides and 20 standard amino acids. With only rare minor variations the same code maps trinucleotide sequences (codons) to amino acids. Subsequencs in DNA are copied into messenger RNA (mRNA), whose codons attach to molecules of transfer RNA (tRNA) that have been charged with mostly correct amino acids by protein catalysts called aminoacyl-tRNA-synthetases. This universal mechanism had its origins in the earliest beginnings of biological existence. While the mechanism of protein synthesis from DNA templates is well understood, its complexity, the dependence of protein synthesis on protein catalysts that are themselves products of the synthesis, and the near universality of the standard Genetic Code collectively pose the question that is at the heart of my research:

How did any, let alone one dominant Genetic Code bootstrap itself into existence and maintain its stability in the face of error-prone replication and translation?

Taking a molecular biological perspective in attempting to explain the origin of the genetic code, different researchers have proposed three broad classes of hypotheses:

- adaptation of the code through selection for some measure of optimality, such as minimisation of the physicochemical effects of single mutational or translational errors
- stereochemical associations between nucleotide sequences and amino acids
- coevolutionary paths that resulted from the biosynthesis of certain amino acids

In addition, the Frozen Accident theory proposed by Crick in his influential review paper (Crick, 1968) held a dominant position for the succeeding two decades until evidence was found for continuing evolution of the genetic code and in support of the above three models (Knight et al., 1999).

These theories have not been sufficient to fully describe the development of the genetic code starting from a prebiotic environment in which there was no coded assignment.

A fourth approach begins with Eigen's (1971) work on hypercycles, systems of mutually autocatalytic components. It considers the general question of under what conditions such a system can self-organize to a dynamic stability. The present research is on this path, looking at the genetic code and protein synthesis as a molecular information processing system, following the work of Bedian (1982) and Wills (1993). We have developed a simulation that models a replicating information storage, analogous to genes, that patterns the synthesis of functional components, which in turn catalyse the replication and synthesis processes. The purpose is to investigate what is required for such a model to self-organise into a stable autocatalytic coding system.

The initial phase of this research has followed on the work of Wills (1993) and Füchslin and McCaskill (2001) as described in more detail in Markowitz et al. (2006). This phase uses a discrete event step simulation to model what Füchslin and McCaskill call a GRT (Gene Replicase Translatase) system. Their system includes gene replication with mutation catalysed by a "replicase" protein, protein synthesis with coding catalysed by a "translatase" and two alternate translatase proteins that implement non-target codings.

Co→AA	**Translatase**	*** Target gene sequences**
00→W	WYZWZXWYYZYY	
00→X	WWXZWWWWYWYY	
00→Y	**ZYZZXZXZZWYX**	*** 100010101110111010010011**
00→Z	YZWYZWZYYYYY	
01→W	**ZYWYYXZXZWWZ**	*** 100001000011101110010110**
01→X	YYZZZXYWZYXY	
01→Y	WYXXXXWZZZYZ	
01→Z	XYXZWXYXZXZW	
10→W	ZZZWYWYXZZZX	
10→X	XWWXWZYYYZZY	
10→Y	XWYWZZYYYZZY	
10→Z	**XWYYWXYWWZYZ**	*** 110010100111000101100010**
11→W	XZWYZXZXYZWZ	
11→X	**WZXZWWWZWYXW**	*** 011011100101011001001101**
11→Y	WYYYZWYYXWYY	
11→Z	YZYWZXZZWWZW	
Replicase	**WYZWZZYXXWWY**	*** 010010011010001111010100**

Table 1: *An example coding. For each of the 16 possible translations from codons in the alphabet* {0,1} X {0,1} *to amino acids in* {W,X,Y,Z} *a random translatase sequence is chosen as a point in a 12-dimensional protein sequence space. Gene sequences are shown for one randomly selected coding.*

C→A	**Translatase**	**gene code1**	**gene code2**
0→Y	YYZYZZYY	00101100	
0→Z	YYYYYZZZ		11111000
1→Y	ZZZYZZZZ		00010000
1→Z	ZYZZYYZY	10110010	
Replicase	ZZZYYYZY	11100010	00011101

Table 2: *Both coding systems in the smaller model. For each of the 4 possible translations from codons in the alphabet* {0,1} *to amino acids in* {Y,Z} *a random translatase sequence is chosen as a point in a 8-dimensional protein sequence space. Gene sequences are shown for the two possible coding systems.*

My work extends research in this area by modelling a GRT system that is more complex and more realistic. Separate translatases for each codon-to-amino acid assignment are embedded as catalytic centres in the space of protein sequences. The reaction employs a separate translatase for each amino acid (like the real process) whereas Füchslin and McCaskill employed a single translatase for the entire synthesis of a new protein. Their model was embedded on a three dimensional lattice with one molecule per node and diffusion between nodes. My model also demonstrates stabilization through reaction-diffusion coupling, but uses a one-dimensional lattice of ”well-mixed” compartments containing variable numbers of molecules, with diffusion between compartments. In this, the results can be compared to research that has shown that a combination of chaotic flows and spatial constraints is required for evolutionary improvement of replication and translation efficiency (Scheuring et al., 2003).

Research to date

Starting with preliminary source code from Kay Nieselt and Peter R. Wills (unpublished), I have written a purpose-built C program that models genes and protein as 24-bit sequences contained in a closed one-dimensional loop of 1500 cells. Each cell contains a maximum of two five-gene “genomes” and 100 proteins. There are four types each of codons and amino acids. Simulations have been run for on the order of 10^{10} event steps, which takes several days running on a 3.4 GHz dual Xeon processor with 1Gb RAM. While a networked cluster was used for the simulation, each computer ran its own copy of the simulation with different parameters. The current software does not contain any parallel or distributed processing capability.

As described in Markowitz et al. (2006), at the start of the simulation sixteen 24-bit sequences are randomly selected to be the “catalytic centres” for protein translation, corresponding to the sixteen possible assignments of codon/amino acid pairs. Table 1 shows an example of one “coding system”, of a possible 24, which is a set of five catalytic centre proteins and five corresponding genes that implement a consistent set of mappings from codons to amino acids in this model. When the initial state contains random sequences of genes and proteins with a single genome that encodes a target coding system, the model eventually reaches a state in which the genes and proteins that implement the target coding system predominate.

Recent results

More recently, I have reduced the size of the model by using two types each of codons and amino acids in eight bit sequences of genes and proteins. The simplified model is initialised by selecting 5 random points in an 8-dimensional protein sequence space to be catalytic centres, which results in 2 possible coding systems. Table 2 shows an example of coding systems in such a model. Preliminary results indicate that these simplified systems demonstrate similar emergence of a dominant coding system. The simulation can be initialised with only random sequences and no preferred coding system, as the smaller sequence space can be fully explored within a feasible time scale. This result is significant, as it demonstrates the emergence from a random coding state of a simple coding system, which may then evolve to a more complex coding system as shown by (Wills, 2004). Fig. 1 shows the emergence of two competing “species” in such a system over 10^8 event steps.A future paper will explore the range of parameters in the model that allow the emergence of one dominant coding system and the maintenance of stability. The paper will also include results from computing the spatial autocorrelation of protein catalytic activity as a way of distinguishing the effects of reaction diffusion coupling from purely stochastic processes in the model.

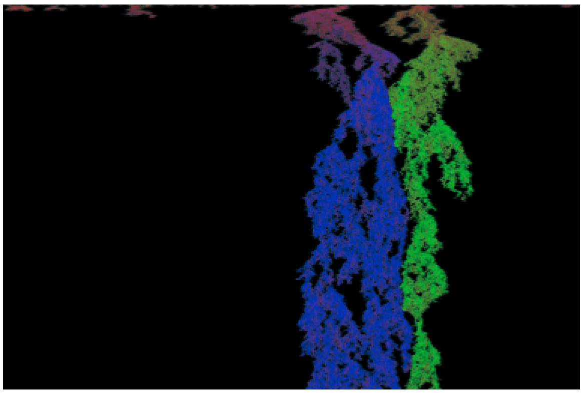


Figure 1: **Protein catalytic activity in compartments initialized using random sequences.** Each row shows protein activity in the 1500 compartments at time *t*, event 0 at the top. Only the first 10^8 event steps of the full simulation are shown in this figure. Red indicates the presence of protein sequences with low catalytic activity. Blue and Green indicate protein catalysts of the two possible coding systems. The initial random population distributed across compartments and with low catalytic activity can be seen to spread due to diffusion. More active proteins are seen clumped together as genes coding them replicate and produce protein faster than mutated genomes in other regions. Regions of genes in each coding system appear and compete. By 10^9 event steps in this simulation the system finds an apparent attractor state with one dominant coding system.

Future work

This paper describes a discrete event step simulation of a GRT system in one dimension that demonstrates an apparent attractor state for some range of parameters. Future work will extend the model in the following directions:

- Continue working with the discrete event step simulation, extending the dimensionality of the lattice to two and three dimensions.

- Explore the parameter space, looking for regions that lead to stability, and attempt to formulate general principles regarding stability in self-organising systems of this type.

- Investigate ways to simulate larger systems and for more event steps on clusters of computers, possibly by using optimistic asynchronous distributed discrete event simulation techniques (Lin and Fishwick, 1996).

- Derive systems of differential equations that characterise the model and apply numerical modelling methods to perform simulations, if possible using a software package such as XMDS (Collecutt and Drummond, 2001).

- Investigate models in which simple codes self-organise into increasing complexity, following the work of Wills (2004).

References

Bedian, V.: 1982, The possible role of assignment catalysts in the origin of the genetic code, *Orig Life* **12(2)**, 181

Collecutt, G. and Drummond, P. D.: 2001, Xmds: extensible multi-dimensional simulator, *Computer Physics Communications* **142(1-3)**, 219

Crick, F. H. C.: 1968, Origin of genetic code, *Journal of Molecular Biology* **38(3)**, 367

Eigen, M.: 1971, Self-organization of matter and evolution of biological macromolecules, *Naturwissenschaften* **58(10)**, 465

Füchslin, R. M. and McCaskill, J. S.: 2001, Evolutionary self-organization of cell-free genetic coding, *Proceedings of the National Academy of Sciences of the United States of America* **98(16)**, 9185

Knight, R. D., Freeland, S. J., and Landweber, L. F.: 1999, Selection, history and chemistry: the three faces of the genetic code, *Trends in Biochemical Sciences* **24(6)**, 241

Lin, Y. B. and Fishwick, P. A.: 1996, Asynchronous parallel discrete event simulation, *Ieee Transactions on Systems Man and Cybernetics Part a-Systems and Humans* **26(4)**, 397

Markowitz, S., Drummond, A., Nieselt, K., and Wills, P. R.: 2006, Simulation model of prebiotic evolution of genetic coding, in L. M. Rocha, M. Bedau, D. Floreano, R. Goldstone, A. Vespignani, and L. Yaeger (eds.), *ALIFE X: 10th International Conference on the Simulation and Synthesis of Living Systems*, pp 152–157, MIT Press, Bloomington, Indiana

Scheuring, I., Czrn, T., Szab, P., Krolyi, G., and Toroczkai, Z.: 2003, Spatial models of prebiotic evolution: Soup before pizza?, *Origins of Life and Evolution of Biospheres (Formerly Origins of Life and Evolution of the Biosphere)* **33(4 - 5)**, 319

Wills, P. R.: 1993, Self-organization of genetic coding, *Journal of Theoretical Biology* **162(3)**, 267

Wills, P. R.: 2004, Stepwise evolution of molecular biological coding, in J. Pollack, M. Bedau, P. Husbands, T. Ikegami, and R. A. Watson (eds.), *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, pp 51–56, MIT Press, Boston, MA

Exploring Better Techniques for Diagram Recognition

Rachel Patel

Exploring Better Techniques for Diagram Recognition

Rachel Patel
rpat088@ec.auckland.ac.nz

1. Introduction

Imagine having the ability to sketch a diagram on a Tablet PC and have that diagram recognised and converted to a formal diagram in a user specified format. Although computers can be used directly to produce such formal diagrams, in the beginning of a project, it can be better to use pen and paper. The reason is that these simple tools are more flexible, which encourages creativity, which in turn produce better designs in the end. However a computer offers easy editing and distribution features and greater formality to the look of a diagram and so sometimes the important pen and paper design stage is overlooked.

InkKit is a sketch tool that has been designed to bridge the gap between pen and paper and computers. InkKit allows you to sketch any type of diagram. Its recognition engine then identifies each component of the diagram and transforms it into a formal diagram in a specified format such as HTML or a Word document.

2. Motivation

The most important part of InkKit is being able to reliably recognise the diagrams. The purpose of my research is to investigate and refine the recognition algorithms that currently exist in InkKit.

One of the reasons why recognising a diagram is difficult, is because we are dealing with words and shapes at the same time. Figure 1, shows a typical user interface design that can be used to highlight this problem. For example, what is it that makes the circles in front of the words “Male” and “Female” circles indicating radio buttons and not the letter ‘o’?

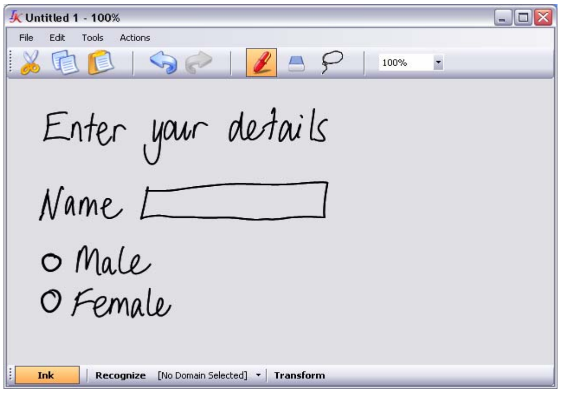


Figure 1. Sketch of a user interface design

The existing system automatically divides the ink into words and shapes, and then recognises the words using the Tablet Operating System recognizer and shapes using a variant of Rubine’s algorithm (Rubine 1991). Once the basic shapes and words have been recognised they are combined back together to suggest the most probable diagram component. This method of dividing the words and shapes is preferred over using a modal interface to separate the two as it preserves our natural tendency on pen and paper.

The intention of my research has been to concentrate on improving the algorithm that divides the diagram, simply because this is the area where we can expect the most improvement.

3. Method

In order to identify significant features of diagrams that may improve the divider, data from sketches have been collected and analysed. Sketches were gathered from 26 people. Each person completed a set of 9 sketches. The sketches were then processed within InkKit to obtain the required data for subsequent analysis.

In particular, the use of pressure and time data has been explored as this data is now more readily available to us and has not been investigated in the past. In regards to pen pressure, if significant differences exist in the pressure applied when writing words and when drawing shapes then this could be identified as an additional feature to improve the divider. In regards to time, the time between different types of strokes could prove to be valuable information. For example it is expected that there is less time between letter and word strokes than between shapes. This is because there is a smaller cognitive shift in thinking when writing.

In addition to this work on InkKit's divider, time data also has the potential to improve basic shape recognition. Research has suggested that pen speed is slower when drawing corners (Sezgin, Stahovich et al. 2001). This feature could help distinguish shapes with corners and those without for example recognizing the difference between circles and squares.

However my research has not been limited to pressure and time features only. Many already existing and possible features have been investigated to allow an overall picture of the significance of these sketch features to recognition.

4. Experimental Results

After data collection and analysis of various types of sketches the following empirical observations were made.

Pressure seems to increase with the duration of the stroke regardless of whether it is a shape or a word stroke. However shape strokes appear more stable in nature where the pressure increases at the beginning of the stroke, then stabilises, and eventually decreases at the end; whereas there are more oscillations in pressure for text strokes. This is illustrated in figure 2.

As for the time observed between letters, words and shapes, from the data collected there seems to be less time between letters and words than between shapes as expected.

The results from a comparison of the speed of rectangle and circle strokes has shown strong support for the hypothesis that there are large decreases in speed at

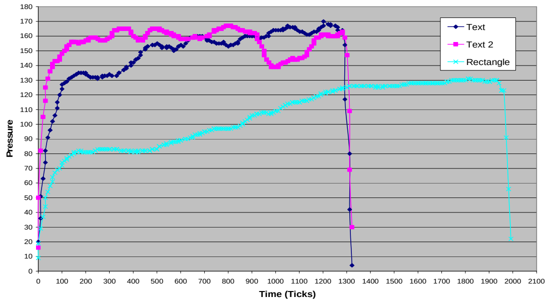


Figure 2. Pen pressure for text and shape strokes

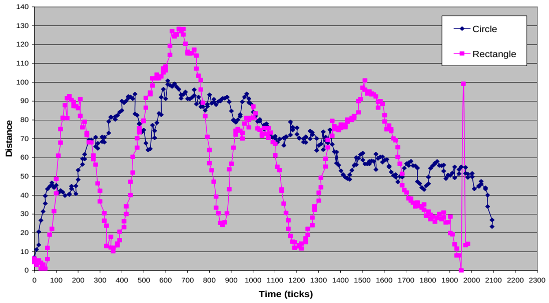


Figure 3. Speed of circle and rectangle shape strokes

corners and no such decrease when there are no corners. This is illustrated in figure 3 where we can see that there are 4 clear minima shown on the speed graph (excluding the one at the beginning of the stroke which marks the point where the pen is put down initially) which correspond to the corners of the rectangle and no such extreme minima for the speed of the circle.

After further statistical analysis using a partitioning technique, a decision tree has been constructed to divide strokes into shapes and words as shown in figure 4. The width of a strokes bounding box has been found to be the most significant feature separating word and shape strokes. Almost 85% of strokes in the dataset could be correctly classified using this feature alone.

Pressure was not found to be significant in dividing these strokes. However time on the other hand does appear in the decision tree as an important classifying feature. More specifically the speed and time to the next stroke has been found to be significant.

5. Future Work

This decision tree is being implemented into InkKit at present. It will then be evaluated against InkKit's existing divider and the Microsoft divider to determine if any improvement has been made.

In regards to improving basic shape recognition, further statistical analysis of the data is being carried out to establish the most significant features so that they can eventually be implemented into InkKit.

Acknowledgements

Thanks to Ross Ihaka for all his work towards the statistical analysis of my data and to my supervisors Beryl Plimmer and John Grundy for all their support.

References

Rubine, D. (1991). Specifying gestures by example. Proceedings of Siggraph '91, ACM.

Sezgin, T. M., T. Stahovich, et al.(2001). Sketch based interfaces: early processing for sketch understanding. Proceedings of the 2001 workshop on Perceptive user interfaces, Orlando, Florida, ACM Press.

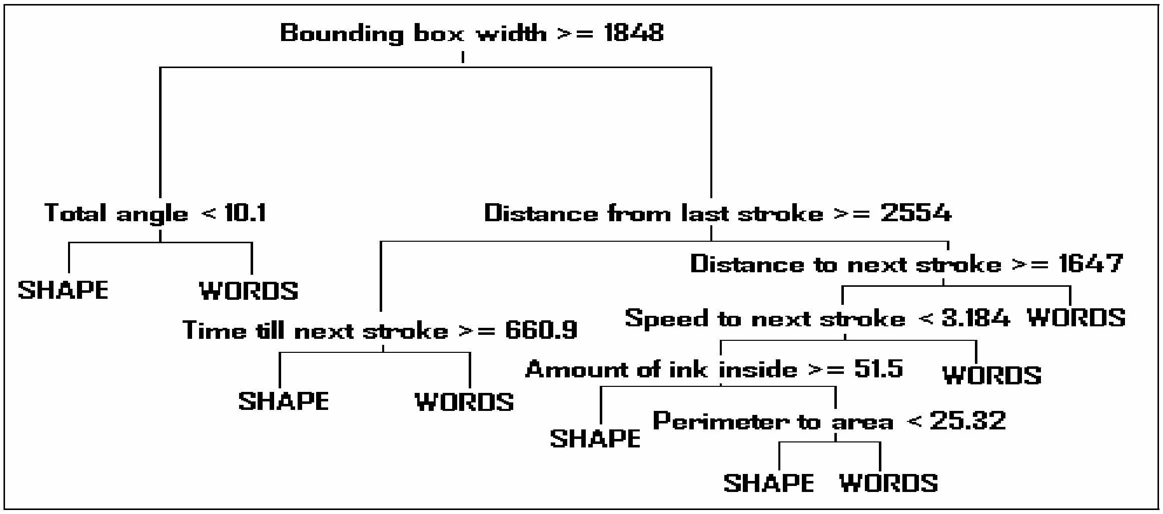

Bounding box width >= 1848
Total angle < 10.1
SHAPE
WORDS
Distance from last stroke >= 2554
Distance to next stroke >= 1647
Time till next stroke >= 660.9
SHAPE
WORDS
Speed to next stroke < 3.184
WORDS
Amount of ink inside >= 51.5
WORDS
SHAPE
Perimeter to area < 25.32
SHAPE
WORDS

Figure 4: Decision tree for word/shape divider

Applying Artificial Intelligence Techniques to the Game of No Limit Texas Hold ’em

Jonathan Rubin

Applying Artificial Intelligence Techniques to the Game of Texas Hold‘em.

Jonathan Rubin[1] and Ian Watson[2]
Department of Computer Science, University of Auckland, New Zealand
[1] jrub001@ec.auckland.ac.nz
[2] ian@cs.auckland.ac.nz

Overview

This extended abstract refers to research which is currently being undertaken as part of a Masters thesis into the investigation of artificial intelligence techniques applied to the game of poker. This research will focus specifically on a variation of poker known as Texas hold‘em (described below). Texas hold‘em provides a non-deterministic, hostile environment in which players must deal with incomplete information and uncertainty. Decision making in this type of environment has not been well addressed by A.I games research in the past and it is believed that advances made in games of this sort will also reap rewards in real-world problems as well.

Games and Artificial Intelligence

Games provide a well suited domain for artificial intelligence research. This is due to the fact that a game is usually composed of several well defined rules which players must adhere to. For a large majority of games the rules imposed are quite simple, yet the game play itself involves a large number of very complex strategies. This is especially true of games such as chess and checkers which offer opportunities to make very sophisticated and intricate plays. This statement is also true of the game of Texas Hold‘em and is nicely summed up by a popular quote coined by Mike Sexton which states “The name of the game is No Limit Texas Hold ‘em, the game that takes a minute to learn but a lifetime to master”. Another reason why games offer a beneficial environment for artificial intelligence research is the fact that goals and objectives of the game are clearly defined. This is advantageous to research as a performance metric is implicitly embedded in the game. Success can easily be measured by factors such as the amount of games won, the ability to beat certain opponents or, as in the game of poker, the amount of money won.

Up until recently artificial intelligence research has mainly focused on games such as chess and checkers. Successes like *Deep Thought*, *Deep Blue* and *Chinook* are usually the first to come to mind when contemplating A.I and games. Games such as chess, checkers and backgammon are classified *as two-person, zero-sum* games with *perfect information*. This means that there is one winner and one loser (zero-sum) and the entire state of the game is accessible by both players at any point in the game (perfect information), i.e. both players can look down upon the board and see all the information they need to make their playing decisions. These types of games have achieved their success through the use of fast hardware processing speeds, selective search, effective evaluation functions and better opening books and endgame databases. While these achievements are remarkable, their scope is rather limited. They offer little insight into other areas where A.I. techniques may be useful.

Games such as poker on the other hand are classified as stochastic, imperfect information games. The game involves elements of chance, the actual cards which are dealt, and hidden information in the form of other player’s *hole cards* (cards which only they can see). This ensures that players now need to make decisions with uncertain information present. This is still an open research question in the A.I community and research efforts are likely to be beneficial outside the realm of poker itself. For A.I. to be useful for most real world problems, challenges that imperfect information and a stochastic environment offers need to be addressed.

The Game of Poker

There are numerous variations of the game of poker available. The games differ by various aspects such as the number of *hole cards* dealt (cards which only the player can see and use to make their best hand), the number of community cards dealt (cards which all players can see and use to make their best hand), the order in which players bet and the limits imposed on a players bet. There are two variations which control the amount that a player may bet: *limit* and *no limit.* In a *limit* game player's bets are restricted to a certain amount; this amount usually doubles in later rounds of betting. Conversely, in *no limit* there is no restriction on the amount that a player can bet. A player's betting decision can be to *fold*, *check*, *call*, *bet* or *raise*. These are described below:

Fold: A player can *fold* their cards if they are facing a bet by another player, but they don't wish to match the bet. Once a player *folds* they are no longer involved in the current hand, but can still participate in any future hands.

Check/Call: When it comes time for a player to make his/her decision they can *check* if there have been no bets made by other players. *Checking* means the player does not need to invest any of their money into the pot to stay in the current hand. If, however, an opponent has made a bet then a player can *call* the bet by adding to the pot the exact value of the current bet. By contributing their own money to the pot they are able to stay in the current hand.

Bet/Raise: A player can invest their own money to the pot over and above what is needed to stay in the current round. If the player is able to *check,* but they decide to add money to the pot this is called a *bet*. If a player is facing a bet from an opponent, but instead of deciding to just *call* the bet they decide to add more money to the pot then this is called a *raise*.

The Game of Texas Hold 'em

In the game of Texas hold'em players are dealt two *hole cards* and five community cards are used in total. This strikes the right balance in terms of information availability (Harrington and Robertie, 2005) and offers opportunities for better strategic play than other poker variations allow for. Texas hold'em also offers a better skill-to-luck ratio than is offered by other forms of poker. An expert hold'em player has more of an advantage because the best hand holds up more often than in any other poker variation (Sklansky and Malmuth, 1994). Play in hold'em proceeds in the following stages: *preflop*, *flop*, *turn* and the *river*. These are described below:

Preflop: The game of Texas hold'em begins with each player being dealt two *hole cards* which only they can see. Betting order is determined by assigning one player at the table the status of *dealer*. Betting proceeds round the table in a clockwise manner. The minimum size of a bet is determined by the *big blind*. If a player, wishes to play then they must pay at least the *big blind* to enter into the pot. As long as there are at least two players left then play continues to the next stage. During any stage of the game if all players, except one, fold their hands then the player who did not fold his/her hand wins the pot (without having to reveal their *hole cards*) and the hand is over.

Flop: Once the *preflop* betting has completed three community cards are dealt. Players use their *hole cards* along with the community cards to make their best hand. Another round of betting occurs. The player classified as *dealer* is always the last to act (if the *dealer* is no longer in the hand the first active player to the right of the *dealer* becomes the last player to act). As long as there are at least two players left then play continues to the next stage.

Turn: The *turn* involves the drawing of one more *community card*. Once again players use any combination of their *hole cards* and the community cards to make their best hand. Another round of betting occurs and as long as there are at least two players left then play continues to the next stage.

River: During the *river* the final community card is dealt proceeded by a final round of betting. If at least two players are still active in the hand a *showdown* occurs in which both players reveal their hole cards and the player with the highest ranking hand wins the entire pot (if both players hold hands of the same value then the pot is split between both players).

Proposed Solution

Any attempt to develop a strong poker player needs to address many areas of the game. Several key components required for strong poker play have been identified (Billings et al, 2001). These include **hand strength**, **hand potential**, **betting strategy**, **bluffing**, **unpredictability** and **opponent modeling**. The strength of a particular hand and the potential strength of a hand needs to be determined given the *hole cards* that a player possesses, the current community cards, the type and number of opponents the player is up against and the likely cards these opponents might be holding. The ability to vary play is also an essential requirement for strong play. Any static strategy that is unable to adapt to the game conditions will be at risk of being exploited by strong opponents. Conversely, a strong player needs to be able to spot weaknesses in their opponents play and successfully exploit those weaknesses. This ensures the need for an *opponent modeling* component. Other issues such as *bluffing* (trying to deceive opponents about the strength of a hand by playing a weak hand strongly) and *slow-playing/trapping* (trying to deceive opponents about the strength of a hand by playing a strong hand weakly) also need to be considered. A strong player needs to know when *bluffing* or *slow-playing* may be successful and when they won't be, as well as knowing if an opponent is *bluffing* them or trying to *trap* them.

At the present the use of case-based reasoning is being considered to handle *opponent modeling*. Case-based reasoning attempts to solve new problems by reusing or adapting solutions to old problems (Watson and Marir, 1994). We believe this type of approach is well suited to *opponent modeling* as most players tend to not vary their play too much. Keeping track of how a particular opponent has been playing will provide useful information when making decisions about how to act against that opponent. We will also investigate a case-based reasoning approach to the *preflop* and *postflop* stage of the game. With this approach we hope to overcome deficiencies encountered in previous research (Billings, 2001) which employed the use of a static expert system. Case-based reasoning should be able to improve such an approach due to its ability to learn. We also hope to investigate other machine learning approaches such as using neural networks to predict which *hole cards* an opponent may be holding and using this information to inform a betting decision. It is hoped that with the combination of the above and other techniques we can construct a program that plays strong poker.

References

[1] Darse Billings, Computer Poker, M.Sc. research essay, University of Alberta, 1995.

[2] Darse Billings , Lourdes Peña , Jonathan Schaeffer , Duane Szafron, Learning to play strong poker, Machines that learn to play games, Nova Science Publishers, Inc., Commack, NY, 2001.

[3] D. Harrington and B. Robertie. Harrington on Hold 'em. Expert Strategy For No-Limit Tournaments. Volume 1: Strategic play. Two Plus Two Publishing, 2005.

[4] Schauenberg, Terence, Opponent Modelling and Search in Poker, M.Sc. thesis, University of Alberta, 2006.

[5] D. Sklansky. The Theory of Poker. Two Plus Two Publishing, 1992.

[6] D. Sklansky and M. Malmuth. Hold'em Poker for Advanced Players. Two Plus Two Publishing, 2nd edition, 1994.

[7] Watson, I., & Marir, F. (1994). Case-Based Reasoning: A Review. The Knowledge Engineering Review, Vol. 9 No. 4: pp. 355-381

A Software Tool for Integration of Colour-Coded Diffusion Tensor Data into a Neuro-Navigation System

Falk Uhlemann

A Software Tool for Integration of Colour-Coded Diffusion Tensor Data into a Neuro-Navigation System

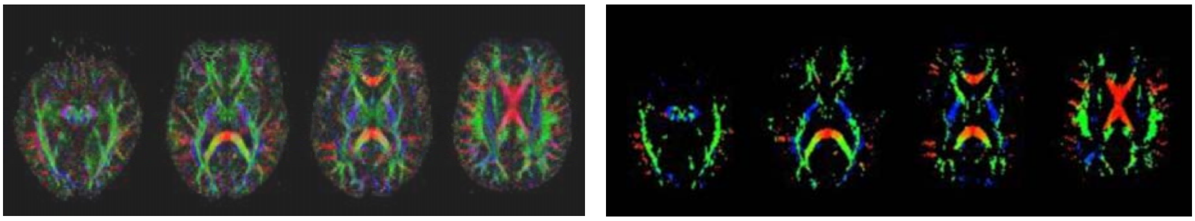
Falk Uhlemann

1 Introduction

Recently numerous new three-dimensional imaging techniques emerged in the field of biomedical imaging. One of these is diffusion tensor imaging (DTI) which allows to measure local anisotropic molecular diffusion processes (Le Bihan 2002). It is commonly assumed that these data provide information about shape and orientation of brain White Matter structures.
If applied to neurosurgical resection of brain tumours infiltrating white matter fibre structures, it may allow the sparing of eloquent fibre tracts resulting in a reduction of postoperative morbidity (Coenen 2003). Therefore a great demand exists concerning the transfer of DTI images into neuro-navigation systems, which allow to visualize the position of tracked neurosurgical instruments relative to preoperatively created image data during the surgical intervention. But commercially available systems do not support the integration of colour coded DTI data yet.

2 Method

Amongst various possibilities to visualise tensor data, i.e. the direction and value of the local vectors, is by colour hue and brightness respectively. Some software packages (e.g. DTI Taskcard, MGH, Boston, USA) support the export and transfer of the resulting true colour coded directional data in the commonly used DICOM format. In figure 1a slices created by an 1.5 Tesla Sonata scanner applying a diffusion weighted EPI sequence (TR/TE: 4000/129 ms; FOV: 230x230 mm; thickness: 2 mm; matrix: 1282; EPI-factor: 128; b-value 750, 6 direct.) on a healthy volunteer are shown.



a) Slices of original colour coded DTI data

b) Corresponding recoded DTI data slices

Figure 1 Original (DICOM) and recoded DTI data slices

The software developed during this project (dic2openmind) allows to convert this DICOM data into the so-called OpenMind format which can be read by the planning software of our BrainLAB neuro-navigation system. This special OpenMind format opens great possibilities for research purposes because it is very flexible and easy to implement. But currently it supports only grey value volume data. Therefore the idea was to approximate the original true colour (RGB) data by an indexed colour table which is available on the navigation system. From the colour maps preinstalled on the navigation system the so called “Rainbow” map was selected, because it contains the widest dynamic range of hue and brightness values. The necessary approximation of the original RGB values by the colours of the Rainbow map was performed using a nearest colour recoding algorithm (Floyd 1975, Thomas 1991). The result of this conversion step is visualised in figure 1b.

It can be seen that especially dim structures are not transformed well and that some hues appear to be different from the original data. These effects are due to the reduced number of colours in the transformed data (256 in the Rainbow map versus 256^3 = 16.777.216 in original RGB space).

To improve visualisation quality an interpolation is performed by the navigation system. Due to complexity, integration and stability concerns this interpolation step is not externally accessible. For usual linearly coded grey value data this does not impose any problem. But for colour coded (indexed) data this causes a conflict between the linear interpolation and the non-linear colour representation. Experiments demonstrated that the resulting interpolation/ visualization related artefacts can be significantly reduced by an increased resolution employing interpolation prior to colour coding which was therefore included in dic2openmind as a pre-processing step to increase the original volume size from 128x128x20 to 512x512x80.

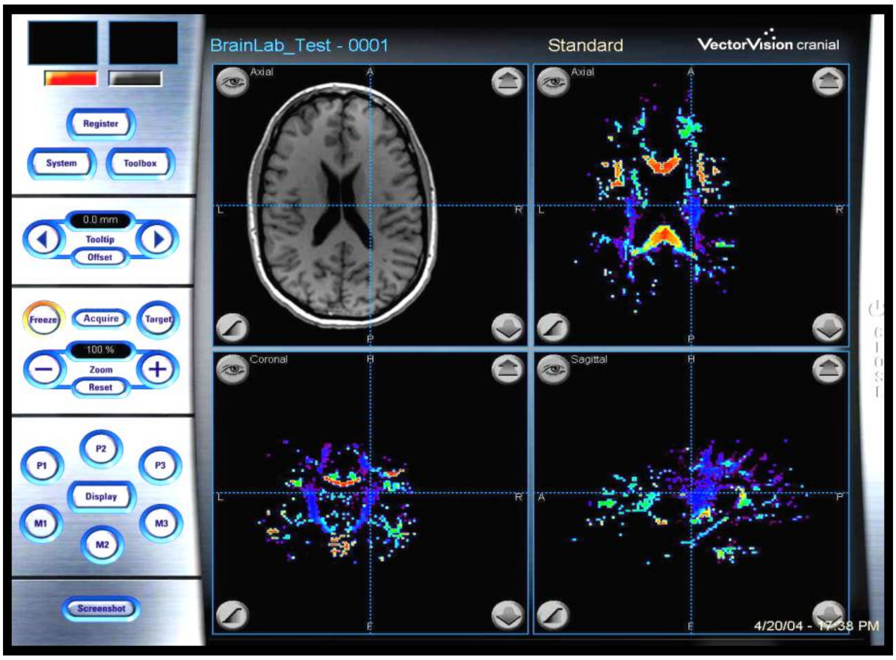


Figure 2 Display of the colour coded data on navigation system

To identify possible colour differences, both the original and the converted data as well as numerical difference images can be displayed to the user by the dic2openmind software for visual comparison. In the last conversion step the information provided by the transformed DICOM data is used to create an OpenMind compliant header and data file automatically.

After successful conversion, import of the converted data and registration with other imaging modalities, export for the navigation system can then be performed by the BrainLAB software PatXfer (OpenMind) and VVPlanning according to the standard procedures. Finally the corresponding Rainbow colour table has to be selected for the grey value coded tensor data to display it in colours in the navigations system's screen (see figure 2).

3 Results

This feasibility study project demonstrates the successful integration of colour coded data (such as DTI) into a commercially available neuro-navigation system for routine procedures. This data can then be easily used for planning and intra-operative navigation. Even though dim fibre structures can currently not be represented accurately, bright and saturated voxels of greater fibre tracts contain only minor transformation errors.

Several experiments demonstrated that interpolation and colour reduction artefacts due to indexing in the navigation system's colour map can be further reduced by a higher resolution of the original DTI data employing a prior external interpolation and a customized colour map. Integration of overlaid fibre structures in clinical routine would improve the interpretation of the anisotropy information significantly. Until this development becomes clinically available the presented method appears to be a well suited alternative.

4 Discussion

Because there exists only a limited number of colours in the colour maps currently available on the BrainLAB navigation system, only high intensity voxels are represented in the converted data accurately. According to our radiologists and neurosurgeons this seems not to be a disadvantage for surgical use since bright voxels are considered to represent locations with high anisotropic diffusion and are therefore the most important data points from the diagnostic point of view.

To evaluate the representation accuracy, a semi-quantitative approach by visualising deviations of the transformed from the original data set, was chosen. Colour information was coded by the hue, saturation, and value triple (HSV) colour space. In a colour coded DTI data set hue represents then the principle Eigenvector direction. This analysis revealed that structures with high anisotropy show a slight overestimation in value.

Bibliography

Coenen V A, Krings T, Weidemann J, Spangenberg P, Gilsbach J M, Rohde V: Diffusionsgewichtete MRT kombiniert mit navigiertem 3D-Ultraschall und fMRT zur Entfernung eines Kavernoms der Sehstrahlung. Zentralbl Neurochir, vol. 64, 2003, pp. 133-137.

Floyd RW, Steinberg L: An adaptive algorithm for spatial grey scale. Int. Symposium Digest of Technical Papers. Soc. for Information Displays, 1975, p. 36.

Le Bihan D, Zijl, P: From the diffusion coefficient to the diffusion tensor. NMR Biomed., vol 15, 2002, pp. 431-434.

Thomas SW: Efficient inverse colour map computation. Graphics Gems II, Academic Press Boston, 1991.

Acknowledgement

I would like to thank the foundation "Studienstiftung des Deutschen Volkes" for the financial support, Hagen Kitzler, Annett Werner, George Ogrezeanu for their valuable insights, data and feedback, Greg Sorensen and Ruopeng Wang, MGH, Boston, USA for providing the DTI Taskcard software and Ute Morgenstern as well as Ralf Steinmeier for the great support throughout this project. For further information please contact: imaging@uhlemannweb.de