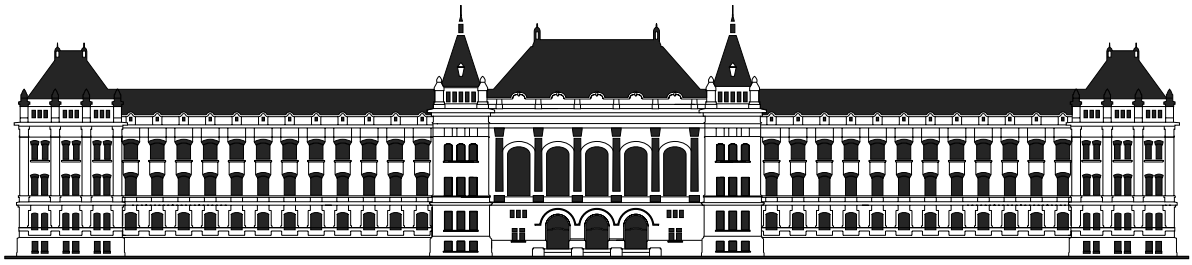
PROCEEDINGS
OF THE
13TH PHD MINI-SYMPOSIUM

FEBRUARY 6–7, 2006.
BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
BUILDING I, GROUND FLOOR 017.



BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
DEPARTMENT OF MEASUREMENT AND INFORMATION SYSTEMS

Head of the Department: Prof. Dr. Gábor PÉCELI

Conference Chairman:
Béla PATAKI

Organizers:
Ágnes ERDŐSNÉ NÉMETH
Péter CSORDÁS
Dániel DARABOS
János LAZÁNYI
István PILÁSZY
Attila SÁRHEGYI
András SZÉLL

Homepage of the Conference:
http://www.mit.bme.hu/events/minisy2006/index.html

Sponsored by:
IEEE Hungary Section (technical sponsorship)
Schnell László Foundation
ChipCAD Electronic Distribution Ltd. (www.chipcad.hu)

ISBN 963 420 853 3

FOREWORD

This proceedings is a collection of the extended abstracts of the lectures of the 13th PhD Mini-Symposium held at the Department of Measurement and Information Systems of the Budapest University of Technology and Economics. The main purpose of these symposiums is to give an opportunity to the PhD students of our department to present a summary of their work done in the preceding year. Beyond this actual goal, it turned out that the proceedings of our symposiums give an interesting overview of the research and PhD education carried out in our department. The lectures reflect partly the scientific fields and work of the students, but we think that an insight into the research and development activity of the department is also given by these contributions. Traditionally our activity was focused on measurement and instrumentation. The area has slowly changed during the last few years. New areas mainly connected to embedded information systems, new aspects e.g. dependability and security are now in our scope of interest as well. Both theoretical and practical aspects are dealt with.

The papers of this proceedings are sorted into seven main groups. These are Biomedical Measurement and Diagnostics, Information Mining and Knowledge Representation, Intelligent and Fault-Tolerant Systems, Machine Learning, Measurement and Signal Processing, Embedded Systems, Model-based Software Engineering. The lectures are at different levels: some of them present the very first results of a research, because some of the first year students have been working on their fields only for half a year. The second and third year students are more experienced and have more results.

During this thirteen-year period there have been shorter or longer cooperation between our department and some universities and research institutes. Some PhD research works gained a lot from these connections. In the last year the cooperation was especially fruitful with the Vrije Universiteit Brussel, Toshiba R&D Center, Kawasaki, Computer and Automation Research Institute of the Hungarian Academy of Sciences, Budapest.

We hope that similarly to the previous years, also this PhD Mini-Symposium will be useful for both the lecturers and the audience.

Budapest, January 17, 2006.

Béla Pataki

Chairman of the PhD Mini-Symposium

LIST OF PARTICIPANTS

Participant	Advisor	Starting Year of PhD Course
Márta ALTRICHTER	Gábor HORVÁTH	2005
András BALOGH	András PATARICZA	2004
Péter BOKOR	András PATARICZA	2005
András BÓDIS-SZOMORÚ	Tamás DABÓCZI, Alexandros SOUMELIDIS, Zoltán FAZEKAS	2005
Károly János BRETZ	Ákos JOBBÁGY	2003
Péter CSORDÁS	Ákos JOBBÁGY	2004
Dániel DARABOS	Gábor HORVÁTH	2004
Péter DOMOKOS	István MAJZIK	2003
Ágnes ERDŐSNÉ NÉMETH	András PATARICZA	2004
András FÖRHÉCZ	György STRAUSZ	2004
László GÖNCZY	Tamás BARTHA	2003
Gábor HAMAR	Gábor HORVÁTH, Tibor VIRÁG, Zsuzsanna TARJÁN	2005
Gábor HULLÁM	György STRAUSZ, Péter ANTAL	2005
Zsolt KOCSIS	András PATARICZA	2004
Dániel László KOVÁCS	Tadeusz DOBROWIECKI	2003
László LASZTOVICZA	Béla PATAKI	2003
János LAZÁNYI	Béla FEHÉR	2004
Zoltán LUDÁNYI	Gábor HORVÁTH	2005
András MERSICH	Ákos JOBBÁGY	2004
Zoltán MICSKEI	István MAJZIK	2005
András MILLINGHOFFER	Tadeusz DOBROWIECKI, Péter ANTAL	2004
Károly MOLNÁR	Gábor PÉCELI	2003
Tamás NEPUSZ	Fülöp BAZSÓ, György STRAUSZ	2005
István PILÁSZY	Tadeusz DOBROWIECKI	2004
Attila SÁRHEGYI	István KOLLÁR	2004
Péter SZÁNTÓ	Béla FEHÉR	2004
András SZÉLL	Béla FEHÉR	2004
Ákos SZŐKE	András PATARICZA	2003
Gábor TAKÁCS	Béla PATAKI	2004
Balázs TÓDOR	Gábor HORVÁTH	2004
Norbert TÓTH	Béla PATAKI	2003
András ZENTAI	Tamás DABÓCZI	2004

Program of The MINI-SYMPOSIUM

Embedded Systems

Biomedical Measurement and Diagnostics

Intelligent and Fault-Tolerant Systems

Machine Learning

Information Mining and Knowledge Representation

Measurement and Signal Processing

Model-based Software Engineering

Conference Schedule

Time	February 6, 2006	Time	February 7, 2006
8:30	Conference Opening Opening Speech: Gábor Péceli Embedded Systems	8:30	Information Mining and Knowledge Representation
10:00	Biomedical Measurement and Diagnostics	10:00	Measurement and Signal Processing
Launch break		11:40	Model-based Software Engineering
13:30	Intelligent and Fault-Tolerant Systems		
16:00	Machine Learning		

PARALLEL SORTING ALGORITHMS IN FPGA

András SZÉLL
Advisor: Béla FEHÉR

I. Introduction

Sorting algorithms have been investigated since the beginning of computing. There's a lot of well-known and thoroughly analyzed, optimized algorithms for different problem sets, with different average runtimes and memory needs. For specific input data and specific computer architectures, there isn't a general "best of all" algorithm, sorting algorithms like quick sort that tend to be fast on average, may be less efficient in special cases.

In this paper I describe some major sorting algorithms and architectures taken into consideration and present a method to create fast, hardware-efficient and deterministic sorting solution for a specific data set in molecule data clustering on a Virtex-II FPGA. As data clustering is an often applied method in bioinformatics, efficient sorting is a vital intermediate step.

II. Sorting architectures

Most sorting algorithms consist of three main steps: read, compare and store – sorting is a memory intensive task. For finding a good sorting solution, memory operations has to be performed fast and the algorithm must be smart enough to reduce the number of necessary operations. As general purpose processors have extremely fast L1 and L2 caches and clock frequencies several times higher than in any processor implemented in FPGA, their sorting performance can be achieved only by higher level of parallelism in embedded applications.

For parallel operations, there are several different possibilities [1]. Sorting networks (see Figure 1) have a great performance but only for limited data count, as their hardware requirements increase exponentially. Sorting with mesh-connected processor arrays is an interesting method, but only for limited data sizes. Another way for parallel sorting is using several processors and a more sophisticated data distribution mechanism. These processors can be either MicroBlaze or PicoBlaze processors (according to the complexity of the sorting algorithm), a Virtex-II 6000 may implement 8-12 MicroBlaze processors and much more PicoBlazes.

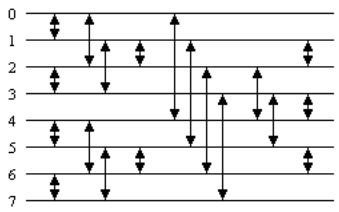


Figure 1: Sorting network

All these solutions are limited by the speed of the input/output memory and the hardware resources of the FPGA, so for the optimal performance, these attributes and the type of the input data has to be considered.

III. Input data and hardware limitations

The input data is a 256 megabyte set of 64 bit elements, as on Figure 2. Data has to be sorted in A, R, B order; storage also follows this order. This way either 32/64 bit integer comparisons or bit-wise comparisons can be carried out. The algorithm which generates the data performs a bucket sort on index A with a bucket size of 128 indexes, and in our case sorting by B index is unnecessary when using stable sorting algorithms ([2], e.g. merge sort).

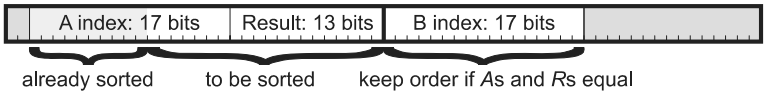


Figure 2: Input data format

Input data is stored in a 100 MHz, 64 bit wide external SDRAM memory, while the parallel sorting process can use the internal block RAMs and distributed RAMs. In Virtex-II 6000 there are 144 block RAMs with 18Kbit in each block. They are best utilized as 512x64 bit dual port modules by coupling them, this way a 100 MHz clock speed can be easily achieved with double speed for read operations. Another external memory is present for storage of partial results.

Input data has been thoroughly analyzed by simulations on data sets extracted from real molecule databases. The deviation of the number of pairs at a specific A index is much higher than the deviation in buckets of 128 A indexes, resulting in sub-optimal hardware utilization, though due to the greater element count, sorting will take much more time in the latter case. Merge block sizes are set according to the results of the analysis.

IV. Parallel merge sorting

The main concept of the recursive merge sort is the following: divide the unsorted elements into two equal size sublists, sort the sublists and merge the two sorted lists into one.

Merge sort is an *O(n log(n))* linear algorithm [2]. (Real parallel algorithms may have an asymptotic complexity of *O(n)*, but they need extreme hardware resources, so they are not feasible for our data set, where data is read in from a linear memory.) In general it is slower than quicksort, but there is smaller difference in its best and worst case run lengths unlike in quicksort, making hardware timing optimizations easier, and its simplicity results in a smaller hardware complexity. Merge sort needs *2n* memory for sorting *n* elements – a big disadvantage that has to be addressed as memory limitation is a bottleneck of the sorting procedure.

The proposed algorithm is a two-round merge sort with a previous 8-element sorting network to reduce recursive complexity. Unsorted data is written into the 32 merge cells via this net; then in $log_2(512/8)=6$ merge cycles, data is sorted by these units in parallel (each cell sorts its 512 elements in 2x512 places). Storing the results of the cells is done via a binary tree which sorts the 32*512 elements on the fly; while storing the results, the next set of data is read into the empty half of the merge cells. After sorting each 16384-element block of a bucket, these blocks are merged from the temporary memory and the sorted bucket is written over the unsorted input.

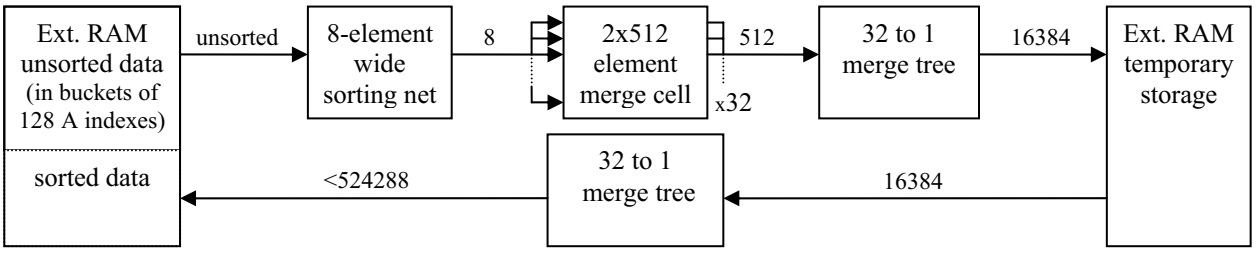


Figure 3: Sorting steps – numbers between sorting steps show the size of sorted blocks

V. Conclusion

The proposed architecture sorts the input data in 1 memory read cycle (33M steps), 1 memory write cycle (33M) and 6 merge cycles per 16K-blocks (>6.3M steps) time, with deterministic run times. The architecture depends on input data properties and hardware features, but the algorithm itself is much more general. The objective to get the sorting speed of Pentium 4 class processors on Virtex-II FPGAs was reached.

References

[1] H.W. Lang, *Sequential and parallel sorting algorithms*, Fachhochschule Flensburg, 2000, URL: http://www.iti.fh-flensburg.de/lang/algorithmen/sortieren/algoen.htm

[2] *Sorting algorithm*, Wikipedia, 2005, URL: http://en.wikipedia.org/wiki/Sorting_algorithm

ANTIALIASING IN SEGMENTED RENDERING

Péter SZÁNTÓ
Advisor: Béla FEHÉR

I. Introduction

The market for 3D graphics accelerators can be roughly divided into three segments: workstations, desktop systems and embedded systems. The former two segments have quite a lot of common requirements and parameters, but embedded systems are a little different. However, antialiasing is a feature which is compulsory everywhere; in embedded systems due to the still limited screen resolution (which only increases about 10% a year). There are many different steps in 3D rendering where incorrect sampling causes annoying aliasing effects, this article concentrates on antialiasing the edge of objects.

In Section II. some feasible solutions are introduced briefly, while Section III. reviews how these algorithms fits into a segmented renderer.

II. Antialiasing methods

As mentioned in the introduction, the goal of edge antialiasing is to reduce the artifacts caused by the finite screen resolution (jagged object edges, polygon popping).

A. Supersampling

Supersampling is probably the most straightforward option for antialiasing: the frame is rendered at multiplies of the screen resolution and then scaled down to the final resolution, typically with simple box filtering. The clear advantage is that this method not only reduces edge aliasing, but also effectively removes other sampling problems, such as texture aliasing. However, rendering the frame at a much higher resolution increases computational and storage – memory – requirements considerably.

B. Multisampling

Multisampling also uses some sort of oversampling, but in contrast to supersampling, only geometry is sampled with higher resolution, color – and therefore textures – are handled only with the final screen resolution. So, coverage and depth tests are done at a higher resolution, generating the visibility test result with subpixel accuracy. Further parts of the rendering pipelines can use this information to compute the color of a given pixel. When all subpixels belonging to a pixel are covered by a single object, color is computed only once for the pixel center. In all other cases different color values are computed for the subpixels and averaged together to form the final output value.

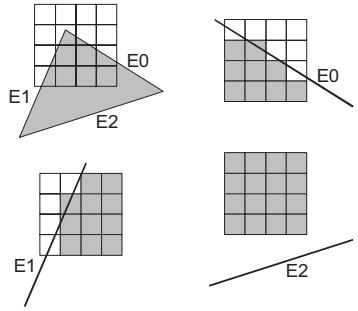
Compared to supersampling, multisampling requires the same processing power for covering determination and depth tests, but a lot of shading (output color computing) calculation is saved. Picture quality wise supersampling produces better results as this method does not reduce texture aliasing. However, texture aliasing can be reduced with much less performance hit by other techniques.

C. Multisampling with object detection

It is obvious that multisampling still does much more coverage and depth computation than what is really necessary. To further reduce requirements the following multi-pass algorithm can be used. First, visibility testing is done with the screen resolution. During this pixel resolution testing,

adjacent pixels are inspected to find out if they are not covered by the same object. In the second pass, a small area around these places is tested with higher resolution. The main problem is the correct identification of the object edges.

D. Coverage mask



As opposed to the former methods, aliasing with coverage mask is a kind of pre-filtering algorithm. A look up table (LUT) stores a subpixel resolution binary mask for every possible edge orientation and distance from the pixel center. Addressing this LUT according to the three edges forming the triangle and using bitwise AND on the three results gives a binary string containing exactly as many '1's as the number of covered subpixels. Figure 1 shows an example where every pixel is divided into 16 subpixels. Small rectangles represent subpixels while the big rectangle is a single pixel. For every triangle edge, subpixels marked with gray are the positions where the coverage mask contains '1'.

III. Antialiasing in hardware

Supersampling and multisampling requirements are the same in the unit responsible for removing hidden surfaces: visibility and coverage tests should be done at a higher resolution. Compared to traditional rendering architectures, using these kinds of antialiasing does not increase the required external memory requirements, as visibility testing is done using on-chip buffers. However, performance decreases linearly with the number of subpixel samples taken into account. Using a small, programmable unit to compute triangle edge functions and depth value increments allows using arbitrary number of subpixels and arbitrary subpixel positions (of course, as long as the internal buffers have enough capacity to store the subpixels).

Using coverage mask is quite different. Although performance wise it may be a good option (performance does not decreases linearly with the number of subpixels), it has several drawbacks. When using supersampling or multisampling, only the sign of the edge functions are required to determine coverage; with coverage mask, the distance between the pixel center and the edge also plays a role – therefore edge functions have to be normalized, which requires division. Another drawback is the required memory to store the coverage masks – as the proposed rendering architecture has several Hidden Surface Removal Units working in parallel to increase performance, all of them need a separately addressable coverage mask LUT. And finally, despite larger number of subsamples may be used, quality may not be higher, as depth test is only done once per pixel. This means that the color value of a single object can be weighted according to its coverage, but no other objects influence the final color.

IV. Conclusion

When concentrating on the HSRU part of the 3D pipeline, for segmented rendering, multisampling may offer the best compromise. It does not increase external memory bandwidth requirement, allows selecting different performance/quality trade-offs by using flexible sampling numbers, and fits well into the rendering pipeline architecture.

References

[1] T. Akenine-Möller and J. Ström, *"Graphics for the Masses: "A Hardware Rasterization Architecture for Mobile Phones"*, ACM Transactions of Graphics, Vol. 22., Issue 3., 2003.

[2] D. Crisu, S.D. Cotofana, S. Vassiliadis, P. Liuha, *"Efficient Coverage Mask Generation for Antialiasing"*, Computer Graphics International, 2004

MEMORY PROFILING BASED HARDWARE – SOFTWARE CO-DESIGN IN FPGA

János LAZÁNYI
Advisor: Béla FEHÉR

I. Introduction

Embedded software design methodology is merging from various hardware dependent assembly languages towards the well-defined and commonly used C. Most algorithms are evaluated with standard PCs using high-level programming languages. There is a natural need to use this code in the design too. During the implementation phase hardware and software components must be separated from each other with a well defined interface description. The goal is usually to put the most frequently used and/or resource demanding processes into dedicated hardware accelerator and let a general microprocessor do the rest. Today's FPGAs enable to use the same design methodology while either they include a hard-macro processor core or alternatively they have enough resources to implement soft-core controller(s) beside the dedicated hardware accelerator unit(s).

There is an intense research in the field of hardware / software co-design in FPGAs. Most papers present techniques on high-level source code *Data Flow Graph* (*DFG*) or instruction level profiling. [1][2] Both approaches use bottom-up design methodology to evaluate the complexity of the design, based on instruction level information. After a certain level of system complexity the DFG becomes un-manageable, while many of the graph search algorithms are *NP*-hard problems. These methods are really good in finding the optimal implementation of a dedicated algorithm, but fail to analyze the total complexity of a design.

There are several attempts, to enable automatic RTL level synthesis from high level source code. The first approach is to use a smaller subset of C language (Handel-C) or use the common C/C++ functions for primarily simulation (System-C). In this case, we can use many standard functions, but unfortunately the code is not always synthesizable.

In this paper I will introduce a top-down design methodology to assign a dedicated part of the algorithm to a *Computational Unit* that can be optimized later on with the above described techniques.

II. Difficulties on FPGA implementation of C algorithms

C language had been developed to run on standard Neumann architecture computers, and has no native support for other architectures (like Harvard-architecture, *Very Long Instruction Word*, *Single Instruction Multiple Data*) that are commonly used in embedded systems. One big disadvantage is that there is no "universal" language syntax for describing parallelism. In contrast to this FPGAs can be configured freely allowing thousands of parallel *Computational Units*.

Usual high-complexity, computational demanding algorithm's (e.g.: mpeg video decoding) source code contains hundreds of functions. Let us declare that the code is optimal, optimally separated into functions, and the execution time is the most important constraint. We could gain the fastest execution time in dedicated hardware (like FPGA), if we allocate each function to a dedicated hardware or software Computational Unit that is optimal. This is usually not possible, because we have limited resources; and not necessary due to the data dependencies in the program. Another optimal solution is: if we find all the function dependencies, and ensure that data are ready when it is necessary.

A. Pointers

C language has a pretty efficient construct called pointers or references. As there is only one Computational Unit, and all data are stored physically in the same chip (or chip array) we can call functions without moving operands and results. All we have to transfer is a pointer to the array we would like to use. If we implement the same function call in FPGA, and the functions are assigned to two different Computational Units we have two choices, we share memory area among these two Computational Units and transfer the pointer or alternatively we copy the output data of a subsequent function to the antecedent. Sharing memory among two Computational Units is relatively easy with Dual-Ported BRAMs, registers, or direct wiring but the limited number of FPGA resources restricts the use of them. Anyhow, the best choice is always depending on many other parameters too, like: When does the function use the transferred variable first, how often this function call happens.

B. Stack

Compiling C code to microprocessors will transfer the operands and results through the stack during a function call, used as a scratchpad. (If we don't take compiler optimization into account) Stack is also the place for the local registers. During simple FPGA implementation there are nothing like global-, local variables or stack, we can use only BRAMS, registers and wires to transfer values, that is why we have to dedicate a register for each transfer.

C. Heap

One of the biggest challenges is to track the heap usage. As the memory allocation is done during run-time, we can only rely only on run time profiling information. Fortunately heap is usually used as working buffers among programmers.

D. Library functions

Transferring library functions (like floating point arithmetics, other mathematical functions, I/O functions etc.) into the FPGA is also a key issue, but these questions are out of the scope of this document. One implementation possibility is to use an embedded microprocessor core, running parts of the code. If a compute intense part is covered by these parts, we can also accelerate the execution, with other options like: instruction set extension or dedicated hardware accelerator.

III. Recommended process for C code analyses

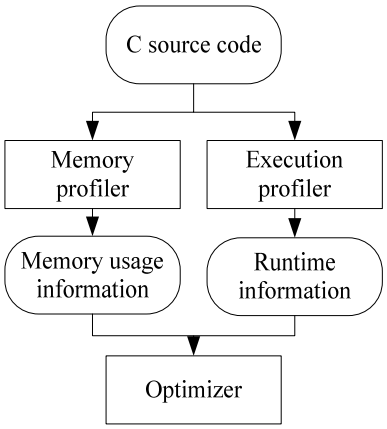


Figure 1: Design methodology

We can build a memory access profiler that tracks the stack usage, heap allocation, local and global variable accesses per function basis. We have to monitor the memory cells accessed by references during code execution and mark them as "accessed" if they have been read or "dirty" if they have been modified. This information can be combined together with standard profiling information containing the distribution of CPU usage of each function.

Based on these two information, we can optimize the function communication inside FPGA, and decide the physical implementation technique. (Figure 1.)

References

[1] S. Wuytack et al., "Flow graph balancing for minimizing the required memory bandwidth.", *Proc. of the Int'l Sympopsium on System Synthesis*, pp. 127--132, Nov. 1996.

[2] Celoxica website, *http://www.celoxica.com/*

MICROCALCIFICATION ASSESSMENT USING JOINT ANALYSIS OF MAMMOGRAPHIC VIEWS

Márta ALTRICHTER
Advisor: Gábor HORVÁTH

I. Introduction

In screening X-ray mammography two different views of both breasts – CC „head-to-toe" and MLO a lateral oblique view – are captured. In the four X-ray images some special signs of cancer (mainly microcalcifications and masses) are looked for. In computer aided cancer detection the first step is to analyse the individual images. However as breast cancer detection using X-ray mammography is an ill-defined problem, obtaining high hit rate while keeping the number of false positive detections low is extremely difficult. [1] To achieve better results we try to reassess calcification clusters found by preliminary algorithms on individual images by joint analysis of the the mammograms.

There are two ways of joint analysis: the two views from the same breast and the similar views of the two different breasts can be analysed jointly. This paper proposes a simple new procedure for the joint analysis of one breast's two views. The procedure is based upon the experiences of radiologists: masses and calcifications should emerge on both views, so if no matching is found the given object is almost surely a false positive hit.

One of the major signs of cancer is microcalcification cluster. Microcalcifications have a higher X-ray attenuation coefficient than the normal breast tissue therefore they appear as a group of small localized granular bright spots. (See Fig. 1) Calcification clusters do not have significantly different texture characteristics than the surrounding tissues, therefore no texture analysis is made, calcification clusters on the two views are only matched according to their position. To have a frame system of the breast to position within, we made a simple, "2.5 D" position system.

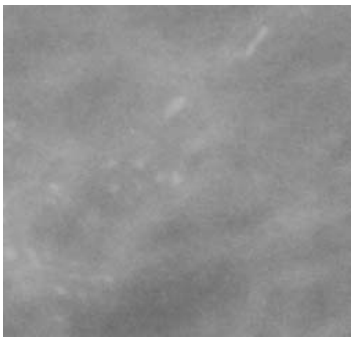


Fig. 1.: Microcalcification cluster

Fig 2.: Mask of the cluster

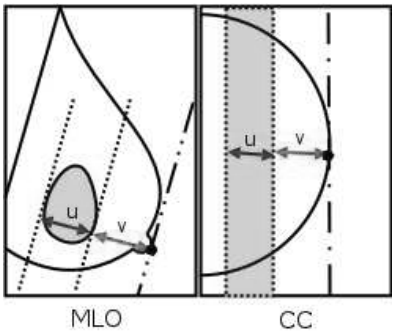


Fig. 3.: Positioning system for positioning

II. Assessment of Calcification Detection Algorithm's Result

The preliminary calcification detection algorithm developed in a combined CAD project of Budapest University of Technology and Economics and of Semmelweis Medical University produces Calcification Masks from suspicious regions (see Fig. 2.) [2]. To further refine the output of the preliminary algorithm a probability value is assigned to each cluster. This value represents the probability of the cluster to be a true positive (TP) hit. We want to develop a method which assigns low probability value to false positive hits (FP) and high values to TP clusters.

The probability value is determined by a neural network. 35 parameters are extracted from a cluster as an input vector to the neural network. These parameters include minimum, maximum and average distance between calcifications; min., max., avg. calcification intensity; intensity difference between calcifications and surrounding tissue ... The preliminary algorithm produced 48 TP clusters (radiologists marked the region as a calcificated region) and 862 FP ones. 60 clusters made up of 30 TP and 30 FP clusters were used for teaching the net (10–20–1 neurons in the three layers) with early stop based on a test set having 10-10 TP-FP clusters. The net classifies FP hits to -1 and TP hits to 1. We can chose a threshold (like 0), and consider the remaining range (like 0-1) equally distributed to be the probability of being a TP hit.

The results were validated on the remaining 8 TP clusters and 812 FP hits. All the TPs were kept by the net (assessed to be above 0) and 436 FP hits were dropped.

These results look promising but as the number of TP cases are so low (more run of the preliminary algorithm will increase it in the future) these results need further analysis.

III. Positioning System

In X-ray mammography perfect 3-D reconstruction is impossible due to deformation. There exist some methods to reconstruct 3-D breast to certain extent, but these methods need additional informations (like the distance of the compression plates during X-ray screening), therefore our main aim was only to build a simple "2.5-D" positioning system. CC and MLO views are both two-dimensional projections of the three dimensional object. Therefore a stripe will correspond to a region on the other image. (See Fig. 3.)

For positioning in the breast three landmarks are named in the literature: the pectoral muscle, the nipple and the boundary of the breast. The position of the nipple is determined by laying a tangent on the breast border parallel with the pectoral muscle in MLO views, and parallel with the vertical axes in CC views. The pectoral muscle was gained by separating it from other edges in an edge map provided by Edgeflow algorithm [3]. The stripe corresponding to a region on the other view is established by measuring up the distances of the region's two boundary pixels – from the nipple perpendicular to the pectoral muscle – to the nipple of the other view (see Fig 3).

The probability accompanied to a calcification cluster is modified with the area ratio of the stripe corresponding to it and of other calcification clusters found on the other view: p'=p-c1*(1-area_rat).

Matching of calcification clusters was tested on 188 cases (376 pairs of mammographic images). At this time the probability assessment of the cluster was not according to the method described in Section II., but with a much simpler formula including calcification number and intensity information. The combination of the methods in Section II and III need further future analysis.

During matching of the 188 cases (1/3 TP and 2/3 FP cases), 3.17% of true positive cases were lost (96.8% kept), while the original 0.8% of found normal cases increased to 12.4%.

IV. Conclusion

The first results show that using this approach (inspired by skilled radiologists) the number of false positive detections can be reduced significantly while the decrease of true positive hits is relatively small.

References

[1] G. Horváth, J. Valyon, Gy. Strausz, B. Pataki, L. Sragner, L. Lasztovicza, N. Székely: "Intelligent Advisory System for Screening Mammography", *Proc. IMTC 2004,* Como Italy, 2004, Vol. I. pp. 2071-2076.

[2] Gábor Horváth, Béla Pataki, Ákos Horváth, Gábor Takács, Gergely Balogh: "Detection of Microcalcification Clusters in Screening Mammography", *Embec 2005*, Prague, Czech Republic, 20-25 Nov, 2005

[3] R.J. Ferrari, R. M. Rangayyan, J. E. L. Desautels, R. A. Borges, A. F. Frére: Automatic Identification of the Pectoral Muscle in Mammograms *IEEE Trans on Im. Proc.* Vol. 23, No 2, pp. 232–245, February 2004.

MEDICAL IMAGE PROCESSING IN DETECTION OF BREAST CANCER

Zoltán LUDÁNYI
Advisor: Gábor HORVÁTH

I Introduction

Breast cancer is one of the most frequent cancers and the leading cause of mortality for women, affecting almost one eighth of them and giving one third of cancers. Evidences show that X-ray mammography is the only reliable screening method giving nearly 95% chance of survival within 5 years due to early detection. [1]
Due to the huge number of images captured per year and the high number of false positive diagnoses done by doctors (80-93%), development of mammographic decision support systems is under heavy research.

II Importance of joint analysis

During X-ray screening two different views are captured of each breast: a CC (cranio-caudal) from above and a leaning lateral view, the MLO. The two most important symptoms are microcalcifications (small spots that have high intensity compared to their environment) and masses (big, high intensity blobs). Our results and other publications on this topic show that obtaining high hit rate while keeping the number of false positive detections low is extremely difficult. [1]
Microcalcification and mass detector algorithms developed in our department have quite a good hit rate (near 95%) but the false positive hits per image are 3 per image in the case of microcalcifications and 6 in the case of masses. [2,3] This rate can be achieved if – in addition to the individual analysis – joint analysis of the images is done similar to the way done by radiologist experts. Finding a pair to a mass- or microcalcification-candidate can increase the probability that the hit is a true positive one.
Since in X-ray mammography perfect 3-D reconstruction is impossible due to breast deformation, we implemented a simple "2.5-D" positioning system between CC and MLO images for this joint analysis. This means that we can assign a stripe on the MLO image to every mass-candidate on the CC image and vice versa. The stripe is based on the position of the nipple and the angle of the pectoral muscle. According to this reference system we could make a hypothesis: "the *distances* of a mass (measured from the tangent that is parallel to the pectoral muscle and placed in the nipple) in the CC and MLO pictures are equal".
The correctness of the reference system and our hypothesis were tested by a statistical analysis. Results showed that the assumption was correct though there is some variance caused by the failures of the algorithm, wrong radiologist assessment or the flaw of the hypothesis (because of breast deformation) for a few cases. To compensate these effects the width of the stripe can be increased by a constant or by a number relative to the width of the stripe to counteract the deviation of the algorithm.

Since masses have characteristic texture, the reference system can be improved by textural analysis. This is done through the following steps. First the image is segmented by EdgeFlow [5], and then texture features are calculated for each segment, k-means clustering is applied to these features resulting a better segmentation. Once we have a good quality segmentation, we recalculate features for the new segments. After these preliminary steps we establish the reference system. Then for each mass-candidate we do pairing by computing the corresponding stripe and – based on texture

features – we choose the most similar segments within the stripe. (Note that based on some rule-based laws this pairing may result no pair at all. This decision mainly contains size, intensity, texture similarity based rules.)

Results proved that in the case of microcalcifications by loosing 3.1% of true positive hits we can gain a decline of 13.1% in false positive hits, while in the case of masses we found these numbers to be 4% and 13.3% respectively. (Performance in the latter case should be better due to the textural analysis but since false positive hits are higher, so is the probability of random pairing.) [4]
As it can be seen from the results above – it proved to be true that such an analysis can improve performance but since running times are high, the approach cannot be used yet. Therefore lately we started focusing on quickening algorithms while keeping performance at least at the same level. This is partly a technical, partly substantive question since for better running times it is not enough to use clever coding techniques but some algorithms had to be substantially modified. (However – faster running has one benefit. There are always some free parameters that affect performance, and therefore should be carefully tuned. With less running time a more exhaustive testing can be done during the same amount of time.)
Up to the time of present article running times declined to half, while – according to the tests – performance is still as good as it was. This means about 10 minutes of running time per pairs of images.

III Using Our Algorithms on Chest X-ray images

Lung cancer is also an important factor of mortality and (since it affects both genders) is an even more important factor in the medical budget. Since some of our algorithms can be applied for general image processing, trying our algorithms on these images seemed to be a rational step.

EdgeFlow and the segmentation algorithm based on it seemed to be useful in analysis of chest X-ray images. EdgeFlow has proved to be useful for edge detection, however our experiments and national publications show that directional filters can better be used for this aim since a priori information is given in fields where such an algorithm is needed (eg. bone identification). We found the same in the case of segmenting the lung from the background. A simple radix transformation-based algorithm implementation gives just as good results as EdgeFlow with much better running times.

References

1 L. Tabár: "Diagnosis and In-Depth Differential Diagnosis of Breast Diseases" Breast Imaging and Interventional Procedures, ESDIR, Turku, Finland, 1996.
2 M. Heath, K. Bowyer, D. Kopans, R. Moore, K. Chang, S. Munishkumaran and P. Kegelmeyer "Current Status of the Digital Database for Screening Mammography" in Digital Mammography, N. Karssemeier, M. Thijssen, J. Hendriks and L. van Erning (eds.) Proc. of the 4th International Workshop on Digital Mammography, Nijmegen, The Netherlands, 1998. Kluwer Acamdemic, pp. 457-460.
3 G. Horváth, J. Valyon, Gy. Strausz, B. Pataki, L. Sragner, L. Lasztovicza, N. Székely: Intelligent Advisory System for Screening Mammography, *Proc. of the IMTC 2004 – Instrumentation and Measurement Technology Conference*, Como, Italy, 2004, Vol. pp.
4 Ludányi Zoltán: Mammográfiás képek együttes kezelése (foltdetektálás) – diplomamunka. BME MIT, 2005.
5 Wei-Ying Ma, B. S. Manjunath: EdgeFlow: A Technique for Boundary Detection and Image Segmentation. *IEEE Trans. on Image Processing*, Vol. 9, No 8, pp. 1375–1388, August 2000.

NEW ASPECTS OF THE ESSENTIAL TREMOR MEASUREMENT

Károly János BRETZ
Advisor: Ákos JOBBÁGY

I. Introduction

The present work was designed to describe measuring procedures, review devices and complex equipment, developed partly in our laboratory. Further purpose of this study is to summarize our results of investigating and evaluating tremor, body sway and stress characteristics.

Tremor is defined as an involuntary, rhythmic shaking movement of some part of the body. Essential tremor is a chronic condition characterized by its frequencies and amplitudes, occurring most typically on the hands and arms. Affected persons may have difficulties in performing fine motor manipulations. A possible classification of tremor, using the dominant frequencies, may be summarized as follows:

3 - 4 Hz	tremor of cerebellum origin
4 - 6 Hz	essential tremor* -, Parkinson tremor-, and tremor of muscle tone origin,
6 - 12 Hz	essential**-, physiological-, and in stance occurring tremor, as well as tremor of muscle tone and / or psychic origin.

* In older age (above 65 years), ** in younger age.

In accordance with the conclusion of Spyers-Ashby and Stokes (2000) we found reliability of repeated measurements of normal physiological tremor. In case of the essential tremor we are confronted with some difficulties. Examining the state of art it can be mentioned that according to the up-to-date information of the WEMOVE Institution (204 West 84th Street, New York, NY 10024) "there is no definitive test for essential tremor".

The result of our novel investigation makes it possible to propose a multi-parametric approximation for testing essential tremor and model these phenomena. The essence of the matter is the monitoring of paroxysm character of the essential tremor. Our results and findings suggest formulating theses (claims) on our multi-channel measuring system and the collected data.

II. Methods

Our complex investigations on physiological and essential tremor, as well as body sway, include different techniques: e.g., device for work ability investigation, finger force measuring device, accelerometer, force platform, stabilometry system, laser gun and pointer, 2 D movement analyser system, The equipment is completed by a special device for heart rate and physiological stress measurement, namely CardioScan (Energy Lab. Technology GmbH, Germany), as well as a force and disjunctive reaction time measuring instrument.

The results of patient interviews and the responses to questionnaires of Spielberger help to measure the disability, and the impact on quality of life. Amplitude/intensity and frequency, measures of severity of tremor have a direct correlation with functional disability.

A special device containing a microcontroller and model of a miniature screw-driver was constructed for screening of candidates, applying for jobs in the precision instrument industry. Accelerometer adapter was adapted to a microcomputer for simultaneous investigations of the hand tremor. The actual stress reaction of the tested person was measured. Relationship between the stress level and hand tremor was examined.

In the experiments participated sharpshooters (n=14), three investigations were absolved within one year, engineer students (n=22), psychiatry patients in two investigations (n1+n2=11+26=37), neurology patients (n=12), elementary school teachers (n=92), physical education students (n=27).

III. Excerpts of results (averages)

Junior sharpshooters: body sway in Romberg position r = 5,2 mm, with gun aiming: r (g) = 3,5 mm. "Competition" stress reaction S = 32,2 % on 0-100 % scale.

Engineer students: disjunctive reaction time t = 274 ms, stress reaction S = 29 %, moderate physiologic tremor.

Psychiatry patients: disjunctive reaction time t = 390 ms, stress reaction S = 15 %, due to the drug therapy. Essential tremor was measured.

Neurology patients: body sway: r = 18 mm, stress reaction S= 49 %. Essential tremor was observed.

Teachers: Stress reaction: S = 23 %.

Physical education students: disjunctive reaction time t = 273 ms, stress reaction: S = 22 %.

Significant correlation was found between tremor and emotional excitement.

Remarks: "body sway" means "vibration of the centre of mass",
"stress reaction": measurable physiological effect of external or internal stresses,
"disjunctive reaction time": reaction time measurement, performed with two kinds of photo stimulation, using green and red LED-s, which light up stochastically.

IV. Discussion

Several methods have been proposed in the literature for clinical evaluation. It is typically based on a thorough study ofmedical history and assessment of the patient's symptoms including:

Topography of the body's areas affected by tremor as follows fingers, hands, arms, legs, head, chin, trunk, voice.

Tremor type and appearance condition.

Amplitude/intensity and frequency-i.e., measures of severity that have a direct correlation with functional disability.

Analysing our experimental results we could find and discriminate effects on the measured parameters: effects of the circumstances (e.g. competition and university examination), effects of drug therapy (at patients), and the effect of the high level motivation (at engineer and physical education students), on the level of the recorded parameters.

References

[1] Bretz, K.J. Sipos, K. (2003) Tremor and stress during college examination. Kalokagathia. Budapest. Vol. 41. No.1. 111–115. p.

[2] Bretz, K.J., Skripko, A.D., Sipos, K., Bretz, K. (2004) Measurement and correlation analysis of hand tremor, body sway, stress and anxiety. III. International Conference on Medical Electronics. Ulashika, V.S. and Batury, M.P. (Eds): Medelektronika. Minsk. 217-222 p.

[3] Bretz K.J. (2005) A testlengés és a kéz tremor méréstechnikája. Híradástechnika. Vol. LX. 2005/4. 18–21. o.

[4] Bretz, K.J., Jobbágy, Á. (2005) Evaluation of hand tremor. T he 3rd European Medical and Biological Engineering Conference. EMBEC'05. Prague, November 20-25, 2005. ISSN: 1727-1983 -2005. IFMBE. 1-4 p.

[5] Jobbágy, Á., Harcos, P., Karoly, R., Fazekas, G. (2005) Analysis of the Finger Tapping Test. Journal of Neuroscience Methods, Vol 141/1 pp 29-39.

[6] Spyers-Ashby J.M., Stokes M.J. (2000) Reliability of tremor measurements using a multidimensional electromagnetic sensor system. Clinical Rehabilitation. 14 (4), 425-432.

ACCURATE BLOOD PRESSURE MEASUREMENT FOR HOME HEALTH MONITORING

Péter CSORDÁS
Advisor: Ákos JOBBÁGY

I. Introduction

We elaborated a new method for short-term continuous blood pressure monitoring based on the classical Riva-Rocci technique and on photoplethysmographic (PPG) signal. The method assures better characterisation of blood-pressure than the most popular oscillometric measurement. The drawbacks of the oscillometric technique are discussed, demonstrating its inaccuracy. A new possibility to assess the cardiovascular state of the patient has also been found.

As a result of a co-operation, this research was supported by measurements made on a physical model of the arterial system. The model has been built at the Department of Hydrodynamic Systems [1].

The new methods are being implemented in a device for home health monitoring. This instrument stores the recorded physiological signals (ECG, PPG) for medical postprocessing. In addition to the new parameters, some well-known important indices (e.g. heart rate variability, augmentation index) can also be calculated. The algorithms for the calculation of relevant indices are under development.

The possibility for *continuous* blood pressure monitoring is investigated. After presenting the measurement methods, I will analyze the typical pressure signals recorded using the physical model.

II. The oscillometric blood pressure measurement

The vessels have special pressure-volume characteristics. The higher is the pressure loading of the wall, the lower is the compliance ($C=\Delta V/\Delta P$). This means that if the pressure outside the vessel is set equal to the inner mean pressure, the vessel diameter changes caused by pressure waves are maximal. In other words, the oscillations in the occluding cuff's pressure are maximal at minimal transmural pressure, i.e. if the cuff-pressure is equal to mean arterial pressure (*MAP*).

The oscillation amplitudes are directly related to C which is a well defined mechanical parameter of the vessel. However, to the best of our knowledge the methods used in practice use only some discrete points of the oscillation curve. Evaluating the whole shape could be a new possibility to describe the state of the arteries.

It is evident that with the oscillometric method, only an instantaneous *MAP* value can be measured. The diastolic and systolic blood pressure is *estimated* by means of statistical data. However, the potential users of our instrument can have a cardiovascular state differing from "statistical standard". We try to consider this by examining of other physiological parameters. The disappearance of the PPG signal measured distal to the upper arm-occluding cuff indicates the systolic pressure. Our newest results prove that the diastolic pressure can be determined by means of tracing the delay between ECG and PPG signals.

Our measurements done at Semmelweis University demonstrate that after an appropriate re-scaling, the PPG signal can be interpreted as a short-term continuous blood pressure curve. Namely, a tonometer, as the most accepted non-invasive blood pressure measurement device, gave a pressure signal nearly coincident with the scaled PPG curve. Although the occlusion of the upper arm during measurement acts as a considerable intervention in the measured system, evaluating the PPG signal recorded before the occlusion can solve this problem. For the scaling however, determining of two

coherent blood pressure values (e.g. systolic and diastolic pressure) is needed. Because of the variability of blood pressure, this is not a trivial task.

III. Materials and Methods

We have made numerous records with participation of young and elderly healthy subjects. The pressure of a cuff on upper arm, the signal of PPG sensor fixed on fingertip, and the Einthoven I. ECG have been recorded and processed with MATLAB. However, the evaluation our new algorithms based on these records is difficult, because of the continuously changing physiological effects, like breathing.

The physical model of the arterial system built at the Department of Hydrodynamic Systems is made of highly flexible silicon elastomer tubes and artificial vessels. The heart is modelled by a membrane pump, the arm tissue is substituted with fiber-like fluid-filled elastic tubes. During the slow deflation of the cuff, the intravascular pressure was recorded at three locations of the arterial network: at the heart (aorta), at the shoulder (arteria axillaris) and at the thigh (arteria femoralis). These recordings were also processed with MATLAB.

IV. Evaluating the oscillometric method by means of the physical model

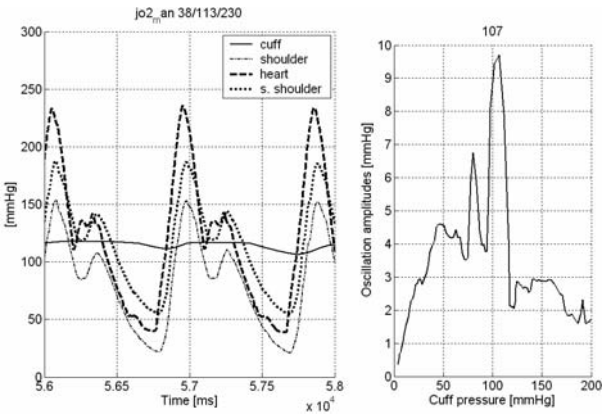


Figure 1: Oscillometric blood pressure measurement on the physical model

Fig. 1. demonstrates the typical result of an oscillometric blood pressure measurement executed on the physical model. The pressure values are out of the physiological range. This implies that the statistical methods for the calculation of systolic and diastolic pressure fail. The shape however, is realistic. The pressure values (diastolic/MAP/systolic) are: 38/113/230 mmHg. The oscillometric algorithm gives 107 mmHg, which is acceptable considering the fact, that the maximum of the oscillometric amplitudes reaches almost 10 mmHg. This indicates at the same time, that the appropriate decomposition of the cuff pressure to static and dynamic components is a critical part of the algorithm. Using an envelope for the oscillation points could also increase the accuracy.

V. Future Plans

Further examination of the oscillometric method can be done as described above. By means of the physical model, the cross effects between the measured parameters can be analyzed. Replacing parts of the arterial network the stiffening of vessels will be simulated. Applying a PPG sensor, the PPG scaling can be validated. After the prototype of the home health monitor is finished, measurements with participation of patients suffering from cardiovascular diseases can be done.

References

[1] Ferenc Molnár, Sára Till, Gábor Halász, “Arterial blood flow and blood pressure measurement an a physical model of human arterial system”, Hozman J., Kneppo P. (Editors). IFMBE Proceedings, Vol. 11. Prague: IFMBE, 2005. ISSN 1727-1983. (Proceedings of the 3rd European Medical & Biological Engineering Conference - EMBEC´05. Prague, Czech Republic, 20-25.11.2005)

COMPUTER AIDED EVALUATION OF CAPILLARY MICROSCOPIC IMAGES

Gábor HAMAR
Advisors: Gábor HORVÁTH, Tibor VIRÁG, Zsuzsanna TARJÁN

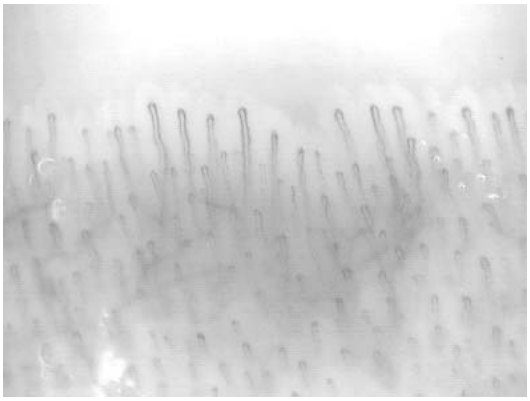
I. Introduction

Capillary microscopic examination means examining the smallest vessels of the human organ, the capillaries. The peripheral blood circulation is very sensible for certain illnesses e.g.: autoimmune diseases, diabetes. In many cases the deformations in the blood circulation can be observed before other symptoms, therefore capillary microsopic tests play an important role in the early identification of the diseases.

In our presentation we will introduce the first results of a computer aided evaluation system. The aim is to develop computer algorithms, which can process capillary microscopical images automatically or with little human intervention, furthermore to give definitions of objective and quantitative measures with which subjective attributes can be replaced. In addition we also would like to create a cheap and widely applicable appliance with which high quality images can be created.

The presentation describes attributes of the capillary microscopical images in short, reviews the main steps of image processing, the difficulties, the edge-detecting algorithm that is used for detection capillaries, and the first results of the validation.

A. The main attributes of capillary microscopical images



1. figure Healthy pattern

The capillary microscopical pattern of a healthy subject can be seen on figure 1. In the picture the vessels are ordered into rows, their shape is regular “hairpin”, pointing to the same direction. Every capillary has two parallel stems: a thinner called arterial section, and a wider called venous section. They are connected with a winding part called apical section.

The most important parameters that can be extracted from the picture:

- The arrangement of the vessels
- Sizes of the hairpin (length, distance of the two stem, diameters)
- Shape of capillaries
- Linear density
- Occurrence of micro hemorrhages
- Visibility of SVP (Subpapillary Venous Plexus)

In certain diseases the healthy pattern changes. In many cases the regular arrangement breaks up. If the vessels become dilated, giant- or megacapillaries come into existence. The hairpin shape can also be changed: the medical literature classifies the modified shapes into the following groups: meandering, bushy, ball, tortuous and ramified. The linear density decreases in general, in certain cases micro hemorrhages can be observed, and the visibility of the SVP increases.

II. Computer aided evaluation

A. The method of image recording

There is a generally accepted method for capillary microscopical examinations, hence we have also followed this method [1], [2], [3]. Our present database was created with a stereo microscope. We used paraffin oil for increasing the transparency of the skin, and a light source (intralux 6000) with cold light. The direction of the light was approximately 45º.

B. Image processing

The aim of the computer support is to support image storage and evaluation. At this time only the first steps have been solved. We have already developed an input system, with which images and other information can be stored in a directory structure. We have already implemented an edge-detecting algorithm for locating capillaries and we have given a solution for the following problems:

- Separation of the tree sections of capillaries
- Measurement of linear capillary density
- Detection of micro hemorrhages

C. Test method

We checked two parameters: detection of vessels and the line selected by the algorithm for measuring linear capillary density. We used two sets of test images: a set of high quality images (8 images), and a set of lower quality images (50 images).

D. Results

For the lower quality images 84.8% of the vessels have been detected, the line selected for linear density measurement is good in 86.4% of the test cases, and not optimal but acceptable in 10%. For the higher quality set 95.6% of the capillaries have been detected, the selected line is good in 37.5% of the cases, acceptable in 45.8%, and not acceptable in 16.7%.

III. Conclusion

Considering the test results currently our system can not be used for fully automatic density measurement, but it is capable for capillary detection, hence it can be used with human control.

References

[1] A. Bollinger, B. Fagrell, Clinical Capillaroscopy, Hogrefe & Huber Publishers, ISBN 0-88937-048-6

[2] P. Dolezalova, S. P. Young, P. A. Bacon, T. R. Southwood, Nailfold capillary microscopy in healthy children and in childhood rheumatic diseases: a prospective single blind observational study, Annals of the Rheumatic Diseases pp. 444-449, 2003.

[3] Zs. Tarján, É. Koó, P. Tóth, I. Ujfalussy, Kapillármikroszkópos vizsgálatok, Magyar Reumatológia, vol. 42, pp. 207-211, 2001.

[4] J. C. Russ, The Image Processing Handbook, Fourth Edition, ISBN 0-8493-1142-X, CRC Press, 2002.

[5] Open Source Computer Vision Library, Reference Manual, URL: http://sourceforge.net/projects/opencvlibrary/

[6] V. Székely, Képfeldolgozás, id. 55067, publisher: Műegyetemi kiadó, Budapest, 2003

NON-INVASIVE ASSESSMENT OF BLOOD VESSEL PROPERTIES

András MERSICH
Advisor: Ákos JOBBÁGY

I. Introduction

Aortic stenosis, thrombosis and stroke are common fatal diseases which could be predicted if a cheap, easy-to-use, non-invasive measuring method was available that estimates the state of arteries. Up to date ultrasound Doppler is the only such reliable way. Goal of this research is to asses the biomechanical properties of blood vessels via a modified photoplethysmographic (PPG) signal. An equivalent electrical model is being introduced which simulates the pressure and flow transients in the aorta and the left arm. Effects of a cuff wrapped around the upper arm are also included. The cuff-induced changes in flow conditions are measured by PPG sensor on a fingertip. The diagnosis comprises of the estimation of model parameters from the measured data.

II. Materials and Methods

A. Measurement Set-up

Photoplethysmographic signals originate from optical reflections (or transmission) of blood vessels. Volume changes in an artery generated by the pulse alter the reflecting conditions of tissues according to Lambert-Beer law. They are estimated from the arterial pressure by means of non-linear function compliance (in first approach treated as a 2/2 polynomial). Commercial PPG sensors operate in a frequency range of 0.1-20 Hz. Detection of the relatively slow transients however requires a measuring device with a useful bandwidth of 0-40 Hz. Before any diagnosis could be made a custom built PPG had to be constructed. The special pressure profiles used in the research made the development of a regulator for cuff inflation and deflation also necessary.

B. Electrical Model of Circulation

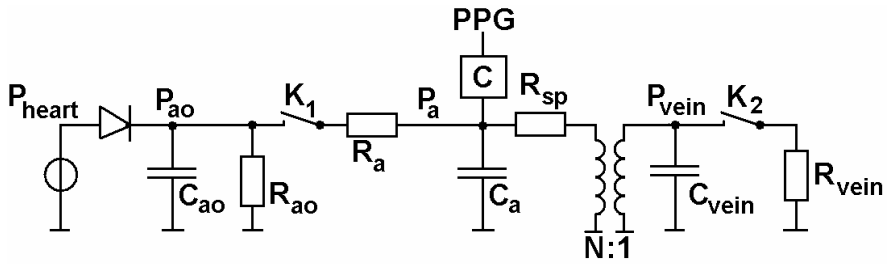


Figure 1: Simplified model of circulation

Figure 1 is a simple electrical model of the heart, aorta and the left arm artery and veins. In this representation pressure is replaced by voltage, flow by current. Capacitors represent the buffer effect of aorta, arteria brachialis and veins. Diode stands for the valve, transformer for the capillary net and resistors for flow-resistance. The pressure of the cuff is qualified into 3 different states: pressure below the venous pressure, between the venous and the systolic pressure, over the systolic value. The switches K1 and K2 are controlled according to this quantization: both closed, K1 closed but K2 open, both open. The model neglects BP auto-regulation and assumes that during measurement the vessel-properties and P_{ao} are unchanged. Viability was confirmed by comparative measurements on a physical circulation-model at the Department of Hydrodynamic Systems [1].

III. Results

Identification of the model parameters comes from two different measuring protocols. In the first one the cuff was fast inflated above the systolic pressure, held there for about 15 sec and then fast deflated. In terms of the abstract model this means the turning off and after 15 sec the simultaneous turning on of both switches. Figure 2a shows the measured data. The second cuff-intervention applied, as presented in Figure 2b, is fast inflation until 50 mmHg (below diastolic), holding it for about 40 sec, then fast deflation.

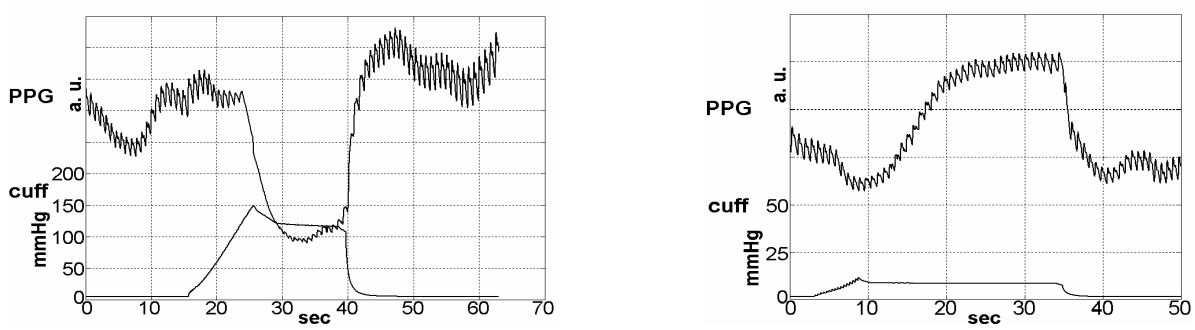


Figure 2: Measurements 1 and 2; response of a healthy patient to the excitation. PPG, cuff pressure.

The identification itself is an iterative algorithm (A detailed description can be found in [2].):

1. Measure the mean arterial pressure (MAP) by means of oscillometric method and set the initial compliance polynomial linear.
2. Transform the PPG signals to pressure values via compliance polynomial.
3. Apply ARMAX frequency domain identification for measurement data 1 according to $P_1 = \frac{1}{s} MAP \frac{s^2\alpha + s\beta + \gamma}{s^2 + s\delta + \varepsilon}$ and determine the model parameters α, β, γ_1, δ_1 and ε_1.
4. Apply ARMAX frequency domain identification for measurement data 2 according to $P_2 = \frac{1}{s} MAP \frac{s^2 + s\delta + \gamma}{s^2 + s\delta + \varepsilon}$ and determine the model parameters γ_2, δ_2 and ε_2.
5. Change the compliance parameters in a way that the gradient of estimation error, $\left[(\gamma_1, \delta_1, \varepsilon_1) - (\gamma_2, \delta_2, \varepsilon_2)\right]^2$, becomes negative.
6. If estimation error is exceptional small then STOP, else GOTO step 2.

IV. Conclusion

A simple model of circulation including heart, aorta and left arm was developed; effects of a cuff wrapped around the upper arm were investigated. Two different cuff pressure profiles were used: one blocking the whole circulation of the arm the other impeding only back flow through the veins. A measurement set-up comprising of a DC-coupled PPG sensor and a cuff-pressure controller was constructed. Model parameters were estimated from PPG signals recorded on a fingertip. The method makes evaluation of six circulatory parameters (R_a, R_{sp}, R_{vein}, C_a, C_{vein}, N) for every individual patient possible. As future work the switches should be replaced by components able to represent continuous cuff pressure and the compliance should be described by a more suitable function.

References

[1] F. Molnár, S. Till, G. Halász, "Arterial blood flow and blood pressure measurements on a physical model of human arterial system," in Conf.*Proc. of the IFMBE EMBEC'05*, paper no. 2043 on CD-ROM. ISSN: 1727-1983, Prague, Czech Republic, November 20–25 2005.

[2] A. Mersich, P. Csordás, Á. Jobbágy, "Non-invasive assessment of aortic stiffness," in Conf.*Proc. of the IFMBE EMBEC'05*, paper no. 1426 on CD-ROM. ISSN: 1727-1983, Prague, Czech Republic, November 20–25 2005.

SIMULTANEOUS LOCALIZATION AND MAPPING

Balázs TÓDOR
Advisor: Gábor HORVÁTH

I. Introduction

A key problem of the area of autonomous robot navigation is that the robots have no knowledge about their position and heading. The present article introduces a novel approach to this so called localization problem.

II. Problem formulation

Let's assume that we have a robot equipped with some cheap range sensors, one motor on each rear wheel, and that it has some computing capacity as well. Now, let's give the robot something to do: let's send it to a room nearby. Finding a path between the starting point and the destination is an easy task, since there are several well known path planners that can operate in such continuous environments.

However, following the calculated path is much harder, because we have no explicit knowledge about the robot's position and heading, and these pieces of information have to be calculated using the range sensors' signals.

In order to help this calculation, the measurements are stored in a memory in a structure called the world model. There are three popular concepts in this area: browsing through the literature one can find topological, metric and hybrid methods.

The topological world models are based on graphs that store the relevant points of the environment as nodes and paths between them as arcs. They are hard to build, because they represent relatively high level information as opposed to the metric ones. However, this can be an advantage when it comes to path finding and tracking, since navigation is much easier on graphs. The most obvious reason is that such world models are lacking the mass of lower level information that can be overwhelming for a navigation algorithm.

On the other hand, the metric models represent the lowest level of information that can come out of the sensors. The simplest example of a metric model is a grid, or a top view map of the environment, in which each cell contains the probability of being occupied by an obstacle. These models are easy to build, since they are just evenly sampled (and integrated) versions of a joint obstacle probability density function of the environment.

III. The world model

Even though the previous world model types are quite well known [1], this article presents a new one that was meant to eliminate the disadvantages of the metric models while maintaining the same precision and speed.

When it comes to storing sampled probability density functions (*pdf*'s) there is only one method that can be simpler than the grids: instead of even sampling, we use random sampling with varying sample density. This way the world model should have less data with the same amount of information because we can have less samples in the less important regions; a solution that should make all navigation algorithms run faster.

The next two subsections will present the world model "writing" (building) and "reading" (matching) processes.

A. World model building

As mentioned above, building a metric model is quite a simple task, and that is also true for the model the Particle Swarm Localization (PSL) uses. The algorithm takes a measurement from a range sensor, and calculates the a posteriori obstacle probability density function for it. The point of having such a function instead of one discreet measurement value is that this way we can easily handle the probabilistic sensor behavior. The next step is sampling: the building process picks out random areas of the sensor space and calculates the probabilities of their being obstructed.

These area-value pairs are then stored in the world model in a way that allows for averaging the effects of each measurement taken since the robot's power was turned on.

B. Pose matching

If we choose the right averaging algorithm the robot will have a stable world model after a few measurements, at which point it can start moving and use its previous information to find out its position and heading. To make the problem easier to solve, the pose matching process gets some pose suggestions and assigns a number to each one depending on its distance from the robots real position and heading. The procedure takes one sensor at a time, calculates a posteriori obstacle *pdf* from the current measurement, and matches this function curve to the world model data. The theory behind this matching is that if the pose suggestion is correct, the relevant world model samples should represent a sampled version of the current a posteriori *pdf*.

The sum of absolute differences of the corresponding samples is then treated as the pose error. Although there is no mathematical evidence yet the tests have shown that most of the times this measure can indeed be used as a fitness value.

IV. Particle Swarm Localization

Particle Swarm Localization, or PSL for short, is based on a modified version of Particle Swarm Optimization (PSO), a simple yet effective algorithm for finding the global minimum of even highly nonlinear functions. In the localization problem it is used as a data fusion component that helps us handle those situations when the pose error is not a correct fitness measure. For example if the current pose isn't the global minimum of the pose match function, the PSL will be used to keep the error as low as possible.

The PSL's other function is to speed up the calculations by reducing the number of tried pose suggestions.

Most localization methods are based on the robot's physical model to increase effectiveness. In this solution, the PSO was modified to include such information in the pose selection process which so far seems to be a simple yet effective method; furthermore its flexibility allows us to incorporate other types of information into the process, like environment dependent robot parameters.

V. Conclusion

The preliminary tests show that the new localization algorithm (the PSL) is capable of tracking small pose changes without relying on the robot's physical motion model if the immediate surroundings contain obstacles of varying shape and size. However, on a higher level, the PSL can be effectively used to correct the odometry errors.

References

[1] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, Eds., *Principles of Robot Motion: Theory, Algorithms, and Implementation*, MIT Press, Cambridge, Forthcoming, 2005.

DESIGN OF COLLECTIVE BEHAVIOR IN MULTI-AGENT SYSTEMS

Dániel László KOVÁCS
Advisor: Tadeusz DOBROWIECKI

I. Introduction

The problem of designing a given social behavior (e.g. cooperation, compromise, negotiation, altruism) in Multi-Agent Systems (MAS) is a well known issue, still there is no general approach to solve it. In fact, there is no general and comprehensive theory connecting the individual behavior of agents with the collective behavior of the MAS. Nonetheless there are theories, which capture some profound aspects of the problem (e.g. game theory [1], theory of implementation of social choice rules [2], satisficing games [3]). Inspired by these theories a high-level model of agent decision mechanism, called virtual games, was developed [4]. The new concept overcomes most of the weaknesses of its predecessors, and provides a tractable solution to the problem.

II. The new approach

MAS are usually considered from the perspective of intelligent agents. An *agent* "can be anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors." [5]. Now, the high-level agent-model proposed [4] is the following (see. Fig. 1).

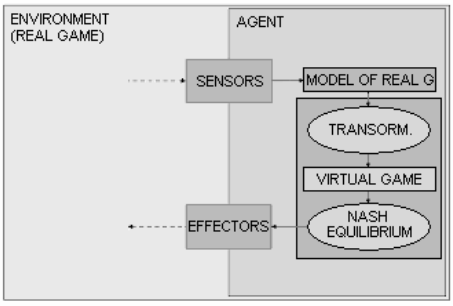


Figure 1: The new approach to the implementation problem

In multi-agent environments agents must consider the activity of other agents for effective operation. Such situations of strategic interaction are commonly modeled by game theory. Thus, the MAS environment is seen, as if it was a Bayesian game [6], where agents are *players*; the possible architectures for running agent-programs are the possible *types* of the players; agents' beliefs correspond to probability distributions over the types of the players; agents' programs correspond to *strategy profiles* of the players; and agents' plans are *strategies* of the players. Strategy profiles associate strategies to the types of the players.

Now, the lifecycle of an agent is the following: (1) first it senses the environment, i.e. the *real game*. (2) From that percept it creates a representation of the environment, i.e. the *model of the real game*. (3) This Bayesian game is the input of its *decision mechanism* choosing among possible plans, i.e. strategies. Finally, (4) the agent acts according to the strategy recommended by its decision mechanism, and then continues at step (1).

The decision mechanism of the agent has three parts: a *transformation*, a *virtual game*, and a *function for selecting a Nash-equilibrium* [6]. The transformation is responsible for generating the virtual game from the model of the real game. It may use any aspect of the model of the real game

(e.g. strategies, types, utilities). Thus the virtual game has strategies, called *virtual strategies*, and utilities, called *virtual utilities*, which may be different from those found in the model of the real game, because this way the incentives, private valuation and preferences over the possible outcomes of every single agent can be represented, and agents' individual rationality is connected with the rationality of the collective. While the concept of real utility is inherently selfish, virtual utility may reflect not only an agent's own interest, but the interest of others as well. Thus, a virtual game is "virtual" in a sense, that it isn't intended to describe or model the real game. It is only intended to "guide" the decision making of the agent in a sense, that the third component of the decision mechanism (which is responsible for selecting the strategy played by the agent), the function for selecting a Bayesian Nash-equilibrium is based upon it. Nash equilibrium is an inherently non-cooperative concept for maximizing expected profit, but since cooperative aspects are incorporated into the virtual utilities, it is appropriate. Thus the decision mechanism is eventually controlled by the virtual game, mainly by the virtual utilities. Lately it was shown, that – even with binary virtual utilities – any collective behavior in a MAS can be implemented exactly.

III. Comparison with some common approaches

Theory of games [1] provides an elaborate description framework, but does not specify how the agents' decision mechanism works. This makes game theory inappropriate for the design of collective behavior in MAS, where agents should act according to a specified rule of behavior. A new branch in game theory, *theory of implementation of social choice rules* [2] tries to implement a given collective behavior by constructing a *mechanism* centered above the agents, which produces the necessary outcomes by interacting with the collective. It considers agents to be given, and therefore specifies the decision mechanism not inside, but outside of them. This causes some fundamental difficulties (e.g. generally only approximate implementation is possible), which may be overcome, if the mechanism is distributed among the agents, like in the new approach. *Theory of satisficing games* [3] is one of the latest approaches to address the problem. It has essentially the same potential as the new approach [4], but it is more complex and the preferences of agents are represented with orderings, not utilities. The lack to represent cardinal relationships (e.g. degree of superiority) between the goodness of different outcomes is also a weakness.

IV. Conclusion

The article presented a novel method to describe, design, and analyze collective behavior in MAS. The goal was to develop a general method that overcomes the weaknesses of the previous approaches. It was shown, that arbitrary collective behavior can be implemented exactly and in general, which is a significant step in the theory of implementation. Nevertheless, the design principles enabling the construction of MAS operating according to a given social behavior are still under development. Thus, future research will mainly concentrate on synthesis: connecting the concept with existing low-level agent architectures and making practical MAS design possible.

References

[1] J. von Neumann, and O. Morgenstern, *Theory of games and economic behavior*, Princeton University Press, 1947.

[2] R. Serrano, "The Theory of Implementation of Social Choice Rules," *SIAM Review*, 46:377–414, 2004.

[3] W. C. Stirling, and M. A. Goodrich, "Satisficing games," *Inf. Sci,.* vol. 114, pp. 255-280, 1999.

[4] D. L. Kovács, "Virtual Games: A New Approach to Implementation of Social Choice Rules," *Lecture Notes in Computer Science*, Vol. 3690, Springer Verlag, pp. 266–275, 2005.

[5] S. Russell, and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 1995.

[6] J. C. Harsányi, „Games with incomplete information played by Bayesian players I-II-III," *Management Science*, Vol. 14, pp. 159–182, 320–334, 486–502, 1968.

GUIDANCE IN AD HOC POSITIONED NETWORKS

Dániel DARABOS
Advisor: Gábor HORVÁTH

I. About ad hoc positioned networks

With large wireless networks (such as sensor networks) it is often the case that it would be very useful to know the positions of the nodes but the nodes do not have a direct means to acquire this information (such as GPS). A simple example would be a sensor network of thermometers. In this case the measurements (temperatures) are not really useful without positional information. With associated positional information the temperature data could be displayed and analysed in more detail.

Ad hoc positioning methods are methods for gathering positional information in the above case. These methods rely on the topographic information present in the communication. Given that radio communication is the typical means of data exchange in wireless networks a simple example for deducing topographic information via communication could be "if two nodes are able to communicate their physical distance is not greater than the radio range".

A number of ad hoc positioning methods have been published recently (two of them are described in [1] and [2]). They have different preconditions and thus address a wide range of different scenarios. Some are assuming that the distance between two nodes can be determined with some accuracy based on the strength of the received radio signal while others base the positioning algorithm only on the connectivity graph. They are also different in their priorities with some valuing energy efficiency over positional accuracy and others the opposite. Characteristic of each method is also what level of connectivity they optimally operate at. Usually an average connectivity 5-10 is desirable (meaning that nodes have 5-10 other nodes within communication range in average). Most methods focus on isotropic networks (networks with a roughly uniform structure) and non-moving nodes, but there are some exceptions.

The accuracy of these methods is usually expressed as the ratio of the average positional error and the communication radius. Typical values are in the 10%-100% range (influenced largely by the network structure and the preconditions).

II. The guidance problem

Determining the position of the nodes in a large wireless network can be useful in a number of ways. Positional information is essential for example for representation of measurement data and can be used to optimize network routing (thus conserving power). It can also be used for navigation. The fundamental assumption in this use case is that the connectivity graph is to some extent coincident with the traversability graph.

The guidance problem can be expressed as the following. A mobile agent is stationed within the communication range of at least one network node. A node within the network is distinguished as the goal node either by itself or by the mobile agent. The problem is for the network to guide the mobile agent to the goal node with robust, distributed algorithms.

This is a unique navigation problem in that there are several "beacons" which are both statically placed and uniquely identifiable (which is very useful for localization) while at the same time their position can only be deduced via communication and only with limited accuracy. Trivial solutions for this problem can easily be found, but there is much room for improvement. The accuracy of the localization for both the mobile agent and the network nodes can be improved with delicate distributed algorithms.

III. The proposed solution

My solution to the guidance problem is based on the ad hoc positioning algorithm developed last year. In a nutshell this is an iterative algorithm which reconstructs the topology of the network based on the connectivity graph by simulating it with a spring and mass system. Depending on the properties of the network up to 10% positional accuracy can be achieved.

For efficient localization of the mobile agent a hybrid method was conceived. One part of the solution is based on the ad hoc positioning algorithm which has a significant constant error, but no accumulative error. It is complemented by the other part of the solution which is based on odometry. Odometry is the estimation of the relative position from the starting point based on measuring for example the rotation of the agents wheels. This method of position estimation is accurate for short distances but can not be used in itself because it is highly susceptible to accumulative error.

To understand how the two localization methods are joined a few words have to be said about the spring and mass system used for the nodes. In this system nodes are represented by point masses and the state of connectivity between two nodes is represented by a spring between them. Between connected nodes the spring is used to enforce a maximal distance constraint while for disconnected nodes the spring enforces a minimal distance constraint. For most nodes this leaves an area in which all springs are relaxed (all constraints are satisfied) and the node is free to move.

The general idea of the hybrid method is representing the mobile agent as a node and moving this node according to the odometry information. Assuming zero error in both node positioning and odometry the node will always reach the boundary of a free movement region at the time that its connectivity status changes and thus the free movement region changes so that it once again contains the node. However if the odometry realistically accumulates error, the connectivity status changes sooner or later than the border of the region is reached. In this case the springs will exert force on the node and the error in odometry is continually corrected. If the positioning error for the nodes is also taken into account the correcting forces have some error themselves and can even introduce error in the case of perfect odometry. This means that the error from the two methods has to be estimated (on a case by case basis) and an optimal combination found.

A moving node also provides information about the topology of the network and by passing through it increases the accuracy with which the network can be localized.

IV. Future work

So far the algorithm performs well in a rudimentary simulated environment – one with only a limited set of measurement errors simulated. However the real test of the new method will be determining its robustness against the errors in a better simulator and eventually in a physical experiment. Work on the realistic simulator is already underway and plans for the physical experiment are also getting shape. Work on the simulation of a practical application (aiding the movement of a crane) has also begun.

The continuous communication employed in the localization method of the network is energetically inefficient. There are simple ways however to make it practically applicable. In practice once the nodes are positioned to an acceptable degree of accuracy the communication can be kept to guidance use. Specifics for a more energy conservative variant will be laid out as the work on the physical implementation progresses.

V. References

[1] D. Darabos and B. Tódor, “Constraint-based Localization of Nodes in Wireless Networks,” *Proc. of the IDAACS*, Sofia, Bulgaria, 2005.

[2] C. Savarese, J. Rabay, and K. Langendoen, “Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks,” in *USENIX Technical Annual Conference*, Monterey, CA, USA, June 2002.

IMPLEMENTATION OF REDUNDANCY PATTERNS USING AOP TECHNIQUES

Péter DOMOKOS
Advisor: István MAJZIK

I. Introduction

Software fault tolerance techniques are designed to allow a system to tolerate *software faults*. Software fault tolerance techniques are based on *design diversity*, that is, critical functions of the system (or even the entire system) are implemented by several *variants* designed and implemented by independent developers possibly using different design techniques and tools. An *adjudicator* is used to accept or to reject a result. An adjudicator can e.g. compare the results of different variants (*voter*), or implement an *acceptance test*. An *executive* orchestrates the variants and the adjudicator to realize the behavior of the fault tolerance (FT) pattern. FT patterns are e.g. *NVP* (N-Version Programming) and *RB* (Recovecry Block).

Aspect-oriented programming (AOP, [1]) is an emerging programming paradigm that tries to overcome the weakness of object-oriented programming (OOP) by modularizing features that crosscut the boundaries of objects into *aspects*. *Join points* are points of the original program code (*core concern*), where additional code (an *advice*) is executed. Advices are the implementation of the *crosscutting concerns* modularized in the aspects. An advice may be a *before* advice (executed before the join point), an *after* advice (executed after the join point), or an *around* advice. An around advice is executed instead of the code at the join point. In this case, the *proceed* pseudo-method can be used to execute the original code at the join point within the around advice. Join points are designated by special language constructs (*pointcuts*). AspectJ is an implementation of the AOP concepts for the Java language [2]. AspectJ allows to write Java programs that use AOP constructs.

In this paper, we discuss the injection of FT patterns into legacy software using AspectJ. First, a small subset of the concepts is introduced, then some aspects and the experiences of a pilot implementation are presented.

II. Implementation of FT Patterns Using AOP

In this section, the main problems and proposed solutions are introduced that arise during the implementation of FT patterns in legacy software using AOP.

The implementation of the critical functionality is supposed to be available via a single method, called *FT method* in the following. (Otherwise, the code can become overloaded by pointcuts and difficult to overview.) An *around advice* is created that is executed instead of the FT method. This advice implements the executive (the logic of the FT pattern), that is, it orchestrates the variants and the adjudicator. If there is an acceptable result, it returns that result to the caller, otherwise, the failure is indicated. If the original code implements error handling, the failure can be indicated the same way as the FT method does it (e.g. returning null or throwing an exception of the same type).

Also the FT method must be modified, if it influences the program state or flow *outside* the FT method, e.g. by sending messages or terminating the program. This behavior is normal in the case of legacy software (e.g., in case of an error the program is terminated). However, in the case of the FT system, this is not acceptable. Therefore, error handlers must be analyzed, and possibly suppressed using an around advice that does not call the *proceed* method.

The variants must not take actions that affect the state of the program or its environment outside the variants (or, it must be ensured that the state can be restored before the execution of the next variant).

For this purpose, outgoing messages must be buffered by the executive and only sent if the adjudicator accepts the results (and in this case, the message must be sent exactly once, and not by all variants).

If the variants use volatile data (e.g. random numbers, incoming messages), it must be ensured, that all variants receive the same data. For this purpose, these data must be buffered and forwarded to the variants by the executive.

If the variants use global data, either checkpointing must be used in order to be able to restore the state after the execution of the variant, or the global environment must be emulated to the variant. This is necessary because all variants must be executed in the same initial environment, and in case of successful execution, the global environment must be modified by exactly one variant.

III. Pilot Implementation

To examine to usability of the concepts, a pilot implementation was made. Sun Microsystems' Open Service Gateway Initiative (OSGi, [3]) implementation, called JES (Java Embedded Server) provides a framework for multiple applications (called bundles). One of the applications is the HTTP service. This service was made fault tolerant by the application of the RB pattern. The original implementation is referred to as *jesHTTP*, while the FT implementation is referred to as *ftHTTP*.

The bundle must take some actions when it is started and when it is shut down. These actions are re-implemented in the ftHTTP, and supressed in the jesHTTP. The HTTP bundle provides an interface to other bundles to register and unregister servlets and static resources that will be made available through HTTP. The implementation of this interface is provided by the ftHTTP, which forwards the requests to all variants so that all variants can keep track of the currently registered servlets and resources.

The entire HTTP service was made fault tolerant, not only parts of it. Therefore, the prerequisite that the legacy variant is available through a single method, was violated. As a consequence, several parts (related to global data, starting up and shutting down the application) of the jesHTTP had to be re-implemented in the ftHTTP.

IV. Conclusion

According to the experiences, AOP can be used for the implementation of fault tolerance techniques in legacy software with certain restrictions. Such a restriction is that source code must be available (however, this requirement does not stem from the use of AOP).

The legacy code consist of 6.9KLoc (33 classes). The code was modified directly at 19 points in order to be able to access some classes and members, that is, only the visibility of some classes and fields had to be modified. No other direct modifications were performed on the original source code.

All other necessary modifications were carried out using aspects. 5 structural modifications were made by the introduction of fields in order to make navigation possible, and 15 behavioral modifications were made to prevent or modify the execution of an action.

The reusability of the resulting code (aspects) is restricted to the core logic of the FT pattern and the application requires extensive analysis of the original code. However, this originates mainly in the fact that the components of the redundancy pattern must be customized to the application, and is not the result of the use of AOP. The extensive analysis can not be avoided if the original code is modified directly.

References

[1] G. Kiczales et. al., "Aspect-Oriented Programming" In *Proc. European Conference on Object-Oriented Programming (ECOOP)*, LNCS 1241, Springer Verlag, 1997.

[2] AspectJ, http://www.aspectj.org

[3] Open Service Gateway Initiative, http://www.osgi.org

SPEED MEASUREMENTS OF DATABASE REPLICATION PREPARING FOR APPLICATION DEPENDENT PARTIAL REPLICATION OF DATABASES

Ágnes ERDŐSNÉ NÉMETH
Advisor: András PATARICZA

I. Introduction

With the increasingly widespread use of web-based services, databases constitute more and more critical parts of applications. System crashes involving the database lead to large recovery times, and accordingly to a low availability of the applications relying on it. The redundancy needed to assure the fault tolerance of large-scale databases, whether a complete logic replication of the database content or a RAID-like partial one (RAIDb) is used. The latter is in the scope of this article.

However, critical services use frequently only a minor fraction of the data in the database heavily and the larger part serves other purposes, e.g. logging. As the user perceived availability depends both on the frequency of use a particular service and its criticality and availability a cheap solution was searched to assure a high availability for the most critical services at the price of a moderate redundancy.

The aim of our work is to propose an application-dependent partial replication scheme assuring fast recovery of the most important data and consequently a high availability of the most critical services; while the recovery of less critical services is postponed until the main database becomes available again after a longer time.

For the composite measurements it is important to choose the type of the database server, a partial replication tool and a pilot application with critical web-based services.

II. Results of the analysis

RAIDb solutions offer a database independent solution in the terms of functionality, as this middleware communicates with the core database only via standard JDBC calls. However, the selection of a particular core database type may influence the performance of the RAIDb setup by reacting differently to the redundancy in the requests. Accordingly, at first, a small JAVA application was used to analyze different database types and replication methods:

- Implementations of the same database structure over two MySQL implementations offering different level of services (MyISAM offering only simplest database functions, and InnoDB supporting advanced features like atomic transactions as elementary operations) served as one group of baseline performance measurements. Additionally, IBM DB2 represented high-performance commercial database engines. 1000 and 10000 “insert/update” statements served as basic workload in this performance comparison experiment estimating the maximal performance by not involving any middleware or replication. The simple MySQL (MyISAM) and the commercial DB2 implementations had nearly the same response time, while the InnoDB solution was 20-35 times slower than the first two implementations.
- The next experiment aimed at the estimation of the performance overhead originating in a full replication scheme corresponding to a high level of fault tolerance executing an update of the replica immediately after a write operation to the database. This setup is quite similar to mirroring, but the performance test executes only write-type operations in order to avoid interferences from read speedup mechanisms. Measurements were performed both by using

the built-in full database replica mechanisms of the different database engines, and externally driven by Sequoia.[1][2]. Here once again MySQL with MyISAM type tables and IBM DB2 Q-Replication offer a similar performance both with the built-in and Sequoia-based replication, but the InnoDB is slower by a factor of 10-15 times. (See Figure 1.)

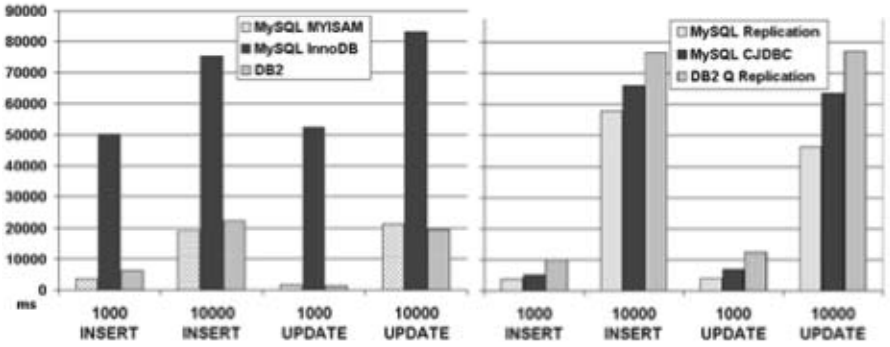


Figure 1: The measurements data

III. Conclusion

Partial replication is only available for IBM DB2 as a built-in feature. Q-Replication can execute a partial replication as fine granular, as the smallest unit of replication is a single attribute primarily used for database federation. The missing support in MySQL necessitates the use of Sequoia. Once again, the performance measures of MySQL for MyISAM and DB2 are quite similar.

The performance related experiences can be summarized in such a way, that simple databases offering a very basic set of services and a commercial one with sophisticated services may reach a similar performance for those basic operations, which are needed for replication.

IV. Future work

For the composite measurements, Sequoia with MySQL MyISAM tables and IBM DB2 Q-Replication were selected as database platform for partial replication (RAIDb-2) implemented with Sequoia and the TPC-W benchmark as pilot application (see figure 2.) [4] [5].

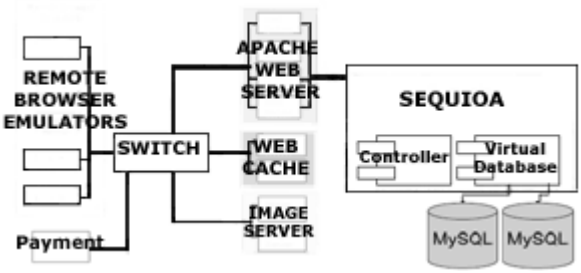


Figure 2: A The architecture of the TPC-W benchmark implementation with Sequoia

References

[1] E. Cecchet, *RAIDb: Redundant Array of Inexpensive Databases*, LNCS 3358, 2004, pp. 115 – 125, ISPA 2004

[2] E. Cecchet, J. Marguerite, W. Zwaenepoel, *Partial Replication: Achieving Scalability in Redundant Arrays of Inexpensive Databases*, LNCS 3144, 2004, pp 58 – 70, OPODIS 2003

[3] G. Pintér, H. Madeira, M. Vieira, I. Majzik, A. Pataricza, *A Data Mining Approach to Identify Key Factors in Dependability Experiments*, LNCS 3463, 2005, pp 263 – 280, EDCC 2005

[4] Specification of TPC-W: Transaction Processing Performance Council - http://www.tpc.org

[5] Implementation of TPC-W: http://www.ece.wisc.edu/~pharm/ & http://www.cs.cmu.edu/~manjhi/tpcw.html

ROBUSTNESS TESTING OF HIGH AVAILABILITY MIDDLEWARE SOLUTIONS

Zoltán MICSKEI
Advisor: István MAJZIK

I. Introduction

Lately dependability became a key factor even in common off-the shelf computing platforms. High availability (HA) can be achieved by introducing *manageable redundancy* in the system. The common techniques to achieve minimal system outage can be implemented independently from the application, and can be exposed as a *HA middleware*. Recently the standardization of the functionality of such middleware systems has begun (for example the AIS [1] specification, and its open source implementation, OpenAIS). The benefit of an open specification would be for example the easier integration of different off-the shelf components.

With multiple products developed from the same specification the demand to compare the various implementations naturally arises. The most frequently examined properties are performance and functionality, but especially in case of a HA solution the *dependability* is also, or even more important. This paper outlines an approach to *compare the robustness* of HA middleware systems.

II. Related work

Robustness is defined as the degree to which a system operates correctly in the presence of *exceptional inputs* or *stressful environmental conditions*. In the past ten years several research projects examined the robustness of different kinds of applications.

Earliest works used software implemented fault injector tools to simulate hardware faults. Console applications were tested using randomly generated streams searching for input combinations that can crash the system under test. In *Ballista* [2] the POSIX API was examined, and the robustness of fifteen POSIX operating systems was compared. The method used in Ballista was to develop for each type in the API a test generator that can produce valid and exceptional values. The API functions were called with the combinations of the values returned by the generators. The goal of the European Commission project *DBench* [3] was to define a general framework for dependability benchmarking. The procedure of the benchmarking is to characterize a workload representing normal operation and a faultload with injected faults.

III. Robustness testing of HA middleware systems

One of the earliest phases of developing a test strategy is to identify potential places where faults can occur in the system, i.e. develop the fault model of the system. Figure 1 illustrates a typical node in a HA distributed system and the identified specific fault types.

1. *External errors*: These errors are application-specific, thus they are not included in our tests.
2. *Operator errors*: In general, operator errors mainly appear as erroneous configuration of the middleware and erroneous calls using the specific management interface.
3. *API calls*: The calls of the components using the public interfaces of the middleware can lead to failures if they use exceptional values, e.g. NULL or improperly initialized structures.
4. *OS calls*: the robustness of a system is also characterized by how it handles the exceptions returned by the services it uses, in this case the operating system errors.
5. *Hardware failures*: The most significant HW failures in a HA middleware-based systems are

host and communication failures (that has to be tolerated in the normal operating mode) and lack of system resources.

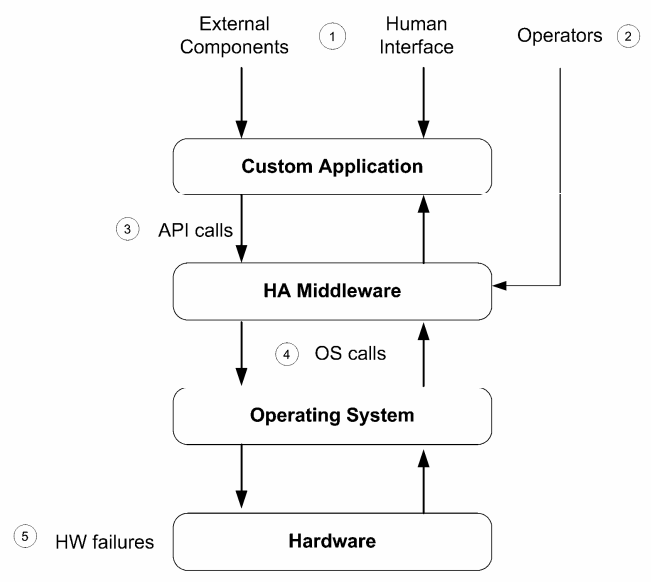


Figure 1: HA middleware fault model

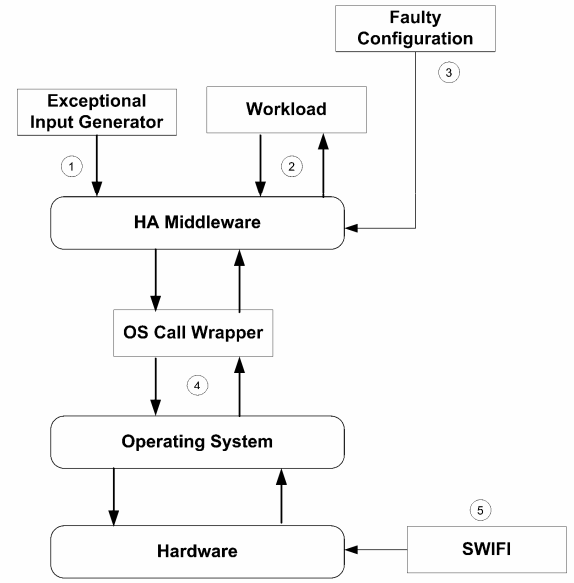


Figure 2: Architecture of the test environment

According to the fault model described above we developed the method illustrated in Figure 2 to test the robustness of a HA middleware. The method can be implemented in two separate phases.

- Phase I: testing exceptional inputs in API calls with the following techniques (1):
  - Generic input generators using fixed domain of input for all types.
  - Type-specific exceptional input generators using the knowledge of the specific types.
  - Scenario-based testing using the sequence diagrams in the specification to reach such system states in which exceptional inputs can be provided.
- Phase II: testing stressful environment conditions. The steps needed for this are the following.
  - Defining a workload according to the normal operation (2).
  - Constructing a faultload representing various faults from the environment (3, 4, 5).

To demonstrate the feasibility and efficiency of this method we developed tests for the OpenAIS middleware. Code examples were created with generic and type-specific input generators, and robustness test cases were generated from the available functional test cases and sequence diagram specifications using *mutation techniques*. The tests found several robustness failures, mostly related to improper pointer handling (e.g. null pointers). The generation of test cases with generic testing was partially automated using templates.

IV. Conclusion

In this paper we examined methods used for robustness testing of different implementations of the same specification of a HA middleware. We presented the fault model and proposed a process to test the robustness of the middleware at different layers. The implementation of the test programs has been started based on OpenAIS with promising preliminary results.

References

[1] Service Availability Forum, URL: http://www.saforum.org/

[2] P. Koopman et al, "Automated Robustness Testing of Off-the-Shelf Software Components," in *Proc. of Fault Tolerant Computing Symposium*, pp. 230-239, Munich, Germany, June 23-25, 1998.

[3] K. Kanoun et al, "Benchmarking Operating System Dependability: Windows 2000 as a Case Study," in *Proc. of 10th Pacific Rim Int. Symposium on Dependable Computing*, Papeete, French Polynesia, 2004.

INDEXING TECHNIQUES FOR TEXT CATEGORIZATION

István PILÁSZY
Advisor: Tadeusz DOBROWIECKI

I. Introduction

Nowadays through the sudden growth of the Internet and on-line available documents, the task of organizing textual data becomes one of the principal problems. A major approach is text categorization (TC), the task of which is to automatically assign documents into their respective categories. TC is used to classify news stories, to filter out spam and to find interesting information on the web. Until the late '80s the most popular methods based on knowledge engineering, i.e. domain experts created rules by hand. In these days the best TC systems use supervised machine learning approach: the classifier learns rules from examples, and evaluates them on a set of test documents.

The task of supervised machine learning (ML) is to learn and generalize an input-output mapping. In case of text categorization the input are the documents, the output are their respective categories (labels). To evaluate a classifier, one needs a set of training examples and another set of testing examples. An ML algorithm creates a classifier from the training set, which is applied then to the testing set, and gives the predicted labels. Performance measures are based on the comparison of the predicted and the real labels. However, in practice we do not know the labels of the testing documents that is why we apply ML.

The most popular machine learning framework for TC are Support Vector Machines (SVMs), the aim of which are to find a linear decision surface that separates positive and negative examples: it tries to minimize a fixed linear combination of the reciprocal of margin and the training error. The margin is the smallest distance of the examples and the decision surface. Training error is generated by examples which are not beyond the corresponding margin. There is a non-linear extension of SVMs that exploit one interesting porperty of SVM: it does not uses examples as they are, but in the inner-product with other examples. The non-linear extension replaces the inner product with a more general kernel-funtion. However, in case of TC, the non-linear approach is only a wee bit better compared to the linear approach [1].

II. Indexing techniques

The most commonly used indexing technique for TC is the so-called tf-idf term-weighting scheme [2]. Roughly speaking, terms are words, but a term is a more general concept, it may mean phrases, etc. In tf-idf term-weighting scheme documents are represented as vectors, each dimension corresponds to a term. Document vectors (columns) form together the term-document matrix:

$$TD(t,d) = TF(t,d) \cdot IDF(t), \quad IDF(t) = \log \frac{N}{DF(t)}, \quad DF(t) = \sum_{d:\, TF(t,d)\neq 0} 1, \quad N = \sum_{d} 1 \qquad (1)$$

where $TF(t,d)$ is the number of occurences of the tth term in the dth document, DF means document-frequency, IDF means inverse document-frequency.

III. Considerations

There are lots of methods for TC. Which one is the best, it depends on the nature of the task. These tasks are very diverse. Methods can be sophisticated or theoretically established, but the only way to say that one is better than another is evaluating each one in a concrete task.

TC systems may have a lot of free parameters, which influence the performance. Our task is not to say that the ultimate choice for parameter X is Y, because it is different for each task, but to identify important parameters, and give some idea how to set them.

Other days other ways – other days other language. When evaluating a TC system, train and test examples must be separated. It is always the case that people randomly select examples for training and testing. We suggest to split with a document-creation-time threshold, because it is more practical: in practice often it is the case that newly created documents' category must be determined based on formerly created ones.

In the following we propose two indexing methods for a concrete task. First, we would like to see, what if we change the importance of words based on their position in the document. Second, we want to reduce the high dimensionality caused by the lots of terms.

IV. Two proposed indexing methods and preliminary results

To evaluate text classifiers labeled documents are required: We used the `music.hu` portal's articles, classified by their genres: 1482 training examples, 585 testing examples, and 8 categories. The used performance measures are micro F_1 (MiF_1) and macro F_1 (MaF_1) [3].

Splitting train and test examples randomly leads to MiF_1=0.7511, MaF_1=0.6136. Splitting by creation time leads to MiF_1=0.6733, MaF_1=0.4977. Furtheron, this later split will be used.

First, we examined, what if we index the title and the body of a document separately. This was achieved by prefixing a "b_" or "t_" to words, indicating whether they are in the title or in the body. To achieve the best results, TF was multiplied by 5 in case of title-words, to give higher importance to these words, and in DF-based term selection (i.e. keeping only a fraction of words, based on their DF values) 4 was used instead of 1 in the summation in eq. 1, to keep a considerable amount of title-words. With these modifications MiF_1=0.7096, MaF_1=0.5895 was achieved.

Second, we examined a more complex idea: Train and test documents have different words, especially when splitting by creation time. This leads to performance degradation. To overcome this problem, we suggest the following: create small bundles of test examples, each containing 20 documents. For one bundle, train the classifier by keeping only those words that exist in the bundle. Throw out train examples with very few (≤ 6) words. With this modification MiF_1=0.7139, MaF_1=0.5974 was achieved.

V. Conclusion, further work

In this paper we proposed two indexing methods for TC. Approx. 4% improvement can be achieved in MiF_1, and 10% in MaF_1, which is quite notable and encourages further investigations, e.g.:

- examine how other parts of texts should be given less or more importance;
- examine how to combine these diverse methods into a sophisticated system or in a mixture of experts framework;
- create better bundles in the 2nd method, e.g. by clustering test examples by the label assigned with a usual classifier.

References

[1] T. Joachims, "Text categorization with support vector machines: learning with many relevant features," in *Proceedings of ECML-98, 10th European Conference on Machine Learning*, C. Nédellec and C. Rouveirol, Eds., number 1398, pp. 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[2] A. Aizawa, "An information-theoretic perspective of tf-idf measures," *Information Processing and Management: an International Journal archive*, 39(1):45–65, 2003.

[3] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, 34(1):1–47, 2002.

NONLINEAR INPUT DATA TRANSFORMATION

László LASZTOVICZA
Advisor: Béla PATAKI

I. Introduction

This paper presents a method for improving the performance of classification algorithms.

A nonlinear transformation is applied to the input data before the classification is performed. A variant of this method is also presented which uses the same idea as behind kernel methods [1] as it extends the dimensionality of the input e.g. increases the dimensionality of the feature space. I show experimentally that the proposed method is capable of improving the performance of different classification algorithms. The experiments were conducted on well-known machine learning benchmark datasets [2].

The method is rather simple since it is only a nonlinear mapping of the input data through a nonlinear function. The extension is a little bit more complicated though as it consists one or two transformation steps of the input vectors and the concatenation of two vectors.

II. Mathematical considerations

Let us consider a usual formulation of a K class classification problem in the case of a given empirical data of N samples,

$$(\mathbf{x}_1, d_1),.....,(\mathbf{x}_N, d_N). \tag{1}$$

where $\mathbf{x}_i \in \mathbf{R}^n$ and $d_i \in (1,2,...,K)$ are class labels of K classes. The probability of being correct in the case of a general classifier is given by [4],

$$P(correct) = \sum_{i=1}^{K} \int_{R_i} p(\mathbf{x} \mid d_i) P(d_i) d\mathbf{x}, \tag{2}$$

where $P(d_i)$ are the a priori probability of the classes and $p(\mathbf{x} \mid d_i)$ are the class conditional probability distributions and R_i represents the decision region formed by the classifier. Let $f : \mathbf{R}^n \to \mathbf{R}^n$ be a nonlinear function, so that $f(\mathbf{x}) = f([x_1,...,x_n])$, and let us transform the data

$$(\mathbf{x}_1, d_1),.....,(\mathbf{x}_N, d_N) \to (f(\mathbf{x}_1), d_1),.....,(f(\mathbf{x}_N), d_N). \tag{3}$$

This mapping gives a new dataset to be classified by the classifier. Considering this transformation as a probability distribution transform, it changes the probability distribution of the data. For monotonic, bijective function it follows as

$$p(y) = p(x)\left|\frac{dx}{dy}\right| = p(f^{-1}(y))\left|\frac{df^{-1}(y)}{dy}\right|, \tag{4}$$

which can be generalized. In the same time the probability of being correct given by Eq. 2 also changes. An extension of this method can be reached if one uses for example the following transformation,

$$\Phi : \mathbf{R}^n \to \mathbf{R}^{kn}, \tag{5}$$

which, if $k = 2$, gives the following transformation,

$$(x_1,....,x_n) \mapsto (f_1(x_1),...,f_1(x_n), f_2(x_1),...,f_2(x_n)), \tag{6}$$

that is, Φ is the concatenation of two differently transformed input vectors. The extension of the dimensionality of the input space can be arbitrary. As transformation functions many functions can be used, Table 1 contains a few examples.

Sigmoidal	$\tanh(s \cdot \mathbf{x} + c) = \tanh(s \cdot [x_1, ..., x_n] + c)$
Logistic	$1/(1 + \exp(-s \cdot \mathbf{x} + c)) = 1/(1 + \exp(-s \cdot [x_1, ..., x_n] + c))$
Gaussian	$\exp(-(\mathbf{x}^2 + s)/c) = \exp(-([x_1, ..., x_n]^2 + s)/c)$

Table 1: Suitable functions

III. Experiments and results

The experiments were conducted on well-known machine learning benchmark datasets using a 10-fold cross-validation scheme. I used neural networks, decision trees and subspace methods (CLAFIC, ALSM) [3] as classifiers. The settings of the different classification algorithms are the same, except when using the input extension when the weights in the input layer of the neural networks are doubled. As nonlinear function I used only the hyperbolic tangent function in the basic case ('Tr'), and I used the hyperbolic tangent and Gaussian functions in the extended case ('ExtTR'). The results are shown in Table 2.

Dataset	**Neural network**			**Decision tree**			**CLAFIC**			**ALSM**		
	N	**Tr**	**ExtTr**	**N**	**Tr**	**ExtTr**	**N**	**Tr**	**ExtTr**	**N**	**Tr**	**ExtTr**
breast-cancer	3.65	3.21	2.92	6	3.43	-	15.47	8.16	4	11.32	8.35	3.63
cleveland	19.07	16.41	15.55	25.38	24.21	-	22	19.78	19.37	17.87	17.16	17.29
crx	14.67	13.36	13.19	17.19	14.29	-	17.86	15.28	15.19	15.25	14.14	14.43
glass	32.43	31.82	29.17	30.82	30.28	-	44.13	38.65	38.18	32.52	30.53	31.42
hepatitis	19.4	19	17.5	23	18.2	-	16.73	16.63	15.53	16.73	16.63	14.63
hypo	5.97	2.01	1.91	0.61	0.59	-	15.94	13.9	11.96	3.77	3.72	2.8
sonar	18.06	14.17	10.86	30.16	27.9	-	22.64	20.5	16.14	19.83	17.9	16.44

Table 2: Error rates for the different classification algorithms and transformations

Table 2 contains three columns for each algorithms, 'N' stands for the run without transforming the input data. 'Tr' stands for the simple transformation (see Eq. 3) and 'ExtTr' stands for the extended transformation (see Eq. 5). As one can see, the results with transformed input vectors are better for each classification algorithm in all cases and in many cases they are better significantly.

IV. Conclusions

Applying the proposed transformation the performance of different classification algorithms can be improved. Extending the dimensionality of the input space can also increase the performance, but it changes the classification problem significantly. The main question is how to determine the parameters of the transformation and which functions are the most suitable.

References

[1] Schölkopf, B.: *Statistical Learning and Kernel Methods. Data Fusion and Perception*, Technical report No.(23), 3-24. (Eds.) Della Riccia, Lenz and Kruse, Springer, Redmond, WA (2000)

[2] J Blake, C.L. & Merz, C.J. (1998). *UCI Repository of machine learning databases*, Irvine, CA: University of California, Department of Information and Computer Science http://www.ics.uci.edu/~mlearn/MLRepository.html].

[3] E. Oja, *Subspace methods of pattern recognition*, Research Studies Press, 1983

[4] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification*, John Wiley and Sons, Inc. 2001

DECISION TREES WITH UNCERTAINTY

Norbert Tóth
Advisor: Béla Pataki

I. Introduction and inspiration

A Medical Decision Support System for Mammography is being developed in cooperation with radiologists in the Budapest University of Technology [1] for the automatic evaluation of the mammograms. In the system several detection algorithms are working parallel to each other, looking for different kinds of abnormalities (e.g. microcalfications, masses) or different kinds of features to detect the same type of abnormality. Since markings (spots that show the location of an abnormality) created by the detection algorithms cannot be 100% certain, a confidence value was introduced to the system. Each marking is accompanied by this value, showing the diagnoses certainty. Each algorithm produces this confidence value, although in different ways.

One of the methods uses decision trees to classify a certain number of features at a location of the image [2]. The result of this classification can be normal tissue or abnormality. If the features are classified as abnormal tissue a marking is generated. To generate the confidence value the original decision tree algorithm was modified to handle classification uncertainty.

II. Dealing with classification uncertainty

One of the most widespread used decision tree frameworks is the Classification and Regression Trees (CART, 1984) [3] developed by Breiman et al. Their work was used as basis for the enhancements to the decision tree methodology. Decision trees produced by the CART algorithm have some favorable properties compared to other methods. They are easily interpretable and can be used to classify data very quickly. Another good property is that the decision boundary can be easily identified. A new algorithm was developed to provide a confidence value to the classification result of the tree. This algorithm makes use of the clear structure of the trees and the explicitly defined decision boundary.

Dealing with the classification uncertainty or classification confidence the main assumption is that the confidence of the classification is proportional to the distance from the closest decision boundary that splits the differently labeled regions.

According to the previous assumption, to get a classification certainty value we need to measure the shortest distance to the closest decision boundary that splits different classes, or equally the shortest distance to the closest region with different class label.

The proposed algorithm to measure the shortest distance from the closest relevant decision boundary is the following:

1) First the actual data is classified using the decision tree: a leaf node is reached, which defines a section in the input space and an output label.
2) To get the distances to the other sections the input data point is projected to all of the decision boundaries. The projection rules are calculated only once, right after the tree growth process and stored together with the tree structure. If the input space contains N variables than the decision boundaries of a section are maximum N-1 dimensional hyperplanes.
3) The distance between the projected points and the input point is calculated.
4) Take out the projected point that has minimal distance from the input point.
5) Check if that projected point is on a decision plane that splits different classes.

6) If yes, the output certainty value is the distance between the projected point and the input data. If no, take out the next projected point with minimal distance and repeat steps 5 and 6.

From the distances of the original training points the true classification rate can be estimated for a given distance (Figure a, upper curve). Assuming a correct classifier this function increases with the distance, ultimately reaching 1. p_D=1 at a given distance D indicates that the training points whose distance to the decision boundary is greater than D are all correctly classified.

However using this estimated classification rate p_D as classification uncertainty will give high confidence to points that are far away from the decision boundary. But how can we estimate the classification certainty if there were no training data farther away than the actual input point? To overcome this problem the 95% lower confidence interval φ (Figure a, lower dotted curve) is calculated for the p_D function. φ is a function of p_D, N (number of point used to estimate p_D) and α (the significance level, which is 0.05 in the current implementation). As D increases p_D will (usually) also increase and this will cause φ to increase also. However at a certain distance the number of points used to estimate p_D will decrease in the way that it will cause φ to fall. Using φ as classification certainty will give zero confidence to points that are farther away than the training points.

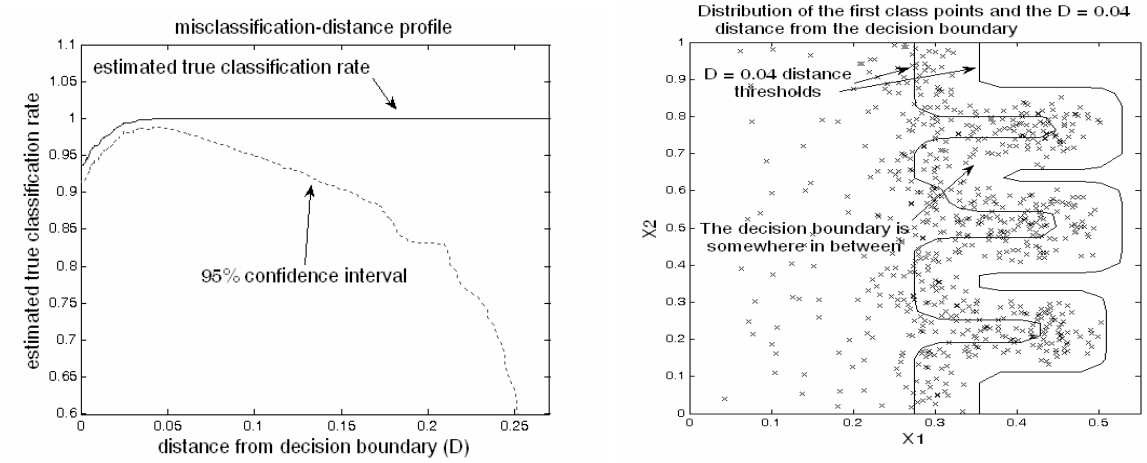


Figure a (left): estimated classification rate and the 95% confidence interval. The confidence interval falls to zero as the number of training points decreases. Figure b (right): Distribution of the training points around the decision boundary. Only class 1 points are displayed. The D=0.04 distance thresholds from the decision boundary is also shown.

III. Results and Conclusion

A novel method was presented in this paper to extend the decision tree framework. The proposed extension to the decision tree framework gives the possibility to determine classification certainty for each input sample. The method is experimental, testing will be concluded in the next months.

References

[1] G. Horváth, J. Valyon, Gy. Strausz, B. Pataki, L. Sragner, L. Lasztovicza, N. Székely, "Intelligent Advisory System for Screening Mammography" in *Proc. of the IEEE Instrumentation and Measurement Technology Conference, IMTC '2004*. Como, Italy, May 18-20. Vol.3. pp. 2071-2077.

[2] N. Székely, N. Tóth, B. Pataki, "A Hybrid System for Detecting Masses in Mammographic Images" in *Proc. of the IEEE Instrumentation and Measurement Technology Conference, IMTC '2004*, Como, Italy, 18-20 May 2004, pp. 2065-2070.

[3] L. Breiman, J. Friedman, R. Olshen, C. Stone, *Classification and Regression Trees*, Chapman & Hall, 1984.

LOCAL HYPERPLANE CLASSIFIERS

Gábor TAKÁCS
Advisor: Béla PATAKI

I. Introduction

The concept of margin (distance between the decision surface and the closest training point) proved to be a useful approach in machine learning, and led to new classifiers that have good generalization capabilities. The most famous ones are Support Vector Machines (SVMs) [1] that produce a linear decision surface with maximal margin for linearly separable problems. For the non-linearly separable case SVMs map the input into a higher (possibly infinite) dimensional feature space with some non-linear transformation and build a maximal margin hyperplane there. The trick is that this mapping is never computed directly but implicitly induced by a kernel. While non-linear SVMs can solve any separable classification problem with zero training error, the large margin in the feature space does not necesserily translate into a large margin in the input space [2].

A natural idea is to try and build a large margin decision surface directly in the input space. Would it be possible to find an algorithm that produces a non-linear decision surface which correctly separates the classes and maximizes the margin in the input space? Surprisingly the plain old Nearest Neighbor (NN) algorithm does precisely this. But NN classifiers have too much capacity (VC-dimension) therefore they are often outperformed by SVMs in practice. Figure 1 shows a geometrical explanation of this phenomenon. A possible approach to improve the generalization capability of NN classifiers is to somehow fantasize the missing training samples based on the local linear approximation of the manifold of each class [3]. This paper presents a new algorithm for defining the local hyperplanes and discusses some performance-boosting issues related to this topic.

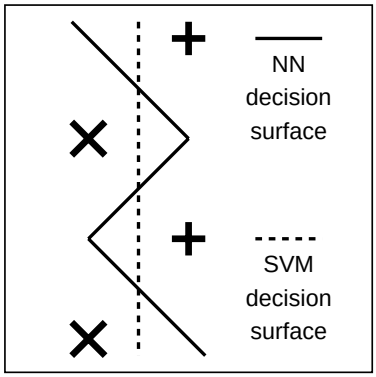


Figure 1

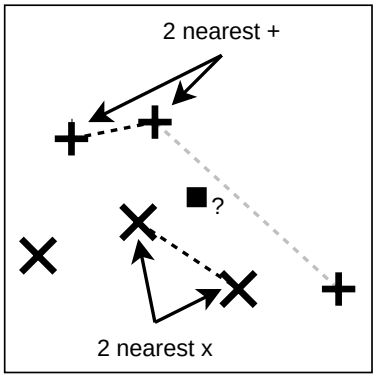


Figure 2

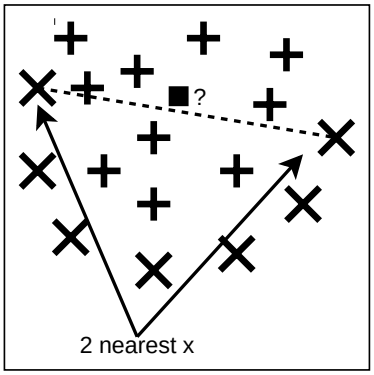


Figure 3

II. Defining the Local Hyperplanes

Given a testing point $\mathbf{y}$, a straightforward way to define the local hyperplane for class c is to choose the K training points from c whose distance to $\mathbf{y}$ is the smallest (K-c neighborhood). However this method (used in [3]) is simple it can easily lead to unwanted classification result (Figure 2). Therefore we propose a more sophisticated algorithm for defining the local hyperplanes:

- *Step 1*: Initialize the generator points of the local hyperplane for class c with the K-c neighborhood of $\mathbf{y}$. ($\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K], \mathbf{x}_i \in \mathbb{R}^N, K \leq N$)
- *Step 2*: Compute the nearest point to $\mathbf{y}$ on the local hyperplane.
  ($\mathbf{p} = \mathbf{X}\mathbf{t}$, where $\mathbf{t} = \operatorname{argmin}_{\mathbf{s}:\sum s_i=1} ||\mathbf{X}\mathbf{s} - \mathbf{y}||$)

- *Step 3*: Stop if the nearest point is located inside the convex hull of the generator points. (if $\forall i : t_i \geq 0$ then return $\mathbf{X}$)
- *Step 4*: Compute the location where the line connecting the center-of-gravity and nearest point exits the convex hull. ($\mathbf{p}_{exit} = \bar{\mathbf{x}} + (\mathbf{p} - \bar{\mathbf{x}})/(1 - Kt_{min})$, where $\bar{\mathbf{x}} = \frac{1}{K}\sum \mathbf{x}_i$ and $t_{min} = \min_i t_i$)
- *Step 5*: Replace the generator point corresponding to the most violating t_i with the next closest training point from class c. ($\mathbf{x}_{min} = \mathbf{x}_{next}$)
- *Step 6*: If the iteration count is less than or equal L, then go to *Step 2*. Otherwise return the hyperplane whose exit point was closest to $\mathbf{y}$.

In [3] the nearest point to y on the local hyperplane is computed by eliminating the constraint and solving a linear system of equations in K - 1 variables. Our idea is to try and solve the constrained optimization problem directly, so thus the convex hull containment question can be decided easily. It can be shown that the minimization can be performed with gradient method without violtating the constraint.

III. The Proposed Classifier

Local hyperplane classifiers are classifying machines that make their decisions based on the local hyperplanes. The members of this family differ in the way of defining the hyperplanes and the method of decision making. Now we proceed with some questions of the decision making part.

It is not worth classifying all inputs based on the distance from local hyperplanes (Figure 3). For testing points close to the decision boundary (*critical inputs*) the local hyperplane classification is useful, but for testing points far from the decision surface (*non-critical inputs*) the plain NN classification is more reliable. Therefore the proposed classifier splits the testing set into a critical and a non-critical part and applies the nearest local hyperplane rule only for the critical inputs. The partitioning algorithm is the following:

- *Step 1*: Label the nearest outclass neighbor of each training point as a border pattern. (The nearest outclass neighbor of a training pattern is the closest training point from a different class.)
- *Step 2*: Label testing points as critical if their nearest training point is a border pattern.

IV. Experimental Results

Our classifier was tested with some artificially generated non-linearly separable problems (e.g. double spiral). This non-exhaustive test showed that the new classifier can outperform NN classifiers in accuracy and SVMs in computational efficiency.

V. Conclusion, Future Work

A local hyperplane classifier architecture was presented in this paper. The uniquene elements of the approach are a new local hyperplane finding algorithm and a partitioning algorithm that splits the testing set into a critical and a non-critical part. Although the results are promising for some artificial datasets, the classifier should be tried for real-life problems and the domain of its applicability should be assigned.

References

[1] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

[2] S. Akaho, "Svm maximizing margin in the input space," in *Proceedings of the 9th International conference on Neural Information Processing*, vol. 2, pp. 1069–1073, 2002.

[3] P. Vincent and Y. Bengio, "K-local hyperplane and convex distance nearest neighbor algorithms," in *Advances in Neural Information Processing Systems 14*, pp. 985–992, Cambridge, MA, 2002. MIT Press.

THESAURUS REVIEW SYSTEM FOR COLLABORATION

András FÖRHÉCZ
Advisor: György STRAUSZ

I. Introduction

In the European Union citizens are welcome to study or work abroad. We should take advantage of the integrated European labour market, but to take part in such an international cooperation is somewhat difficult. One of the most important factors for both parties is to clearly understand the expected skills and competencies.

In recent years EU institutions have developed tools to establish the transparency of qualifications, like the Europass CV [1]. But all these suffer from the same weakness: the core concepts of these tools – terms for skills and competencies – are neither standardised nor internationally compatible, thus they can only offer limited support.

II. Project background

The Leonardo da Vinci Pilot Project DISCO (European Dictionary on Skills and Competencies, [2]) intends to fill this gap by providing a common terminology, a multilingual thesaurus on vocational skills and competencies. Having strong international background, the project has operational partners from Austria, the United Kingdom, Germany, France, Belgium, the Czech Republic, Lithuania and Hungary. Most of the countries already have national thesauri or at least a compilation of possible qualifications. I took part in this project by supporting the development of the methodology and creating a thesaurus review system for partner collaboration.

III. Project goals

The main result should be a translation tool for skills and competencies, which is missing from European transparency tools at the moment. The translation tool consists of a multilingual thesaurus and an online tool for individuals, which offers querying, browsing and supporting automatic translation of qualifications.

The common terminology means not only a standardized English thesaurus which describes vocational skills and competencies, but it contains the national translations as well. By translation the terms' meaning may slightly change, and although it cannot be completely avoided, it has to be kept down as much as possible. Certainly, the semantic distortions should be documented.

We expect to have approximately 2.000–3.000 terms and 1.000–5.000 synonyms per language. Besides different translations, terms should be mapped to concepts of existing thesauri related to qualifications. There are two tightly related standard collections. ISCED [3] is established by UNESCO for the compilation of national and international education statistics. It also contains the hierarchy of the fields of education. ISCO [4] organises occupations in a hierarchical framework. Although it has a looser connection with qualifications, the higher levels should be mapped to DISCO, in order to assist in searching required skills for a given occupation.

IV. Mapping and merging

As mentioned earlier, the purpose of the project is to complete a multilingual thesaurus on the basis of already available national collections in the domain of skills and competencies.

"Thesauri are designed as an agreement and compromise on a set of shared terms for common concepts." [5] The project partners already have their more or less strict conception about the assignment between terms and objects, now we have to achieve interoperability between these terminologies.

Thesaurus mapping is the process of identifying approximately equivalent terms and hierarchical relationships from one thesaurus to another. It has limited interoperability, depending on the number of found equalities. For example, if a field of interest is not mentioned in a foreign terminology, we will not be able to translate it at all.

Full integration can be achieved with thesaurus merging: construction of a common thesaurus with all concepts involved, and mapped from the source terminologies. Certainly, this task is substantially complex: besides finding similarities, we also have to resolve differences.

V. Methodology

There was a lot of effort made developing collaboration tools, especially in the field of ontology compilation. These tools make it possible for different users to modify the same knowledge base at the same time (e.g. Ontolingua, KAON Engineering Server).

The project does not gain as much from a real collaboration tool as it would seem at first. The size and complexity of the thesaurus implies that it must be composed in a centralized manner. Otherwise, it would be impossible to guarantee consistency (e.g. to avoid duplicates).

After translating national thesauri into English, a charged partner tries to merge them into a central English terminology – only one sector at a time. Certainly, this cannot be performed without errors, because the partner may not be aware of all national specialities, possibly whole fields specific to a country. The final terminology is developed through alternating merging and review phases.

Review phases are performed with a custom online tool, which is able to import and track thesaurus changes. This way the review phase is separated from the engineering tools, the partner merging the thesaurus can use advanced software of own choice. All partners can suggest modifications and discuss accurate semantics for individual terms on the review interface.

VI. Recent and future work

The project is just entering the review phase. A specification of the review system has been written and accepted based on the described methodology and former discussions between the project partners. I have developed the review system, but we have not got yet any feedback on its usability.

If all partners start to comment on the current pilot thesaurus, we will see how suitable the concept of this methodology is, and how hard it is to fulfil the partners' requests to merge content. It can happen that the growing size of the thesaurus forces us to refine the applied methodology, and also improve the review system in parallel.

In the meantime I can start the development of a public user interface. End users with no former experience in this field need a more intuitive tool to explore the content and use for translation.

References

[1] The Europass Curriculum Vitae, URL: http://europass.cedefop.eu.int/

[2] European Dictionary on Skills and Competencies, URL: http://disco.youtrain.net/

[3] International Standard Classification of Education, ISCED 1997, URL: http://www.unesco.org/education/information/nfsunesco/doc/isced_1997.htm

[4] International Standard Classification of Occupations, ISCO-88, URL: http://www.warwick.ac.uk/ier/isco/isco88.html

[5] M. Doerr, "Semantic Problems of Thesaurus Mapping," *Journal of Digital Information*, 1(8), Apr. 2001. URL: http://jodi.ecs.soton.ac.uk/Articles/v01/i08/Doerr/

INCORPORATING TEXTUAL INFORMATION INTO BAYESIAN NETWORKS

András MILLINGHOFFER
Advisors: Tadeusz DOBROWIECKI, Péter ANTAL

I. Introduction

Today, Bayesian networks (BNs) are one of the most popular tools for representing and handling uncertainty. This is primarily due to that (1) they can expressively represent human knowledge and (2) can normatively combine it with observations. However, the learning of Bayesian networks from data is often inapplicable because of the lack or the high cost of data, and the evaluation of the resulting posterior distribution also can be problematic. To come around this difficulty, the paper proposes a text-mining method through which we can construct the structure of the domain model, which can be used directly or as a starting point (prior) for further refinement.

We also propose an extension to Bayesian networks, through which we gain an annotated knowledge representation method able to handle queries containing textual information concerning the domain.

II. Bayesian networks

Bayesian networks (see [1]) model the relevant quantities of the world as probabilistic variables. Our knowledge about the domain is represented by their joint distribution. A BN consists of two parts: (1) the directed acyclic graph G represents the variables with its nodes and direct probabilistic dependencies with its edges; (2) the local *conditional probability distributions* associated to each node, describe the node's distribution conditional on its parents in the graph. The joint distribution can be computed by multiplying the local models: $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|Parents(X_i))$.

A BN can be interpreted as (1) an effective representation of the joint distribution, (2) the map of probabilistic dependencies of the domain, or (3) the causal description of the domain, with edges representing direct cause-effect connections.

III. Text mining with Bayesian networks

Since the data needed for learning are often inaccessible or very expensive and expert provided information is difficult to handle for a knowledge engineer, a third source of information, namely scientific publications would be worth considering. An ambitious goal could be to build models based on textual data that represent a background knowledge comparable to those provided by the experts, i.e. to extract knowledge encoded in articles and papers, or even to determine the direction of further research into the domain (see [2]).

A. Causality appearing in Bayesian networks

The basic ideas about the connection between publishing and domain mechanisms are the followings. As the exploration of a research area proceeds, the considerations of the publications change. The main phases are: (1) settling the relevant factors (variables), (2) exploring associative or causal connections, and (3) determining numeric parameters.

We are mainly interested in processing papers of the second class: since these consider connections between variables, we await that those variables will appear together in a paper which depend on each other. This suggests that the dependency structure of the distribution describing the co-appearance of the concepts will be similar to that of the real-world domain. Hence if we can learn the generative

models of publications (w.r.t. what variables are mentioned together), then these models will fit the corresponding real-world area as well.

B. Learning domain models based on textual data

The main steps of learning are, as follows:

- Input data are: free-text articles and the set of short "kernel" descriptions of the variables.
- These are converted into binary vectors representing which words are contained in them.
- The relevance of a keyword to an article is determined by the similarity of the vector of its kernel description and the one of the article. Based on this, we assign to each article a binary vector representing which variables are relevant to it (for details see [3]).
- The model structure can now be learned by any standard learning algorithm, see e.g. [4].

C. Comparing results with expert knowledge

The structure of the network encode qualitative relations of the domain. To evaluate a model w.r.t. an other, we may consider how many of the pairwise causal relations of the nodes remained in it w.r.t. the model provided by the expert. The possible relations of two nodes are (in weakening order): (1) there is an edge between the variables, (2) there is a directed path between the variables, (3) the two variables have a common ancestor, and (4) none of the previous. The ideas of this section were discussed in details in [5].

IV. Annotated Bayesian networks

As we have seen above, prior knowledge provided by experts can take an important role in model construction. The basic idea of annotated Bayesian networks is to extend BNs by assigning textual descriptions to nodes and/or structures. Using these annotations, we can formulate expressions like: $\forall X_1, X_2$*: the annotations of* X_1 *and* X_2 *contain* $string_1$ *and* $string_2$*, and* $X_1 = X_2$ or regarding the structure $\forall X_1, X_2$*: if the annotations of* X_1 *and* X_2 *contain* $string_1$ *and* $string_2$*, then there is a directed path between* X_1 *and* X_2.

Since a BN defines a distribution over atomic events (the full instantiations of the nodes), and there exists a posterior distribution over structures conditional on observations ($P(G|D)$), through the equation $P(expr|D) = \sum_{G:\ expr\ is\ true\ in\ G} P(G|D)$ we can compute the probability of any such expressions being true, assuming that only non-textual objects are quantified.

Annotated Bayesian networks provide a first-order, yet finite extension of BNs, capable of incorporating textual information concerning the variables (like kernel descriptions) or even the domain itself (annotations of networks structures). Hence, they provide a computationally tractable, free-text-based first-order knowledge representation language, able to coherently deal with uncertainty through probabilities. For details, see [6].

References

[1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Francisco, CA, 1988.

[2] C. Yoo, V. Thorsson, and G. Cooper, "Discovery of causal relationships in a gene-regulation pathway from a mixture of experimental and observational dna microarray data," in *Proceedings of Pacific Symposium on Biocomputing*, vol. 7, pp. 498–509, 2002.

[3] P. Antal, G. Fannes, Y. Moreau, D. Timmerman, and B. D. Moor, "Using literature and data to learn Bayesian networks as clinical models of ovarian tumors," *Artificial Intelligence in Medicine*, 30, 2004.

[4] D. Heckerman, D. Geiger, and D. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, 20:197–243, 1995.

[5] P. Antal and A. Millinghoffer, "Learning causal Bayesian networks from literature data," in *Proceedings of the 3rd International Conference on Global Research and Education, Inter-Academia'04*, 2004.

[6] P. Antal and A. Millinghoffer, "A probabilistic knowledge base using annotated Bayesian network features," in *Proc. of the 6th Int. Symp. of Hungarian Researchers on Computational Intelligence*, pp. 366–377, 2005.

DISCOVERY OF CAUSAL RELATIONSHIPS IN PRESENCE OF HIDDEN VARIABLES

Gábor HULLÁM
Advisors: György STRAUSZ, Péter ANTAL

I. Introduction

Causal relationships have a more and more accepted role in modern science. In the majority of models and mechanisms there is a casual (cause–effect) relationship between the components and in many domains (e.g.: medicine), the major goal of scientific research is the identification of the causes of certain phenomena. One of the most commonly used tools to represent causality is the *Causal Bayesian Network* (CBN), which is a directed acyclic graph, where vertices represent domain variables, while edges represent the causal relationships between them.

II. Constructing CBNs

The most straightforward way of constructing a CBN is to conduct controlled experiments. In order to decide whether a variable X_i has causal influence on some variable X_j we compute the marginal distribution of X_j by using the *truncated factorization* formula [1], under interventions $do(X_i = x_i)$, –that is we set certain variables X_i deliberately to values x_i– and test whether the distribution is sensitive to x_i.

In most cases control is not possible, due to financial, technical or ethical reasons. Then we have to resort to methods that are capable of finding an appropriate CBN –in terms of representing the domain– using only observational data. One possible option is to apply a series of transformations based upon the *do calculus* [1], which provides a method of verifying claims about interventions. Basically, in certain cases it enables us to determine the effects of $do(X_i = x_i)$ without the actual experimental data.

Another established approach to learn CBNs or at least to infer causal relations uses observational data by assuming the so called Causal Markov condition (all relations are causal), stability (exact representability with BNs) and the absence of hidden variables. However, the assumption of absence of hidden variables is not realistic in many domains.

III. Hidden Variables

In many cases we can observe only a portion of variables that describe the relevant aspects of a certain domain. Therefore the variable set of a domain usually consists of observed variables whose values are specified in the data set and a set of hidden variables that –for some reason– were not observed. If we assume incorrectly, that the domain in question consists of observable variables only and that is not the real case, then it can have an adverse effect on the learning procedure. However, not every hidden variable should be taken into consideration. If a hidden variable H is a leaf or a root with only one child or if H is isolated then H does not contribute to the representation of the domain, thus it can be ignored. We can conclude that taking hidden variables into account during a learning process can only be beneficial if they are connected to many –but at least two– observed variables. In the following sections we present different methods to handle hidden variables.

A. Inductive Causation

The IC* algorithm is one of the first algorithms that aimed the recovery of latent structures (i.e. causal structures with hidden variables). It is based upon the observation that any latent structure has a

projection where every hidden variable is a parentless common cause of exactly two nonadjacent observed variables (for further details see [1]). So instead of searching through the unbounded space of all possible latent structures, the algorithm has to perform a search confined to graphs with such structures. As the first step, the IC* identifies adjacent variables and places an undirected link between them, then it searches for V-structures (variable triplets, where two non-adjacent variables have converging arrows towards the third one) and directs links accordingly. Finally, it adds arrowheads and marks to certain undirected links based on the directionality rules (described in [1]) and thus, it creates an annotated partially directed graph. The marked links are guaranteed to be genuine (i.e. there is no hidden variable influencing them) while the others represent an ambiguity (i.e. it is possible that a hidden variable mediates the dependence between them).

B. Score-based approach

Another possible approach is to use scoring functions that evaluate each candidate network with respect to the training data and then select the best according to that metric. There are many existing scoring metrics such as the *belief scoring function* [2] or the one based on the principle of *minimal description length* (MDL) [3] and the *Model Selection EM* algorithm [4] which originates from the latter.

The MDL score consists of two terms. The first term is the log-likelihood of the Bayesian network given a certain data set. The higher the log-likelihood is the closer the network is to modeling the probability distribution in the data set. The second term is a penalty term which encodes the notion: the simpler the better, hence we generally favor simpler networks in accordance with the semantic form of Occam's razor. In case of a complete data set, where all variables are observable and all have a value assigned to them, there is a closed form solution for the parameters that maximize the log-likelihood score given a specific network structure. However, in the presence of hidden variables we must find an optimal parameter setting first in order to evaluate the score (for details see [4]). Therefore a parametric search such as EM is needed to find the best choice of parameters for each network structure candidate.

The MS-EM algorithm uses a scoring metric that is based on the MDL principle. Since some of the values needed for the evaluation of the score are not present in the data, due to unobserved variables, only an expected score can be calculated by taking expectation over all possible values the unobserved variables might take. At each stage MS-EM chooses a model and parameters that have the highest expected score given a previous assessment.

IV. Conclusion

These inference and learning methods were developed for the so called monolithic Bayesian network representation, in which there are no regularly repeating substructures. However, in many application domains such Bayesian networks are appropriate, for example in temporal probabilistic models as in the compact representation of a Hidden Markov Model. The extension to such highly-structured probabilistic representations are yet unsolved, which constitutes the primary goal of our future research.

References

[1]J. Pearl, *Causality: Models, Reasoning, and Inference*, Cambridge University Press, 2000.

[2]D. Heckerman, D. Geiger, D. M. Chickering**,** "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data ", *Machine Learning*, 20(3): 197-243, 1995.

[3]W. Lam, F. Bacchus, "Learning Bayesian Belief Networks: An Approach Based on the MDL Principle", *Computational Intelligence*, 10: 269-293, July, 1994

[4]N. Friedman, "Learning Belief Networks in the presence of Missing Values and Hidden Variables", *Proc. 14th International Conference on Machine Learning*, pp.125—133, Nashville, Tennessee, USA, July 8-12 1997.

EFFICIENT FRAMEWORK FOR THE ANALYSIS OF LARGE DATASETS REPRESENTED BY GRAPHS

Tamás NEPUSZ
Advisors: Fülöp BAZSÓ (MTA RMKI), György STRAUSZ

I. Introduction

Many complex systems in engineering, society or nature can be characterised as networks or graphs, where the graph vertices represent functional components or modules and the edges represent links or relations between them. In computer sciences, they can be used to describe the router-level structure of the Internet [1], the connections of individual web sites [2] or the graph of functions and procedures calling each other in a large software project [3]. In social sciences, graphs describe the relationships among members of a social structure; in biology, nerve cells or brain areas form diverse complex networks; in linguistics, graphs can be used to represent co-occurences of words in a sentence, revealing some kind of regularity in an otherwise irregular structure called the human language.

The study of complex systems has witnessed a great increase of interest during the last few years. Several free software packages have been developed to help the work of researchers involved in complex networks [4] [5], but all of them have shortcomings like platform-dependency, closed source code or the inability to cope with large datasets consisting of tens of thousands of vertices and millions of edges. To overcome these drawbacks, I will introduce an open-source and platform-independent software called SNA ("Sixdegrees Network Analyser") [6] designed particularly for the analysis of large graphs and networks. SNA aims to be a comprehensive application implementing most graph algorithms used in the field of graph analysis and providing an easy-to-use, Matlab-like environment to sketch and implement new algorithms in a high-level language.

II. Design considerations and software architecture

Since the software is dealing with large datasets, the first aim of the implementation is efficiency above all. Another design goal was to allow the user to program graph algorithm prototypes quickly, without the need to implement it in C (which can otherwise be unavoidable because of performance reasons). These two requirements may seem controversial, because the first one suggests using a low- or medium-level language, while the latter one is easier to achieve in a high-level scripting language. SNA uses a hybrid approach: most of the core functionality related to graphs are provided by a C library called `igraph`, while the actual functionality of `igraph` is exported to the Python language [7] through a dynamically loadable module. Users of SNA use an embedded Python interpreter to access the functionality of `igraph` on a window-based GUI.

On the implementation level, graphs inside `igraph` are represented with edge lists, but the basic edge list structure is extended with redundant members to ease the query of incoming and outgoing edges for a given vertex, while the storage requirements for a graph with v vertices and e edges is still only $O(v + e)$. Attributes can be used to store information not related directly to the structure of the graph, like weights of the edges. Besides the basic functionality, the `igraph` library includes deterministic and non-deterministic functions for the generation of different structures (rings, trees, Erdős-Rényi random graphs and so on) and for the calculation of the most common measures (e.g. graph diameter, closeness centrality, edge and node betweenness). Since the language of SNA is based on a real scripting language, one can easily implement various graph algorithms without losing the efficiency of the `igraph` library. To illustrate this, one can take a look at the implementation of the

Google PageRank algorithm [8] in SNA: it is only 15 lines long.

III. Benchmarks

SNA has been compared to several frequently used graph analysis softwares. All test cases were executed on the same directed Barabási-Albert (BA) graph with 1000 nodes and an out-degree of 20 for every node. BA graphs were chosen because they show most of the basic characteristics of real-world networks (e.g. small-world property), while they are easy to implement. The test cases were the following: calculation of the diameter of the graph (assuming that the edges are undirected), the Google PageRank of every node, an unweighted spanning tree, betweenness centralities (the amount of shortest paths passing through a given vertex) and closeness centralities (the sum of distances from a given vertex to all other vertices) for each vertex.

The tests were executed on a notebook with Intel Celeron M 1.4 GHz processor and 512 Mbytes of RAM. The results are presented in Table 1. Missing results mean the lack of the given functionality in the tested graph analysis software. Execution times in Pajek [4] and Visone [5] were measured by stopwatch because of the lack of exact time measurement functionality.

Table 1: Benchmark results for different graph analysis softwares

	Diameter	PageRank	Spanning tree	Betweenness	Closeness
Visone [5]	–	$\approx$ 56 sec	–	$\approx$ 16 sec	$\approx$ 14 sec
NetworkX [9]	27.16 sec	3.1 sec	4.73 sec	–	33.03 sec
Pajek [4]	$\approx$ 5 sec	–	–	$\approx$ 5 sec	$\approx$ 6 sec
SNA	2.46 sec	0.04 sec	0.01 sec	3.58 sec	6.47 sec

IV. Conclusion

A prototype of an efficient graph analysis framework has been presented. SNA allows the user to create and manipulate graphs using a simple language based on Python, while outperforming most of the common graph analysis packages for large datasets. Further work needs to be carried out to optimize some of the calculations and provide user documentation.

References

[1] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *SIGCOMM*, pp. 251–262, Cambridge, MA, USA, Sept. 1999.

[2] R. Albert, H. Jeong, and A.-L. Barabási, "Diameter of the world-wide web," *Nature*, (401):130–131, Sept. 1999.

[3] C. R. Myers, "Software systems as complex networks: Structure, function and evolvability of software collaboration graphs," *Physical Review E*, 68(4):046116.

[4] V. Batagelj and A. Mrvar, "Pajek – program for large network analysis," *Connections*, 21(2):47–57, 1998.

[5] M. Baur, M. Benkert, U. Brandes, S. Cornelsen, M. Gaertler, B. Köpf, J. Lerner, and D. Wagner, "visone - software for visual social network analysis," in *Proc. of the 9th International Symposium on Graph Drawing*, pp. 463–464. Springer-Verlag, 2002.

[6] T. Nepusz, "Sixdegrees network analyser," 2005-2006,
URL: http://sna.sourceforge.net.

[7] G. van Rossum, *An Introduction to Python*, Network Theory Ltd., Apr. 2003.

[8] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[9] A. Hagberg and P. Swart, *NetworkX Manual*, Los Alamos National Laboratory, 2005,
URL: http://networkx.lanl.gov.

EFFICIENT SPECTRAL OBSERVER WITH UNEVENLY SPACED DATA

Károly MOLNÁR
Advisor: Gábor PÉCELI

I. Introduction

The theory of spectral observers is a well-studied area, where the Fourier spectral analysis is performed in real-time, by ongoing recursive calculations [1]. The efficient algorithms are based on the state-variable formulation and the results of observer theory. This approach is highly attractive, but it generally requires the ongoing calculation of the observer gain.

In case of conventional signal sampling, when the samples are evenly placed in time, explicit expressions are already developed to calculate the gain values [2]. In this case all the coefficients can be calculated offline, in advance, which efficiently simplifies the algorithm, making it usable also in hard real-time embedded systems, typically implemented on digital signal processors.

Spectral observers can also be used in sensor networks, which are distributed signal processing systems, where the cooperating nodes perform real-time data acquisition and signal processing. The nodes are interacting by a common communication medium, such as Ethernet, ZigBee radio, etc. These communication channels are not error-free, effects such as latency and packet loss have to be considered if the signal samples are transmitted between two nodes, or between a sensor node and a processing node.

In this paper we propose an efficient spectral observer that operates with unevenly placed signal samples. Such an observer is not sensitive to irregularities in sampling such as delayed or missing samples due to latency and packet-loss. Our solution is based on the idea presented in [3]. Unfortunately, in the general case, the observer gain calculation involves numerous matrix multiplications, that prevents the implementation of the algorithm in hard real-time embedded systems. We propose an efficient, realizable solution based on complex-resonators.

II. Spectral observer with unevenly spaced data

The discrete-time model of the observed system is:

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) \tag{1}$$

$$y(t_k) = \mathbf{c}^T \mathbf{x}(k),$$

where the time instants t_k are not necessarily spaced evenly. In the related system

$$\mathbf{z}(k+1) = \mathbf{A}(k)\mathbf{z}(k) + \mathbf{g}(k)[\, y(t_k) - \mathbf{c}^T \mathbf{z}(k)] = \tag{2}$$

$$= [\mathbf{A}(k)\mathbf{z}(k) - \mathbf{g}(k)\mathbf{c}^T]\, \mathbf{z}(k) + \mathbf{g}(k) y(t_k) = \mathbf{F}(k)\mathbf{z}(k) + \mathbf{g}(k)\, y(t_k).$$

$\mathbf{F}(k)$ is termed the identity observer of the system (1). The error between the state variables of the observed system and the observer is: $\mathbf{x}(k+1)$-$\mathbf{z}(k+1) = \mathbf{F}(\mathrm{k})[\, \mathbf{x}(k)$-$\mathbf{z}(k)]$. This error can be made to be zero if the observer gain values ($\mathbf{g}(k)$) are chosen to ensure that $\mathbf{F}(n\text{-}1)\ \mathbf{F}(n\text{-}2)\ldots\mathbf{F}(0) = 0$. Such an algorithm is given in [3], which calculates the $\mathbf{g}(k)$ values in an ongoing manner.

III. Simplified observer gain calculation

Based on the above presented structure, we propose a spectral observer where the observed system is modeled by complex resonators. The initial complex values of the state variables are the

Fourier coefficients and the $\mathbf{A}(k)$ matrix is diagonal: $\mathbf{A}(k) = \text{diag}< z_0(k), z_0(k)^*, z_1(k), z_1(k)^*,\ldots>$. Where $z_m(k) = e^{j2\pi m\Delta(k)/N}$ and $\Delta(k)$ is the time difference between t_{k+1} and t_k. (* means complex conjugate.) In this case, considerable simplifications can be made, and it is possible to derive explicit formulas to calculate $g(k)$. If $A(k)$ is a 2x2 matrix (one complex resonator), then $g(k)$ is the following:

$$g_0(k) = z_0(k)\, z_0(k\text{-}1) / [z_0(k\text{-}1) - z_1(k\text{-}1)] \qquad (3)$$

$$g_1(k) = g_0(k)^*.$$

Note that the matrix multiplications are eliminated and the g values depend only on the previous z values. Explicit formulas for more systems with more Fourier components are not yet developed, as due to complexity of the calculations, the formulation is not trivial.

The complex-resonator based observer inherently reproduces the spectral components of the input signal at its output, therefore the two signals can be compared in the time domain. In Fig.1. a sine wave $y(t_k)$ is shown versus the estimated output of the 2x2 observer. The '+' signs represent the $y(t_k)$ values, the circles are the calculated values of the spectral observer. $y(t_k)$ is a sine wave with unevenly spaced samples, with a step in the amplitude at sample nr. 125. Note that the error between the two signals disappears after 3 samples following the step in the input.

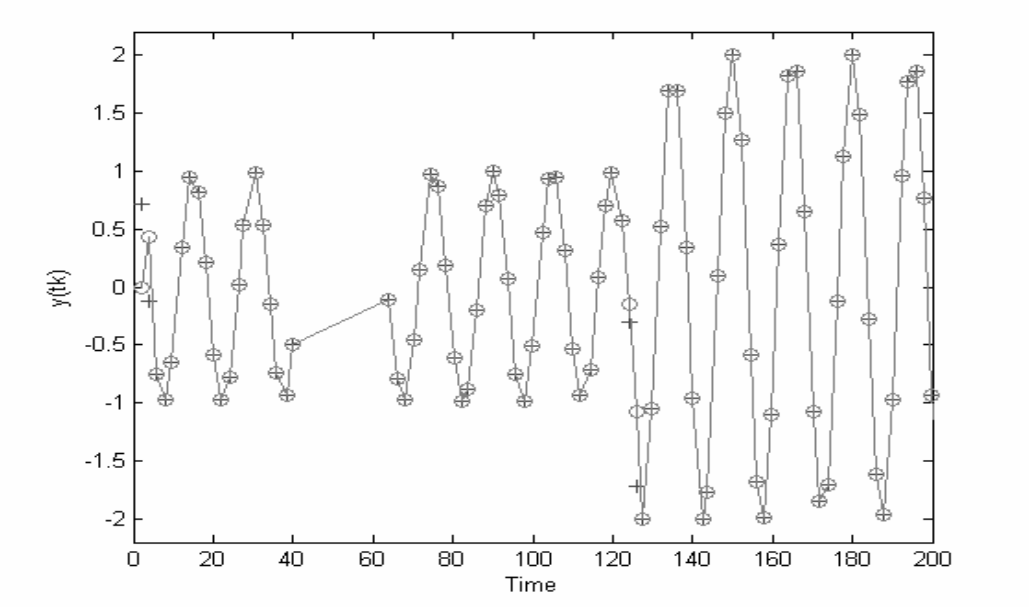


Fig.1. Unevenly spaced signal and output of the spectral observer

IV. Conclusion

In the paper we propose an efficient spectral observer with unevenly spaced samples. The complex-resonator based structure and the proposed algorithm works in Matlab simulation environment, however the efficient observer gain calculation is only formulated for simple cases. The next step is to formulate a more general calculation for arbitrary number of Fourier components.

References

[1] G. H. Hostetter, "Fourier Analysis Using Spectral Observer," *Proceedings of the IEEE*, Vol. 68, No 2, Feb. 1980.

[2] G. Péceli, "A Common Structure for Recursive Discrete Transforms," *IEEE Trans. on Circuits and Systems*, Vol. CASS-33, No 10, Oct. 1986.

[3] G. H. Hostetter, "Recursive Discrete Fourier Transformation with Unevenly Spaced Data," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-31, No.1, Feb. 1983.

ROBUST SINE WAVE FITTING IN ADC TESTING

Attila SÁRHEGYI
Advisor: István KOLLÁR

I. Introduction

The IEEE standard 1241-2000 [1] defines ADC testing methods which make use of least squares (LS) sine wave fitting algorithms. Theoretically, the full–scale input would be the best solution during the test. If so, we get samples from every quantum level. It is also needed in order to get reasonably accurate parameter estimations.
Adjusting a suitable amplitude is difficult and not always possible. On the one hand, too high amplitude results in signal degradation, on the other hand, the smaller the amplitude the more quantum level will be missed. Moreover, if the amplitude of the input signal is too small the ”pseudo quantization noise” model is getting to lose its validity. This means that quantization noise becomes less uniformly distributed and becomes more dependent on the input signal.

II. The Least-Squares Sine Wave Fitting Method

The standard [1] proposes two sine wave fitting procedures. Both of them minimize the following sum of the errors:

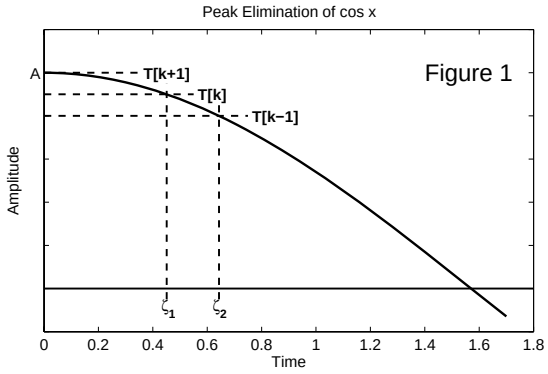
$$S = \sum_{n=1}^{M} (y_n - A\cos(\omega t_n) - B\sin(\omega t_n) - C)^2 = (\mathbf{y} - \mathbf{D}\hat{\mathbf{x}})^T (\mathbf{y} - \mathbf{D}\hat{\mathbf{x}}) = \mathbf{e}^T\mathbf{e}, \qquad (1)$$

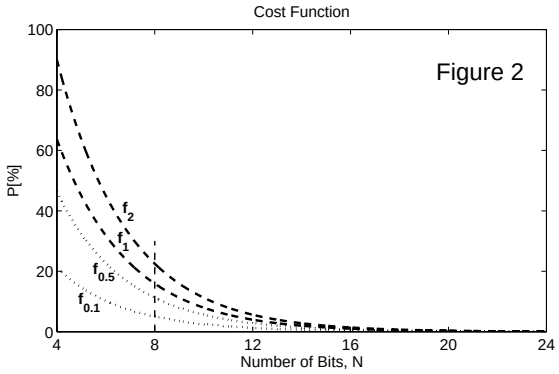
where $\hat{\mathbf{y}} = \mathbf{D}\hat{\mathbf{x}}$ is the fitted model which is linear in $\hat{\mathbf{x}} = [A\ B\ C]^T$ if ω is known. Otherwise, it is nonlinear in $\mathbf{e}$ and an iterative procedure is executed. In this paper only linear least squares fitting is being considered. The estimate of $\hat{\mathbf{x}}$ minimises S with respect to $\hat{\mathbf{x}}$, which is $\hat{\mathbf{x}} = \left[\mathbf{D}^T\mathbf{D}\right]^{-1}\mathbf{D}^T\mathbf{y}$. If the error sequence was random, zero-mean Gaussian and white, the estimate would be unbiased, minimum variance and would coincide with the maximum likelihood estimation. If Gaussian conditions are not met the estimate becomes non–minimum variance and might become biased. Nevertheless, for the general case LS estimate is asymptotically unbiased and asymptotically minimum variance as M tends to infinity.

III. Peak Samples Elimination

In sine wave testing of ADC’s the above mentioned assumptions are far from being true, especially when the sine waves are covering less than, say, 60 quantum levels. The quantization error looks more or less like sawtooth except the sine peaks which are responsible for the strong peaks in its probability density function (PDF). The position of this peaks depend on the amplitude and the dc value. So, this PDF is neither uniformly distributed nor zero mean which are needed for pseudo quantization noise model. This effect can be decreased by eliminating the samples around the peak of the sine wave [2]. By this elimination of the ’pathological’ bins, the fit yields approximately unbiased amplitude estimation, but the variance of the estimation increases.
The variance increase and the number of ’pathological’ bins are the subject of recent interest. Let us observe the modified algorithm if only the amplitude is unknown. In the present case $[\mathbf{D}^T\mathbf{D}]^{-1} = 1/\sum_{n=1}^{M}\cos^2(\omega_0 t_n)$, $\mathbf{D}^T\mathbf{y} = \sum_{n=1}^{M} y_n\cos(\omega_0 t_n)$ and the variance is directly proportional to the first expression, i.e. $\sigma_A^2 = K\cdot(1/a)$ where $a = \sum_{n=1}^{M}\cos^2(\omega_0 t_n)$. The sample elimination

decreases the value of a by $b = \sum_p \cos^2(\omega_0 t_p)$, where p denotes the peak samples. If so, the new variance is $\sigma^2_{A^\star} = K \cdot (1/[a-b])$ and after some manipulation $\sigma^2_{A^\star} = \frac{1}{1-b/a} \cdot \sigma^2_A$. Here, b/a represents the proportion of the eliminated energy. Thus, after rewriting to continuous time the $P = \frac{b}{a} = \frac{\int_0^{\zeta_c} \cos^2(x)}{\int_0^{\pi/2} \cos^2(x)}$ cost function seems acceptable for further analysis. In Figure 1 a cosine function (FS=2A) and three code transition levels ($T[k-1]$, $T[k]$, $T[k+1]$) can be seen and these 'peak bins' are filled. This figure shows that if samples in one or two 'peak bins' are dropped, $[0, \zeta_1]$ or $[0, \zeta_2]$ intervals will be eliminated in the cost function (P). For instance, if the quantizer bit number N=8 and FS=2A, the code bin width will be $Q = \mathrm{A}/2^{N-1}$ and $\zeta_1 = 1/\sqrt{2^{N-2}} = 1/8$ radians. Hereafter, P can be expressed as a function of the quantizer bit number $P_{\zeta_1}[N] = f_1, P_{\zeta_2}[N] = f_2$ (see Figure 2).





$$f_1 = \frac{4}{\pi} \cdot \frac{1}{\sqrt{2^{N-2}}} \quad f_2 = \frac{4}{\pi} \cdot \frac{1}{\sqrt{2^{N-3}}} \tag{2}$$

Full–scale input is just a dream, but f_1 and f_2 functions give the worst case solutions if one or two 'peak bins' are dropped. For example, suppose that the peak bin is partly filled i.e. the amplitude is somewhere between two transition level $T[k] \leq A \leq T[k+1]$. In the present case if only one peak bin dropped the variance increase will be less than according to f_1. This can be seen in Figure 2. if the peak bin is only half–filled ($f_{0.5}$) or 10%–filled ($f_{0.1}$).
According to the above the variance increase can be estimated by $\sigma^2_{A^\star} \leq \frac{1}{1-P[N]} \cdot \sigma^2_A$ where $\sigma^2_{A^\star}$ denotes the variance of the modified estimation.

IV. Results and Conclusion

- This elimination of the peak samples renders the quantization noise approximately uniformly distributed, hence, its the LS estimate become less biased.
- It can be seen in Figure 2 the higher the number of bits the lower the variance increment this algorithm results. This is not so surprising but prove that this modification does not spoil the estimate significantly at higher number of bits.
- Until now, I have investigated the noiseless ideal quantizer, but this method can equally well used for real ADC's, if the noise is zero–mean, white and Gaussian or if it has a special PDF.

References

[1] IEEE Standard 1241-2000, *IEEE Standard for Terminology and Test Methods for Analog–to–Digital Converters*.

[2] I. Kollár and J. Blair, "Improved determination of the best fitting sine wave in adc testing," *IEEE Trans. on IEEE Instrumetation and Measurement*, 54(5):1978–1983, Oct. 2005.

SENSORLESS ROTOR POSITION ESTIMATION BASED ON INDUCTIVITY MEASUREMENT

András ZENTAI
Advisor: Tamás DABÓCZI

I. Introduction

In electric power assisted steering (EPAS) systems an electric machine is connected to the mechanical steering system to reduce driver's work by generating torque in the appropriate direction. Electric actuators have great advantages compared to hydraulic systems. Steering properties can be changed on-line according to vehicle speed, quality of road surface or other significant parameters. They are also environment-friendly, because of their higher efficiency and because they do not contain environmentally harmful hydraulic steering fluid. Nowadays three phase permanent magnet synchronous machines (PMSM) are used in EPAS systems. In PMSM type of machines rotor position information (angle between rotor permanent magnet axis and magnetic axis of stator winding) is essential in motor control application, because without knowing the exact rotor position the required torque cannot be generated [1]. In conventional applications a position measurement sensor is dedicated to deliver angle position information. There are different methods to omit position sensor [2],[3]; in this paper a method using inductivity measurement is described.

II. Measurement method

To estimate rotor position without a position sensor there is no need for hardware modification, only the motor control algorithm should be modified. Sinusoidal voltage output signals should be superponed with short ($\sim 50\ \mu s$) measurement pulses. 6 different pulses should be generated. 3 if one of the three phases is connected to U_{DC} and the other two is connected to GND. Other 3 if two of the three phases is connected to U_{DC} and one is connected to GND, as it can be easily seen in Figure 1a. The 6 different pulses are the measurement signals of three different resultant inductivity of the machine phases. This inductivity values are varying as a function of rotor position, but their sinusoidal components phase which is depending of rotor position are shifted with $\pm 2\pi/3$ radian. From current answer to the voltage pulses resultant inductivity of the machine phases can be calculated using (1), where Δu is the measurement signal amplitude, Δi is the current answer, Δt is the measurement pulse with. To estimate rotor position using inductivity change of the stator phases a specially designed electrical machine is needed. Inductivity varies as a function of rotor position almost in every type of PMSM. But if this phenomenon is not emphasized during design period, inductivity change can not be measured in automotive environment. In the test environment inductivity variation is 30% and approximately sinusoidal as a function of rotor position (2), where L is the resultant inductivity of the machine phases, L_0 is the average constant part of the inductivity, L_d is the amplitude of the sinusoidal part of the inductivity, γ is the rotor angle, ϑ is the phase offset. Main disadvantage of this method is that inductivity varies two times faster as the rotor angle. This phenomenon can result π rad failure in position estimation. Main goal of research was to improve an existing position detection algorithm. For this reason a measurement environment was built to test new implementations.

$$L \approx \frac{\Delta u}{\Delta i/\Delta t} \tag{1}$$

$$L(\gamma) = L_0 + L_d \cdot \sin(2\gamma + \vartheta) \tag{2}$$

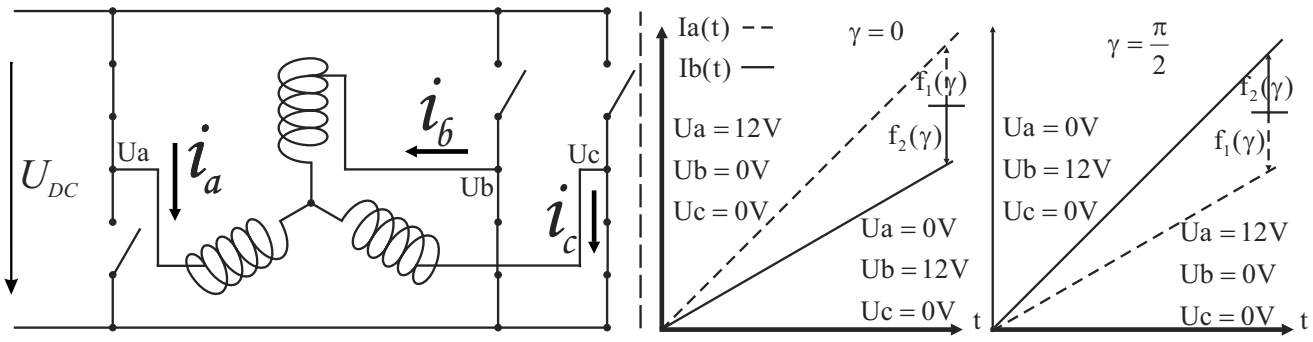


Figure 1: Electric circuit of motor windings and current response in different angle positions

III. Building a test environment

Efforts to build a test environment using dSpace Autobox hardware and MATLAB/Simulink software was also done. Main advantage of Autobox hardware is that Simulink models – which are previously used for simulations and calculations – can be compiled to its target processor language and run real-time, directly without extra programming work.

Autobox board is based on a Power PC controller, contains A/D measurement card with 20 A/D converter, PWM controller card, and other hardware for I/O functionality. Setting up real-time measurement in Autobox environment was a challenging job, because voltage step signals are usually very short pulses. A/D converters are also not suited for such high speed signals. That is the reason why each channel is captured by 4 A/D converters; each of them has different conversion start phase. Using this solution current answer for the short measurement pulses can be captured.

IV. Conclusion

Main goal of this paper was to present a driving method for permanent magnet synchronous motors without using a dedicated position sensor. Main advantages of the method are:

- sensor cost can be eliminated,
- weight and volume of the construction can be reduced.

Main disadvantages of the method are:

- it can have π failure in the position measurement,
- generates hearable noise,
- it requires more computation capacity,
- not tested thoroughly in safe-critical applications.

Acknowledgment

The author thanks László Naszádos and Ferenc Illés (ThyssenKrupp Nothelfer Kft.) for helping better understanding theoretical background of the method. Financial, technical, and theoretical support of ThyssenKrupp Research Institute Budapest is also appreciated.

References

[1] J. M. D. Murphy and F. G. Turnbull, *Power Electronic Control of AC Motors*, Pergamon Press, Oxford, 2000.

[2] P. Vas, *Sensorless Vector and Direct Torque Control*, Oxford University Press, Oxford, 1998.

[3] M. Schrödl and M. Lambeck, "Statistic properties of the INFORM-method in highly dynamics sensorless PM motor control applications down to standstill," *European Power Electronics and Drives*, 13(3):22–29, Mar. 2003.

FPGA-BASED DEVELOPMENT ENVIRONMENT FOR AUTOMATIC LANE DETECTION

András BÓDIS-SZOMORÚ
Advisors: Tamás DABÓCZI (MIT)
Alexandros SOUMELIDIS, Zoltán FAZEKAS (MTA SZTAKI)

I. Introduction

Vision sensors and computer vision have become particularly important in traffic applications due to their easy installation, maintenance and their ability to provide detailed information about the scene[1]. Two major fields of their application are automatic vehicle guidance and traffic monitoring.

Automatic lane detection (ALD) plays an important role in vision-based lateral control of vehicles. Active research in the field resulted a high number of specific video processing approaches and techniques. However, the different environmental conditions in which they work safely still needs to be widened. Such systems have to meet severe requirements of safety, robustness and real-time processing and therefore they require lots of computational resources.

The automatic understanding of scene content in computer vision is realized by performing an abstraction of information together with a reduction of the quantity – in the pixel sense – of the information [2]. To perform the preprocessing stage in an automatic vehicle guidance system, FPGA devices seem to be suitable for an extreme reduction of pixel data by extracting edges from a video on-the-fly. Thus, as a first step, we focused on the development of a simple FPGA-based environment in which algorithms to be developed in the future can be easily embedded and tested. In the meantime a highly reconfigurable FPGA- and Linux processor-based network camera has been selected for future use in our research and development process.

II. The developed FPGA-based image preprocessing environment

A. Design of a VGA DAC extension board

For the preparation of a simple development environment for real-time image preprocessing, a Starter Kit Development Board was used with a Spartan-3 200 kgate FPGA chip. Since this board is unable to drive a display with a satisfactory quantity of colors and no extension board has been found for this purpose, a new extension board has been designed. The new card contains a 24-bit 300 MHz RGB DAC and can be interfaced to the FPGA development board and to a VGA-compatible display.

B. FPGA configuration design

Currently, the system implemented in Verilog HDL can display a 640x480 24-bit RGB still image in a 800x600@72Hz screen. An image can be downloaded and reread in RGB RAW format through serial port. The design contains an asynchronous transceiver, which organizes the 3 byte pixel data sequentially into 32-bit packets. This data is then paralelly written into the external SRAM by the SRAM controller implemented. In display mode, the FPGA generates the synchronization signals for the display and redirects the 24-bit pixels extracted from the 32-bit memory data to the DAC. The RGB data flows through a PixelProcessor module which is able to perform some basic operations on each pixel separately, on-the-fly. Currently, this module is not equipped with a cache memory, but it will be in the near future. This is required for executing image filtering operations on an intensity map in image space e.g. by using basic noise filtering and differential filtering masks for edge emphasis. An appropriate color space transformation and its inverse transformation has to be implemented as well, to be able to work on the intensity map without changing the chromaticity.

III. FPGA-based automatic lane detection

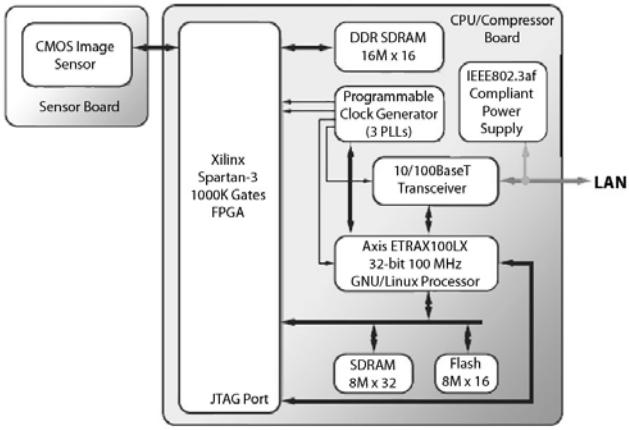
A. Automatic lane detection stages

The preprocessing stage for the extraction of vehicle location information from the acquired video consists of the following steps: Bayer-encoded image transformation to intensity-chromaticity space, noise filtering on the intensity map in image space, edge enhancement by using a gradient operator, Laplacien or compass mask, thresholding the resulted magnitudes and edge aggregation [1].

Higher level processing usually uses the powerful model-driven approach by estimating the parameters of a deformable road- or lane-template (e.g. Bayesian optimization procedure supposing a stochastic line-, spline- or clothoid-based model) [1]. In feature-driven approaches the road or lane features are extracted after edge aggregation and are arranged in feature vectors [3]. In the feature space, a predefined metric and a motion predictor can be used to detect estimation errors and make the system more robust. In stereo vision systems – and potentially this will be our case - the lane markings can be reprojected on the ground plane to extract vehicle coordinates.

B. Introducing a reconfigurable network camera

For performing the mono processing stages of the full detection process, an FPGA-based high resolution, high frame rate camera has been chosen. This contains a CMOS sensor, a Spartan-3 1000 kgate FPGA and a Linux-optimized processor with Ethernet support. By default, an open-source Theora OGG encoder, a memory controller, peripherals, a processor bus interface and a camera interface are configured into the FPGA. The default configuration will be modified to integrate specific lane detection preprocessing capabilities into the chip. Higher-level processing tasks will run in the processor.



C. Usage of the Reconfigurable Camera

The camera(s) will be installed on a car provided by a cooperating automotive partner company for testing. They will be connected to a host laptop in the development phase and therefore specific software is required on the host side which receives data from the camera(s) and displays and records a rough version of the original video stream together with the identified lanes for evaluation. The software will perform the final processing by combining information from the two sources, but later this algorithm will be placed into an embedded processor. The lane detection system will provide vehicle location information to the servo guidance system.

IV. Conclusion

An FPGA-based environment has been designed to test simple image preprocessing implementations in hardware. A network-capable reconfigurable camera also based on an FPGA chip was analyzed and selected for further development of the dual-camera lane detection system. We are at the beginning of a long-term development process and we have just entered a well progressed research field. A detailed and systematic analysis of the state of the art is still required in the future.

References

[1] V. Kastrinaki, M. Zervakis, K. Kalaitzakis, “A survey of video processing techniques for traffic applications”, *Image and Vision Computing*, pp. 359-381, Jan. 2003.

[2] A. Watt, F. Policarpo, *The Computer Image*, Addison-Wesley Longman, New York, 1999.

[3] Young Uk Yim, Se-Young Oh, “Three-Feature Based Automatic Lane Detection Algorithm (TFALDA) for Autonomous Driving”, *IEEE Trans. on Intelligent Transportation Systems*, 4(4):219-225, Dec. 2003.

DEPENDABILITY ANALYSIS AND SYNTHESIS OF WEB SERVICES

László GÖNCZY
Advisor: Tamás BARTHA

I. Introduction

Web service-based system integration became recently the mainstream approach to create composite services. Accordingly the dependability of such systems becomes more and more crucial. The service integrator (i.e. the provider of the main service) typically includes external services that are frequently out of its control. Nevertheless, it has to guarantee a predefined service level for its own service, thus he has to be aware of the dependability of the invoked services. Therefore, methodologies are needed for both the analysis and the synthesis of composite Web services. Hereby I propose a methodology for the dependability analysis of composite Web services by evaluating them as Multiple Phased Systems [1].

II. Dependability Analysis by Model Transformations

In [2] and [3] a methodology was suggested to evaluate dependability indicators of Web service flows.

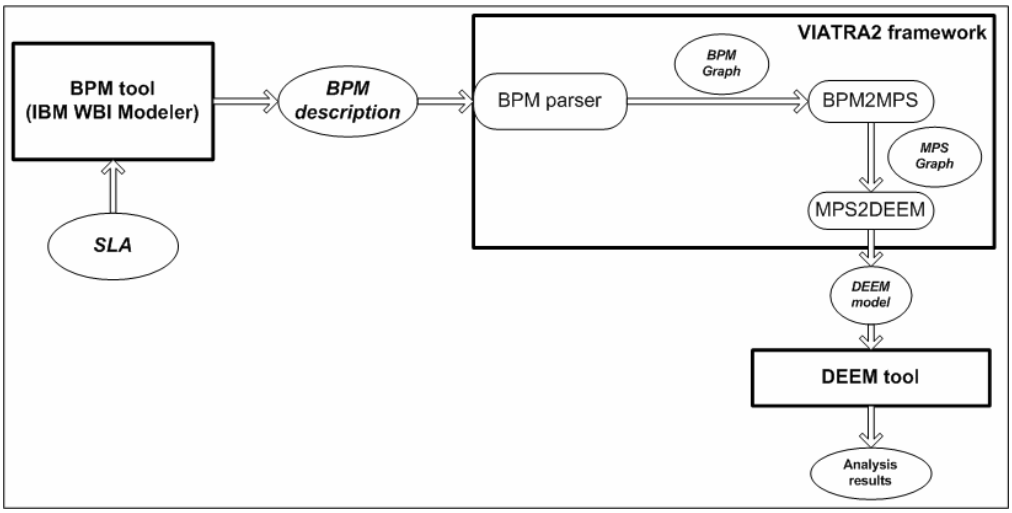


Figure 1: Dependability analysis of composite Web services by model transformations

This analysis takes a business process model, extended with the Service Level Agreement descriptions [4] of the external services and internal resources as input. This model is transformed into a Multiple Phased System (MPS) for dependability analysis. MPS is a class of systems whose operational life can be partitioned in a set of disjoint periods, called ”phases”. The configuration of MPS may change over time, in accordance with performance and dependability requirements of the phase being currently executed, or simply to be more resilient to a hazardous external environment. Business process models are considered as MPS because their operational life also consists of periods with different requirements and resource characteristics. The DEEM tool evaluates the characteristics of MPSs described as General Stochastic Petri Nets by using Markov Regenerative Process as solution algorithm.

The transformation is implemented in the VIATRA2 model transformation framework [5] in two steps as illustrated in Figure 1. The graph representation language of VIATRA (VTCL) is used to store both the metamodels of the business process description and the MPS model, together with the corresponding model instances. First, the business process description is parsed to VTCL, then a graph-based representation of the MPS is created. Finally, the source code of an MPS is generated for the particular modeling tool (DEEM).

The dependability measures of interest are −among others− the following:

- The probability that a client request fails (for different types of clients).
- Performability metrics which show the cost of dependability, i.e. which external services to invoke at given QoS parameters and price.

Sensitivity analysis is also performed for the SLA parameters such as response time and expected failure rate. Figure 2 shows the results of a sensitivity analysis and the evaluation of a performability measure.

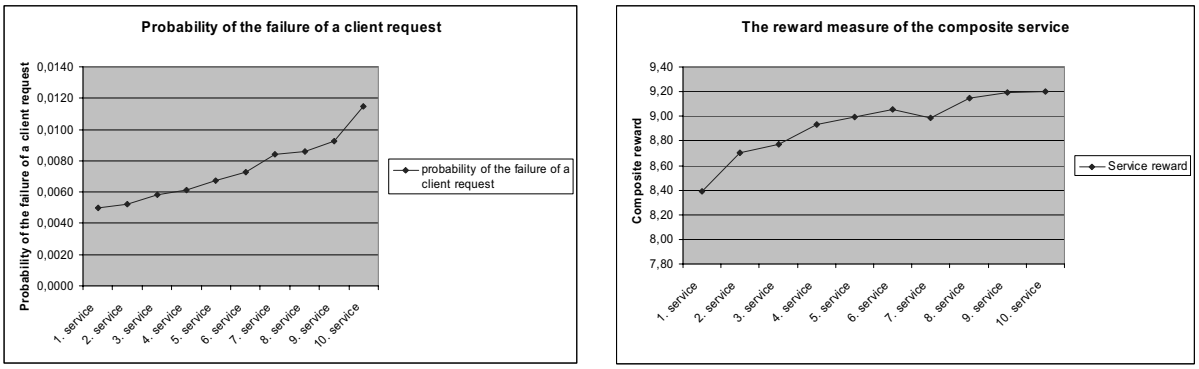


Figure 2: The results of the dependability analysis

III. Conclusion

The presented methodology aims at performing dependability analysis on Web service-based process models. This provides a very useful support to the service provider in choosing the most appropriate service alternatives to build up its own composite service. Future research objective is the support of dependability-driven service composition by code generation, which allows the designer to build high level models and then generate the XML descriptors of the functional and non-functional characteristics of the service. This work will be carried out in the SENSORIA Integrated Research Project [6].

References

[1] A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, and I. Mura, "Dependability modeling and evaluation of multiple-phased systems, using DEEM," *IEEE Transactions on Reliability*, 2004.

[2] L. Gönczy, "Dependability Analysis of Web Service-based Business Processes by Model Transformations", In Proc. of the *First YR-SOC Workshop*, Leicester, United Kingdom, 2005.

[3] L. Gönczy, S. Chiaradonna, F. Di Giandomenico, A. Pataricza, A. Bondavalli, T. Bartha, "Evaluating Dependability Indicators of Web Service-based Processes", submitted to the *Performance and Dependability Symposium* of the DSN 2006 conference.

[4] D. Menasce, V. A. F. Almeida. *Capacity Planning for Web Services: Metrics, Models, and Methods*. Prentice Hall, 2001

[5] *The VIATRA2 Model Transformation Framework*, Generative Model Transformer Project, The Eclipse Foundation. http://eclipse.org/gmt/

[6] *Software Engineering for Service-Oriented Overlay Computers* (SENSORIA), FP6-2004-IST Integrated Project, http://sensoria.fast.de/

ON USING ABSTRACTION TO MODEL CHECK DISTRIBUTED DIAGNOSTIC PROTOCOLS

Péter Bokor
Advisor: András Pataricza

I. Introduction

In my master thesis I presented a method to verify a specific diagnostic algorithm (called DD [1]) by using formal methods. I applied model checking as the means of verification, which performs exhaustive simulation of the system model. Even for few diagnostic nodes (processing elements, also called PEs) the simulation yielded state space explosion. In order not to lose on generality of the verification we introduced an abstraction of the system. The novelty of the approach was to exploit the symmetry of the problem which was arisen from the fact that non-faulty nodes (PEs) show similar behaviour. In fact, by applying the *one-correct-node abstraction*, i.e. despite modeling each correct node separately we defined one abstract node representing the correct ones, the size of the state space became feasible. To prove the soundness of the abstraction we pointed out that by non-deterministic assignment of the node variables all scenarios of the real system can be modeled.

Our ongoing research is aimed at verifying another diagnostic protocol. In our former work, even for the abstracted system, we restricted ourselves to a certain number of nodes assuming that the requirements (see in Section II.B.) also hold for an arbitrary number of PEs. The new diagnostic protocol also makes more realistic assumptions on the communication network (duplicated bus instead of fully connected network) and exposes further challenging issues (e.g. maintenance-oriented diagnosis).

II. The Diagnosis Problem

A. Application Field

The diagnostic protocol under verification has been developed in the DECOS project. DECOS [2] (Dependable Embedded Components and Systems) is an EU project aimed at establishing a framework for integrated design of dependable (embedded) systems. Instead of dealing with the *federated approach*, where each service of the system is implemented by a dedicated component, DECOS distributes the services (called DASs, Distributed Application Subsystems in the DECOS terminology) on the installed nodes (i.e. processing units equipped by local memory, network connector device, sensors, actuators, etc.). The *integrated architecture*, in contrast to the federated approach, has many advantages, e.g. it facilitates the application of COTS (Commercial-Off-The-Shelf) components in dependable systems and also enhances the cost-effectiveness of the design process.

B. Requirements

Generally, if verifying a diagnostic protocol we may want to show that the protocol is *complete* and *correct*. While correctness requires that non-faulty nodes will never be accused to be faulty (safety property), completeness assures that every faulty node will be detected eventually (liveness property). In many cases correctness and completeness are contradictory requirements that need a tradeoff to be satisfied. In fact, in case of a non-restrictive fault model ensuring correctness implies that completeness only holds under certain restrictions to the fault manifestation.

C. The Diagnostic Protocol

If dealing with verification of diagnostic algorithms the requirements completeness and correctness are general properties. However, under which constraints they are to be guaranteed depends greatly

on the system model (communication, fault hypothesis, etc.). While many approaches assume a fully connected network (like DD also does), which is not applicable in case of bigger systems, DECOS applies a Time Triggered Architecture (TTA). In TTA each node is connected to a duplicated bus and a scheduler a priori (at design time) assigns time slots to the nodes, where special devices (bus guardians) assure that in each slot only one node (the a priori scheduled one) issues messages to the bus. A detailed introduction of the DECOS diagnosis would be out of the scope of this paper, thus we only give the basic assumptions: (1) no restrictions on communication faults, (2) host faults are symmetric and (3) the diagnostic service may be either fail-silent or Byzantine (design decision).

Count-and-threshold [3] mechanisms are widely used in many application areas. The main idea is that each time a fault effect is recorded, a penalty value to the corresponding DUT (device under test; in DECOS the DUT is the node) will be assigned to indicate an increased evidence of the presence of the fault. When a fault-free behaviour is recorded after the detection of some fault effect, the penalty is progressively decreased, denoting a major trust on the absence of a permanent fault. In the DECOS diagnosis count-and-threshold will be a core part of the protocol (unlike in DD), hence its consideration in the verification process is of crucial importance.

III. Verification

A. Generality of the Verification

While verifying DD the straightforward encoding of the algorithm turned out to be infeasible for model checking (even for 4 nodes). With the one-correct-node abstraction we managed to reduce the state space, however, the model remained bounded to a given number of nodes: the model variables encoding the number of PEs could not be defined unbounded for finite model checkers. Now we are aimed at proving the protocol without any loss on generality.

B. Verification Environment and Techniques

As a verification platform we are using the SAL framework [4], which provides a "family" of model checkers. Beside the standard approach (symbolic model checking) it also provides a *bounded model checker*. BMC (Bounded Model Checking) [5] may increase the efficiency of model checking and also supports *infinite model checking*. However, the approach is only applicable for proving invariants (note that completeness is an eventual property), and also the implementation in SAL is an open issue. More research on that is planned to be done in the upcoming months.

References

[1] C. Walter., P. Lincoln, and N. Suri, "Formally verified on-line diagnosis," *IEEE Transactions on Software Engineering*, Nov. 1997.

[2] H. Kopetz, R. Obermaisser, P. Peti, and N. Suri, "From a federated to an integrated architecture for dependable real-time embedded systems," Tech. Rep.

[3] A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, and F. Grandoni, "Threshold-based mechanisms to discriminate transient from intermittent faults," Technical Report B4-17-06-98, IEI-CNR, June 17 1998.

[4] S. Bensalem, V. Ganesh, Y. Lakhnech, C. Muñoz, S. Owre, H. Rueß, J. Rushby, V. Rusu, H. Saïdi, N. Shankar, E. Singerman, and A. Tiwari, "An overview of SAL," in *LFM 2000: Fifth NASA Langley Formal Methods Workshop*, C. M. Holloway, Ed., pp. 187–196, Hampton, VA, June 2000. NASA Langley Research Center.

[5] A. Biere, A. Cimatti, E. M. Clarke, and Y. Zhu, "Symbolic model checking without BDDs," in *TACAS '99: Proceedings of the 5th International Conference on Tools and Algorithms for Construction and Analysis of Systems*, pp. 193–207. Springer-Verlag, 1999.

MODEL-DRIVEN DEVELOPMENT OF REAL-TIME EMBEDDED SYSTEMS

András BALOGH
Advisor: András PATARICZA

I. Introduction

Model-Driven Architecture (MDA) [1] has become a main trend in software development. It defines a development methodology, which highly relies on modeling and model-transformations. Using these techniques, the designer can concentrate on the functional aspects of the system, while all implementation-related information and the implementation of the system itself is automatically generated by software tools. The traditional application area of MDA is the domain of enterprise information systems.

However, there are several other domains that need support for handling the growing complexity of systems. The application of model-driven development methodology in embedded systems raises several challenges to tool developers. In contrast with the traditional EIS domain, where the main focus was on the functional requirements, embedded systems domain also requires the specification of non-functional (or Quality-of-Service) properties of systems. These properties have to be collected, and maintained during the development cycle, and must be enforced by the tools during code generation.

These additional requirements lead to a new, more complex development methodology and tool chain. We propose a model-centric approach for this problem in this paper that is based on the VIATRA2 [3] model transformation framework. We apply the basic approach to a safety-critical distributed domain, that is currently under development in project DECOS (Dependable Embedded Components and Systems – an EU Framework 6 Integrated Project) [4].

II. The Target Domain

Currently, in the automotive and avionics industry, the various functionalities of the on-board electronics are provided by separate, federated subsystems. This approach results in huge number of electrical control units (ECUs) that leads to increased costs and power consumption. These systems include both safety critical (SC) (e.g. x-by-wire) and non safety critical (NSC) (e.g. entertainment subsystem) applications.

To decrease the cost and complexity of on-board electronics, the most suitable solution is to integrate the subsystems into a cluster built from a relatively small set of computing nodes. This results in a mixed set of SC and NSC subsystems running in a common environment. To allow this mixed structure, the runtime platform has to ensure the proper temporal and spatial separation of different subsystems, while supporting the communication between the specified components.

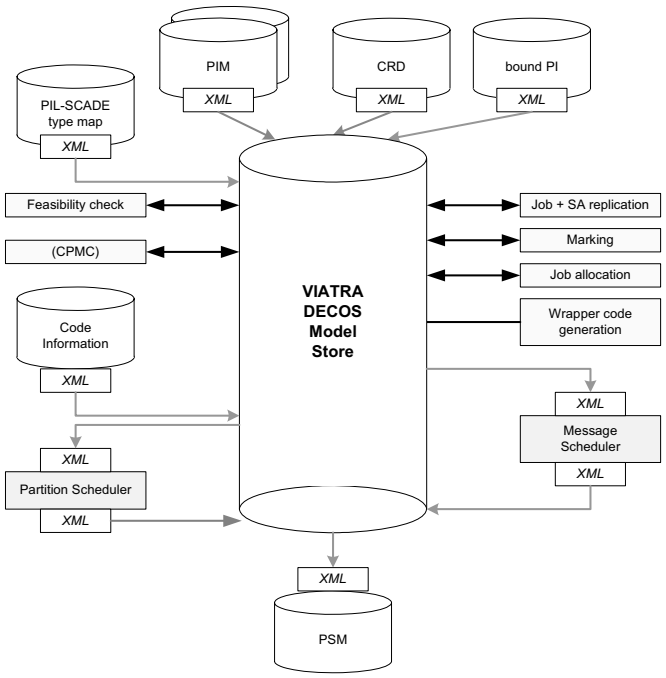
DECOS aims at integrating the various federated embedded computing subsystems into a single integrated system by a clear architecture that defines the required set of middleware services (e.g. messaging, group membership, global time) and the properties of platform operating systems (guaranteed separation of subsystems) [2].

III. Model-Driven Development in DECOS

The DECOS architecture supports complex applications; therefore we have to support the development in order to let the developers deal with this complexity. We have defined a domain-

specific platform-independent meta model (PIM) for distributed embedded systems that can be used to describe the functional, performance, and dependability aspects of the system, and we have also developed a platform-specific meta model (PSM) for the DECOS reference platform that also specifies all implementation-related properties of the systems.

The most important point of the development workflow is the mapping between the PIM and PSM, because we have to (semi-)automatically map the functional and also the non-functional requirements contained by the PIM to the actual hardware configuration (contained by the Cluster Resource Definition - CRD). We also have to take into account the actual Platform Interface parameters (timings, offered services, etc.).



The process requires a set of models be present simultaneously; therefore we have to use a generic model store. VIATRA2 [3] has been selected for implementation of the model store. The mapping itself cannot be full-automatic, because the developer may want to make some special restrictions (such as assigning specific jobs to specific HW nodes) called markings. These markings are integrated into the model and used by the subsequent steps.

The PSM generation starts with the job allocation, then the communication and task scheduling of nodes follows. The schedulers are third-party legacy tools. To allow the customization in each step, the user can review the results and make new markings, if needed. The consistency and feasibility of the PSM is continuously monitored by several special model transformations that search possible problems in the system model.

After the PSM generation is completed, the PSM can either be exported from the model store, or can be used directly to generate deployment configuration and wrapper code for the configuration of node operating system and middleware.

IV. Conclusions

MDA in Real-Time systems introduces new challenges for tool developers. Using the generic model store and model transformation infrastructure of VIATRA2 and some legacy tools we have succeeded to develop a PIM to PSM mapping that supports the specification of both functional and non-functional aspects of the systems. In contrast of the traditional full-automatic mapping, we also introduced a marking mechanism to support the manual, fine-grained customization of the system.

References

[1] The Object Management Group, *Model-Driven Architecture Information portal* http://www.omg.org/mda

[2] H. Kopetz, R. Obermaisser, P. Peti, N. Suri: *From a Federated to an Integrated Architecture fo Real-Time Embedded Systems*, - DECOS Technology paper

[3] *VIATRA2* – An Eclipse GMT subproject http://www.eclipse.org/gmt

[4] DECOS – An EU FW 6 IP http://www.decos.at/

AUTHORIZATION FRAMEWORK FOR SERVICE ORIENTED ARCHITECURE

Zsolt KOCSIS
Advisor: András Pataricza

I. Introduction

Setting up secure application architecture is very challenging. The Service Oriented Architecture design allows and requires centrally manageable security services, among them the authorization service is the key to build model based security infrastructure. Although the theory of different security models are well-known, the definition and coding of the authorization rules are not complicated, due to the lack of robustness and high performance the solutions based on these models are still not in everyday use. This paper introduces an industry scale solution based on standard Tivoli technology to set up a universal authorization service.

II. Authorization engine based on IBM Tivoli technology

There are several security models developed like the Bell- La Padula, Clark- Wilson model, or Role Base Access Control[2].Common in these models is that the runtime evaluations of the necessary rules are very resource consuming, considering that each object access needs to be verified.

The solution is based on Tivoli Access Manager (TAM). The TAM uses ACLs to evaluate the access requests, provides an authentication framework, and includes a robust authorization engine. Nevertheless the solution does not support the evaluation a business condition.

The standard TAM provides the following authorization logic:

- Role Based (standard ACLs)
- Protected Object Policy - POP
- Rule Based (dynamic condition evaluation)

The rule evaluation is rather slow, the POP conditions are rather limited, therefore we had to work out a solution being able to evaluate flexible business logic decisions and providing central authorization service to make Allow-Deny decisions.

An application model is composed of the following objects:

- Business objects – these are the protected objects
- Operations - business operations available on the Business Objects
- Users in Roles

Special runtime conditions (business conditions) must be evaluated before each authorization decision. These decisions are necessary to define which users can execute a given operation on an object, and under what conditions

A very important development criteria was that the authorization engine must be powerful enough to serve a very high load of authorization requests. I designed a special, stored result Boolean arithmetic to solve the problem, implemented as special ACLs in the TAM's authorization database, serving the authorization request through the provided authorization API.

The solution is able to evaluate request calls passing the grouped Boolean or Integer runtime conditions, make virtually any arithmetic operation with these conditions, and provide the Allow-Deny result.

We worked out a special management interface - as an add-on to the TAM - to allow defining the special rules. It processes the rules-input as human-readable regular logical expressions and uploads them as special ACLs to TAM.

The application itself contains a small plug-in that converts the runtime conditions received from the calling application into standard TAM requests and receives the formal authorization result. All authorization decisions are made with the native TAM throughput.

There was one TAM feature to consider: it is the 32 bit internal word -length. This is a bottleneck to make the solution fully generic on one hand, but on the other hand with the evaluation method detailed below the solution is very well usable in most environments.

I defined the following building elements to implement the condition evaluation engine:

a. Condition Group

Basic building block, implemented as a special TAM ACL type.
One *condition group* means five binary variable. Within one condition group any combination of the conditions can result in Allow decision.
Input range: 5 bit Boolean, or 0-31 Integer, or any combination.
Sample : [A] = (A and not B) or C or D and not E , or N=1,2,3, N<17 etc.
Output: Boolean, Yes/No
Number of calls: 1 call

b.Condition Chain

To one protected object several (max. 32) *condition groups* can be attached *in chain* ,and the result of the evaluation for all groups can be performed with a single system call.
The result is a logical AND for all the chained condition groups.
Sample: {ABC} = [A] and [B] and [C]
Output: Boolean, Yes / No
Number of calls: 1 call / chain

c. Condition Relation

Relations are freely definable based on the result of five formerly evaluated Condition Chains. This is very flexible, all exeption like condition can be accomplished by this module.
Sample CR= {A} and {B) or ({C} and {D}))and not {E}, etc.
Number of calls: 6 calls

A. Model Based Security

We set up a working version of the environment – including a simple test application sending authorization requests. The framework itself is general enough to provide a solution to implement different security models. We plan that having this framework we can start to implement the model based verification according to a given security model. As the first step we are going to start with the Wilson-Clark authorization model.

B. Authorization service for Service Oriented Architecture

The authorization engine layer is easily accessible from any application requiring authorization decisions , therefore the solution provides a framework to externalize the definition and evaluation of all authorization requests needed. As a further step the definition of this service will be done, and the solution will be tested in live environment.

References

[1] John Ganci, Hinrich Boog ,Melanie Fletcher,Brett Gordon : *Develop and Deploy a Secure Portal Solution*, IBM Redbook 2004

[2] David Clark, David Wilson:*A Comparison of Commercial and Military Computer Security Policies*, Proceedings of the IEEE Computer Society Symposium on Security and Privacy, Oakland, 1987.

PROJECT RISK MANAGEMENT BASED ON BAYES BELIEF NET USING EMF

Ákos SZŐKE
Advisor: András PATARICZA

I. Introduction

Although software metrics is nearly 30 years old it could not become a silver bullet to project management (PM). The key reason is that metrics does not provide information to the main objective of PM: decision support during software development [1]. Some exhaustive investigation [2,3,4] concluded that Bayes Belief Nets (BBNs) provide far the best solution to this managerial problem. This paper describes an extensible and powerful model based risk management tool which intends to help governing the development of today's complex software intensive systems.

The present investigation is related to a GVOP tender (GVOP-3.3.3.-05).

II. Quantitative Risk Management Based on Bayes Belief Net

In everyday project management, decisions are based on mostly subjective information, which is usually uncertain and incomplete. Adapting agile development methodologies such as Extreme Programming (XP) require even more precise and robust decision-making, which implies support for risk management in an easy way. In mature risk management, the decisions should be based on quantitative information which can be gained through the widespread CMU SEI PM level software metrics [5]. Unfortunately, metrics based approaches do not support decision-making; they just provide some useful data for risk analysis.

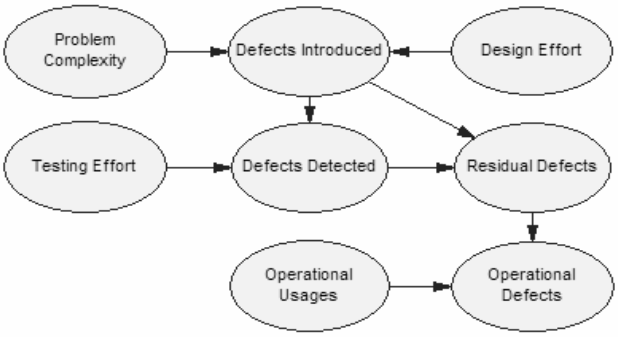


Figure 1: A defect model expressed by BBN

BBNs have proven to be an extremely powerful technique for reasoning under uncertainty. A BBN is a directed acyclic graph (DAC), where nodes of a BBN represent uncertain variables and the arcs are the causal links among them. Between each node, a set of conditional probability functions (CPF) are defined to model the uncertain relationships.

A defect model [3] expressed by BBN can be seen in Figure 1. Like any BBN, this model contains a mixture of variables where some values are known, and others are interested. The power of BBN is that it will compute the probability of every variable irrespectively of the amount of known variables to support decision-making.

III. Modeling BBNs with EMF

EMF (Eclipse Modeling Framework), a core part of the Eclipse IDE, is an implementation of OMG's MOF [6], and besides it is a Java framework and code generation facility for building tools based on structured data models. These models are simply a set of related classes used to handle the data in an application.

One of the main components of EMF is ECore framework which is responsible for basic model generation. Although, ECore is a model, it is a metamodel, or meta-metamodel depending on how it

is used. In this presented approach, BBN models are defined with ECore in multiple abstracted way: ECore is a meta-metamodel (M3 model) because it is used to define BBN metamodel (M2 model), and this metamodel defines how BBN models (M1 model) can be created.

IV. Overview of the Model-driven BBN-based Decision Support

The proposed solution consists of an Eclipse plug-in for BBN (meta)modeling and a popular BBN engine (Hugin Decision Engine 6.6) [8] for probability computations and BBN updates.

A BBN model describes managerial risk management problem to be supported. In the solution architecture its position can be seen in Figure 2. Hence the BBN engine can import BBNs in its own structured language and the BBN model is in XMI (XML Metadata Interchange) format, an XMI to BBN engine language transformation is needed. This is realized by a very powerful tool for generating source code: JET (Java Emitter Template).

In order to create and edit a BBN model, a BBN metamodel should be composed with EMF. This metamodel (*.ecore) consists of a package, a few data types, enumerations, and classes, which are in fact instances of model elements in ECore.

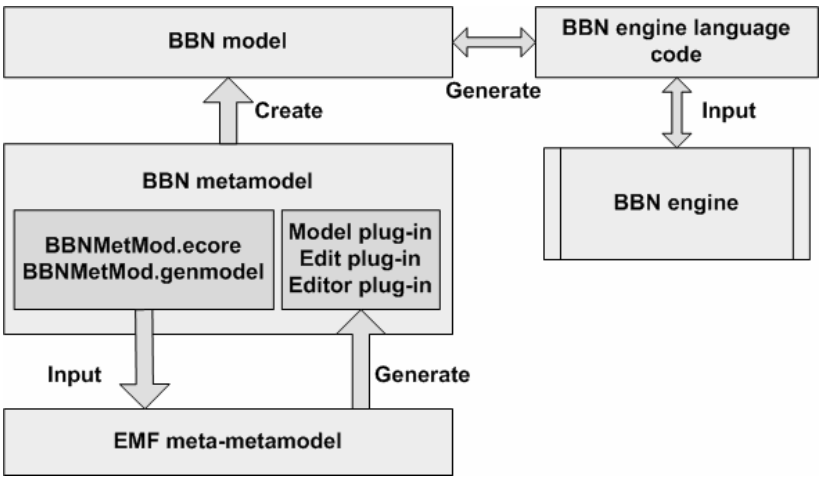


Figure 2: Solution architecture

From this metamodel, with the ECore's EMF.edit framework a generator model (*.genmodel) generates the desired Eclipse editor plug-in, as an Eclipse platform standard extension, for BBN creating and editing.

This integrated solution provides a flexible and extensible solution for decision-making.

V. Conclusion

The commonly used regression models may lead to inappropriate risk management decisions. The presented solution provides an extensible and powerful predictive model, where metrics are incorporated into cause-effect relationships (expressed by BBNs), to provide accurate predictions.

Our next objectives are to facilitate automatic BBN node set up with information gained through PM level software metrics and extend functionality with graphical editing capability using Eclipse's Graphical Editing Framework (GEF).

References

[1] V.R. Basili, H.D. Rombach, *Towards Improvement-oriented Software Environments*, IEEE TSE, p758-773, 1988

[2] N. E. Fenton, M. Neil, *A Critique of Software Defect Prediction Models*, IEEE Transactions on SE, p675-689, 1999

[3] N. E. Fenton et al, *Making Resource Decisions for Software Projects*, Proc. 26th ICSE, IEEE 2004

[4] C-F. Fan, Y-C Yu, *BBN-based Software Project Risk Management*, The Journal of Systems and Software, 2004

[5] Carnegie Melon University Software Engineering Institute, *Software Metrics Guide*, URL: http://sunset.usc.edu

[6] T. Aven, *Foundations of Risk Analysis: A Knowledge and Decision-Oriented Perspective*, John Wiley & Sons, 2003

[7] C. G. Wu, *Modeling Rule-Based Systems with EMF*, Eclipse Corner Article 2004

[8] B. Moore et al, *Eclipse Development using the GEF and the EMF*, IBM Redbooks 2004

[9] Hugin Expert A/S, *HUGIN API Reference Manual*, URL: http://www.hugin.com