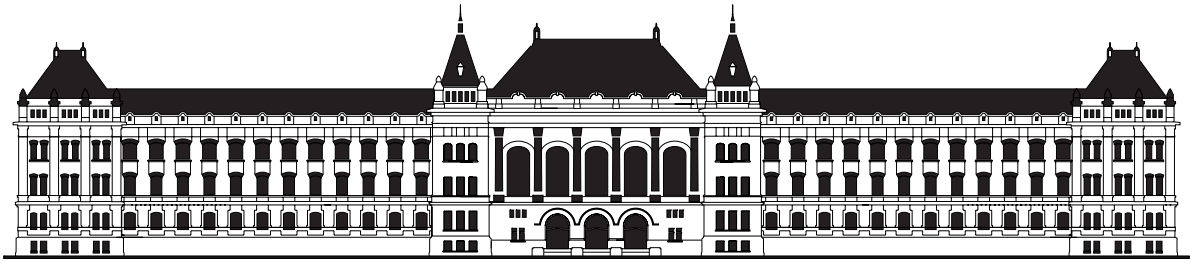
PROCEEDINGS
OF THE
14TH PHD MINI-SYMPOSIUM

FEBRUARY 5–6, 2007.
BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
BUILDING I



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:
Márta ALTRICHTER
Péter BOKOR
András FÖRHÉCZ
Gábor HULLÁM
Zoltán MICSKEI
Tamás NEPUSZ

Homepage of the Conference:
http://www.mit.bme.hu/events/minisy2007/

Sponsored by:
IEEE Hungary Section (technical sponsorship)
Schnell László Foundation
Albacomp zrt

ISBN 978-963-420-895-2

FOREWORD

This proceedings is a collection of the extended abstracts of the lectures of the 14th 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 six main groups. These are Biomedical Measurement and Diagnostics; Information Mining and Knowledge Representation; Intelligent Systems; Fault-tolerant and Dependable Systems; Embedded Systems, Measurement and Signal Processing; Model-based Software Engineering. The lectures are at different levels: some of them present the very first results of a research, because most of the first year students have been working on their fields only for half a year, therefore they submitted two-page papers. The second and third year students are more experienced and have more results; therefore they have four-page papers in the proceedings.

During this fourteen-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, Belgium; Toshiba R&D Center, Kawasaki, Japan; University of Darmstadt, Germany; University of Coimbra, Portugal; LAAS-CNRS, France; Computer and Automation Research Institute of the Hungarian Academy of Sciences, Budapest; ThyssenKrupp Nothelfer Ltd. R+D Institute, Budapest, Hungary.

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 12, 2007.

Béla Pataki

Chairman of the PhD Mini-Symposium

LIST OF PARTICIPANTS

Participant	Advisor	Starting Year of PhD Course
ALTRICHTER, Márta	HORVÁTH, Gábor	2005
BALOGH, András	PATARICZA, András and VARRÓ, Dániel	2004
BÓDIS-SZOMORÚ, András	DABÓCZI, Tamás; SOUMELIDIS, Alexandros and FAZEKAS, Zoltán	2005
BOKOR, Péter	PATARICZA, András	2005
CSORDÁS, Péter	JOBBÁGY, Ákos	2004
DARABOS, Dániel	HORVÁTH, Gábor	2004
FÖRHÉCZ, András	STRAUSZ, György	2005
GADOS, Dániel	HORVÁTH, Gábor	2006
HAMAR, Gábor	HORVÁTH, Gábor; TARJÁN, Zsuzsanna and VIRÁG, Tibor	2005
HORVÁTH, Ákos	VARRÓ, Dániel	2006
HULLÁM, Gábor	ANTAL, Péter and STRAUSZ, György	2005
KISS, Gergely	FEHÉR, Béla	2006
KOCSIS, Imre	PATARICZA, András	2006
KOCSIS, Zsolt	PATARICZA, András	2004
KOVÁCS, Máté	MAJZIK, István; VARRÓ, Dániel and GÖNCZY, László	2006
LAZÁNYI, János	FEHÉR, Béla	2004
LUDÁNYI, Zoltán	HORVÁTH, Gábor	2005
MAKAI, Tamás	PATAKI, Béla	2006
MERSICH, András	JOBBÁGY, Ákos	2004
MICSKEI, Zoltán	MAJZIK, István	2005
MILLINGHOFFER, András	ANTAL, Péter and DOBROWIECKI, Tadeusz	2004
MOLNÁR, Károly	PÉCELI, Gábor	2004
NEPUSZ, Tamás	STRAUSZ, György and BAZSÓ, Fülöp	2005
OROSZ, György	PÉCELI, Gábor and SUJBERT, László	2006
PÁSZTOR, Péter	PATARICZA, András	2006
PILÁSZY, István	DOBROWIECKI, Tadeusz	2004
RÁKOSI, Bálint	PATARICZA, András	2006
RÁTH, István	VARRÓ, Dániel	2006
SÁRHEGYI, Attila	KOLLÁR, István	2004
SZÁNTÓ, Péter	FEHÉR, Béla	2004
TAKÁCS, Gábor	PATAKI, Béla	2004
TÓDOR, Balázs	HORVÁTH, Gábor	2004
TÓTH, Dániel	PATARICZA, András	2006
ZENTAI, András	DABÓCZI, Tamás	2004

Program of The MINI-SYMPOSIUM

Biomedical Measurement and Diagnostics

Information Mining and Knowledge Representation

Intelligent Systems

Fault-tolerant and Dependable Systems

Embedded Systems, Measurement and Signal Processing

Model-based Software Engineering

THE REFINEMENT OF MICROCALCIFICATION CLUSTER ASSESSMENT OF MAMMOGRAPHIC X-RAY IMAGES

Márta ALTRICHTER
Advisor: Gábor HORVÁTH

I. Introduction

Computer Aided Diagnosis of x-ray mammographic images has a long history, many algorithms exist to find the two most important pathological signs: the masses and the microcalcifications. In this article we will address microcalcifications, which appear as small, intense, white spots within the mammographic tissue and are serious signs of benign or malignous cancer only if they form a cluster.

The algorithms searching for microcalcification clusters on individual X-ray images in the literature [1] are common in a sense that they usually find the positive cases with large probability – sensitivity is about 90-95% – but the number of false positive hits per picture is too high – 1.5 FP/image, specificity 0-5%.

To improve the specificity/sensitivity ratio of a microcalcification searching algorithm [2] we propose a simple way to reassess already found clusters. The basic idea of the reassessment is that first we determine a preliminary confidence value of the cluster, then we deter from the normal way of only analyzing one image at once, and use out the additional information residing in CC and MLO view by Joint analysis.

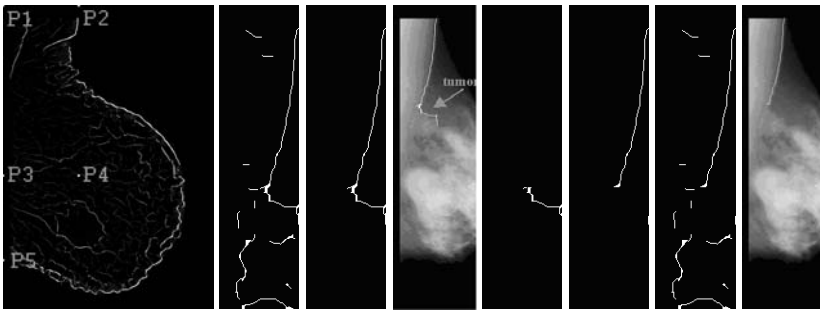
Joint analysis is important as the signs of a cluster have to appear on both views (MLO and CC). In practice a 3-D reconstruction of the breast would be the best, but the full 3-D reconstruction is impossible, as only two images are availagble, and those are made of the same but differently compressed breast. As a result instead of a full 3-D reconstruction we suggest a relatively simple procedure ("2.5-D" correspondence) to pair clusters of MLO and CC. Calcificated tissue and normal tissues are rather similar in texture, therefore their matching will only use the positioning system.

In this paper we focus on describing how the "2.5-D" positioning system can be built, and how it can be used to refine the assessment of a microcalcification cluster. We also show a method newly developed to calculate an initial confidence value based on more features of the cluster than initially used in the Joint analysis.

II. Building and Analysis of the "2.5-D" Positioning System

To match calcificated regions we have to build a simple model, that gives a suggestion on where the corresponding region can be at all on the other view. As the breast in CC view is exposed to x-ray from a different angle than in MLO, we can assume that a stripe will cover the possible corresponding regions on the MLO to a region taken from the CC view (this assumption works backward as well, thus an MLO region transfered to the CC view is a stripe) (the matching is illustrated at Fig. 2.). To find the corresponding stripe we use the positioning landmarks that segment the breast into its anatomical regions – the pectoral muscle, the nipple and the boundary of the breast – and three simple hypotheses:

1. The pectoral muscle's angle on a CC image is the vertical axis.
2. The position of the nipple can be estimated by laying a tangent on the breast border parallel with the pectoral muscle.
3. The distance covered from the nipple perpendicular to the pectoral muscle on MLO approximately corresponds to the distance measured up on the horizontal axis from the nipple on CC.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 1: (a) Region Of Interest (b) Edgeflow result, (c) Longest line selected, (d) Line on the image, (e) The bad part, (f) Removed suspicious segment, (g) New edge-map, (h) Final result

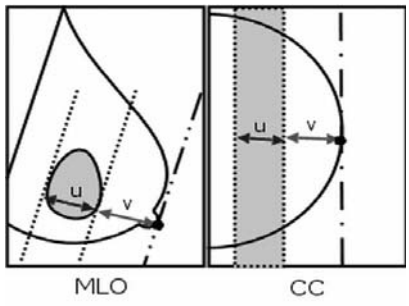


Figure 2: The corresponding stripe on the CC of a selected region on the MLO

The first step of the algorithm is to find the angle enclosed by the pectoral muscle and the horizontal axis on MLO views. To find the angle the slightly modified method in [3] was used where after the edge detection – special edge detection method: Edgeflow [4] – we cut up the line we got and removed segments which could not be part of the muscle line. As a result we did not find the complete pectoral muscle, but as we only searched for it's angle, the cutting and deletion of improbable line segments increased the robustness of the algorithm. (See Fig. 1.)

The nipple is marked out by the tangent parallel to the pectoral muscle laid on the breast border. To find the corresponding stripe we first measure the distances of the observed region from the tangent – u and v on Fig. 2. The same distances are measured up on the perpendicular line to the tangent from the nipple of the other view. The two points and the angle of the pectoral muscle mark out the stripe where the corresponding region can occur.

The success of the Joint analysis depends on how good estimation the three hypotheses provide. To test its effectiveness a statistical analysis was made. 1159 cases with 400μ/pixel resolution from the DDSM database [5] were selected, so that these cases contained only one pathological growth on each views according to the radiologists' assessments. As a result we can assume that the two marks on the two views are the 2-D projections of the same object.

The pixel corresponding to the centroid of the growth on the MLO was determined, and the deviation of the result from the centroid of the growth on the CC was measured in pixels. Fig. 3. shows the histogram of the deviations. There is some variance caused by wrong pectoral muscle finding, wrong radiologist assessment or the flaw of the hypotheses (because of breast deformation) for a few cases, but generally the hypotheses look to be standing (mean is around zero). To compensate the effect of variance the width of the stripe is increased by 10%.

A simple sensitivity test was constructed to see how much the system is sensitive to errors in finding the pectoral muscle angle and the nipple position. 90 versions of the above mentioned statistical analysis was made by slightly changing the originally found pectoral muscle angle from -50 to +40 in each system. This influenced the calculated nipple position as well. The Fig. 4. shows the percentage of cases under a given error in centroid matching (27, 52, 72 pixels – which were the average-variance, average, average+variance cluster sizes found by the microcalcification searching algorithm). The figure shows two important facts: (a) there is no inbuilt constant error in the muscle-angle finding algorithm, (b) there is a big tolerance (around -10 to 10 degree) in finding the pectoral muscle angle.

III. Microcalcification cluster matching and reassessment

Initially we have suspicious calcification clusters on both views. Depending on the calcification number and mean intensity difference of calcifications and surrounding tissue we assign an initial

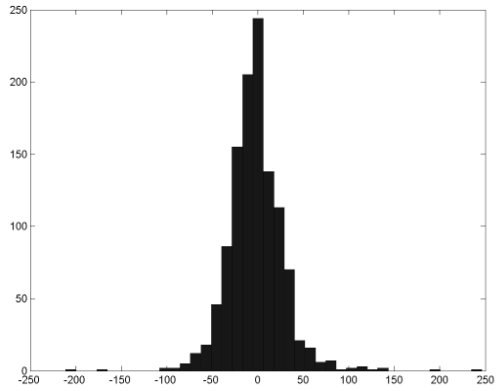


Figure 3: Histogram of pixel errors, number of cases 1159

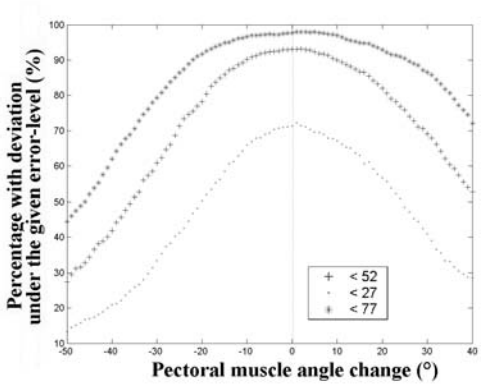


Figure 4: The percentage of the 1159 cases which are below the constant (27,52 or 77) plotted along the error in the pectoral muscle angle imbued by the experiment

credibility value to each cluster: P_{calc} (Range: 0-255, 255 - highly suspicious region, P_{calc} (150 is considered to be a calcification).

The credibility value is modified according to the area ratio of the stripe corresponding to it and of other calcification clusters found on the other view: A_{ratio}. The new credibility is: $\hat{P}_{calc} = P_{calc} - cons_1 * (1 - A_{ratio} + cons_2)$. The probability is decreased by subtracting an amount instead of simply multiplying with A_{ratio} as sometimes we only find a TP hit on only one view. If that cluster is highly suspicious it won't be removed by the final thresholding even if no pair is found. The constant $cons_2$ is to ensure, that at high correspondence the suspicion is raised.

The performance of the calcification matching was analyzed over 188 cases. 66 out of these cases contained malignant calcifications. Table 1. shows the results of the matching in a case level. Thus the 13.1% increase in specificity means that the algorithm cleared all the FP hits on the four images in those cases.

IV. Credibility calculation with MLP (Multilayered Perceptron)

The original credibility value depended on only the calcification number and intensity values. In this new credibility calculation method we tried to find a better estimate for P_{calc} value, including other features as well: area parameters (like the area of calc.), shape parameters (average length of the major axis of the calc. etc.), distance parameters (like the average distance between the calcifications) etc. Features were extracted from the calcification masks (see Fig. 6.).

We have acquired a 1390 true positive cluster set determined by the microcalcification algorithm. Each of these clusters overlap a TP mark of a radiologists. A false positive sample set was gathered from the clusters found that did not overlap radiologists' mark. 13 parameters (out of 35) of the clusters were used to teach a simple MLP with Levenberg-Marquart algorithm. Early stop was implemented

Table 1: Table of results for Joint analysis:

	Original microcalcification algorithm	Calcifications after matching reassessment	Percentage change
Sensitivity	95.5%	92.4%	-3.1%
Specificity	0.8%	13.9%	$+13.1\%$
FP/image	3.25 FP/image	1.57 FP/image	

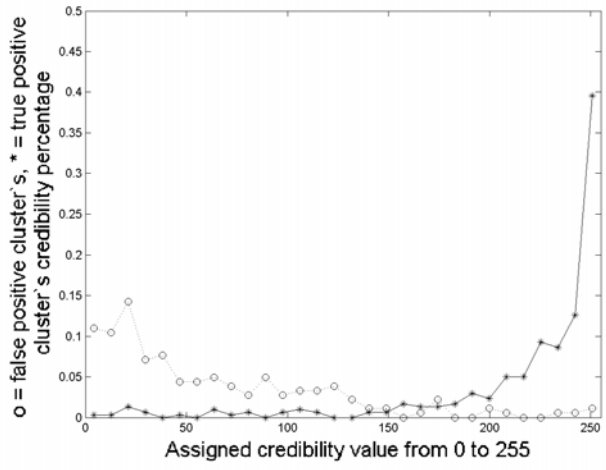
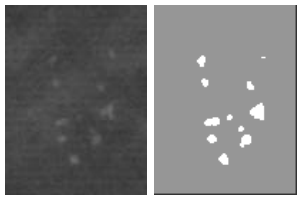


Figure 5: The percentages of true positive and false positive cluster's credibility values, circles are FPs, cross TPs



(a) (b)

Figure 6: Original cluster and calcification mask

with a testing set of 254-254 clusters, the learning set contained 834(TP)-834(FP) clusters. The remaining 302(TP)-284(FP) data were used for validation of the network.

A threshold (-1) was applied so that clusters which fall below are automatically dropped as false positive hits. 1 TP and 102 FP clusters were dropped by the network because of the treshold. The credibility values for the remaining clusters were determined according to the distance they resulted from the value 1 (surely TP). Fig. 5. shows the histogram of TP and FP percentage with the given P_{calc} value. We can see, that TP and FP initial P_{calc} values are promising.

V. Conclusions

Joint analysis is a simple way of combining the results of microcalcification detection algorithms applied for individual X-ray breast images (CC and MLO). It follows the procedure of skilled radiologists: if a suspicious area can be found in one view, usually its corresponding pair should be detected in the other view of the same breast. According to our first results this approach reduces the false positive detection significantly while the decrease of true positive hits is relatively small. Moreover a more sophisticated method to calculate the credibility was examined to improve primal P_{calc} value.

The proposed system is still tested. The improvement of the primal calcification searching algorithm [2], and the further analysis of the pairing (FROC curve) is needed. Also we intend to test the merging of the new credibility value assignment with the pairing, to see how the results can further change the TP/FP ratio.

References

[1] S. Yu and L. Guan, "A CAD system for the automatic detection of clustered microcalcifications in digitized mammogram films," *IEEE Trans. on Medical Imaging*, 19(2):115–126, Feb. 2000.

[2] G. Horvath, B. Pataki, A. Horváth, G. Takacs, and G. Balogh, "Detection of microcalcification clusters in screening mammography," in *Proceedings of the 3rd European Medical & Biological Engineering Conference (EMBEC'05)*, Prague, Czech Republic, Nov. 20–25 2005.

[3] R. Ferrari, R. M. Rangayyan, J. E. L. Desautels, R. A. Borges, and A. F. Frére, "Automatic identification of the pectoral muscle in mammograms," *IEEE Trans. on Image Processing*, 23(2):232–245, Feb. 2004.

[4] W.-Y. Ma and B. S. Manjunath, "EdgeFlow: A technique for boundary detection and image segmentation," *IEEE Trans. on Image Processing*, 9(8):1375–1388, Aug. 2000.

[5] M. Heath, K. Bowyer, D. Kopans, R. Moorea, K. Chang, S. Munishkumaran, and P. Kegelmeyer, "Current status of the digital database for screening mammography," in *Proc. of the 4th International Workshop on Digital Mammography*, pp. 457–460, Nijmegen, The Netherlands, June 7–10 1998.

AN IMPROVED OSCILLOMETRIC METHOD

Péter CSORDÁS
Advisor: Ákos JOBBÁGY

I. Introduction

The home health monitoring device (HHMD) built at our department stores some physiological signals, like ECG and photoplethysmographic (PPG) signals of the patient. The records can be used for later medical analysis and research. The currently running research is focused on accurate non-invasive blood pressure measurement and on characterization of blood vessel properties. In this paper, the second area is outlined, while the first one is discussed in detail. After clarifying the main concept of the wide-spread oscillometric blood pressure measurement method (OBPM), the efficiency of the recently developed technique is demonstrated through the evaluation of a HHMD record.

II. Materials and Methods

To perform a standard measurement with the HHMD, the patient only has to fix a cuff on the left upper arm, press a button and put the hands on ergonomic designed platform. The Einthoven I. lead ECG signal is captured by means of the electrodes under the palms. (Optionally, the Einthoven II. lead ECG can be recorded using an extra electrode applied on the patients left leg). The index fingers rest upon PPG sensors. The pulsation in the PPG signal gives information about the volume changes of the vessels. For the left hand (which is occluded by the cuff during the measurement), two PPG signals are recorded, where the wavelength of the excitation light is different. Since the two signals are calibrated, the measurement of blood oxygen saturation is possible. The DC components of the left PPGs are also recorded. These deliver information about the changes of the *total* blood volume in the capillary vessels (arterioles and venules) of the fingertip. Naturally, the cuff pressure (CP) is also recorded. All these signals are sampled with 1 kHz.

The measurement process is fully automated. In the first 25 seconds, the ECG and PPG signals are recorded with deflated cuff. This makes the investigation of the physiological signals possible, while the patient is in rest. Optionally, it can be checked, whether the cardiovascular system is in steady-state, or some transient (caused e.g. by physical stress) is decaying [1].

After the rest phase, begins the inflation of the cuff. In case of the OBPM, the better accuracy can be achieved, the slower is the CP change. The applied approximately 6 mmHg/sec. is a compromise, which can be accepted by the users. The maximal CP depends on the patient's blood pressure, which is measured on-line. A maximal upper limit can be configured. Although the slow deflation phase is not strictly required by our algorithms, it is kept as a source of redundancy. After deflation, another 20 seconds of rest-state is recorded.

Currently, only a basic oscillometric algorithm is implemented. The results (calculated systolic and diastolic blood pressure) are indicated (spoken and displayed) at the end of measurement. Regarding the research, it is more important, that all of the recorded signals are stored on an MMC card for post-processing.

By means of a PC application, the records can be 'replayed'. This serves for testing and improving the algorithms, implemented in the HHMD. The thorough analysis of the signals and the development of further algorithms are aided by a user-friendly, easily extendable framework, written in MATLAB.

For the validation of the implemented oscillometric method, we have made measurements with a commercial oscillometric blood pressure meter (OMRON MX3) subsequent to the application of the HHMD. In a few cases, parallel measurements with a tonometer (COLIN CBM 7000) have been taken.

III. The Oscillometric Blood Pressure Measurement

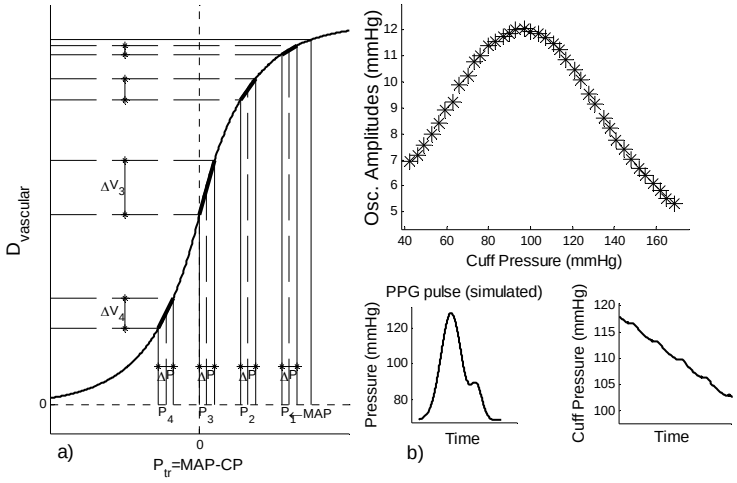


Figure 1: The principle of the OBPM

The principle of the OBPM is illustrated in Figure 1. The model of the vascular wall is simplified to the highest degree. Fig. 1/a shows the diameter changes of the brachial artery as a function of transmural pressure. It can be seen, that the volume pulsation caused by pressure waves ($C=\Delta V/\Delta P$ - Compliance) is maximal, if the external pressure, i.e. CP equals to mean arterial pressure (MAP). Thus, the pressure oscillation in the cuff (oscillometric amplitude - OA) is also maximal, if the volume-pressure characteristic of the cuff is assumed to be linear. This implies that the MAP can be detected as the maximum of OA values (Fig. 1/b.). However, systolic and diastolic (SYS, DIA) pressure can only be calculated. In most cases, they are assigned to CP values, where OA reaches $k_{sys}\cdot OA(MAP)$ and $k_{dia}\cdot OA(MAP)$, respectively. The parameters are determined with statistical methods. They are not well defined: most of the device manufacturers are using different values [3]. Some special properties of the vessel walls (e.g. collapse tendency) can cause slight changes in OA, when CP is near to SYS or DIA. The so called derivative OBPM is based on this phenomenon [5].

The algorithm implemented in the HHMD is presented in Figure 2. The detection of the oscillometric pulses is demonstrated in the right upper corner of Fig. 2. Since the ECG signal is recorded, the QRS detection is a great help for the segmentation of the CP signal (dotted line). After the subtraction of the pressure trend (dashed line) for each heart cycle, the OAs are easy to calculate. After the MAP is detected, two straight lines are fitted (with LS fit) on the OA points to estimate SYS and DIA. These are assigned to the CP values, where the fitted lines pass the SYS and DIA threshold levels. The parameters, used for the threshold calculation: k_{sys}=0.7, k_{dia}=0.6. The measured and estimated pressure values for the presented record:
DIA / MAP / SYS: 86/118/133 (mmHg).

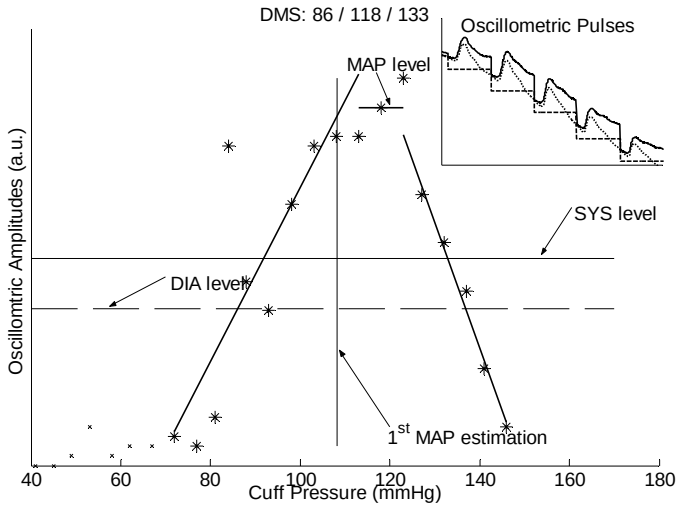


Figure 2: The implemented OBPM, running on a HHMD record.

IV. Improvement of the Method

The difference between the ideal (Fig. 1/b) and the real (Fig 2) CP-OA characteristics is conspicuous. This is only partly caused by measurement noise. The 'physiological noise' has an important role: the blood pressure is changing beat to beat (e.g. because of the respiration).

The techniques, presented below, have the following objectives:

- Eliminating the effect of respiration, getting coherent DIA, MAP and SYS values.
- Estimating the continuous blood pressure signal, calculating the blood pressure variance
- Validating and customizing the parameters k_{sys} and k_{dia} for the OBPM.

A. Examining the correlation of the PPG and CP signals

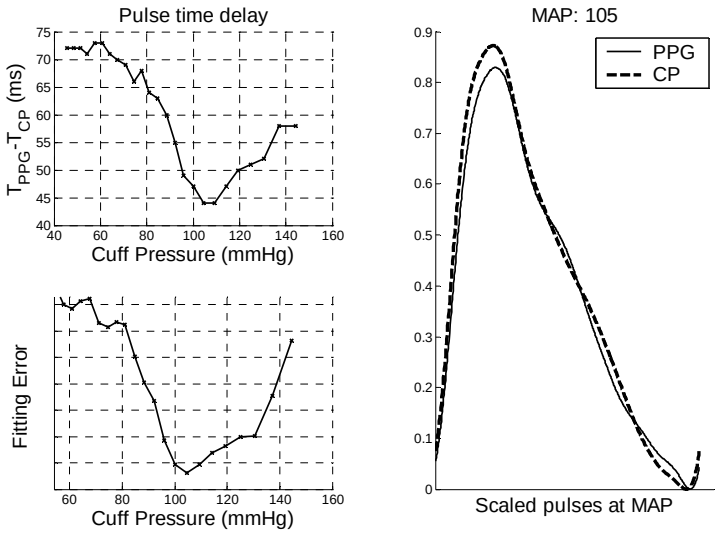


Figure 3: Correlation of CP and PPG signals

We assume, that the D-P (Fig. 1/a) function of the capillary vessels between systolic and diastolic pressure is nearly linear. If the pressure transfer between macro- and microcirculation is also considered to be linear, the PPG signal can be regarded as a rescaled pressure signal. Thus the effects, modulating blood pressure, are modulating the PPG as well. Besides, based on Fig 1/a, it is assumed, that regarding to linearity and augmentation, the blood pressure - cuff oscillation transfer is the better, the smaller is the transmural pressure. This means, that if CP=MAP, the D-P characteristic of the brachial artery is linear with maximal slope. Therefore, the pressure pulses are transferred to the cuff with maximal amplitude and with minimal shape distortion. The measured oscillation pulses and the PPG pulses, recorded from the fingertip of the free (not occluded) arm, are nearly identical, apart from a scaling factor. Practically, after an appropriate rescaling, the distance of the two signals can be calculated for each beat. The algorithm has the following steps:

- Detection of pulse upstrokes, beat to beat segmentation
- Getting the pulsatile components of the signals: $PPG^{\sim}$,$CP^{\sim}$
- Calculation of pulse amplitudes for each beat. ($PPG^{\wedge}(k)$, and $CP^{\wedge}(k)$ for beat k)
- Calculation of fitting error. N(k) stands for the length of beat k (in samples), $PPG^{\sim}(k,p)$ and $CP^{\sim}(k,p)$ for point p in the beat k, on the $PPG^{\sim}$ and $CP^{\sim}$ signals, respectively.

$$FE(k) = \sqrt{\frac{1}{N(k)} \cdot \sum_{p=1}^{N(k)} \left(\frac{CP^{\sim}(k,p)}{\max(CP^{\wedge})} - \frac{PPG^{\sim}(k,p)}{\max(PPG^{\wedge})} \right)^2}$$

The minimum of the error function indicates the MAP. Moreover, there are significant changes at DIA and SYS, coming up to our expectation. (If the CP is between SYS and DIA, complete pressure pulses are transferred to cuff pulses, therefore, FE is smaller than in other cases). The main advantage relative to standard oscillometric method is the compensation of the effects, modulating the blood pressure. The result of the described correlation check is illustrated in Figure 3. (DIA, MAP, SYS: 81,105,131 mmHg, respectively). The error function and the delay between the coherent

PPG and pressure pulses is plotted on the left side. The right side shows the normalized pulses at CP=MAP.

B. Continuous Blood Pressure Measurement

As described in section IV.A, the PPG signal can be considered as an uncalibrated pressure signal. Knowing at least two momentary pressure values (DIA, MAP or SYS), the PPG signal can be rescaled. For this, a reference PPG beat is needed, which is obtained as the average of a few subsequent beats, practically of those, which have been recorded, during the reference pressures have been measured. The averaging helps decreasing the effect of blood pressure modulation.

The result of a PPG rescaling is remarkable. While the calibration has been done by means of SYS and DIA values, the MAP calculated from the scaled PPG equals to the MAP obtained in IV.A. This means, that the assumption about linear pressure-PPG pulse transfer is admissible.

By means of the short term continuous blood pressure signal, the statistical parameters of the blood pressure can be investigated.

V. Results and Future Plans

Ten prototypes of HHMD have been assembled and they are ready for everyday application. Beyond the methods discussed above, there are several other techniques aiding accurate blood pressure measurement at home [1]. Using these together, the customization of the standard OBPM is possible. Moreover, extra information about the patient can be obtained. The statistical properties of the blood pressure, the validated parameters of the OBPM are not less important for the characterization of the vessels state. At this field, the investigation of the pulse wave shape can be a further step. Another aim is the non-invasive measurement of the compliance function for the brachial artery [5]. Since the HHMD is ready for use, measurements with participation of patients suffering from cardiovascular diseases can be done. Thanks to the cooperation with other researchers in the field, the new measurement techniques can be validated by means of a physical model of the human arterial system [2].

Acknowledgement

The research was supported by the GVOP-2004-05-0340/3.0 (Economic Competitiveness Operational Programme) project *Home Health Monitoring*.

References

[1] Á. Jobbágy, P. Csordás, A. Mersich, *Blood Pressure Measurement at Home*, Conf. Proc. World Congress of Medical Physics and Biomedical Engineering 2006. pp. 3319-3322.

[2] F. Molnár, S. Till, G. Halász, *Arterial blood flow and blood pressure measurement on a physical model of human arterial system*, IFMBE Proc. of 3rd EMBEC'05, 2005.

[3] A. Ball-llovera, R. Del Rey, R. Ruso, J. Ramos, O. Batista, I. Niubo, *An Experience in Implementing the Oscillometric Algorithm for the Non-Invasive Determination of Human Blood Pressure*, Proc. 25th Annual International Conference of the IEEE EMBS, 2003. pp. 3173-3175.

[4] E. O'Brien, *State of the market for devices for blood pressure measurement*, Blood Pressure Monitoring. 6(6):281-286, December 2001.

[5] Drzewiecki, G. *Noninvasive Arterial Blood Pressure and Mechanics.* The Biomedical Engineering Handbook, Second Edition. CRC Press LLC, 2000.

SEGMENTING THE LUNG FIELDS IN CHEST RADIOGRAPHS

Dániel GADOS
Advisor: Gábor HORVÁTH

I. Introduction

Chest radiography is the most frequently used method in medical examination and screening of the lungs. Although there are lots of other imaging techniques, chest radiography is relatively cheap and quite fast, so it will remain the most common procedure for the next several years.

The cancer statistics call attention to the necessity of early diagnosis, which can be made by regular screenings. To cope with the remarkable number of patients and the improper diagnoses made by doctors a CAD system is needed to reduce the work of radiologists.

Thus the computer program should discover the structures located on the images and understand their meanings (i.e. lung fields, the ribcage, the clavicle, and many others), it should contain image enhancement features (such as subtracting the ribcage or reducing the shadow of the heart), but first the region of interest (ROI) must be determined. In this case it means segmenting the lung fields or in other words finding their contours, which is much more difficult than it seams. First of all large interpersonal anatomical variations can be noticed in the pictures, also the tube voltage and the amount of inhaled air have remarkable effect on the images (for example the visibility of the bones mainly depends on the former while the latter has impact on the contrast of the lungs), moreover chest radiographs are projection pictures, so anatomical objects are superimposed [3].

II. Heuristics and rules for segmentation

The lung segmentation in our sense differs from the trends accounted in literature (where the area hidden by the heart is ignored), because the left border of the left lung is located beneath the heart (anatomically there is some part of the lung hidden by the heart, which cannot be observed in the X-Ray images without image processing). Thus in general the lung contours are in coincidence with the usual contours, but the inner border of the left lung is much fainter than in other places.

The heuristic which gives an acceptable approximation of the lung fields is organized as follows [2]:

- determination of some parameters of the picture and the lung (excluding the areas that do not belong to the body, finding the centers of the lungs, finding the backbone, which separates one lung from another, applying the so called Relief Model [2] to exclude even more areas)
- approximating the lung fields (searching for the points whose intensity is higher than all points that are closer to the picture border in a given direction – horizontal, vertical and at 45 degree angle are applied –, using flood-fill algorithm to find the remaining points around the lung centers)
- refining the results (finding the diaphragm, the mediastinum, the “lower corners of the lungs”, smoothing the obtained point-lists describing the lung contours)

Using these rules we can obtain an approximation of the lungs, which can be a foundation to advanced algorithms to refine the results. We use the so called ASM (Active Shape Model), which is more precise, but an initial approximation is needed to start with (it is pretty slow when the whole search space is taken into account).

III. Applying ASM

ASM is a simple iteratively searching approach to object detection combined with an object model [1]. Let $\mathbf{x}_i$ describe the i-th object example (right or left lung in this case, which is determined by humans to obtain a model), which contains n x-coordinates and n y-coordinates (i.e. $\mathbf{x}_i = (x_1^{(i)}, y_1^{(i)}, \ldots, x_n^{(i)}, y_n^{(i)})^T$). The covariance matrix of the examples is

$$\mathbf{S} = \frac{1}{s-1}\sum_{i=1}^{s}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \,, \tag{1}$$

where s is the number of examples, $\bar{\mathbf{x}}$ is the mean of examples. Some eigenvectors of $\mathbf{S}$ belonging to the largest eigenvalues form matrix $\mathbf{\Phi}$:

$$\mathbf{\Phi} = [\,\boldsymbol{\varphi}_1,\, \boldsymbol{\varphi}_2,\, \ldots,\, \boldsymbol{\varphi}_t\,]^T \,,\ t < n\,. \tag{2}$$

Thus the vector describing the lung might be approximated as

$$\hat{\mathbf{x}} = \mathbf{\Phi}^T\mathbf{\Phi}(\mathbf{x} - \bar{\mathbf{x}}) + \bar{\mathbf{x}}\,, \tag{3}$$

which is easier to evaluate than the original approximation rule, since no eigenvalues must be taken into account during the iterative search. So having n landmarks the ASM starts with an initial approximation, and then replaces the landmarks according to the fit function (using gradients on the line perpendicular to the lung-curve). These steps are iterated until a good solution is obtained.

IV. Results

An experimental run of the combination of the heuristic and the ASM is sown in Figure 1. The result is acceptable, but some refinement should be done in the future. In comparison to the heuristics described above, the results are somewhat better and more stable. Testing the results is in progress on a set of 125 images.

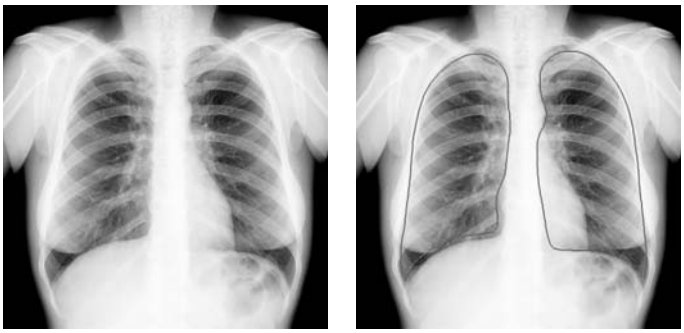


Figure 1: A normal X-Ray image (on the left) and the result of lung segmentation (on the right).

V. Conclusion

We have produced novel rules that are fast and provide a good approximation to the lung contours. The lung models assist the stability of the results, though the fit functions used in the ASM should be improved.

References

[1] T. F. Cootes, C. J. Taylor, "Statistical Models of Appearance for Computer Vision", Technical report, University of Manchester, Wolfson Image Analysis Unit, Imaging Science and Biomedical Engineering, Manchester, United Kingdom, 1999.

[2] D. Gados, "Mellkas röntgenfelvételek elemzése (tüdőárnyék körülhatárolása)", M.Sc. thesis, G. Horváth and Á. Horváth (advisors), Budapest University of Technology and Economics, 2006.

[3] B. van Ginneken, "Computer-Aided Diagnosis in Chest Radiography", Ph.D. thesis, University Medical Center Utrecht, Wageningen, 2001.

MARKOV CHAIN BASED EDGE DETECTION ALGORITHM FOR EVALUATION OF CAPILLARY MICROSCOPIC IMAGES

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

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 microscopic tests play an important role in the early identification of these diseases [1].

Today the main problem is that there is not a cheap, easily accessible instrument, which is capable not only of the image or video recording, but has a support for computer aided evaluation. This is very important because the exact and objective evaluation requires much more time than the examination itself, and this is why quantitative measures are rarely used.

The first problem of image evaluation is the detection of capillaries, this is essential for any further image processing steps. We have to achieve high hit rate with low number of false positive hits, despite of the low image quality. In our presentation we will introduce an edge-detection method which can solve this problem with relatively high performance.

II. Materials and Methods

A. Capillary microscopic images

The capillary microscopic pattern of a healthy patient is examined by many researchers, hence it is precisely defined by the medical literature. As one can see in Figure 1(a) the vessels are arranged into rows, they are regular hairpin shaped, with the same orientation. A capillary loop 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 and 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, they are called giant- or mega-capillaries according to their size. 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.

B. The method of image recording

There is a generally accepted method for capillary microscopic examinations [2], hence we have also followed this method. 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 45°.

C. Edge detection

As it can be seen of figure 1(a) the capillaries do not have sharp edges, the image is very blurred and noisy. These properties are mainly come from the recording method, because capillaries are observed

through the skin. Classical edge detectors such as Sobel, Laplace and Canny operators have given bad result. It was very hard to separate real edges from noise, and edge detectors found not only the borders but the whole area of the capillaries because this intensity changes continuously in perpendicular direction to the capillary. Considering this properties we decided to search not the border but the centre line. It can be located more precisely and more robustly, with the algorithms described later.

D. "Walking" algorithm

A exact definition can be given for the centre points with the second derivatives of the image. Let us consider the images as a 3D surface, where the intensity is the third dimension. Capillaries are valleys on this surface, a centre line is the bottom of a valley. Our task is to give definition for these bottom points. A point is a bottom point of a valley, if the curvature of the image is significantly different in two perpendicular directions, and the point is a local minimum along a line which is perpendicular to the direction of the valley. The direction of the valley can be calculated with the following method. Let $f(x, y)$ be the intensity of the image in the pixel at point (x, y), let $\mathbf{A}_{x_0,y_0}$ be the matrix of the second derivatives at the (x_0, y_0) point:

$$\mathbf{A}_{x_0,y_0} = \begin{bmatrix} \frac{\partial^2}{\partial x^2} f\Big|_{(x_0,y_0)} & \frac{\partial^2}{\partial x \partial y} f\Big|_{(x_0,y_0)} \\ \frac{\partial^2}{\partial y \partial x} f\Big|_{(x_0,y_0)} & \frac{\partial^2}{\partial y^2} f\Big|_{(x_0,y_0)} \end{bmatrix}. \tag{1}$$

The eigenvector, which belongs to the smaller eigenvalue, points to the direction of the valley. The knowledge of the direction of the vessels can be used to create an iterative algorithm, which walks along the vessel from a given point. During the walk the algorithm makes little steps to the direction of the valley calculated from a little surrounding of the current point. The pseudo code of NEXT_POINT$(f, x, y, \mathbf{v_0})$, which calculates a step of the walk, is the following:

NEXT_POINT$(f, x, y, \mathbf{v_0})$
1. Calculating $\mathbf{A}$ in point (x, y)
2. $\mathbf{v} \leftarrow$ the normalised eigenvector of $\mathbf{A}$ belonging to the smaller eigenvalue
3. **if** $\mathbf{v}_0^\mathsf{T} \cdot \mathbf{v} < 0$ **then** $\mathbf{v} \leftarrow -\mathbf{v}$
4. $\mathbf{v}_0 \leftarrow \mathbf{v}$
5. $\mathbf{p} \leftarrow (x, y)^\mathsf{T} + \mathbf{v} \cdot const$
6. $l \leftarrow$ section trough $\mathbf{p}$, which is perpendicular to $\mathbf{v}$
7. $\mathbf{q} \leftarrow$ minimum point of $f(x, y)$ from the points of l
8. **return** $\mathbf{q}$

Lines 1 – 2 calculate the direction of the valley according to the second derivatives, as described above. There are two possible ways in a valley where we can move, therefore we must choose from them (lines 3 – 4). For this decision we need the direction of the previous step ($\mathbf{v_0}$). To avoid cycles, the direction is chosen, so that the angle of two subsequent steps is less than $90°$. The performance of the algorithm can be improved if we search the local minimum of the section which is perpendicular to the estimated direction of the valley, because this technique decreases the probability of going out of the valley (lines 7 – 8).

During a walk the algorithm performs steps one after the other. The end point of a step is the start point of the next one:

WALKING_ALGORITHM(f)
1. $\mathbf{s} \leftarrow$ START_POINT$()$
2. $\mathbf{v_0} \leftarrow 0$
3. $w \leftarrow (\mathbf{s})$

4. **repeat**
5. $\quad \mathbf{s_1} \leftarrow$ NEXT_POINT$(f, \mathbf{s}, \mathbf{v_0})$
6. $\quad \mathbf{v_0} \leftarrow \mathbf{s_1} - \mathbf{s}$
7. $\quad \mathbf{s} \leftarrow \mathbf{s_0}$
8. $\quad w \leftarrow \mathbf{s}$ appended to w
9. **until** STOP_CONDITION()
10. **return** w

Lines 1 – 2 perform the initialisation. The $\mathbf{d_0}$ vector is the direction of the previous step. In the repeat –until iteration (lines 4 – 9) the NEXT_POINT procedure calculates the point of the walk, and in line 8 each calculated point is added to the walk.

The performance of this method highly depends on the two procedures: START_POINT and STOP_CONDITION. The implementation of these procedures can contain complex conditions and must be based on the properties of the used images. The main problem of this approach is that it is very difficult to implement these procedures so that they give good results on high variety of images.

E. Random walk

Both of the two procedures can be got around by using a probabilistic approach. Instead of starting the walk form one ore more specified positions, we can start it from a random position. Instead of follow each walk separately we can calculate the probability of staying in a certain position after the 1^{th}, 2^{nd} ... M^{th} step. This is similar to the approach, used by the early Google for web page ranking [3]. In the initial state we assume that we can be in any pixel position with the same probability. During an iteration of the algorithm we modify this distribution according to the walk. If $i_1, i_2, \ldots, i_n$ are the pixels from where we can step to j with the walking algorithm, than the probability of staying in pixel position j in the $(k+1)^{\text{th}}$ iteration is:

$$P_{k+1}[j] = P_k[i_1] \cdot p_{i_1,j} + P_k[i_2] \cdot p_{i_2,j} + \cdots + P_k[i_n] \cdot p_{i_n,j}, \tag{2}$$

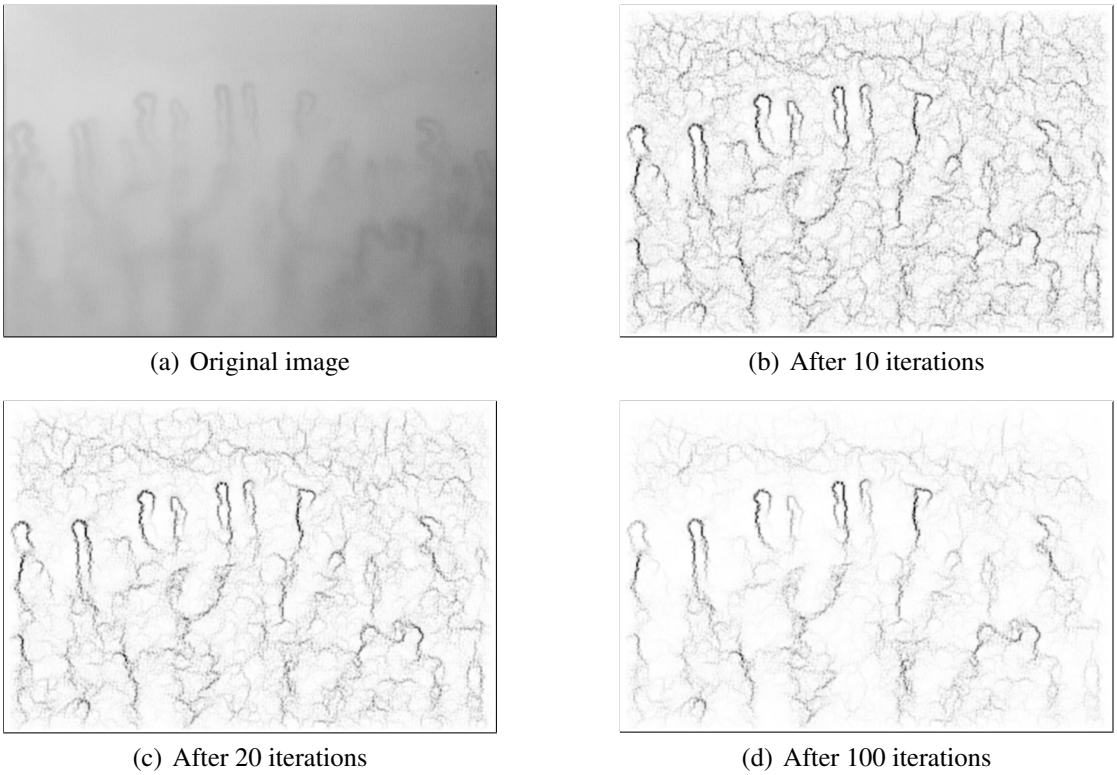
where $p_{i,j}$ is the probability of stepping from i to j if we are in pixel position i. Our assumption is that if we come into a point of a valley during the walk, we stay in the valley with high probability. If it is outside the valley the direction of the step is nearly random, so the probability of staying in these points is uniform, therefore relatively low. After some iterations the probability of staying in a pixel which belongs to a valley becomes significantly higher than in other pixels.

This problem can be formalised as a finite state homogeneous Markov chain. Let us label the pixel positions with positive numbers form 1 to N. Let X_k be a random variable, where $X_k = i$ means that the walk is in the pixel position $i \in \{1, 2, \ldots, N\}$ after the k^{th} step. Let $\mathcal{X}$ be the stochastic process: $\mathcal{X} = (X_0, X_1, \ldots, X_M)$, and let $p_{i,j} = P(X_{k+1} = j | X_k = i)$ that is the probability of moving to the pixel position j if we are in pixel position i. If this probability is independent from k for every i and j, then $\mathcal{X}$ is a homogeneous Markov chain. If $\mathbf{P}_k = (P(X_k = 1), P(X_k = 2), \ldots P(X_k = N))$, is the probability distribution of X_k, and $\mathbf{T} = [p_{i,j}]$ is the state transition matrix of the Markov chain, then $\mathbf{P}_{k+1}$ can be calculated with a vector – matrix multiplication:

$$\mathbf{P}_{k+1} = \mathbf{P}_k \mathbf{T}. \tag{3}$$

$\mathbf{P}_0$ is the uniform distribution, so that $P_0[i] = \frac{1}{N}$ for every i. The matrix $\mathbf{T}$ can be derived from the NEXT_POINT procedure. From pixel position (x, y) we can move to two possible destinations according to the procedure. In our algorithm we allowed to move not only to these two points but, to a little neighbourhood of them. The probability to move two points of this neighbourhood is calculated according to a 2D Gaussian distribution. Each probability is written to a position in the matrix. The value of $t_{i,j}$ is the probability of moving from point i to point j, when we are in point i.

III. Experimental results



(a) Original image (b) After 10 iterations

(c) After 20 iterations (d) After 100 iterations

Figure 1: Result of the edge detection

The resulting probability distribution can be viewed as a grey-scale image, after scaling pixel values to the 0 – 255 interval. Figure 1 shows the result on a typical capillary microscopic image. We have run the procedure for 10, 20 and 100 iterations. As it can be seen with increment of the number of iterations the noise becomes less, but the vague capillaries also begin to disappear. It is quite difficult to precisely measure the efficiency of the algorithm, because there is no exact reference with which it could be compared, we can only use the opinion of a human observer. We have applied this algorithm on 30 images, with variable quality. After a post-processing step (edge detection, noise filtering and edge connection) the results are compared with the human's opinion. The vessels detected by the algorithm was 91% of the vessels which are seen in the image.

IV. Conclusion

We have introduced a novel method of edge detection, which uses a different edge definition than traditional edge detectors. A definition based not directly on local properties of the image, but utilises the relation between pixel positions, which can be calculated using local properties. We have tested our solution on a special image set, but our method can be easily modified to detect other types of edges. Generalising and testing the modified algorithm remains a future research problem.

References

[1] A. Bollinger and B. Fagrell, Eds., *Clinical Capillaroscopy*, Hogrefe & Huber Publishers, 1990.

[2] Zs. Tarján, É. Koó, P. Tóth, and I. Ujfalussy, "Capillary microscopic examinations (kapillármikroszkópos vizsgálatok)," *Magyar Reumatológia*, 42:207 – 211, 2001, (In Hungarian).

[3] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Tech. Rep., Stanford University, 1999.

JOINT ANALYSIS OF DIFFERENT VIEWS OF MAMMOGRAPHIC IMAGES

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.
According to the importance of the topic numerous researches are going on about the topic including development of computer aided diagnosis (CAD) systems – a special part of which I have started investigating during writing my master degree.
This article is going to follow the order of this research: first I started and investigating the importance of joint analysis, then cooperating with Márta Altrichter we created a reference system to pair areas on different views of mammographic images, later on we realized that microcalcifications and masses need separate methods for joint analysis due to their different characteristics, and in my latest publication I have been focusing on improvement of results by giving stressed attention to spiculated massed that could be paired with less efficiency. During my work I also tried to keep algorithms usable by shortening running times.

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) – note that since X-ray images are inverted, high intensity spots are shades in fact. 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 false positive rate can be reduced if – in addition to the individual analysis – joint analysis of the images is done similarly to the way done by radiologist experts. Finding a pair to a mass- or microcalcification-candidate on the other view can increase the probability that the hit is a true positive one, therefore it should be kept. On the other hand – if no pair is found, the candidate can be regarded as a false positive one.
(We assumed that this neglection should be done considering breast density. Up to this time the latter detailed algorithm works with similar performance for all groups of densities.)

III Basics of Joint Analysis

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 nearly equal".
We used EdgeFlow [5] for finding the pectoral muscle since this edge detecting algorithm is also used in further steps, but according to other authors lots of other techniques can also be used, for example Hugh transformation.

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. According to our experiments a margin of 30 pixels have proven to be the best choice with an offset of 10 pixels. (We used images with 400 micron resolution, since on this resolution all masses are visible but running times are still tolerable.)

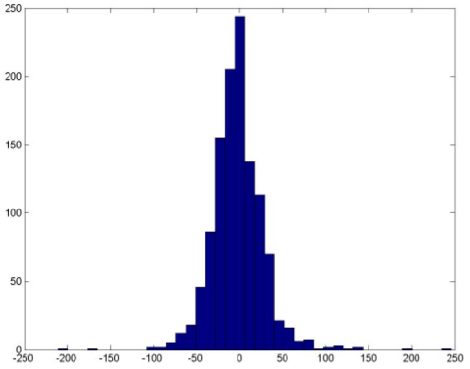


Figure 1. Histogram of error
(horizontal axis: error of hypothesis in pixels
vertcal axis: number of occurences in test set)

IV Joint Analysis for Masses

Since masses have characteristic texture, the reference system can be improved by textural analysis. (Some parts of the breast within the stripe can obviously not be the pair – based on texture features – of the examined mass candidate.) This is done through the following steps (see Fig. 2 also):

1. First the image is segmented by EdgeFlow [5], and then texture features (based on intensity, co-occurrence and gray-level-differences) are calculated for each segment, k-means clustering is applied to these features resulting a better segmentation.
2. Once we have a good quality segmentation, we recalculate features for the new segments.
3. After these preliminary steps we establish the reference system.
4. Then for each mass-candidate (b) we do pairing by computing the corresponding stripe (e) and – based on texture features – we choose the most similar segments within the stripe (f,g&h). (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.)
5. For the given mass candidate we check if there are pairs colliding with candidates on the other image. This pairing is done in both ways. Only those mass-candidates are kept which have a pair found. (Since (h) confirms candidate on (d), we keep it.)

With this algorithm we could achieve that 92% of true positive hits are kept while 23% of false positives are neglected.

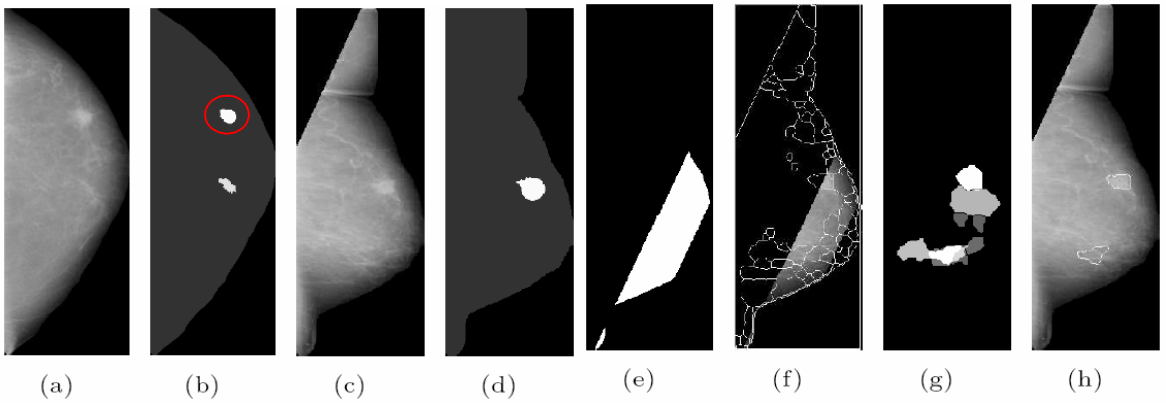


Figure 2. (a&c) CC & MLO image, (b&d) mass candidates on CC & MLO (marked candidate examined), (e) corresponding stripe to marked candidate, (f) segmentation (highlighted within stripe), (g) segments on MLO (intensity proportional to textural similarity), (h) possible pairs

V Focusing on Spiculated Masses

We have found that some spiculated masses cannot be well distinguished from normal tissue, since most of our previously used texture features are based on some derivatives of intensity and these masses have no high intensity core but only a stellar form causing our algorithm to draw false conclusions. Therefore we investigated the related references for clever techniques or distinctive texture features. After some unsuccessful trials we have found that ALOE invented by Kegelmayer et al. [6] can be used for our aims.

ALOE (Analysis of Local Oriented Edges) is based on the following facts. Spiculated masses have a stellar form, while normal breast tissue usually has mostly parallel linear structures (at the scale of usual segments) aiming at one direction. Therefore a histogram of pixelwise edge orientations having numerous local maxima is characteristic for spiculated segments, and a histogram having some local maximums should belong to normal segments. This difference can be well characterized by the variance of this histogram.

ALOE can be parametrized along edge detection: one can use any kind of edge detecting algorithm to gain pixelwise local orientation. References mention two groups of filters: classical edge detecting algorithms (filters of Prewitt, Roberts, Laplace, etc.) and Gaussian derivatives. We have also tried these methods and naturally investigated the usability of some results of EdgeFlow.

Both in our early experiments on cut image segments, and latter detailed tests when we built this feature in the pairing algorithm proved that the best choice is the EdgeFlow based ALOE.

This is due to the fact that classical edge detecting algorithms result mainly in noise for “flat” images segments (where linear structures are further than the scope of the differential filter). Therefore the histogram of local orientations is also flat, just like it should be for spiculated masses. All in all – the main problem with these filters is that the number of pixels having meaningful information about local orientation is low.

However – phase image produced by EdgeFlow has meaningful information even far from edges and also more reliable due to its noise reduction techniques. (Phase detection is based on an error-prediction scheme, and its clever edge-energy vector propagation also has noise reducing effects.)

Comparing results of this modified pairing (including ALOE as texture feature) to our previous results we have found that false positive neglecting rose from 23% to 31% while still keeping 92% of true positive mass candidates. For the sake of comparison: the gain is merely 1-2% with classical edge detecting algorithms; and by only an extra 20% running time in pairing 5% with Gaussian derivatives. This extra running time came from the fact that using pure Gaussians results in 5% gain only if we use 32 orientations instead of 8 used for EdgeFlow. This should be a 4 times factor in

runnig times but since EdgeFlow has further steps, it is causes only this 20% gain. However – note that the results of EdgeFlow are given – due to earlier steps!
In my experiments I have also tried to use a function that is based on both pixelwise energy and orientation – that is a kind of "weighted histogram" – but this led to no increase in performance.
Note that the ALOE value is also low for normal high intensity masses since they have a clear contour, resulting in a wide range of orientations in the filtered image. So we could say that ALOE is conservative regarding our previous feature space. It still keeps non-spiculated masses close in feature space while separates spiculated ones from normal tissue – that had previously proven to be too close.

VI Performance in Time

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 we also focused our efforts on accelerating 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 to the orginal, while – according to the tests – performance is still as good as it was in the beginning. This means about 10 minutes of running time per pairs of images.

VII Summary and Future work

During my research I have managed to prove that there is a way of applying joint analysis on different views of mammographic images even though perfect 3D reconstruction is impossible. With the help if this joint analysis a remarkable percentage of false positive mass-candidates can be eliminated by much smaller loss of true positive ones. I also succeeded in improving a special texture feature for spiculated masses.
Lately I have been working on applying intelligent learning methods for pairing instead of rule based systems and also plan numerous tests to draw detailed ROC curves on performance.

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.
6 W. P. Kegelmeyer Jr., J. M. Pruneda, P. D. Bourland, A. Hillis, M. W. Riggs, and M. L. Nipper, "Computer-aided mammographic screening for spiculated lesions," Radiology, vol. 191, pp. 331–336, May 1994.

ASSESSMENT OF AORTIC STIFFNESS

András MERSICH
Advisor: Ákos JOBBÁGY

I. Introduction

High blood pressure, arterial stenosis and aneurism are the precursors of fatal cardiovascular diseases. However unreliable oscillometric BP measurement is usually still a sufficient enough tool for early detection of hypertension. Estimation of blood vessel properties on the other hand is still a blank spot. Although arterial stiffness would be a vital diagnostic parameter there is no reliable-enough non-invasive estimation method. There are several attempts to assess arterial stiffness through a cuff-based BP measurement, but these yet have not been proved to be reliable mainly because of the lack of directly measured signals and a sufficient model. Goal of this research is to fill in this gap and develop a simple, non-invasive diagnostic method assessing the stiffness of the aorta by means of a cuff and a DC-coupled photoplethysmographic (PPG) sensor.

II. Materials and Methods

A. Aorta vs. transmission line

The major arterial trunk of the body is called aorta. It is a muscular blood vessel with 2.5 cm diameter starting in a hairpin bend from the left ventricle of the heart running along the spine and ending in the bifurcation of the two femoral arteries. Along its way it gives branches to the head and the arms to the liver and kidneys and other intestines. A. brachialis, main blood vessel of the arm has by comparison only 0.4 cm diameter which ensures that only a marginal part of the total aortic flow is passed to the arms; a fact that will play a significant role in our further investigations, meaning that blocking the circulation of an arm will not cause any significant change in the aortic flow.

Stiffness of the aorta and its peripheral load in the form of branching out arteries and mainly the femoral bifurcation determine the form and propagation speed of the pressure wave called pulse. Figure 1. shows two aortic pressure pulses recorded in a senior (solid line) and junior (dot) person. Major difference between the two is found in the age related elasticity-change of the aorta.

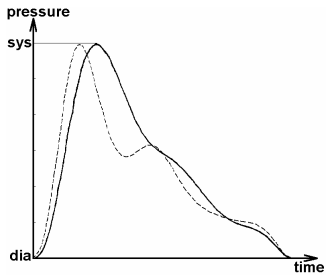


Figure 1: Aortic pulse waveform alter with age

There are lots of complex models describing such flow profiles in hydrodynamic studies [1] however their practical applicability due to numerous immeasurable parameters is rather questionable. For EE engineers the recorded signal corresponds to the output of a non-ideally terminated transmission line provided that pressure equals voltage and flow current. Several papers suggest the use of a transmission line as an equivalent for arterial flow [2]. Basic difference between hydrodynamic models and transmission lines is turbulent flow which will be neglected in this study.

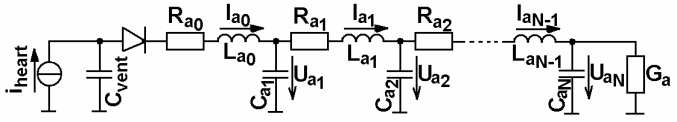


Figure 2: Transmission line representing the aorta.

In our representation heart is replaced by an ideal current source providing sinusoidal flow, a capacitance C_{vent} stands for the buffer effect of the left ventricle and a diode substitutes the semilunar valves. Aorta is modeled by an RLC transmission line with N discrete sections and a terminal load in the form of conductance G. As a convention N is the height of the patient in cm divided by six. Westerhof et al. [2] have determined a minimum N for given accuracy which is conveniently exceeded in our choice. State-variables U_{an} and I_{an} are the local pressure and flow values in the n^{th} section respectively. In each discrete length-unit C_{an}, R_{an} and L_{an} are functions of U_{an}.

Before proceeding with the discussion of the hydrodynamic equivalent of each RLC parameter let us consider a small but significant side-track. Provided that the aorta is not homogenous - the C value is drastically different in a single section for example - the output waveform will be different also. This suggests the chance of detecting and locating aneurisms.

B. Nonlinear parameters

Strangely enough the comparison of RLC and hydrodynamic parameters has not been thoroughly investigated in the past. Before proceeding, the term transmural pressure has to be defined. In a blood vessel it is the difference between internal and external pressure: $P_{trm}=P_{int}-P_{ext}$. In our representation each U_{an} is a local P_{trm}. Now the volume of an elastic tube as a function of P_{trm} is that of a sigmoid depicted in Figure 3. Compliance, a parameter commonly used in physiology, by definition is: $C=\Delta V/\Delta P$. In our context where voltage corresponds to pressure, electrical current to flow and electrical charge to volume capacitance is the equivalent of compliance: $C=dQ/dU=dV/dP$. From the $V(P_{trm})$ curve the flow resistance of the tube can also be derived according to Hagen-Poiseuille:

$$R(P_{trm}) = \frac{8l\mu}{\pi r^4} \cong c_1 \frac{1}{r^4} \cong c_2 \frac{1}{[V(P_{trm})]^2}, \qquad (1)$$

where l is the length of the tube, r its radius, μ the viscosity of blood; $c_{1,2}$ for practical applicability are considered constant. The exact equivalent of L and its relation to P_{trm} is yet unknown, it is in a way connected to the mass and inertia of the blood flow. For now on it is to be considered a constant.

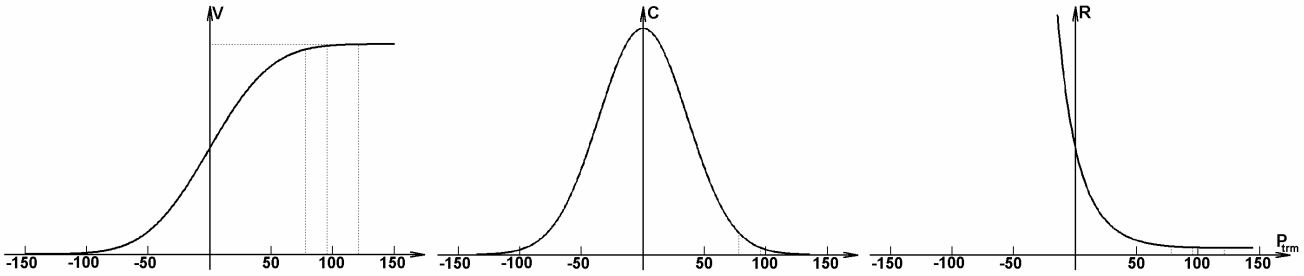


Figure 3: Volume, compliance and flow resistance of a blood vessel as function of transmural pressure. Dashed lines mark the normotensive physiological BP (120/80, MAP=93mmHg).

Introduction of the term transmural pressure becomes instantly necessary when the external pressure is changed either by a cuff or through a forced abdominal pressure in the so called Valsalva-maneuver. The drastic increase in abdominal pressure results in a decrease in the transmural pressure which in turn changes the compliace and resistance value of the aorta and forces it in a different steady state. The intervention is so impressive that its effects can be observed even in the microcirculation of a fingertip and thus recorded on a PPG.

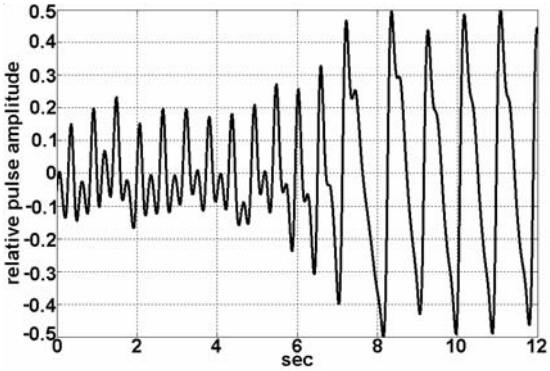


Figure 4: PPG during and after Valsalva-maneuver.

C. Measurement setup

Identifying model parameters in case the pressure P_{ao} is directly measured at a given point of the aorta is investigated thoroughly [3]. However in non-invasive diagnostics this obviously cannot be achieved. In our research we used a DC-coupled PPG, attached to a left hand fingertip, as a replacement for the lost P_{ao} data. Further we applied a cuff on the same arm to intervene in the circulation through P_{trm}. All this necessarily brought with itself the need to extend the model.

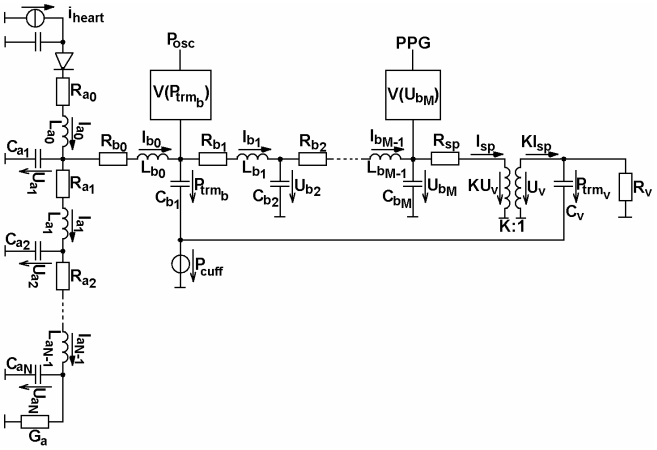


Figure 5: Extended model including heart, aorta, left arm arteries and veins, cuff and PPG sensor.

The extended model consists of heart, aorta and the circulation of the left hand complete with cuff and PPG sensor. Brachial artery is represented by a second transmission line with length height/4. The cuff intervenes in the arterial and venous system by changing the P_{trm} of two sections. Inflating the cuff will first block the venous circulation through R_v which becomes infinite shortly after P_{cuff} exceeds the absolute venous pressure U_v. Over the systolic value the cuff impedes the arterial flow too through R_{b1}, but still allows the generation of oscillometric pulses. The PPG signal, as P_{osc}, is volume-change based. Both are derived from their local P_{trm} by means of the $V(P_{trm})$ curve. Transformer substituting the capillary net is needed to prevent veins from oversaturating when $R_v=\infty$. In situ validation of the model was performed by a Colin CBM7000 tonometer. For measurements a unique protocol was developed: 20 sec rest, then Valsalva-maneuver for 20 sec, then another 20 sec rest and after that cuff inflation above systolic pressure and finally a slow deflation. P_{sys} was determined during cuff inflation at the pressure where PPG pulses faded away and MAP was measured in an oscillometric way during cuff deflation.

III. Results

Diagnosis comprises of the estimation of the model parameters based on a measurement according to the above described protocol in 3 discrete steps. First the $V(P_{trm})$ curve of the brachial artery was estimated during cuff deflation. Taking advantage of the fact that occlusion of the left arm does not intervene in the circulation of the entire body aorta was replaced by an ideal voltage source providing general pressure waves between P_{sys} and P_{dia}. Now the parameters of $V(P_{trm})$ were tuned according to gradient method to fit the measured oscillometric wave amplitudes.

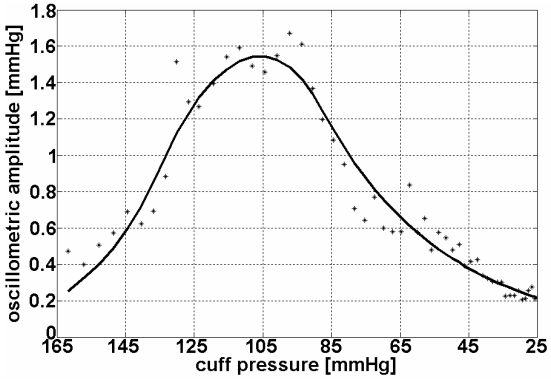


Figure 6: Oscillometric amplitudes during cuff deflation. Measured (dot), simulated (line).

As a second step the changes in the PPG baseline were investigated during cuff intervention. DC-coupled sensors enable the measurement of low frequency changes in the blood volume of the microcirculation. These changes are contributed to breathing and the intervention of a cuff. When cuff pressure exceeds about 20 mmHg the venous back-flow is blocked however the arterial input is not compromised. This leads to the saturation of the venous system which in turn slightly increases the MAP in the brachial artery. This saturation induced offset can be observed in the PPG baseline as depicted in Figure 7. PPG waves are in hardware derived from this very signal by means of a 0.28 Hz RC highpass filter and amplified by 28. During the estimation process aorta was still replaced by the ideal source providing however only constant MAP pressure this time. Parameters C_b, L_b, R_{sp}, C_v and R_v were identified while the assumptions R_b=1 and K=3 were made.

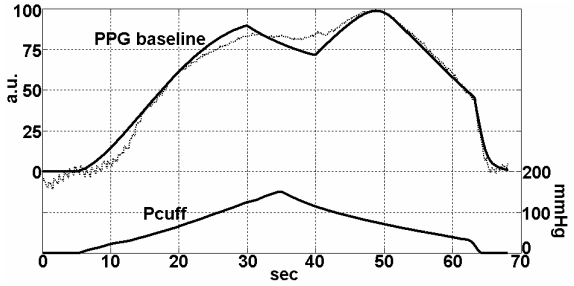


Figure 7: PPG baseline during cuff intervention. Measured(dot), simulated (line).

As a last step the parameters of the aorta were identified from an averaged PPG pulse at rest and during Valsalva-maneuver. P_{ao} was restored from the PPG signal through the by now ID-d model of the left arm. Heart was assumed to provide sinusoidal flow. Pressure dependency of parameters was neglected; R_a, C_a, L_a, G_a were considered constant. Differences between the two records are attributed to different C_a values.

IV. Conclusion

Identifying a system with unknown input and directly immeasurable output signals where the chance of intervening in the system is also quite limited is a mission impossible in itself. This is exactly the situation when estimating parameters of the aorta during a non-invasive diagnosis. However applying a cuff and a DC-coupled PPG sensor the chances are improved. The above introduced measuring protocol and identification process results in an estimated $V(P_{trm})$ curve of the brachial artery and five parameter values (R_a, L_a, G_a and C_a at rest and during Valsalva – maneuver) describing the aorta. A prototype device was developed which is about able to assess arterial stiffness by applying the above described model. This research was based on measurement series of numerous healthy persons where in vivo arterial pressure was also recorded through a Colin monitor. Further research is needed to investigate the possibility of neglecting parameters and thus speeding up the diagnosis process. Moreover a way has to be found to present medical doctors with the results.

Acknowledgement

The EU has supported the reported research work, GVOP grant 3.3.1.-2004-05-0340/3.0.

References

[1] A. P. Avolio, “Multi-branched model of the human arterial system,” *Med. Biol. Eng. & Comput.*, 18:709-718, 1980.

[2] G. N. Jager, N. Westerhof, A. Noordergraaf, “Oscillometry flow impedance in electrical analog of arterial system: Representation of sleeve effect and non-Newtonian properties of blood,” *Circ. Res.*, 16:121-133, 1965.

[3] C. M. Quick, D. S. Berger, R. H. Stewart, G. A. Laine, C. J. Hartley, A. Noordergraaf, “Resolving the homodynamic inverse problem,” *IEEE Tr. on Biomed. Eng.*, 53:no. 3, 2006.

MODELING HUNGARIAN LAW USING DESCRIPTION LOGICS

András FÖRHÉCZ
Advisor: György STRAUSZ

I. Introduction

The complexity of legislation and regulation is increasing significantly. Modern technology and business relations demand for more complex laws, also countries joining the European Union has to deal with new requirements. The faster development of technology results in more changes, faster fluctuation in regulations.

Citizens have difficulties understanding their rights and obligations, and applying regulations even for their own situation. Finding the appropriate articles in law may not be enough, as it is difficult to interpret legal terms, and resolving contradictory legal qualifications is far from trivial. New legislation from the European Union complicates application of law further in member states. However all citizens are expected to know and abide by the law.

Dealing with the tremendous amount of legal material raises bureaucracy and business costs as well. To solve these issues we need a wide spectrum of new methods and automated solutions. Professionals in the legal domain may need different services compared to citizens for their tasks, and also the main fields of work are also broad: drafting, implementing law, argumentation and judging.

Legal knowledge representation systems should help managing the growing complexity of legislation, supplying services for people with a diverse level of legal experience. The European ESTRELLA project (IST-2004-027665) is to develop an open, standards-based platform allowing public administrations to develop and deploy comprehensive legal knowledge management solutions. ESTRELLA will support both legal document management and legal knowledge systems, to provide a complete solution for public administration requiring the application of complex legislation and other legal sources. The interoperability of proprietary products of particular vendors should also be provided by a common Legal Knowledge Interchange Format (LKIF).

II. Domain and goals

In the ESTRELLA project we are about to construct a pilot application built upon the Hungarian Act on Value Added Tax (VAT). Selecting the tax material was driven by various aspects, providing a good starting point for experimenting with knowledge representations. VAT includes many different entities with complicated relationships. People not familiar with law has severe difficulties to understand or apply it for their own business activity. VAT contains calculations in connection with determining tax rates, which can be handled with reasoning and conventional computing methods. Compared to other fields of law (e.g. criminal law) VAT has less modalities (e.g. beliefs) and reifications: constructions which are hard to handle with lightweight reasoning facilities, as description logics (DL).

At the moment source documents are available as legal text in Word or HTML format. Later preprocessed XML (e.g. Metalex) documents can be used, which are to be produced in a parallel work. We expect that usage of XML documents makes development easier, as entities in the formal model and the source legal document can be interrelated through references. Also the quality of the knowledge base should not be affected.

The two main knowledge representations investigated in ESTRELLA are rule-based systems and description logics. Former legal expert systems used rules to describe knowledge. Frame-based production systems also include hierarchies of entities. In this experiment we investigate another ap-

proach, the expressive power of description logics. The final model of ESTRELLA should combine both representations.

The modeling language used is the DL subset of the OWL Web Ontology Language: it is not only theoretically well described and tractable (e.g. decidable) [1], but also well supported from the practical side. As OWL DL is widely used for ontology development, there are many knowledge base editors and reasoning engines freely available.

This is an experiment investigating the possibilities in OWL DL: how much of the legal material can be exactly described, and what kind of questions can the knowledge base answer. We will try to express as much as possible using OWL DL, also keeping the ontology maintainable. Artificial entities caused by the lack of language constructs are avoided, we also try to keep class definitions clean and simple.

The Hungarian Act on Value Added Tax addresses to answer three basic questions for tax payers: who has to pay tax, how much should be paid and in what form (rules of procedure). At first we try to answer the question of scope: reasoning on the ontology should decide whether a person has to pay tax in a given situation.

III. Foreseen limitations

Now as we have specified the knowledge representation as OWL DL, we can address some limitations of description logics.

Legal cases examine some legal action, brought about by a participant. The actor taking part in the action has to be qualified regarding legal rules. In a legal case three main relations have to be examined: has the actor brought about, intended to bring about and expected to cause the action. Intentions and expectations have a crucial role in criminal law, but also appear quite often in civil law.

Modeling intentions and beliefs is not possible with description logics, as no modal operators are available. Terminological knowledge representations can only describe time-independent facts using a set of instances. There is no way to reason in independent worlds, like reality and an agents beliefs, and have the two interrelate. Also possible modal extensions of OWL DL are investigated as in [2], in this paper we would stick to standard OWL DL.

Another issue follows from the distinct nature of rules and descriptive definitions as syntax of regulations is rule-based. Most of the articles formulate preconditions with a consequence. Looking at the semantics other features common to rule-based systems can be identified. When two statements conflict, contradictions are resolved in most cases by priority of rules. Priority may be based on level of statute (lex superior), level of restriction (lex specialis) and date of enactment (lex posterior). Also exceptions are very common in law. Priority and exceptions can be formulated using rules in a compact way, but obfuscate the descriptive representation.

We can manage some of the limitations by extending OWL DL with a rule language, not affecting the advantageous properties of the formalism. We will address rule-based extensions using SWRL later.

Since description logics has a limited expressive power, it is used to qualify events, classifying into legal categories. Inferring a final judgement based on the qualifications will be handled by a non-monotonic reasoning instrument.

IV. Methodology

We developed the OWL knowledge base using former experiments in ontology development [3]. The basic terminology is laid down using a top-down approach. Ontology guidelines are very useful in the whole process, also when extending the knowledge base with DL axioms.

First we identified main concepts in the legal material. The most important terms are required to

formulate the question and describe current situation or case, indispensable for reasoning. Some terms have a definition already in the legal text, although most have only implicit definitions. In case of VAT, key concepts include: economic activity, taxable person, personal scope, material scope, goods.

Concept and relation definitions formalize legal material, whereas instances describe case in question. Activities commited by an actor are represented by instances and instance relations, and qualified through classification. The reasoning can be effective if all classes have full definition, neccessary and sufficient conditions.

When defining conditions, obviously new concepts have to be added to the knowledge base. The most conspicuous differences between rule-based and descriptive representation emerge during this step. Handling exceptions demand for the introduction of artificial concepts (e.g. "occasional sale of new means of transport into another member state"). There exists no brief semantic for such classes, possibly they should remain as part of the class definition of a higher level concept. OWL supports unnamed classes for this purpose, but current tools lack features to handle them transparently.

The knowledge base should follow the semantic structure of the legal text. We try to map entities in the law to class definitions. Conditions should reflect the structure of the text, also keeping redundant assertions. This is necessary to verify correctness later, matching the DL expressions with the original legal material. Every OWL class references the articles (fully or partially) covered by the definition, helping the verification process.

After building the knowledge base we have to verify the semantics. Using a reasoning engine automatic analyzis of the model is possible. At first we should verify consistency: unsatisfyable classes pointing out errors in the model. Advanced tools like SWOOP are able to provide explanations in a readable manner, helping the user understand the cause for inconsistencies.

Classification is another tool to verify correctness. All subclass relations are inferred and a new class hierarchy is provided. Unexpected results may be caused by some implicit rule in the legal text or by unsound class definitions.

The final verification of the model is done using test cases. Similar to programming code analysis, test examples for every possible outcome of axioms should be included in a test set. The cases are prepared by a legal expert, and formalized using the terminology of the knowledge base. Creating an exhaustive test set is recommended, as running the tests correctness of the knowledge base can be verified automatically.

V. Reasoning example

Our VAT ontology tries to answer the question: does a person have to pay Value Added Tax in a given situation? This can be modeled through the personal and material scope of the law. Another common question is whether a person is a "taxable person", which is a similar class like the personal scope, but not a sub- neither a superclass of that.

Part of the class hierarchy detailing the economic activities are shown on figure 1. Material scope is inferred as subclass of economic activity, as this fact was introduced in a neccessary assertion. Concepts with gray background indicate full definitions, so entailment of individuals can be inferred.

To demonstrate the reasoning capabilities of the VAT knowledge base we have included two individuals with different activities. The two natural persons are Paula Brown and John Smith. The economic activities include acquisition of corn in Ontario and tobacco export. For automated reasoning we used Pellet integrated to Protégé ontology editor through the DIG interface.

The OWL knowledge base produced all the expected results. An example of non-trivial inferred statements are the following: both persons are taxable, as they take part in economic activity. Acquisition of corn is ,,importation of goods", as it is an acquisition from a third country territory (Canada). Tobacco export is not in the material scope of the law, as tobacco is governed under the "Excise Act," making it an exception for VAT.

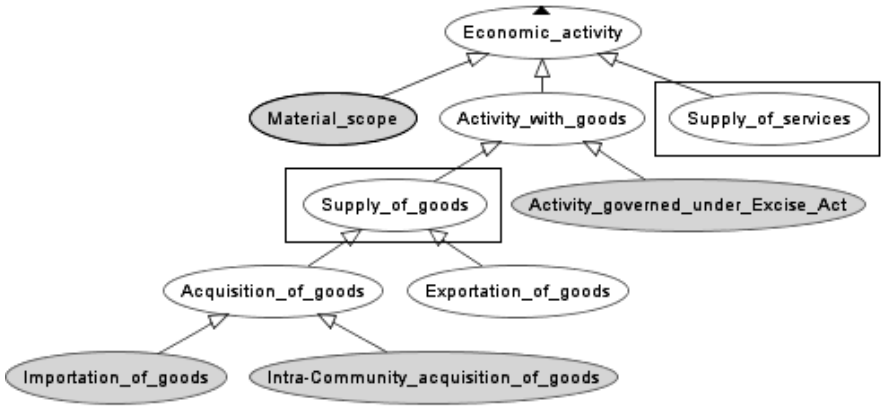


Figure 1: Inferred class hierarchy

Our first legal knowledge base proved that description logic is a useful tool for legal modeling. Some complex structures however are hard to model, e.g. contracting parties and other relevant properties of a contract. Especially assertions on class properties are hard or not possible to formulate.

VI. Future work

In our work now we try to overcome the limitations of description logics by extending the formalism. We know that OWL DL is equivalent to a DL, namely $\mathcal{SHOIN}(\mathbf{D})$. However OWL DL is already established to support most DL constructs while remaining decidable and tractable.

A heavy rule-based extension of OWL DL is SWRL, unfortunately SWRL does not inherit the tractable property. A new approach in extending OWL DL with rules while still remaining in the decidable subset is SWRL DL-safe [4]. We are now trying to exploit the possibilities of this rule formalism in legal modeling to overcome the mentioned limitations. Support for SWRL DL-safe is very limited. Available tools for reasoning include Hoolet and Pellet DL-safe.

Hoolet is a simple tool, which maps OWL DL axioms and SWRL rules into first order logic (FOL) expressions. The FOL knowledge base is processed by a first order theorem prover through the TPTP interface, also Vampire is included. As this is a naive approach, performance suffice only for small illustrative examples. However as for the FOL conversion Hoolet is not required to restrict SWRL to the decidable subset, the whole expressive power of SWRL may be used. Hoolet is the only reasoner freely available supporting the whole OWL DL + SWRL language.

The Pellet OWL DL reasoner has recently been extended with DL-safe reasoning facilities [5]. The current implementation is far from efficient, but unlike Hoolet it is based on the Tableaux algorithm, so later improvement of Pellet reasoning performance is expected.

References

[1] I. Horrocks and P. Patel-Schneider, "Reducing OWL entailment to description logic satisfiability," in *Proc. of the 2nd International Semantic Web Conference (ISWC)*, 2003.

[2] F. Baader and A. Laux, "Terminological logics with modal operators," in *Proc. of the 14th International Joint Conference on Artificial Intelligence*, pp. 808–814, Montreal, Canada, 1995.

[3] N. F. Noy and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology," .

[4] B. M. Pascal Hitzler, Jürgen Angele and R. Studer, "Bridging the paradigm gap with rules for OWL," in *Proc. of the W3C Workshop on Rule Languages for Interoperability*, Washington D.C., USA, Apr. 2005.

[5] B. P. Vladimir Kolovski and E. Sirin, "Extending SHOIQ(D) with DL-safe rules: First results," in *Proc. of the International Workshop on Description Logic*, 2006.

A New Approach to the Clustering of Directed Graphs

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

I. Introduction

Scientists of all disciplines have to deal with ever increasing amount of empirical data. Large portion of these data can efficiently be represented as graphs. Some well-known examples are social networks [1], technological networks such as the Internet at its router level, or biological networks – e.g. the major brain areas and their neural connectivity patterns [2]. Several methods have been developed to extract underlying relevant structural properties of such networks and to aid the visualization of such data sets. A common objective for these methods is to group ”similar” and densely interconnected vertices together in clusters.

This approach has proved its usefulness, but there can still be a lot of information in these data sets that can't be identified with traditional clustering methods. For instance, it is practically impossible to identify bipartite or almost bipartite subgraphs. Most of these methods are also difficult to extend to directed graphs.

In this paper, a new, stochastic approach to the clustering of directed graphs is presented. First, a short overview of the current state of the art will be given. Then the underlying theory of this alternative method will be explained and a possible implementation will be proposed. To prove the validity of this approach, benchmark results on computer-generated graphs are presented.

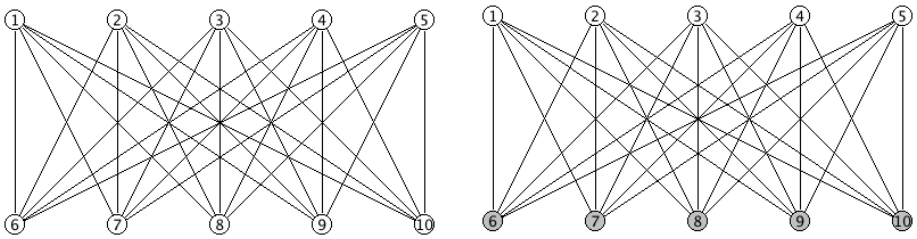
II. Methods of cluster detection

A. Traditional methods and their shortcomings

In the simplest case, “cluster” is equivalent to a strongly connected component of a graph. However, this definition is of no use if the graph is connected. As a workaround, one might progressively remove edges of the graph in a given order until the desired number of clusters is reached, or start with no edges and add them one by one. This simple approach is the basis of the family of hierarchical graph clustering methods. A notable member of this family is the Girvan-Newman algorithm [3], where the edges are removed in decreasing order of edge betweenness (the number of shortest paths passing through the given edge).

Another large family of graph clustering algorithms are the flow-based methods where a large amount of random walks are conducted along the edges of the graph starting from randomly selected nodes. A cluster is defined as a set of nodes that are interconnected ”densely enough”, so it is hard to escape from them with a random walk. A fast and unsupervised variant of this algorithm family is the Markov Cluster Algorithm [4].

It is important to emphasize that all graph methods presented so far define clusters as subsets of the vertices with a dense connectivity. This approach is meaningful, but there are cases when a set of vertices can be interesting or important for the researcher because they are not connected at all. Consider Fig. 1 as an example. The graph shown here is a complete balanced bipartite graph with 10 vertices. A meaningful view of this graph is the one where the structurally relevant groups are the white and gray vertex sets (see the bottom part of the figure) based on the fact that all edges go between a white and a gray vertex – but this contradicts the traditional definition of clusters as a set of vertices with many edges between them!



(a) Traditional methods (b) Proposed method

Figure 1: Clustering of a bipartite graph. (a) – clustering with a traditional method. (b) – clustering according to human perception and the proposed method

All methods presented so far tried to – explicitly or implicitly – create a partition of the vertex set V such that intra-cluster edge densities are maximal and inter-cluster edge densities are minimal. To formalize this statement, let us introduce the concept of edge density and the edge density matrix. The edge density of two given vertex sets $V_1, V_2 \subseteq V$ and a graph $G(V, E)$ is the number of edges connecting vertices in V_1 with vertices in V_2 divided by the product of the cardinality of these two sets:

$$d(V_1, V_2) = \frac{|\{(v_i, v_j)|v_i \in V_1 \wedge v_j \in V_2 \wedge (v_i, v_j) \in E\}|}{|V_1||V_2|} \tag{1}$$

The density matrix $\mathbf{D}(G)$ for a given partition $V_1, V_2, \ldots, V_n$ is built from the $d(V_i, V_j)$ values for all (V_i, V_j) pairs. Diagonal elements of $\mathbf{D}(G)$ represent the intra-cluster while off-diagonal elements are the inter-cluster edge densities. By maximizing intra-cluster and minimizing inter-cluster edge densities, traditional clustering methods fail to unveil the bipartite structure of the graph shown on Fig. 1. The partition shown on the right side of Fig. 1 actually results in a density matrix where the diagonal elements are zero and the off-diagonals are 1.

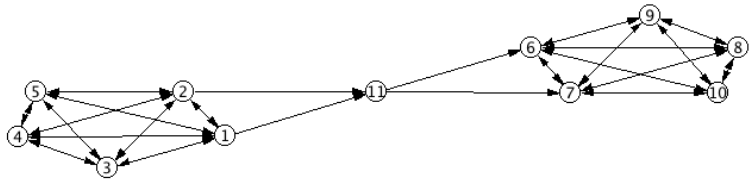


Figure 2: Graph with two clusters and a single vertex as a bridge between the clusters

Another problem arises when one tries to apply these algorithms to directed graphs. For instance, consider a graph composed of two full graphs and a vertex which receives all of its incoming connections from one of the full graphs while sending all outgoing connections to the other (see Fig. 2). Most algorithms put node 11 in either into a separate cluster (hiding the fact that it's actually a bridge between the two clusters) or arbitrarily group it together with one of the two large clusters. The algorithm presented here is able to overcome this difficulty by putting such bridge-like nodes into two different clusters at the same time if necessary.

B. The proposed method

Formally speaking, the problem considered here can be formulated as follows: given an arbitrary (directed or undirected) graph $G(V, E)$ with no multiple or loop edges and the desired cluster count k, find two partitions $U_1, U_2, \ldots, U_k$ (output partition) and $V_1, V_2, \ldots, V_k$ (input partition) of V such that the likelihood of the graph (considering the edge densities obtained from the partitions as probabilities

of edge existence between sets of vertices) is maximal. The likelihood of the graph is defined as:

$$L = \prod_{i=1}^{n} \prod_{j=1}^{n} \begin{cases} d(U_i, V_j) & \text{if } (v_i, v_j) \in E, v_i \in U_i, v_j \in V_j \\ 1 - d(U_i, V_j) & \text{if } (v_i, v_j) \notin E, v_i \in U_i, v_j \in V_j \end{cases} \quad (2)$$

where n is the number of vertices. Informally, for every edge (v_i, v_j), we find the set U_i where v_i belongs to and the set V_j where v_j belongs to, calculate the density with respect to these two sets and multiply them. At the same time, for every (v_i, v_j) pair with no edges between them, calculate the density, subtract it from one and multiply them together with the previous product. This measures the likelihood that our specific graph would emerge from a model where the edges are placed with the given probabilities. Our goal is to maximize the likelihood.

According to the notation above, U_i-s and V_j-s together define our clustering. U_i is a partition of the vertex set based on their outgoing edges, while V_j is a partition based on their incoming edges. Undirected graphs and directed graphs without bridge-like nodes yield the same partition for both U_i-s and V_j-s. Bridge-like nodes will belong to different sets among U_i-s than among V_j-s. Let's take Fig. 2 again as an illustration! The partition belonging to this graph with $k = 2$ is the following:

$$U_1 = \{v_1, v_2, v_3, v_4, v_5\}, U_2 = \{v_6, v_7, v_8, v_9, v_{10}, v_{11}\}$$
$$V_1 = \{v_1, v_2, v_3, v_4, v_5, v_{11}\}, V_2 = \{v_6, v_7, v_8, v_9, v_{10}\}$$

Note that v_{11} belongs to different sets among U_i-s than among V_j-s due to the different incoming and outgoing connectivity pattern of v_{11}.

C. Implementation details

It can be proven (but it extends well beyond the scope of this paper) that this problem is NP-complete. Due to the huge size of the parameter space, probably the best approach to maximize the likelihood is to use stochastic discrete optimization methods.

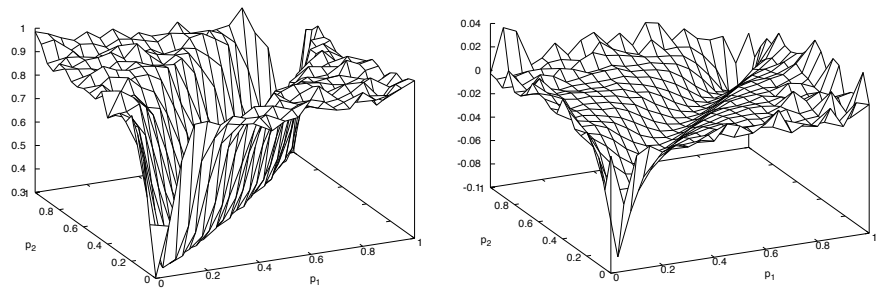
The algorithm we used consists of an initialization and a core with two alternating steps: a deterministic, greedy optimization and a stochastic mutation phase. The initialization is responsible for generating a solution which is sufficiently near to the global optimum in the parameter space. The core iteratively improves this solution until the global optimum is found or the allocated computational resources (time, memory etc.) run out.

Finding a good starting point is crucial for all stochastic optimization methods. In this particular problem, the singular vectors obtained from the singular value decomposition (SVD) of the adjacency matrix of the graph have proven to be particularly useful.

The deterministic phase tries to advance along the steepest descent in the state space using a simple heuristic until no further improvement is possible. After conducting a successful deterministic phase, the current optimal solution is randomly mutated in m substeps. Every substep involves putting a randomly chosen vertex into another set in either one of the partitions (U or V), or swapping two randomly chosen vertices both in U and V. The mutated clustering is then fed back into the input of the deterministic phase and the process starts again until the allocated resources (time, space) run out.

III. Test results

To test the validity of the algorithm, several benchmarks have been run on various computer-generated test graphs. For each test graph, the original and the calculated cluster structure was compared and a hit rate was calculated as the ratio of the number of vertices classified correctly and the total number of vertices. The ratio of the inverse log-likelihood of the test graph with respect to the original (expected, L_e) and the resulting (observed, L_o) cluster structure was also calculated according to the following formula: $r = \frac{L_o - L_e}{L_e}$. When r is exactly zero, the likelihood of the graph with respect



(a) Hit rate (b) Likelihood ratio

Figure 3: Average hit rates and likelihood ratios (r) plotted against p_1 and p_2

to the resulting cluster structure perfectly matches the expected one. Positive r values show that the algorithm failed to precisely discover the underlying cluster structure while negative r values indicate that the algorithm found a cluster structure which results in a higher likelihood than the original graph.

Test graphs were generated using a simple stochastic model based on vertex types. Each vertex is assigned an integer (the "type" of the vertex) from 1 to 4 with equal probability. A 4×4 preference matrix is generated with each element of the matrix being either p_1 or p_2 with a probability of 0.5. The preference matrix represents the connection probabilities between given types of vertices. p_1 and p_2 directly influence the clustered organization of the graph: if one is high and the other is low, vertex pairs with a probability of the higher one will tend to connect to each other. We expect the algorithm to perform well in the cases when $p_1 \gg p_2$ or $p_1 \ll p_2$, with r being close to zero and the hit rate close to 1. If $p_1 \approx p_2$, the graph will not have a clustered organization at all.

20 test graphs with 128 vertices were generated for each (p_1, p_2) pair with a step size of 0.05 and the algorithm was run on each test case for at most 100 steps. Hit rates and likelihood ratios were averaged for every (p_1, p_2) pair. The results are presented on Fig. 3. The algorithm detected the underlying cluster structure correctly in the majority of test cases, provided that such structure existed. When no feasible cluster structure was present in the original graph, the algorithm usually fitted a clustering which resulted in a higher likelihood than what was calculated with respect to the original vertex types.

IV. Conclusion

In this paper, an alternative method for identifying structurally relevant group of vertices in a graph was presented. The method performs well in finding densely connected clusters as well as bipartite subgraphs, and it is able to consider a vertex as part of two clusters depending on its incoming and outgoing connections.

Acknowledgement

The author is grateful to Gábor Tusnády for useful ideas and discussions.

References

[1] P. Holme, C. R. Edling, and F. Liljeros, "Structure and time-evolution of an internet dating community," *Social Networks*, 26:155–174, 2004.

[2] L. Négyessy, T. Nepusz, L. Kocsis, and F. Bazsó, "Prediction of the main cortical areas and connections involved in the tactile fuction of the visual cortex by network analysis," *European Journal of Neuroscience*, 23(2):1919–1930, 2006.

[3] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," in *Proc. Natl. Acad. Sci. USA*, vol. 99, pp. 7821–7826, 2002.

[4] S. van Dongen, "A stochastic uncoupling process for graphs," Tech. Rep. INS-R0011, National Research Institute for Mathematics and Computer Science in the Netherlands, 2000.

AUTONOMOUS MAPPING IN IN-DOOR ENVIRONMENTS

Balázs TÓDOR
Advisor: Gábor HORVÁTH

I. Introduction

In the recent years several algorithms have been proposed to solve the problem of autonomous robot mapping. Most of these are based on either metric or topological world models [1]. The latter store key features of the environment relative to each other, so in most cases they are represented with graphs, and as such, contain only high level information about the environment. This means that while they are harder to build, they provide an easy way to navigate.

The metric models are quite the opposite: in most cases they are represented as grids, top view maps of the environment. Each cell contains an occupation probability, a value that can be easily obtained from the reverse sensor model functions. A major disadvantage of the grid is that in order to keep the precision high, they require a huge amount of storage space that presents a maximum size limit for robot applications, and they are harder to navigate than graphs.

The algorithm presented in this article tries to combine the advantages of both worlds.

II. The measurement process

After a measurement is taken, the result (a scalar value) is converted and then inserted into the agent's world model. A measured distance becomes an obstacle hypothesis that is placed in the most likely place where an obstacle in the real world would be.

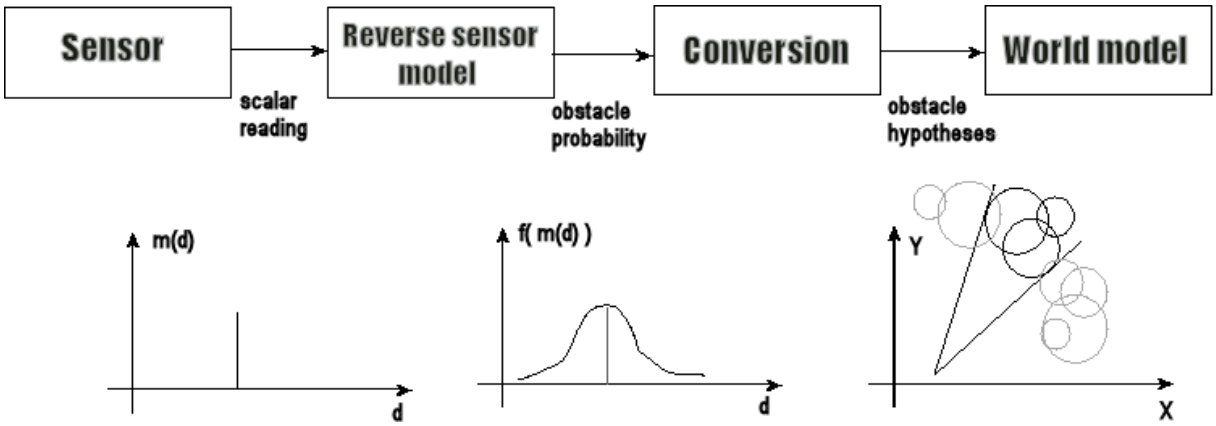


Figure 1: The measurement process

The range sensor's random errors are handled by assigning a probability density function to the scalar measurement. This is done by the reverse sensor model which tries to estimate the density function $f(d|m(d))$, where *d* is the real distance from the obstacle causing the reading and *m(d)* is the reading itself. This step is pretty well researched, by for example Thrun [1].

III. The obstacle hypotheses

The world model space obstacle hypotheses are formed by sampling this density function and placing virtual obstacles in the corresponding places in the world model. Each such hypothesis is represented by a position, a direction and a confidence value.

By carefully filtering the sensor readings the error that remains can be modeled as a linear function of the measured distance. The smallest detectable object size is also a linear function of the distance. Under normal circumstances it is safe to assume that there is a linear relationship between the size of the hypothesized object and the measurement error estimate.

This also means that the size of an obstacle in the world model is a confidence value as well.

IV. Data combination

When a measurement is taken in an already visited part of the world, the resulting hypothetical obstacles have to be combined into the world model instead of just being simply stored. This way the information level in the model will increase while the required computation times will remain low.

The combination of two world model obstacles is the weighted average of their positions and radii.

$$\begin{aligned} \vec{P}' &= \vec{P} \cdot (1-\alpha) + \vec{P}_2 \cdot \alpha \\ r' &= r \cdot (1-\beta) + r_2 \cdot \beta \\ \vec{A}' &= A \end{aligned} \tag{1}$$

In Equation (1) P is the center of the object hypothesis, r is the size (or confidence) and A is the unit axis. The two coefficients consist of three main terms: a constant, a radius-dependent part and a direction sensitive part.

$$\begin{aligned} \alpha &= \mu \cdot \left(0.01 + 0.09 \cdot \frac{r - r_{MIN}}{r_{MAX} - r_{MIN}} \right) \cdot \left(\vec{A} \cdot \vec{A}_2 \right)^2 \\ \beta &= \mu \cdot \left(0.01 + 0.09 \cdot \frac{r - r_{MIN}}{r_{MAX} - r_{MIN}} \right) \cdot \left(\vec{A} \cdot \vec{A}_2 \right) \end{aligned} \tag{2}$$

As mentioned above, the obstacles have an axis property. This stores the direction from which the measurement was taken. This is important since the reading errors are mostly radial. By taking this into account the world model building process will be more efficient.

In an in-door environment we can assume that most obstacles are large and that they are close to the robot. They are close in the sense that the associated hypotheses are smaller than the real obstacles. This means that in most cases the two different sides of a real world object are represented by different hypothetical objects in the world model. This is very fortunate because this allows us to skip the data fusion process for all objects that were detected from the other direction, simply because they belong to some other side of a real world obstacle, and thus can't be affected by the current measurement.

The experiments have shown that the position calculation is more sensitive to the measurement direction than the radius. This is why the directional component in the weight is squared when calculating α.

V. Unrelated hypotheses' combination

There is a special case to the hypotheses combination. There are areas covered by a sensor reading that might have obstacles in the world model whereas in the real world they don't. By creating a special method of combination for this case we can let the world model slowly remove these false readings.

The strength of relation between two hypothetical obstacles shows the relative overlapping of their confidence circles:

$$S(W_1, W_2) = 1 - \frac{\left|\vec{P}_1 - \vec{P}_2\right|}{r_1 + r_2} \tag{3}$$

When *S* is less than zero, the two hypotheses are considered unrelated. This happens when a measurement reports an obstacle far from what the world model suggests. In this case the false reading's confidence should slowly decrease. This can be achieved by combining a very low confidence hypothesis into the false reading. This way its radius will slowly grow while its center moves in the right direction. Sooner or later it will move into the area where it should've been in the first place.

There is one very rare case when this doesn't happen. If the reading was caused by some one time phenomenon in a large open area, the false hypothesis might never get close to any real world obstacle. In this case its size will continue to grow, and eventually it will get filtered out by the world model cleaning algorithm. This is provided as a last resort only, since the basic assumption of in-door environments makes this case really rare.

VI. Clean-up algorithm

The efficiency of the algorithm decreases over time as the world model grows. There are two problems that have to be handled here.

The first is the case of the growing false reading hypotheses which might cover all the free space in the model if they were allowed to. This is easy to handle: a simple check is run every few seconds, and it removes all hypotheses that have too high a radius.

The other case is more complex. It is assumed that each measurement covers more than one obstacle, each of which could cause the same reading alone. However, if the size of the real world obstacle in the view cone of the sensor is smaller than the total size of the hypotheses, it will cause the hypothetical objects to slowly converge to the same place. They will asymptotically move into the same position.

This is normal, because the world model elements are hypotheses that are independent of each other, so nothing stands against them occupying the same place. However, this degrades performance, so every now and then one of each highly overlapping pair has to be removed.

VII. Topological model

The world model is a topological model itself since the hypotheses themselves can be treated as landmarks. The only problem is that there are many more of them than in pure topological models. This makes classic algorithms hard to use. (However, the continuous space A* path finding algorithm should work fine.)

VIII. Metric model

It is more interesting to convert this world model into a grid. This is done by a simple rasterization algorithm that treats each cell independently.

For each cell, an impact value is calculated for each hypothesis, and the maximum is stored as the cell occupation probability (4).

The impact value is based on the strength of relation between the cell and the hypothetical object and weighted by the latter's radius.

$$\mathrm{CellValue} = \max_{W}\left(S(W, \mathrm{Cell}) \cdot e^{1.25 \cdot \frac{r - r_{MIN}}{r_{MAX} - r_{MIN}}} \right) \tag{4}$$

The result can be seen in the image below. The dark boxes are the real world obstacles. The hypotheses themselves are not shown because the picture would have been too cluttered. The white spots are the grid cells above the 1.0 threshold. These can be treated as fully occupied cells. The white areas do not cover all the area of the real world objects, since the range sensor used has a very wide field of view, and thus, a very low resolution.

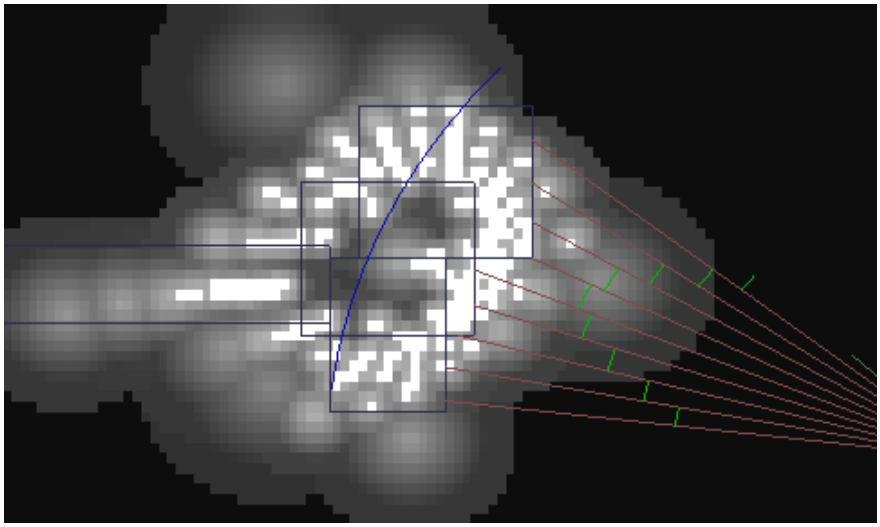


Figure 1: World model

Even if it is not possible to make out the shapes of the real world objects, it can be seen that the resolution of the world model can be higher than that of the sensor.

IX. Conclusion

The goal of this algorithm was to build a world model that has a higher resolution than the used sensor. While the research is not even close to complete the preliminary results show that it is possible to create such an algorithm that is more efficient than the grids and simpler than the topological world mapping methods.

X. Future work

The world model building algorithm should be tested in real life situations. The final goal is to create a simultaneous mapping and localization method, which this method will be an integral part of.

Acknowledgement

The author would like to thank András Széll for his useful comments during the research.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, Cambridge, MA, 2005

A FIRST-ORDER BAYESIAN LOGIC FOR HETEROGENEOUS DATA SETS

Gábor HULLÁM
Advisors: Péter ANTAL, György STRAUSZ

I. Introduction

In many areas of intelligent data analysis, especially in the field of biomedicine, the statistical inference about semantic properties of a model using heterogeneous data sets became an important issue. The combination of first-order probabilistic logic (FOPL) and the Bayesian framework provides a wide variety of different alternatives for the specification of semantic priors, models and queries. This field however still has many open problems.

Our long-term goal is to provide a unified probabilistic framework for inferring with heterogeneous data using hierarchical models and the power of first-order logic for formulating queries about the models. In this paper we describe a methodology behind a system we are currently working on, which aims to implement such inferences. The methodology contains the following major components: (1) A hybrid Bayesian semantics over first-order sentences which enables a model-based embedding of complex posteriors. (2) A stochastic annotated graph-grammar to define priors for hierarchical Bayesian networks. (3) A probabilistically linked model-network for the fusion of heterogeneous models. Here we focus on the hybrid Bayesian semantics and on the fusion of heterogeneous models.

II. FOPL

In the fields of knowledge representation information is often expressed in logic statements since they have clear, easily interpretable semantics. They lack however the ability to represent uncertainty which can be done efficiently by another set of tools: probabilistic graphical models, more specifically Bayesian networks (BN). Since many applications require both the descriptive power of first-order logic and the ability to represent uncertainty, the interest in combining the two approaches and developing a unified representation has grown significantly in the recent years.

A. First-order logic

First-order logic allows the use of predicates and quantifiers, thus it supports statements about groups of objects. This expressive power is needed in many application areas despite its computationally demanding algorithms. Furthermore, it prevents such obvious incorporations of probabilistic information as in propositional logic.

The basic inference problem in first-order logic is to determine whether a knowledge base (KB) entails a formula F, (i.e., if F is true in all worlds where KB is true). Unfortunately, inference in first-order logic is only semidecidable. Therefore, most of the knowledge bases are constructed using a restricted subset of first-order logic which in turn affects the representational capabilities in a negative way. Due to these limitations, pure first-order logic has limited applicability to practical problems.

B. Bayesian Networks

Bayesian networks(BNs) represent the modeled domain by a directed acyclic graph (DAG), where the nodes are stochastic variables (the entities of the domain) and the edges can be regarded as direct probabilistic dependencies. Besides the DAG structure there is a conditional dependency model assigned to each variable (in the following, we will use the terms variable and node as synonyms), describing how the given node depends on its parents. In a sense, the Bayesian network itself implements

a hierarchic model structure: the structure of the DAG is the higher-order discrete component, while the local dependency models, described through numeric parameters form the lower-order continuous component. In case of discrete Bayesian networks (where every variable is discrete) the most common form of the local dependency models are the *conditional probability tables* (CPTs), which describe the probability distribution of the child variable conditional on its parents.

C. Earlier Works

Two of the early works that attempted to unify logic and probability was [1], which introduced a combination of the domain-frequency approach and the possible worlds approach, and [2], which described knowledge-based model construction (KBMC). Its underlying concept is to order dedicated networks to different queries or modeling situations by constructing them on-the-fly from a background knowledge base.

The Relational Bayesian network (RBN) [3] is another possible way to represent probabilistic first-order statements. In RBNs every predicate is represented by one node in the network and each of them has a probability formula assigned to it describing its probability distribution conditional on its parents.

Another FOPL approach is the BLOG (Bayesian Logic) language [4], which defines a probability distribution over model structures of a typed first-order logical language.

Other examples of probabilistic extentions to first-order reasoning are Stochastic logic programs [5], Bayesian logic programs [6], and Markov logic networks [7].

III. FOPL by model-based embedding of posteriors

The general approaches summarized earlier define distributions from elementary probabilities over worlds (object, relations, etc.), possible worlds (literals) or knowledge bases (syntactic statements), where the result can be interpreted as an extended monolithic BN defined hierarchically and/or recursively. This section describes a more modest hybrid approach for fusing a certain logical knowledge, which mainly contains factual free-text information and complex unnormalized posteriors for hierarchically defined BNs.

To introduce a hybrid approach to FOPL first consider the probabilistic model-based semantics for propositions. In propositional logic, the joint value-assignments (atomic events) $\boldsymbol{x}$ are the canonic representations of the worlds under various interpretations (resulting in the possible worlds). The set of models of a propositional knowledge base $\mathcal{K}$ is the set of worlds $\boldsymbol{x}$ where $\mathcal{K}$ is true in a given interpretation. The knowledge base semantically entails a sentence α (denoted with $\mathcal{K} \models \alpha$) if α is true in all models of $\mathcal{K}$. That is the knowledge base defines a truth-value for the possible worlds and the truth-value of the sentence is defined by $\forall x\ \mathcal{K}(\boldsymbol{x}) \rightarrow \alpha(\boldsymbol{x})$. The definition of probability as truth-value of a sentence is similar to this, as the probabilistic knowledge base defines a distribution over the models and the truth-value of the sentence can be defined as the expectation $E_{p(\boldsymbol{x}|\mathcal{K})}[\alpha(\boldsymbol{x})]$ expressing its coverage by the models of $\mathcal{K}$ (normalized to the model of the KB).

In first-order logic this approach requires a distribution over worlds with interpretations M containing potentially varying number of objects and predicate and functional relations between them. Whereas this task is an open research problem, earlier works addressed several restricted cases, such as the works on the relational probability models [8, 9].

A frequently applicable hybrid approach defines the distribution over the models $p(M)$ of a logical knowledge base using an additional probabilistic model. The logical knowledge base $\mathcal{K}^l$ describes the certain knowledge in the domain and defines the set of models (possible worlds) $\mathcal{M}(\mathcal{K}^l) = \{M : \mathcal{K}^l(M)\ is\ true\}$. The probabilistic knowledge base $\mathcal{K}^p$ expresses the remaining uncertain knowledge over these worlds by defining a distribution over these models $p(M|M \in \mathcal{M}(\mathcal{K}^l)$. That is the uncertain knowledge only weights models but it does not narrow the set of models any further. So the probability

of a sentence α is defined as the expectation of its truth in valid worlds $\boldsymbol{M}$.

$$p(\boldsymbol{M} : \alpha(\boldsymbol{M})|\mathcal{K}^l, \mathcal{K}^p) = E_{p(\boldsymbol{M}|\mathcal{K}^l,\mathcal{K}^p)}[\alpha(\boldsymbol{M})] = \sum_{M \in \mathcal{M}(\mathcal{K}^l)} \alpha(\boldsymbol{M}) p(\boldsymbol{M}|\mathcal{K}^p) \tag{1}$$

If the models vary only in a well-defined respect such as a given object, this regularity can be used to define the distribution over the models based on a distribution over this respect.

IV. Augmented BNs for multiple data sets and models

The need for the joint analysis of multiple, heterogeneous data sets calls for the usage of Bayesian statistics and probabilistic graphical models, especially Bayesian meta modeling, which offers a normative solution for the problem.

We propose two assumptions to derive a practical Bayesian meta model. First we do not model uncertainty over the structural and parametric dependency of the models, i.e. we fix a prior S^M, θ^M and we do not perform Bayesian inference/learning over them. Second we assume exclusive correspondence between the random variables of a domain and its model class. Furthermore to simplify presentation we omit the parameters assuming that they are averaged out for the discrete models.

These assumptions result in a two-layered structure using the augmented BN representation. The upper layer is a BN with nodes $\mathcal{M}_1, \ldots, \mathcal{M}_K$ representing model classes, and their joint distribution $p(\mathcal{M}_1, \ldots, \mathcal{M}_K)$. The lower layer contains the observations $D_{N_i}^{(i)}$ connected as children to the appropriate model class without interconnections, which conforms to our assumption about exclusivity and exchangeability.

An important consequence of these assumptions that inference is decomposed as follows:

$$p(\mathcal{M}_1, \ldots, \mathcal{M}_K|D_{N_1}^{(1)}, \ldots, D_{N_K}^{(K)}) \propto p(D_{N_1}^{(1)}, \ldots, D_{N_K}^{(K)}|\mathcal{M}_1, \ldots, \mathcal{M}_K) p(\mathcal{M}_1, \ldots, \mathcal{M}_K) \tag{2}$$

$$= \prod_{i=1}^{K} \prod_{l=1}^{N_i} p(D_{N_i}^{(i,l)}|\mathcal{M}_i) \prod_{i=1}^{K} p(\mathcal{M}_i|Pa(\mathcal{M}_i)). \tag{3}$$

It means that the effect of observations can be computed independently for each model class and can be incorporated as a virtual evidence into inferences at the upper layer among the model classes, e.g. to compute a marginal for a given model class $p(\mathcal{M}_i|D_{N_1}^{(1)}, \ldots, D_{N_K}^{(K)})$ using MC methods.

In integrated learning from heterogeneous sources however, rescaling of belief for the sources is advisable to express our confidence in them, which can be easily achieved by flattening/peakening the virtual evidences for the model classes.

The specification of the local probability dependency models $p(\mathcal{M}_i|Pa(\mathcal{M}_i))$ in the upper layer of the model-augmented BN can be done by using feature sets, such as parental sets $Pa(\mathcal{M}_i)$, the directed edges and undirected edges.

V. The Bayes^3 system

The Bayes^3 system, upon which we are currently working on, aims to offer a unified probabilistic framework for inferring with heterogeneous data using hierarchical models and the power of first-order logic for formulating queries about the models. It is based on the before mentioned components: the hybrid Bayesian semantics (see Section III.), the probabilistically linked model-network (see Section IV.) and posteriors for BNs $p(M_i|D_{N_i}^{(i)})$, potentially with so called hyper-posteriors corresponding to their hierarchical, modular definition H_i, $p(M_i, H_i|D_{N_i}^{(i)})$.

The extension of this hybrid method to the general case when the models of the $\mathcal{K}^l$ vary in multiple, but well-defined respects is straightforward.

A. *Inference*

We can identify two inference methods suited for this model-based FOPL. The first is basically an enumeration method, iterating over the models, evaluating the truth-value of the sentence weighted with the probability of the model or iterating through only the models where the target is true. The second method directly estimates the expectation of its truth using an MC method with the indicator function of the truth-value of the target sentence.

B. *Examples*

For example using the posterior over BN structures given a particular data set the following statement has well-defined probability given the data sets

$$\forall v_1, v_2 : Var(v_1) \wedge Var(v_2) \wedge SimilarText(v_1, v_2) \Rightarrow \neg Independent(v_1, v_2, targetDAG) \quad (4)$$

where the predicate $Independent(v_1, v_2, G)$ means that the variables v_1, v_2 are not confounded or causally not connected in the DAG G.

Analogously, the posterior of a sentence including elements of the hierarchical definition of BNs exist, such as the occurrence of the ***M'*** module in the target model M_i

$$p(M' \subseteq M_i | D_{N_1}^{(1)}, \ldots, D_{N_K}^{(K)}) = \sum_{\substack{M_j \\ i \neq j}} p(M_1, \ldots, M' \subseteq M_i, \ldots, M_K | D_{N_1}^{(1)}, \ldots, D_{N_K}^{(K)}) \quad (5)$$

VI. Conclusion

The fusion of heterogeneous information resources, particularly the integration of electronic prior knowledge such as knowledge bases and free-text with expertise and experimental data is of vital importance. We presented two components of the Bayes[3] framework that will allow a practical and still normative Bayesian fusion of literature, experimental data and expertise. Currently we are working on the implementation of these methods.

Acknowledgement

I am grateful to Peter Antal for many valuable discussions relating to this topic.

References

[1] J. Y. Halpern, "An analysis of first-order logics of probability," *Artificial Intelligence*, 46:311–350, 1990.

[2] M. Wellman, J. Breese, and R. Goldman, "From knowledge bases to decision models," *The Knowledge Engineering Review*, 7(1):35–53, 1992.

[3] M. Jaeger, "Relational bayesian networks," *Proc. of the 13th Conference on Uncertainty in Artificial Intelligence (UAI-1997)*, 1997.

[4] B. Milch, B. Marthi, and S. Russell, "Blog: Relational modeling with unknown objects," in *ICML 2004 Workshop on Statistical Relational Learning and Its Connections to Other Fields*, 2004.

[5] S. Muggleton, "Stochastic logic programs," in *Proceedings of the 5th International Workshop on Inductive Logic Programming*, L. De Raedt, Ed., pp. 254–264. Department of Computer Science, Katholieke Universiteit Leuven, 1995.

[6] K. Kersting and L. D. Raedt, "Bayesian logic programs," in *Proceedings of the Work-in-Progress Track at the 10th International Conference on Inductive Logic Programming*, J. Cussens and A. Frisch, Eds., pp. 138–155, 2000.

[7] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, 62:107–136, 2006.

[8] M. Jaeger, "Complex probabilistic modeling with recursive relational bayesian networks," *Annals of Mathematics and Artificial Intelligence*, pp. 297–308, 2002.

[9] A. Pfeffer and D. Koller, "Semantics and inference for recursive probability models," in *Proc. of the 17th National Conference on Artificial Intelligence (AAAI), Madison, Wisconsin*, 2000.

ISSUES OF A BAYESIAN NETWORK-BASED FIRST-ORDER PROBABILISTIC KNOWLEDGE REPRESENTATION LANGUAGE

András MILLINGHOFFER
Advisors: Péter ANTAL, Tadeusz DOBROWIECKI

I. Introduction

In the area of knowledge representation one of the most widely spread systems is that of Bayesian networks. Despite their advantageous properties shown in the handling of informal, subjective knowledge and uncertainties, they have some drawbacks which prevent them from being applicable in larger domains. The currently used monolithic Bayesian networks do not support the reuse of previously acquired knowledge (i.e. already constructed model parts) and are unable to perform inference about multiple objects simultaneously; briefly, they lack object-orientedness and first-orderedness. Recently there has been many attempts to overcome these disadvantages, in this paper we overview the most important ones, and motivated by them we summarize the desired properties of a representation language and introduce a framework for handling the instances of the above mentioned models.

The paper is organized as follows: in Section II. we overview the framework of Bayesian networks and the underlying principles of Bayesian statistics, in Section III. the main trends of extensions to Bayesian networks are enumerated. In Section IV. we introduce the elements of the implemented system, and in Section V. we sketch the planned direction of further developments.

II. The Bayesian framework

A. Bayesian statistics

Uncertainty is an inherent property of knowledge acquisition processes: it may result from many sources, e.g. the applied method itself, the way how data is gathered or the lack or ignorance of knowledge. In the framework of Bayesian statistics these uncertainties are described through probabilistic quantities, accepting their subjectivist interpretation, considering them measures of prior beliefs about the domain, contrary to the frequentist interpretation which is more prevalent in engineering practice. In Bayesian statistics, observation data is considered to be generated by an ensemble of models parameterized by stochastic variables. In practice, like in the case of Bayesian networks, this parametrization is often structured hierarchically.

The basic task in Bayesian statistics is to calculate the posterior probability of an event conditional on the prior knowledge and observations. In case of *predictive inference*, we are interested in the probability of a generic event, the probability of which can be calculated by a sum or integral over the possible models (in the below example discrete model structures with continuous parameters):

$$P(x|D) = \sum_k P(G_k|D) \int P(x|\theta_k, G_k) P(\theta_k|G_k, D) d\theta_k \tag{1}$$

Parametric inference is the task of computing the probability of a given model, the corresponding equation can be derived from Bayes' theorem:

$$P(\theta, G|D) = \frac{P(D|\theta, G)P(\theta, G)}{P(D)} \tag{2}$$

$$P(G|D) = \frac{\int P(D|\theta, G)P(\theta, G)d\theta}{P(D)} \tag{3}$$

Since the above equations are intractable in practice, inference is performed using Monte Carlo methods, taking into account only some representative model instances, or, as an extremum, only the most probable one.

The main advantages of Bayesianism over classical statistics can be summarized as follows:

- Information about the parameters are described as a probabilistic distribution over them, hence every statistical inference is a direct probabilistic statement.
- Parameter estimation can be regarded as an inversion task formalized by Bayes' theorem, i.e. inference is based purely on data.
- Prior distribution are capable of representing any kind of knowledge or even the total lack of it.
- Posteriors can be used as priors in the consecutive steps of knowledge updating, and can be considered the representations of the phases of knowledge acquisition.
- Through Bayes' theorem, Bayesian inference combines information in prior knowledge and observation data in a normative manner.
- Using posterior distributions instead of point estimations as results of inference takes into account not only the most probable configuration.

B. Bayesian networks

The most common implementations of the above principles are Bayesian networks. They represent the modeled domain by a directed acyclic graph (DAG), the nodes of which are (discrete) stochastic variables (the entities of the domain); while the edges can be regarded as direct probabilistic dependencies. Besides the structure of the graph there is a conditional dependency model assigned to each variable, describing how the given node depends on its parents, hence a Bayesian network implements the previously mentioned hierarchic model structure. These local dependency models suggest an important property of Bayesian networks: the represented distribution $P(X_1, ..X_n)$ factorizes w.r.t. the DAG G if

$$P(X_1, ..X_n) = \prod_{i=1}^{n} P(X_i|Parents(X_i)). \tag{4}$$

Hence a Bayesian network can be defined as the minimal DAG obeying the above condition, assuming that each of its nodes represent a domain variable. Though this definition concentrates on the independencies of the domain, in practice, Bayesian networks are regarded as pairs $BN(G, \theta)$ of the DAG structure G, and the set of corresponding parameters θ describing the local dependencies.

III. Extensions of Bayesian networks

There are two main trends aiming at unifying the capabilities of first-order logic and Bayesian networks. The one which starts from the basis of logic tries to extend it with probabilistic statements. As described in [1] the two possible semantics introducing probabilities about logical statements are those based on object-frequency and on possible world. In the KBMC methodology [2] Bayesian networks are constructed from a logical knowledge base to support the current query. Stochastic logic programs [3] and Bayesian logic programs [4] stem from inductive logic programming.

The other basic approach, which is closer to ours, is to introduce object-oriented properties in Bayesian networks. The basic idea of object-oriented Bayesian networks (OOBNs) [5] is to build up larger networks from smaller objects, network fragments. These can be constructed separately from each other, so forming a library of reusable elements. Among the defined classes the usual inheritance and containment relations may exist, what further extends the knowledge engineering advantages of the framework. Models of a given situation can then be built up from these smaller fragments.

Probabilistic relational models (PRMs) [6] are very similar to OOBNs, but apart from the differences between their formal definitions, they further extend OOBNs with structural uncertainties. In case of *reference uncertainty* the members of the relations cannot be exactly identified, only a set of them; *existence uncertainty* means inferring about unknown objects, defined by its relations.

Bayesian logic (BLOG) [7] models also resemble the structure of OOBNs and PRMs, although they are defined from the basis of predicate logic. They complete the types of uncertainties with *identity uncertainty*: in the case of this, the objects of the domain do not have a unique identifier attribute, so they can be distinguished from each other only probabilistically.

A. *Desired properties of a knowledge representation language*

Motivated by the above trends we can now enumerate the desired properties of a language describing a knowledge base about a domain, which successfully extend monolithic Bayesian networks towards first-orderedness.

- The possible worlds of the language are the model structures and their parameterizations, that is, only the domain-related elements are probabilistic, the basic pieces of knowledge, like logical axioms and functions are not.
- A distribution over them: according to the principles of Bayesian statistics, any kind of domain knowledge is represented by a distribution over the corresponding domain elements.
- Textual annotations assigned to model elements: these may contain important informal knowledge, and tractable first-orientedness can be achieved by means of these.
- The query language, beyond expressing basic queries about the instantiations of the model variables, must be supplemented with string manipulation functions, in order to make it capable of handling queries which are first-ordered in the referred annotations.
- Evidently, the system must have effective algorithms for inference, and as an application of generic inference methods for the calculation of posteriors over parameters and structure, i.e. for learning.

Furthermore, the model structures and parameterizations must follow the paradigm of object-orientedness: they must support a hierarchic, decomposed description of network modules and inheritance between them. The description also has to be able to handle the structural uncertainties described in Section III..

IV. The implemented framework

A. *Elements of the representation language*

The system implemented so far mainly concentrates on the object-oriented properties of the model. The description language consists of the following elements:

- type: simply speaking, defines the range of the corresponding variable, their task is to ensure that only compatible variables will be connected in the model.
- CPD (conditional probability distribution): defines how a child variable depends on its parents, can be regarded as a probabilistic function mapping the current values of the parental variables, to the distribution of the child. It can be a table containing the conditional probability values in the simplest case, or it can be some more complex function e.g. a decision tree.
- TDM (typed dependency map): similarly to CPDs, describes a probabilistic function from some input types to some output types, however, it has an inner structure, that is, it may consist of multiple CPDs and TDMs, defining a Bayesian network module or class.

Of the above elements, TDMs are the key of modularity: by the inclusion of further TDMs they support a hierarchically decomposed structure description.

B. *The implemented system*

The framework currently consists by two parts: a graphical user interface for editing and visualizing models, developed within the confines of the project Pythia, and an inference engine capable of loading models stored in XML files, and performing inference in them, implemented by the author.

V. Prior definition using graph-grammars

An important factor of the inference equation Eq. 2 is the term $P(G, \theta)$, encoding the prior knowledge and beliefs about the domain. In real-world applications it would be intractable to define this distribution by an exhaustive enumeration of the possible structures, however, the language based on TDMs offers an easy and tractable method for this task. Since the model structure hierarchically decomposed by the TDMs naturally resembles a top-down description, it is a straightforward to define a graph grammar that generates the set of possible structures.

Such a grammar would consist of a set of rewrite rules of the form:

$$C_i \rightarrow A_{i1}(P_{i1})|...|A_{in}(P_{in}), \tag{5}$$

where C_i, the condition part consists of a nonterminal subgraph and the corresponding annotation, which, if matched by a part of the current graph may be replaced by one of A_{ij}-s, with probability P_{ij}. Since the matching of the model parts is based on the textual annotations, this grammar-based model generation procedure may be extended to other elements of the model, e.g. the conditional dependency models. In this way the (possibly unnormalized) prior probability of a model can be defined as the product of the probabilities of the rules applied in its derivation.

For the sake of simplicity, the grammar should be retained at a context free level, so that the grammar-related computations remain tractable, and conflicts between overlapping rule conditions can be avoided.

The grammar-related part of the system has to support the following types of queries:

- parsing: deciding whether a given structure can be derived by the grammar, and the probability of it.
- membership: deciding whether a symbol occurred in the parsing tree of the model.
- parameter learning: finding the set of probabilities assigned to the rules, which maximize the posterior probabilities of the models.

VI. Conclusion

In the paper we have overviewed the recent trends aiming at extending monolithic Bayesian models towards first-order logic. Motivated by these we could enumerate the necessary properties of a language by which we can describe a knowledge base containing uncertain knowledge. We have introduced the basic elements of the language and sketched the future steps of its development.

References

[1] J. Y. Halpern, "An analysis of first-order logics of probability," in *Proceedings of IJCAI-89, 11th International Joint Conference on Artificial Intelligence*, pp. 1375–1381, Detroit, US, 1989.

[2] M. P. W. nad J. S. Breese and R. P. Goldman, "From knowledge bases to decision models," *The Knowledge Engineering Review*, 7(1):35–53, 1992.

[3] S. Muggleton, "Stochastic logic programs," in *Proceedings of the 5th International Workshop on Inductive Logic Programming*, L. De Raedt, Ed., p. 29. Department of Computer Science, Katholieke Universiteit Leuven, 1995.

[4] K. Kersting and L. D. Raedt, "Bayesian logic programs," in *Proceedings of the Work-in-Progress Track at the 10th International Conference on Inductive Logic Programming*, J. Cussens and A. Frisch, Eds., pp. 138–155, 2000.

[5] D. Koller and A. Pfeffer, "Object-oriented bayesian networks," in *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97)*, pp. 302–313, 1997.

[6] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *IJCAI*, pp. 1300–1309, 1999.

[7] B. Milch, B. Marthi, and S. Russell, "BLOG: Relational modeling with unknown objects," in *Proceedings of the ICML-04 Workshop on Statistical Relational Learning*, 2004.

LARGE MARGIN METHODS FOR CONSTRUCTING POLYTOPE CLASSIFIERS

István PILÁSZY
Advisor: Tadeusz DOBROWIECKI

I. Introduction

There are many tasks that require the prediction of missing data. Some of them are text document categorization, spam filtering, handwritten digit recognition, credit screening, weather forecasting, etc. The missing data for these tasks are the labels of the documents, classification as spam or not spam, the value of digit, decision to grant or not to grant credit, or the weather for the next day.

Many methods have been developed to yield faster or more accurate solutions to these problems [1].

In this paper we propose a new machine learning (ML) algoritm for binary classification tasks. The algorithm is a modification of a multiclass classification method, suggested in [2]. We will show some preliminary results, based on artificial datasets.

II. Machine learning methods

To be able to compare our new method to the existing ones, we need some classification of classification algorithms. *Supervised machine learning* is the learning of a mapping from input-output patterns. [3] We distinguish between binary classification and multiclass classification. In case of binary classification, we choose between two classes, the positive (e.g. spam) and the negative (e.g. not spam), as opposed to multiclass classification, when we have more classes. In both cases each example belongs to exactly one class.

One of the most important and simplest classifiers are the linear classifiers. They require the examples to be represented in $\mathbb{R}^n$, and use the following decision function:

$$y = \operatorname{sign}(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b) \quad \mathbf{w} \in \mathbb{R}^n,\ \mathbf{x} \in \mathbb{R}^n,\ b \in \mathbb{R} \tag{1}$$

The $\mathbf{w}$ vector and the b scalar are the parameters of the model. There are a plenty of methods to find a suitable $(\mathbf{w}, b)$ solution: Support Vector Machines, Centroid, Adaline, Perceptron, Naive Bayes, Logistic Regression, etc. [4][3][1].

III. Support Vector Machines

During the last few years Support Vector Machines (SVMs) have gained a lot of popularity, and have been proven as one of the most powerful learning algorithms for various problems. Suppose we are provided with examples $\mathbf{x}_i$ and their labeling $y_i \in \{-1, 1\}$. The H hyperplane separating positive and negative examples is formulated by:

$$H = \{\mathbf{x} \mid \mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0 \quad \mathbf{x} \in \mathbb{R}^n\} \tag{2}$$

In case of an SVM, $\mathbf{w}$ and b are the solutions of the following convex quadratic programming (QP) problem:

$$\text{minimize: } V(\mathbf{w}, b) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_i \xi_i \quad \text{subject to: } (\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b)\cdot y_i \geq 1 - \xi_i,\ \ \xi_i \geq 0 \tag{3}$$

Where C is a user-defined constant, which should be set to trade off between large margin ($1/||\mathbf{w}||$) and small training error ($\sum_i \xi_i$). Here ξ_i is a slack variable representing the training error on the ith example.

The dual of this QP is the following optimization problem [4]:

$$\text{maximize: } W(\alpha) = -\frac{1}{2}\sum\sum \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j \tag{4}$$

$$\text{subject to: } 0 \leq \alpha_i \leq C, \ \sum \alpha_i y_i = 0 \tag{5}$$

The optimal $\mathbf{w}$ is equal to $\sum \alpha_i \mathbf{x}_i y_i$ [4]. Those vectors, for which $\alpha_i > 0$, are called support vectors. Only these vectors influence the model.

IV. Multiclass SVMs

There are many approaches to solve multiclass problems. Almost all of them rely on the binary classification algorithms. In [2] a different approach is proposed:

Let $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_m, y_m)\}$ be the set of training examples, and $Y = \{1, ..., k\}$ be the class labels ($y_i \in Y$) and $M = \{(1, \mathbf{w}_1, b_1), ..., (j, \mathbf{w}_j, b_j), \ldots, (k, \mathbf{w}_k, b_k)\}$ be the multiclass model. For a given example $\mathbf{x}$, the decision function is defined as:

$$f(\mathbf{x}) = \arg\max_{j \in Y} \mathbf{w}_j^{\mathrm{T}} \mathbf{x} + b_j \tag{6}$$

The M model is the solution of the following QP-problem:

$$\text{minimize: } V(M) = \sum_{j \in Y} \frac{1}{2} \mathbf{w}_j^{\mathrm{T}} \mathbf{w}_j + C \sum_{i=1}^{m} \xi_i \tag{7}$$

$$\text{subject to: } \forall_{i=1}^{m} \forall_{j \in Y \setminus y_i} \quad (\mathbf{w}_{y_i}^{\mathrm{T}} \mathbf{x}_i + b_{y_i}) - (\mathbf{w}_j^{\mathrm{T}} \mathbf{x}_i + b_j) \geq 1 - \xi_i, \ \ \xi_i \geq 0 \tag{8}$$

V. Polytope classifiers

Convex polytope is the convex hull of a finite set of points [5]. Convex polytopes can be represented as the intersection of a finite number of half-spaces. Note that in case of SVMs SVM partitioned the space into two halves: one for positive examples, the other for negatives. Consider the example depicted in fig.1a. This problem cannot be solved with a linear classifier. Nevertheless SVM has a non-linear extension, which is able to solve non-linearly separable problems using the following function:

$$f(\mathbf{x}) = \text{sign}\left(b + \sum \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i)\right) \tag{9}$$

Where $K(\cdot, \cdot)$ is a user-supplied kernel-function, which must hold some properties [4]. Roughly speaking, the more similar the $\mathbf{x}$ and $\mathbf{x}_i$, the higher the value of $K(\mathbf{x}, \mathbf{x}_i)$. However, when we are given a lots of training examples, to decide a category of an unseen example, the method needs to calculate $K(\mathbf{x}, \mathbf{x}_i)$ for all i, and this could be slow, compared to the linear approach (in that case, $K(\cdot, \cdot)$ is the scalar product, and the support vectors can be summed up).

To overcome this problem, we may use *convex polytope classifiers*: given a convex polytope, it classifies points inside the polytope as negative (inner class), otherwise positive (outer class).

In [6] there are some negative results for constructing classifiers: „It is NP-complete to recognize whether two sets of points in general space can be separated by two hyperplanes." But „For every

fixed k in any fixed dimension, it takes polynomial time to recognize whether two sets of points can be separated with k hyperplanes." Thus in arbitrary high dimensions we can give up fixing k for a small (minimal) value. Note that if a binary classification problem can be solved with a convex polytope classifier, it is very easy to construct one:

- for each $\mathbf{x}_i$ point in the outer (positive) class, separate it from the inner class. This yields $\mathbf{w}_i$,b_i.
- for an unseen point $\mathbf{x}$, compute $\mathrm{sign}(\max_i \mathbf{w}_i^{\mathrm{T}}\mathbf{x} + b_i)$, which indicates the class label.

VI. Proposed method

The multiclass SVM described in section IV. can easily be adopted to construct a binary convex polytope classifier: We create a separate class for each positive example (but do nothing on negatives): we ignore the constraint of eq.8 between positive classes. Thus, each positive example will be separated from each negative (if feasible), but there may be mistakes between positive class labels (but we do not take care of it).

Formally: suppose we are given with a binary classification task ($Y = \{-1, 1\}$) and the outer class is the positive, and contains m^+ examples. If not, then negate the class labels. Let the first m^+ example be positive, and the rest negative. Let $Y = \{-1\} \cup \{1, \ldots, m^+\}$, and $\forall_{i=1}^{m^+}\ y_i := i$. The M model is the solution of the following QP-problem:

$$\text{minimize: } V(M) = \sum_{j \in Y} \frac{1}{2}\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j + C\sum_{i=1}^{m}\xi_i \tag{10}$$

$$\text{subject to: } \forall_{i=1}^{m^+} \quad (\mathbf{w}_{y_i}^{\mathrm{T}}\mathbf{x}_i + b_{y_i}) - (\mathbf{w}_{-1}^{\mathrm{T}}\mathbf{x}_i + b_{-1}) \geq 1 - \xi_i, \tag{11}$$

$$\forall_{i=(1+m^+)}^{m}\forall_{j=1}^{m^+} \quad (\mathbf{w}_{y_i}^{\mathrm{T}}\mathbf{x}_i + b_{y_i}) - (\mathbf{w}_j^{\mathrm{T}}\mathbf{x}_i + b_j) \geq 1 - \xi_i, \ \ \forall_{i=1}^{m}\xi_i \geq 0 \tag{12}$$

The classification of an unseen example is performed by: $f(\mathbf{x}) = \mathrm{sign}\left(\arg\max_{j \in Y} \mathbf{w}_j^T\mathbf{x} + b_j\right)$.

Note that the optimal solution can be found in polynomial time, and the proposed method is a large margin method.

VII. Preliminary results

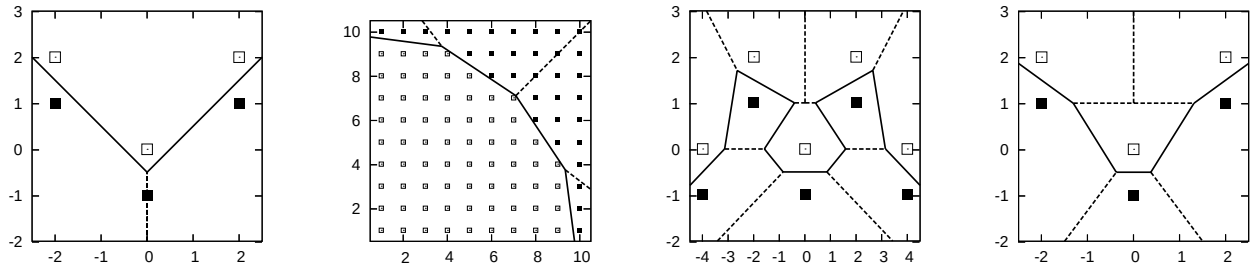
To demonstrate the capabilities of the proposed method, we have evaluated it on some two-dimensional artificial datasets.

In fig.1a there is a very simple classification problem. Linear SVMs are unable to solve it. The proposed approach can solve it ($C = 10$, see fig.1a). With a Gaussian kernel ($K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma||\mathbf{x}_1 - \mathbf{x}_2||^2)$) SVM finds a similar (but smooth) solution. In fig.1b negative examples are inside a circle, positives are outside (100 examples total, $C = 10000$). Our method solves it with 4 weight vectors (plus bias), SVM with a gaussian kernel uses 93 support vectors and one bias. This means a factor of 7-8 in speed and memory consumption when classifying an unseen example. We can generalize the proposed approach: we create a separate class even for each negative example (fig.1c-d, $C = 100$).

VIII. Conclusions

Advantages of the proposed algorithm:

- It can easily be generalized to solve even more general problems: if we create separate classes for each example (not only for positives), it is even more general (i.e. not only convex polytopes). Multiclass (not only binary) polytope classification tasks can also be solved.
- The classification phase is fast, the learning method is deterministic, the optimum point always exists, and can be found in polynomial time.



a) b) c) d)

Figure 1: Classification problems and their solutions. Dashed line means inner-class separation

- It solves the convex separability problem with a large-margin method.
- The solution is ,,intelligent", i.e. it can merge positive classes, and uses a few number of facets.

Disadvantages:

- It is terribly slow when the outer class contains a lots of points.
- We do not know in advance, how many facets are required (we could not know). We have no control over the number of facets.
- Maximizing the margin has no mathematical background as opposed to SVMs (i.e. minimizes the expected classification error).

IX. Future work

The results are very promising, however we need to do further investigations:

- We should try out the kernelized version of multiclass classification kernel functions, to create even more general classifiers (possibly with few support vectors).
- Speed up the proposed method by simplifying the QP-problem: Some clustering on the outer class would speed up the method because of the fewer classes. We could use a small C and solve the QP problem. From this, we hope to have a small number of positive classes. We could consider only these classes as the covering of the positive set, and solving this new problem (fewer positive classes) with a higher C. Another approach is to cut down with some hyperplane as many outer points as we can, create a cluster for them, and repeating this greedy method until each point is in a cluster.
- There are more open-source implementations of the multiclass SVM, with lots of speed-up tricks. Can they be modified to construct polytope classifiers while preserving their speeds?
- If we put some labeled points in the space, and assign each point of the space to the nearest of these points, we get a nearest neighbour classifier. The relationship between k-NN and polytope classifiers should be investigated.

References

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*, Wiley-Interscience, 2000.

[2] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, 2:265–292, Dec. 2001.

[3] S. Russell and P. Norvig, *Artifical Intelligence: A Modern Approach*, Prentice-Hall, Englewood Cliffs, NJ, second edition, 2002.

[4] H. Gábor, "Support vektor gépek (support vector machines)," 2002,
URL: http://www.mit.bme.hu/˜horvath/supportvector.pdf.

[5] H. S. M. Coxeter, *Introduction to Geometry*, Wiley, New York, second edition, 1969.

[6] N. Megiddo, "On the complexity of polyhedral separability," *Discrete and Computational Geometry*, 3:325–337, 1988.

AN EFFICIENT ALGORITHM FOR DECIDING THE CONVEX SEPARABILITY OF POINT SETS

Gábor TAKÁCS
Advisor: Béla PATAKI

I. Introduction

One of the main goals of *machine learning* is to provide tools for building *intelligent systems*, that gain knowledge, make predictions or make decisions based on a set of data. The heart of such a system is often a *classifier* (e.g. in spam filtering, computer-aided breast cancer recognition, postal code recognition, etc.). The classifier has to guess the value of a high-level discrete attribute (e.g. healthy or sick, spam or nonspam) from the values of some observable, low-level features (e.g. pixel intensities of an X-ray image, words of a text file).

If all the d low-level features are represented by real numbers, then the observations can be treated as points in the d-dimensional Euclidean space. Observations belonging to the same class form a set of points. An important step of building a classification system for a real-life problem is exploring the data. A typical question is whether the classes are *linearly separable* or not. We will see that it can be decided in polynomial time, and there are good practical algorithms performing it.

Another interesting question is whether the classes are *convexly separable* from each other or not. In theory it can be decided in polynomial time too, but conventional methods fail for large tasks in practice due to running time problems. In this paper we will propose an efficient algorithm for deciding the convex separability of point sets. In contrast to the other algorithms available, our method is scalable even for very large problems. We will compare the running time of our algorithm with conventional, linear programming based methods.

II. Notation, Definitions

Let $\mathcal{S}$ be a set of points in the d-dimensional Euclidean space. The *convex hull* of $\mathcal{S}$ denoted by conv($\mathcal{S}$) is the minimal convex set containing $\mathcal{S}$. If $\mathcal{S}$ is finite, then conv($\mathcal{S}$) is a convex polyhedron. A convex polyhedron in d dimensions can be given either by its vertices or its $(d-1)$-dimensional facets. The former is called *vertex representation*, the latter is called *halfspace representation*. The halfspace representation typically cannot be computed in high dimensions, because the number of $(d-1)$-dimensional facets grows exponentially with d.

Let $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ and $\mathcal{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m\}$ be two sets of points in $\mathbb{R}^d$. The convex hull of $\mathcal{P}$ and $\mathcal{Q}$ is denoted by conv($\mathcal{P}$) and conv($\mathcal{Q}$).

Definition 1. $\mathcal{P}$ and $\mathcal{Q}$ are called *linearly separable* if $\mathrm{conv}(\mathcal{P}) \cap \mathrm{conv}(\mathcal{Q}) = \emptyset$. Or equivalently: $\exists \mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R} : \mathbf{w}^T\mathbf{p}_i + b < 0$ for $i = 1, 2, \ldots, n$ and $\mathbf{w}^T\mathbf{q}_j + b > 0$ for $j = 1, 2, \ldots, m$.

Definition 2. $\mathcal{P}$ and $\mathcal{Q}$ are called *convexly separable* if $\mathrm{conv}(\mathcal{P}) \cap \mathcal{Q} = \emptyset$ or $\mathcal{P} \cap \mathrm{conv}(\mathcal{Q}) = \emptyset$. If $\mathrm{conv}(\mathcal{P}) \cap \mathcal{Q}$ and $\mathcal{P} \cap \mathrm{conv}(\mathcal{Q})$ are both empty, then $\mathcal{P}$ and $\mathcal{Q}$ are *mutually convexly separable*. If $\mathrm{conv}(\mathcal{P}) \cap \mathcal{Q}$ is empty, but $\mathcal{P} \cap \mathrm{conv}(\mathcal{Q})$ is not, then $\mathcal{P}$ is called the *inner set* and $\mathcal{Q}$ the *outer set*.

Note that linear separability implies mutual convex separability, but the reverse is not true.

III. Conventional Methods

A. Linear Separability

Linear separability can be formulated as a *linear programming* (LP) problem as the following:[1]

$$\begin{array}{rcl} \text{variables} & : & \mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R} \\ \text{minimize} & : & 1 \\ \text{subject to} & : & \mathbf{w}^T\mathbf{p}_i + b > 0, \quad i = 1, 2, \ldots, n, \\ & & \mathbf{w}^T\mathbf{q}_j + b < 0, \quad j = 1, 2, \ldots, m. \end{array} \tag{1}$$

$\mathcal{P}$ and $\mathcal{Q}$ are linearly separable if and only if the constraints can be satisfied. Since there exist polynomial algorithms for solving LP ([1, 2]), linear separability can be decided in polynomial time. Problem (1) has no objective function and such a problem is often called a *linear feasibility problem*. Although it might look a simpler problem to solve, it is polynomially equivalent to the general LP to solve it. Unfortunately, this formulation is often unsuitable in practice, because general LP solvers (e.g. MATLAB's) usually fail on large tasks (say $d > 100$ and $n = m > 1000$) due to running time problems.

Fortunately, there is another polynomial formulation, that can handle much larger problem sizes in practice. Linear separability can also be decided with a linear *support vector machine* (SVM) [3]:

$$\begin{array}{rcl} \text{variables} & : & \mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R} \\ \text{minimize} & : & \mathbf{w}^T\mathbf{w} \\ \text{subject to} & : & \mathbf{w}^T\mathbf{p}_i + b \geq 1, \quad i = 1, 2, \ldots, n, \\ & & \mathbf{w}^T\mathbf{q}_j + b \leq 1, \quad j = 1, 2, \ldots, m. \end{array} \tag{2}$$

$\mathcal{P}$ and $\mathcal{Q}$ are linearly separable if and only if Problem (2) has a solution. This quadratic programming (QP) problem seems harder than the previous LP one, but for this special type of constrained optimization there exist very efficient solvers (e.g. libsvm [4], svm-light [5]).

B. Convex Separability

$\mathcal{P}$ and $\mathcal{Q}$ are convexly separable if $\mathrm{conv}(\mathcal{P}) \cap \mathcal{Q} = \emptyset$ or $\mathcal{P} \cap \mathrm{conv}(\mathcal{Q}) = \emptyset$. Because the two tests are symmetric, assume that we want to decide whether the intersection of $\mathrm{conv}(\mathcal{P})$ and $\mathcal{Q}$ is empty or not. At first we give a simple but exponential method that works well only for small problems:

1. Compute a halfspace representation of $\mathrm{conv}(\mathcal{P})$.
   This yields $\mathbf{w}_k$ -s and b_k -s $(k = 1, 2, \ldots, r)$ such that:
   $\mathrm{conv}(\mathcal{P}) = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{w}_1^T\mathbf{x} \geq b_1,\ \mathbf{w}_2^T\mathbf{x} \geq b_2,\ \ldots,\ \mathbf{w}_r^T\mathbf{x} \geq b_r\}$. (3)
2. $\mathrm{conv}(\mathcal{P}) \cap \mathcal{Q} = \emptyset$ if and only if $\max\limits_{j=1,\ldots,m} \{\min\limits_{k=1,\ldots,r} \{\mathbf{w}_k^T\mathbf{q}_j - b_k\}\} < 0$.

The second method is based on LP:

$$\begin{array}{rcl} \text{variables} & : & \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m \in \mathbb{R}^d,\ b_1, b_2, \ldots, b_m \in \mathbb{R} \\ \text{minimize} & : & 1 \\ \text{subject to} & : & \mathbf{w}_j^T\mathbf{p}_i + b_j > 0, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m, \\ & & \mathbf{w}_j^T\mathbf{q}_j + b_j < 0, \quad j = 1, 2, \ldots, m. \end{array} \tag{4}$$

[1] Strictly speaking this is not an LP, because of the $<$ and $>$ signs. In practice $\leq$ and $\geq$ are used instead, and zeros are replaced to small nonzero numbers.

The necessary and sufficient condition of $\mathrm{conv}(\mathcal{P}) \cap \mathcal{Q} = \emptyset$ is the existence of a solution. This method is polynomial in time, but fails for large problems in practice.

The third method is an alternative LP formulation. The LP problem for any fixed $j \in \{1, 2, \ldots, m\}$ is the following:

$$\begin{array}{rcl} \text{variables} & : & \alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jn} \in \mathbb{R} \\ \text{minimize} & : & 1 \\ \text{subject to} & : & \mathbf{q}_j = \alpha_{j1}\mathbf{p}_1 + \alpha_{j2}\mathbf{p}_2 + \cdots + \alpha_{jn}\mathbf{p}_n, \\ & & \alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jn} \geq 0, \quad \alpha_{j1} + \alpha_{j2} + \cdots + \alpha_{jn} = 1. \end{array} \tag{5}$$

This formulation exploits the fact that using a vertex representation $\mathrm{conv}(\mathcal{P})$ can be written as $\{\mathbf{x} \in \mathbb{R}^d : \mathbf{x} = \sum_{k=0}^{n} \beta_k \mathbf{p}_k,\ \beta_1, \beta_2, \ldots, \beta_n \geq 0,\ \sum_{k=0}^{n} \beta_k = 1\}$. $\mathrm{conv}(\mathcal{P}) \cap \mathcal{Q} = \emptyset$ if and only if Problem (5) is infeasible for any $j \in \{1, 2, \ldots, m\}$.

IV. The Proposed Method

At first we introduce a fast algorithm that performs approximate convex separation:

1. Initialize signed distances with $s_1 = s_2 = \cdots = s_m = \infty$.
2. Compute the centroid of $\mathcal{P}$ as $\bar{\mathbf{p}} = \frac{1}{n}(\mathbf{p}_1 + \mathbf{p}_2 + \cdots + \mathbf{p}_n)$.
3. Choose $\mathbf{q}_k$ from $\mathcal{Q}$ such that $k = \arg\max_{j=1,\ldots,m}\{s_j\}$.
4. If $s_k \leq 0$, then stop.
5. Compute $\mathbf{w} = (\bar{\mathbf{p}} - \mathbf{q}_k)/\|\bar{\mathbf{p}} - \mathbf{q}_k\|$ and $b = \min_{i=1\ldots n}\{\mathbf{w}^T\mathbf{p}_i\}$.
6. $s_k \leftarrow 0$.
7. For all $s_j > 0 : s_j \leftarrow \min\{s_j,\ \mathbf{w}^T\mathbf{q}_j - b\}$.
8. Go to step 3.

(6)

This algorithm translates at most m hyperplanes to their "optimal" positions. (The number of q_j-s on negative side of the hyperplane is maximized while all p_i-s have to be on the positive side.) The algorithm can be called the *centroid method*, because the hyperplanes are defined by connecting the centroid of $\mathcal{P}$ with the elements of $\mathcal{Q}$. This extremely simple and fast method is often able to separate most of the elements of $\mathcal{Q}$ from $\mathrm{conv}(\mathcal{P})$. The disadvantage of the algorithm is that it does not guarantee to find a convex separation for convexly separable problems.

At second we straightforwardly extend the SVM method to the case of convex separability:

$$\begin{array}{rcl} \text{variables} & : & \mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w}_m \in \mathbb{R}^d,\ b_1, b_2, \ldots b_m \in \mathbb{R} \\ \text{minimize} & : & \mathbf{w}_1^T\mathbf{w}_1,\ \mathbf{w}_2^T\mathbf{w}_2,\ \ldots,\ \mathbf{w}_m^T\mathbf{w}_m \\ \text{subject to} & : & \mathbf{w}_j^T\mathbf{p}_i + b_j \geq 1, \quad i = 1, 2, \ldots n, \quad j = 1, 2, \ldots m, \\ & & \mathbf{w}_j^T\mathbf{q}_j + b_j \leq -1, \quad j = 1, 2, \ldots m. \end{array} \tag{7}$$

$\mathrm{conv}(\mathcal{P}) \cap \mathcal{Q} = \emptyset$ if and only if the above m constrained optimization problems all have a solution. The disadvantage of this method is enormous running time for large m. (However it is faster than the conventional LP-based method.)

A very efficient algorithm can be obtained by combining the previous two methods. At first the size of $\mathcal{Q}$ is reduced by applying the centroid method. Denote the set of unseparated $\mathbf{q}_j$-s by $\mathcal{Q}_U$. At second $\mathrm{conv}(\mathcal{P}) \cap \mathcal{Q}_U = \emptyset$ is checked by running the SVM method.

Table 1: Running times. (N/A: no answer in reasonable time.)

	d	n	m	LP1	LP2	SVM	Centroid+SVM
Votes	16	267	168	33.0 s	15.9 s	0.25 s	0.15 s
Wisconsin	9	444	239	63.9 s	23.4 s	0.40 s	0.39 s
MNIST01	196	6903	7877	N/A	N/A	25.3 s	0.86 s
MNIST02	196	6903	6990	N/A	N/A	325.8 s	11.4 s
MNIST79	196	7693	6958	N/A	N/A	919.0 s	145.8 s
MNIST49	196	6824	6958	N/A	N/A	1235.1 s	268.2 s

V. Comparison

We compared the running time of our method with conventional LP-based methods in 6 real-world problems. The first two test problems originate from the UCI machine learning repository [6]. The *Voting Records* dataset includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes. The class attribute is whether the voter is democrat or republican. The *Wisconsin Breast Cancer* dataset contains measurements on 239 malignant tumors and 444 benign findings in women's breast. Both datasets are convexly separable, but not mutually.

The other test problems were extracted from the the *MNIST* handwritten digit recognition database [7]. There is a linearly separable (digit 0 vs. digit 1), a mutually convexly separable (0 vs. 2), a convexly separable (7 vs. 9) and a convexly nonseparable (4 vs. 9) among them.

The algorithms had to return the number of elements in $\mathrm{conv}(\mathcal{P}) \cap \mathcal{Q}$ and $\mathcal{P} \cap \mathrm{conv}(\mathcal{Q})$, because this information is more useful in practice than a plain yes-no answer. Fortunately, all algorithms can easily be modified to return these 2 numbers instead of a boolean value. The algorithms in the comparison were the following:

- LP1: The LP-based formulation given in (4), implemented in MATLAB.
- LP2: The alternative LP-based formulation given in (5), implemented in MATLAB.
- SVM: The SVM-based formulation given in (7), implemented in Java using libsvm [4].
- Centroid+SVM: The proposed algorithm, implemented in Java using libsvm.

The results can be seen in Table 1.

VI. Conclusion

Convex separability is an interesting question in the data exploration phase of building classification systems for real-life problems. In this paper we proposed an efficient algorithm for deciding the convex separability of two point sets in $\mathbb{R}^d$. Our algorithm was more than 50 times faster than conventional LP-based methods in all test problems.

References

[1] L. Khachiyan, "Polynomial algorithm for linear programming," *Dokl. Akad. Nauk SSSR*, 224:1093–1096, 1979.

[2] N. Karkamar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, 4:373–395, 1984.

[3] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[4] C. Chang and C. Lin, *LIBSVM: a library for support vector machines*, National Taiwan University, 2001, Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[5] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. MIT Press, 1999.

[6] D. Newman, S. Hettich, C. Blake, and C. Merz, *UCI repository of machine learning databases*, University of California, Irvine, CA, 1998, Available at http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[7] Y. LeCun, *The MNIST database of handwritten digits*, 2003, Available at http://yann.lecun.com/exdb/mnist.

VERIFICATION OF HARDWARE-SOFTWARE INTEGRATION PROCESSES

András BALOGH
Advisors: András PATARICZA, Dániel VARRÓ

I. Introduction

Model-driven development (MDD) attempts to uniform the IT system development processes. On the other hand, developer needs are strongly separated to support hard- and soft-(or non-)realtime system development. MDD is continously emerging in several domains (like enterprise software development), however it lacks certain aspects, like dependability and other non-functional requirements modeling and validation, needed for realtime and dependable system development.

An MDD framework for embedded systems will be presented in this paper with a special focus on the verification and validation of non-functional properties during the development. The solution introduces an open, extensible framework that can handle a diverse set non-functional properties like timeliness, power consumption, etc. by a proper set of plugins.

II. Problem domain

A. *Distributed, dependable embedded systems*

The project DECOS (Dependable Components and Systems) aims at the development support for safety-critical, distributed embedded systems, with a special focus on automotive, aerospace, and industrial control applications.

These systems consist of several independent components (each having a communication controller, and one or multiple processing units) - called DECOS nodes - interconnected via a core network relying on the time-triggered principle. Other optional networks (called field busses) may interconnect a DECOS node with several field bus nodes (for instance, intelligent sensor, or mechatronical actuators). The field busses can rely on an arbitrary combination of event-triggered and time-triggered protocols.

The main goal of the development is the generation of the following artifacts:

- Communication schedule of the core network,
- Configuration of the operating system and fault-tolerant middleware on DECOS nodes and configuration of communication controllers integrated into the DECOS nodes,
- Generation of the integration interfaces for field bus communication (implemented outside of the framework),
- Automated generation of the individual application job code from their respective high-level models.

B. *Hardware-software integration tool chain*

The DECOS tool chain aims at the complete coverage of the workflow related to model-driven development of systems running on the DECOS platform including modeling, model transformation, and code generation.

The tool chain [1] covers four main aspects of development: specification, design, implementation, and V&V (Verification and Validation). The current paper focuses on the architecture design of the system. Itsupports the modeling of the interfaces and capabilities of the hardware platform in the form of a cluster resource description, and interfaces and interactions of the software components in a platform-independent way.

Our main focus is the implementation of the hardware-software integration design step by architectural design tools. The purpose of this step is the semi-automated generation of a platform-specific model (PSM) from the platform-independent model (PIM) of the applications that contain the architectural description of the software, and the cluster resource description (CRD). This integration and deployment process underlies both functional constraints (the functional specification of the target system) and non-functional ones (temporal, and dependability related constraints). A similar approach for non-functional constraint modeling can be found in [2]. The resulting PSM includes all allocation, scheduling, and configuration information needed for deployment of the software on the hardware, and the process guarantees the fulfillment of both the functional and the non-functional constraints.

III. Taxonomy of contraints

The complexity related to the design of safety-critical systems is well characterized by the fact that even the collection, categorization and formalization of typical constraint classes requires an extensive work. Extensive research (e.g. [3] and [2]) has been done regarding the collection of various design constraints for embedded systems. The categorization introduced in this paper supports the development of library-like constraint sets at general, domain-specific and even at application-specific level. We will categorize constaints based on their type and scope.

A. Constraint type

The type dimension of constraints is divided into two categories:

- *Qualitative constraints* are logical statements that define requirements for several system properties. These constraints are mostly structural ones, and do not refer to attribute values. For instance, a typical safety constraint is "two replicas of a given software component must not run on the same computing node".
- *Quantitative constraints* confine the design space by correlating attributes of the system components. Typical representatives are limitations of the overall system characteristics (temporal, dependability related, power consuption, etc.) expressed as functions of the system architecture and component attribute values.

B. Constraint Scope

The categorization according to the scope follows a similar logic as standardization.

- *Overall dependability-related* costraints are design rules valid for all dependability-related systems. For instance a constraint that has to be fulfilled by all systems relying on robust partitioning.
- *Domain specific* constraints are derived from standards and recommendations of a given application domain (like automotive or aerospace).
- *Application specific* constraints express the pecularities of an individual application.

The DECOS tool chain is an integrated development environment that contains several interactive design steps. To ensure that the constraints are met during the entire process, we have to introduce integrated constraint checking techniques and tools.

IV. Constraint checking

A. Qualitative constraints

Qualitative constraints can be represented using first order logic formulas. Using these formulae, never claim-like negated formulae can be generated that should never be satisfied by the target system (model).

These formulae are mapped into graph patterns that represent the counter-examples of the constraint. If one of the patterns can be matched on the system model, the associated constraint is violated.

A limited subset of constraints can also be handled by a second method that is supported by the ontology-based checking of constraints. This requires the transformation of the system model to the input of the ontology tool [4], and the mapping of constraints to ontology queries. This method has limitations due to the expression power of the background mathematics used in ontologies.

B. Quantitative constraints

Qualitative expressions can be generated using the actual system structure and attribute values. The resulting formulae -usually containing open variables- can then exported to an appropriate mathematical solver that can either find a possible solution for the open attributes, or it can state that the equation system is inconsistent.

The first case means that the constraints are fulfilled, the second case means that some of them are not fulfilled. This check can be done with a simple constraint solver package or using an external mathematical tool.

V. Implementation of the DECOS tool chain

Verification of design constraints is an integral part of the DECOS software-hardware integration tool chain.

A. Basic Integration Framework

The basis of the framework is the Eclipse development platform that is an open framework for tool development and integration. The DECOS tool chain relies also on Eclipse. VIATRA2, a model-transformation tool relying on a simple transformation language[5], is used for model-to-model and model-to-code transformation, and also for graph pattern-based constraints checking.

A single transformation serves to export PIM and PSM models into an ontology tool. Specific queries are created to verify qualitative constraints using the ontology reasoner. Several examples of PSM constraints that can be checked by this approach: each replica of a job must run on a separate computer; a message can be sent only by a single job; each job has to be assigned to a computer, etc.

The formulae reflecting the qualitative constraints and actual parameters are also generated by model transformations and are exported to an external mathematical package (for instance, ILog CPLEX). For example, in case of system availability constraints, the actual availability calcualtion formula depends on the system structure. According to [6] the formula with all the parameters can be dynamically generated and evaluated using the external mathematical tool.

B. Open constraint checking architecture

Adaptation to new application areas by incorporation of domain-specific constraints as validation objectives, moreover the extensability by application-specific constraints all require extension points receiving custom constraint modules. These extension points support at the same time the frequently very sophisticated analysis tools developed for the validation of a particular aspect. An extension point in the Eclipse terminology is a well-defined interface where software components implementing specific tool functions can be plugged in. Several extension points have been defined that can be used for extensions (see Fig. 1):

- *Structural extensions:* Attribute providers defining new, custom attributes for model elements can be defined. This is then used for a) to specify this attribute in the model, b) to be able to refer to this attribute in constraints. The attribute provider is responsible for handling the attribute values for all corresponding model elements.
- *Behavioral extensions:* Custom constraint providers can define new constraints that must be fulfilled by the system. The provider is responsible for the generation, maintanance and evaluation of its constraints. It is also possible to extend the set of constraint solver tools (ontologies, graph patterns, and mathematical packages) that enables the integration of state-of-the art domain-specific analysis tools and packages.

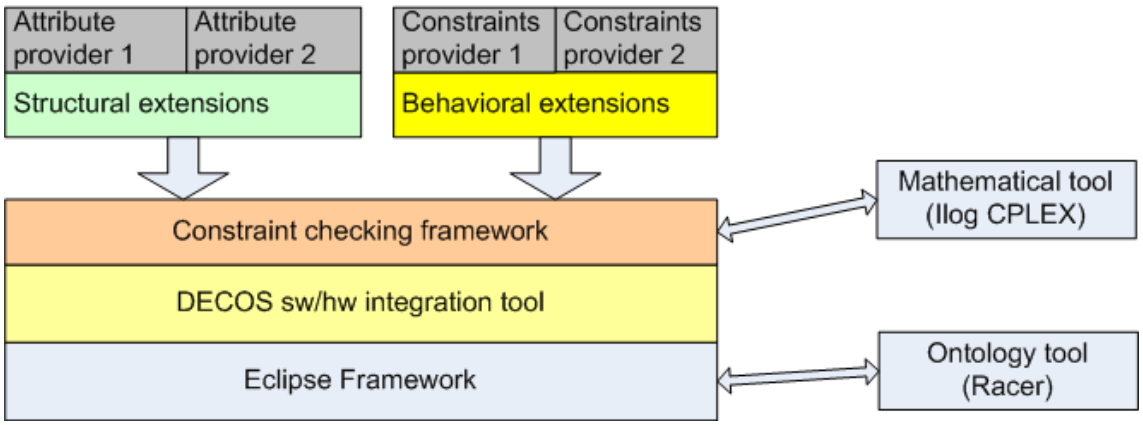


Figure 1: Constraint checking framework

Standards define correlated sets of constraints, which may appear on extension point level, where several types of constraint packages can be defined. (like all the constraints needed for an drive-by-wire application)

VI. Conclusion

The presented development framework for safety-critical, distributed embedded systems incorporates a open, extendable constraints checking subsystem. This tool is able to receive a large variety of system constraints including quantitative and qualitative ones. Our solution can be used during the software-hardware mapping process to ensure the early recognition of design problems and constraint violations and helps improving the productivity.

VII. Acknowledgement

This work was partially supported by EU FW 6 IP DECOS (Dependable Components and Systems).

References

[1] A. B. G. C. W. Herzner, B. Hubert, "The DECOS tool-chain: Model-based development of distibuted embedded safety-critical real-time systems," in *Proc. of the DECOS/ERCIM Workshop on SAFECOMP 2006.*, Gdansk, Poland, September 2006.

[2] G. M. L. L. A. S.-V. J. R. Rong Chen, Marco Sgroi, "Embedded system design using UML and platforms," in *Proceedings of Forum on Specification and Design Languages 2002 (FDL'02)*, September 2002.

[3] K. Ramamritham, "Where do time constraints come from and where do they go," *International Journal of Database Management, Vol. 7, No. 2, pp. 4-10, 1996.*, 7, 1996.

[4] S. V.-G. A. B. G. C. András Pataricza, Balázs Polgár, "Formal checking of metamodels and models," Invited talk at DECOS/ERCIM workshop on SAFECOMP 2006, Gdansk,Poland, September 2006.

[5] A. Balogh and D. Varró, "Advanced model transformation language constructs in the VIATRA2 framework," in *ACM Symposium on Applied Computing — Model Transformation Track (SAC 2006)*, 2006, In press.

[6] A. P. András Balogh and D. Varró, "Model-driven optimization of enterprise application and service deployment," in *Proc. Of the 2nd International Service Availability Symposium (ISAS 2005)*, pp. 84–98, 2005.

A NOVEL ABSTRACTION APPROACH TO MODEL CHECK DISTRIBUTED, FAULT-TOLERANT PROTOCOLS

Péter BOKOR
Advisor: András Pataricza

I. Introduction

We present an *abstraction* method to reduce the state space in *model checking* when verifying *fault-tolerant distributed protocols* meeting specified symmetry assumptions. The main idea of abstraction is that instead of storing the tuple of local states for each node of the system explicitly, we only store the sets (we call them *consistency sets*) of possible states a correct process can have at every round of communication. Representing sets instead of tuples leads to a considerable reduction of the state space, as shown in our experiments.

The abstraction is *sound*, i.e., if the model checker proves a property in the abstract model then it also holds in the non-abstracted one. Furthermore, for protocols where the latest communication (e.g., broadcast messages) between nodes fully determines the current system state (we call such protocols *memoryless*), soundness and *completeness* can be shown, i.e., a property holds if and only if the abstract property holds in the abstract system.

Due to information loss during the abstraction, we restrict ourselves to properties that require all processes to satisfy a certain property (we call such properties *uniform*). We make use of *temporal logic* to define properties that constrain the protocol's behavior in different stages (we call them *rounds*) of execution.

We also present results of experiments that were carried out with the SAL model checker.

II. Specifying Fault-Tolerant, Distributed Protocols

In this section we introduce the specification language (an adaptation from [1]) for the class of protocols we concentrate on, i.e., synchronous, frame-based and symmetric ones.

We consider fault-tolerant, distributed protocols containing n processors (or nodes). We assume that the number of faulty processors is a. The protocol proceeds through rounds, the number of rounds is finite, we denote it by r. At each round, every processor generates a message of the domain D and *broadcasts* to the others. We assume *synchronous* communication, i.e., at the round boundaries every message is supposed to be available. At every round, each processor maintains a local state of the domain S, which, at round 0, is one of the initial states ($S^I \subseteq S$). According to the current state, each node computes the message to send in the next round. This is done through the *deterministic* message generation function $m : S \rightarrow D$. Faults of faulty nodes manifest only through the messages they send. Faulty nodes do not obey the message generation function, they can send arbitrary messages, even *asymmetrically*.

Each processor computes its next state according to the state transition relation $\tau : S \times D^{n-a} \times S$. τ contains tuples of the *current* state of a correct processors, the messages received from other *correct* nodes and the *next* state of the processor. It is important to note that nodes also send messages to themselves. The messages of faulty nodes can affect the next states of the correct processors as follows. τ is a function in a fault-free environment: given the current state and the vector of messages, it contains exactly *one* tuple determining the next state of the node. In case faulty nodes are present, τ captures those state transitions that are affected by the messages sent by faulty nodes. Accordingly, nodes having the same input messages (from correct nodes), even with identical current states, can

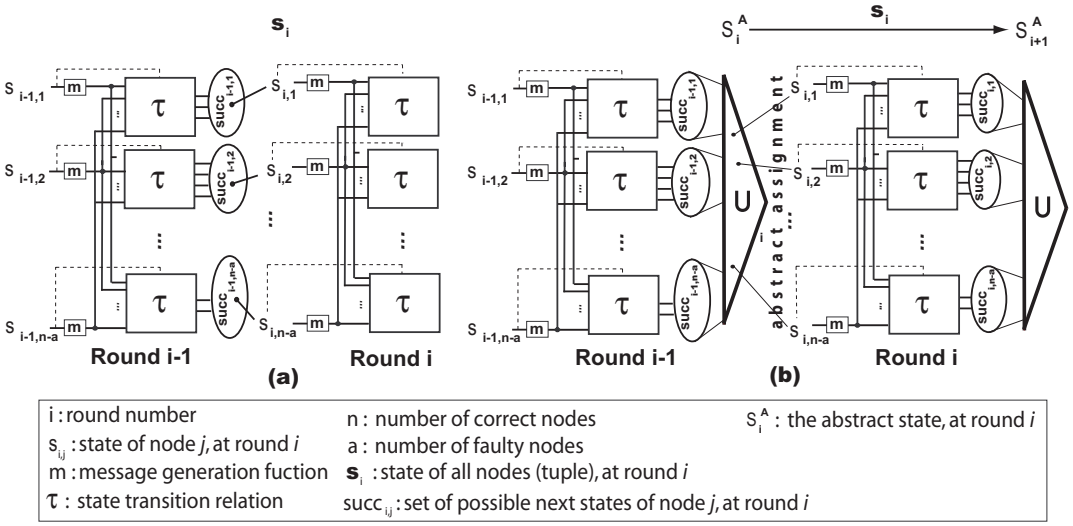


Figure 1: **(a)** Tuple-based mode **(b)**The abstraction with consistency sets

proceed to different states. By definition, every processor maintains the *same* τ. Based on the previous definitions we define a distributed protocol as a tuple: $DP = (S, S^I, D, n, a, r, m, \tau)$.

III. The abstraction principle

The non-abstracted model (we call it *concrete* model) is a straightforward derivation from the description of the protocol, which will turn out to be infeasible to verify properties against systems of larger size and/or in case of complex protocols. We make the following assumptions:

(A1) We assume a *pessimistic (Byzantine) fault model*, i.e., faulty nodes can send asymmetric information to correct nodes, and faults cannot be detected locally.
(A2) Correct nodes are allowed to start with *different initial states* (at round 0).
(A3) Faulty nodes are not modeled.

The main idea is that we model the state of each processor explicitly: a *tuple* of states will be updated at every round, where each element of the tuple stores the current state of the corresponding processor. Due to (A3) only correct nodes are modeled, accordingly, the size of the tuple is $n - a$. Figure 1(a) shows how the concrete model proceeds from round $i - 1$ to round i. The tuple $(s_{i,1}, ..., s_{i,n-a})$ is called the *(global) state* of the concrete model, where $s_{i,j}$ denotes the state of processor j at round i (which we call *local state* of processor j). Initially, every node starts from an initial state, which can be different for each node, in accordance with (A2). Every node broadcasts a message to the others, which is obtained by $m(s_{i,j})$.

If the current state $(s_{i-1,j})$ and the messages sent by correct nodes are available at the node's input, the set of possible next states will be computed: $succ_{i-1,j}$ is called the successor set of processor j, at round $i - 1$, containing all states that processor j can take at round i. These sets can vary from node to node, depending on the current state of the node and the messages sent by the faulty nodes, which are captured by τ. The global state at round i is $\mathbf{s}_i = (s_{i,1}, ..., s_{i,n-a})$, where $s_{i,j}$, the current state of processor j at round i, is taken from $succ_{i,j}$. Note that $s_{i,j}$ can be any of the possible states, since Byzantine nodes can send arbitrary messages, even asymmetrically (A1). Further development of the global state follows iteratively.

A. Consistency sets

In case of fault-tolerant, distributed protocols, the tuple-based model can easily blow the state space to exponential size, due to (1) the variety of possible *faulty behavior*s and (2) the case explosion

induced by the *permutation* of processors. To tackle case explosion we propose an abstraction, which *aggregates* the states that different nodes can have, instead of modeling the state of every correct processor explicitly.

The core idea is to use *consistency set*s, i.e., sets containing all states of *any* correct processors. More precisely, we define the abstract state to be a set containing all possible states that any correct processor can assume, given a certain communication pattern (i.e. the sequence of messages sent among the nodes). In other words, there is no run where two correct processes have two states owned by two different consistency sets. Such a set is considered *consistent* in the sense that it is an appropriate, common domain for all correct processors to take the current state from.

The idea of abstraction is depicted in Figure 1(b). Similarly to the tuple-based scheme, the successor set $succ_{i-1,j}$ is the set of all possible next states for processor j, at round i. In the concrete model the 1st processor proceeds to a particular state in $succ_{i-1,1}$, the 2nd processor to a state in $succ_{i-1,2}$, etc., and these states build a tuple constituting the global state of the next round. In the abstract model we aggregate these successor sets by building the *union* of them. We call this union (which is a set of states) consistency set, and we consider it as the *abstract state* (S_i^A). It is intuitive to see that such a set is consistent, since it contains all states that *any* correct node can assume at the given round.

To proceed from the abstract state S_i^A to S_{i+1}^A, the abstract model builds a tuple of states from S_i^A, which corresponds to a possible global state of the concrete system ($\mathbf{s}_i$). We call this global state $\mathbf{s}_i = (s_{i,1}, ..., s_{i,n-a})$ *abstract assignment*, where $s_{i,j}$ is chosen to be the next state of processor j. According to the states of $\mathbf{s}_i$ the next abstract state (S_{i+1}^A) is computed as the union of the successor sets $succ_{i,j}$. Note that the abstract assignment is an auxiliary construct that represents a possible assignment of states to correct processes. It is used to determine the set of messages produced at the next round. It is important to remark that the abstract assignment is chosen *non-deterministically*, since no assumption is made concerning what messages are sent by the faulty nodes.

Abstract assignments lead to unreachable global states, which is due to the fact that not every state in a consistency set is necessarily valid for a given processor. Such possibly unreachable states affect the completeness of the abstraction, which will be addressed later.

IV. The property space

Our basic model is that processors maintain program *variables* to run the protocol. Due to the assumption that nodes execute the same protocol, the set of variables are identical for every processor. Furthermore, we assume certain operators/relations over the domains of the variables to constrain their values. For example, if a state variable stores integer values, then we assume the definition of the ordinary operators $+, *$, etc. and relations $=, <, \leq$, etc. over integers.

We add a new interpretation to the notion of state. While the concrete model proceeds through the rounds of the protocol, the local state of the processors may differ at every round. Since at each node the state variables encode the distributed protocol DP, a state can be considered as an assignment of values to the variables. For example, one variable could be the program counter, which will be incremented in the next state at every node. A *property* is constructed upon state variables composed together via the pre-defined operators/relations.

In many practical cases, the desired properties define *uniform* requirements, i.e., it is required that *all* of the processors satisfy a certain property. For example, in case of a consensus protocol, agreement requires that every node adopts the same value after the protocol terminates.

Properties constrain the protocol's behavior at a particular round, i.e., in a particular state. Protocols often define requirements to constrain the behavior *over* rounds, i.e., for the sequence of states. For example, validity of consensus defines that if a node is correct then its proposed value (in the initial round) will be adopted by the others (in the final round). We make use of *Linear Temporal Logic* (LTL, originally introduced in [2]) to define such requirements.

The abstract counterpart of a property (we call it *abstract property*) is interpreted over consistency sets. We assume that the property is uniform and we define that an abstract property is true in an abstract state if all states in the consistency set satisfy the property.

V. Property preservation

An abstraction is *(strong) property preserving* if a property is true in the concrete model if and only if the abstract property holds in the abstract one. *Weak* preservation means the following: the fact that an abstract property holds in the abstract model implies that the corresponding property is true in the non-abstracted model, however, nothing is guaranteed if the abstract property is violated. It can be proven that our proposed abstraction is weak property preserving with respect to temporal uniform properties. The formal proof of this property is beyond the scope of this paper.

Furthermore, strong preservation can be proven if we restrict the class of distributed protocols. Generally, the next state of a processor depends on its *current* state and the *messages* received from the other processors. Memoryless protocols are protocols, where the next state is *not* dependent on the current state of the node.

VI. The main application: model checking

The main application of our abstraction is model checking. We assume a model checking environment that supports verification of LTL formulas. Instead of model checking if a property holds in the concrete model, we propose to build the abstract model and verify the abstract property. Initial experiments show that the abstract model reduces the size of the state space by several magnitudes also speeding up the model checking significantly.

The following table (Table 1) contains the results of model checking a 2-rounds, binary consensus protocol [3]. The protocol tolerates one malicious (Byzantine) sender if the number of nodes is greater than 3. The experiments were carried out with SAL's symbolic model checker [4].

nodes	Checking **agreement**				Checking **validity**			
	Concrete model		Abstract model		Concrete model		Abstract model	
	States	Time (s)	States	Time (s)	States	Time (s)	States	Time (s)
4	$1.34 \cdot 10^{08}$	13.48	1152	0.17	$1.34 \cdot 10^{09}$	12.92	4416	0.28
5	$3.43 \cdot 10^{11}$	23.87	5440	0.30	$5.84 \cdot 10^{12}$	21.55	$2.65 \cdot 10^{04}$	0.47

Table 1: Results of model checking a consensus protocols (the concrete and abstract models, resp.)

References

[1] N. A. Lynch, *Distributed Algorithms*, Morgan Kaufmann, 1996.

[2] A. Pnueli, "The temporal semantics of concurrent programs," in *Proceedings of the International Sympoisum on Semantics of Concurrent Computation*, pp. 1–20, London, UK, 1979. Springer-Verlag.

[3] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *J. ACM*, 27(2):228–234, 1980.

[4] L. de Moura, S. Owre, H. Rueß, J. Rushby, N. Shankar, M. Sorea, and A. Tiwari, "SAL 2," in *Computer-Aided Verification, CAV 2004*, R. Alur and D. Peled, Eds., vol. 3114 of *Lecture Notes in Computer Science*, pp. 496–500, Boston, MA, July 2004. Springer-Verlag.

AUTHORIZATION FRAMEWORK

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.

Centralizing authentication and authorization has now become the cornerstone of any solid enterprise-wide access management solution: centralized management of each user's access permissions, roles, and detailed audit logs of their actions and activities is fundamental to the success and security of the large-scale enterprise. While it simplifies management, centralizing such indispensable and performance critical services can easily become the reason for degraded throughput accompanied by underutilization of other components.

The authorization service has always been a critical component of IT infrastructures, since the unavailability of the access control service would impact the whole system's availability. The performance of the authorization service also has a direct effect on business throughput as it could easily become a bottleneck in any critical business process. These needs require the implementation of a robust, high performance authorization service with straightforward authorization logic simple enough to provide qualities in both performance and manageability, yet providing the ability to involve flexible plug-ins or external rule engines to enable the authorization process to satisfy more complex policies.

Business needs furthermore require the authorization process to handle security policies that apply to the protected objects based on a variety of runtime conditions, such as context and environment, while still providing the performance to allow the desired utilization of the rest of the system.

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 [3], Clark- Wilson model [4], or Role Base Access Control (RBAC) [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.

Performance, scalability and flexibility are indispensable properties where large scale access control is required. Core RBAC is a security model that can be implemented to provide enterprise access control in a well performing manner. Core RBAC also satisfies traditional flexibility requirements, but as business needs require authorization based on dynamic conditions, vendors have to seek a solution outside of Core RBAC.

The lack of a standardized and well recognized method of handling dynamic behavior of security policies has lead to custom extensions provided by vendors of access management products. The absence of proper formalization is a serious drawback in environments where the scale and complexity of the security policy disables manual verification thereof.

As a response to these issues the Core RBAC security model has been extended to provide a generic approach to model dynamism in the authorization decision process. The DRBAC (Dynamic Role Based Access Control) model has been crafted to enable the straightforward implementation of a high performance authorization service without placing any constraints on the environments it can be applied to.

The decision has been taken to implement a DRBAC layer on top of the robust and scalable IBM Tivoli Access Manager. The development efforts have started with the primary goal of creating a Tivoli Access Manager based simple proof-of-concept implementation for the DRBAC model. Due to a couple of months' work, the mission has matured to a project consisting of a stable, highly optimized authorization service accompanied by a convenient thin client administration interface and a J2EE demo application.

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.

The standard TAM provides the following authorization logic:

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

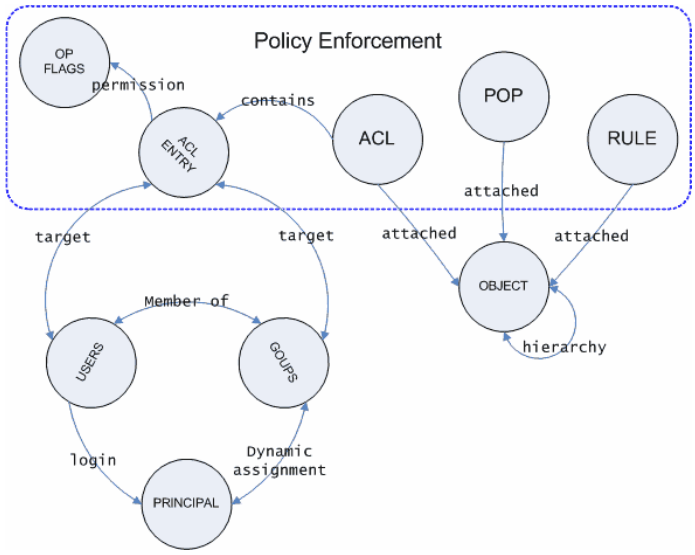


Figure 1: The Tivoli Access Manager RBAC model

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.

A typical 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.

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.

Tivoli Access Manager supports up to 32 action groups, each consisting of 32 individual actions allowing to represent a total of 1024 RBAC operations. ACL entries store the set of permitted actions within Access Manager. ACL entries' permission portion is a 1024 long Boolean array that represents the privilege of performing the 1024 actions.

In the proposed solution the authorization process converts the textual representation of the requested operations to the binary 1024 bit representation. Then, this bit array is masked against the 1024 bits of the ACL entries to decide whether access should be granted by the ACL attached. This task is executed with enormous performance that does not depend on the amount of actions or action groups that are used within the system. Those 1024 bits are compared by a single, atomic instruction.

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

III. Standalone versus plug-in architecture

The immplementation of a standalone DRBAC authorization server running on a separate node is the first alternative, but centralizing request evaluation can lead to difficulties if many endpoints share the service provider. To eliminate this issue, the DRBAC service provider layer has been implemented as a security plug-in that has to be deployed into the secured application's code space.

The advantages and drawbacks of the two architectures from various aspects are:

- A standalone server that exposes a platform independent communication interface can be utilized by applications independent of whether they utilize the .NET framework, J2EE, of native libraries. However, implementing the communication on the client side is always involved. The plug-in, as implemented, only targets Java applications without additional interfacing; however, promoting the functions to a remote interface can turn the plug-in into a standalone server.

- The fact that the plug-in has to run on the secured host raises a security objective: the risk of exploits through byte-code manipulation that are absent in the case of a standalone server that is only communicated with through network protocols. Serious effort has been made to ensure integrity and avoid vulnerabilities.
- As far as performance is concerned, the plug-in has obvious advantages over the standalone operation mode. The processing overhead of the DRBAC attribute mapping is performed in the client JVM and does not impact the host main Tivoli Access Manager components run on. This way, the additional load is split up among the secured nodes rather than a central authorization server. This allows for better scalability and performance as well. In addition, unlike the standalone mode, no supplementary network traffic is involved in performing dynamic authorization, since the attributes only undergo in-memory operations on the secured host.
- The more redundant plug-in architecture also has advantages in term of high availability. The failure of the plug-in – possibly caused by hardware or JVM failure – does not affect other plug-ins, which reside in other hosts' JVMs. Since the protected J2EE application and the security plug-in share the JVM, and so the same hardware resources, both the protected and the protecting code would become inaccessible due a hardware failure, but other secured application would not be affected.
- Manageability of the security solution is not affected by the redundancy of the plug-in architecture as long as all the configuration data is kept in the Tivoli Access Manager authorization database – that means, if no composition is utilized. In the case of composition-aware DRBAC plug-ins, the administration of authorization metadata becomes more complex since the XML configuration file has to be maintained for each plug-in. However, this task could be accomplished by automating the process of distributing the updated configuration file to each host the plug-in runs on.

Future work

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] J. Ganci, H. Boog ,M. Fletcher, B. Gordon : *Develop and Deploy a Secure Portal Solution*, IBM Redbook 2004

[2] Draft: *Role Based Access Control Implementation Standard*; January 2006
URL: http://csrc.nist.gov/rbac/draft-rbac-implementation-std-v01.pdf

[3] D.E.Bell, L.J. La.Padula: *Secure Computer System: Unified Exposition and Multics Interpretations*, Technical report prepared for Command and Management Systems, United Stated Air Force, Bedford, Massachusetts, 1976. URL: http://csrc.nist.gov/publications/history/bell76.pdf

[4] 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.

ROBUSTNESS COMPARISON OF HIGH AVAILABILITY MIDDLEWARE SYSTEMS

Zoltán MICSKEI
Advisor: István MAJZIK

I. Introduction

Recently availability 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 manage redundancy and achieve minimal system outage can be implemented independently from the application, and can be put on the market as a *HA middleware*. The standardization of the functionality of such middleware systems has begun as the leading IT companies joined to the Service Availability Forum (SA Forum) to elaborate the Application Interface Specification (AIS [1]). The benefit of an open specification would be for example the easier integration of different off-the shelf components.

With multiple middleware 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 HA products the *dependability* is also an important property to be considered. This paper outlines an approach to compare one of the attributes of dependability, *the robustness* of HA middleware systems.

II. Robustness testing approach

Robustness is defined as the degree to which a system operates correctly in the presence of *exceptional inputs* or *stressful environmental conditions*. Based on earlier robustness testing projects (like *Ballista* [2]) and dependability benchmarks (like *DBench* [3]) we elaborated an approach for high availability middleware systems, which was presented in [4].

The first step of developing the test strategy was to identify the potential sources for activating robustness faults in the HA middleware. Figure 1 illustrates these sources, considering a typical computing node of a HA distributed system as follows:

1. *External errors*: These errors are related to the application, thus they are not covered by the robustness testing of the middleware.
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 application components using the public interfaces of the HA middleware can lead to failures if they use exceptional values, e.g. NULL pointer or improperly initialized structures.
4. *OS calls*: The robustness of a system is also characterized by its ability to handle the exceptions or error codes returned by the OS services it uses.
5. *Hardware failures*: The most significant HW failures in a HA systems are host and communication failures (that has to be tolerated in the normal operating mode of the HA middleware) and lack of system resources.

From the above sources the following ones were selected to be included in the first version of the dependability benchmark suite:

The standardized middleware API calls are considered as a potential source of activating robustness faults. Because of the high number of possible exceptional value combinations and scenarios, the elements of the robustness tests suite were automatically generated by tools. The

challenge in testing the API calls was that most of the AIS interface functions are state-based, i.e. a proper initialization call sequence, middleware configuration and test arrangement is required, otherwise a trivial error code is returned.

The failures of the OS system calls were included for the following reason. They do not only represent the faults of the OS itself (which has lower probability for today's mature operating systems), but failures in other software components, in the underlying hardware and in the environment also could manifest in an error code returned by a system call. Possible examples of such conditions are writing data to a full disk, communication errors when sending a messages, etc.

Studies show that operator errors cause also a significant part of service unavailability, however the configuration of the HA middleware and the system management interface are still under standardization by the SA Forum, thus they were not included in the benchmark suite.

III. Testbed tools and benchmark suite

Taking into consideration the potential sources of activating robustness faults, a set of tools was developed to assist the activation of these faults by generating proper test values and performing the test calls. This dependability benchmark testbed is depicted on Figure 2. In the following, we describe these tools and their application during robustness testing.

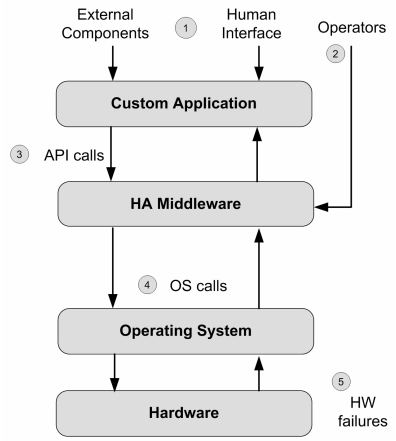


Figure 1: HA middleware fault model

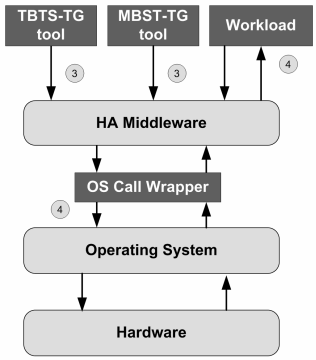


Figure 2: Testbed tools

A. Template-based type-specific test generator

The template-based type-specific test generator (TBTS-TG) uses the following approach to generate robustness test cases that realize calls to the HA middleware API with exceptional values. Instead of defining the exceptional cases one by one for each function, the exceptional values are defined for the parameter *types* that are used in the API functions. From the description of these types, the tool generates a *test program* for each API function, that calls the given function with all combinations of the specified values. Each combination is executed in a new process to separate the test cases from each other, and the result code of the call is logged after completion. The test case is considered to detect a robustness failure if it has caused a segmentation fault or a timeout. To help diagnosing the robustness faults, in the first calls only a single parameter was assigned an exceptional value (using valid values in the case of the remaining parameters).

The input for the tool consists of the following elements. The skeleton of the test program is given as an XSL template. The metadata of the functions and types to test are specified in XML files. The exceptional and valid values are defined as C code snippets. For simple types, e.g. numbers and enumerations, values recommended by traditional testing techniques were selected, like valid values, boundary values and values outside the domain of the given type. For complex structures, each member is assigned a combination of invalid values while the others remained valid.

B. *Mutation-based sequential test generator*

While the TBTS-TG tool tests mostly individual functions, the mutation-based sequential test generator (MBST-TG) could be used to generate complex call sequences. The basic idea of the tool is that mutation operators representing typical robustness faults, like omitting a call or changing the specified order of calls, are applied to valid functional test programs using the HA middleware. In this way a large number of complex robustness test cases can be obtained automatically.

The challenge of implementing the MBST-TG tool was the parsing and modification of the input source files. Parsing standard C files is not as trivial as it seems, the available free parsers encountered various problems when system header files were included in the input files. Thus instead of obtaining the full parse tree (that is required for compilation) a light-weight method was used. The srcML tool was used to build an XML file representing *only the syntactic structure* of the input source files. This syntactic structure is enough to implement the common mutation operators.

Currently five mutation operators are implemented. The inputs of the MBST-TG are the source files to be mutated and a configuration file that describes the parameterization of the mutation operator, e.g. the filters to be used when searching for a call to apply the mutation. Note that the mutation may result in such source code that cannot be compiled (data flow analysis is not performed, this way, for example, changing of function calls may result in using variables that were not assigned a value before). Finally 92 valid mutants were included in the test suite.

C. *OS Call wrapper*

The OS call wrapper intercepts system calls executed by the HA middleware and injects exceptional values into them. To include all important system calls used by the middleware a workload application is needed, which generates calls to the middleware resembling typical application activities.

The OS call wrapper can be configured to intercept or delay selected system calls. The return value of an intercepted call could be (i) the actual value returned by the original system call, if the call was also forwarded to the OS, (ii) a predefined valid or exceptional value or (iii) a randomly selected value from the possible error codes of the function. The wrapper is implemented using the Unix LD_PRELOAD variable, which can be used to load predefined libraries instead of system libraries.

As a workload to explore the OS system calls from the middleware, a general application was used that implements a search and index engine. The application utilizes the failover and checkpoint service of the middleware. Using the *strace* utility the system calls of the middleware were monitored, and 10 calls were selected for interception.

IV. Robustness comparison results

Using the benchmark suite created by the above tools, the robustness of *openais*, an open source implementation of the AIS specification was evaluated. Two versions of openais were used in the tests: 0.80.1 (the latest stable release) and the trunk (the latest development version directly from the source control system of the project).

Results from the type-specic tests: Just by trying to compile the test suite on the system under test, several discrepancies were found: The header files used in openais differ in several places from the official header files of the AIS specification, and thus from the header files used by the test suite.

Table 1 summarizes the results obtained by executing the test cases. Segmentation faults definitely indicate robustness failures, since in a HA middleware even invalid inputs should be handled correctly. Timeout could indicate normal behavior, because some of the API functions could be parameterized to wait for an event to dispatch. However, the large number of timeouts in openais-0.80.1 is not reasonable taking into account the concrete values used in the benchmark.

Some of the test cases caused fatal error in the middleware. The tests for 5 functions produced an internal *assertion violation* and the middleware exited. Different assertion violations were observed, they were produced at the registration phase of the application component.

In openais-0.80.1 segmentation faults were observed in 12 functions while timeout were found in 7 functions. These numbers were slightly lower in openais-trunk (9 and 2, respectively), which indicates an improvement achieved during the development of the openais implementation.

Table 1: Resulting status code of test cases for two versions of openais

Status code	Number of test calls in openais-0.80.1	Number of test calls in openais-trunk
0 (success)	25610 (91%)	27863 (96,1%)
11 (segmentation fault)	1336 (4,8%)	1048 (3,6%)
14 (timeout)	1183 (4,2%)	94 (0,3%)
Total	28129	29005

Results from the mutation-based testing: Some of the mutants produced a third kind of assertion violation in the middleware, while others resulted in timeout. In the case of openais-trunk, the test results were the following: assertion violation 47,6%, timeout 33,3% and finished 19%.

Results from the OS wrapper: For each of the identified system calls a separate test was executed, where the workload application was started and after a while a failover was forced. During the execution the system calls were forwarded to the OS, and with a predefined probability a random error code was returned (the probability was adaptive as it depended on the frequency of the call). In the case of two OS functions no robustness failures were observed, another two functions were not called by the actual version of the workload. For three OS functions the CPU utilization was raised to 100% and the test could only be terminated by a hard reset. In two cases the middleware detected the failure and exited because it could not recover in other way. In one case an assertion was triggered during the execution. These latter results indicate robustness failures.

V. Conclusion

In this paper a robustness testing approach for HA middleware systems was presented. The novelty of the approach is the application of automatic tools that construct the test cases systematically on the basis of the standard interface specification (API functions) and existing functional test suites. The robustness testing of an open source HA middleware demonstrated that these tools can be used efficiently and their test results are complementary as they detect distinct failure types. It turned out that there are still several robustness problems both in version 0.80.1 and in the trunk version of the openais implementation. It is important to emphasize that robustness testing can be used only to observe these problems, and further work is needed to find the causes and to turn the observations into dependability benefits, e.g. by identifying the wrong implementation approaches or coding errors that shall be corrected.

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.

[4] Z. Micskei, I. Majzik and F. Tam, “Robustness Testing Techniques For High Availability Middleware Solutions,” in *Proc. of Int. Workshop on Engineering of Fault Tolerant Systems*, Luxembourg, Luxembourg, 2006.

Using Virtual Computers in Developing High Availability Systems

Péter László PÁSZTOR
Advisor: András PATARICZA

I. Introduction

High availability systems require extensive testing to prove their tolerance to faults. This paper serves the purpose to show the possibility of using virtualization techniques to enhance the testing process of high availability systems.

A. Virtualization in general

Virtualization is an emerging technology between simulation and emulation of computer systems. It offers the best of both worlds encapsulating the system, while having nearly native performance without requiring additional expensive hardware by using the fact that the hardware to be emulated has a similar architecture to that of the emulating system. This conception enables the use of more efficient layers of software between the virtual machine and the physical one than compared to simulation, while permitting the use of generic hardware. Virtualization lowers time-to-market, and the product cost in software development, and overall IT costs in enterprise environments by lowering the power and space requirements by optimizing the usage of the IT resources.

B. Virtualization and high-availability

Availability may also benefit from utilizing virtualization, enabling a physical computer to provide more than one functionality without compatibility issues caused by more services being present in one environment. Using a virtualization tool, every service can have a separate environment, even if they are running on physically the same machine, resulting also in a finer service granularity, so in case of a service failure, the other services are not directly affected. In case of a physical computer failure, the other physical machines incorporating the same technology are able to take the services over, and continue providing them.

II. Testing of high availability systems

A. Fault injection

High availability systems are rather difficult to test, since availability testing needs faults to be injected, to examine the response of the whole system. Such faults might be CPU failures, memory failures, disk read or write errors, errors in other peripherals, power or network failures, or any kind of failure that can appear. In physical testing environments, it is cumbersome to test some of these.

Alternatives to injecting physical faults are using a software simulator – being too slow – or an emulation environment – which is unavailable in most cases, and expensive when available, and virtualization – falling in between the two – is a viable solution.

B. Fault injection using a virtualization layer

All virtualization tools share the same principle of isolating the hardware from the software by using a software layer of different width. For example the VMware products completely encapsulate the guest systems, almost entirely hiding away the actual hardware, with as thin software layers as possible, as opposed to Xen, which basically lets the guest system to use the bare metal in their own time window, just controlling the access to the devices.

Both implementations could influence the operation of the systems, and inject faults without requiring a faulty processor, or modification of the application, just the specification of the fault, and

when, or how often to inject it, only a fault injection API should be added to the virtualization tool – which is possible in case of the open-source Xen.

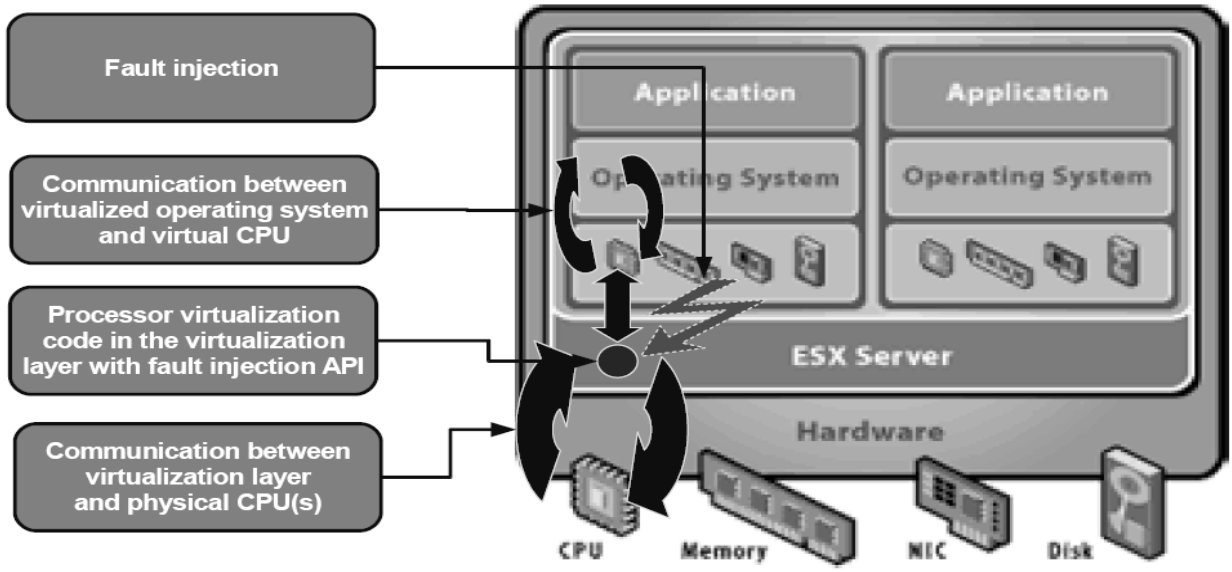


Figure 1: Model of a virtualized environment with fault injection (VMware ESX server, original:[1])

As it is clearly visible on Fig. 1, from the virtualization layer (labeled ESX server in the picture) it is possible to inject failures to every hardware device a virtual machine can see without compromising the integrity of the other virtualized systems. Using hardware virtualization in Xen the whole application itself can be tested without any modifications, just like it was running on the hardware itself – only faintly slower, and with some resource limitations.

C. Testing using virtualization tools

No matter what the result of a test was, in short time the whole virtual system can be restored to the same point (snapshot) and another test can be run. This process can be automated, which means that large number of tests can be done without continuous supervision, and the results accumulated and processed.

D. Drawbacks

Virtualization has remarkable advantages in testing high availability systems, but has a few downsides also. Unlike Xen, most of the other virtualization platforms isolate the virtualized system from the physical hardware, and this means that the virtualized system needs to have specialized device drivers – so it is not exactly the same system.

Even the virtualization layer can modify the behavior of the applications so that it makes the test provide inappropriate information, for example the faintly slower operation might hide away a timing fault emerging from multiple faults in a short period, so the effect of the virtualization layer on the result of fault injections must be examined.

III. Conclusion

This work is currently at the stage of experimentation, and the evaluation of the available solutions. A testing environment is being prepared for this examination that will use the fault injector described by Volkmar Sieh in [2].

References

[1] VMware ESX Server product overview, URL: http://www.vmware.com/products/vi/esx/

[2] V.Sieh: Fault-injector using UNIX ptrace Interface, Internal Report, Erlangen, Dec. 1993

HIGH-ACCURACY CALIBRATION OF LANE DETECTION SYSTEMS USING STEREOVISION

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

I. Introduction

Automatic lane detection plays an important role in the autonomous lateral control of on-road vehicles. The objective of lane following systems is to the increase safety by determining the exact position of the vehicle within the actual lane, detecting the lane's curvature, setting a viable vehicle trajectory and actuating through the electronic braking system (EBS) or through the electronic power assisted steering system (EPAS). For lane detection, the application of visual sensors is usually preferred to laser sensors and radar, as vision sensors are of non-intrusive nature and they do not require additional infrastructure. Stereo systems (e.g. [1]) are capable of achieving a higher accuracy in 3D lane geometry reconstruction than mono systems having to rely on simplifying assumptions do. However, the fulfillment of this potential heavily depends on the quality of the camera calibration. Therefore, firstly a high-accuracy far-range camera calibration method for stereo lane detection systems has been devised and has been investigated for its accuracy. Secondly, a real-time, FPGA-based stereo lane detection system is under development. This system is also presented here.

Figure 1: Stereo matching: the correspondent of the highlighted points in the left image is searched along epipolar lines in the right image

Recent stereo systems reconstruct road geometry by first detecting feature points that belong to the current lane within a predicted region of interest in each of the two images [1]. Then they search for point correspondences using correlation techniques (stereo matching) [2]. The pair of a certain feature point appearing in one of the images must lie on a specific line (called epipolar line) in the other image [2]. This constraint significantly simplifies the search and is due to the rigid stereo geometry of the cameras being fixed to the vehicle chassis. The matching point-pair is reprojected into the 3D space by triangulation [2] and a lane model is fitted to the resulting point-cloud and also to the features in the images [1]. The fitting is done in two main steps. In the first step the lane's vertical profile is determined, in the second step its horizontal profile is computed. If the camera parameters are not determined with reasonable accuracy during the camera calibration procedure, the computed epipolar line is shifted and stereo matching may fail [3]. The triangulation may result intolerable inaccuracies in the reprojected point coordinates. Therefore, a precise initial far-range camera calibration is indispensable for such systems.

II. Far-range camera calibration

A. The proposed two-step calibration method

The most common approach for calibrating a single camera is to take snapshots of a known high-contrast repetitive pattern, such as checkerboard, held in different orientations and then based on its camera image, estimate the intrinsic camera parameters (relative focal lengths, principal point, skew, distortion coefficients) and extrinsic camera parameters (camera location and orientation in the world reference frame) by minimizing the geometric error in the image [4,5]. This relies on the assumption

that the marker arrangement (checkerboard pattern) is printed accurately and is planar [4]. In the case of far-range camera calibration this is not the case since a precise pattern or marker arrangement is quite expensive to set up. We therefore propose a method for measuring the marker locations in 3D space with relatively low-cost but high-precision laser distance meters followed by a two-step calibration method: first the intrinsic parameters and then the extrinsic parameters are computed.

B. Intrinsic parameter calibration

The intrinsic parameters are calibrated independently for the two cameras with the method proposed by Z. Zhang [4]. This method is used because of its simplicity and high accuracy with respect to the internal parameters. The noise present in the square corner locations measured in the images of the checkerboard is assumed to be isotropic (direction-independent) Gaussian and uniform (the distribution is the same everywhere in the image). No noise is supposed to be present in the real corner locations, the printed pattern is assumed to be accurate. The deviation in the image can be approximated with the standard deviation of the measured corner locations from the reprojected ones. The error in the intrinsic parameters is computed by linearizing the camera model in the vicinity of the estimated parameters and by transporting backwards the measurement error to the parameters. Practically, Gaussian noise is an acceptable assumption in the case of close-range calibration (the isotropic nature of the pattern and of the corner detector, small depth range) but is not realistic in far-range calibration as in this case significant noise might be present in the measurement of 3D-marker locations, as well, that do not translate into a Gaussian noise in the image.

C. Measurements for extrinsic parameter calibration

The vehicle with the two cameras fixed to the chassis is driven into an open flat area of about 10 meters wide and 30-35 meters long. X-shaped marker plates are placed perpendicular to the ground within the field of view of the cameras. The location of the markers is suggested to be measured by using high-precision laser-based distance meters from two reference points. These points can be designated anywhere in vehicle coordinate system (VCS). From the two measured distances d_{Ai} and d_{Bi} and from the known reference point locations R_A and R_B, the 3D location $\mathbf{W_i}=(x_i,y_i,z_i)^T$ of the i-th marker can be computed. The height coordinate is supposed to be constant and known for all the markers. The planarity of the area is required, otherwise the measurement of the marker centers' y-coordinate would require at least 3 reference points and the problem would be very ill-posed. The errors in x_i, y_i, z_i coordinates can be calculated by evaluating the Jacobian of the function that maps (d_A,d_B) into (x,y,z) at (d_{Ai},d_{Bi}) and by estimating the errors in the location of R_A, R_B and in d_A and d_B.

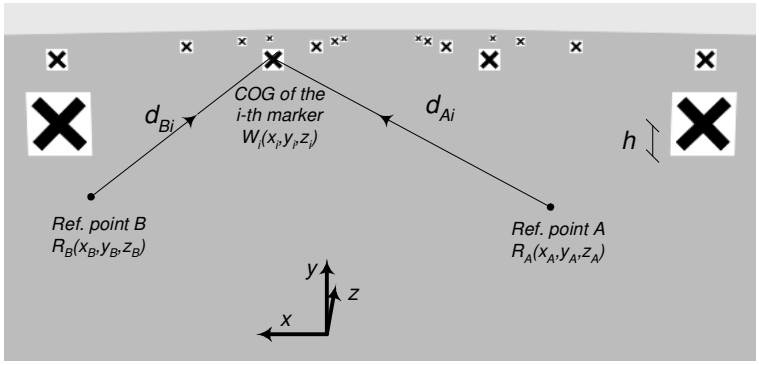


Figure 2: Marker arrangement for extrinsic calibration and the two-point distance measurement arrangement.

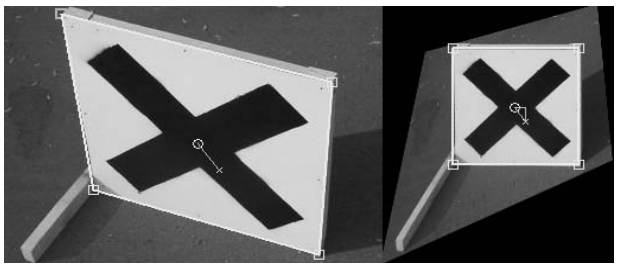


Figure 3: Automated aiming error detection on the snapshots created with a digital camera: original image and metrically rectified image. (o – center of gravity, x – laser dot)

An improvement of this method is also proposed. It takes into consideration the aiming errors $\boldsymbol{\delta_A}=(\delta_{Ax}, \delta_{Ay})^T$ and $\boldsymbol{\delta_B}=(\delta_{Bx}, \delta_{By})^T$ measured on each marker. $\boldsymbol{\delta_A}$ and $\boldsymbol{\delta_B}$ can be measured by taking a close-range snapshot (with a digital camera) from each marker when it is aimed with the laser beams. With the assumption that the exact

sizes of the marker plates are known, $\boldsymbol{\delta_A}$ and $\boldsymbol{\delta_B}$ errors can be automatically measured with the metrical rectification [3] of the markers in the snapshots. The precision of $\boldsymbol{\delta_A}$ and $\boldsymbol{\delta_B}$ depends on the resolution of the snapshot and is tipically in the sub-millimeter range.

Consequently, these measurements provide the 3D locations $\mathbf{W_i}=(x_i,y_i,z_i)^T$ of the markers' COG and their uncertainties $\Delta\mathbf{W_i}=(\Delta x_i,\Delta y_i,\Delta z_i)^T$.

D. Algorithm of extrinsic calibration for far-range

Each camera is modeled by a nonlinear mapping φ of 3D points into 2D points in the image:

$$I = \varphi(p_{\text{int}}, p_{ext}, W) \tag{1}$$

Here **W** represents the 3D point (marker COG), **I** represents the 2D image point, $\mathbf{p_{int}}$ and $\mathbf{p_{ext}}$ are the vectors of 9 intrinsic and 6 extrinsic parameters (3 for orientation, 3 for translation) of one camera, respectively. Radial lens distortion is the most significant type of distortion in this mapping and therefore a polynomial radial distortion model is included. $\mathbf{p_{int}}$, **W** and their uncertainty are already determined by using the methods described in sections C and D. **I** in the image of each of the two cameras is extracted by an automatic regression-like method that is able to detect the COG of the markers in their images. The related uncertainty can be estimated by running the implemented X-detector on multiple synthetic images and by evaluating the distribution of the error. The X-detector can be implemented with sub-pixel accuracy (as well as the corner detector at the checkerboard method). $\mathbf{p_{ext}}$ is determined by iterative minimization of a non-linear cost function. In the literature, the most commonly used cost function is the reprojection error:

$$C(p_{\text{ext}}) = \sum_{i=1}^{n} \left\| I_i - \varphi(p_{\text{int}}, p_{\text{ext}}, W_i) \right\|^2 \tag{2}$$

The usage of this cost function means that inaccuracies in the (3D) metrical marker coordinates are all translated into a geometric error in the image. If nothing is known about the noise structure, this is the best we can do. In [4] an alternative method is presented. They apply nonlinear equation solving and therefore they minimize an algebraic error that is not related to any physical quantity. Furthermore, they do not deal with radial distortion models.

We have information about the measurement procedure, the uncertainties in the metrical measurements and the errors of the X-detection in the image are known. Therefore, a different solution is being investigated which exploits all the experimental information we can have about the noise structure, which minimizes geometrical errors and which can deal with radial distortion models due to its generality. Using the proposed method we minimize errors in the 3D metrical space and in the images simultaneously and we sum the terms by weighting them with the inverse of their confidence level. Therefore we suggest building in the inaccuracy information $\mathbf{\Delta I_i}$ and $\mathbf{\Delta W_i}$ resulted from our preliminary experimental considerations into the cost function. In fact, this is the best we can do in our case, because the noise distributions are not exactly known. The uncertainty of the external parameters ($\mathbf{p_{ext}}$) can be estimated by evaluating partial derivatives of the camera model and by using the already-known $\mathbf{\Delta p_{int}}$, $\mathbf{\Delta I_i}$ and $\mathbf{\Delta W_i}$ uncertainties. The Direct Linear Transformation (DLT) usually gives a good initial guess for the nonlinear minimization [2].

III. A lane detection system under development

In order to record videos in different traffic situations, lighting conditions and road types a stereo camera system is installed on a test car with the cameras fixed to the side mirrors. This system uses analog CCD cameras and can provide 720x576@25fps videos for lane detection algorithm development. This system will be calibrated with our new calibration method. We should emphasize that this system is only used for video recording, not for real-time processing.

Parallel to the work related to camera calibration, a real-time stereo vision system is under development that is applicable for stereo lane detection (and for other stereo applications as well).

This is based on two grayscale high-resolution network cameras with a Spartan3 FPGA and an additional 100 MIPS RISC processor optimized for Linux. As the software and the IP core are open-source, these intelligent cameras can be reconfigured to perform pipelined image processing steps that are executable independently for the two views. The data provided by the cameras are transmitted through standard 100 Mbps Ethernet interfaces into a high-performance Laptop computer (CoreDuo, 2 GHz) running Linux. Specific TCP/IP receiver software will couple the corresponding frames and pass them to the stereo vision algorithm. A performance of 640x480@15fps is expected from this system (although the cameras are high-resolution and high-speed). Considering vehicle dynamics as well, this is enough supposing a maximum vehicle speed of about 130 km/h [1].

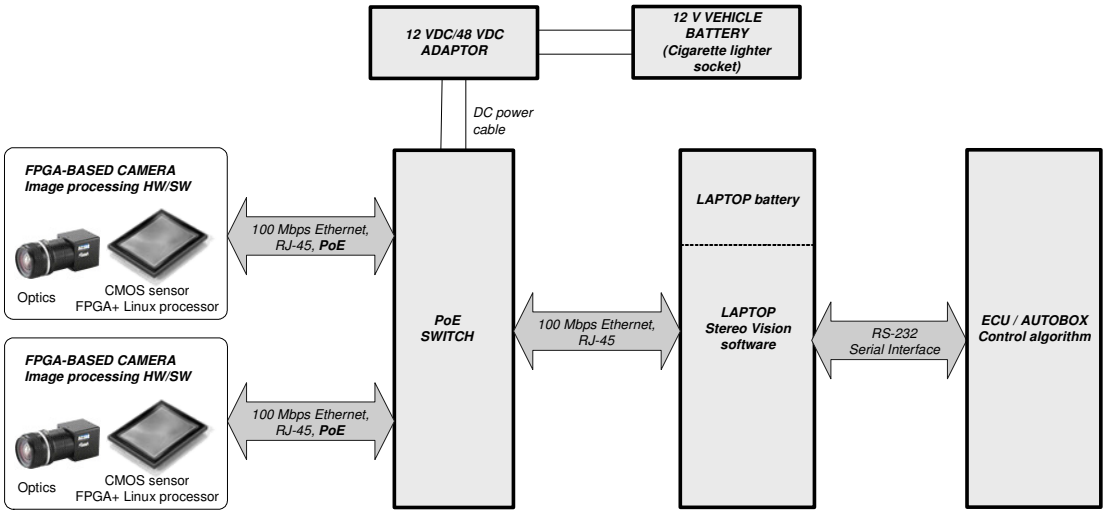


Figure 4: The architecture of the real-time stereo vision system under development.

IV. Conclusions and further work

A new off-line far-range camera calibration method has been investigated and briefly presented in this paper. This includes marker location measurements and calculation with high-precision laser-based distance meters, aiming error correction and error estimation of the measurements. The two-step calibration algorithm is under implementation in MATLAB. The presented calibration method will be easily adapted to other stereo vision applications that require high accuracy.

In our case, the high-precision calibration of the preliminary stereo-recording system will be used to record videos in different environmental conditions. The synchronized videos will be used in the development and testing of lane detection algorithms as well as for evaluating the precision of some on-line stereo autocalibration methods investigated in the future. On-line autocalibration is required because camera parameters may change due to shocks and vibrations and these changes might cause significant errors in lane reconstruction.

A stereo vision system under development has been presented, as well. This system will be applicable for real-time stereo lane detection in the future.

References

[1] Nedevschi, S., et al., "Driving Environment Perception Using Stereovision" in *Proceedings on IEEE Intelligent Vehicles Symposium*, pp. 331-336, Las Vegas, USA, June 2005

[2] Hartley, R., Zissermann, A., *Multiple View Geometry in Computer Vision*, Second Edition, Cambridge University Press, March 2004

[3] Marita, T., Oniga, F., Nedevschi, S., Graf, T., Schmidt, R., "Camera Calibration Method for Far Range Stereovision Sensors Used in Vehicles" , *Intelligent Vehicles Symposium*, pp. 356-363, Tokyo, Japan, 13-15 June 2006

[4] Zhang, Z., *A flexible new technique for camera calibration*, Technical Report, Microsoft Research, Microsoft Corporation, Redmond, USA, Aug. 10, 2002

[5] Bouguet, J.-I., Caltech Camera Calibration Toolbox for Matlab, 18 October 2006, URL: http://www.vision.caltech.edu/bouguetj/calib_doc/

DISTRIBUTED SIMULATED ANNEALING IN SENSOR NETWORKS

Dániel DARABOS
Advisor: Gábor HORVÁTH

I. Introduction

Sensor networks have gained the attention of a large number of researchers in the last few years. With the advent of low cost, low energy consumption microprocessors developed in other fields of engineering (such as mobile phones, mobile gaming devices, mobile music players and PDAs) and more attention focused on wireless communication (again thanks to research on mobile telephony, wireless internet solutions and Bluetooth devices) wireless sensor networks became a viable solution for a number of problems. Some of these are problems for which wired solutions were available, but there are also problems for which no appropriate solution was given before. These problems include the surveillance of a large area (as in ecological monitoring systems) and the measurement of a single event from multiple vantage points (as in counter-sniper systems[1]). Wireless sensor networks have a number of advantages over wired solutions which include the ease of deployment, flexibility of configuration and higher fault tolerance.

These sensor networks however give rise to a number of problems themselves. Because of their ease of deployment it can be desirable to be able to distribute them over an area without much consideration. In these scenarios the locations of the individual network nodes will have to be determined by the network itself. Examples of research concerning this problem can be found in [2] and [3]. Another problem which is important to sensor network design is the routing problem. Without wires and with a large number of nodes communications can be routed in a very large number of ways. Because of concerns for energy efficiency and lossy communication determining the optimal routing is far from trivial. Also the source and sometimes the destination of the communication can be any one of a large number of nodes. An example of a routing protocol for mass data transfer is described in [4].

Currently the solution for a number of the above mentioned problems is centralized. In these solutions all necessary information is collected from the nodes of the network to a special node (most often a personal computer) and the bulk of the computation takes place on this node with only some preprocessing performed on the other nodes (examples include [1] and [3]). While at this time such centralized solutions may well be optimal in terms of energy efficiency and speed, in many cases this is going to change in the future.

Complexity of the problems which centralized solutions solve are generally proportional to the size of the network – meaning either the number of nodes or the number of edges in the communication graph or both. While the computational power of the central node is going to increase as the power of CPUs increase, the size of the networks is going to increase at a much faster pace as the cost of network nodes decreases. What this means is that sensor network applications are approaching a barrier beyond which centralized solutions are not going to continue to be viable.

The total processing power present in the network nodes themselves however obviously grows with the size of the network. As technology improves the processing power present in a single node is going to increase as well. These observations suggest that after a certain point is reacehed in the development of wireless sensor network technology, distributed computing solutions are going to replace a significant number of current centralized solutions.

At the core of a large number of algorithms pertaining to sensor networks an optimization problem can be found. This is actually the case in all of [1], [2], [3] and [4]. In many cases a specialized optimization algorithm is the best choice, but there are also general optimization algorithms with a

history of providing good solutions for a large variety of problems. One such general algorithm is Simulated Annealing[5] and in the current article we would like to propose a distributed version of it for application in sensor networks.

II. Simulated Annealing

Simulated Annealing is an optimization algorithm for state spaces in which the relative value of neighboring states can be calculated. Let us denote the state space in which the search occurs with X and the objective function giving the value of state $x \in X$ with $f(x)$.

The basic steps of SA are the following. Initially a random x_0 state is picked. Then in every iteration a candidate state y_k is randomly picked according to a $D(\cdot; x_k)$ distribution (the distribution is usually uniform across the neighboring states of x_k, but heuristics can be injected into the algorithm at this point). Then the value function is evaluated for the candidate state and the difference in value from the current state is calculated ($\Delta_k = f(y_k) - f(x_k)$). Next an acceptance function is evaluated which gives the probability of accepting the candidate. With this probability the new state is either accepted ($x_{k+1} = y_k$) or rejected ($x_{k+1} = x_k$). Typically the probability is 1 for $\Delta_k > 0$, the most well known acceptance function being the Metropolis function

$$A(\Delta, T) = \min\{1, e^{\frac{\Delta}{T}}\} \tag{1}$$

The parameter T of this function is the "temperature" of the system and the acceptance function ensures that random steps taken in the wrong direction occur less likely with the decrease of temperature. The idea is that at high temperatures the system will roughly move into the region of the global optimum and then at lower temperatures finer adjustments will be made and the system will settle in the closest local optimum – which is by then hopefully a global one. The temperature value used in each iteration (T_k) is given by a function called the cooling schedule.

Thus there are three parameters of a particular SA algorithm and they are the distribution function for choosing the candidates, the acceptance function for accepting or rejecting them and the cooling schedule for determining the temperature at a given iteration. Other than that the stopping conditions of the algorithm have to be determined. With sensor networks however it is often the case that there is no need for explicitly stopping the algorithm. In these cases the anytime behaviour of SA can be exploited to get increasingly accurate results over time.

SA was first introduced as a combinatorial search algorithm[5][6] and proven convergent as such[7] but more recently variants of SA dealing with continuous global optimization problems have been developed and proven convergent (as in [8]). In the field of sensor networks both the combinatorial and the continuous variants are likely to find their applications.

III. Distributed Simulated Annealing

As the pace of improvement appears to be slowing down for single processor computing power and networks grow faster and larger by the day many researchers have turned their attention toward distributed algorithms. Arshad and Silaghi present a survey of distributed stochastic search algorithms and introduce a form of distributed simulated annealing (DSAN) in [9]. Based on experiments described DSAN appears to be competitive with other algorithms.

For DSAN to work the state space of the search needs to be decomposable. This is not a property of the majority of general optimization problems, but the problems arising in sensor networks are more often of this kind. The reason for this is that often the base of the optimization is the data acquired by the sensors, and often these pieces of data are more closely related the closer the acquiring nodes are to each other. If this property is indeed presented, then each node of the sensor network can work on part

of the problem and only communication with the nearest nodes (those within direct communication range) is necessary.

Distributed simulated annealing then works as follows. Every node ($n \in N$) maintains a part of the state space $x_{n,k}$) so that the entire state space is the product of these parts ($x_k = \prod_{n \in N} x_{n,k}$). Every node can also independently choose a next candidate state and decide about moving to it or not.

Most often however the value function can not be independently evaluated and in this part the cooperation of neighboring nodes is required. If the value function can not be approximated without knowing the entire state space then this algorithm is probably not the best choice. In sensor networks however an approximation can often be based on knowing the current states of just the neighbors, i.e. $x_{m,k}$ for $m \in v(n)$ where $v(n)$ is the set of neighboring nodes of node n.

In cases where the entire state space has to be known to approximate the objective function the algorithm can still be used if past values are acceptable for nodes further away. This can be arranged by the nodes retransmitting the states of neighboring nodes so that this information propagates through the entire network. This can possibly be an unacceptable drain on energy, but in some cases energy can be saved in a variety of ways. Examples include a lower rate of retransmission of information coming from further away and retransmission of information from further away nodes with stronger lossy compression or lower sampling rate (e.g. images of reduced resolution).

IV. An Example

Distributed simulated annealing can be applied to some problems while not to others. When it can be applied there are a number of parameters that have to be decided on. There are also some non-trivial details of translating a centralized optimization problem into a distributed one. While some parts of the methodology can be described, an example can also be valuable in understanding the process.

An application of the distributed simulated annealing is described in [10] as part of the proof of convergence of the author's localization algorithm. The relevant parts of the article will be repeated here to serve as an example of using simulated annealing for solving a distributed localization problem.

In this article the problem considered is determining the relative positions of the nodes based on signal strengths. That means that the state space is $\mathbb{R}^{dN}$ if there are N nodes in a d-dimensional space. A natural decomposition of this problem is finding the position of each node, i.e. $x_{n,k}$ will be the position of node n at the kth iteration.

This problem is particularly well suited for distributed simulated annealing because the objective function in the centralized approach is

$$f(x_k) = -\sum_{i \in N} \sum_{j \in v(i)} (C_{i,j,k} - D_{i,j})^2 \tag{2}$$

where $D_{i,j}$ is the assumed distance between nodes i and j based on the measured signal strength and $C_{i,j,k} = \|x_{j,k} - x_{i,k}\|$ is the calculated distance based on the position of the nodes in the state space. This function is the sum of terms that are functions of only the state of a single node and its neighbors. Therefore the partitioned value function is simply

$$f(x_{n,k}) = -\sum_{j \in v(n)} (C_{n,j,k} - D_{n,j})^2 \tag{3}$$

In centralized simulated annealing algorithms the cooling schedule is most often predetermined and independent of any property of the search process. In a distributed variant however the cooling schedule can be different for every node, and can be heavily dependent on different local properties. In [10] for example the temperature of a node is a function of the sum of squares of the differences between the

assumed and calculated distances:

$$T_n, k = g(\sum_{j \in v(n)} (C_{n,j,k} - D_{n,j})^2) \tag{4}$$

where g is a monotonously increasing function.

Figure 1 shows the results of a simulation run of the discussed algorithm. Although not a pure SA algorithm, its practical behavior is very similar to centralized SA variants.

V. Conclusion

As the example illustrated distributed simulated annealing can be particularly well suited for problems emerging in sensor networks. These problems are generally easily partitioned with the relationships that connect the partitions often constrained to neighboring nodes.

For evaluation purposes future work includes applying this approach to a number of different problems in the field of sensor networks and comparing the resulting distributed algorithms to present centralized ones in terms of communication costs, rate of convergence and scalability.

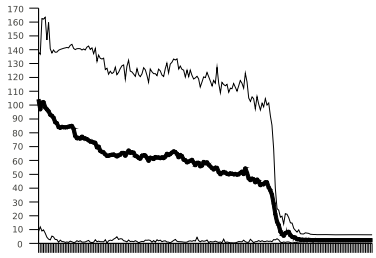


Figure 1: Decrease in maximal, average and minimal angular error during a simulation of the algorithm described in Section IV.

References

[1] G. Simon, M. Maróti, A. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton, "Sensor network-based countersniper system," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 1–12, New York, NY, USA, 2004. ACM Press.

[2] D. Darabos and B. Tódor, "Constraint-based localization of nodes in wireless networks," in *IEEE Third International Workshop on Intelligent Data Acquisition and Advanced Computing Systems*, 2005.

[3] M. Kushwaha, K. Molnar, J. Sallai, P. Volgyesi, M. Maroti, and A. Ledeczi, "Sensor node localization using mobile acoustic beacons," *The 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2005)*, 2005.

[4] P. Volgyesi, A. Nadas, A. Ledeczi, and K. Molnar, "Reliable Multihop Bulk Transfer Service for Wireless Sensor Networks," *Engineering of Computer Based Systems, 2006. ECBS 2006. 13th Annual IEEE International Symposium and Workshop on*, pp. 112–122, 2006.

[5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, 220(4598):671–680, May 1983.

[6] V. Cerny, "Thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, 45(1), 1985.

[7] D. Mitra, F. Romero, and A. Sangiovanni-Vincentelli, "Convergence and finite-time behavior of simulated annealing," *Advances in Applied Probability*, 18:747–771, 1986.

[8] M. Locatelli, "Convergence of a simulated annealing algorithm for continuous global optimization," *Journal of Global Optimization*, 18:219–233, 2000.

[9] M. Arshad and M. C. Silaghi, "Distributed simulated annealing," *Distributed Constraint Problem Solving and Reasoning in Multi-Agent Systems*, 2004.

[10] D. Darabos, "Convergent ad hoc positioning with simulated annealing," in *WiOpt*, 2007.

FPGA-BASED SYSTEM SUPPORTING ACOUSTIC EVENT DETECTION AND PROCESSING

Gergely KISS
Advisor: Béla FEHÉR

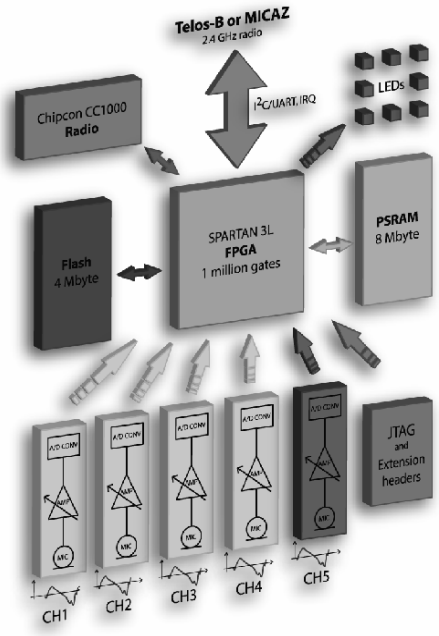
I. Introduction

Hardware-based implementation of acoustic localization algorithms requires a flexible test environment. The Sensor Board 2 is a new custom designed board which has been implemented to be the basis of a joint research project of several universities. A high performance wireless sensor network can be built by connecting the board to the Telos-B [1] or Mica2/MicaZ [2] motes supporting distributed signal processing.

II. Structure of the Sensor Board 2

The Sensor Board 2 consists of a high performance FPGA, on-board memories, one precision and four fast analog channels and an ISM-band radio module.

The system is built around the Xilinx Spartan-3L FPGA having 1 million equivalent gates. The configuration process of the FPGA starts automatically from a Xilinx Platform Flash after applying the power source. The digital and analog parts of the board have distinct power supplies. The SB2 can be powered by either the LOGSYS development cable or four AA batteries. System uptime varies between 20-50 hours depending on the application.



The SB2 has two types of memories. The first is a 8 Mbyte (4Mx16) pseudo-SRAM having 70 ns access time, the other is a 90 ns 4 Mbyte (2Mx16) NOR-type flash. Independent connection of the memories makes it possible to create real Harvard architecture. Operating from the on-board 20 MHz quartz crystal oscillator the maximum memory bandwidth for the PSRAM is 107 Mbit/s in single word mode, which means that data from all analog channels can be stored real-time. The PSRAM can be used as sensor data buffer or as temporary storage for intermediate variables. Program code and data storage is possible in the flash.

The Sensor Board 2 has five analog channels. As high temporal resolution is necessary for transient event detection applications, four of them have 12-bit A/D converters supporting sampling frequencies up to 1MSPS. These are operated synchronously from the FPGA. The fifth channel has 16-bit A/D converter and supports sampling frequencies up to 100 kSPS and has separate control lines. It can be used for vehicle tracking where preciseness is more important than speed. All five channels have a two-stage gain amplifier with an overall gain of 165-1815x (44-65 dB) which can be controlled separately for all the channels in 64 steps with digital potentiometers. In order to reduce noise coming from the digital switching parts of the SB2, the analog channels have separate power and ground planes. The SB2 supports two types of microphones by installing or removing a resistor.

The SB2 can be used alone or together with the TELOS-B or Mica2/MicaZ motes. The power supply for the FPGA and the second stage of the analog channels can be switched on and off by either motes. As the first stage is still operational, the TELOS-B can monitor the fifth analog channel and the Mica2/MicaZ the four other channels with their A/D converters via dedicated lines for activity. The SB2 connects to the motes via I^2C, SPI or UART. Several general purpose input/output and interrupt pins of the motes are also connected to the FPGA.

The Chipcon CC1000 radio module can be used for interference based sensor localization algorithm [3] thanks to a very special feature, the high resolution frequency tuning. Measuring the envelope of the received signal strength is necessary for the algorithm and can be done with a 12-bit A/D, which is of the same type that is used for the four analog channels.

The three extension headers on the SB2 allow the connection of optional components.

III. System architecture

The system is built around the FPGA and the drivers for each component must have been implemented first. The drivers for the analog-digital converters, the digital potentiometers, the PSRAM and the flash were created in VHDL paying attention to modularity and standard interfaces. These IPs occupy only small part (<5% of slices) of the FPGA and can be used for different applications.

IV. Sample application – transient event detection and recording

In order to be able to compare the properties of different localization algorithms in MATLAB, real data is required. The full system consisting of Telos-B nodes with attached sensor boards can be controlled via a graphical user interface implemented in MATLAB. The communication is bidirectional, commands can be issued and status messages along with data can be received. The user commands are forwarded to the base station which broadcasts them to the single nodes. Data from the nodes can be collected similarly, using the base station as a bridge. Time synchronization is implemented in the motes. The SB2 and the Telos-B communicates via UART and dedicated lines for time critical signaling (eg. requesting timestamp). When a start command is issued by the user, all the sensor boards start sampling and the detectors on each channel are looking for transient events. Meanwhile, a short period of samples is stored in temporary circular buffers. If the detectors signal the onset of a transient event, the contents of the buffers are copied to the SRAM along with the new samples, until the detection of the end. Late false detections are eliminated by disabling detectors after most of the sensor boards detected. Data from SRAM is forwarded to the base station on request, and also stored in flash with metadata (recording size, timestamp) and can be read out later directly to a PC.

V. Results, future plans and acknowledgement

The newly developed SB2 is working properly. Future plans include collecting data with the created system and evaluating multiple acoustic source localization algorithms based on the recordings. The ultimate goal is the hardware implementation of these algorithms inside the FPGA.

I am grateful to Mo Azimi who made it possible to spend a year at Colorado State University, Fort Collins, USA as a research associate.

References

[1] Telos-B platform, URL: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf

[2] MicaZ platform, URL: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf

[3] G. Miklos Maroti, Branislav Kusy, Gyorgy Balogh, Peter Volgyesi, Karoly Molnar, Andras Nadas, Sebestyen Dora, Akos Ledeczi: *Radio Interferometric Geolocation*, SenSys 2005, San Diego, September 2005.

HIGH PRECISION FLOW METER USING FPGA TECHNOLOGY

János LAZÁNYI
Advisor: Béla FEHÉR

I. Introduction

This paper gives an introduction to the most widely used ultrasonic *Time-Of-Flight* (**TOF**) measurement methods based on the simplest *threshold technique* and their implementation possibilities using FPGA technology. The main objective was to build a precise but inexpensive handheld ultrasonic flow meter. Although this paper focuses mainly on the theoretical background of ultrasonic flow meters, some part of the practical implementations will be presented too.

II. Ultrasonic Flow Meter Theory

The most widely used ultrasonic flow meter structure is depicted on *Figure 1*[6]. There are two ultrasonic transducers located at a distance *L*. Initially *TR 1* is operating as a transmitter and *TR 2* as a receiver. The ultrasonic source transmits a burst of ultrasonic pulses, and *TR 2* receives them. After the signal packet decays following multiple reflections from all surfaces, the roles are commutated, *TR 2* acts as an ultrasonic source, and *TR 1* as a receiver.

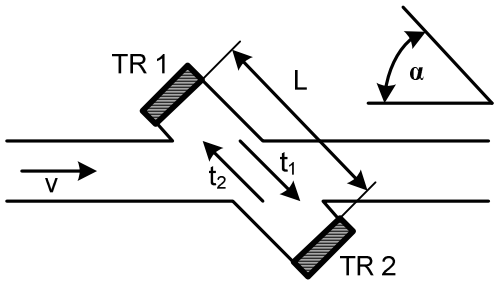


Figure 1: Ultrasonic flow meter

Transit time of ultrasonic signal can be expressed as the following: (***t1, t2*** are the time-of-flights, ***L*** is the distance of transducers, ***c0*** is the speed of sound at the given conditions, ***v*** is the velocity of the gas (flow) traveling through the tube and ***α*** is the inclination angle of the transducers.)

$$t_1 = \frac{L}{c_0 + v\cos\alpha} \qquad t_2 = \frac{L}{c_0 - v\cos\alpha} \tag{1}$$

There are two existing techniques to determine ***v***. Assuming that $c_0^{\,2} >> v^2\cos^2\alpha$, the flow can be expressed by the *difference* of the two TOF.

$$v \approx \frac{c_0^{\,2}}{2L\cos\alpha}(t_2 - t_1)\,. \tag{2}$$

The system under development will be used in spirometry applications (meaning the measurement of respiration), where the basic lung functions are assessed. [1] Speed of sound wave (***c0***) is dependent on the gas traveling through the tube, and it may vary during a measurement cycle. The difference between the speed of a sound traveling through 36 °C (exhaled, body temperature) and 21 °C (inhaled, room temperature) air at normal atmospheric conditions is around 9 m/s. This value is in

the range of the air-flow under measurement (***v***), hence *Eq. 2* cannot be used to determine the flow speed, because it depends on c_0.

Alternatively flow speed can be expressed as *Eq. 3.* In this case flow measurement is independent from the speed of sound, but we need the *absolute* value of the two TOF.

$$v = \frac{L}{2\cos\alpha}\frac{t_2 - t_1}{t_1 t_2}. \tag{3}$$

As a conclusion successful operation of most ultrasonic ranging systems relies on accurate time-of-flight measurements. [2]

III. Time Of Flight Measurement Techniques

Figure 2. illustrates a typical waveform recorded at the receiver output. Multiple echoes can be recognized, since the physical dimensions of the handheld flow meter are commensurable with the wave-length ($\lambda = \frac{c}{f}$). During the tests, cheap f=40 KHz unfocused transducers were used, where $\lambda \approx 8.5$ mm.

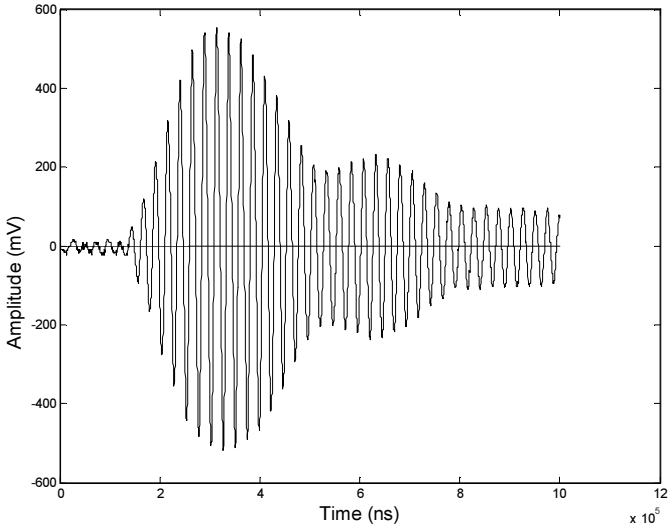


Figure 2: Typical received signal after amplification
(The transmitter was transmitting a burst of 7 pulses, starting from 0 ns.)

A. Threshold method

One of the conventional ways to determine the TOF is the threshold method. This routine is simple and fast. The TOF is determined when received signal exceeds a threshold level for the first time. Obviously, this level must be over the noise level. The main problem with this method is, that the TOF obtained by the measurement is larger than the actual TOF, which corresponds to the starting point (onset) of the echo signal. This is a consequence of the relatively long rise-time of the echoes produced by currently available low-bandwidth ultrasonic transducers for operation in air, and renders more difficult to measure the absolute value of TOF need in *Eq 3*. [2][3]

The main benefit of the threshold method is it's simplicity. There is no need to digitize the analogue signals; hence a simple comparator is enough to measure TOF. *Figure 3* demonstrates a sample block diagram. Modern microcontrollers include programmable capture units, with integrated comparator functionality enabling to implement all the digital functionalities inside a single microcontroller.

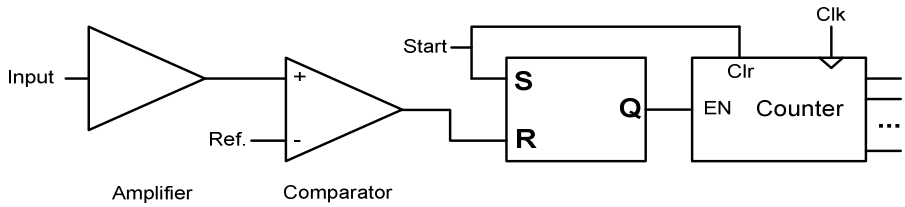


Figure 3: Typical system using threshold method

IV. Improved threshold method

Precision can be improved by averaging multiple samples. Multiple reflections and inertia of the transducers limit the number of bursts to be transmitted down to some hundreds of measurements per second. One of the criteria during spirometer design is the fast response time; hence subsequent measurements cannot be averaged to improve precision.

Time-Of-Flight can be separated into two parts, integer number of wavelengths plus a part representing the phase.

$$t = (n + \frac{\varphi}{2\pi})T\,, \quad T = \frac{1}{f} \tag{6}$$

Where: t time of flight, n integer number of wavelength, φ phase of the received signal, f the frequency and T the corresponding period of time of the received signal.

The system mechanics can be designed in a way, that at zero flow $t = (n + 0.5)T$, ($\varphi = \pi$). Ensuring that the flow will not result in bigger phase than π, TOF measurement can be implemented with phase measurement, using a simple counter with a period equal to the transducer.

Phase value can be evaluated multiple times and averaged during a single measurement.

$$t = (n_k + \sum_{i=1}^{N} \mathrm{mod}_T\, t_i)T\,. \tag{7}$$

Where: N Number of phase evaluations, n_k known integer number of wavelength, t_i elapsed time.

As 40 KHz transducers were used the period is relatively large which results in rare phase-hop. (With the given mechanics the first hop will occur at v≈ 14 m/s) As the human body cannot generate rapid flow change, phase-hop can be detected.

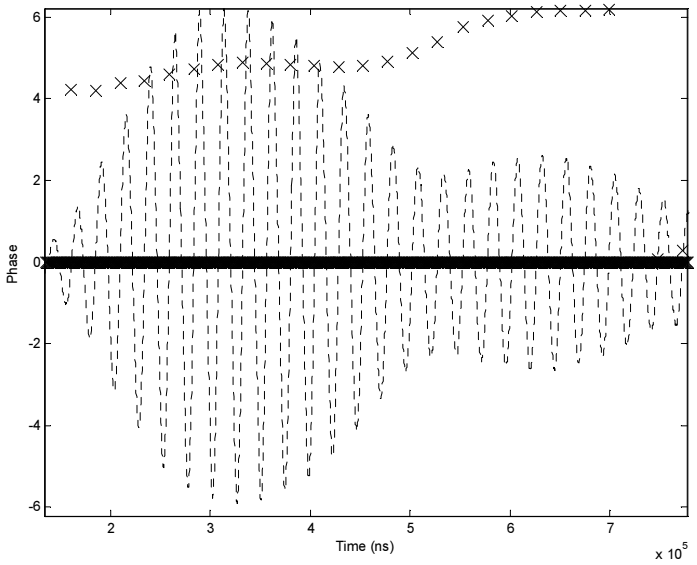


Figure 3: Calculated phase values

Figure 3. shows the calculated phase values at each period (large cross). When the burst signal starts to excite the transducer, it oscillates at a slightly lower frequency due to its inertia. Starting from the 8th period, several phase values could be averaged to gain precise TOF value. Around the 13th period reflections arrive, resulting in additional phase shift.

A. Determination of the received signal frequency

Eq. 6.-7. uses the frequency of the received signal. Measurements show, that the receiver transducer is oscillating at its own resonant frequency, rather than the frequency of the ultrasonic transmitter. This can be proven by a Dirac-like, single-pulse excitation.

There are several methods to determine the exact resonance frequency, ranging from traditional impedance measurement to four parameter sine-fitting.

V. Results

A *phase capture unit* implementing the above described measurement algorithm was synthesized in a Spartan-3 FPGA running at 50 MHz in a fully configurable manner. Transducer oscillation frequency, transmit burst width, phase capture starting period and burst repetition time can be set through PC software. To test the algorithm 1000 cycles have been captured with the following settings: burst length: 5 cycles, phase capture 12th -15th period (best four), flow speed 0.

The first row shows the mean of time elapsed (in unit of 20 ns) until the given period zero crossing has been detected.

Table 1: Measurement results (Unit: 20ns)

	12th period	13th period	14th period	15th period	**Average of four periods**
Mean of 1000 measurements	15,541.68	16,768.59	17,996.64	19,222.52	-
Variance of 1000 measurements	11.23	8.03	7.03	6.80	**3.31**
Difference between max-min values	30.00	22.00	21.00	16.00	18.25

Table 1. shows that by averaging the phase of four consecutive periods the variance of the phase measurement significantly reduces, resulting in more precise TOF measurement. It is important to mention, that the spectra of the additive noise is Gaussian-like.

The phase capture unit was connected to an embedded processor -synthesized inside the FPGA- through *On-Chip Peripheral Bus* (***OPB***) [5]. Moreover the FPGA contains internal memory, graphical LCD driver and computer interface resulting in a real single-chip low cost embedded solution without using A/D chip.

References

[1] Spirometry – Wikipedia, URL: http://en.wikipedia.org/wiki/Spirometry

[2] B. Barshan, "Fast processing techniques for accurate ultrasonic range measurements", *Measurement Science and Technology*, vol.11, no.1, pp.45-50, January 2000.

[3] R. Queiros, P. S. Girao, A. Cruz Serra, "Cross-Correlation and Sine-Fitting Techniques for High Resolution Ultrasonic Ranging", *Proc. of Instrumentation and Measurement Technology Conference,* pp. 552-556, 2006.

[4] B. Barshan and B. Ayrulu, "Performance comparison of four time-of-flight estimation methods for sonar signals", *Electronics Letters*, vol.34, no.16, pp.1616-1617, 6 August 1998.

[5] Platform Studio Documentation – Xilinx Inc., URL: http://www.xilinx.com/ise/embedded/edk_docs.htm

[6] Kouji Ichino, "Socket Structure For Mounting Ultrasonic Gas Flow Measuring Device with Respect to Gas Flow Pipe", US Patent No: 4823612, 1989

MODELING ELECTRICITY CONSUMPTION BASED ON CONSUMPTION LEVELS

Tamás MAKAI
Advisor: Béla PATAKI

I. Introduction

The liberalization of the Hungarian energy market caused a major change in the sales of electricity. Before the liberalization the consumers bought the electric energy directly from the single central power distributor of the country. Presently, they buy it from electricity suppliers, who have to order it from the central power distributor on the preceding workday. If the actual electricity consumption of the consumers, receiving electricity through a given supplier, is more than ordered the missing amount is available only at a higher fee. In the opposite case i.e. the ordered amount is over the actual consumption the whole ordered amount has to be paid for. Therefore a good prediction of the following day's consumption is needed. The prediction of the next day's consumption has to be made for every quarter of an hour. The prediction relies currently only on data collected from previous consumption of the consumer.

There are many different approaches to short term load forecasting, there are several conventional methods like the non-linear models [1], and many that forecasted the load using intelligent methods mainly different types of neural networks [2,3].

For better prediction a good model of the consumption (the long-term behavior of the consumer) and the short term changes (noises) affecting the customer's usual behavior is needed. In this paper a model is shown and the model is evaluated.

II. Detection methods

Weekly time series of two different consumers are shown below (Figure 1).

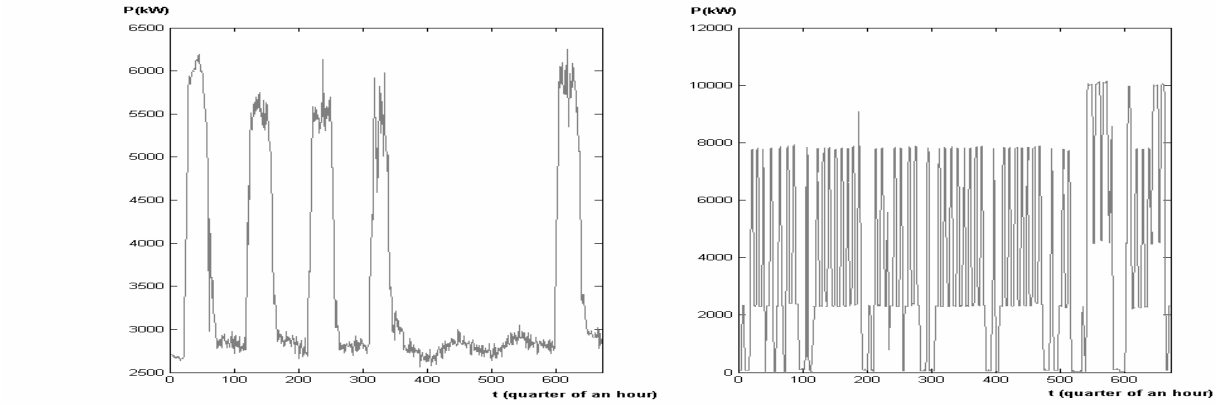


Figure 1: The weekly consumption of two different consumers

For both consumers every quarter of an hour can be assigned to one of some levels, based on the amount consumed during the period. These levels are characteristic to the individual consumers. In Fig. 1a. there are two important levels, the first has a lower consumption. This is the amount of electricity that the consumer uses during night hours, weekends and holidays. The other level has a higher consumption it belongs to daytime consumption on workdays. Similar levels can be identified in the time series of the second consumer. The standard deviation of the consumed energy within the

levels of Fig. 1b. is smaller, but there are more levels than in Fig. 1a. Identifying these levels would help model the electricity consumption of the individual consumers.

The data points assigned to the same level should have two basic characteristics: (a) the amount of electricity consumed during the interval should lay in a close range of a mean value specific to that level and (b) a level should contain several temporally close data points. The first examinations of the time series showed that the classification should be done on a daily basis. Every individual consumer has many different levels with similar means. These levels usually occur on separate days. It would be difficult to separate these levels if several days were investigated together. Further it would be harder to filter out the transients, which arise when the consumer's behavior changes and its consumption switches to another level. Using a daily basis for classification these transients will be assigned to levels containing only a few data points, which supply no relevant information. Finally any method that attempts to determine the levels of several days simultaneously would not work efficiently around the fast and unexpected changes, which occur in the time series. To use data collected from several days is advantageous on weekends, holidays and on any periods when the consumption stays on a lower level for a longer time. The changes, which happen on these days are relatively small compared to the average consumption of the previous days, but are high compared to the average consumption of the weekend.

The algorithm used to assign every quarter of an hour to a level was based on edge detection [4], because between the consumption of two different levels there is always a radical change. The only problem is that in some cases these transients have the same magnitude as the changes inside a level resembling short impulse like noises. This could be compensated by first sorting the 96 values of a day in an increasing order and detecting edges only afterwards. Sorting the data reduces the changes inside every level, but does not affect the transients significantly. A level has to contain at least 5 data points, otherwise it is discarded, and the data points previously assigned to the discarded level are not classified. The algorithm was tested on 300 consumers, the results are shown in the table below.

Table 1: Results of classifying the data points

Maximum amount of levels created for given consumer	1	2	3	4	5	All
Number of consumers	10	83	173	32	2	300
Relative average error	44.8%	11.5%	11.9%	9.6%	9%	12,7%
Unclassified data points	3.7%	5%	5.5%	5.6%	6.4%	5.3%

III. Conclusion

A model of the consumption was created and was validated on the data from 300 consumers. The model proved to be valid for one third of the consumers within the 10% error rate.

After determining the levels belonging to each day, an algorithm will be developed in the future to match the levels from separate days.

References

[1] D. G. Infield, D. C. Hill, "Optimal Smoothing for Trend Removal in Short Term Electricity Demand Forecasting", *IEEE Transactions on Power Systems*, Vol. 13, No. 3, 1998.

[2] M. Espinoza, J. Suykens, B. Moor, "Fixed-size Least Squares Support Vector Machines: A Large Scale Application in Electrical Load Forecasting", *Computational Management Science*, Vol. 3, issue 2, pp. 113-129, 2006.

[3] Guoqiang Zhang, B. E. Patuwo, M. Y. Hu, " Forecasting with artificial neural networks: The state of the art", *International Journal of Forecasting*, Vol. 14, pp. 35–62, 1998.

[4] J. Canny, "A computational approach to edge detection", *IEEE Transactions on Patternt Analasys and Machine Inteligence (PAMI)*, Vol. 8, pp. 679-698, 1986.

ADVANCES IN SPECTRAL OBSERVERS WITH UNEVENLY SAMPLED 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 spectral observer structure is derived from the state-variable formulation and it is essentially a deadbeat state observer. Spectral observers can be used as parametric signal estimators when a measured signal is considered as an output signal of a system represented by state space model. Fig.1. shows the block diagram of the spectral observer structure. A computationally efficient realization is already developed for the regularly sampled case in [2]. The main advantage of this structure is its realizability in hard real-time embedded systems, as the coefficients can be computed offline and only modest amount of calculations need to be performed online.

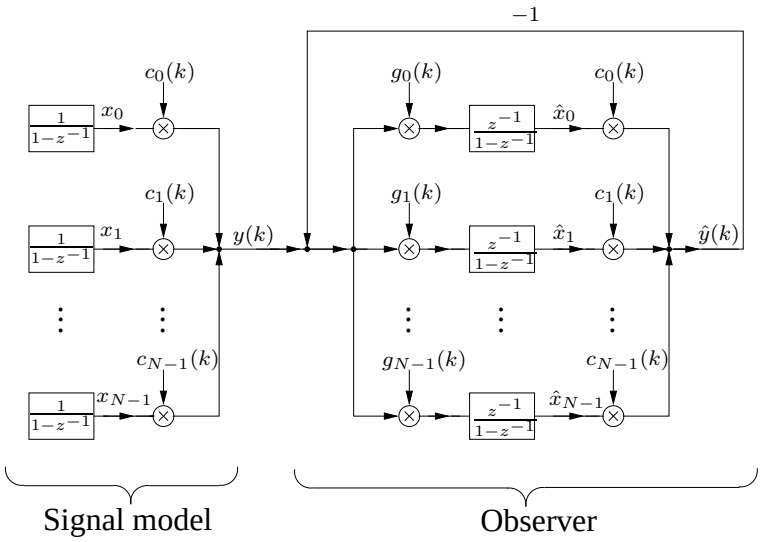


Figure 1: Spectral observer

Unfortunately, the efficient solution heavily rests on the assumption that the input signal is sampled regularly. However, in some cases this assumption does not hold as it is possible that due to implementation issues or resource constraints only uneven samples are available. This problem is critical in embedded and sensor network environments where the designer has to deal with serious resource constraints. First, the processor has to share its limited amount of CPU cycles among its tasks. Hence, it can not be guaranteed that sampling is performed precisely at a given time instant if it is not the highest priority task. Second, in case of wireless sensor networks (WSNs) the communication bandwidth is very limited and error prone, e.g. packet losses can result in missing samples if the signal samples are transmitted between two nodes of the network.

This paper presents the current state of the work in the area of spectral observers with unevenly sampled data. First, in section II. we summarize the foundations of our work. This is useful as the notations introduced here will be used in following sections. In section III. we examine the possibilities of improving an inefficient but theoretically sound solution to the general case of the uneven sampling

problem. In section IV. we present an efficient solution to the special case of periodic uneven sampling. Finally, simulation results are shown and the conclusions are drawn.

II. Spectral Observers with Unevenly Sampled Data

This section presents the foundations of spectral observer theory in order to introduce the notations and the problem domain of our work. The block diagram of the conceptual signal model and the basic observer structure is shown in Fig.1. Let's denote $y(k)$ the sample of the input signal at sampling time instant t_k. Similarly, the values $c_m(k)$ and $g_m(k)$ denote the coefficients of the model at the time instant t_k. Note that in case of regular sampling $t_k = kT_s$ where T_s is the sampling interval.

Several different observers can be realized by the above structure, depending on the values $c_m(k)$. If $c_m(k) = e^{j(2\pi/N)mk}$, then a spectral observer is realized. Let's denote $\mathbf{c}(i) = [c_0(i)\ c_1(i)\ \dots c_{N-1}(i)]^T$ and $\mathbf{g}(i) = [g_0(i)\ g_1(i)\ \dots g_{N-1}(i)]^T$, $\mathbf{x}(i) = [x_0(i)\ x_1(i)\ \dots x_{N-1}(i)]^T$ similarly.

The observer is called deadbeat if $\mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1) = 0 \quad \forall k > N$. From the state equations it can be derived that:

$$\prod_{i=k-N+1}^{k} [\mathbf{I} - \mathbf{g}(i)\mathbf{c}^T(i)] = 0 \quad \Longrightarrow \quad \mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1) = 0 \qquad \forall k > N \tag{1}$$

It is shown in [2], that if the signal is evenly sampled, then (1) is satisfied for every basis/reciprocal basis system. In case of the spectral observer $c_m(k) = e^{j(2\pi/N)mk}$ and $g_m(k) = \frac{1}{N}e^{-j(2\pi/N)mk}$.

In case of uneven sampling there is no constant sampling frequency, hence explicit time indexing has to be used. If we multiply the exponent by $f_sT_s = 1$ than f_s can be eliminated from the coefficients:

$$c_m(k) = e^{j(2\pi/N)mk} = e^{\frac{j2\pi f_s}{N}m(kT_s)} = e^{j\omega_1 m t_k}$$

where $\omega_1 = (2\pi f_s)/N$ is the circular frequency of the resonator with lowest nonzero circular frequency (the first spectral component).

It is important to note that observer theory assumes deterministic signals (i.e. signals without noise). The model does not address the problem of noise or measurement uncertainity in the theoretical level. Hence, theoretical results must be also examined from this point of view as these problems inevitably occur in practical applications.

III. General Case - Observer Gain Calculation Algorithm

In [3] Hostetter proposed an algorithm that ensures the deadbeat property of the spectral observer working with arbitrarly sampled data. This solution uses the same observer structure, the only difference is that the observer gain values $\mathbf{g}(k)$ are not known a priori, they are calculated online for every new time instant. The system equations of this structure are equivalent to the system depicted in Fig.1.

The proposed algorithm is proved to ensure a deadbeat observer (see [4] for the proof), but the amount of required online calculations makes this solution unrealizable in embedded or WSN environments. A promising goal of our future work is to develop an efficiently realizable variant of this algorithm.

IV. Special Case - Periodic Uneven Sampling

Although an efficient spectral observer is not known so far for the general case, we show that it does exist for a special case, namely if there are repeating patterns in the sampling time instants. If there is periodicity in the sampling process, then the sampling time instants follow each other with arbitrary difference, but the time difference between t_k and t_{k+N} is constant: $t_{(k+N)} - t_k = T_p \quad \forall k = 0..\infty$.

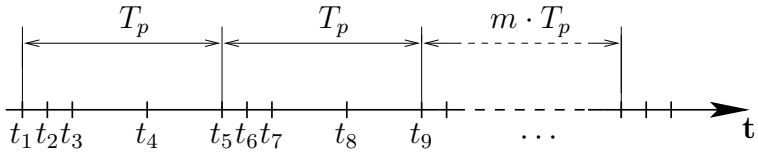


Figure 2: Periodic uneven sampling, N=4

Fig.2 shows an example of periodic uneven sampling with $N = 4$, where $t_1, t_2, ...t_i$ are the sampling time instants and T_p is the period. This situation, when a pattern N long is repeated periodically in the sampling time instants is realistic in certain applications, e.g. in case of burst sampling.

If the above property is true and the repeating pattern is known a priori, then the spectral observer can be utilized to reconstruct the measured signal in a deadbeat fashion i.e. with zero error. Denote $\mathbf{C}(k)$ the N x N matrix formed from the last N $\mathbf{c}(k)$ column vectors: $\mathbf{C} = [\mathbf{c}(k)\, \mathbf{c}(k-1)\, \mathbf{c}(k-2)\, \ldots\, \mathbf{c}(k-N+1)]$ and similarly $\mathbf{G} = [\mathbf{g}(k)\, \mathbf{g}(k-1)\, \mathbf{g}(k-2)\, \ldots\, \mathbf{g}(k-N+1)]$.

With these notations it can be easily shown that if we choose $\mathbf{G}(k)$ to statisfy $\mathbf{G}(k) = \mathbf{C}(k)^{-1}$, then Eq. 1 is satisfied $\forall k > N$. As $\mathbf{C}(k)$ is an N x N matrix formed from the last N $\mathbf{c}(i)$ column vectors ($i = k \ldots k-N+1$), therefore $\mathbf{C}(k+1)$ is formed from the column vectors $\mathbf{c}(i), i = k+1 \ldots k-N$, i.e. it is a shifted version of the columns of $\mathbf{C}(k)$ except for the last column. If we ensure that the last column of $\mathbf{C}(k+1)$ is the same as the first column of $\mathbf{C}(k)$ for all k, than all $\mathbf{C}(i)$ matrices will consist of the same columns, but they are circularly shifted. In this case all of the corresponding inverse matrices $\mathbf{G}(i)$ will consist of the same rows which are shifted circularly. This means that the $\mathbf{G}(k)$ can be calculated offline and the actual $\mathbf{g}(k)$ values can be read out easily.

In the following, we show that the columns $\mathbf{c}(i+N)$ and $\mathbf{c}(i)$ are equal if and only if the time difference between the sampling time instants t_i and $t_i - N$ is a multiplier of T_1, the period of the lowest frequency resonator.

$$
\begin{aligned}
\mathbf{c}(i+N) = \mathbf{c}(i) \iff c_m(i+N) &= c_m(i) \qquad \forall m \in 1 \ldots N-1 \\
e^{jm\omega_1 t_{i+N}} &= e^{jm\omega_1 t_i} \\
jm\omega_1 t_{i+N} &= jm\omega_1 t_i + k2\pi \qquad k \in \mathbf{N} \\
t_{i+N} &= t_i + \frac{k2\pi}{m\omega_1} = t_i + \frac{kT_1}{m} \\
t_i - t_{i-N} &= k\frac{T_1}{m} \qquad \forall m \in 1 \ldots N-1 \qquad k \in \mathbf{N}
\end{aligned}
$$

This has to be true for all $m \in 1 \ldots N-1$. But as k can be arbitrary, if the property is true for $m = 1$ then it is true for all other m, as an appropriate k multplier can always be found to satisfy the equation. We can summarize this as follows:

$$
c(i+N) = c(i) \iff t_i - t_{i-N} = T_p = kT_1. \tag{2}
$$

An attractive property of the spectral observer is that the frequencies of the resonators can be tuned easily by using the Adaptive Fourier Analyzer (AFA) algorithm proposed in [5]. Combining the AFA with the above proposed observer gain calculation algorithm the property (2) can be ensured by tuning the frequency of the first resonator to $\frac{1}{T_p}$.

Another condition that falls out from the above deduction is that exactly N arbitrary samples are needed from the T_p period. If there are more samples available, N of them has to be chosen in a way that the N different $c(k)$ vectors are linearly independent of each other to ensure that the $C(k)$ matrix is invertible.

It is important to point out a possible numerical problem of the solution. In case the $\mathbf{C}(k)$ matrix is ill-conditioned i.e. its condition number is so large that its reciprocal is close to the numerical precision

of the computing machine, then the calculated $\mathbf{G}(k)$ values can be very unprecise. Therefore, if there are more than N samples are available from the T_p period, those samples have to be chosen which provide the minimum condition number for $\mathbf{C}(k)$. In case this minimum condition number is still too large, then the spectral observation is not possible by the proposed solution.

V. Simulation Results

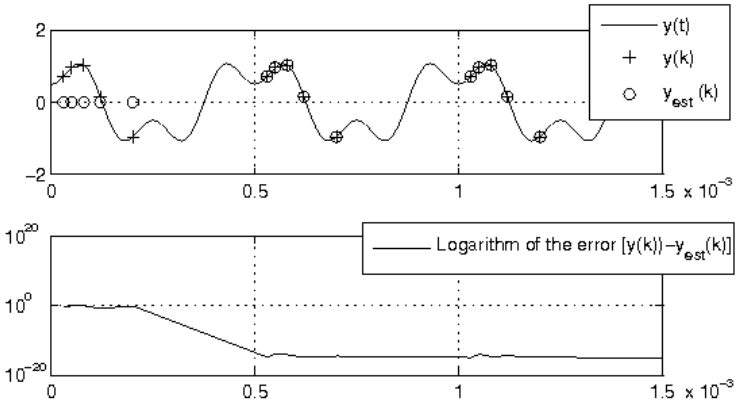


Figure 3: The estimated signal samples and the error of the observer $[y(k) - \hat{y}(k)]$

In Fig. 3 we present an example with an input signal sampled periodically and unevenly ($N = 5$). The input signal is a simple periodic signal formed from two sinusoid components, a base harmonic and the third upper harmonic. The original analog signal $y(t)$ is plotted with a solid line, the sampled signal $y(k)$ is represented by the "+" signs, and the output of the observer $\hat{y}(k)$ is shown by the "o" signs. The observer error $[e(k) = y(k) - \hat{y}(k)]$ is plotted with a logarithmic scale. It is clear that the observer error is reduced to zero (within numeric precision) after N steps, proving the deadbeat property.

VI. Conclusion and Future Work

The well-known spectral observer structure can be modified to work with unevenly sampled data easily. The algorithm mentioned in section III. is an inefficient but formally correct solution to realize a spectral observer with arbitrary signal samples. The main reason of the inefficiency is the online calculation of the observer gain values. Section IV. shows an efficient solution to this problem for the special case of periodic uneven sampling, where the observer gain values can be calculated offline.

In the future, first we will examine the mentioned numerical limitation of the solution proposed in section IV. and work out a formal bound on the usability of the solution. Second, we will try to develop an efficient and realizable algorithm based on Hostetter's algorithm for the general case.

References

[1] G. H. Hostetter, "Fourier Analysis Using Spectral Observers," *Proceedings of the IEEE*, 68(2), February 1980.

[2] G. Péceli, "A Common Structure for Recursive Discrete Transforms," *IEEE Trans. on Circuits and Systems*, CAS-33(10), October 1986.

[3] G. H. Hostetter, "Recursive Discrete Fourier Transformation with Unevenly Spaced Data," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-31(1):206–209, February 1983.

[4] K. Molnár, "Correctness Proof of an Observer Gain Calculation Algorithm for Unevely Spaced Data," Tech. rep., Budapest University of Technology and Economics, to be published in 2007.

[5] F. Nagy, "Measurement of Signal Parameters Using Nonlinear Observers," *IEEE Trans. on Instrumentation and Measurement*, IM-41:152–155, February 1992.

TESTBED FOR WIRELESS ADAPTIVE SIGNAL PROCESSING SYSTEMS

György OROSZ
Advisors: Gábor PÉCELI, László SUJBERT

I. Introduction

Nowadays the investigation of wireless sensor networks' (WSN) application is a relevant research topic, because WSN are utilized with success in various fields [1]. In this paper a testbed is introduced, which can be a common base of current investigations on the applicability of wireless sensor networks in closed loop multiple-input multiple-output (MIMO) control systems. The proposed generic application for testing is an active noise control (ANC) [2] system, the feedback part of which comprises a WSN for sensing the noise to be suppressed. Because of the inherent and inevitable problems of radio communication (e.g. packet loss, uncertainty in data transfer time) the deployment of WSN in closed loop applications is not spread yet [3]. The novel feature of this testbed is that it offers a framework for the investigations to eliminate the disadvantageous effects of radio communication.

This arrangement poses strict demands on the main components of the testbed (WSN and ANC algorithms) for ensuring the overall reliability of the system: the sensor network has to provide efficient data transmission towards the signal processing algorithm, which in turn has to be adaptive for the anomalies in data transfer. Because of the wide variety of ANC algorithms and their particular sensitivity against anomalies in the feedback loop, this system is appropriate for test purposes and comparative analysis of results. Major topics that can be studied in this field with the testbed are summarized in this article.

II. The proposed testbed

The block diagram of the testbed can be seen in Fig. 1. The system basically consists of two units. The main parts of signal processing algorithms are implemented on a DSP evaluation board of type ADSP-21364 EZ-KIT LITE [4], which includes an ADSP-21364 (SHARC) processor. The DSP is a 32 bit floating point one with a maximal clock frequency of 333 MHz. The DSP is connected to an AD1835 codec that has two analog input- and eight analog output channels, through which signal can be fed to the loudspeakers. The relatively great number of output channels ensures the possibility of realization of extensive systems. Other loudspeakers can be utilized to generate external sound/noise for test purpose.

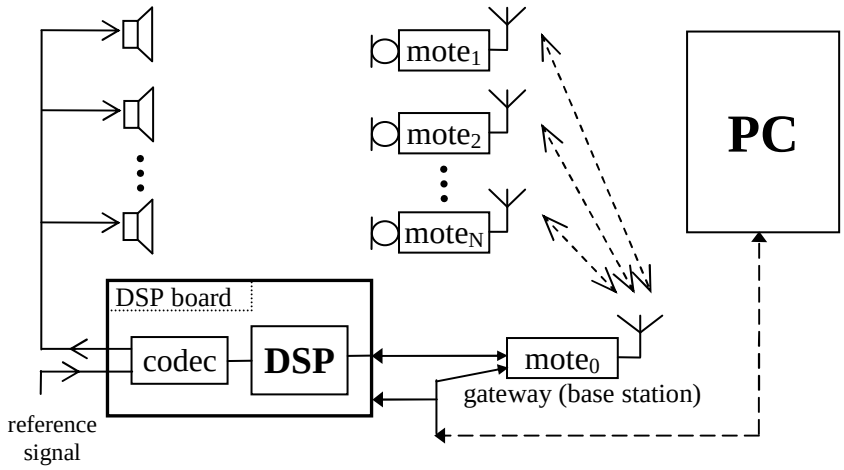


Figure 1: Block diagram of system

The acoustic signal is sensed by the WSN ($mote_1$…$mote_N$), which is built up of Berkeley micaz motes [5]. These motes are intelligent sensors that consist of an ATmega128 eight bit

microcontroller, a CC2420 ZigBee compatible radio transceiver and an MTS310 sensor board. The sensor board includes a microphone, as well, the output signal of which is converted to digital domain by a 10 bit analog to digital converter (ADC) of the microcontroller. A mote ($mote_0$) serves as gateway between the WSN and DSP. The software for motes is developed in TinyOS embedded operating system, but modifications should be carried out in order to decrease the code efficiency.

The PC basically serves as developing and debugging tool for both platforms.

III. Utilization possibilities of testbed

The testbed is suitable for investigating how a WSN can be applied in fast digital signal processing- and closed loop control systems. Topics that can be investigated are listed below:

- Algorithm fitting for specialities of wireless data collections: e.g. handling of data loss and non-equidistant signal transmission are fundamental problems, since traditional algorithms are not prepared for these anomalies.
- Synchronization: crucial task, since the uncertainty in the time instants of the sampling can cause the instability of the whole system. This makes the ANC system appropriate for testing the synchronization algorithms, but the handling of unsynchronized sensors' data on data processing algorithms' side can be investigated, as well.
- Distributed signal processing: the benefit of preprocessing of data is twofold. It can reduce the amount of data to be transmitted, and the extension (e.g. decomposition) of signal processing algorithms on WSN can lead to robust systems, since primary data are processed/transformed (e.g. Fourier transformation) on motes, so primary data loss caused by pocket loss in radio network can be avoided. In connection with this topic, algorithms that work directly on these transformed domain data can be studied.

IV. Results

Recent experiments on the testbed proved its operability, the cooperation of WSN and signal processing algorithm was appropriate. The system contained four motes – operating with the sampling frequency of 1.8 kHz – and four loudspeakers, but the resources make possible further extension of number of the sensors and actuators. As noise source we applied a loudspeaker driven by a signal generator. The noise was measured by an external microphone – placed at the noise sensing sensors – the output signal of which was processed with a digital oscilloscope with FFT function. The noise suppression was nearly the same as in the case we applied wired sensors.

As this example shows the utilization of WSN in closed loop applications can be a feasible goal, and this testbed can serve as a platform for investigations in the field of wireless control. Further investigations will aim at the extension of the system and the reduction of constraints and fault sources emerging due to the wireless data acquisition.

References

[1] Akyildiz, I. F., W. Su, Y. Sankarasubramaniam, and E. Cayirci, „Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393-422, 2002

[2] Kuo, S. M., D. R. Morgan, „Active Noise Control: A Tutorial Review," in *Proceedings of the IEEE*, vol. 87. No. 6., pp. 943-973, June. 1999.

[3] Mathiesen, M., G. Thonet, N. Aakwaag, „Wireless ad-hoc networks for industrial automation: current trends and future prospects," in *Proceedings of the IFAC World Congress*, Prague, Czech Republic, July 4-8, 2005

[4] Analog Devices Inc., *ADSP-21364 SHARC EZ-KIT Lite Evaluation System Manual*, One Technology Way, 2005

[5] *MPR-MIB User's Manual*, www.xbow.com/Support/Support_pdf_files/MPR-MIB_Series_Users_Manual.pdf

MAXIMUM LIKELIHOOD ESTIMATION OF NOISE VARIANCE AND CODE TRANSITION LEVELS IN ADC TESTING

Attila SÁRHEGYI
Advisor: István KOLLÁR

I. Introduction

In 1965, Gordon Moore predicted that the number of transistors on a chip doubles about every two years. Since computers are digital systems and the world is analog, the interface between them also becomes more and more important. Moore's law is valid not only for microchips but also for our everyday life. We are surrounded by more and more electrical facilities e.g. when doing the laundry or keeping track if the milk has gone bad. Due to this quick growth, companies are developing newer and newer analog to digital converters to meet the requirements. Hence fast, efficient, and standardized test methods are essential. Moreover, these would reduce the time needed for the buyers of ADCs to validate the performance of the circuits they bought. This is what Standard IEEE-1241 is made for.

Imagine that we have a 12-bit A/D converter. It has $2^{12} = 4096$ output codes and between them $2^{12} - 1$ comparison levels, called code transition levels. One of the purposes of ADC testing is to determine these levels. Once code-transition levels have been measured, then all static parameters, including integral and differential nonlinearities, missing codes, gain, and offset can be computed. Code transition levels are defined as [1] *Quantitatively, a code transition is the value of the converter-input parameter which causes half of the digital output codes to be greater than or equal to, and half less than, a given output code.* There exist some procedures to directly measure these but they are either time-consuming (about 1 hour for all levels of a 12-bit ADC) or prone to dynamic errors.

In this paper a prospective statistical test method to determine all parameters of the ADC and of the input signal is analyzed. In Figure 1 the output signal of a one-bit quantizer can be seen in response to a linear ramp input signal $y(t)$ with some additive noise $\xi(t)$. The input signal is not necessarily linear,

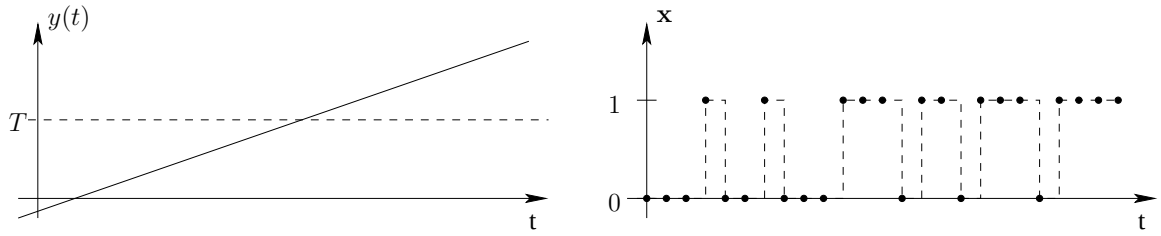


Figure 1: A code transition of an ADC, T is the code transition level.

it could be any arbitrary function which can be described by a small parameter vector $\mathbf{w}$. The main goal is to create an estimator which gives a "good" estimation for the noise variance and the transition level(s) from an input-output data series with the help of estimating also the input signal via estimating $\mathbf{w}$. We will use the maximum likelihood method (MLE), because of its desirable properties.

II. The quantization of the probability density function

Before the analysis let us create the model. The input signal is $y(t)$, and a noise $\xi(t)$ is superposed to it. $\xi(t)$ is Gaussian distributed with zero mean and variance σ^2, $\xi(t_i) \sim N(0, \sigma)$ with identical and

independent probability density functions (PDFs):

$$f(z_i) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{1}{2}\left(\frac{z_i - y(t_i)}{\sigma}\right)^2}. \tag{1}$$

This PDF is quantized at every time instant. In the simplest case (1 comparison level) the quantizer is a step function

$$H(l) = \begin{cases} 0 & \text{if} \quad l \leq 0 \\ 1 & \text{if} \quad l > 0 \end{cases}. \tag{2}$$

$\mathbf{x} = (x_1, x_2, \ldots, x_M)$ is the output signal (i.e. the vector of the observations) which consists of M samples. Thus, equation $x_i = H[y(t_i) - T + \xi(t_i)]$ will describe the operation.

A. Constant noiseless input

Suppose that the the input signal is constant: $y(t) = c$. If so, the noisy input signal can be treated as samples of a Gaussian distribution with mean c. This can be seen in Figure 2 where T is the code transition level which has to be determined. After quantization of this PDF a Bernoulli distribution is

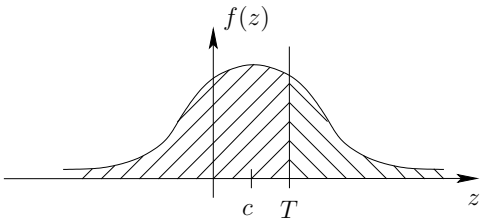


Figure 2: Normal distribution

obtained having two possible outcomes labelled by $n = 0$ and $n = 1$, in which $n = 1$ ($y(t) + \xi(t) \geq T$) occurs with probability p and $n = 0$ ($y(t) + \xi(t) < T$) occurs with probability $q = 1 - p$, where $0 < p < 1$. The probability function is

$$P(n) = \begin{cases} 1 - p & \text{for} \quad n = 0 \\ p & \text{for} \quad n = 1 \end{cases} \tag{3}$$

which can also be written as $P(n) = p^n \cdot (1 - p)^{(1-n)}$. In other words, p is the probability of that the output signal is equal to 1 if the noiseless input signal is equal to c. The relation between the continuous and discrete distributions is as follows:

$$p = P(z > T) = \int_T^{+\infty} f(z)\mathrm{d}z = \frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{T - c}{\sqrt{2}\sigma}\right)\right] \tag{4}$$

where

$$\mathrm{erf}(k) = \frac{2}{\sqrt{\pi}} \int_0^k e^{-\alpha^2} \mathrm{d}\alpha. \tag{5}$$

B. Generalization for $y(t)$ time dependent

Following the line, suppose that $y(t)$ changes with t. Since discrete samples are treated, we will use symbol y_i for $y(t_i)$. Thus, in this case the probability p is also changing in time. This will be denoted by p_i, given by

$$p_i = \frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{T - y_i}{\sqrt{2}\sigma}\right)\right]. \tag{6}$$

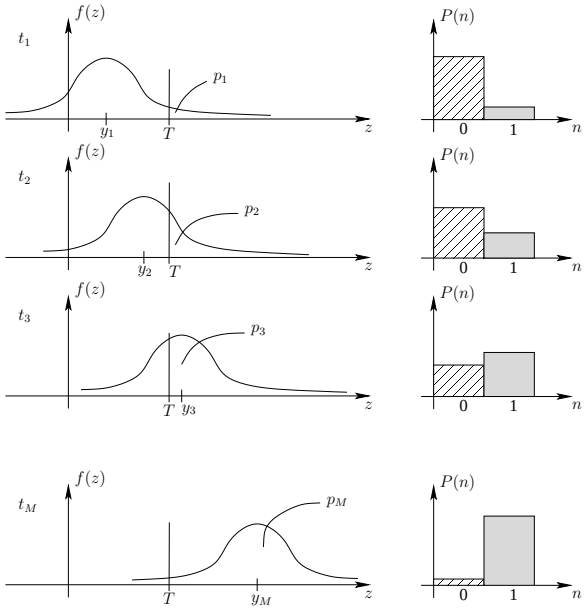


Figure 3: The relation between the continuous and discrete distributions

Figure 3 shows the relations between the continuous and discrete distributions for different time instants. In the first column a series normal distributions can be seen which is transformed into a discrete Bernoulli distribution series (second column) due to quantization. T is the code transition level which has to be determined. Each row is a particular solution of Eq. (6) at instant t_i which results in probabilities $p_1, p_2, p_3 \dots p_M$. According to Eq. (3) these are the probabilities $P(y(t_i)+\xi(t_i) \geq T)$=$P(n_i = 1)$.

C. Numerical example

Suppose that the input signal is $y(t) = 0.24 \cdot t + 0.1$, the code transition level is $T = 1.6$ and $\sigma = 0.3$. According to Eq. (6) the calculated probability is plotted against time in Figure 4. It can be seen that if

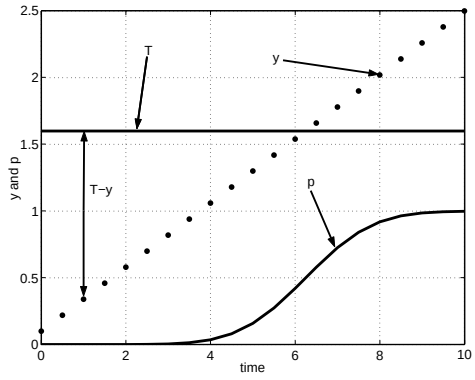


Figure 4: The probability p as a function of time

$y(t)$ is far below the transition level, p is small which is reasonable because in this case the probability of exceeding the transition level is negligible. Conversely, if the probability p is given, the difference $T - y$ is also fixed. Furthermore, since the input signal y is estimated, the transition level T can also be estimated. The shape of the p-curve depends not merely on the transition level and σ but also on the shape of the input signal. Thus, the group of parameters (T, σ, y_i) and the probability p_i mutually determine each other.

III. The likelihood function

Finally, let us determine the likelihood function which yields the MLE for the parameters of the input signal, for the transition level and σ:

$$\max_{\mathbf{w},T,\sigma} L(\mathbf{w},T,\sigma|\mathbf{x}) \Longrightarrow \hat{\mathbf{w}},\hat{T},\hat{\sigma}.$$

This is executed in two steps. The relation $(T,\sigma,y(\mathbf{w},t_i)) \leftrightarrow p_i$ has already been considered. The second step is to estimate the likelihood function from the p_i probabilities i.e. $L(\mathbf{w},T,\sigma|\mathbf{x})$ and maximize this likelihood function according to $\mathbf{w},T,\sigma$ parameters.

From a statistical viewpoint, the data vector $\mathbf{x} = (x_1, x_2, \ldots, x_M)$ is a random sample from an unknown population. [2] The goal of data analysis is to identify the population (defined by the parameters) that is most likely to have generated the samples. As the parameters change in value, different probability distributions are generated. Formally, a model is defined as the family of probability distributions indexed by the model's parameters. So, if a set of parameter values is given, the corresponding PDF will show that some data are more probable than other data. However, we have already observed the data. Accordingly, we are faced with an inverse problem: given the observed data and a model of interest, find the PDF, which is most likely to have produced the data. To solve this inverse problem, we define the likelihood function by reversing the role of the data vector $\mathbf{x}$ and the parameter vector $\mathbf{u} = (\mathbf{w},T,\sigma)$ in $f(\mathbf{x}|\mathbf{u})$ i.e. $L(\mathbf{u}|\mathbf{x}) = f(\mathbf{x}|\mathbf{u})$. Thus, $L(\mathbf{u}|\mathbf{x})$ represents the likelihood of the parameter $\mathbf{u}$ if the observed data $\mathbf{x}$ is given.

$$L(p_i|x_i) = \prod_{i=1}^{M} \left[p_i^{x_i}(1-p_i)^{1-x_i}\right] \quad x_i = (0,1). \tag{7}$$

The difficulty of this estimation is that the input signal is not constant. Hence, only the sample x_i belongs to probability p_i and its statistics cannot be created by one sample. Moreover, the probabilities p_i depend on each other due to the mean value changing deterministically. This makes the estimation unnecessarily difficult. And last, we are interested in the parameters $\mathbf{w},T,\sigma$ and not in p_i. These problems can be solved if p_i is substituted by Eq. (6):

$$L(\mathbf{w},T,\sigma|\mathbf{x}) = \prod_{i=1}^{M} \left[\left(\frac{1}{2}\left[1 - \operatorname{erf}\left(\frac{T-y(\mathbf{w},t_i)}{\sqrt{2}\sigma}\right)\right]\right)^{x_i} \left(\frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{T-y(\mathbf{w},t_i)}{\sqrt{2}\sigma}\right)\right]\right)^{1-x_i} \right]. \tag{8}$$

This function has to be maximized only according to $\mathbf{w},T,\sigma$. Analytically, this is not solvable, thus the next step is to maximize numerically by using MATLAB.

IV. Summary and further work

A quantization model and according to it a likelihood function have been created. It is too complex to obtain a solution in analytic form. In such situations, the ML estimation must be sought numerically using nonlinear optimization algorithms.

The above estimator going to be evaluated and extended for multi-bit quantizers. In that case, the above estimation will become more complicated and numerical problems may emerge in MATLAB. If solved effectively, this will become the "ultimate" solution to the ADC testing problem.

References

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

[2] I. J. Myung, "Tutorial on maximum likelihood estimation," *Journal of Mathematical Psychology*, 47:90–100, 2003.

RANK FILTER FOR HIGH DEFINITION VIDEO APPLICATIONS

Péter SZÁNTÓ
Advisor: Béla FEHÉR

I. Introduction

Rank order filtering is a non-linear filtering technique, which selects an element from an ordered list of tap number of samples. In the two-dimensional (2D) case filtering takes place on the contents of a rectangular window (or more generally, an arbitrary shape), which slides across the image – the number of samples within the filter window hence equals to the tap number. Every time the window is moved by one pixel column, a set of obsolete elements are discarded and a set of new elements are inserted. The samples within the window are sorted and the element with the specified rank replaces the output element of the window. Most typical ranks are median, minimum and maximum, but the selection can be easily tailored to the needs of any application. Compared to other filters – such as FIR, Laplacian or blur filters – rank filters can effectively remove impulse like noises while preserving the edges of the original image. This can be very useful for various applications, for instance removing certain types of transmission noises, or pre-processing for edge detection.

II. Proposed Architecture

When filtering images or video, the filter window is slided horizontally on the input image. In case of a simple rectangular window, to generate a valid output, *WV* (vertical size of the filter window) new input samples should be processed. Therefore, for non bit serial implementations, an important classification criterion is the level of input parallelization.

Word-serial architectures can process one input sample per clock cycle. This is the typical structure for filtering 1D inputs, but it is also applicable for 2D filtering. In this case the filter should operate at *WV* times of the input pixel frequency and generates a valid input sample every WV^{th} clock cycle.

The other extremity is the full-parallel approach – these filters can generate valid output every clock cycle, irrespectively of the number of input samples required to achieve this. Consequently, such filters process *WV* new samples in a single clock cycle. Hence the required operating frequency equals to the input pixel frequency, while at the same time hardware resource requirements are greatly increased. Previous papers typically considered fully parallel architectures as 2D filters, however, as this paper proves, using recent FPGA technologies this solution is sub-optimal due to the inefficient resource utilization.

Multi-word architectures are hybrid solutions: in one cycle they can handle more than one input samples, but less then the full-parallel implementation (from now on, let *NI* denote the number of new input samples in a single cycle). This solution allows finding a good balance between operating frequency and hardware complexity. Using a given filter window and input pixel frequency (FS), the required operating frequency (FO_{max}) can be computed:

$$FO_{max} = FS \frac{WV}{NI} \tag{1}$$

When processing color images using the full per-pixel information (e.g. full RGB or YCbCr values) is not a convenient solution. Filtering these components independently not only increases computational requirements, but may introduce blur effects, as it may generate new color values which were non-existent on the input image. A better solution is to use a magnitude-like value, e.g.

luminosity. If the input format does not contain such a component, it can be generated within the filter.

A. Global Filter Architecture

The proposed architecture consists of five main components (illustrated on Fig. 1): the Line Buffer, the optional Filter Value Generator (FVG), the Delay Line, the Filter Core and the Control Unit (CNTRL). The Line Buffer stores *WV*-1 lines of the original input frame in internal memory.

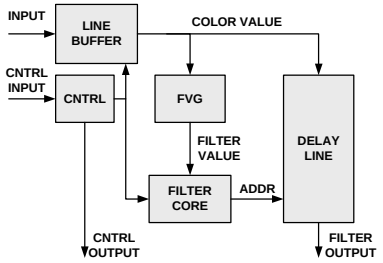


Figure 1: Top level architecture

The Filter Value Generator is only required if the input format does not contain magnitude-like component, for YCbCr or YUV input representations this module can be omitted as the Y component lends itself well for magnitude ordering. For RGB input, luminance, a typical magnitude value can be calculated.

The Delay Line is an addressable FIFO which stores the full per-pixel information of the pixels residing inside the Filter Core. The Filter Core itself uses the values generated by the FVG and generates the appropriate address for the Delay Line. The Control Unit generates properly delayed synchronization signals and output valid signals. Henceforward, only the Filter Core and its extensions are discussed in details.

B. Word-serial Filter Core

The operation of the Filter Core is based on observations introduced in [4]. As a first assumption the filter contains TAP number of different samples. For each sample, an index value is generated, which equals to the number of samples which are smaller than the given sample. This results in TAP distinct values for the TAP samples, which range from 0 (smallest sample) to TAP-1 (largest sample). The ranked sample is the one which has the index value equal to the required rank. The block diagram on Fig. 3 illustrates the hardware implementation of the algorithm for TAP=5. The *D*[3:0] data registers store older filter values, while the new data value is saved into the ND register. In every cycle, these registers shift their data to the left. Older values are compared with the new value (result is '1' if the new value is smaller than the older, '0' otherwise), and the comparison result is saved into the LSB position of TAP-1, TAP wide registers (*CR*[3:0]). The MSB positions of these CR registers are updated with the value of the previous CR register, so:

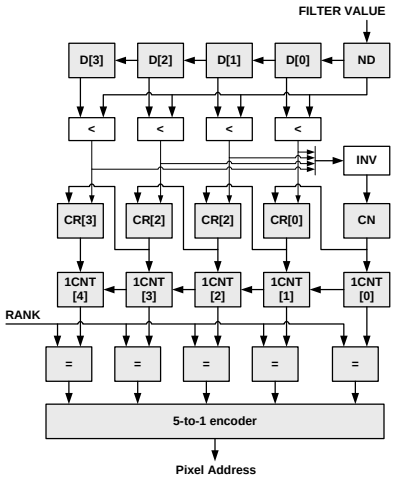


Figure 2: Filter Core

$$CR[k] = \{\, CR[k\text{-}1](TAP\text{-}2{:}0),\ C[k]\} \qquad (4)$$

where (:) denotes bit selection, {} denotes concatenation and $C[k]$ denotes the k^{th} comparison result. Expressively, the comparison result of a given value moves to the left together with the filter value. Therefore, at any given time, $CR[k]$ stores the comparison results of $D[k]$ with all the other values within the filter. The TAP wide register for the new value (CN) is computed differently: it is generated using the negated result of the comparators – namely, the k^{th} bit is updated with the $(k+1)^{th}$ comparison result. The 0^{th} bit (self-comparison) is set to '0'. Counting the '1's in the CR[] and CN registers gives the number of values which are smaller than the given value. This bit summing operation is done by the 1CNT modules. The straightforward way is to use an adder tree with TAP one bit inputs; for the CN register this is the only solution, as its content can change completely from clock to clock. The CR[] registers, however, offers some optimization possibilities. When generating $CR[k]$, only two bits changes from $CR[k\text{-}1]$: the MSB (comparison result with the discarded sample) and the LSB (comparison result with the new value). Therefore, bit summing can be implemented using an incrementer/decrementer. The result of the bit

summing blocks are compared with the required rank, thus generating a TAP bit result containing exactly one '1' at the position of the cell which contains the required output. An encoder passes this position to the Delay Line as an address.

C. Multi-word Filter Core

The architecture presented in the previous section can be easily extended to be able to process more than one new filter values per clock cycle. Instead of one data position, the data registers (*D[]*) and the comparator result shift registers (*CR[]*) should shift by *NI*. The yet single *CN* and *CR* registers become register arrays with NI elements. The number of comparators is increased, as all old samples should be compared with all new samples and new samples should be compared with each other. The required number of comparators for a TAP sized filter with *NI* new samples:

$$C=(TAP-NI)*NI+\frac{NI*(NI-1)}{2} \quad (5)$$

If *WV* is not an integer multiple of *NI*, the bandwidth of the filter core input supersedes that of the input stream, so in some clock cycles the number of valid new data is going to be less than *NI*. The simplest solution to make the filter capable of processing different number of new samples is to insert multiplexers into the appropriate data paths, in front of *D*[], *ND*[], *CR*[] and *CN*[] registers. Two-to-one multiplexers are sufficient, because during the operation of the filter there are only two different scenarios. Either all *NI* inputs are valid, or there are only (*WV* mod *NI*) legal values (see Fig. 3). Thus the size of multiplexers is limited to 2:1, but still a numerous multiplexers are required.

Another solution is to insert padding samples as necessary, so in every clock cycle *NI* new samples are entered, thus creating a virtual filter (from now referred as virtual filter kernel). Fig. 4 illustrates such kernel for the *WH*=3, *WV*=3, *NI*=2 case (*WH*: horizontal size, *WV*: vertical size, *NI*: number of new input samples in a single cycle). Valid samples in the window are marked with light grey; padding samples are marked with dark grey (the actual value of the padding samples is irrelevant). Obviously, this method makes the virtual kernel size larger than the real filter window, hence requires more hardware resources, as parts of the Filter Core scales with the virtual kernel size.

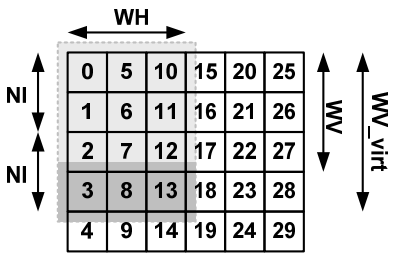


Figure 3: Virtual kernel

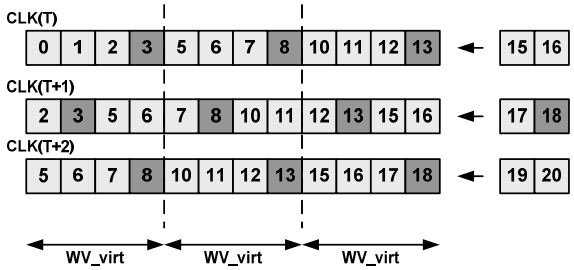


Figure 4: Register masking

Fig. 4 presents the contents of the data registers clock by clock – using the numbers on Fig. 4 – as new inputs are inserted and the filter window is moved horizontally. Valid and invalid (padding) samples are marked just as on the previous figure. Samples on the right are the input samples. As a given cell contains both valid and invalid samples during operation, comparisons are done using all data registers, irrespectively of the validity of the actual sample of the cell, so the number of comparators required scales with the size of the virtual filter kernel. Padding samples are masked after the comparator result registers (*CR*[], *CN*[]), but before the 1CNT blocks. For each older sample, masking is done for 2*NI bits: NI bits mask the comparison results with the NI new samples, and another *NI* bits mask the comparison results of the oldest NI samples. The output ranking part is the same as in the single-word case. The number of required equality comparators scales with the size of the real filter window, as it is sufficient to select the appropriate output when all samples in a new column have been inserted into the filter. In these cycles the locations of the valid samples are well defined.

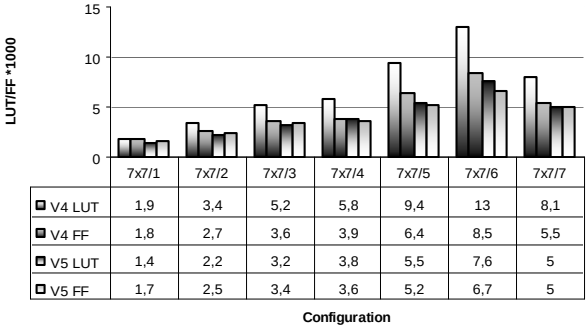
III. Implementation Results

The following implementation results were obtained using 24 bit RGB input, while the FVG was set to sum the three color components and output the 10-bit result. Table 1 summarizes the obtainable operating frequency of the word-serial architecture in different Xilinx FPGA families and different TAP numbers. As the most demanding commercial video format (HDTV 1920*1080p) has a pixel frequency of 75 MHz, the required filter architecture can be easily selected based on this table. For example, a Virtex-4 device can perform real-time filtering on HDTV source using a 49 TAP filter by employing a multi-word Filter Core configuration with 2 input samples per clock cycle.

Fig. 5 summarizes the resource requirements of a 49 TAP rank filter using different Filter Core configurations (configuration: WVxWH/NI). LUT and FF denote the number of LUTs and flip-flops in Virtex-4 and Virtex-5 devices, respectively.

Table 1: Operating Frequency

Family	TAP number		
	9	25	49
XC5V-3	480 MHz	480 MHz	480 MHz
XC4V-10	400 MHz	400 MHz	350 MHz
XC2V-5	235 MHz	225 MHz	215 MHz
XC3S-4	200 MHz	185 MHz	185 MHz



	7x7/1	7x7/2	7x7/3	7x7/4	7x7/5	7x7/6	7x7/7
V4 LUT	1,9	3,4	5,2	5,8	9,4	13	8,1
V4 FF	1,8	2,7	3,6	3,9	6,4	8,5	5,5
V5 LUT	1,4	2,2	3,2	3,8	5,5	7,6	5
V5 FF	1,7	2,5	3,4	3,6	5,2	6,7	5

Figure 5: Resource requirements

As can be seen on Fig. 5, there are multi-word configurations (such as 7x7/5, 7x7/6) which require more resources than the full-parallel architecture (7x7/7). The reason for this is that the virtual filter kernel becomes way larger than the real filter window due to the enormous number of padding samples. Obviously, these configurations should not be used; however, as can be calculated from Table 1, these are not required even in the slowest FPGAs.

IV. Conclusion

An efficient architecture for high performance two dimensional rank filter was presented. Rank order filters, especially median filters, are used extensively for removing non-Gaussian (salt and pepper) noise from images and video feeds. Compared to previous 2D architectures, the size and complexity of the filter structure was considerably reduced by optimally balancing the number of new input samples entered into the core and the possible operating frequency of the filter. The presented architecture can be further generalized to use arbitrarily shaped filter kernel and to perform weighted filtering.

References

[1] R. Roncella, R. Saletti, P. Terreni, “70-MHz 2-mm CMOS Bit-Level Systolic Array Median Filter”, IEEE Journal of Solid State Circuits, vol 28, 1993.

[2] B. K. Kar and D. K. Pradhan, ``A new algorithm for order statistic and sorting,", IEEE Trans. Signal Processing vol 41, August 1993.

[3] F. A. Suhaib, P. Y. K. Cheung, L. Wayne: “Novel FPGA-Based Implementation of Median and Weighted Median Filters for Image Processing”, Field-Programmable Logic and Applications (FPL 2005), 2005.

[4] C. Chakrabarti, “High Sample Rate Array Architectures for Median Filters”, IEEE Trans. on Signal Processing, vol. 42, March 1994.

MODEL BASED TORQUE ESTIMATION

András ZENTAI
Advisor: Tamás DABÓCZI

I. Introduction

This paper reports advances in the design of electric power assisted steering (EPAS) systems focusing on model based torque estimation. In EPAS systems electrical machines are connected to the mechanical system to reduce the driver's work by generating an assisting torque in the appropriate direction. Nowadays, three phase permanent magnet synchronous machines (PMSM) are used in cars. Steering systems are life critical, therefore they have to be designed by maximizing safety, redundancy and reliability. Automotive environment postulates further special requirements for EPAS applications, as e.g. the usage of multiple but usually low voltage levels, providing high torque at high speeds with small size and high efficiency. The system cost also plays an important role in automobile system design, because in high volume production every additional part may dramatically increase the overall cost. Designing cheap and reliable systems forces the engineers to build in as few sensors as possible and extract as much information as possible from the output of the sensors. One possible realization of this design principle is to implement redundant functions – required for safety checking – without redundant sensors but using the available sensor data and model based simulations to calculate the appropriate value.

Model based torque estimation will be described in the following order: firstly the significance of the torque measurement will be discussed, secondly a physical machine model will be expounded, then phase voltage measurement methods will be presented. Thirdly the parameter estimation methods will be detailed. Finally simulation results, further research possibilities and conclusion will be presented.

II. Torque measurement

The main function of EPAS systems is to reduce steering efforts of the drivers. Mechanical work is a product of steering wheel angular speed and torque. It is only possible to reduce the work of the driver by generating an assisting torque in the appropriate direction, because the required position – and therefore the angular speed – of the steering wheel is determined by the driving situation. Because of the previously detailed reasons EPAS systems use a steering assistance control algorithm based on the torque signal. Torque measurement is costly, therefore there is only one torque sensor built into the system, which measures driver torque. It is not necessary to measure electrical machine torque, because it can be calculated from the stator currents and the rotor angle of the electrical machine (1), where T is the electrical torque of the machine. The other terms are detailed in section III.

$$T = i_q K_{gen} \tag{1}$$

The current measurement sensor is the primary input signal to the torque estimation algorithm. A life critical system can not be controlled by using a single measured signal. Some cheap but effective redundancy has to be built into the system which evaluates measured machine currents and calculates machine torque. It is possible to estimate the phase currents if machine input phase voltages are known and a good machine model is available. This method does not require expensive sensors, but only additional CPU time. The precision of the torque estimation depends on the accuracy of the input signals of the model, the complexity of the model and estimation accuracy of the model parameters.

III. Machine model

To estimate currents, a physical model of the machine is used. This model takes the following into account:

- voltage drop caused by the phase windings,
- voltage induced by the rotor magnets,
- voltage induced by the phase windings, and
- the coupling caused by phase windings.

The model does not treat the following:

- resistance changes of the phase windings caused by thermal effects,
- non-linear saturation effects of the iron parts, and
- magnetic working point changes caused by the heating up of the rotor magnets.

These effects can also be taken into consideration in a more complex model if there is more CPU resource available. In the first approach the accurate knowledge of the model parameters is more important than a more sophisticated model.

All type of e-M@chines can be modeled using several different coordinate systems. The most simple model is described in a coordinate system, where the two axes (d,q) are fixed to the magnetic axis of the rotor as it can be seen on Fig. 1 a. Queer (q) component is perpendicular to the rotor's magnetic field, it is used to generate an effective torque. Direct (d) component is parallel with the rotor's magnetic field and is used to decrease the effect of the rotor's magnetic field on the stator windings (field weakening) [1]. In this model, the current and the voltage signals are not sinusoidal even if the rotor is rotating with a constant speed, because the coordinate system of this model rotates synchronously with the rotor. In one mechanical working point, where the torque and the rotor speed are constant all the currents and the input voltages are constant.

It is also possible to describe the machine in the coordinate system fixed to the phase windings (u,v,w) as it can be seen on Fig. 1 b. This model is closer to the physical model of the machine. In this model, voltage and current signals are sinusoidal, the most important parameter is the phase angle between them.

The two machine models are equivalent. Mostly, the (d,q) model is used, because it makes calculations easier with constant signals. To change from physical model to rotor oriented model it is required to convert the data given in the 3 phase stator oriented reference frame (u,v,w) into a magnetising (d) and a torque producing (q) component. To change from stator oriented to rotor oriented reference frame, Clarke (2) and Park (3) transformations are used, where i_α, i_β are real and imaginary currents in a stator oriented complex reference frame, Θ_r is the angle between stator phase flux direction and rotor magnetic flux direction (d) and i_d, i_q are currents in the rotor oriented reference frame [2][3]. Clarke transformation (2) changes from stator oriented 3 axis reference frame (u,v,w) to stator oriented orthogonal 2 axis frame (α,β). Park transformation (3) rotates stator oriented reference frame (α,β), which results in a rotating reference frame (d,q) that rotates synchronously with the rotor.

$$i_\alpha = \frac{2}{3} \cdot \Re e \left\{ i_u + i_v \cdot e^{j \cdot \frac{2\pi}{3}} + i_w \cdot e^{j \cdot \frac{4\pi}{3}} \right\} = i_u \tag{2}$$

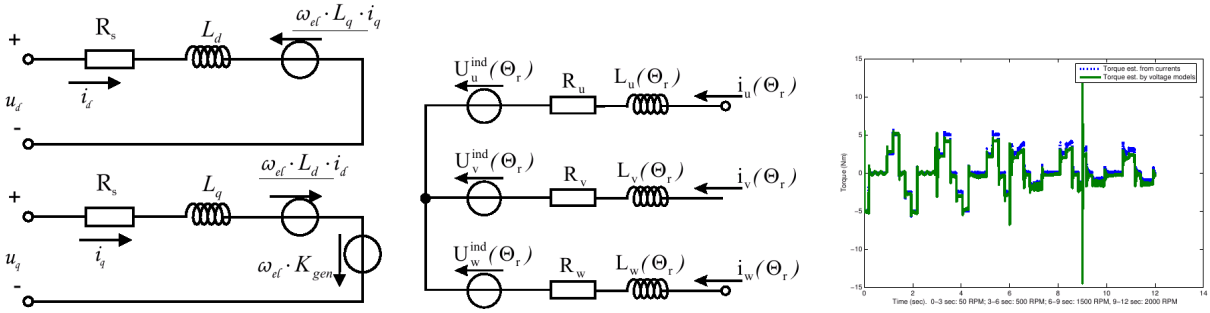
$$i_\beta = \frac{2}{3} \cdot \Im m \left\{ i_u + i_v \cdot e^{j \cdot \frac{2\pi}{3}} + i_w \cdot e^{j \cdot \frac{4\pi}{3}} \right\} = \frac{2 \cdot i_v + i_u}{\sqrt{3}}$$

$$i_d = \cos\theta_r \cdot i_\alpha + \sin\theta_r \cdot i_\beta \tag{3}$$

$$i_q = -\sin\theta_r \cdot i_\alpha + \cos\theta_r \cdot i_\beta$$

If the star point of the windings is not connected to ground, then currents in the u,v and w phase windings can be calculated using the following formula: $i_u + i_v + i_w = 0$, where i_u, i_v, i_w are the phase currents, as it can be seen on Fig. 1 b.

The (d,q) model can be transformed to the (u,v,w) model with the inverse of the previously mentioned Park and Clarke transformations.



a) Rotor oriented ref. frame b) Stator oriented ref. frame c) Torque estimation result

Figure 1: Machine models and torque estimation result

As it can be seen on Fig. 1 a and in (4), the motor model estimates the motor currents i_d, i_q by using the voltage inputs u_d, u_q in the rotor oriented coordinate system, where R_s is the phase resistance, L_d and L_q are the direct and quer inductance and K_{gen} is the generator constant of the machine. ω_{el} is the electrical angular speed of the machine. ω_{el} can be calculated with the following formula: $\omega_{el} = n_p \cdot \omega_{mech}$, where ω_{mech} is the rotor mechanical speed and n_p is the number of rotor magnetic pole pairs [1].

$$u_d = R_s \cdot i_d + L_d \cdot \frac{di_d}{dt} - L_q \cdot \omega_{el} \cdot i_q \tag{4}$$
$$u_q = R_s \cdot i_q + L_q \cdot \frac{di_q}{dt} + L_d \cdot \omega_{el} \cdot i_d + \omega_{el} \cdot K_{gen}$$

IV. Voltage measurement methods

It is clear from the previous sections that a precise voltage measurement is essential in the current based torque estimation. Hopefully, in those EPAS systems which are using PWM modulation to control the current of the electrical machine, phase voltages can be measured easily. There are two methods to determine the phase voltages:

- phase voltage measurement using additional hardware,
- calculating the phase voltages from the measured battery voltage and PWM duty cycles.

The first method is a bit more simple than the second, but it requires a dedicated measurement circuit for each phase. Furthermore, the precision of the measurement is questionable in a noisy automotive environment. The second method does not assume that an extra hardware – unless the battery voltage measurement hardware, which is necessary to other functions – is built into the system and also measurement noises are suppressed very well. The phase voltages are calculated from the phase duty cycles and the battery voltage. Then they are transformed to the rotor-oriented coordinate system.

V. Model parameter estimation

The precision of the current estimation highly depends on the quality of the identified model parameters. In the first step, with a quick method, the parameters can be estimated roughly. These can be used as initial parameters for the iterative, simulation based non-linear optimization algorithm.

If an inverter and a motor control software – including current regulators – is available, then the machine parameter could be determined easily. The quick method identification works in the following

way. The machine is set to different working points, where the parameters can be calculated from the machine equations:

- To estimate phase resistance, the machine angular speed is set to zero, and a relatively high current is forced into the machine windings. From (4) it is clear, that in the steady situation ($di/dt = 0$ and $w_{el} = 0$) only the resistance determines the connection between phase voltages and phase currents.
- In the next step, the generator constant of the machine can be determined if the reference signals of both current regulators are set to zero and if the machine is rotated at a known speed. Then in (4), the only non zero term is the voltage induced by the permanent magnets of the rotor.
- Setting the reference signal of the i_d (or i_q) controller to a non-zero value and knowing R_s and K_{gen} as a result of the previous measurements, L_d (or L_q) can be identified.

Using the previously determined machine parameters as initial values, they can be refined by an iterative minimization algorithm. The goal of the optimization is to minimize the difference between the simulated (i_d^*,i_q^*) and the measured (i_d,i_q) machine currents (5). A steepest descent minimization algorithm is used in the following way:

- simulate the machine with the current parameters,
- determine the negative gradient of (5)
- calculate the new machine parameters
- if necessary, repeat the above the steps.

$$min\left((i_d - i_d^*)^2 + (i_q - i_q^*)^2\right) \tag{5}$$

VI. Conclusion and further research possibilities

On Fig. 1 a it can be seen that the torque estimation works well in the whole operation range of the machine. Model based estimation does not cover only the static operation of the machine, but also the dynamical changes.

The main goal of the research was to find out how the torque of an electrical machine can be estimated if no torque measurement sensor and no current measurement sensor are used. A model has been presented which resulted a good torque estimate. In this paper a method has been shown, that is suitable for machine parameter estimation. Also some theoretical background has been presented to give a better understanding of voltage measurement and model transformation techniques.

It may be advantageous to improve this model based torque estimation method in the following directions:

- improve model complexity by introducing temperature-dependent parameter variations,
- improve model complexity by introducing non-linear saturation effects in the simulation model,
- implement on-line parameter estimation if there is more computational resource available.

Implementing model based torque estimation using the model and testing the method described in this paper within an industrial environment could help determining the further research directions.

Acknowledgment

Financial, technical, and theoretical support of ThyssenKrupp Research Institute Budapest is appreciated.

References

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

[2] The Matworks, Inc., *Clarke Transformation (Embedded Target for Texas Instruments C2000 DSPs)*, 2005.

[3] The Matworks, Inc., *Park Transformation (Embedded Target for Texas Instruments C2000 DSPs)*, 2005.

MODEL TRANSFORMATION PLUGINS ON EMF MODELS

Ákos HORVÁTH
Advisor: Dániel VARRÓ

I. Introduction

Nowadays, model driven software development (MDSD [1]) is an emerging paradigm in software development. Based on high-level modeling standards (e.g. UML), MDSD separates business and application logic from underlying platform technology by using platform independent models (PIM) to capture the core functionality of the target system, and platform specific models (PSM) to specify the target system on the implementation platforms (Java, C#, C++). PSMs and platform-specific source code are automatically generated from PIM and PSMs, respectively, by using model transformation (MT) techniques, thus, the role of MT is unquestionable for the overall success of MDSD.

As the complexity of model transformations is growing, a new demand has been arisen to separate the design from transformation execution by using high-level models at design time and by automatically deriving source code for the target platform of the platform execution from these high-level models.

II. Model Transformation by Graph Transformation

Informally, a graph transformation [2] (GT) rule performs local manipulation on graph models by finding a matching of the pattern prescribed by its left-hand side (LHS) graph in the model, and changing it according to the right-hand side (RHS) graph. GT rules are capable of manipulating the model locally, but every MT concept which is based on GT rules connects additional control structure to support complex MT constructs, so more and more complex MTs can be captured by a sequence of GT rules. But regardless of the approach of the model transformation, these concepts should enable a cost and time efficient design of (i) manipulations within a single modeling language (or domain) (ii) mappings and synchronization between different modeling languages and/or source code (iii) semantic translations into various mathematical domains to carry out formal model analysis.

III. Eclipse Modeling Framework

Eclipse Modeling Framework (EMF) [3] is currently one of the defacto standards for modeling used by leading software companies (e.g., IBM, SUN, etc.) and supported by the Eclipse foundation. EMF is a Java framework and code generation facility for building tools and other applications based on a structured model. EMF provides a metamodel for describing structured models. Using these structured models EMF provides tools and runtime support to produce a set of Java classes representing the model in Java, a set of adapter classes that enable viewing and a basic editor.

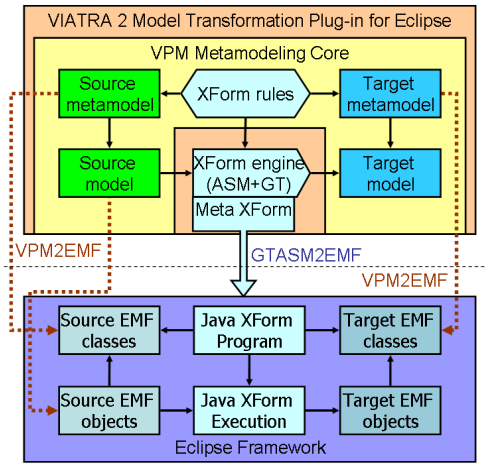
The ECore framework is used to create the metamodels. ECore is essentially the class diagram subset of UML. It is based on the Object Management Groups (OMG) Meta Object Facility (MOF) specification. From an ECore model, the generator of EMF can create a corresponding set of Java implementation classes. Every generated EMF class extends from the framework base class, EObject, which enables the objects to be integrated and appear in the EMF runtime environment.

IV. Overview of the Approach

The current paper describes a new approach using high order (meta [4]) transformation rules for the source code generation from high-level model transformation specifications defined by a combination

of graph transformations and abstract state machine (ASM) constructs (as used within the VIATRA2 [5] framework, a general Eclipse based modeling framework).

The essence of the approach is to store model transformation rules (`XForm` rules) as ordinary models in the model space. This way the source code generator (`Meta XForm`) of the transformations can be implemented within the modeling framework. As a result, the code generator can be reused by replacing only the output generation rules in order to port the transformations to new execution platforms.



The proposed solution consists of two main transformations. The *VPM2EMF* transformation is responsible for mapping the models and metamodels described in VIATRA2 into EMF models. The derivation contains two steps: (i) the `Source` and `Target metamodels` are mapped to `Source` and `Target` EMF classes, (ii) the generated Java classes representing the ECore model are instantiated based on the `source model` within the VIATRA2 framework.

Now that the `source` and `target EMF classe` are derived from the VIATRA2 framework, the *GTASM2EMF* transformation maps the high-level transformation specification of the framework into native Java code. Because of the joint representation of models and transformations this can also be done within VIATRA2 using meta-transformations (i.e., a transformation having transformations as its input and output). The model manipulation operations of the specification are mapped to reflective API calls provided by the EOBject.

This approach enables not only to run transformations as standalone application but also to integrate them with other java (Eclipse) based applications, which in case of specific tasks can significantly shorten development time (e.g., XML-to-XML mapping).

V. Conclusion

In the current paper, we proposed to meta-transformations for generating EMF-based transformer plugins from transformation specifications given by the combination of graph transformation rules and abstract state machines in the VIATRA2 framework. The completion of the implementation and the optimization of the model manipulation is an ongoing research.

In the future we would like to use this generator as a base platform for advanced transformation verification such as Proof Carrying Code [6] (PCC).

References

[1] J. Bettin, "Model-driven software development," URL: http://www.softmetaware.com/mdsd-and-isad.pdf.

[2] G. Rozenberg, Ed., *Handbook of Graph Grammars and Computing by Graph Transformation, volume 1: Foundations*, World Scientific, 1997.

[3] "Eclipse Modeling Framework," URL: http://www.eclipse.org/emf.

[4] D. Varró and A. Pataricza, "Generic and meta-transformations for model transformation engineering," in *Proc. UML 2004: 7th International Conference on the Unified Modeling Language*, T. Baar, A. Strohmeier, A. Moreira, and S. Mellor, Eds., vol. 3273 of *LNCS*, pp. 290–304, Lisbon, Portugal, October 10–15 2004. Springer.

[5] *VIATRA2 Framework*, An Eclipse GMT Subproject, URL: http://www.eclipse.org/gmt/.

[6] G. C. Necula, "Proof-carrying code," in *Proc. POPL '97: The 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 106–119, Paris, France, 15–17 1997.

TOWARDS THE MODEL BASED DESIGN OF SUPERVISORY SYSTEMS

Imre KOCSIS
Advisor: András PATARICZA

I. Introduction

Today, as the core business and business support of organizations relies more and more on their IT environment, the resilience of big, heterogeneous IT systems is of ever growing concern. One established way of increasing resilience is using supervisory systems: specialized system management software solutions supporting the detection and the alleviation of divergences from the expected nonfunctional characteristics – most notably, service level objectives and information security policies.

Currently there is no widespread design methodology for translating business needs imposed on the IT environment into a verified and validated supervisory system deployment and configuration. Our vision is to introduce the use of the Model Driven Architecture initiative [4] for this purpose. This paper outlines the research challenges and reports on the first steps taken.

II. Managing IT Infrastructure and Services

IT infrastructure and service management is generally addressed with standards based, enterprise grade Operational Management Product (OMP) families covering various so-called *management domains*. A management domain is a specific aspect of the systems and services to be managed; our research focuses on the availability and performance domain, as the most significant factor in resilience beside security management.

A. Business Driven IT management

Historically, IT management was IT resource centric; in the terms of availability and performance management this meant the supervision of the state and allocation of IT resources. Business driven *IT service management* recognizes the services provided by the IT infrastructure as first-class concepts and advocates management for so-called Service Level Objectives, agreed upon in Service Level Agreements. However, the mapping of service-level availability and performance goals into operational management configuration and deployment is a practically unsolved problem.

OMP-s can be categorized as tools for a) monitoring of hosts and network elements, b) distributed transaction monitoring, c) centralized, rule-based event management d) historical data collection and service level tracking and e) configuration management and automated infrastructure provisioning.
In the enterprise world, products developed by IBM, HP, CA and BMC are the most prevalent. In our research we employ IBM Tivoli products as standards-based management platform.

B. IBM Tivoli Change and Configuration Management Database

Of particular interest for us is the Tivoli CCMDB [2], a product maintaining an on-line infrastructure model through automated infrastructure discovery. Tivoli CCMDB is the strategic IBM middleware integrating the OMP level with IT service management. CCMDB represents the model of the infrastructure elements and their relationships in a standard-based language and publishes it via API-s; this way, it is capable to support on-line system level resilience analysis solutions.

III. Model Based Design of Supervision

The design task of supervision differs from established MDA approaches in certain key areas.

- Modeling covers particularly many aspects in the engineering domain; hardware, software, middleware, applications, and services with varying levels of abstraction needed. Applicable existing modeling languages provide only partial solutions at best.
- For the qualitative analysis of nonfunctional properties, not only novel algorithms but a sufficiently general and scalable error/incident model is needed.
- Novel system management approaches are to be supported – most notably, Autonomic Computing [5] and policy based management.

Some of these topics are being elaborated in the European FP6 IP project DESEREC [3], which aims to utilize the inherent structural reconfigurability of IT systems to devise supervisory mechanisms that guarantee IT service resilience against faults and security intrusions.

IV. Qualitative error modeling and analysis

Evolving model-based design for dependability approaches rely heavily on quantitative modeling, where complexity is handled by spatial and/or temporal compaction of system state (sub)vectors into partitions of a given fault/error/failure semantics. [1] introduces a suitably general, first order logic based error model and identifies possible approaches for their automated generation from system models accompanied with runtime platform description. [1] also gives constraint satisfaction and model checking based general approaches for the static and dynamic error propagation analysis of the models. A research goal is to improve the performance of the general algorithms, by finding heuristics and exploiting the facts that the number of qualitative error partitions is small for practical cases and that the events of interest for observation and possibly supervisory intervention are rare.

Additionally, incremental constraint solving on qualitative static error propagation models offers the possibility of incremental, on-line error propagation analysis; on-line constraint solving approaches are to be elaborated that scale well with model size and complexity.

V. Conclusion

In this paper, the first research steps and the initial background work has been reported on. Based on the evaluation of the available technologies a demonstrator is currently being developed for system-level qualitative error analysis to enable the evaluation of improved algorithmic approaches, with a current emphasis on on-line analysis. The demonstrator will be integrated with standards-based industrial products, and thus assessable for real life scenarios.

References

[1] András Pataricza, *Model Based Dependability Analysis*, thesis submitted to the Hungarian Academy of Sciences

[2] Tivoli Change and Configuration Management Database – Product Overview page, Dec. 2006, URL: http://www-306.ibm.com/software/tivoli/products/ccmdb/

[3] DESEREC FP6 IP EU project – public project page, Dec. 2006, URL: http://www.deserec.eu

[4] Joaquin Miller and Jishnu Mukerji. MDA guide version 1.0.1. Technical Report. omg/2003-06-01, OMG, 2003

[5] IBM Autonomic Computing Initiative. URL: http://www.research.ibm.com/autonomic/

MODELING BPEL WORKFLOWS WITH FAULT HANDLING

Máté KOVÁCS
Advisors: Dániel VARRÓ, László GÖNCZY

I. Introduction

The classical way of administering the activities of an organization involves data that are kept on paper and decisions that are made according to that data. There are also atomic activities that carry out the modifications on work items, thus modifying the information on the paper.

Today these concepts serve as a model of electronically executed business processes. The information on products is stored in databases, and the workflow is executed by a workflow engine. Decisions concerning the activities to be carried out on a work item are made autonomously in many cases in a programmatic way. Business processes are used to coordinate the activities of the organizations, and maintain information on the products.

The correct functionality is crucial to the successful operation of the company, thus business process designers and IT professionals need modeling and verification methodologies to reduce the risk of malfunctioning to a minimum. There are several initiatives that aim to provide a formal modeling technique to workflows, which could contribute to the quality of business processes being enacted. These techniques are usually based on already existing formalisms, such as finite state automata [1], Petri-nets [2] and dataflow networks [3]. However, the fault handling mechanisms are out of the scope of these existing methodologies.

In this paper a business process modeling technique is presented that also deals with the fault and compensation handling features of the Business Process Execution Language (BPEL) [4].

II. The Model of Business Processes

A. The BPEL Language

The Business Process Execution Language (BPEL) is designed to implement business processes. As such, the most important building block of a workflow is the *activity*. *Structured activities* are used to define the control flow of the *basic activities* that carry out modifications on data. Structured activities include *sequence*, *selection*, *iteration* and *parallel execution*.

A workflow implemented in BPEL may be organized into a hierarchy using the concept of *scopes*. Scopes provide a special sort of fault handling for the workflow fragment wrapped by it. According to the terminology used by the standard [4], basic activities may throw *faults* in case of malfunction that are caught by one of the *fault handlers* of the containing scope. The process fragment contained by the fault handler is dedicated to the reversal of the effect of the partially executed workflow fragment of the scope. The effect of a successfully terminated scope may be compensated by its *compensation handler*. The compensation is initialized from outside of the scope.

However, the precise modeling of the semantics defined by the standard is not trivial for several reasons. BPEL provides implicit functionality when either a fault handler or the compensation handler is missing: a snapshot of the state space of the workflow is taken when a scope successfully finishes, and later the compensation handler is run in the snapshot world, not affecting the state space.

B. The Formal Model of the BPEL Language

A BPEL process is mapped into a transition system to represent its states during execution. The initial state of the transition system corresponds to the initialization of the process. Each activity is

mapped into a small automaton depending on its kind. The state space of the workflow model is the composite of that of the constituent automata.

The state spaces of the automata modeling activities have three states in common: *not yet startable, activated*, and *terminated*. The states and state transitions are specific to the actual activity between the states activated and terminated.

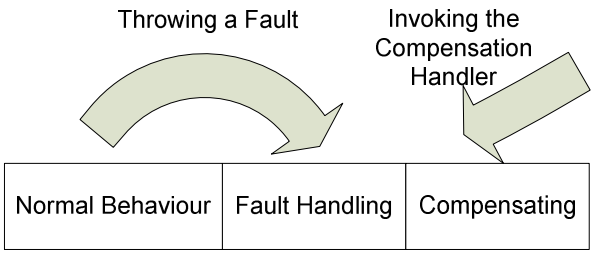


Figure 1: Structure of scopes

Fault handling techniques are closely related to the concept of scopes that have three different running modes with different purpose, namely *normal behavior*, *fault handling* and *compensating*. The finite state automaton modeling a scope maintains this information in its state space as it is shown in Figure 1. Each activity residing in any of the running modes of the scope may be of arbitrary complexity in hierarchy and functionality as well.

First, when the scope is triggered, the activity residing in the region of normal behavior is executed. If an activity in the mode of the normal behavior throws a fault, the running mode is switched into fault handling, and the activity inside the fault handler is triggered. The compensation of a scope may be triggered from the container scope.

III. Conclusion

I have implemented a model transformation to facilitate the practical usage of the business process modeling technique introduced above. The model transformation is implemented as a graph transformation running in the VIATRA2 framework. I have also created the importer of the BPEL language and the exporter of the SAL, which were possible due to the modular architecture of the framework. Furthermore, the metamodels of both of the languages were also created.

Model checking can be carried out on the business process model using the SAL. Requirements against the workflow have to be composed as temporal logical expressions that may also take into consideration the behavior of the process in case of an external fault. SAL is capable of the evaluation of those expressions based on the workflow model.

Our future plans include the extension of the methodology, so that it supports the automatic generation of the temporal expressions representing certain general requirements thus enhancing the usability of the method.

References

[1] J. Koehler, G. Tirenni, S. Kumaran, *From Business Process Model to Consistent Implementation: A Case for Formal Verification Methods* In Proc. *of the 6th International Enterprise Distributed Object Computing Conference*, pp. 96-106, 2002.

[2] Wil van der Aalst and Kees van Hee: *Workflow Management*, The MIT Press 2002.

[3] Máté Kovács and László Gönczy: "Simulation and formal analysis of workflow models", In *Proc. of the Fifth International Workshop on Graph Transformation and Visual Modeling Techniques,* pp 215-224, Vienna, AUSTRIA, Elsevier 2006.

[4] *T.* Andrews, F. Curbera et al., *Business Process Execution Language* Version 1.1. IBM, Microsoft, BEA et al. 2003. URL: ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf

COMPONENT BASED SOFTWARE DEVELOPMENT FOR EMBEDDED SYSTEMS

Bálint RÁKOSI
Advisor: András PATARICZA

I. Introduction

Recent embedded software development techniques tend to reach the limits of their scalability. For instance, the next generation of embedded automotive electronics will have several hundred of devices consisting total of 1 GB of code. Our environment will be dominated by ambient intelligent devices ranging from nanoscale devices up to the talking washing machine. Future embedded systems will coexist and cooperate in a complex and heterogenic system, so problems of software complexity will exceed the problems related to hardware design. Component-based development is a well proven method for hardware design, therefore its applicability for embedded software development should be examined.

II. Main Idea

Reuse of software is one of the most important aspects when developing new applications [1]. Component integration ensures a high level of IP reuse, therefore Component Based Software Development (CBSD) is the dominant design paradigm for desktop environments, enterprise systems, and web-based, distributed applications. However all basic component integration focuses only at the functional aspects of the system. The development of software for embedded systems necessitates the consideration of all the frequently complex non-functional and resource constrains in addition to the pure software quality aspects. A uniform specification and validation paradigm for software components covering non-functional constrains as well as functional aspects becomes to a vital factor from the point of view of the product quality.

III. Core Concepts

The first definition of a component [2] defines it as an executable entity, which can be composed and deployed at runtime. Accordingly, component-based software architecture defines a set of components, a set of component interfaces and their mutual interaction. A component could be seen as fragment of source code which can be derived from a high level abstract language and allows either compile or build time composition, or both. The component interface must include in case of embedded systems non-functional proprieties in addition to functional ones in order to support early, conceptual verification at the system design level.

The most pragmatic solution for small embedded systems suffering typical of a lack of resources is the exclusive use of design time composition and a complete avoidance of run time integration. This way interconnection of components can bee translated directly to function calls instead of using resource and power consuming dynamic event notifications.

Large-scale embedded systems with sufficient resources may rely on a reduced version of the component model on top of a real time operating system.

IV. Requirements

The most important requirement of embedded systems from the point of view of CBSD is the compliance to the required non-functional and extra-functional proprieties:

- Real Time Guarantees: violation of timing requirements even of in the case of a functionally correct reaction prohibits proper system functionality (latency, execution time, etc.)
- Dependability: represents the propriety of the system to function even in case of failures (avoidance of faults, fault tolerance) as characterized by other attributes like reliability, availability.
- Resource constraints: limitations in available resources like energy, computing power etc.
- Life-cycle properties: the system must be able to handle several generations of hardware and software technologies during its lifetime.

The development and maintenance of an embedded system requires a significant effort from the system designer as well as from the system programmer. The use of specification design languages at a high level of abstraction like UML complemented by techniques like Meta-Modeling, Model driven Architecture (MDA) [3], and Model Driven Software Development (MDSD) [4] lead to reduced costs both during system development and product life-cycle maintenance.

V. Component based modeling and design

An already proven method is presented in [5] to develop or design component-based software architectures. The system is partitioned to components and containers where the focus is on the containers which are responsible for aspects like: scheduling, interrupt handling, remote communication, generic driver interface, resource control, advanced error detection, meanwhile the components will contain only the functional aspects, like the application logic,. The source code of the components as well those of the containers can be generated from the abstract languages by the methods mentioned earlier.

VI. Challenges and future work

Currently we are developing a model-based approach for component-based software development, using tools openArchitectureWare [6] for code generation and Enterprise Architect [7] for modeling interfaces and complex data types.

OpenArchitectureWare uses an explicitly programmed metamodel (metaclasses describing the metamodel) implemented in Java. Additional templates (and extensions) define the mapping from the metamodel to the output source code. The Enterprise Architect is an arbitrary XML-based modeling tool. The output of the generator is the source code (skeletons) for the components, the complete container implementations, as well as a make file controlling the compilation and build process of the container and the components.

References

[1] Ivica Crnkovic *Component-based Software Engineering for Embedded Systems,* International Conference on software engineering, ICSE'05 ,ACM, St. Luis , USA, May, 2005

[2] C. Szyperski. Component Software: *Beyond Object-Oriented Programming. Second edition*, ACM, Press and Addison- Wesley, New York, N.Y., 2002.

[3] www.omg.org

[4] Tom Stahl, Markus Völter *Model-Driven Software Development Technology*, Engineering, Management 2006

[5] Markus Völter *Model-Driven Development of. Component Infrastructures. for Embedded Systems*. Version 2.0, Sept 29, 2004. Submission to MBEES Workshop 2005.

[6] http://www.openarchitectureware.org/

[7] http://sparxsystems.com.au/products/ea.html

CHALLENGES FOR ADVANCED DOMAIN-SPECIFIC MODELING FRAMEWORKS

István RÁTH
Advisor: Dániel VARRÓ

I. Introduction

In this paper, I identify main challenges for domain specific modeling frameworks. *Arbitrary abstract to concrete syntax mapping* enables the toolsmith to create visual representation to complex logical models with the possibility to determine what detail to show and what to hide in a flexible way. *Interactive model simulation* based on a model transformation architecture allows for the design time analysis and precise fine tuning of behavioral aspects.

The main goal of language engineering is to integrate the following aspects into a coherent framework: (i) *abstract syntax*, which specifies how logical model elements are stored and represented in the modeling environment; (ii) *concrete syntax*, which specifies the graphical representation of models; (iii) *well-formedness rules*, to describe static and language-specific constraints; (iv) *dynamic (operational) semantics* which describe the behavioral attributes of the system-under-design; and (v) *transformations* which provide formal and declarative support for creating generative bridges between various modeling domains.

Many state-of-the-art tools, such as MetaEdit [1], Microsoft's DSL Tools [2], openArchitectureWare [3] or the upcoming Graphical Modeling Framework [4] solve the traditional problems of custom modeling (the first three aspects) in an elegant fashion. However, there are multiple key areas where existing DSM frameworks typically fall short. The problem of *abstract to concrete syntax mapping* arises from the fact that abstract syntax models tend to be too complicated and difficult to handle for humans. *Interactive model simulation* is another important language engineering aspect which has been lacking support in traditional modeling tools.

With the *ViatraDSM Framework*, our main goal was to push the limits of tool integration further, in order to ease and speed up the construction of rich domain-specific visual languages. The ViatraDSM framework is tightly integrated with the VIATRA2 [5] engine, and relies on its modelspace containment and model transformation facilities to provide support for the evaluation of *well-formedness constraints*, *model simulation* and *interdomain transformations*.

II. Arbitrary abstract syntax to concrete syntax mapping

In constructing domain-specific visual languages, the clear separation of the logical modeling layer and the visualization layer is important. Abstract syntax models tend to be quite complicated, especially if they are tailored to the needs of the code generator, rather than the user. Therefore, the visualization layer needs to be independent, because that way the language engineer is free to construct diagrams that are easy to handle and do not confuse domain experts.

It is important to note the difference between the *model level* and the *metamodel level* separation of logical models and diagrams. On the *model level*, most tools separate the two layers by giving the user (almost) complete control over what parts of the logical model are visualized. The *metamodel level* separation, enables the language engineer to define visualization independently. Logical and diagram metamodels are constructed separately. This means that the GUI-driven editing only affects the diagram models directly, and the required changes to the logical model are generated by a mapping interface.

Our approach to map logical and diagram models, *bidirectional mapping*, makes use of the fact that the user can only apply a limited and well-defined set of modifications to the diagram through the graphical interface. Therefore, this approach directly maps the editing actions of the user to the logical model.

III. Interactive model simulation

Since the ViatraDSM framework is built on the VIATRA2 model transformation infrastructure, interactive model simulation can be implemented more easily. Ongoing research focuses on the following areas:

- Guided simulation using a *step-based execution*. This can be implemented with parametrized, single-step transformation rules. In that case, the toolsmith constructs a model transformation which executes a single transformation step. The framework runs these transformation steps, with support for user interaction at non-deterministic choice points.
- *Debug-style simulation* with support for more precise control over the transformation process. This can be regarded as an evolutionary continuation of the step-based execution approach. From the point of view of the ViatraDSM framework, the most challenging aspect of this approach is to provide a domain-specific interface to the debugger.
- General *trace representation* to preserve the different states of the model during simulation. Tracing is an important aspect of model simulation, because it provides useful feedback on simulation runs. With a trace, the various states of the system and the non-deterministic choices can be regenerated at a later time.

  This research topic is strongly connected to those of debug-style simulation, because traces should ideally have a domain-specific interface as well.

IV. Conclusion

To sum up, current research is active in the following areas:

1. Arbitrary model-to-diagram and diagram-to-model mapping based on a declarative specification, to facilitate the metamodel level separation of diagrams and logical models, and provide a flexible and transparent bidirectional mapping between them.

2. Design time interactive model simulation and constraint evaluation using transparent model transformation techniques, with support for the generation of traces. A domain-specific interface to the generic transformation engine is in the planning stage.

References

[1] Metacase, "MetaEdit+," URL: http://www.metacase.com/mep/.

[2] Microsoft, "DSL Tools," URL: http://lab.msdn.microsoft.com/teamsystem/workshop/dsltools/default.aspx.

[3] M. V. Bernd Kolb, "openarchitectureware and eclipse," URL: http://architecturware.sourceforge.net/data/oawEclipse.pdf.

[4] The Eclipse Project, "Graphical Modeling Framework," URL: http://www.eclipse.org/gmf.

[5] *VIATRA2 Framework*, An Eclipse GMT Subproject, URL: http://www.eclipse.org/gmt/.

AADL, A LANGUAGE FOR MODELING REALTIME EMBEDDED SYSTEMS

Dániel TÓTH
Advisor: Dániel VARRÓ

I. Introduction

The SAE Architecture Analysis & Design Language (AADL)[1] is a modeling language for formally correct description and analysis.[2] Its intended application area is software and hardware architecture modeling of performance-critical real-time embedded systems. Performance-critical systems are systems whose operation strongly depends on meeting non-functional system requirements such as reliability, availability, timing, responsiveness, throughput, safety, and security. Following the trend of modern modeling languages, AADL provides both a textual and a visual syntax. The language is also designed to be extendable to accommodate aspects that the core language does not completely cover.

II. Overview

The AADL standard defines the abstract syntax of the language as a Meta Object Facility (MOF) based metamodel. A trivial mapping exists between the metamodel and both the textual and graphical notations, so the two notations are completely equivalent. The metamodel's entities have been divided into three main categories: *software*, *platform* and *system*. The software category defines *process*, *subprocess*, *thread* concepts and commonly used data sets along with timing requirements and periodicity of each processing task. The platform category defines hardware resources such as *processor*, *memory*, *bus* and *auxiliary device*, each having a set of possible performance properties. The *system* is the only composite category, representing the assembly of software and execution platform components.

A. Levels of description

The standard defines four levels of description. These can be thought of as four metalevels. The first is the predefined metamodel of *category* concepts. The next level is the *type* for external interface specification of the functional elements of the software. The *implementation* level details the content of the implementation of interfaces defined in *type*s. Each type definition can have multiple implementations, supporting fault tolerant designs such as n-version programming. The implementations can have multiple run-time *instances*. *Inheritance* can be defined between type and implementation concepts.

B. Features of components

A *feature* is a part of a component type definition that defines how a component can interface with other components. The main feature components are: *port*, *subprogram*, *parameter* and *subcomponent access interface*. Ports are logical connection points for transfer of control and data. A rich set of *connection* types is available to define the precise semantics and temporal characteristics of transfer between ports. A similar connection concept exists for parameter and subprogram access. Flow graphs can be defined within components. This detailed specification facility enables thorough analysis of timing, delay and other quality of service parameters as well as error propagation paths and reliability.

C. Bindings and modes

The software component associations to pieces of execution hardware are called *bindings*. They can be organized into operation modes, that represent the operating configuration of the system at a given time. Mode transitions are useful for modeling degraded system health caused by component failure

or autonomous dynamic adaptation to load levels.

III. State of the Art

The AADL standard was published in 2004 by the Society of Automotive Engineers (SAE)[1]. There is a two-tiered tool support strategy. Commercial tool providers include the standard developers and key industry users. To facilitate the industry acceptance open source tool sets called OSATE and TOPCASED are also being developed.

Interoperability between different tools and other related standards is a key issue. To avoid the problem of various slightly different vendor specific model formats that was a serious problem of early UML tools the AADL Standards Committee has released a reference XML and XMI schema for AADL model interchange. An ECore metamodel is also a provided for Eclipse based tool development. UML integration will be realized by an AADL UML profile that is being standardized.

OSATE is an Eclipse plugin set based on the reference AADL model[3]. Its main features include a syntax guided textual and visual editor along with a set of plugin extension interfaces. The extensions can take the form of model analysis, generation, transformation and extraction features. Commercial code generation tools are being developed for automated code stub generation in C and ADA languages[3]. There are ongoing research projects of transforming subsets of AADL models into state automata and petri nets for dependability analysis[3]. Several language extensions, *annexes* are also being developed to cover specific problem fields.

IV. Further research issues

The detailed description facility and precisely defined execution semantics of AADL enable very thorough model checking and analysis. At the moment only a small subset of the theoretically proovable properties of AADL models can actually be verified due to the lack of capable verification tools. By translating AADL models into SAL[4] timed automata models the existing model checking tools can be used to verify formalized properties[5]. A focused part of my research is the question of formally correct back annotation. Without it the verification results of a translated model are exceptionally hard to interpret on the original model, defeating the original purpose of the verification.

The second important field of research is the automated implementation code generation and deployment. Up to now, code generators were either incomplete, providing only stubs or ”frames” that a programmer had to fill in by hand, or complete but restricted to a very narrow problem domain. The lack of a universal and at the same time semantically well defined modeling language was a limiting factor. AADL has potential to solve some of these issues as it covers several aspects of integrated system development. Its scope is still limited, but extendable by utilizing annexes. Integrating the formal verification results into proof carrying code and improving the completeness of the generated code by utilizing the annexes is the main goal of my research.

References

[1] As-2 Embedded Computing Systems Committee, *Architecture Analysis & Design Language (Aadl) Standard*, Nov. 2004, AS5506.

[2] J. J. H. Peter H. Feiler, David P. Gluch, *The Architecture Analysis & Design Language (AADL): An Introduction*, Carnegie Mellon University, Feb. 2006.

[3] “The aadl user workshop,” Oct. 2005, the AS2C subcommittee of SAE and Axlog Ingénierie.

[4] L. de Moura, S. Owre, H. Rueß, J. Rushby, N. Shankar, M. Sorea, and A. Tiwari, “SAL 2,” in *Computer-Aided Verification, CAV 2004*, R. Alur and D. Peled, Eds., vol. 3114 of *Lecture Notes in Computer Science*, pp. 496–500, Boston, MA, July 2004. Springer-Verlag.

[5] M. Sorea, “Bounded model checking for timed automata,” (SRI-CSL-02-03).