

120-th European Study Group with Industry (ESGI'120)

*July 25-29, 2016
Sofia, Bulgaria*

PROBLEMS & FINAL REPORTS



ECMI EUROPEAN CONSORTIUM FOR
MATHEMATICS IN INDUSTRY

**FASTUMPRINT
2016**

Organizers:

**Faculty of Mathematics and Informatics
Sofia University “St. Kl. Ohridski”**

Institute of Information and Communication Technologies, BAS

Institute of Mathematics and Informatics, BAS

in cooperation with

European Consortium for Mathematics in Industry

Scientific Advisory Committee:

Prof. Dr.Sc. Svetozar Margenov, Director of IICT, BAS

Assoc. Prof. Dr. Nadya Zlateva, Vice-Dean of FMI, SU

Acad. Julian Revalski, Director of IMI, BAS

Local Organizing Committee

Assist. Prof. Tihomir Ivanov, FMI, SU and IMI, BAS – chairperson

Prof. Dr.Sc. Stefka Dimova, FMI, SU

Assoc. Prof. Dr. Ivan Georgiev, IICT and IMI, BAS

Assoc. Prof. Dr. Vesselin Gushev, FMI, SU

Galina Lyutskanova, FMI, SU

Dragomir Aleksov, FMI, SU

© Sofia University “St. Kl. Ohridski”, 2016

© Institute of Information and Communication Technologies, BAS, 2016

© Institute of Mathematics and Informatics, BAS, 2016

Published by FASTUMPRINT, Sofia, Bulgaria

ISBN 978-619-7223-31-6

Contents

Preface	5
List of participants	7

Problems

Problem 1. <i>MM Solutions AD</i> – Problem 1. Multi-frame Denoising of Still Images	11
Problem 2. <i>STOBET Ltd</i> – Optimal Cutting Problem	15
Problem 3. <i>Adastra Bulgaria</i> – Authenticity Management Algorithm for Digital Images	17
Problem 4. <i>BG Drilling Solutions Ltd</i> – Laboratory Calibration of a MEMS Rate Gyro Sensor	19
Problem 5. <i>TechnoLogica Ltd</i> – Future Cyber Attacks Modelling & Forecasting	22
Problem 6. <i>Gruner AD</i> – Post-Processing for Beam Elements: Calculating the Second Order Work and Strain Energy	23
Problem 7. <i>MelissaClimate</i> – Mathematical Model of Residential Storage Water-heating System	26

Final reports

<i>S. Harizanov, D. Dimov, N. Naidenov, S. Kostadinova.</i> Multi-frame Denoising of Still Images	31
<i>A. Avdzhieva, T. Balabanov, G. Evtimov, I. Jordanov, N. Kitanov, N. Zlateva.</i> Two Dimensional Optimal Cutting Problem	54

<i>T. Balabanov, P. Dojnow, V. Kolev, N. Manev, W. Mudzimbabwe, P. Tomov, I. Zankinski, S. Zhelezova.</i> Authenticity Management Algorithm for Digital Images	59
<i>T. Ivanov, G. Lyutskanova, D. Aleksov, O. Kounchev.</i> Laboratory Calibration of MEMS Rate Sensors	68
<i>Z. Minchev, G. Dukov, D. Boyadzhiev, P. Mateev.</i> Future Cyber Attacks Modelling & Forecasting	77
<i>S. Stoykov, D. Fidanov, M. Datcheva, S. Dimova.</i> Post-Processing for Beam Elements: Calculating the Second Order Work and Strain Energy	87
<i>M. G. Zefreh, M. Dvoriasyna, M. Galabov, L. Yovkov, K. Liolios, S. Darachev.</i> Mathematical Model of Residential Storage Water-heating System	94

Preface

The 120th European Study Group with Industry (ESGI'120) was held in Sofia, Bulgaria, July 25–29, 2016. It was organized by the Faculty of Mathematics and Informatics, Sofia University “St. Kliment Ohridski” (FMI–SU), the Institute of Information and Communication Technologies, Bulgarian Academy of Sciences (IICT–BAS) and the Institute of Mathematics and Informatics, BAS (IMI–BAS) in cooperation with the European Consortium for Mathematics in Industry (ECMI). ESGI'120 was the fourth Study Group in Bulgaria, after the very successful ESGI'113, September 14–18, 2015; ESGI'104, September 23–27, 2014; ESGI'95, September 23–27, 2013.

ESGI'120 was financially supported by the Mathematics for Industry Network (MI-NET), COST Action funded project (TD 1409), which aims to facilitate more effective widespread application of mathematics to all industrial sectors, by encouraging greater interaction between mathematicians and industrialists. We are grateful to the Core Group of MI-NET, and especially to the Chair Dr. Joanna Jordan and the Project Manager Dr. Filipa Vance, for their assistance and support. The event was also sponsored by the companies that posed problems for solving.

ESGI'120 was hosted by the Faculty of Mathematics and Informatics, SU. FMI provided excellent conditions for work and communication. Well-equipped computer laboratories as well as the rich library of FMI were available for the work of the participants. We thank Assoc. Prof. Dr. Nadya Zlateva, Vice-Dean of FMI and member of the Scientific Advisory Committee, for her constant help and support.

Study Groups with Industry are an internationally recognized method of technology transfer between academia and industry. These one-week long workshops provide an opportunity for engineers and industrial developers to work together with academic mathematicians, students and young professional mathematicians on problems of direct practical interest.

The Organizing Committee selected seven problems to work on:

1. *Multi-frame denoising of still images*, MM Solutions AD;
2. *Optimal cutting problem*, STOBET Ltd.;
3. *Authenticity Management Algorithm for Digital Images*, Adastra Bulgaria;

4. *Laboratory calibration of a MEMS rate gyro sensor*, BG Drilling Solutions Ltd.;
5. *Future Cyber Attacks Modelling & Forecasting*, TechnoLogica Ltd.;
6. *Post-Processing for Beam Elements: Calculating the Second Order Work and Strain Energy*, Gruner AG;
7. *Mathematical Model of Residential Storage Water-heating System*, Melissa Climate.

Six of the companies are Bulgarian. The company Gruner AG has had a collaboration with the Institute of Mechanics, BAS.

The participants from Bulgaria (35) and from abroad (2), among which 1 BSc, 1 MSc, and 10 PhD students, were divided into seven groups, each group working as a team on one of the problems. The Bulgarian participants were from various Academic institutions: FMI-SU; IMI-BAS; IICT-BAS; Plovdiv University; Veliko Tarnovo University; University of Blagoevgrad; University of Rousse. The foreign participants were from the University of Genoa, Italy and from the Herriot-Watt University, UK. Prof. Wil Schilders, a former President of ECMI, attended the event for three days. We thank him for the permanent interest and attention to the Bulgarian Study Groups, as well as for the valuable discussions with the participants on most of the problems.

The traditional ECMI Modelling Week, organized this year by FMI as ECMI Educational Center, was designed to run back-to-back (17–24.07) with ESGI'120.

On the last day of the workshop, the working groups presented the progress in solving the problems and the recommended approaches for their further treatment (including generalization, improvements and implementations). The presentations have been a basis of the final reports, which are assembled in this booklet. The Study Group Final Report provides a formal record of the work for both the industrial and the academic participants.

The description of the problems, the last day presentations and the final reports of each working group, as well as a color copy of this booklet, are posted on the website of ESGI'120:

<http://esgi120.fmi.uni-sofia.bg>

As usual the participants were awarded certificates for their valuable contribution to the event.

List of participants

Ana Avdzhieva (Sofia University, FMI)
Dimitar Fidanov (Plovdiv University, FMI)
Dimo Dimov (IICT, BAS)
Doychin Boyadzhiev (Plovdiv University, FMI)
Dragomir Aleksov (Sofia University, FMI)
Galina Lyutshkanova (Sofia University, FMI)
Georgi Evtimov (IICT, BAS)
Georgi Dukov (TechnoLogika)
Ilian Zankinski (IICT, BAS)
Ivan Jordanov (University of National and World Economy)
Konstantinos Liolios (IICT, BAS)
Lyudmil Yovkov (Sofia University, FMI)
Maria Datcheva (IMeh, BAS)
Mariia Dvoriashyna (University of Genoa, Italy)
Masoud Ghaderi Zefreh (Heriot-Watt University, UK)
Mihail Galabov (IICT, BAS)
Nadia Zlateva (Sofia University, FMI)
Nikola Naidenov (Sofia University, FMI)
Nikolai Manev (IMI, BAS)
Nikolay Kitanov (University of Blagoevgrad)
Ognyan Kounchev (IMI, BAS)
Petar Tomov (IICT, BAS)
Peter Dojnow (IGIC, BAS)
Plamen Mateev (Sofia University, FMI)
Savka Kostadinova (Sofia University, FMI)
Stanislav Darachev (Sofia University, FMI)
Stanislav Harizanov (IICT, BAS)

Stanislav Stoykov (IICT, BAS)
Stefka Dimova (Sofia University, FMI)
Stela Zhelezova (IMI, BAS)
Stoyan Poryazov (IMI, BAS)
Tihomir Ivanov (Sofia University, FMI)
Todor Balabanov (IICT, BAS)
Vasil Kolev (IICT, BAS)
Veneta Yosifova (Sofia University, FMI)
Walter Mudzimbabwe (University of Ruse)
Wil Schilders (TU Eindhoven)
Zlatogor Minchev (IICT, BAS/IMI, BAS)

PROBLEMS

Problem 1. Multi-frame Denoising of Still Images

MM Solutions AD

Angel Ivanov, aivanov@mm-sol.com

A sequence of 8 images are captured quickly one-by-one while camera is held in hand and unintentional hand shake is present. When capturing in low light conditions (< 20 Lux), the images appear blurred and noisy because the camera uses long exposure (1/15..1/5 sec) and high gain (ISO). The target is to combine multiple frames to produce an image with less noise and less (ideally no) blur. The processing algorithm should not be too computationally expensive. As usual in the image processing, the target is to get a good looking image, rather than removing the noise and the blur to achieve the ground-truth latent image. So any brain-hacking tricks are acceptable.

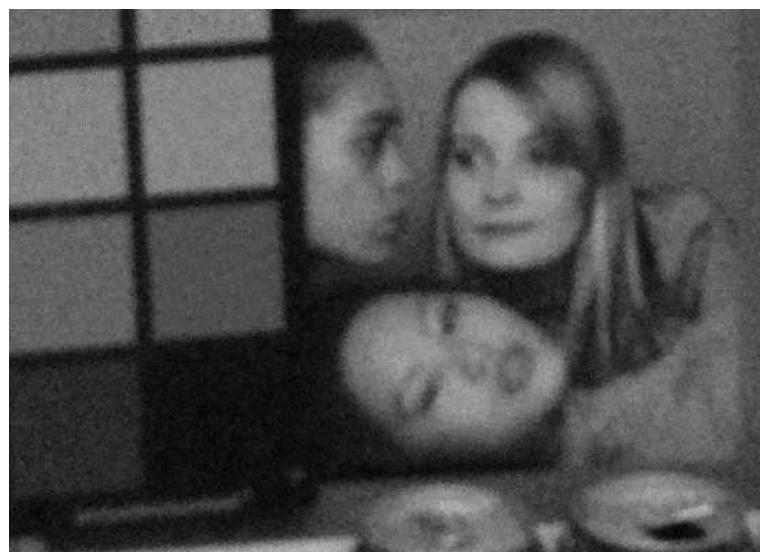
Good-looking image means:

- no unnatural artifacts. Better little less sharpness than having artifacts (e.g. ringing).
- little high-frequency noise (up to STD about 1–2 for 8-bit image) creates an impression of sharper image
- the brain is extremely sensitive to the quality of straight edges and less sensitive to noise, present in image areas with fine details (grass, leaves).

A cropped part of an image captured at 10 Lux with the strongest blur within the series is presented below.



Next image presents the image with least blur in the same sequence.



Noise characteristics

STD = 6.5 (from 8bit pixel values)

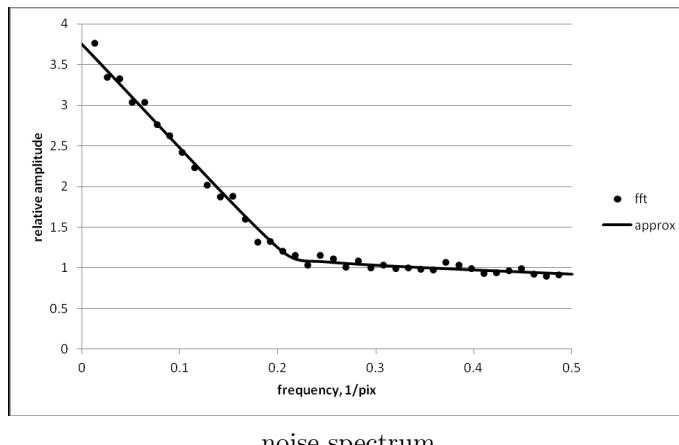
STD changes at different brightness about 2 times. The presented value is the maximum STD, observed at pixel values about 64..128

The noise is Gaussian, but not white – Low frequencies amplitudes are higher.

The next figure show a noise spectrum, measured on a flat (without details) noisy part of one of the images in the same sequence. This is a “pink noise”, typical for electronic circuits like the analog amplifiers embedded in the image sensor.

More noise or less brightness can be expected at lower light, so there is no upper bound of the noise STD – the more noise the algorithm can handle, the better.

For sure, the noise in each individual image is more than the edges we would like to enhance and see in the output image.



noise spectrum

Blur characteristics

Blur kernel changes smoothly within the image – it is not same for the whole image, as opposed to the common assumption in most de-blur articles. This is caused by 2 facts:

- camera shake is mainly rotational;
- contemporary image sensors use rolling shutter, i.e. different rows of the image are exposed in different moments, for same amount of time. Hence, the camera motion during exposures of the rows varies.

In typical hand-held capture case, the blur is up to about 20pix at 4000×3000 image resolution, at 1/15 sec exposure time, close to line.

Alignment

Apart from blur, hand shake causes the images not to be aligned to each other. Alignment is NOT part of that project. We have an algorithm, which aligns the images with up to 2pix error at 4000×3000 image resolution.

Preprocessing

The images are already pre-processed – color conversion and gamma correction is applied, so they are not in linear space (lightness-vs-pixel value is not linear). Finding the gamma applied is possible, but hard, so better to avoid it. The processing parameters are same for all the images in the sequence, including exposure time and gain.

Applying different exposure time and gain to the images is possible, but it would lead to different lightness-vs-pixel response of each image and differences in color. Thus, worse alignment and even worse moving-objects detection.

Moving-objects detection and handling is not part of this project, but yet, it happens somewhere, so we have to keep in mind its presence. Basically, it leaves the moving objects only from one of the images. Another approach is to leave the moving objects to create ghosts – if the exposure periods of the subsequent images are contiguous, this would create a moving object motion blur, as if captured with a very long exposure period, i.e. the image will look consistent.

Approaches

To decrease the blur, exposure time can be reduced, but this leads to darker image, i.e. less SNR.

To decrease the noise, either the gain could be decreased (thus less brightness, less overall SNR) or exposure could be increased while decreasing the gain (keep brightness, but increase blur).

Average all images – decreases the noise up to $\sqrt{\text{number-of-images}}$. Decreases the blur also since different images are blurred in different directions. Yet, the residual blur may be still be more than the one of the least blurred image.

Deblurring – there are many algos, simultaneous deblurring of multiple frames shows good results (in the articles), but is very computationally expensive. Yet, some simple approach is computationally acceptable. Note, deblurring increases the noise, especially the high frequencies. Also, deblurring is very prone to creating “ringing” artifacts near strong edges and overexposed areas.

Problem 2. Optimal Cutting Problem

STOBET Ltd, www.stobet.com

Eng. Georgi Evtimov, gevtimov@abv.bg

Engineering company “STOBET” Ltd makes projects for buildings and structures. Having many such projects, we want to improve the quality and the price of the process of their realization. In our work we use software for design of steel constructions. The software gives the drawings of the different needed profiles. After making all the drawings, the task is to deploy optimally all profiles on a steel sheet.

Task for optimization

We have a large piece of steel sheet with dimensions $X = 1500$ mm, $Y = 12000$ mm. In this big sheet we have to locate maximum number of small plates of given profiles.

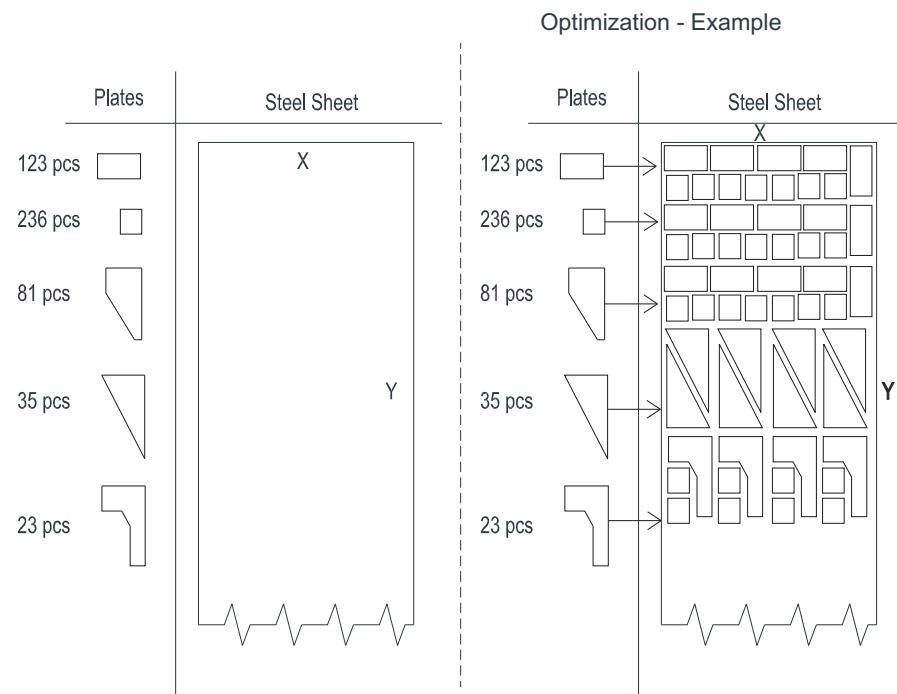
The contour of the plates can be square, rectangle, trapeze, any non crossing closed polyline, consisting of line segments only. The dimensions of the plates are given by the coordinates of the vertices of the polylines as follows:

Plate 1: $(X_1, Y_1; X_2, Y_2; X_3, Y_3)$
Plate 2: $(X_1, Y_1; X_2, Y_2; X_3, Y_3; X_4, Y_4; X_5, Y_5)$
.....
Plate N : $(X_1, Y_1; X_2, Y_2; X_3, Y_3; X_4, Y_4)$

The numbers K_1, K_2, \dots, K_N of the pieces of plates with given profile, which must be situated on the steel sheet, are given.

In the process of optimization some plates can be rotated for best fit.

The aim is to locate maximum small plates in the given big steel sheet. A distance for cutting must be left between the plates. This distance is about 5 mm.



Problem 3. Authenticity Management Algorithm for Digital Images

Adastra Bulgaria, www.adastragrp.bg

Ivan Pavlov, ivan.pavlov@adastragrp.com

Company's overview and our business context

Adastra Group (www.adastragrp.com) is a leading provider of Information Management solutions to global Fortune 1000 companies. One of Canada's 50 Best Managed companies, Leveraging over a decade of experience across industries and technologies, Adastra bridges the gap between pressing business needs and technology consulting to protect existing investments and extend and unlock their business value. Adastra has over 800 employees worldwide with headquarters in Canada and the Czech Republic and offices in Russia, Germany, Slovakia and Bulgaria.

Adastra Bulgaria (www.adastragrp.bg) was founded in the year 2007 and currently has two offices in the cities of Sofia and Varna. With over 140 employees, our portfolio includes various projects in the areas of Data Warehousing, Business Intelligence, Data Integration, Data Quality, Quality Assurance, Master Data Management, Big Data and Data Science for leading companies around the globe.

Problem – Authenticity Management Algorithm for Digital Images

An Android application is used for taking pictures. Those pictures need to be digitally signed in order to verify their authenticity later. The task is to create an algorithm which adds a digital watermark to an image and a corresponding algorithm which verifies if a given image is watermarked by the original algorithm.

The following conditions must be met:

- The algorithm must be able to detect if the image has been tampered with
 - e.g. re-saving, cropping, resizing, editing small parts of the image in a photo-editing software, copying and moving regions, etc.
- There should be no visibly discernible difference between the original image and the watermarked image.
- Watermark verification should not require the original image.

- The watermark embedded in an image generated by using a particular marking key must be detected only by providing the corresponding information to the verification algorithm. All other side information provided to the verification algorithm should fail to detect the mark.
- The insertion of a mark by unauthorised parties should be difficult.
- The watermark should be capable of being embedded in the compressed domain.

An optional requirement is that the verification algorithm should be able to locate and characterize the changes made to a watermarked image.

The winning solution will consist of:

- A comprehensive description of a fast watermarking algorithm.
- A comprehensive description of a verification algorithm.
- Theoretical justification of the algorithm.
- Estimation of RAM usage.
- Android implementation of the watermarking algorithm and benchmarking using different image sizes.

Problem 4. Laboratory Calibration of a MEMS Rate Gyro Sensor

BG Drilling Solutions Ltd

Shteryo Lyomov

Introduction

In oil, gas or geological exploration borehole drilling and logging work, acquiring the direction, inclination and orientation of the drilling tool as well as the borehole is provided by surveying tools. Data from surveying tools is gathered in real-time and transmitted to surface or stored into internal tool's memory.

This information is used to the decision-making control side to adjust the action of drill bit accurately, which will efficiently improve the quality of drilling engineering or other application.

For those applications, where high accuracy or stability is required, MEMS sensors – accelerometer, magnetometer and gyroscope are used. To maximize performance, the sensors used in these applications are generally calibrated and externally temperature compensated.

Simple borehole surveying technology is shown in Figure 1. In the example accelerometers and gyro sensors are employed for precise direction (azimuth) and inclination (dip) measurement in order to determine borehole 3D coordinates.

Background

Calibration is essential for providing the maximum of sensor specification. It includes calculation of couple parameters (constants) used to normalize measured data and most often include bias (offset), sensitivity (scale factor) and misalignment.

Here we will focus on gyroscopes calibration. 3D gyroscopes measure the angular velocity in inertial space around three perpendicular sensitivity axes. Thus it allows for the orientation of an object to be determined. In general, the orientation of a rigid body is the position of its coordinate system observed relative to a reference coordinate system with the same origin. The orientation can be described by a rotation that would move the rigid body's coordinate system, which is initially aligned with the reference coordinate system, to its new position. When working with gyroscope measurements, we consider the gyroscope as the rigid body and the inertial space coordinate system as the reference system.

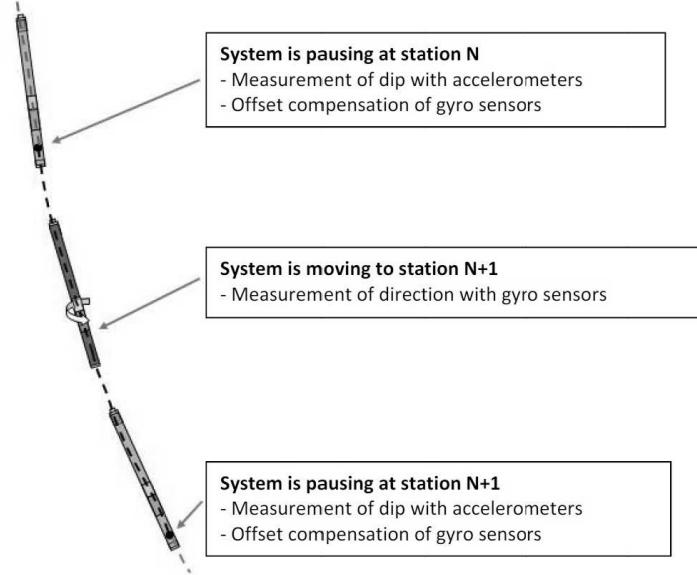


Figure 1. Borehole surveying

The measured angular velocity determines the rotation required to move the sensor to its new position.

In the real world a physical body (surveying tool with sensors) has its position and orientation expressed by the column vector Q of the real physical quantity values in the basic coordinate system of the tool expressed by:

$$Q = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix}. \quad (1)$$

Then the values detected (measured) by a 3D sensor along its three sensitivity axes in a column vector can be expressed by:

$$Q_s = \begin{bmatrix} q_{s,x} \\ q_{s,y} \\ q_{s,z} \end{bmatrix}. \quad (2)$$

In the noiseless model, the accuracy of the measured values Eq. (2) with MEMS gyroscope depends on 1) the zero level offset; 2) the sensitivity, and 3) the alignment of the three sensitivity axes of the sensor. In order to calibrate

a 3D sensor we must determine the values of 12 parameters defined by the 3D sensor model, the nine elements of the calibration matrix include the sensitivities of the 3 axis and misalignment and the three elements of the zero level offset vector.

The zero level offset is the value detected by the sensor when the measured physical quantity is equal to zero. To do this, it is needed to bring the gyro sensor up to the intended operating temperature (allowing sufficient soak time) and measure the null output voltage for all axes.

The sensitivity of the sensor is equal to the ratio between the change in the detected value and the change in the real value, assuming that the sensor characteristic is full-scale linear.

Due to imprecise manufacturing, the sensor sensitivity axes can be misaligned and thus deviate from the sensor coordinate axes. We consider the orientation of each sensor sensitivity axis to be independent of the remaining two axes.

Problem formulation

There are several methodologies that provide solution for the problem of mutual calibration of MEMS sensors – gyroscopes. Those methods, however are quite complex and computationally expensive, require very expensive test equipment, long time for development and assume some initial calibration of the sensors.

A simple and quick solution for laboratory-based calibration is needed, that to be used as a quick calibration, followed by a high-precision calibration, which should cover the temperature and aging deviations of the calibration parameters.

Problem 5. Future Cyber Attacks Modelling & Forecasting

TechnoLogica Ltd, www.technologica.com

Eng. Georgi Dukov, gdukov@technologica.com

Company Profile

TechnoLogica Ltd. is a leading Bulgarian software company, which since 1990 has been developing a comprehensive range of information technology services including implementation of information systems, software development, consulting and specialized training. The company has offices in the Bulgarian cities of Sofia, Plovdiv, Varna and Burgas, in Serbia and in the Republic of Macedonia. Its team numbers more than 200 experts in various fields, including experts with the highest levels of certification: PMI Project Management Professional, Oracle Master, etc. More information about us is available at: <http://www.technologica.com/>

Problem Description

Modern digital world is constantly presenting new challenges, resulting from cyber-physical clashes. One of the greatest problems in the present context is adequately to react towards new threats and expected attacks in the evolving modern world. Both social and technological system components have to be studied in this sense.

A useful approach in support of the problem general coping is fusing expert, observation and reference data in weighted graph-based analytical models. Further, these however have to be adequately assessed and studied from multiple cyberattack dynamic perspective.

Five key steps towards solving the problems could be defined:

1. Selection of typical cyberattacks for further exploration in accordance with future digital space threats evolution prognosis;
2. Models definition for the selected cyberattacks in a suitable future exploration context;
3. Multicriteria selection for models overall holistic assessment;
4. Software working environments selection and models machine prototyping;
5. Numerical simulations of the developed models and results discussion.

Problem 6. Post-Processing for Beam Elements: Calculating the Second Order Work and Strain Energy

Gruner AD, <http://gruner.ch/en>

1. Company's Overview

The Basel-based Gruner Group is an independent market leader in engineering and planning consultancy services. Founded over 150 years ago, we now have more than 20 companies at over 30 sites in Switzerland and all over the world.

Gruner is a byword for construction services that set benchmarks in terms of quality. A broad set of competencies, a committed workforce, in-depth expertise and many years of project experience enable us to provide support for complex building projects throughout their entire life cycle. The interdisciplinary teamwork adopted by Group companies guarantees that customers receive individual solutions for challenging construction projects.

Through our cooperation with both higher education institutions and public and private research teams, our knowledge is in a constant state of development.

2. Definition of the problem

Modern design of excavation pits includes often the simulation of the problem using Finite Element (FE) models. Primary objective for this kind of models is to predict deformations of the excavation pit support and the surroundings. As a secondary objective, it would be very useful to know, if the modelled structure is near to a failure state or not.

For this kind of modelling task, usually a commercial software FE package is used as a black-box model with no access to the source code and no option for altering the algorithms included in the code. This is the case because this kind of software packages are widely known, well tested and accepted by the professional engineer community.

To model excavation pits and especially their support usually a plain-strain analysis is performed where beam elements are employed. The software package used by the company (Plaxis [2]), and this FE code is widely used in geotechnical engineering. This software package provides a beam element based on Mindlin's theory of plates.

3. Task description

A well-known criterion for failure (bifurcation in the solution) for models using elasto-plastic formulation is the Hill's failure [1] criterion based on the second order work. The second order work is defined at the material point by:

$$u = d^2W = d\sigma : d\varepsilon,$$

$$[u] = \frac{N}{m^2} = \frac{J}{m^3}.$$

Hill's condition of stability (Hill 1957) is defined by:

$$d^2W > 0$$

A related measure is the strain energy, which is the integrated value of the second order work over a volume V :

$$U = \int_V udV = \int_V d\sigma : d\varepsilon dV,$$

$$[U] = J = N \cdot m.$$

Unfortunately, the software package Plaxis does not calculate the second order work. Furthermore, it only provides a small set of output variables at the element nodes. The objective is to calculate the second order work for this kind of beam elements as a post-processing after finishing the FE-simulation based only on the nodal variables provided in the programs output. The output provided at each node:

- u_x : total nodal displacements in direction x
- u_y : total nodal displacements in direction y
- φ : total nodal rotation
- N : Normal force extrapolated to the node*
- G : Shear force extrapolated to the node*
- M : Bending moment extrapolated to the node*

*The extrapolation function is unknown.

From NFEM Ch11 [3] we found a formula to calculate the strain energy W for a Timoshenko beam element. Is it also valid for a beam based on Mindlin's theory of plates? If not, please provide the correct solution.

$$\begin{aligned} W = \int_{L_0} \left(N^0 \cdot e + \frac{1}{2} E \cdot A_0 \cdot e^2 \right) dX + \int_{L_0} \left(G^0 \cdot \gamma + \frac{1}{2} G \cdot A_0 \cdot \gamma^2 \right) dX \\ + \int_{L_0} \left(M^0 \cdot \kappa + \frac{1}{2} E \cdot l_0 \cdot \kappa^2 \right) dX \end{aligned}$$

4. Expected Results

- Show a way to calculate the second order work for this kind of beam element with the data given.
- Show a way to calculate the strain energy of one beam element.

Literature

- [1] R. Hill, On uniqueness and stability in the theory of finite elastic strain. journal of the Mechanics and Physics of solid, V. 5, 4, pp. 229–241, 1957.
- [2] Information form the Plaxis manual on beam element defition based on Mildlin plate theory.
- [3] <http://www.colorado.edu/engineering/cas/courses.d/NFEM.d/NFEM.Ch11.d/NFEM.Ch11.Slides.d/NFEM.Ch11.Slides.pdf>

Problem 7. Mathematical Model of Residential Storage Water-heating System

MelissaClimate, www.seemelissa.com

Kristiyan Boyanov, kristiyan.boyanov@gmail.com

About the company and business context

MClimate is an IoT company for intelligent climate control. Our aim is to develop devices that can retrofit to your old appliances and make them smart. But what does smart exactly mean? It all comes down to our philosophy. All of our products obey the same rules:

- know your preferences and habits / will know what's needed to make you comfortable;
- know when are you going to come home;
- analyze your heating & cooling systems in order to make them more energy-efficient.

Introduction

The low-cost commonly available water heater is inherently a lossy device and this means that a good percentage of the energy consumed heating water ends up being lost as heat to the surroundings. Informal measurements and practice show that if a family of four persons switches on and off the water heater just when required savings of up to 80% are recorded.

Our goal is to use model of the water heater to perform energy saving calculations and help the customer operate their water heaters in the most energy-efficient way.

Modelling is nothing but converting physical information in to mathematical form. Mathematical model plays vital role for identification and analysis of the system. The model updates the simulated temperature of the mass of water periodically where the rate of change in temperature depends on the system inputs, outputs and losses.

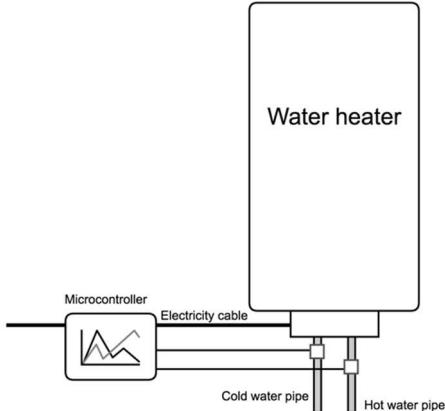


Fig. 1

Problem – Mathematical model of home water heating system

An electric water heater consists of: an inner steel tank, that holds the water being heated, insulation that surrounds the tank so as to decrease the amount of heat loss, pipe to allow cold water to enter the tank, pipe to allow hot water to leave the tank, thermostat that reads and controls the temperature of the water inside the tank, heating element that heats the water by means of electricity and other components for safety and maintenance.

Water temperature inside the heater is controlled by the mechanical thermostat.

The temperature may usually be set by the user somewhere in between 40 and 70 °C.

A microcontroller is used to gather realtime information from a water heater. The information is collected by different sensors (Fig. 1) and consists of data about: current temperature of the cold water pipe, temperature of the hot water pipe, environment temperature (home temperature), electric current and voltage. Important notice – the temperature sensors of the cold and hot water pipes are installed ONTO the pipe itself. There is significant temperature loss depending on the pipe diameter, material and others.

The task is to create model of the dynamics of the water heater:

- The water heater model should simulate the temperature variations over time.
- For the model we can assume that the user has set the mechanical thermostat to 70 °C.
- The model should take into account that the temperature in the tank is updated by:
 - Energy input from the heater;
 - Heat loss through the insulation;
 - Heat loss due to cold and hot water mixture when consuming hot water;

- Unknown consumption of hot water throughout the day.
- The water heater model should be validated against measured data (Fig. 2).
- Measured data and more information can be found at
seemelissa.com/esgi2016

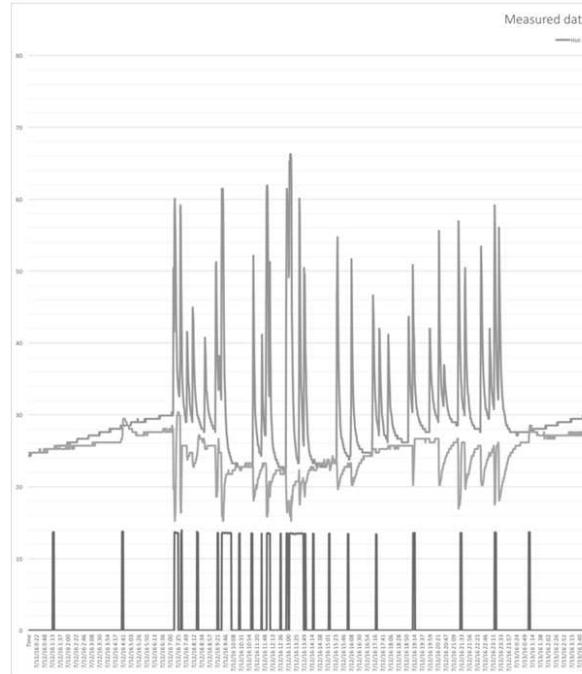


Fig. 2

FINAL REPORTS

Multi-frame Denoising of Still Images

Stanislav Harizanov, Dimo Dimov, Nikola Naidenov, Savka Kostadinova

1. Problem formulation

The representative of *MM Solutions AD* Mr. Angel Ivanov posted the following problem to the European Study Group with Industry'120 (*ESGI'120*): “A sequence of 8 images are captured quickly one-by-one while camera is held in hand and unintentional hand shake is present. When capturing in low light conditions, the images appear blurred and noisy. The target is to combine multiple frames to produce an image with less noise and less (ideally no) blur. The processing algorithm should not be too computationally expensive.”



Figure 1. Single-frame noisy & blurry 2592×4608 image

On Fig. 1 one of the captured image frames is illustrated. One can see that the input data are of high resolution (2592×4608 pixels) and contain great variety of shapes and patterns. Indeed, there are figures, letters, shiny objects, texture, moving objects, repetitive patterns, etc. The noise is “pink”, typical for

electronic circuits like the analog amplifiers embedded in the image sensor. It is Gaussian, but not white, since low frequency amplitudes are higher. The blur kernel changes smoothly within the image due to the predominantly rotational camera shake and the usage of rolling shutter in the image sensor.

The first half of this report is devoted to multi-frame denoising. We construct and numerically solve a penalized energy-minimization optimization problem. The efficiency of the proposed algorithm lies on the observation that our multi-frame denoising approach is equivalent to a single-frame one, for which the input image is the corresponding convex combination of the eight original frames. Therefore, we do not need additional memory resources when the number of frames increases and the execution time is also independent of this number.

Apart from blur, hand shake causes the images not to be aligned to each other. Alignment was not part of that project as the given images were already pre-processed and an algorithm, which aligns the images with up to 2pix error at 4000×3000 image resolution, had already been applied. Nevertheless, in the second half of this report we perform our own alignment on the raw data, based on inverse filtering. This allows us to determine the *hodograph* of the frame centers (see Fig. 10), thus it provides us with additional information on the characteristics of the hand shake, respectively of the blur kernel.

2. Multi-frame denoising via penalized TV norm minimization

2.1. Mathematical model

Denote by $\{\mathbf{f}_i\}_{i=1}^m$ the set of single-framed images of consideration. We consider the following penalized optimization problem

$$(1) \quad \bar{\mathbf{u}} = \underset{\mathbf{u} \in [0, 255]^N}{\operatorname{argmin}} \left\{ \| |\nabla \mathbf{u}| \|_1 + \sum_{i=1}^m \lambda_i \|\mathbf{u} - \mathbf{f}_i\|_2^2 \right\}.$$

The box constraint $\mathbf{u} \in [0, 255]^N$ is due to input data characteristics, namely we are given 8-bit gray-scale bitmap images. Here N corresponds to the image size. Note that the gray-scale intensity of the output $\bar{\mathbf{u}}$ is not integer-valued as the discretized version of (1) is hard and numerically expensive to solve. Instead, as post-processing, we round the intensity value of each pixel to the nearest integer.

The regularization term $\| |\nabla \mathbf{u}| \|_1$ measures the total variation (TV) norm of the discrete gradient ∇ of \mathbf{u} . For the computation of $\nabla \mathbf{u}$ we use forward finite differences and Neumann boundary conditions. This energy norm, also known as *mixed $\ell^{2,1}$ norm* since it is the sum of the Euclidean lengths of the 2D pixel-wise intensity gradients, was first introduced in [1]. The authors showed

improved robustness of such type of filters towards the input image noise in comparison to the corresponding standard ℓ^2 -based filters and until now the TV norm remains the most favorite choice for a regularization term in optimization denoising models.

The data fidelity term $\sum_{i=1}^m \lambda_i \|\mathbf{u} - \mathbf{f}_i\|_2^2$ controls the regularization process. Its choice is statistically motivated, as least-squares are the Maximum Likelihood Estimate for Gaussian-modeled noise. The penalizers $\{\lambda_i\}_1^m$ are free, non-negative parameters (weights) that are manually chosen by the user. The higher the λ_i the smaller $\|\bar{\mathbf{u}} - \mathbf{f}_i\|_2^2$, thus the closer $\bar{\mathbf{u}}$ to \mathbf{f}_i . Choosing adequate values for the penalizers is crucial for the quality of the result. Too large λ_i gives rise to incomplete noise removal, while too small λ_i leads to oversmoothing of the output and loss of important structural information. Later, we propose a practical algorithm how to optimize these values.

Let $\Lambda = \sum_i \lambda_i$. Since

$$\begin{aligned} \Lambda \|\mathbf{u} - \sum_{i=1}^m \frac{\lambda_i}{\Lambda} \mathbf{f}_i\|_2^2 - \sum_{i=1}^m \lambda_i \|\mathbf{u} - \mathbf{f}_i\|_2^2 &= \\ \Lambda \|\mathbf{u}\|_2^2 - 2\Lambda \left\langle \sum_{i=1}^m \frac{\lambda_i}{\Lambda} \mathbf{f}_i, \mathbf{u} \right\rangle + \Lambda \left\| \sum_{i=1}^m \frac{\lambda_i}{\Lambda} \mathbf{f}_i \right\|_2^2 - \sum_{i=1}^m \lambda_i \|\mathbf{u}\|_2^2 + 2 \sum_{i=1}^m \lambda_i \langle \mathbf{f}_i, \mathbf{u} \rangle - \sum_{i=1}^m \lambda_i \|\mathbf{f}_i\|_2^2 &= \\ \Lambda \left\| \sum_{i=1}^m \frac{\lambda_i}{\Lambda} \mathbf{f}_i \right\|_2^2 - \sum_{i=1}^m \lambda_i \|\mathbf{f}_i\|_2^2 & \end{aligned}$$

is independent of \mathbf{u} , the minimizer $\bar{\mathbf{u}}$ of (1) coincides with the minimizer of

$$(2) \quad \bar{\mathbf{u}} = \underset{\mathbf{u} \in [0, 255]^N}{\operatorname{argmin}} \left\{ \|\nabla \mathbf{u}\|_1 + \Lambda \|\mathbf{u} - \sum_{i=1}^m \frac{\lambda_i}{\Lambda} \mathbf{f}_i\|_2^2 \right\}.$$

Hence, the proposed multi-frame denoising model can be interpreted as a single-frame penalized denoising of a convex combination of the input multi-frame data. The choice of the convex combination is again user-dependent and may vary from application to application. Once it has been fixed, the quality of the output depends solely on the parameter Λ , which tuning is much faster than the tuning of the whole set $\{\lambda_i\}$. In all our numerical simulations, we set $\lambda_i = \lambda, \forall i$. Therefore, we apply a denoising procedure on the averaged input image, the one that has been so far considered by *MM Solutions AD* as potential candidate for denoised output. From this point of view, the proposed denoising procedure improves the quality of the final result.

Both $\| |\nabla \mathbf{u}| \|_1$ and $\|\mathbf{u} - \cdot\|_2^2$ are convex, proper, lower-semi-continuous functions. Therefore, there is a one-to-one correspondence between the penalized problem (2) and the constrained one

$$(3) \quad \bar{\mathbf{u}} = \operatorname{argmin}_{\mathbf{u}} \| |\nabla \mathbf{u}| \|_1 \quad \text{subject to} \quad \|\mathbf{u} - \sum_{i=1}^m \frac{\lambda_i}{\Lambda} \mathbf{f}_i\|_2^2 \leq \tau_\Lambda \cap \mathbf{u} \in [0, 255]^N.$$

In other words, for each $\Lambda > 0$ there exists a unique $\tau_\Lambda > 0$ such that the minimizers of (2) and (3) coincide. The constrained formulation of the optimization problem allows us to analyze existence and uniqueness of the minimizer. It is well known that as long as the constrained set is non-empty and do not contain a global minimizer of the cost function, problem (3) has a unique solution that lies on the boundary of the constraint. Since the global minimizers of $\| |\nabla \mathbf{u}| \|_1$ are the constant images and the constrained set is never empty (the input image is always inside) we conclude that there exists $\Lambda_{min} > 0$ such that for every $\Lambda > \Lambda_{min}$ problem (2) admits a unique solution that additionally satisfies $\|\mathbf{u} - \sum_{i=1}^m \frac{\lambda_i}{\Lambda} \mathbf{f}_i\|_2^2 = \tau_\Lambda$.

Apart from the well-posedness of the mathematical problem, the above analysis gives us another perspective, namely we realize that $\bar{\mathbf{u}}$ is the most regular image (with respect to the TV norm) on the intersection of an N -dimensional sphere of (unknown) radius $\sqrt{\tau_\Lambda}$, centered at the convex combination $\sum_{i=1}^m \frac{\lambda_i}{\Lambda} \mathbf{f}_i$ of input frames, and the N -dimensional cube $[0, 255]^N$. The radius $\sqrt{\tau_\Lambda}$ needs to be small enough so that no constant images lie inside the ball, thus Λ needs to be big enough (the $\Lambda \leftrightarrow \tau_\Lambda$ correspondence is reciprocal).

In theory, we do not know how to exactly estimate Λ_{min} , but from a practical point of view the check if the chosen Λ is admissible or not is quite simple - we just check if the output $\bar{\mathbf{u}}$ is constant or not. Anyway, we are interested in $\Lambda \gg \Lambda_{min}$ since we want to preserve all the edges of the input image and not to oversmoothen it, therefore the theoretical peculiarities around Λ_{min} are outside the scope of our application.

2.2. Numerical algorithm

We apply *Alternating Direction Method of Multipliers* (ADMM) to the reformulated version

$$\operatorname{argmin}_{x, y_1, y_2} \left\{ \iota_{[0, 255]^N}(x) + \|y_1\|_1 + \sum_{i=1}^m \lambda_i \|y_2 - \mathbf{f}_i\|_2^2 \right\} \quad \text{s.t.} \quad \begin{pmatrix} L := \nabla \\ I \end{pmatrix} x = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

of (1). Here, $\iota_{[0, 255]^N}$ is the indicator function of the N -dimensional cube $[0, 255]^N$, which takes value 0 inside the set and $+\infty$ outside it.

Algorithm 0.0.1 (ADMM) *Input:* $\{\mathbf{f}_i\}$ (a sequence of input frames) and $\{\lambda_i\}$ (the corresponding weights). *Output:* $\mathbf{u} = x^K$ (denoised image).

For $k = 1, \dots, K$.

$$1. \quad x^{(k+1)} = \operatorname{argmin}_x \left\{ \iota_{[0,255]^N}(x) + \frac{\gamma}{2} (\|b_1^{(k)} + Lx - y_1^{(k)}\|_2^2 + \|b_2^{(k)} + x - y_2^{(k)}\|_2^2) \right\}$$

$$x^{(k+1)} = P_{[0,255]^N} \left\{ (I + L^T L)^{-1} (L^T (y_1^{(k)} - b_1^{(k)}) + (y_2^{(k)} - b_2^{(k)})) \right\}.$$

$$2. \quad y_1^{(k+1)} = \operatorname{argmin}_y \left\{ \|y_1\|_1 + \frac{\gamma}{2} \|b_1^{(k)} + Lx^{(k+1)} - y\|_2^2 \right\}$$

$$y_1^{(k+1)} = (I - P_{B_{\|\cdot\|_*}(1/\gamma)}) (b_1^{(k)} + Lx^{(k+1)}).$$

$$3. \quad y_2^{(k+1)} = \operatorname{argmin}_y \left\{ \sum_{i=1}^m \lambda_i \|y - \mathbf{f}_i\|_2^2 + \frac{\gamma}{2} \|b_2^{(k)} + x^{(k+1)} - y\|_2^2 \right\}$$

$$y_2^{(k+1)} = \frac{1}{\gamma + 2\Lambda} \left(\gamma(b_2^{(k)} + x^{(k+1)}) + 2 \sum_i \lambda_i \mathbf{f}_i \right)$$

$$4. \quad b_1^{(k+1)} = b_1^{(k)} + Lx^{(k+1)} - y_1^{(k+1)}, \quad b_2^{(k+1)} = b_2^{(k)} + x^{(k+1)} - y_2^{(k+1)}.$$

The algorithm theoretically converges for any choice $\gamma \in (0, 1)$ of its free parameter. In general, tuning γ may improve the convergence rate, but in our particular setting we didn't observe significant influence of the γ value on it. Therefore, we always set $\gamma = 0.3$ in our experiments. We also set $K = 150$.

In Step 1. we use direct numerical solver for the linear system

$$(I + L^T L)^{-1} (L^T (y_1^{(k)} - b_1^{(k)}) + (y_2^{(k)} - b_2^{(k)})).$$

The matrix $I + L^T L$ is symmetric, positive definite and well-conditioned, so we do not experience numerical instability. Even though the used solver is one of the most efficient ones for such systems, the computational time heavily depends on the image size. Thus, such a direct approach becomes impractical for high resolution data. As a future work, we plan on applying an iterative numerical solver that takes as an initial guess the already computed $x^{(k)}$ at the previous step. The orthogonal projection $P_{[0,255]^N}$ is straightforward and component-wise. Step 2. is performed by coupled soft shrinkage with threshold γ^{-1} , again component-wise. Steps 3. and 4. involve only basic arithmetic operations.

2.3. Experimental results

We are given five 2592×4608 noisy frames ($m = 5$) like the one on Fig. 1. We consider equal λ_i (i.e., we denoise the averaged input image). On Fig. 2 we plot the denoised output of our algorithm with properly chosen Λ .



Figure 2. 5-frame denoised 2592×4608 image

Compared to the noisy input, the resulting image has no visible noise, yet all the objects in it are clearly visible and do not look artificial (little high-frequency noise creates an impression of sharper edges and the human brain classifies the image as “good looking”). The only problem is the desktop screen, which contains moving objects. There the multi-frame denoising gives rise to additional motion blur and the output is fuzzy and unnatural. We will address this problem later in the section.

Measuring the TV norm of inputs and output gives us a quantitative information regarding the quality of the denoising process. Indeed, the TV norm of the five individual input frames varies between $8.92\text{e+}07$ and $1.34\text{e+}08$, the TV norm of the averaged input image is $5.36\text{e+}07$, while the TV norm of the denoised output is only $1.77\text{e+}07$. By definition, noise is unstructured and highly irregular. Therefore, energy decrease is a strong evidence of successful denoising.

$$TV(\mathbf{f}_1) = 7.14e+06$$



$$TV(\mathbf{f}_2) = 1.20e+07$$



$$TV(\mathbf{f}_{avg}) = 5.13e+06$$



$$\Lambda = 10^5, TV(\mathbf{u}_{exact}) = 2.74e+06$$



$$\Lambda = 2 \cdot 10^4, TV(\mathbf{u}_{over}) = 2.02e+06$$



$$\Lambda = 6 \cdot 10^5, TV(\mathbf{u}_{under}) = 3.85e+06$$



Figure 3. Visual and energy comparison of various input and output images

As expected, averaging is also a denoising procedure and we observe two times smaller energy for the averaged image than for the single-frame ones. On top of that, we managed to decrease the former energy three times more via applying our algorithm without losing any significant structural information.

For the next experimental setup, we choose the blurruest part of the input image from Fig. 1, which is of size 940×1225 . On Fig. 3 we illustrate the different outputs with respect to too small Λ (oversmoothing), optimal Λ , and too large Λ (undersmoothing). When $\Lambda = 10^5$ is optimal we calculate two times smaller energy for the output \mathbf{u}_{exact} compared to the averaged input \mathbf{f}_{avg} . When $\Lambda = 2 \cdot 10^4$ is too small, apart from the noise we remove also the *Coca Cola* and *Schweppes* texts from the drink cans, the result looks unnatural and bad. When $\Lambda = 6 \cdot 10^5$ is too large, the output is visually OK, but the presence of noise in it is captured by the TV norm value, which is significantly bigger than the one for \mathbf{u}_{exact} .

A simple, but practical criterion for choosing a good Λ is based on segmenting difference images, as shown on Fig. 4. When Λ is too small, the segmented difference image $|\mathbf{u}_{over} - \mathbf{f}_{avg}| > 8$ between output and input captures some of the structural information, meaning that these pieces of information might be missing in the output (i.e., oversmoothing). For the particular example, most of the image edges are sharp enough and survive the oversmoothing, but the low-contrast ones such as the text on the drink cans do not. When Λ is (close to) optimal, the corresponding image $|\mathbf{u}_{exact} - \mathbf{f}_{avg}| > 3$ is unstructured and irregular. No image edges are clearly visible, thus there are no indications of oversmoothing and pure denoising is performed. To test for undersmoothing, we replace the input image \mathbf{f}_{avg} by the oversmoothed one \mathbf{u}_{over} , choose larger Λ than the one for \mathbf{u}_{over} and if there is still unstructured data in the corresponding segmented difference image $|\bar{\mathbf{u}} - \mathbf{u}_{over}| > \varepsilon$ then a residual noise is present in $\bar{\mathbf{u}}$ and we decrease Λ . When Λ is (close to) optimal, the difference image looks like a sketch of the original one – only original image edges should be visible, like in the case for $|\mathbf{u}_{exact} - \mathbf{u}_{over}| > 5$ on Fig. 4.

Another criterion for the meaningfulness of the Λ choice deals with the computation of various averaged energies and the Central Limit Theorem. Let us denote by \mathbf{u}_i the output of the single-frame denoising process

$$\mathbf{u}_i = \operatorname{argmin}_{\mathbf{u} \in [0, 255]^N} \left\{ \| |\nabla \mathbf{u}| \|_1 + \frac{\Lambda}{m} \|\mathbf{u} - \mathbf{f}_i\|_2^2 \right\}, \quad \forall i = 1, \dots, m.$$

Note that, when all the λ_i are equal, they are exactly $\lambda_i = \Lambda/m$. Now, for every $i = 1, \dots, m$ we generate the averaged output $(\mathbf{u}_1 + \dots + \mathbf{u}_i)/i$ and compute its

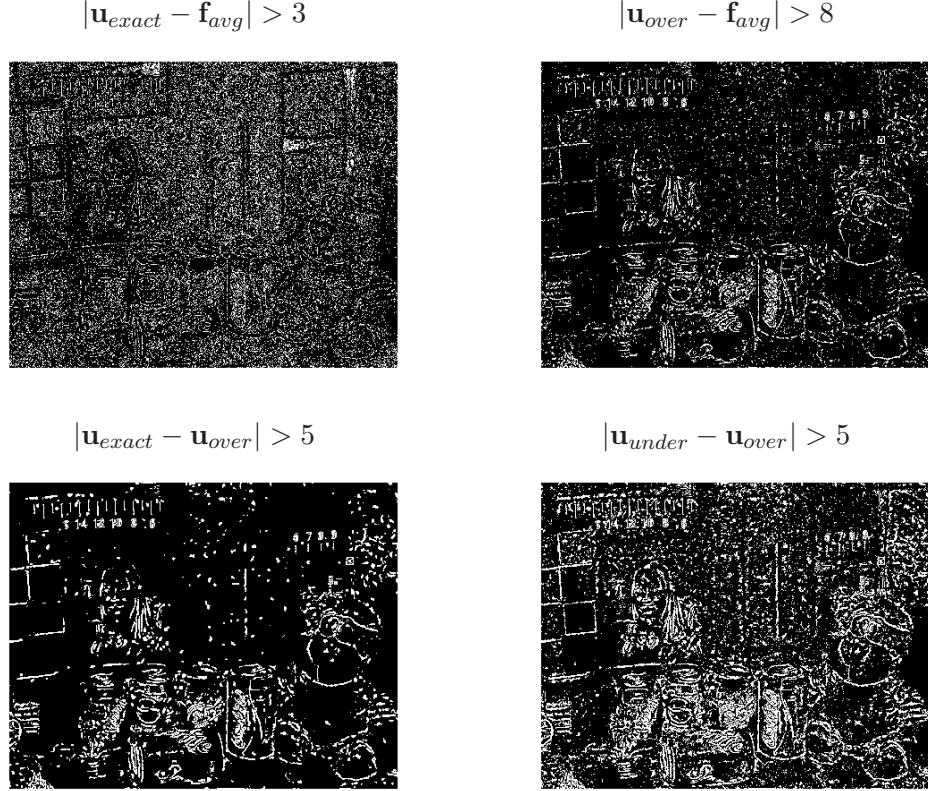


Figure 4. Practical visual criterion for choosing a good parameter Λ

TV norm. If the TV norm gradually decreases when i increases then the chosen Λ is too large and there is still residual noise in the outputs \mathbf{u}_i , see Fig. 5. The largest Λ for which the TV norm remains stable during the averaging process is the optimal one.

A very important remark here is that, unlike the input scenario where taking a multi-frame input (1) or its corresponding single-frame convex combination (2) does not affect the output, the averaged image $(\sum_{i=1}^m \mathbf{u}_i)/m$ significantly differs from the minimizer $\bar{\mathbf{u}}$ of (2) for the same Λ . The reason is hidden in (3) and the non-linear relation $\Lambda \leftrightarrow \tau_\Lambda$ between penalizers and constraints. We observe that $\tau_{\Lambda/m} \gg m\tau_\Lambda$ so the constrained set that corresponds to the single-frame denoising process is too large and gives rise to oversmoothed solutions \mathbf{u}_i . Comparing the

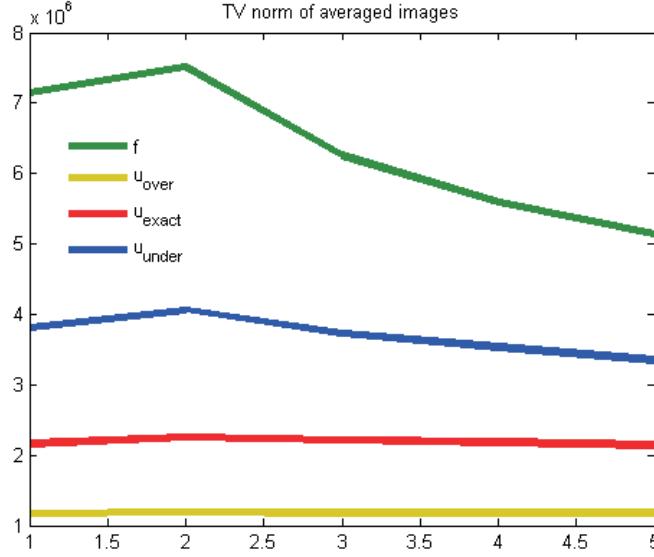
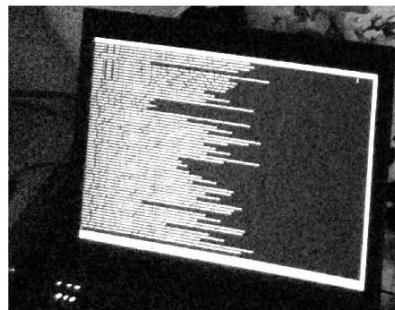


Figure 5. Quantitative criterion for choosing a good parameter Λ

numerical results on Fig. 3 with the numerical results on Fig. 5 we confirm the above argument. Indeed, for $\Lambda_{\text{under}} = 6 \cdot 10^5$ we have $TV(\mathbf{u}_{\text{under}}) = 3.85e+06$, while $TV((\sum_{i=1}^5 \mathbf{u}_i)/5) = 3.35e+06$. For $\Lambda_{\text{exact}} = 10^5$ we have $TV(\mathbf{u}_{\text{exact}}) = 2.74e+06$, while $TV((\sum_{i=1}^5 \mathbf{u}_i)/5) = 2.14e+06$. For $\Lambda_{\text{under}} = 2 \cdot 10^4$ we have $TV(\mathbf{u}_{\text{over}}) = 2.02e+06$, while $TV((\sum_{i=1}^5 \mathbf{u}_i)/5) = 1.17e+06$. Therefore, the proposed multi-frame denoising approach is conceptually better than simply averaging single-frame denoised images.

Finally, we make a short discussion on how to get rid of the additional motion blur in moving objects. Since our denoising approach is based on data regularization, the blur characteristics do not improve throughout the process and the output might be even blurrier than the input. This is especially the case with moving objects as the text on the desktop in the bottom left corner of the original frame, see Fig. 1. Therefore, if we want to control the blur impact it is better to apply single-frame denoising to the least blurry input frame as illustrated on Fig. 6. Since the noise is predominantly “pink”, its characteristics are strongly related to the camera parameters and should not change from frame to frame, when the camera parameters remain untouched. Hence, in absence of blur, the TV norms of all the frames \mathbf{f}_i should be alike. However, the hand-shakes in between the frames change their direction and amplitude and so does the blur.

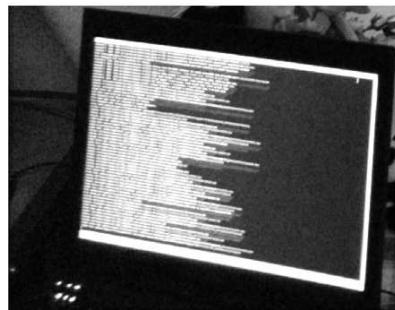
$$TV(\mathbf{f}_2) = 1.77e+07$$



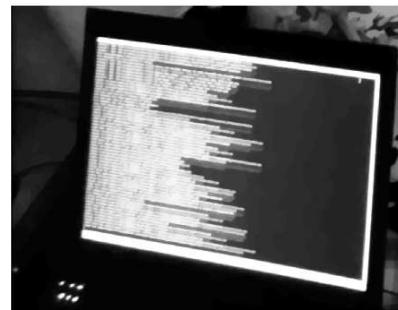
$$TV(\mathbf{f}_5) = 1.03e+07$$



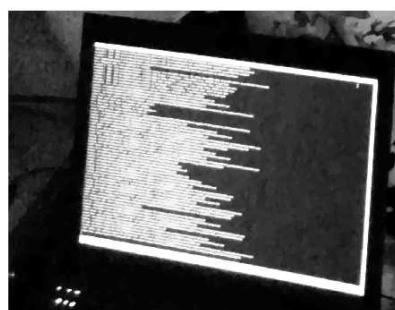
$$TV(\mathbf{f}_{avg}) = 8.36e+06$$



$$TV(\mathbf{u}_{exact}) = 3.23e+06$$



$$TV(\mathbf{u}_2) = 5.37e+06$$



$$TV(\mathbf{u}_5) = 3.00e+06$$

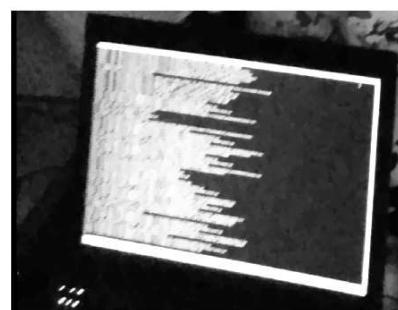


Figure 6. Motion blur: Comparison of different multi-frame and single-frame denoisings

Blurring is also a form of data regularization, so the input frames with higher TV norm possess less blur. When we apply single-frame denoising to the frame with largest TV norm (in our case this is \mathbf{f}_2) and with λ , larger than the optimal one Λ_{exact}/m we are in an undersmoothing regime but we control the blur (see bottom, left of Fig. 6). The choice of \mathbf{f}_2 is significant, because if we take the bluriest frame \mathbf{f}_5 that has the smallest TV norm we end up with a quite blurry output (see bottom, right of Fig. 6).

3. Inverse filtering approach for reducing the noise from camera motion

In this section we perform our own alignment on the raw data, based on inverse filtering. This allows us to determine the *hodograph* of the frame centers (see Fig. 10), thus it provides us with additional information on the characteristics of the hand shake, respectively of the blur kernel. The latter should be useful for successfully deblurring the denoised output of our multi-frame denoising algorithm, which is the next step in our future work on the project.

3.1. Setting of the method and notations

To be more precise, let the initial frame series be f_1, \dots, f_n , where each $f_i = f_i(x, y)$ is a gray-scale image of pixels (x, y) of range $(X \times Y)$ in the plane. Next, we shall denote by $\mathcal{F}(f_i)$ the discrete Fourier transforms of the frames, i.e. $\mathcal{F}(f_i) = \mathcal{F}(f_i)(u_x, u_y)$, $(u_x, u_y) \in (X \times Y)$, where $(X \times Y)$ plays here the role of the frequency definition area, and $\vec{u} = (u_x, u_y)$ is a frequency vector.

By (x_i, y_i) , $i \in \{1, \dots, n\}$, we will denote the position of the frame f_i in the plane \mathbb{R}^2 , i.e. the origin of its local coordinate system. Then, we will use the notations $(\Delta x_i, \Delta y_i)$, $\Delta x_i = x_i - x_{i-1}$, $\Delta y_i = y_i - y_{i-1}$, $i = 2, \dots, n$ for the shifts between the positions of two consecutive frames f_{i-1} and f_i . Sometimes we will omit the index “ i ”, when the pair (f_{i-1}, f_i) can be understood from the context. For example, we will write the shift $(\Delta x_i, \Delta y_i)$ as (τ_x, τ_y) where the values τ_x and τ_y are usually obtained from the data. Also, we will use vector notations $\vec{\tau}$ for simplicity, and for suggesting the idea that similar considerations are valid in \mathbb{R}^3 (e.g., for 3D modeling of the considered scene).

It is essential for the proposed approach that we know the actual order of the frames $(f_i, i = 1, 2, \dots, n)$. For definiteness we work with this sequence, although, by symmetry, one can use the same but in the reverse order.

3.2. The inverse approach

We assume that the movement of the video camera between two successive frames $f_{i-1} \rightarrow f_i$ is evenly and linear. So if we know the shift $\vec{\tau}$ between the

two frames (it can be considered like a “video stabilization” problem, see [2]), we have to model only the movement noise by the formula, [3]:

$$(4) \quad H(\vec{u}) = \int_0^T \exp(-j 2\pi \vec{u} \circ (\vec{\tau}/T)t) dt = T \frac{\sin(\pi \vec{u} \circ \vec{\tau})}{\pi \vec{u} \circ \vec{\tau}} \exp(-j\pi \vec{u} \circ \vec{\tau}),$$

where $H(\vec{u})$ is the Fourier spectrum of the modelled noise, \vec{u} is an arbitrary vector in the frequency domain \mathbb{R}^2 , $\vec{\tau}/T$ is the velocity of the image caused by camera movement (an uniform rectilinear motion) during the time interval $(0, T]$ of open shutter, $j^2 = -1$, and $\vec{u} \circ \vec{\tau}$ is the scalar product of the vectors.

For the length of the time interval we have $T = \mu/\langle \text{fps} \rangle$ where $\langle \text{fps} \rangle$ is the value of the parameter “frames per second” of the camera, and μ , $0 < \mu < 1$ is a specific constant of the camera. On account of electronic realization, the time-interval $(1 - \mu)/\langle \text{fps} \rangle$ of closed shutter is (or can be selected) negligible compared to T . So, in our case we will take $\mu \approx 1$, i.e. $T = 1/\langle \text{fps} \rangle$. For definiteness we will assume that $\langle \text{fps} \rangle = 15s^{-1}$, i.e. $T = 0.066s$. Actually, if we normalize the considered images according to their intensity (which is an established practice), then we can exclude T from consideration, accepting that $T \equiv 1$.

This model of the noise reflects the physical interpretation of blurring the image in a given frame f_i , as a uniform accumulation of the scene captured during the time of exposure, starting from an empty accumulator (i.e. the 2D input pixel matrix of the camera) in the position of the previous frame f_{i-1} . This means that we can use the “inverse filtering” approach (see [3]) to get a restored version \hat{f}_i of the frame f_i relatively to the position of the previous reference frame f_{i-1} , namely:

$$(5) \quad \mathcal{F}(\hat{f}_i)(\vec{u}) = \frac{\mathcal{F}(f_i)(\vec{u})}{H(\vec{u})}, \quad \vec{u} = (u_x, u_y) \in (X \times Y) \subset \mathbb{R}^2,$$

where $\mathcal{F}(\hat{f}_i)$ and $\mathcal{F}(f_i)$ are the Fourier-spectra of \hat{f}_i and f_i , respectively.

However, the direct inverse filtering $(H(\vec{u}))^{-1}$ does not work properly if there is external noise (beyond that caused by motion), for example, noise from temperature and/or wind conditions in the real scene, moving objects therein, distortion of optics, etc. For those real situations, there are known additional filtering approaches (of high frequencies suppression) like Gauss filter, Butterworth filter, Wiener filter, etc., [3].

Indeed, the 2D surface $|H(\vec{u})|$, i.e. modulus of the filter $H(\vec{u})$, is a 2D cylinder of element curve: $T \cdot \text{Sinc}(\alpha) = T \frac{\sin(\alpha)}{\alpha}$, $\alpha \in \mathbb{R}$, which direction is $\pi/2$ rotated towards the shift $\vec{\tau}$, and into this direction the filter asymptotically approaches

zero:

$$(6) \quad |H(\vec{u})| = 0 \Rightarrow \vec{u} \circ \vec{\tau} = k \Leftrightarrow u_y = -\frac{\tau_x}{\tau_y}u_x + \frac{k}{\tau_y},$$

where k is an arbitrary integer. Of course, in the perpendicular direction $\vec{\tau}$ the filter has a constant value, but in the considered bounded area of frequencies $(u_x, u_y) \in (X \times Y)$ this is not a problem. Outside $(X \times Y)$ the spectrum is considered as $\equiv 0$, and for the application of the FFT it is adopted formally a periodicity of the image (horizontally and vertically).

On the basis of experiments we will use the following *heuristic approach*: We consider reverse time as well as the corresponding interpretation of the model, namely – as a scene for accumulation during the open shutter, instead of the restored image \hat{f}_i , we take the one in the given frame f_i , i.e. just contrary of [3]. Thus, from a standpoint of computability, we consider the reverse process – from the image f_i , whose position $(-\tau_x, -\tau_y)$ relatively to the previous frame f_{i-1} is known, we decumulate it uniformly to zero, i.e. reaching back the position of f_{i-1} :

$$(7) \quad \mathcal{F}(\hat{f}_i)(\vec{u}) = \mathcal{F}(f_i)(\vec{u}) \cdot H(\vec{u}), \quad \vec{u} = (u_x, u_y) \in (X \times Y) \subset \mathbb{R}^2,$$

where the filter $H(\vec{u})$, see (4), now has the desired property of “suppressing the high frequencies” (see Fig. 7 to compare the spectra).

At this stage, the experiments confirm the above heuristics, which we will consider as a practical innovation only, whose theoretical confirmation (or rejection for some cases) is left for a future work.

3.3. Positioning of the restored frame

It can be shown (and this is one of the innovations offered here) that: *the restored version \hat{f}_i is positioned exactly in the middle of the two frames: the previous f_{i-1} and the present f_i , $i = 2, \dots, n$.*

Thus, to compare the given frame f_i with the restored version \hat{f}_i we have to transfer them (both) in a reference point (x_0, y_0) . Of course, the latter may be chosen arbitrary, before starting the comparison (or sequence of comparisons). An appropriate choice of (x_0, y_0) is the position of the first frame (x_1, y_1) or the geometric center of the all frames: $(x_0, y_0) = \frac{1}{n} \sum_{i=1}^n (x_i, y_i)$.

3.4. The shifts between the frames

What concerns the preliminary assessment (calculation) of the shift $\vec{\tau} = (\tau_x, \tau_y)$ between two frames, it can be seen as an accompanying problem for “video stabilization of unstable (manual) camera”. A similar task for the video

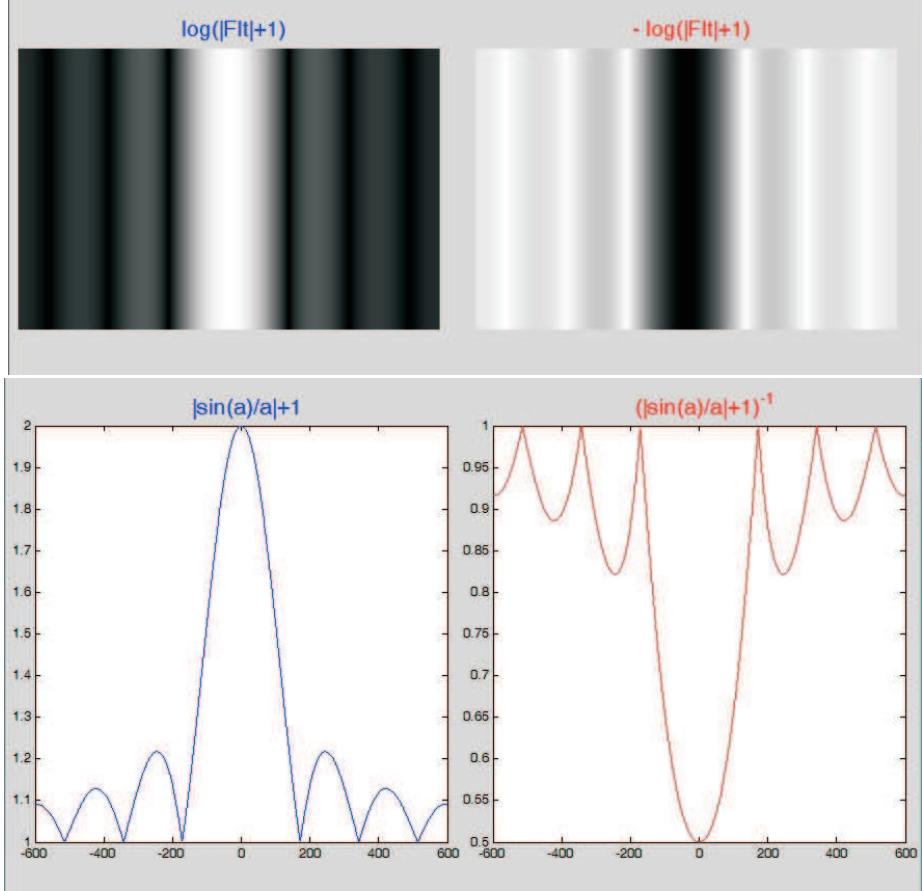


Figure 7. Spectrum of filters H and H^{-1} : First row: spectra of the Fourier modules of the direct (left, (a)) and the inverse (right, (c)) filters; Second row (b,d): 1D element curve of the corresponding spectra, which are 2D cylinders. For better visualization a purely horizontal shift $\vec{\tau} = (\tau_x, 0)$ is chosen

camera of a modern mobile smart-phone, in real-time, was successfully resolved and described in [2]. This solution is very appropriate for the present case as well, but it is realized on C/C++, and the transcoding of the algorithm to MatLab is proved laborious. Therefore, for the illustrative experiments here, we use the classical approach by “maximum cross-correlation” between two frames (see e.g. [3]).

More precisely, the position of this maximum is defined by the shift $\vec{\tau} = (\tau_x, \tau_y)$, which we are looking for for each pair $(f_{i-1} \rightarrow f_i)$, $i = 2, \dots, n$. Of course, the cross-correlation has also to be calculated in the frequency domain by fast Fourier transform (FFT), which is a known practice for efficiency of the calculation (although insufficient in the sense of [2]).

Basic algorithm for a pair of consecutive frames: (f_{i-1}, f_i) , $i = 2, \dots, n$.

1. Find the shift $\vec{\tau} = (\tau_x, \tau_y)$ of the current frame f_i relatively to the previous one f_{i-1} .
 - Calculate the current frequency spectrum $\mathcal{F}(f_i)$ in the 2D frequency domain by FFT. For efficiency, we assume that the spectrum $\mathcal{F}(f_{i-1})$ of the previous frame is already known from the algorithm execution for the previous pair (f_{i-2}, f_{i-1}) .
 - Calculate the 2D cross-correlation $c(x, y) = \mathcal{F}^{-1}(\overline{\mathcal{F}}(f_{i-1})\mathcal{F}(f_i))$, $(x, y) \in (X \times Y)$, where $\overline{\mathcal{F}}(f_{i-1})$ is the conjugate spectrum (having imaginary part with opposite sign).
 - Calculate the shift (τ_x, τ_y) as a position of the maximum of the cross-correlation: $(\tau_x, \tau_y) = \text{argmax } c(x, y)$. If the value of the maximum is repeated, then we choose the position that is closest to the center, i.e. $\tau_x^2 + \tau_y^2 \rightarrow \min$.
2. Perform the inverse filtering to obtain the restored image \hat{f}_i of the current frame: $\hat{f}_i(x, y) = \mathcal{F}^{-1}(\mathcal{F}(f_i) \cdot H_i)$, $(x, y) \in (X \times Y)$, where:

$$H_i = H_i(\vec{u}) = T \frac{\sin(\pi \vec{u} \circ \vec{\tau}_i)}{\pi \vec{u} \circ \vec{\tau}_i} \exp(-j\pi \vec{u} \circ \vec{\tau}_i), \quad \vec{u} \circ \vec{\tau}_i = u_x \tau_x + u_y \tau_y, \quad \vec{u} \in (X \times Y).$$

Normalize \hat{f}_i by average (or maximum) value of the intensity (to reach $T \equiv 1$):

$$\hat{f}_i(x, y) = f_i(x, y) \frac{\sum\{f_i\}}{\sum\{\hat{f}_i\}}, \quad (x, y) \in (X \times Y), \quad \sum\{f\} = \sum_{x=1}^X \sum_{y=1}^Y f(x, y).$$

3. Estimate the effect of the proposed filtering of motion noise through the image triads: the current input f_i , the restored one \hat{f}_i and the previous f_{i-1} :
 - Preliminary translate the restored image \hat{f}_i to the position of the current f_i , i.e. by the vector $\vec{\tau}/2$. This translation can be thought already done at the previous step 2 in the frequency domain via the formula $\mathcal{F}(\text{tran}\{f, \vec{\tau}/2\}) = \mathcal{F}(f) \exp(-j\pi\vec{u} \circ \vec{\tau}/2)$, or directly in the object domain through: $\text{tran}\{f(x, y), \vec{\tau}/2\} = f\left(x + \frac{\tau_x}{2}, y + \frac{\tau_y}{2}\right)$. In the second case, we have to care about the content of emptied (opposite to the translation) horizontal and vertical end-stripes in the translated image.
 - Calculate PSNR (Peak Signal-to-Noise Ratio) for the pair (f_i, \hat{f}_i) : $\text{psnr}_1(i) = \text{psnr}(\hat{f}_i, f_i)$, where $\text{psnr}(g, f)$ is defined like in [4]:

$$\text{psnr}(g, f) = 20 \cdot \log_{10}(255) - 10 \cdot \log_{10} \left(\frac{1}{X \cdot Y} \sum_{x=1}^X \sum_{y=1}^Y (g(x, y) - f(x, y))^2 \right) [\text{dB}].$$

- Translate the previous image f_{i-1} to the position of current f_i , i.e. by the vector $\vec{\tau}$. We denote this translation by \tilde{f}_{i-1} . Next, calculate PSNR for the pair $(\tilde{f}_{i-1}, \hat{f}_i)$: $\text{psnr}_0(i) = \text{psnr}(\tilde{f}_{i-1}, \hat{f}_i)$.
- Calculate the assessment $q(i) = \text{psnr}_1(i) - \text{psnr}_0(i)$.
- End.

The behavior of the quantity $q(i)$ for different pairs of consecutive frames (f_{i-1}, f_i) allows to estimate the effect of the proposed filtering of the noise from camera movement. We expect that $q(i) > 0$ (the more, the better :).

Complete algorithm for the entire sequence or a part of it: $\{f_i\}_{i=1}^m$.

1. Choose a reference point (x_0, y_0) and allocate an image-accumulator \hat{f}_0 , $\hat{f}_0 \equiv 0$.
2. For each pair of frames (f_{i-1}, f_i) , $i = 2, \dots, m$, apply the basic algorithm to obtain the reconstruction \hat{f}_i , which appeared translated into the position of f_i .
3. Translate \hat{f}_i at the reference point (x_0, y_0) and accumulate it in \hat{f}_0 , i.e. currently $\hat{f}_0 = \hat{f}_0 + \hat{f}_i$.
4. If $i < m$, return back to Step 2.

5. Normalize \hat{f}_0 according to intensity, $\hat{f}_0 = \hat{f}_0/(m - 1)$.
6. The complete reconstructed (restored) image is $\hat{f}_0 = \hat{f}_0(x, y)$, $(x, y) \in (X \times Y) \subset \mathbb{R}^2$. End.

3.5. A Comparative analysis

For convenience, we denote the result of the complete algorithm by (**Mine**). For the purposes of comparative analysis, we will define also two simple reconstructions, namely:

(**sAvg**): A direct summation of all frames of the sequence, normalized in number. Next, the result $f_{(sAvg)} = \frac{1}{n} \sum_{i=1}^n f_i$ is translated at the reference point (x_0, y_0) :

$$f_{(sAvg)}(x, y) = f_{(sAvg)}(x - x_0, y - y_0);$$

(**tAvg**): Like (**sAvg**), but before summing up the current frame f_i , $i \in \{2, \dots, n\}$ it is translated at the point (x_0, y_0) . Finally, the result is normalized by $(n - 1)$:

$$f_{(tAvg)}(x, y) = \frac{1}{n - 1} \sum_{i=2}^n f_i(x - x_0, y - y_0).$$

Obviously, we can arrange the three reconstructions in “goodness \prec ”, as follows:

$$(\text{sAvg}) \prec (\text{tAvg}) \prec (\text{Mine}),$$

which can be verified experimentally using the following estimates¹: mean(Img), max(Img), psnr(Img0, Img1), psnr(|Img0 - Img1|, Img1), performed on the three reconstructions, see Table 1 and Table 2.

3.6. Numerical Experiments

Experiments were conducted in MatLab using the generated initial sequence² “multiframe\mdp_8Lux”, consisting of 8 BMP images representing 8 successive frames (snapshots) of a test scene. The result of the evaluation of the shifts between frames, see Fig. 8b and Fig. 10f, shows that the movements of the camera from frame to frame are rather artificial ones than a result of random shake of the hand carrying the camera. Nevertheless, we believe that the main effect of uniform accumulation of the scene in the pixel matrix of the camera is available

¹All these functions are borrowed from MatLab.

²All the necessary photo materials concerning the studied problem have been kindly placed in our disposal by MMSolution Inc., Sofia.

Table 1. Quality rating (in several parameters) for the sequence “mdp-8Lux”: first part – for each pair $(i \leftrightarrow i) = (f_{i-1} \rightarrow f_i)$, $i = 2, \dots, 8$; and second part – for several sub-sequences in lengths: 2, 3, 5 and 8.

\measures\sub- ries\re	Mean	Max	Q	PSNR	PSNR								
	tAvg -Mine	sAvg -Mine	tAvg -Mine	sAvg -Mine	sAvg,Mine sAvg,tAvg								
2 \leftrightarrow 2	7.06	6.24	9.33	65.0	100.0	2.44	29.00	30.02	26.56	14.06	13.89	14.94	
3 \leftrightarrow 3	12.43	9.08	17.12	123.0	167.0	2.93	24.29	26.42	21.36	14.34	13.45	16.93	
4 \leftrightarrow 4	9.03	11.61	17.12	76.0	182.0	184.0	5.64	26.92	24.37	21.28	13.55	13.81	15.54
5 \leftrightarrow 5	6.88	10.37	11.49	62.0	162.0	165.0	4.76	29.25	25.25	24.49	14.04	14.54	15.21
6 \leftrightarrow 6	0.66	13.84	13.95	8.0	163.0	163.0	25.46	48.52	23.12	23.06	14.56	16.96	16.94
7 \leftrightarrow 7	6.63	12.87	16.68	57.0	172.0	173.0	8.03	29.56	23.60	21.52	14.74	15.72	16.80
8 \leftrightarrow 8	5.36	15.01	16.41	54.0	173.0	172.0	9.78	31.40	22.35	21.62	14.87	16.54	16.58
2 \leftrightarrow 2													
2 \leftrightarrow 3	6.90	7.93	12.17	73.0	113.0	159.0	4.94	29.23	27.82	24.29	13.02	13.09	14.59
2 \leftrightarrow 5	5.89	9.88	14.29	47.0	172.0	192.0	7.64	30.70	25.99	23.06	12.96	13.46	15.17
2 \leftrightarrow 8	3.52	10.37	13.06	26.0	133.0	135.0	11.12	35.09	25.83	23.98	13.39	14.49	15.60

Table 2. Quality rating (in several parameters) for the sequence “mdp-8Lux-(cut-off)”, i.e. clipped frames: first part – for each pair $(i \leftrightarrow i) = (f_{i-1} \rightarrow f_i)$, $i = 2, \dots, 8$; and second part – for several sub-sequences in lengths: 2, 3, 5 and 8.

\measures\sub- ries\re	Mean	Max	Q	PSNR	PSNR								
	tAvg -Mine	sAvg -Mine	tAvg -Mine	sAvg -Mine	sAvg,Mine sAvg,tAvg								
2 \leftrightarrow 2	7.40	6.36	10.38	57.0	97.0	121.0	3.03	28.64	29.81	25.62	12.44	12.26	13.40
3 \leftrightarrow 3	15.01	10.05	18.92	130.0	167.0	176.0	2.43	22.91	25.19	20.48	12.53	11.59	14.81
4 \leftrightarrow 4	8.46	15.06	19.57	68.0	183.0	184.0	7.47	27.50	21.97	20.03	11.98	12.64	13.64
5 \leftrightarrow 5	6.48	13.91	14.75	82.0	162.0	165.0	7.40	29.73	22.72	22.33	12.62	13.47	13.84
6 \leftrightarrow 6	0.00	16.33	16.33	0.0	163.0	163.0	Inf	Inf	21.67	21.67	12.74	14.88	14.88
7 \leftrightarrow 7	6.91	15.96	19.55	51.0	170.0	176.0	9.08	29.24	21.69	20.16	13.12	14.13	15.03
8 \leftrightarrow 8	5.39	17.67	19.55	46.0	172.0	171.0	11.20	31.32	20.89	20.12	13.05	14.47	14.81
2 \leftrightarrow 2													
2 \leftrightarrow 3	8.38	9.80	15.97	78.0	105.0	128.0	5.57	27.75	25.99	22.18	11.77	11.83	13.76
2 \leftrightarrow 5	6.36	13.95	18.91	47.0	140.0	147.0	9.44	30.24	23.05	20.80	11.66	12.46	14.06
2 \leftrightarrow 8	4.05	13.75	16.99	26.0	85.0	95.0	12.17	34.13	23.58	21.96	12.04	13.24	14.35

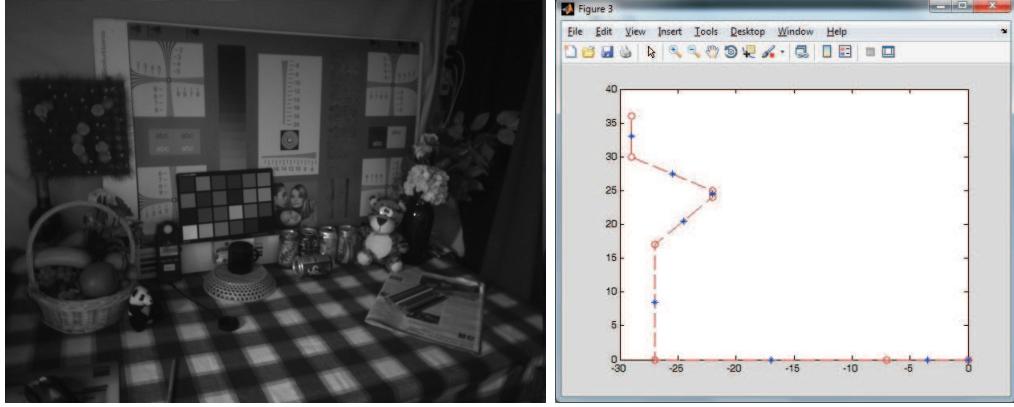


Figure 8. Illustration of the proposed method for reconstruction from motion.

Left (a): the result of the method applied over a series of 8 images; **Right (b):** the hodograph of the respective frame centers. The center of the 1st image is at $(0,0)$. The asterisks mark the reconstructions on the respective pairs of sequential images. All such reconstructions are moved at $(0,0)$ to contribute to the final accumulated result, i.e. the left image, see also Table 1

in this test sequence (mainly because the results of motion noise suppression are positive).

The original sequence “mdp_8Lux” has a frame size of $[4160 \times 3120]$ pixels. This relatively high dimension make difficulties experimenting more precise methods and approaches for noise removal, presented in Section 2 of the report. For the purposes of the common analysis of the results, we use also a reduced version of “mdp_8Lux”, namely “mdp_8Lux_ (cut_off)”, whose frame size is $[1285 \times 965]$.

The result of the complete reconstruction (**Mine**), on the all 8 frames of “mdp_8Lux” is illustrated in Fig. 8a, where the reference point is the center of the first frame f_1 . The intermediate results of the evaluation of the shifts $\vec{\tau} = (\tau_x, \tau_y)$ between the pairs of consecutive frames (f_{i-1}, f_i) , $i = 2, \dots, 8$ are given in Fig. 8b, where the positions of the frames are marked by “circles”, and those of the basic reconstructions in between – by “stars”.

On Fig. 9 some simple reconstructions are shown, namely, (**sAvg**), (**tAvg**) as well as the absolute value of their differences to (**Mine**) (depicted in Fig. 8a).

Visually, the resulting images and comparisons between them is better to be looked by a professional Image-Editor|Viewer. The video improvement tools of MSWord just gloss over these details, that we want to emphasize, i.e. the actually better results look worse, that's why, please, follow simultaneously the attached



Figure 9. Two more possible reconstructions: **Left:** Simple summation of all images of the original series, with the final result translated to (0,0) (top, (a)); each image is translated to (0,0) before summation (bottom, (b)). **Right (c,d):** the absolute differences between the left images and the main reconstruction (see Fig. 8). For better visibility the difference pictures are amplified to maximal intensity. The mean of case (a) is about 3 times higher than the mean of case (b), see also Table 2

two tables, Table 1 and Table 2.

In (**sAvg**), see Fig. 9a and Fig. 9c, obviously much of the noise generated by motion is retained (e.g. the clearly visible contour areas). In (**tAvg**), see Fig. 9b and Fig. 9d, contours are barely visible, i.e. the noise of pure movement is well suppressed, but the noise accumulated during the movement is still kept.

Fig. 10 illustrates 5 partial reconstructions, i.e. full reconstruction, but based on some sub-sequences of the sequence “mdp_8Lux_(cut_off)”. Namely, the reconstructions based on the first 2, 3, 5 and 8 frames are presented. As a first reconstruction we can (quite formally) define the first frame f_1 .

As far as all operations on the images are made in the Fourier discrete frequency domain, in view of the periodicity, the shifts between frames can be traced in the respective reconstructions as relevant horizontal and/or vertical border strips (clearly visible, for example, in Fig. 10d and Fig. 10e). What concerns the hodograph of the shifts in Fig. 10f, we see that it differs slightly from that in Fig. 8b, which can be explained by our disregard for the “occasional” rotations of the camera as it moves from frame to frame. Undoubtedly, the ignoring of the “occasional” rotations in our model of motion noise certainly affects the quality of the expected result, based on modelling “occasional” translations only. And we left this for future extensions of the model.

Note that the processing time for a pair of frames from the sequence “mdp_8Lux” is about 40s, and from “mdp_8Lux_(cut_off)” it is about 4s.

3.7. Possibilities for improving the algorithm

- confirm theoretically (or to reject in some cases) the heuristics (7). Additionally, the adopted inverse approach to be extended with Wiener-filtering and/or other improvements, [3];
- In addition to shifts, in the above described model of motion noise one may include more effects as: rotation, scaling and rolling-shutter (the latter is specific for CMOS cameras);
- A few smoothing the “path” of the frames can help to refine the filtering (on this smooth trajectory);
- The proposed algorithm can be reconciled with any algorithm for video stabilization. Something more, the proposed approach can be applied to suppress the eventual movement noise in a video clip, wherein the “filter” can be considered as a “swimming” one over the frames trajectory of the video clip. The swimming-filter definition domain, measured in number of consecutive frames, will be a suitable constant n or variable $n(t)$ ($n \ll N$, where N is the length of the video clip).

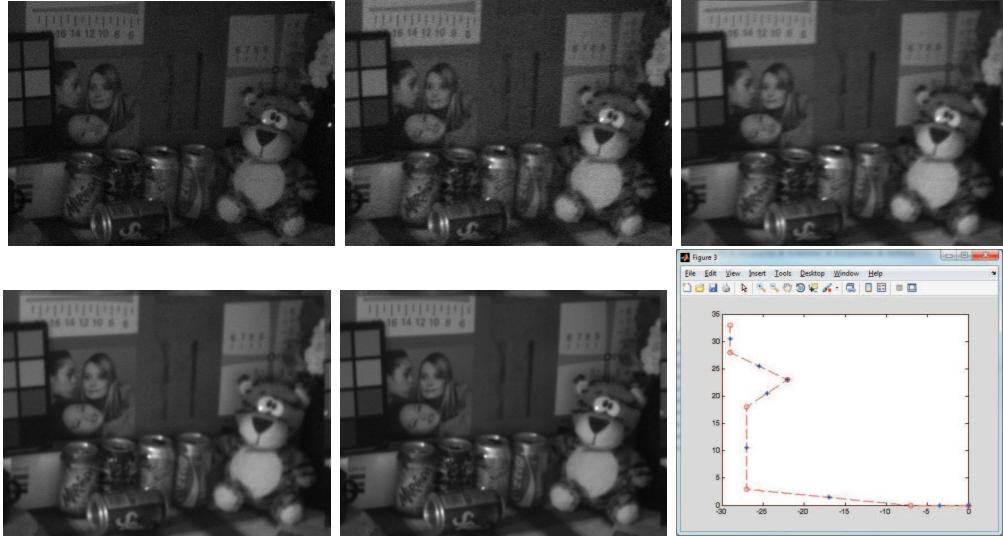


Figure 10. A series of reconstructions by the proposed method, on a series of 8 smaller frames (cropped from the originals, see Fig 8): **Top:** on the first frame (left, (a)); on the 1st and 2nd frames (middle, (b)); on the 1st – 3rd frames (right, (c)); **Bottom:** on the 1st – 5th frames (left, (d)); on the all 8 frames (middle, (e)); and the hodograph of the image centers of the initial series (right, (f)). Each partial reconstruction is moved to (0,0) to contribute to the current accumulated reconstruction

References

- [1] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
- [2] D. Dimov and A. Nikolov. Real Time Video Stabilization for Handheld Devices. *ACM International Conference Proceeding Series* 833, 124–133, ACM Digital Library, 2014.
ISBN:978-1-4503-2753-4, DOI:10.1145/2659532.2659631.
- [3] R. C. Gonzalez and R. E. Woods. Digital Image Processing. *3-rd Edition*, Pearson Edu. Int., 2008.
- [4] https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio.

Two Dimensional Optimal Cutting Problem

Ana Avdzhieva, Todor Balabanov, Georgi Evtimov,
Ivan Jordanov, Nikolai Kitanov, Nadia Zlateva

Abstract

At the 120-th European Study Group with Industry engineering company “STOBET” Ltd set the problem of optimizing the process of arranging plates with different shapes on a steel sheet. This report describes the methods proposed by the study group, and the conclusions drawn from the study of these approaches.

Key words: Two dimensional cutting stock problem, genetic algorithm, greedy algorithm

1. Introduction

Engineering company “STOBET” Ltd creates projects for buildings and structures.

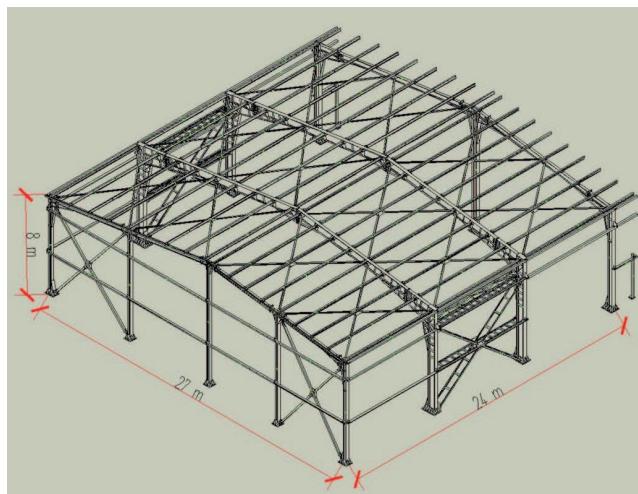


Figure 1. Steel construction

Having a lot of such projects, the company wants to improve the quality of the product and to reduce the price of the process of its realization. “STOBET” Ltd uses software for design of steel constructions. The software produces drawings

of the different needed plates. After making all drawings, the task is to deploy optimally all plates on a minimal number of standard steel sheets.

1.1. Description of the problem

A standard steel sheet with dimensions $X = 1500$ mm, $Y = 12000$ mm is available. On each such sheet we have to locate maximal number of plates of different shapes and dimensions. The contour of a plate might be square, rectangle, trapezoid or any non-crossing closed polygon, consisting of line segments only.

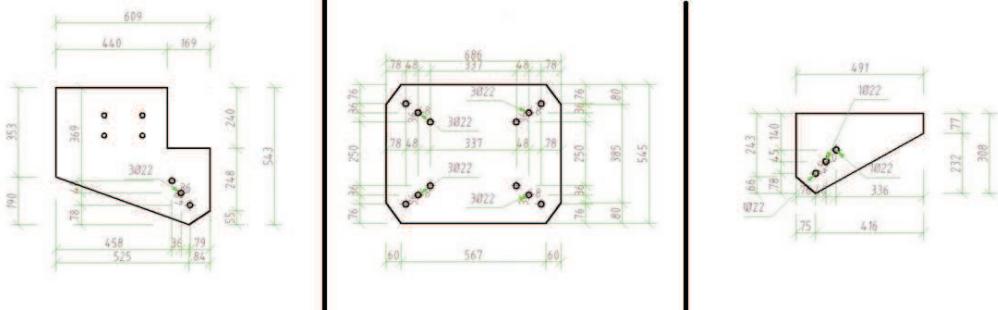


Figure 2. Examples of different plate's shapes

For any plate only the coordinates of its vertices are known. Let, for example, are given n plates with their coordinates:

- **Plate 1:** $(x_{11}, y_{11}; x_{12}, y_{12}; \dots; x_{1i_1}, y_{1i_1})$
- **Plate 2:** $(x_{21}, y_{21}; x_{22}, y_{22}; \dots; x_{2i_2}, y_{2i_2})$
- ...
- **Plate n :** $(x_{n1}, y_{n1}; x_{n2}, y_{n2}; \dots; x_{ni_n}, y_{ni_n})$

The numbers k_1, k_2, \dots, k_n of the pieces of plates $1, 2, \dots, n$ needed for the construction are given as well.

In the process of optimization some plates can be rotated for best fit. The aim is to locate the small plates on the given big steel sheet with minimal waste. For cutting a plate a cut of 5 mm has to be provided.

2. The approaches

Three different approaches to the two dimensional cutting problem in hand are considered by the group.

2.1. A Genetic algorithm

For each polygonal plate a rectangle (box) covering it is considered. The box is easily determined by the coordinates of its lower left vertex (x_{\min}, y_{\min}) and the coordinates of its upper right vertex (x_{\max}, y_{\max}), where x_{\min} is the minimal x -coordinate of the plate, etc. The waste from every box is calculated as the difference between the box area and the polygon area. Then we use the genetic algorithm developed in [1] to deploy boxes on the steel sheet.

Genetic algorithm (GA) is a robust adaptive optimization method based on biological principles. A population of strings representing possible problem solutions is maintained. Search proceeds by recombining strings in the population. The theoretical foundations of genetic algorithms are based on the notion that selective reproduction and recombination of binary strings changes the sampling rate of hyperplanes in the search space so as to reflect the average fitness of strings that reside in any particular hyperplane. Thus, genetic algorithms need not search along the contours of the function being optimized and tend not to become trapped in local minima [4]. Some modifications of the standard form of the GA are accomplished and described in details in Final Reports of ESGI'113 [1].

The total waste is the sum of waste for each box and the waste obtained after application of the genetic algorithm.

2.2. Using Matching plates

The second approach is to arrange first several identical plates in series and after that put the series in boxes. Thus, we optimize the wastage coming from boxes. From the other hand, the resulting boxes might become big and the task to find enough space for putting the big box on the steel sheet becomes more difficult.

2.3. A Greedy algorithm

For each plate we calculate the area of the polygon and we sort plates in decreasing order by their area. The idea is to put the plates on the steel sheet in such a way that its width to be filled first.

We start by considering the contour of the steel sheet as negatively oriented (in counterclockwise direction). As a result we have a sequence of four points describing the negatively oriented contour of the sheet. We orient the contours of all plates positively (in clockwise direction). Calculating the oriented surface of each plate we obtain the orientation of the contour described by the sequence of

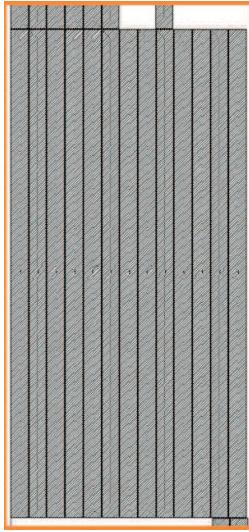


Figure 3. Example of a solution obtained by applying the modified genetic algorithm

vertexes of the plate. If necessary, we easily reverse the orientation by rearranging the vertexes in opposite way. Finally, we have a sequence of points, describing the contour of the plate that is positively oriented.

We start by adjusting the plate with the biggest area into the lower left corner of the steel sheet. The plate can be rotated for best fit. We "cut" the plate. The result is a sequence of points describing the rest of the sheet available for cutting which contour is negatively oriented. On the lower left corner of the rest we put the first plate of the ordered list that fits. If no such plate exists this corner is eliminated for further considerations, and we continue with the following lower left corner of the sheet. The sheet is completely flared when it has no corner in which a plate from the list of non-cut plates could be adjusted.

3. Results

Some experimental tests with real data were made using the suggested modified genetic algorithm. One of them is shown on Fig. 3. The total waste was decreased to 19%.

For a concrete example a genetic algorithm was applied to two-dimensional topology of polygons. The result was worse. The total wastage of the steel sheet was 21.30%.

We have some difficulties to implement the greedy algorithm.

Acknowledgements. This work was supported by the Sofia University Research Fund through Contract No. 30/2016.

References

- [1] Ana Avdzhieva, Todor Balabanov, Detelina Kirova, Hristo Kostadinov, Tsvetomir Tsachev, Stela Zhelezova, Nadia Zlateva, Optimal Cutting Problem, 113-th European Study Group with Industry, September 14–18, 2015, Sofia, Bulgaria, Problems and Final Reports, Fastumprint, 2015, 49–61.
- [2] IICT-BAS, ESGI113, problem 3, a genetic algorithm solver,
[https://github.com/TodorBalabanov/
ESGI113Problem3GeneticAlgorithmSolver](https://github.com/TodorBalabanov/ESGI113Problem3GeneticAlgorithmSolver).
- [3] D. Goldfarb, M. Todd. Chapter II. Linear programming, In: Handbooks in OR & MS, vol. 1 (Eds. G. L. Nemhauser et al.), Elsevier, 1989.
- [4] D. Whitley, T. Starkweather, C. Bogart, Genetic algorithms and neural networks: optimizing connections and connectivity, *Parallel Computing* Volume 14, Issue 3 (1990) 347–361.

Authenticity Management Algorithm for Digital Images

Todor Balabanov, Peter Dojnow, Vasil Kolev, Nikolai Manev, Walter Mudzimbabwe, Petar Tomov, Ilian Zankinski, Stela Zhelezova

Abstract

Nowadays network services are gaining great attention. Therefore authentication of media content is very important. For the considered problem, the group propose an algorithm with client-server architecture. The chosen digital signature algorithm is based on image content and is according to NIST specifications. The most important parameters for an effective tampering detection are discussed.

Key words: authentication, watermarking, tampering detection

1. Introduction

Image authentication is of great importance due to the large number of multimedia applications in various fields. Currently, the amount of digital images transmitted over non-secure channels is growing rapidly. Therefore, the protection of image integrity is of great interest.

A digital watermark is called robust if it resists a designated class of transformations and it is imperceptible if the watermarked content is perceptually equivalent to the original one.

There are active and passive image authentication methods. The active methods extract some information of the image to be authenticated while the passive ones perform the authentication without needing previous information about the authenticated image. Active methods include watermarking and hashing based methods. We extract the needed authentication and tampering detection information from the original picture so we use active methods for the proposed algorithm.

With respect to the embedding method there are different watermarking techniques. Some of them embed message in spatial domain. Other methods use message embedding in a frequency-based representation of a digital image. The proposed authentication schema uses spatial domain.

Because of the particular case in which the watermarking is considered, we employ steganographic technique to embed data. Least Significant Bit (LSB) hiding is one of today's easiest techniques for image steganography. It imply

adding some secret information in the least significant bits of the image pixel. The image quality is distorted for the number of bits embedded in a pixel greater than 3. However, such a technique is very insecure because the watermark can be easily destroyed. But for our problem, the most desirable property is imperceptibility to human senses than robustness therefore it is appropriate. The creation of robust and in the same time imperceptible watermarks has proven to be quite challenging [3].

In the case of mobile devices the signature scheme must be efficient enough without delays. We choose to develop a simple image authentication scheme.

1.1. Definition of the problem

There is an image taken in an Android application. The application have to sign digitally the image without the user's knowledge in order to verify its origin. Apart from this, a digital watermark has to be added such that in case of tampering the modified part of the image is indicated. The application sends the image to the server where a corresponding algorithm part verifies the origin and the integrity of the image.

1.2. Historical Review

A Vector Quantization (VQ) based digital image watermarking scheme is proposed in [4]. The codewords in the VQ codebook are classified into different groups according to different characteristics and then each binary watermark bit is embedded into the selected VQ encoded block. This technique cannot resist geometrical distortion.

There are algorithms based on the discrete Radon transform [5]. The Global Hash estimation of the image is used to establish if the image under analysis was tampered and if this is the case, the local Hash estimation is used for obtaining the exact tampered regions. This algorithm is not suitable for our problem because of the requirement of sending the original Hash values apart from the watermarked image.

Some steganographics techniques also are suitable for hiding an information in image. A steganographic scheme is applied if the user has a secret message that is to be hidden in the image. The authors [7] use a hash function to find out the locations to store the secret message which is encrypted using AES encryption. In the problem that we consider, the secret message has to be the private key of the application. This scheme cannot work if the picture is tampered.

1.3. The Algorithm Requirements

The performance of the proposed algorithm have to meet the following main properties:

- Robustness: It may not be possible without knowledge of the procedure and the secret key to manipulate the watermark. Robustness also means the resistance ability of the watermark information to changes and modifications made to the original file.
- Nonperceptibility: It is important to recognize whether the brought bit sample of the watermark produces perceptible changes optically. A perfect nonperceptible bit sample is present if data material marked with watermark and the original cannot be distinguished from each other.
- Blind: The detection of digital watermark has to be done without the original data, so in order to detect watermark information, blind techniques have to be used.
- Effectiveness: Tampering detection and localization of the changes made to a watermarked image.

2. Our Approach

2.1. Initial statements

The entire algorithm consists of two main steps:

- client part in which the embedding is done;
- server part where the verification and extracting is done.

The initial restrictions are:

- Size of the original image $n \times m$ pixels in Bitmap. Minimal size: 1024×768 . This is the worst case, because the hidden bits place is proportional to the image size.
- Content: interior and car pictures. The content is important in the connection with the hiding bits methods. If the image content is almost the same, i.e. the colors are evenly distributed, hiding of information is much more difficult.
- The value of perceptual transparency is measured with Signal-to-noise ratio (SNR). SNR is used in science and engineering to compare the level of a desired signal to the level of background noise. It is defined as the ratio of signal power to the noise power:

$$SNR = 10 \cdot \log(MAX^2/MSE),$$

MAX – the expected pixel values,

MSE – the standard deviation of the pixel values between the original image (OI) and the watermarked image (WI) bits.

$$MSE = \frac{1}{m \cdot n} \sum_i \sum_j (O_{i,j} - W_{i,j})^2,$$

where $i \in \{1, \dots, n\}$; $j \in \{1, \dots, m\}$ are the dimensions of the original picture.

For the considered problem it is required $SNR > 30$.

The most common way to model color images in Computer Graphics is the RGB color model – each pixel is represented by three values, the amount of red, green and blue. We can use a packed ordering – the three color components are placed together in a single array element. Each color pixel is presented by a 32-bit value according to Fig. 1. Eight bits are used to represent each of the RGB components and 8 bits are reserved for the transparency (α) component. The main input element in our algorithm is the original image (OI). It is an $n \times m$ matrix C , $c_i = (R_i, G_i, B_i)$. Each element of the matrix is $0 \leq R_i, G_i, B_i \leq C_{\max}$, for the most digital images $C_{\max} = 255$. Because there are three color channels, it's possible to store three hidden bits of information in each pixel. But our proposition is to use only one of the color channels because the application does not need more capacity.

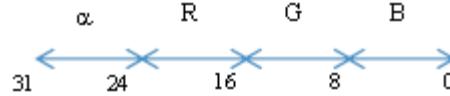


Figure 1. The bits of a color pixel

In order to be able to create a digital signature, we need a private key. For the considered problem, the user must not be involved in the signing procedure. The algorithm aim is to verify that the image is taken with certain device. Its corresponding public key is needed in order to verify the authenticity of the signature. The key pair (private and corresponding public key) generation is done in two parts. The choice of algorithm parameters and the key pair generation

for particular image and user. The algorithm parameters are: q – N -bit prime, p – L -bit prime, $p - 1$ is a multiple of q , g – its multiplicative order modulo p is q . The NIST recommend in Federal Information Processing Standard (FIPS) 186-3 key length pairs (2048, 256), and (3072, 256) for (L, N) – the associated pair of length parameters for a DSA. The algorithm parameters can be used as they are implemented in library according to some provider (as SUN) for instance. Also the `DSAPrivateKeySpec` class can be used to chose specific (p,q,g) parameters.

A particular key pair is generated by `KeyPairGenerator` class. At this step the `String algorithm` (we choose DSA) have to be specified. To initialize two arguments are needed: the key length L_k and the source of randomness. For the second parameter the `threadedSeedGenerator` have to be used several times with different parameters, each of them connected with the particular user, device and time parameters.

So the next input element the algorithm needed is the private key k with length $L_k = 2048$ bits, k – random, $0 < k < q$ and the public key $y = g^k \bmod p$.

2.2. The client part

Run-up the image

From considered `OI` matrix C it is needed to obtain a matrix `ZBI` with zeroes in each bit which we will use later for hiding information. We chose LSB and 3-LSB hiding strategy to achieve the highest possible robustness without degrading image quality.

Next from `ZBI` the sequence is obtained. The order in which pixels are taken is important due to the avalanche effect in hash functions. This can be done in connection with private key. On the server there can be stored different `PATH` schemes for each key pair. A scheme is actually a permutation of $n.m$ variables. All permutations are considered in lexicographical order and each has its consecutive number therefore in variable `PATH` the corresponding permutation number is saved. So the function `MakeSequence` takes as input `ZBI` and `PATH` and returns the “zerobits” image sequence ZBI_s .

Signing

Message digests are secure one-way hash functions that take arbitrary-sized data and output a fixed-length hash value. The standard Message Digest (MD) algorithms are for example MD5, SHA-1, SHA-256 etc. Bear in mind that the first two are compromised [3], [8] for the implementation of the proposed algorithm we suggest SHA3-256, SHA3 is the newest one but it has BITE-only implementation in Java so the length of input sequence has to be divisible by 8. It takes as input `ZBIs` and gives as output $|\text{MD}| = 256$ bits. A common way to sign things is the

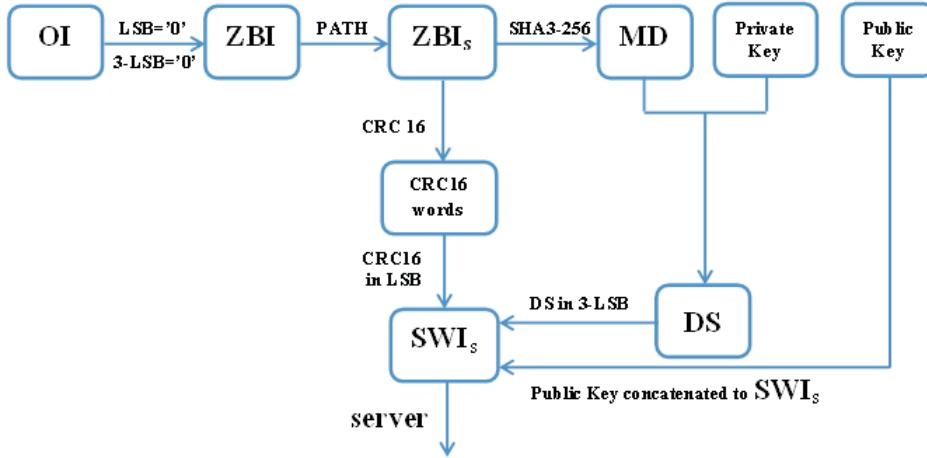


Figure 2. The algorithm client part

Digital Signature Algorithm. The signing is done in `Signature` object from the `Signature` class with the following steps:

- Get a `Signature` object – the signature algorithm name is specified and the name of the message digest algorithm used by the signature algorithm is specified. `SHA3-256withDSA` is a way of specifying the DSA signature algorithm, using the SHA3-256 message digest algorithm (Spongy Castle Crypto package).
- Initialize the `Signature` object – the private key is used.
- Supply the `Signature` object the `ZBIs` to be signed – the *update* method of the `Signature` object is called.
- Generate the digital signature sequence $DS = (r \parallel s)$ applying *sign* method, where $r = (g^{k_i} \bmod p) \bmod q$ and $s = k_i^{-1}(\text{SHA3-256}(ZBI_s) + k.r) \bmod q$, k_i – a random per-image value and L_{DS} is about 2×160 bits but it depends on hash function and converting of integers r and s into strings.

Now the DS and public key are ready to be sent to the server. Usually they are stored in additional files and sent separately. The proposed algorithm embed DS in the `ZBIs`. We store DS not in each 3-LSB but in each $(n.m) \bmod L_{DS}$ 3-LSB according to PATH permutation. The public key is added at the end of `SWIs`.

Watermarking for tampering detection

For the algorithm purpose, the non visible watermark have to be hidden eventually in the ZBI_s . We use wide spread blocking technique. The algorithm apply CRC16 on each different block. The `Crc16` class of Java has to be used. It gives as output 16 bits codeword for each input string. There are different standard CRC polynomial on these parameters included in the library but for each string length we can choose the best one with respect to value of undetected error probability [1]. We have to hide 16 bits in LSB of one color channel so the input string length IL_{CRC16} has to be divisible by 32 and at least $32 \cdot 16 = 512$ bits, i.e. 16 pixels. In this case, each LSB of the chosen color is used. The greater IL_{CRC16} the rougher the tamper detection.

As a result, ZBI_s becomes SWI_s – the image bits sequence with encoded public key (X.509 standard) concatenated at its end, DS bits as 3-LSB and $CRC16$ output bits as LSB. Only the signed and watermarked image sequence SWI_s is sent to the server part of the proposed algorithm.

2.3. The server part

Run-up

The encoded public key is extracted from the end of the obtained SWI_s . Next the DSV from the used 3-LSB and the $CRC16V$ words from LSB are extracted, thus from SWI_s the ZBI_s remains. It is needed because exactly ZBI_s is used to produce signature in the client part of the algorithm.

Verifying a Digital Signature

A `KeyFactory` object in `KeyFactory` class have to be used in order to get DSA public key from its encoding. A signature is verified by an instance of the `Signature` class. For the `Signature` object the same algorithm (DSA) and the same message digest algorithm (SHA3-256) are specified. But now the initialization is with the public key. Then the `update` method of `Signature` object supplies ZBI_s . At the end, `verify` method is applied on the extracted digital signature DSV and the calculated one in the $(r||s)$ form. This method returns “true” only if DSV is the actual signature of the supplied data ZBI_s generated by the private key k_i corresponding to the read public key. It calculates $\omega = s^{-1} \bmod q$, $u_1 = SHA3-256(ZBI_s).\omega \bmod q$, $u_2 = r.\omega \bmod q$ and $v = (g^{u_1}y^{u_2} \bmod p) \bmod q$. So the signature is valid for $v = r$. If the returned value is “true” we can go on with CRC checking else it is clear the image is not taken with the given application.

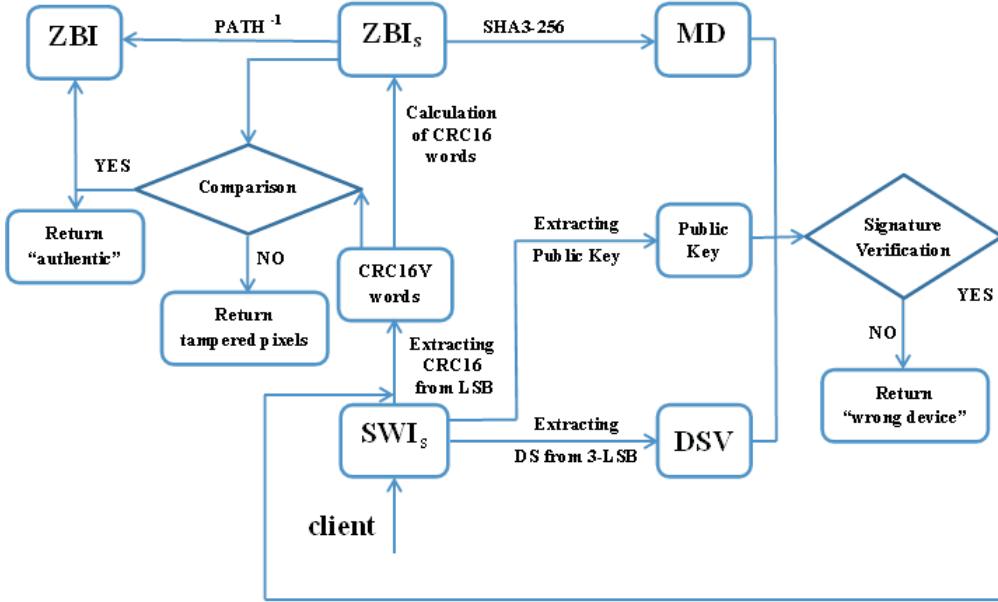


Figure 3. The algorithm server part

Tampering detection

At this step the algorithm compares the extracted $CRC16V$ values and $CRC16$ words obtained from the received ZBI_s . If the image is not tampered it returns “true”. Then applying on ZBI_s the inverse permutation $PATH^{-1}$, ZBI is obtained. It is not the original image, but perceptually indistinguishable.

In the other case, the pixels with wrong $CRC16$ values have to be considered as tampered and a new two color image (the same size as OI) can be done – with black color only at tampered pixels.

Conclusions

We propose a signing and watermarking schema suitable for the features of a particular business problem. The parameters and tools are chosen according to the newest scientific achievements. The experiments with real data are needed in order to evaluate the lowest possible SNR for different parameters and to ensure the best performance of the algorithm. The real media environment have to be considered to supply the algorithm with error protection during the transmission.

References

- [1] Baicheva, T., Dodunekov, S., Kazakov, P., Undetected error probability performance of cyclic redundancy-check codes of 16-bit redundancy, IEE Proc Commun., 147 (5) 2000, 253–256.
- [2] CERT Vulnerability Note VU# 836068, Kb.cert.org., December 31, 2008.
- [3] Cox, I. J., Miller, M. L., Bloom, J. A., Fridrich J., Kalker, T.: Digital watermarking and steganography. Morgan Kaufmann, Burlington, MA, USA, 2008.
- [4] Hsien-Chu Wu, Chin-Chen Chang, A novel digital image watermarking scheme based on the vector quantization technique, Computers and Security, 24, (2005) 460–471.
- [5] Liu, Z., Li, Q., Zhang, H., Peng, X., An Image Structure Information based Robust Hash for Tamper Detection and Localization, Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. Darmstadt, Germany, 430–433.
- [6] <http://csrc.nist.gov/publications/PubsFIPS.html>
- [7] Rinu Tresa M. J., Athira M. B., Sobha T., A novel steganographic scheme based on hash function coupled with AES Encryption, Advanced Computing: An International Journal, 5 (1) (2014).
- [8] Schneier, B., Schneier on Security: Cryptanalysis of SHA-1, February 18, 2005.

Laboratory Calibration of MEMS Rate Sensors

Tihomir Ivanov, Galina Lyutskanova, Dragomir Aleksov, Ognian Kounchev

Abstract

In this paper, we suggest a simple algorithm for calibrating microelectromechanical systems (MEMS) accelerometers. We use a classical relation, depending on 12 parameters, between the raw output from the sensors and the calibrated data, as well as a more complicated relation, derived from physical and geometrical considerations. We obtain the calibration parameters by formulating and solving a least-squares problem. Results of numerical experiments are shown to validate the proposed algorithm.

Key words: MEMS accelerometers, sensors, calibration algorithms, parameter identification, reverse problem.

1. Introduction

In directional drilling, the orientation of the borehole is determined by measuring three angles—toolface, inclination, and azimuth angles (see Fig. 1). In the present work, we are interested in the problem of measuring the first two of them.

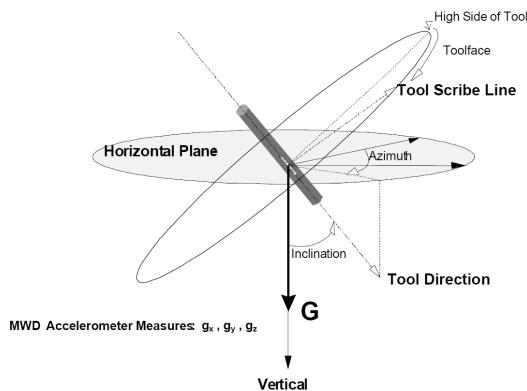


Figure 1. Toolface, inclination, and azimuth angle. [3]

As far as miniature dimensions and full measurement range are concerned, the most advantageous solution is a sensor made of commercial microelectromechanical systems (MEMS) accelerometers, which employs measurements of the three Cartesian components of the vector of gravitational acceleration [4].

Let us denote the acceleration vector, acting on a three-axial accelerometer sensor, with

$$\mathbf{a} = (a_x, a_y, a_z)^T.$$

When the device is in still position, it should be measuring only the gravitational acceleration. Then, using simple geometric considerations, the toolface (tf) and inclination ($incl$) can be easily computed, as follows:

$$(1) \quad tf(\mathbf{a}) = \begin{cases} \frac{360^\circ}{2\pi} \text{atan2}(a_y, a_x) & \text{if } \text{atan2}(a_y, a_x) \geq 0, \\ 360^\circ + \frac{360^\circ}{2\pi} \text{atan2}(a_y, a_x) & \text{otherwise,} \end{cases}$$

where

$$\text{atan2}(x, y) := \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{if } x > 0, \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{if } x < 0 \text{ and } y \geq 0, \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{if } x < 0 \text{ and } y < 0, \\ +\frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0, \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0, \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0. \end{cases}$$

The latter function computes the angle, which the segment through the point (x, y) and the origin makes with the positive x -semi-axis.

Let us remark that when using formula (1) in computer arithmetics, it is wise to substitute the condition for the first case on the right-hand side with $\text{atan2}(a_y, a_x) \geq -\varepsilon$ for some $\varepsilon > 0$, in order to avoid large errors for near-zero angles.

The inclination is given by

$$(2) \quad incl(\mathbf{a}) = \begin{cases} 90^\circ - \frac{360^\circ}{2\pi} \arcsin \sqrt{\frac{a_x^2 + a_y^2}{a_x^2 + a_y^2 + a_z^2}} & \text{if } 90^\circ - \frac{360^\circ}{2\pi} \arccos \frac{a_z}{\sqrt{a_x^2 + a_y^2 + a_z^2}} > 60^\circ, \\ 90^\circ - \frac{360^\circ}{2\pi} \arccos \frac{a_z}{\sqrt{a_x^2 + a_y^2 + a_z^2}} & \text{otherwise.} \end{cases}$$

It is well-known, however, that MEMS accelerometers are subject to different sources of error that should be accounted for, before they can be used in practice. There are two main types of errors—deterministic and random.

The deterministic error sources include the bias (offset) and the scale factor errors [2]. Another issue lies in the fact that formulae (1) and (2) are only valid if the three axes of the sensor are perfectly orthogonal. This, however, can never be the case in practice. The random errors include bias-drifts or scale factor drifts, and the rate at which these errors change with time. Furthermore, all the errors are sensitive to different environmental factors, especially to temperature variations [1].

In the present work, we are interested in the initial calibration in laboratory conditions of the deterministic sources of error. More specifically, we suggest an algorithm for compensating the errors due to nonorthogonalities, shifts and scale factors.

Apart from this, random errors should be accounted for by using an appropriate stochastic model. Many authors suggest using Alan variance as a tool for studying those errors [6]. Also, a temperature model should be used to account for the temperature variations.

2. Calibration approaches

As formulated, e.g., by Aggarwal et al. (2008), calibration is the process of comparing instrument outputs with known reference information and determining the coefficients that force the output to agree with the reference information over a range of output values [2].

Thus, we can divide the problem into two sub-problems:

1. propose a model, relating the raw output data from the sensors to the “real” components of the acceleration vector with respect to some orthonormal coordinate system;
2. determine the parameters in the above model by formulating and solving an appropriate minimization problem.

We shall now consecutively consider those two steps.

2.1. Relation between the measured and calibrated data

We assume the following classical linear relation between the raw data, $\hat{\mathbf{a}} = (\hat{a}_x, \hat{a}_y, \hat{a}_z)^T$, and the calibrated data, $\mathbf{a} = (a_x, a_y, a_z)^T$:

$$(3) \quad \mathbf{a} = \mathbf{M} \cdot \hat{\mathbf{a}} + \mathbf{b}.$$

In the latter, \mathbf{M} is a full 3×3 matrix,

$$\mathbf{M} = \begin{bmatrix} m_{xx} & m_{xy} & m_{xz} \\ m_{yx} & m_{yy} & m_{yz} \\ m_{zx} & m_{zy} & m_{zz} \end{bmatrix},$$

wherein the diagonal elements account for the scale factors and the off-diagonal elements—for the errors due to non-orthogonalities. The vector

$$\mathbf{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$$

contains the offsets.

Another approach is to derive a more accurate form of the matrix \mathbf{M} , using physical and geometrical considerations. It can be shown (see [5]) that

$$(4) \quad \mathbf{M} = \overline{\mathbf{T}}^{orth} \overline{\mathbf{T}} \mathbf{T}, \quad \mathbf{b} = \overline{\mathbf{T}}^{orth} \overline{\mathbf{T}} \mathbf{T} \mathbf{b}$$

where

$$\begin{aligned} \mathbf{T} &= \begin{bmatrix} t_x & 0 & 0 \\ 0 & t_y & 0 \\ 0 & 0 & t_z \end{bmatrix}, \quad \overline{\mathbf{b}} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}, \\ \overline{\mathbf{T}} &= \frac{1}{den} \begin{bmatrix} -\sin^2 \theta & \cos \phi - \cos \theta \cos \psi & \cos \psi - \cos \phi \cos \theta \\ \cos \phi - \cos \theta \cos \psi & -\sin^2 \psi & \cos \theta - \cos \phi \cos \psi \\ \cos \psi - \cos \phi \cos \theta & \cos \theta - \cos \phi \cos \psi & -\sin^2 \phi \end{bmatrix}, \\ den &= -1 + \cos^2 \phi + \cos^2 \psi + \cos^2 \theta - 2 \cos \phi \cos \psi \cos \theta, \\ \overline{\mathbf{T}}^{orth} &= \begin{bmatrix} 1 & \cos \phi & \cos \psi \\ 0 & \sin \phi & \frac{\cos \theta - \cos \phi \cos \psi}{\sin \phi} \\ 0 & 0 & \frac{\sqrt{1 - \cos^2 \phi - \cos^2 \psi - \cos^2 \theta + 2 \cos \phi \cos \psi \cos \theta}}{\sin \phi} \end{bmatrix}. \end{aligned}$$

In the above expressions, the 9 parameters have the following meaning. The three scaling coefficients are denoted by t_x, t_y, t_z , the offsets on the respective axes are denoted by b_x, b_y, b_z and φ, ψ, θ are the angles between the directions defined by the three axes of the MEMS accelerometer. Let us remark that those angles should be close to 90 degrees.

Both approaches have their advantages—in particular, relation (4) gives a more accurate description of the physical problem but it leads to a highly nonlinear minimization problem as can be seen from the next section. The model (3) suggests a more basic description of the relation between the measured and calibrated data but leads to a linear least squares problem with respect to the calibration parameters. We shall present numerical experiments with both models in order to compare their applicability and accuracy.

2.2. Definition of the objective function

Let us define an orthonormal coordinate system $Oxyz$ attached to the device. Let us assume we have obtained raw measurements $\hat{\mathbf{a}}_x, \hat{\mathbf{a}}_y, \hat{\mathbf{a}}_z$ of the device in positions where the gravitational acceleration is aligned with the x -, y -, and z -axes respectively. Further, let $\hat{\mathbf{a}}_i, i = \overline{1, s}$ denote measurements in s different (arbitrary) positions.

We suggest the following objective function that is to be minimized for obtaining the calibration parameters:

$$(5) \quad \begin{aligned} \varepsilon(\mathbf{M}, \mathbf{b}) = & \left(\frac{\|\mathbf{M} \cdot \hat{\mathbf{a}}_x + \mathbf{b} - (g, 0, 0)^T\|}{g} \right)^2 + \left(\frac{\|\mathbf{M} \cdot \hat{\mathbf{a}}_y + \mathbf{b} - (0, g, 0)^T\|}{g} \right)^2 \\ & + \left(\frac{\|\mathbf{M} \cdot \hat{\mathbf{a}}_z + \mathbf{b} - (0, 0, g)^T\|}{g} \right)^2 + \sum_{i=1}^s \left(\frac{\|\mathbf{M} \cdot \hat{\mathbf{a}}_i + \mathbf{b}\| - g}{g} \right)^2. \end{aligned}$$

Minimizing the latter with respect to the calibration parameters, we obtain the best values (in the least-squares sense) such that the computed acceleration is as close as possible to the gravitational acceleration and the coordinate axes are aligned with the natural axes of the device.

3. Numerical results

We are given raw accelerometer outputs obtained in 32 different positions of the sensor, see Table 1. For each position, the toolface and the inclination of the device are known. We use the first 20 of them (the shaded rows in the table) to calibrate the device and the rest is used as a test set. Let us remark that we only give here results from one data set, since other experiments we have carried lead to similar conclusions.

First, we calibrate the sensors, using model (3). For solving the minimization problem (5), we use the Wolfram Mathematica function *NMinimize*. The results are presented in Table 2. In all cases the computed toolface and inclination is within less than 2° from the real value. Let us remark that when the inclination

Table 1. Raw data measured by a three-axial accelerometer

tf	incl	\hat{a}_x	\hat{a}_y	\hat{a}_z
0.00	0.00	-11718.00	966306.00	31738.00
10.00	0.00	-14648.00	938278.40	209862.40
20.00	0.00	-17577.80	891403.80	365428.20
30.00	0.00	-21386.40	815134.20	519431.40
40.00	0.00	-26171.40	710251.80	662986.20
50.00	0.00	-31151.40	583006.00	778122.80
60.00	0.00	-37109.00	440330.40	874997.00
70.00	0.00	-42675.20	277831.00	941500.80
80.00	0.00	-47851.00	112108.80	983199.80
90.00	0.00	-53124.40	58593.00	993258.60
50.00	30.00	271343.57	488907.43	780340.71
50.00	60.00	494627.00	249413.00	780759.00
50.00	90.00	566795.20	-59472.20	779294.00
50.00	120.00	474412.40	-366600.40	775680.80
50.00	150.00	239842.60	-586619.20	771286.80
50.00	180.00	-80956.40	-656833.60	766696.60
50.00	210.00	-378986.33	-561766.00	763750.50
50.00	240.00	-597068.40	-326463.80	763278.60
50.00	270.00	-668943.00	-8984.20	765623.00
50.00	300.00	-573533.40	296189.80	769626.60
50.00	330.00	-337498.60	513182.00	774020.80
0.00	330.00	-503904.00	852243.20	22851.00
0.00	300.00	-884177.20	491209.00	13671.00
0.00	270.00	-1028903.00	4979.80	7324.00
0.00	240.00	-917673.20	-493357.60	4296.40
0.00	210.00	-562888.40	-874118.40	5371.00
0.00	180.00	-69432.80	-1023629.60	9277.00
0.00	150.00	413377.60	-911911.40	16698.60
0.00	120.00	787302.20	-571872.80	24804.40
0.00	90.00	943681.83	-52815.33	31167.67
0.00	60.00	830889.33	433998.83	34504.33
0.00	30.00	483494.60	812399.40	33788.60

equals 90° , the toolface cannot be computed, because the function $tf(\mathbf{a})$ is not defined, when $a_x = a_y = 0$ holds true.

Next, we compare the latter results with those, obtained using model (4), see Table 3. The obtained accuracy is very similar, even slightly worse. On the other hand, the model (4) leads to a much more complex nonlinear minimization problem to be solved. Further experiments should establish if there are cases when

Table 2. Results from the calibration, using the linear model (3)

a_x	a_y	a_z	comp. tf	error	comp. incl	error
0.00000	9.81605	0.00108	0.000	0.000	0.006	0.006
-0.00187	9.60862	1.77483	-0.011	0.011	10.465	0.465
-0.00039	9.20464	3.32969	-0.002	0.002	19.887	0.113
0.00109	8.50745	4.87664	0.007	0.007	29.822	0.178
0.00040	7.52117	6.32738	0.003	0.003	40.073	0.073
0.00156	6.30094	7.50205	0.014	0.014	49.973	0.027
-0.00430	4.91951	8.49967	-0.050	0.050	59.938	0.062
-0.00343	3.32881	9.20115	-0.059	0.059	70.111	0.111
-0.00031	1.69614	9.65744	-0.010	0.010	80.039	0.039
0.00000	0.00108	9.80126	-	-	89.994	0.006
3.05564	5.48063	7.51727	29.141	0.859	50.147	0.147
5.36162	3.18410	7.55978	59.295	0.705	50.482	0.482
6.17725	0.13979	7.61671	88.704	1.296	50.950	0.950
5.34648	-2.95013	7.66863	118.889	1.111	51.470	1.470
3.06713	-5.22806	7.70505	149.601	0.399	51.809	1.809
-0.12023	-6.04927	7.71009	181.139	1.139	51.877	1.877
-3.13003	-5.21794	7.68696	210.958	0.958	51.636	1.636
-5.38272	-2.96114	7.64445	241.184	1.184	51.213	1.213
-6.19794	0.16909	7.59403	271.563	1.563	50.770	0.770
-5.33626	3.24088	7.54617	301.272	1.272	50.397	0.397
-3.04133	5.48927	7.51037	331.011	1.011	50.119	0.119
-4.88847	8.49236	-0.00786	330.074	0.074	0.046	0.046
-8.58255	4.75541	0.03191	298.990	1.010	0.186	0.186
-9.88254	-0.13606	0.10774	269.211	0.789	0.625	0.625
-8.61874	-5.04982	0.19352	239.633	0.367	1.110	1.110
-4.95456	-8.70087	0.26520	209.659	0.341	1.517	1.517
0.02633	-9.99923	0.29185	179.849	0.151	1.672	1.672
4.82159	-8.70299	0.28781	151.013	1.013	1.657	1.657
8.45846	-5.17762	0.24339	121.472	1.472	1.406	1.406
9.86501	0.04465	0.15817	89.741	0.259	0.919	0.919
8.58912	4.84337	0.07855	60.582	0.582	0.456	0.456
4.99960	8.47390	0.01028	30.541	0.541	0.060	0.060

the seemingly more accurate model (4) should be preferred. The current state of our research, however, suggests that the basic relation (4) gives comparable results and, thus, should be preferred because of its simplicity.

Table 3. Results from the calibration, using the linear model (3)

a_x	a_y	a_z	comp. tf	error	comp. incl	error
0.00000	9.80610	0.00000	0.000	0.000	0.000	0.000
0.01071	9.58144	1.79767	0.064	0.064	10.626	0.626
0.02139	9.16502	3.37025	0.134	0.134	20.190	0.190
0.02965	8.45899	4.93054	0.201	0.201	30.237	0.237
0.03271	7.46843	6.38896	0.251	0.251	40.545	0.545
0.03350	6.24978	7.56370	0.307	0.307	50.433	0.433
0.02467	4.87386	8.55637	0.290	0.290	60.333	0.333
0.01843	3.29446	9.24534	0.320	0.320	70.387	0.387
0.01210	1.67631	9.68477	0.414	0.414	80.180	0.180
0.00000	0.00000	9.80610	—	—	90.000	0.000
3.03319	5.43059	7.55588	29.185	0.815	50.538	0.538
5.28494	3.15470	7.55705	59.166	0.834	50.839	0.839
6.06388	0.14401	7.56749	88.640	1.360	51.287	1.287
5.22186	-2.90706	7.57828	119.105	0.895	51.739	1.739
2.96037	-5.15165	7.59054	150.116	0.116	51.947	1.947
-0.18347	-5.95412	7.59552	181.765	1.765	51.894	1.894
-3.13954	-5.12407	7.59530	211.496	1.496	51.649	1.649
-5.33916	-2.88758	7.59335	241.594	1.594	51.362	1.362
-6.11696	0.20791	7.59076	271.947	1.947	51.120	1.120
-5.24463	3.24094	7.58371	301.714	1.714	50.890	0.890
-2.96804	5.45629	7.57160	331.455	1.455	50.636	0.636
-4.82237	8.51211	-0.01023	330.467	0.467	0.060	0.060
-8.48801	4.82994	-0.01057	299.641	0.359	0.062	0.062
-9.80610	0.00000	0.00000	270.000	0.000	0.000	0.000
-8.60074	-4.85926	0.01085	240.534	0.534	0.063	0.063
-5.02271	-8.47755	0.01687	210.646	0.646	0.098	0.098
-0.13020	-9.77458	0.00656	180.763	0.763	0.038	0.038
4.60010	-8.50730	0.00355	151.599	1.599	0.021	0.021
8.20770	-5.03397	-0.00369	121.522	1.522	0.022	0.022
9.63324	0.12261	-0.01938	89.271	0.729	0.115	0.115
8.41512	4.86818	-0.02560	59.950	0.050	0.151	0.151
4.91044	8.46595	-0.02874	30.115	0.115	0.168	0.168

4. Conclusions and discussion

We have proposed an algorithm for calibrating MEMS accelerometers in laboratory conditions that aims to compensate the main deterministic errors. Based on numerical experiments, we suggest that a linear relationship between the raw and the calibrated data, depending on 12 parameters, should be used. We obtain those parameters by solving a least-squares problem, comparing the output

accelerometer vector to the gravitational acceleration.

If the inclination is close to 90° , however, the following issue might arise. In this case the components of the gravitational acceleration on the x - and y -axes are small. Thus, a large relative error in those components might have little effect on the objective function (5), but lead to a substantial error when computing the toolface with formula (1). One possible approach is to adjust the objective function by comparing the toolfaces and inclinations, computed with formulae (1) and (2).

Further, numerical methods should be proposed for the effective solution of the minimization problem. Also, one can consider how to choose the s different positions in order to obtain optimal (in some sense) results.

References

- [1] P. Aggarwal, Z. Syed, X. Niu, N. El-Sheimy, Thermal Calibration of Low Cost MEMS Sensors for Integrated Positioning, Navigation Systems, The Institute of Navigation National Technical Meeting (2007) 22–24.
- [2] P. Aggarwal, Z. Syed, X. Niu, N. El-Sheimy, A standard testing and calibration procedure for low cost MEMS inertial sensors and units. Journal of navigation 61 (2008) 323–336.
- [3] H. Illfelder, K. Hamlin, G. McElhinney, A gravity-based measurement-while-drilling technique determines borehole azimuth from toolface and inclination measurements. In ADDE 2005 National Technical Conference and Exhibition, Houston, Texas.
- [4] S. Luczak, W. Oleksiuk, M. Bodnicki, Sensing Tilt With MEMS Accelerometers, IEEE Sensors Journal 6 (2006) 1669–1675.
- [5] I. Georgieva, C. Hofreither, T. Ilieva, T. Ivanov, S. Nakov, Laboratory calibration of a MEMS accelerometer sensor. ESGI'95 Problems and Final Reports (2013) 61–86.
- [6] H. Hou, Modeling inertial sensors errors using Allan variance. Library and Archives Canada, 2005.

Future Cyber Attacks Modelling & Forecasting

Zlatogor Minchev, Georgi Dukov, Doychin Boyadzhiev, Plamen Mateev

1. Problem Definition

Following leading industrial experience approaches for cyber intelligence intrusion & threats exploration (see e.g. [1], [2]), jointly with some recent research results, concerning the problem area (to note [3], [4]) three key steps have been arranged for future cyber attacks studying: (i) establishment of cyber risks landscape, implementing both intrusions and attack vectors due to expert, and literature data; (ii) supportive models for multiple intrusions risk reassessment towards expected attack vectors; (iii) experimental application of (i) & (ii) results for cyber risks landscape evolutionary prognosis. Practical realization of these steps with more details will be further outlined.

2. Cyber Risks Landscape Establishment

Solving the risk assessment is a complex target [5] that could be started with definition of rectangular matrix of intrusions vs attack vectors $[m \times n]$ $R = [r_{ij}]$, $i = 1 \div m$, $j = 1 \div n$. The matrix R values are real numbers, regarding the risk R^* function values, calculated as follows:

$$(1) \quad R^*(T, V, I) = P(T_k|S_l) \times P(V_i|T_k, S_l) \times P(I_j|T_k, S_l),$$

$$i = 1 \div m, \quad j = 1 \div n, \quad k = 1 \div p, \quad l = 1 \div q,$$

where $P(\dots)$ are the probability values, calculated for the $k^{\text{-th}}$ threat T_k vs certain $l^{\text{-th}}$ attack scenario S_l . The $i^{\text{-th}}$ vulnerability V_i and $j^{\text{-th}}$ impact I_j values are taken vs the explored T_k .

The next important exploration stage is (1) practical implementation. This could be achieved via a probabilistic multiple intrusions modelling towards landscape impacts & vulnerabilities with follow-up experimental matching to risk matrix R . The recalculation of expert beliefs is giving new opportunity for threats T evolution.

The above idea could be expressed with Dirichlet distribution [6], regarding the multidimensional case as follows:

$$(2) \quad L \sim a \text{ priori } Dir(\alpha) \times R,$$

where: L is the probabilistic landscape, produced as a result of expert beliefs from risk matrix R multiplication with a priori Dirichlet probabilistic cyber landscape representation – $Dir(\alpha)$, defined for k -dimensions with shape parameters α_i .

Further exploration of the problem could be achieved by intrusions modelling and cyber risk reassessment, using additional expert and literature data for better problem understanding. A suitable approach in this context is presented in the next paragraph.

3. Supportive Intrusions Modelling for Risk Reassessment

The idea for multiple cyber intrusions modelling, concerning different cyber attacks and risk matrix reassessment could be implemented, using the “*Entity – Relationship*” machine representation, combined with probabilistic forecasting, following the experience from [3], [4] but with some modifications for current tasks specifics.

The general modelling idea, after [7], is to use an oriented graph of m nodes (representing the *Entities*) and n arcs (noting the *Relations* between entities in the model).

The arcs in the graph are marked in a quadratic $[p \times p]$ incident matrix $A = [a_{ij}]$, $i = 1 \div p$, $j = 1 \div p$. The matrix A values are binary numbers, regarding the presence ($a_{i,j} = 1$) or absence ($a_{i,j} = 0$) of an arc between the nodes i and j . For each arc $a_{i,j}$, a probabilistic risk coefficient r ($0 \leq r \leq 1$) is assumed, following R^* assessment from (1).

The resulted classification of the graph nodes for the $k^{\text{-th}}$ arc is calculated, using a multiplication approach (as the studied graph model assumes to represent simultaneous events) for both input $a_{k,i}$ and output arcs $a_{k,j}$, uncertainty correcting constants $c_{k,i}$, $c_{k,j}$ (for coping model uncertainties and noisy expert data) and their $r_{k,i}$, $r_{k,j}$ risk values, producing a resulting forward (input) – R_f and backward (output) R_b probabilistic risk assessments:

$$(3) \quad R_f = \prod_{i=1}^p a_{k,i} \cdot c_{k,i} \cdot r_{k,i}, \quad i = 1 \div p, \quad k = 1 \div p,$$

$$(4) \quad R_b = \prod_{j=1}^q a_{k,j} \cdot c_{k,j} \cdot r_{k,j}, \quad j = 1 \div q, \quad k = 1 \div q.$$

The resulting model probabilistic system risk – R_s with uncertainty constant correction c_s could be defined as:

$$(5) \quad R_s = c_s \cdot (R_f / R_b).$$

Thus, in accordance with the practical necessities of cyber risk assessment a three dimensional R^3 ($x - R_f$, $y - R_b$, $z - R_s$) model probabilistic classification could be accomplished (see (1) and Section 4).

The a posteriori matrix – R' risk values of the a priori one – R (both of size $m \times n$) for the k -th intrusion ($k \leq m$, m – complete number of studied intrusions) are assessed with the new probabilistic risk model results R_{s_i} ($i = 1 \div p$, p – number of entities, referring to the studied risk of interest) as follows:

$$(6) \quad R'_{k,j} = R_{k,j} \times \prod_{i=1}^p R_{s_i,j}, \quad k \leq m, \quad i = 1 \div p, \quad j = 1 \div n.$$

The cyber threats landscape probabilities are reassessed after (2) for the new R' .

Finally, it is also important to note that the presented ideas have to be considered and in the dynamic sense (i.e. vs time – t) as the risk assessment is not a static process, so: $R^3 \rightarrow R^4$.

4. Experimental Application

The studied context is outlined, following the recent and future trends and prognosis, noted in [4], [8], [9]. The “Privacy & Social Engineering”, “Malware & Targeted Attacks”, “Data Breaching & Espionage” and “Compromised by Design Equipment” were the selected attack vectors, matched vs intrusions from: “E-mails”, “Social Networks”, “Web Links”, “Data Sharing” and “Chat”. For further exploration, following the complex nature of modern social networks, the intrusion models were aggregated around: “E-mails”, “Social Networks” & “Data Sharing”.

The probabilistic cyber attacks modelling was organized in Matlab R2011b environment [10].

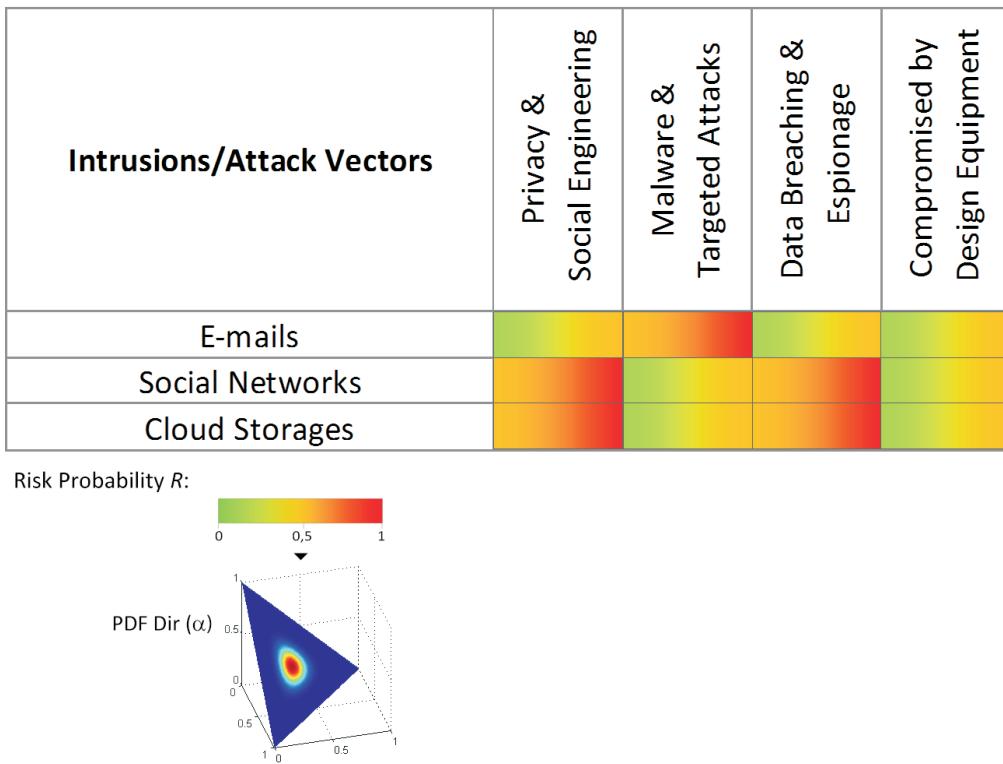
All modeling and risk probabilistic analyses were performed in I-SCIP-RA, v.1.0 software environment. The application was developed, following the “Enti-

ties” – “Relationships” ideas from I-SCIP-SA [7] and (3)–(5) for suitable probabilistic system risk assessment. Overall model entities graphical classification into a “System Risk Diagram” was organized as follows: “non-critical” & “critical”, marked by the north-west/south-east main diagonal of the diagram). Additional entities system subclassification was made for “active” vs “passive” ones (denoted consecutively with white & grey colours).

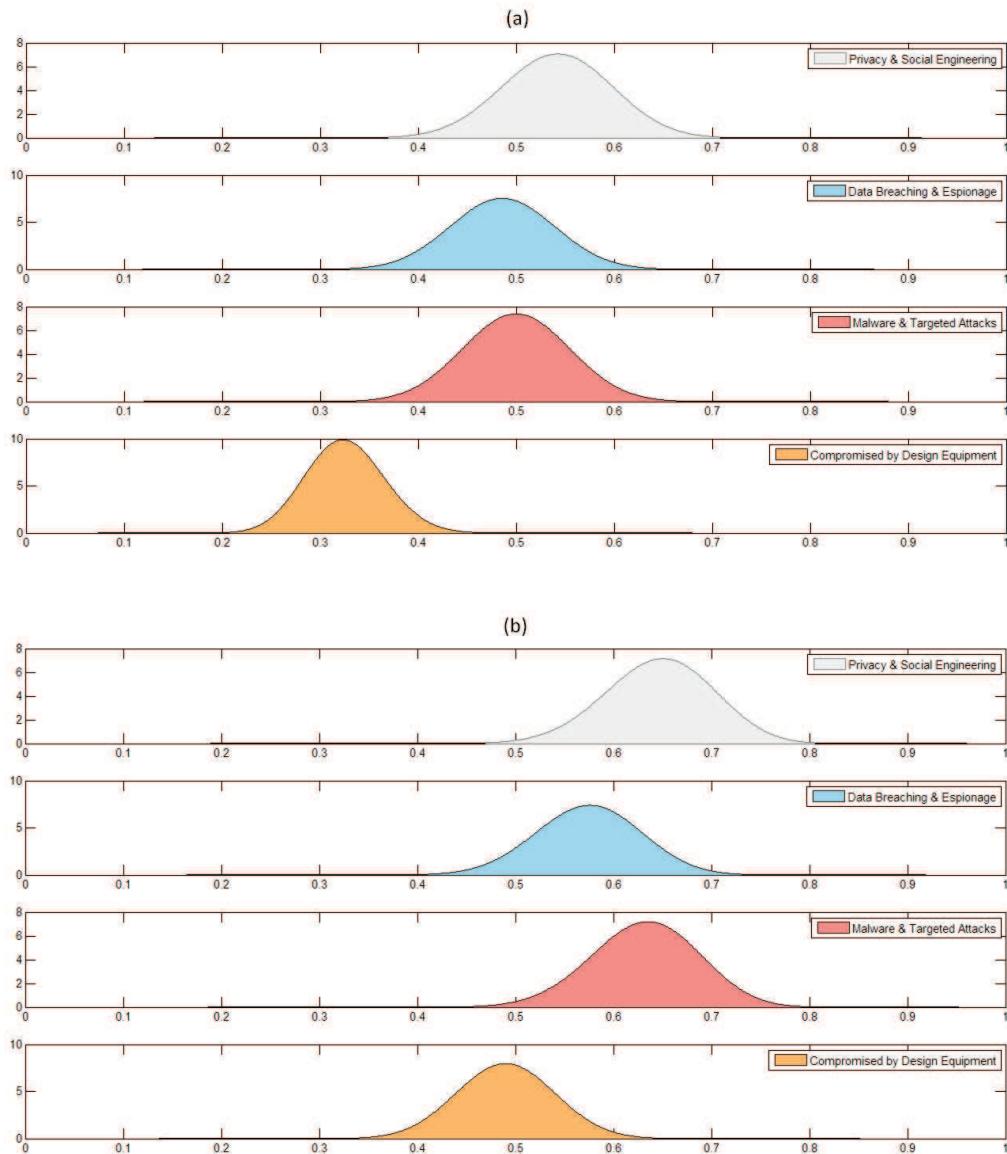
Dynamically, the R^4 space of risk models and their relevant classifications of: R_f , R_b & R_s vs time – t (for a five-step time horizon up to year 2021) was given further for each attack vectors columns and intrusions rows.

Here it should be noted that the a priori cyber landscape L probabilistic risk values and three risk system models (“E-mails”, “Social Networks” & “Data Sharing” intrusions) for a posteriori risk reassessment were developed, using Techno-Logica Ltd. expert data, working group discussions and own research experience.

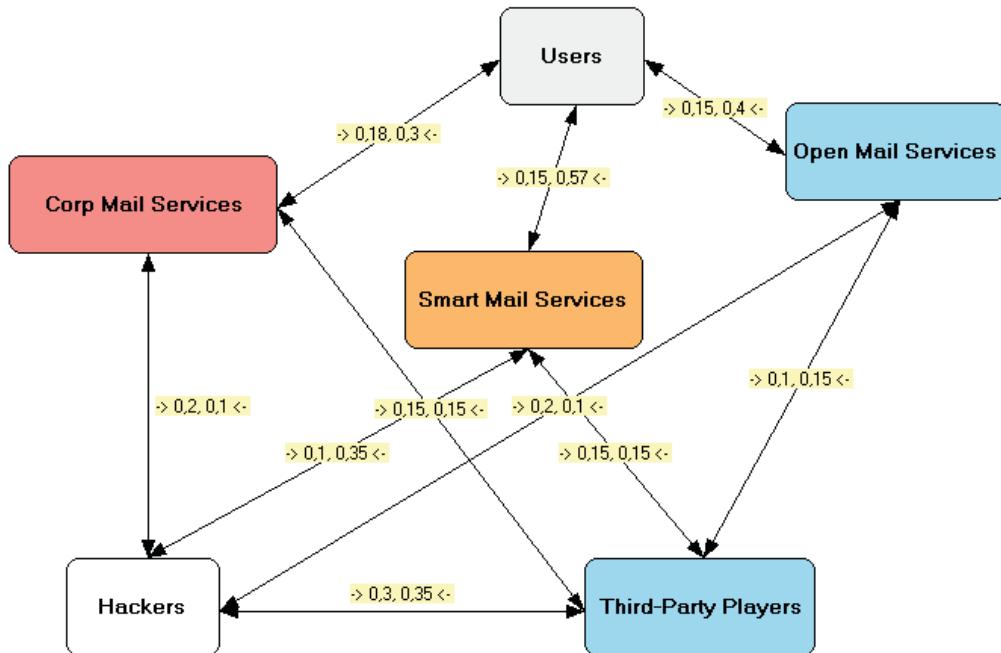
(i) *Probabilistic cyber risks landscape L evolution prognosis 2016 → 2021*



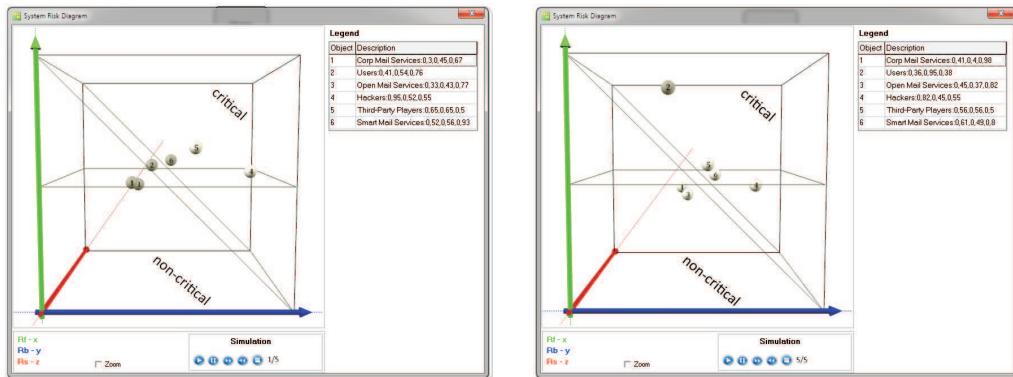
(ii) Cyber risks landscape L attack vectors *a priori* (a) and *a posteriori* (b) probabilistic distributions



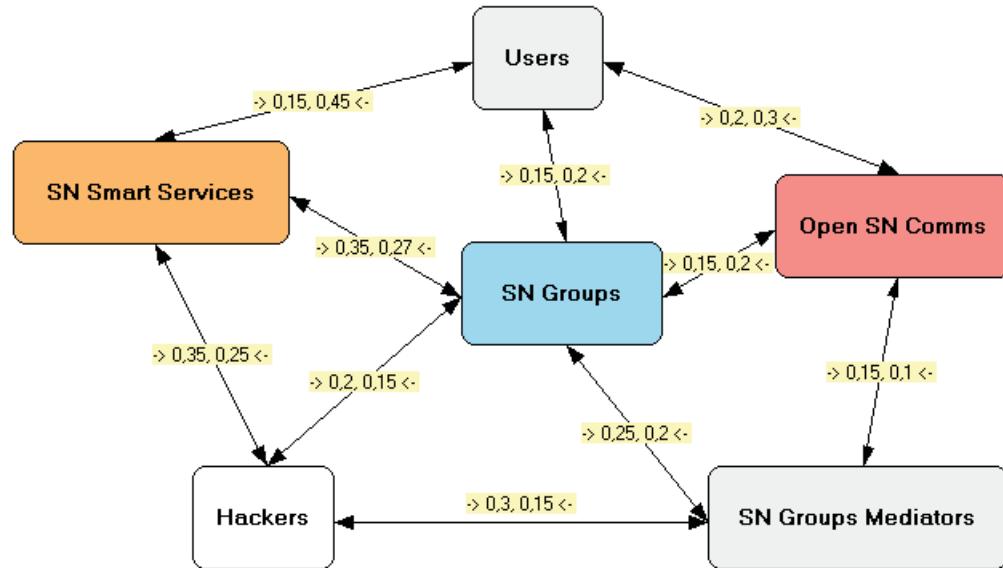
(iii.1) E-mails intrusions modelling for cyber risks landscape L reassessment

**Entities vs Attack Vectors:**

- **Users:** Privacy & Social Engineering
- **Corp Mail Services:** Malware & Targeted Attacks
- **Third-Party Players, Open Mail Services:** Data Breaching & Espionage
- **Smart Mail Services:** Compromised by Design Equipment

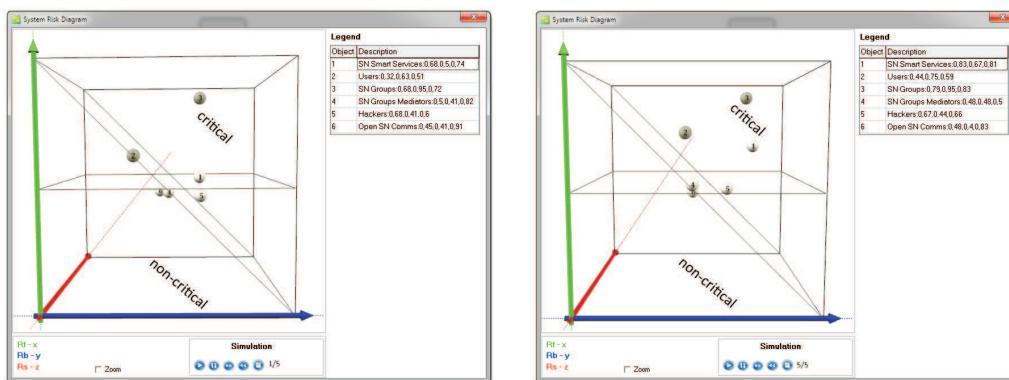


(iii.2) Social networks intrusions modelling for cyber risks landscape L reassessment

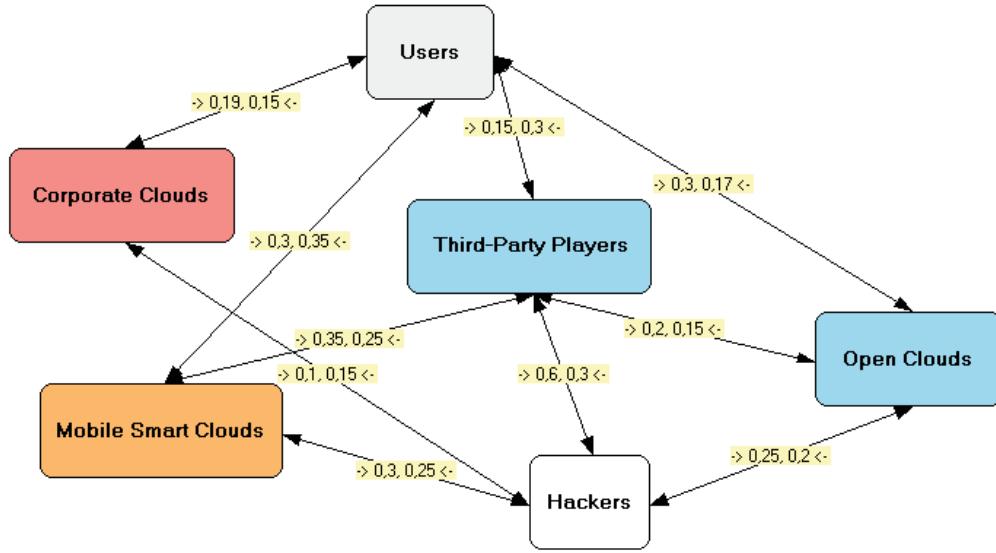


Entities vs Attack Vectors:

- Users, SN Groups Mediators: Privacy & Social Engineering
- Open SN Comms: Malware & Targeted Attacks
- SN Groups: Data Breaching & Espionage
- SN Smart Services: Compromised by Design Equipment

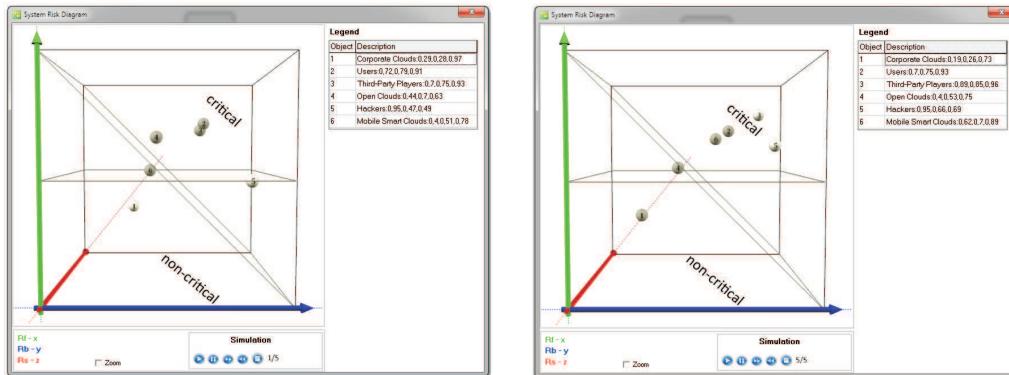


(iii.3) Cloud storages intrusions modelling for cyber risks landscape L reassessment



Entities vs Attack Vectors:

- Users: Privacy & Social Engineering
- Corporate Clouds: Malware & Targeted Attacks
- Third-Party Players, Open Clouds: Data Breaching & Espionage
- Mobile Smart Clouds: Compromised by Design Equipment



5. Discussion

The comprehensive understanding of future cyber attacks is producing numerous multidimensional problems that are difficult to be adequately tackled in

todays' fast evolving digital world. A suitable support in this sense is proposed in the present study, combining data from both experts and literature. Further uncertainty coping is achieved with probabilistic modelling that is practically implemented in an ad-hoc designed and prototyped risk modelling and assessment research environment.

The cyber landscape risk results generalization could be summarized up to year 2021 as follows: (i) attack vectors expecting priorities of: "Privacy & Social Engineering", "Data Breaching & Espionage", "Malware & Targeted Attacks" with moderate ones for "Compromised by Design Equipment"; (ii) intrusions critical points for (i), encompassing: "Users", "Open-", "Mobile-" & "Smart-" Web 3.0 services, noting also "Third-Party Players" key role.

What however stays uncertain is the future cyber risks landscape validation and verification in advance, being generally an arguable problematic area. A useful added value in this sense could be found in the more active role of human-computer hybrid simulations, combined with detailed cyber attacks mathematical modelling.

Acknowledgements

The study is partially supported by SP15-FMIIT-007 project, Faculty of Mathematics & Informatics, Plovdiv University 'Paisii Hilendarski'.

References

- [1] Seven Ways to Apply the Cyber Kill Chain with a Threat Intelligence Platform, Lockheed Martin, 2015, Available at: <https://goo.gl/UA3Bnd>.
- [2] IT Executive Guide to Security Intelligence, IBM Security, 2015, Available at: <https://goo.gl/4kPwBa>.
- [3] Minchev, Z., Dukov, G., Ivanova, T., Mihaylov, K. Boyadzhiev, D., Mateev, P., Bojkova, M., Daskalova, N. Cyber Intelligence Decision Support in the Era of Big Data, In ESGI 113 Problems & Final Reports Book, Chapter 6, FASTUMPRINT, pp. 85–92, 2015.
- [4] Minchev, Z. & Dukov, G. Emerging Hybrid Threats Modelling & Exploration in the New Mixed Cyber-Physical Reality, BISEC 2016, Belgrade Metropolitan University, October 15, pp. 13–17, 2016.

- [5] Rausand, M. Risk Assessment: Theory, Methods, and Applications, John Wiley & Sons, 2011.
- [6] Ng, K. W., Tian, G.-L., Tang, M.-L. Dirichlet and Related Distributions: Theory, Methods and Applications, John Wiley & Sons, 2011.
- [7] Minchev, Z. Methodological Approach for Modelling, Simulation & Assessment of Complex Discrete Systems, In Proceedings of National Informatics Conference Dedicated to the 80th Anniversary of Prof. Barnev, IMI-BAS, Sofia, November 12–13, 2015, pp. 102–110, 2016.
- [8] Minchev, Z. Cyber Threats Identification in the Evolving Digital Reality, In Proceedings of Ninth National Conference “Education and Research in the Information Society”, Plovdiv, Bulgaria, May 26–27, pp. 011–022, 2016.
- [9] Built Environment 2050. A Report on Our Digital Future, BIM 2050 Team, 2014, Available at: <https://goo.gl/iSqnm6>.
- [10] Cho, M., Martinez, W. Statistics in MATLAB: A Primer, Chapman and Hall/CRC Press, 2015.

Post-Processing for Beam Elements: Calculating the Second Order Work and Strain Energy

Stanislav Stoykov, Dimitar Fidanov, Maria Datcheva, Stefka Dimova

1. Introduction and problem formulation

Modern design of excavation pits includes often the simulation of the problem using Finite Element (FE) models. Primary objective for this kind of models is to predict deformations of the excavation pit support and the surroundings. As a secondary objective, it would be very useful to know, if the modelled structure is near to a failure state or not.

A criterion for failure (bifurcation in the solution) for models using elasto-plastic formulation is the Hill's failure criterion [7] based on the second order work. The second order work is defined at the material point by:

$$u = d^2W = d\sigma : d\varepsilon.$$

Hill's condition of stability is defined by:

$$d^2W > 0.$$

A related measure is the strain energy, which is the integrated value of the second order work over a volume V:

$$U = \int_V u \, dV = \int_V d\sigma : d\varepsilon \, dV.$$

For this kind of modelling task, usually a commercial software FE package is used as a black-box model with no access to the source code and no option for altering the algorithms included in the code. This is the case because this kind of software packages are widely known, well tested and accepted by the professional engineer community.

To model excavation pits and especially their support usually a plain-strain analysis is performed where beam elements are employed. The software package Plaxis [1], used by the company Gruner for this aim, is widely used in geotechnical engineering. This software package provides a beam element based on Mindlin's theory of plates.

Unfortunately, the package Plaxis only provides a small set of output variables at the element nodes:

1. u_x : nodal displacements in direction x ;
2. u_y : nodal displacements in direction y ;
3. ϕ : nodal rotation;
4. N : normal force extrapolated to the node*;
5. Q : shear force extrapolated to the node*;
6. M : bending moment extrapolated to the node*.

* The extrapolation function is unknown.

The objectives, posed by the company Gruner, were to show a way to calculate the strain energy of one beam element and the second order work as a post-processing after finishing the FE-simulation. This must be done by using only the nodal variables cited above.

2. Our comments and suggestions

To calculate the strain energy, the equations of motion of the beam are needed. There are different kinds of Mindlin beam and plate models (e.g. [2]–[5]) in the literature, depending on the different hypothesis on the behaviour of the medium and the objects under consideration. From the test example, provided by the company, we found out, that the model used is the 2D plate element, as described on pp. 30, 31 of the Plaxis 2D 2016 Scientific manual [6]. This model uses 3- and 5-node plate elements with three degrees of freedoms (d.o.f.s) – two transitional d.o.f.s u_x , u_y and one rotational d.o.f. ϕ_z . The equations of motion are not explicitly defined there. The model used in Plaxis for 2D analysis is equivalent to Timoshenko's beam model. Thus, we decided to use Timoshenko's beam model with the same 3 d.o.f.s and the same hypothesis about the cross section of the beam as described in [5], p. 242.

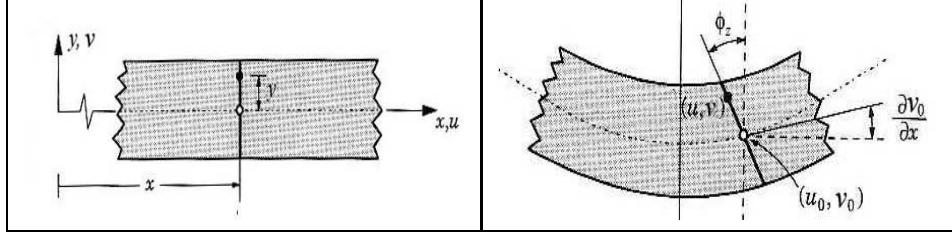
As for the second order work the following considerations can be expressed.

The chapter “Deformation theory” of the Plaxis 2D 2016 Scientific manual [6] begins with the statement (page 7) “A restriction is made in the sense that deformations are considered to be small”. For small deformations and elastic behaviour the problem for loss of stability (bifurcation of the solution) does not arise, because the problem is linear.

The Hill's failure criterion, as described in the paper [7], recommended by the company, is for the case of elastic-plastic solids with geometrical nonlinearity

(nonlinear strain-displacement relations). The Plaxis software uses linear strain-displacement relations. So we have not proposed a way to predict the stability or failure of the beam construction.

3. Equation of motion of Timoshenko beam



The displacement components of 2D beam are denoted by $u(x, y, t)$ and $v(x, y, t)$, where $u(x, y, t)$ represents the longitudinal displacement of the beam and $v(x, y, t)$ represents the transverse displacements. x and y are the space coordinates along the longitudinal and transversal axes of the beam and t is the time. Considering the hypotheses that the cross section, after deformation, remains plane and rotates about the transverse axis z , the displacement components are expressed by the displacements on the middle line of the beam ($y = 0, z = 0$):

$$u(x, y, t) = u_0(x, t) + y\phi_z(x, t),$$

$$v(x, y, t) = v_0(x, t),$$

where $u_0(x, t) \equiv u_x$ is the longitudinal displacement on the middle line, $v_0(x, t) \equiv u_y$ is the transverse displacement on the middle line and $\phi_z(x, t)$ is the rotation of the cross sections about z axis. As the stationary problem is considered below, the dependance of t will be ignored.

The strains are expressed by the displacement components, they are derived from Green's strain tensor, and they are reduced to two dimensional beam formulation:

$$\varepsilon_x = \frac{\partial u}{\partial x} = \frac{\partial u_0}{\partial x} + y \frac{\partial \phi_z}{\partial x},$$

$$\gamma_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} = \frac{\partial v_0}{\partial x} + \phi_z,$$

where ε_x is the direct strain and γ_{xy} is the shear strain.

3.1. Linear elastic case

In the case of linear elastic material behaviour, the stresses are expressed by the strains, considering Hooke's law:

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_x \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} E & 0 \\ 0 & \lambda G \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \gamma_{xy} \end{Bmatrix} = D^e \boldsymbol{\varepsilon},$$

where E is Young's modulus, G is shear modulus and λ is the shear correction factor.

3.2. Elastic perfectly-plastic case

In the case of elastic perfectly-plastic behaviour of the beam the stress-strain relation is of the same kind:

$$\sigma = D\varepsilon,$$

but with matrix D , given by formulae (3.5a,b) from the Plaxis 2D 2016 Material model manual [8], pp. 3–1, 3–2:

$$D = D^e - \frac{\alpha}{d} D^e \frac{\partial g}{\partial \sigma} \frac{\partial f^T}{\partial \sigma} D^e,$$

where $\alpha = 0$ in the elastic case, $\alpha = 1$ in the plastic case, f is the yield function, g is a plastic potential function and

$$d = \frac{\partial f^T}{\partial \sigma} D^e \frac{\partial g}{\partial \sigma}.$$

For more guidance see again [8], pp. 3–1, 3–2.

4. Strain energy

The strain energy of the structure is computed by the stresses and the strains:

$$U = \frac{1}{2} \int_V \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = \frac{1}{2} \int_V \boldsymbol{\varepsilon}^T \mathbf{D} \boldsymbol{\varepsilon} dV,$$

where V denotes the volume of the structure.

For simplicity of the explanation we will derive the strain energy for purely elastic beam.

In this case the strain energy has the following form:

$$U = \frac{1}{2} \int_V (\varepsilon_x \sigma_x + \gamma_{xy} \tau_{xy}) dV =$$

$$\begin{aligned}
&= \frac{1}{2} \int_V \left(\left(\frac{\partial u_0}{\partial x} + y \frac{\partial \phi_z}{\partial x} \right) E \left(\frac{\partial u_0}{\partial x} + y \frac{\partial \phi_z}{\partial x} \right) + \left(\frac{\partial v_0}{\partial x} + \phi_z \right) \lambda G \left(\frac{\partial v_0}{\partial x} + \phi_z \right) \right) dV = \\
&= \frac{1}{2} EA \int_L \left(\frac{\partial u_0}{\partial x} \right)^2 dL + \frac{1}{2} EI_z \int_L \left(\frac{\partial \phi_z}{\partial x} \right)^2 dL + \\
&\quad + \frac{1}{2} \lambda GA \int_L \left(\left(\frac{\partial v_0}{\partial x} \right)^2 + 2 \frac{\partial v_0}{\partial x} \phi_z + \phi_z^2 \right) dL,
\end{aligned}$$

where L denotes the length of the beam, A is the cross sectional area, I_z is the second moment of area:

$$I_z = \int_{\Omega} y^2 d\Omega.$$

It is assumed that the cross section of the beam is symmetric, thus the integral $\int_{\Omega} y d\Omega = 0$ and it is not included in the expression of the strain energy. For the cases of beams with non-symmetrical cross sections, this integral should be included in the expression.

5. FEM discretization

Again for simplicity we will show the FE technic by using 3-node quadratic finite elements. Each finite element, given by nodes x_1, x_2, x_3 , is transformed into the standard element in local coordinate system in the interval $[-1; 1]$. The shape functions are given by:

$$N_1 = -\frac{1}{2}(1 - \xi)\xi,$$

$$N_2 = (1 + \xi)(1 - \xi),$$

$$N_3 = \frac{1}{2}(1 + \xi)\xi,$$

where ξ is the local coordinate. The transformations from local to global coordinate systems and vice versa are expressed by:

$$x = \frac{l}{2}\xi + \frac{(x_1 + x_3)}{2},$$

$$\xi = \frac{2x}{l} - \frac{(x_1 + x_3)}{l},$$

$l = x_3 - x_1$ is the length of the finite element.

The displacement components on the middle line are expressed by their values on the nodal points and the shape functions:

$$u_0(\xi) = u_0^1 N_1 + u_0^2 N_2 + u_0^3 N_3,$$

$$v_0(\xi) = v_0^1 N_1 + v_0^2 N_2 + v_0^3 N_3,$$

$$\phi_z(\xi) = \phi_z^1 N_1 + \phi_z^2 N_2 + \phi_z^3 N_3,$$

where, for example, u_0^1, u_0^2, u_0^3 are the values of the longitudinal displacement in the nodal points of the finite element. The derivatives of the displacements can be expressed on the following way:

$$\frac{\partial u_0}{\partial \xi} = u_0^1 \frac{\partial N_1}{\partial \xi} + u_0^2 \frac{\partial N_2}{\partial \xi} + u_0^3 \frac{\partial N_3}{\partial \xi}.$$

Taking into account the expression for the strain energy derived in the previous section, and using the finite element discretization, the strain energy can be computed by the values of the displacement components on the nodal points:

$$U = \frac{1}{2} \sum_e \left(EA \int_{-1}^1 \left(\frac{\partial u_0}{\partial \xi} \right)^2 \frac{2}{l} d\xi + EI_z \int_{-1}^1 \left(\frac{\partial \phi_z}{\partial \xi} \right)^2 \frac{2}{l} d\xi + \lambda GA \int_{-1}^1 \left(\frac{\partial v_0}{\partial \xi} \right)^2 \frac{2}{l} d\xi \right) \\ + \frac{1}{2} \sum_e \left(2\lambda GA \int_{-1}^1 \frac{\partial v_0}{\partial \xi} \phi_z d\xi + \lambda GA \int_{-1}^1 \phi_z^2 \frac{l}{2} d\xi \right),$$

where summation over all finite elements denoted by e is implemented.

The integration in the local coordinate system can be performed by Gauss quadrature formula. Three Gauss integration points are enough to compute the integrals:

$$\int_{-1}^1 f(\xi) d\xi = c_1 f(\xi_1) + c_2 f(\xi_2) + c_3 f(\xi_3)$$

with nodes $\xi = (-\sqrt{0.6}, 0, +\sqrt{0.6})$ and coefficients $c = (5/9, 8/9, 5/9)$.

In the computation of the strain energy one needs to know the derivatives of the displacement components. One way to obtain them is by using the derivatives of the shape functions and the values of the displacement components on the nodal points, as shown above. The other way is to use the normal and shear forces and the bending moment provided as output from Plaxis.

$$Q = \lambda AG \left(\frac{\partial v_0}{\partial x} + \phi_z \right),$$

i.e. from here the value of $\frac{\partial v_0}{\partial x}$ is known.

$$M = EI \frac{\partial \phi_z}{\partial x},$$

i.e. the value of $\frac{\partial \phi_z}{\partial x}$ is known.

$$N = A\sigma_x = EA \left(\frac{\partial u_0}{\partial x} + y \frac{\partial \phi_z}{\partial x} \right),$$

i.e. from here the value of $\frac{\partial u_0}{\partial x}$ is known.

Remark. The FE technic when using 5-node finite elements is the same. The form functions are those on p. 28 of the PLAXIS 2D 2016 Scientific manual [6].

Acknowledgements. Stefka Dimova thanks to Assoc. Prof. Todor Angelov for the fruitful discussions.

References

- [1] PLAXIS <https://www.plaxis.com>.
- [2] Z. Bittnar, J. Kruis. Modified beam element based on Mindlin theory. S. N. Atluri et al. (eds) Computational Mechanics'95. Springer Verlag 1995, p. 1656.
- [3] Bending of beams – Mindlin theory. <http://mech.fsv.cvut.cz/~zemanj/teaching/mk10/lectures/lecture2.pdf>.
- [4] K. Washizu. Variational methods in elasticity and plasticity. Pergamon press, 1982.
- [5] J. N. Reddy. An introduction to Nonlinear Finite element analysis, 2015, p. 242.
- [6] [https://www.plaxis.com/support/manuals/plaxis-2d-manuals/PLAXIS 2D 2016 – Scientific Manual](https://www.plaxis.com/support/manuals/plaxis-2d-manuals/PLAXIS%202D%202016%20-%20Scientific%20Manual).
- [7] R. Hill. A general theory of uniqueness and stability in elastic-plastic solids. Journal of the Mechanics and Physics of Solids, V. 6, 236–249, 1958.
- [8] [https://www.plaxis.com/support/manuals/plaxis-2d-manuals/PLAXIS 2D 2016 – Material Models](https://www.plaxis.com/support/manuals/plaxis-2d-manuals/PLAXIS%202D%202016%20-%20Material%20Models).

Mathematical Model of Residential Storage Water-heating System

Masoud Ghaderi Zefreh, Mariia Dvoriashyna, Mihail Galabov, Lyudmil Yovkov,
Konstantinos Liolios, Stanislav Darachev

Abstract

The present document reports the work carried on during ESGI 120 in Sofia. The considered problem is *Mathematical Model of Residential Storage Water-heating System*. We strove to model the device from Melisa Climate Ltd which controls the water heater system in the house. We used two approaches to model the problem that complement each other. The first one represents the dynamics inside the heater and the second model considers the control loop run by the device without assuming any delay caused by first block. Although an integrated model is not developed, the ingredients for such one are presented.

Key words: convection/diffusion heat transfer, Boussinesq approximation, feed-back control-loop

1. Introduction

The low-cost commonly available water heater is inherently a lossy device and this means that a high percentage of the energy consumed for heating water ends up being lost as heat to the surroundings. Informal measurements and practice show that when a family of four people switches on and off the water heater just when required, savings of up to 80% are recorded.

Our goal is to develop a model of the water heater to perform energy saving calculations and help the customer to operate their water heaters in the most energy-efficient way.

1.1. Definition of the problem (Melissa Climate)

An electric water heater consists of: an inner steel tank, that holds the water being heated, insulation that surrounds the tank so as to decrease the amount of heat loss, pipe to allow cold water to enter the tank, pipe to allow hot water to leave the tank, thermostat that reads and controls the temperature of the water inside the tank, heating element that heats the water by means of electricity and other components for safety and maintenance.

Water temperature inside the heater is controlled by the mechanical thermostat. The temperature may usually be set by the user somewhere in between 40

and 70 °C. A microcontroller is used to gather real-time information from a water heater.

The information is collected by different sensors (Fig. 1) and consists of data about: current temperature of the cold water pipe, temperature of the hot water pipe, environment temperature (home temperature), electric current and voltage. Important notice: the temperature sensors of the cold and hot water pipes are installed ONTO the pipe itself. There is significant temperature loss depending on the pipe diameter, material and the like.

1.2. Overview

The problem was observed from two different but complementary perspectives. First, a *closed-flow* problem was considered in order to understand the dynamics inside the heater, Section 2. The process is simplified to a 1-dimensional problem along the height of the water tank and spacial temperature distribution is investigated. Then, in Section 3, we analyzed the temperature dynamics during water consumption, assuming uniform temperature distribution inside the heater. The realistic solution would be a combination of both models, since the time required to reach uniform temperature in the second model should be estimated from the first one. Moreover, the assumption of uniform temperature inside the tank does not appear to be imposing any stringent premises and the results in Section 3 seem to comply with those provided by the company.

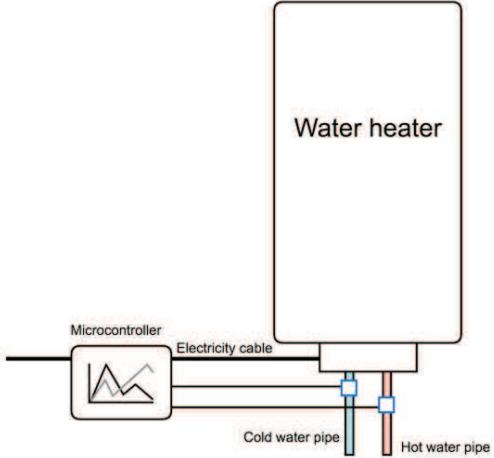


Figure 1. Schematic view of the heater system

2. Temperature dynamics inside the Water Heater

In this section, we develop a model of the temperature distribution inside the water tank. Consider the case when the inflow and outflow valves are closed. The heater is turned on at time t_0 . Changes in temperature impose a density variations along the tank, which drives a water flow due to natural convection.

The density difference is proportional to the temperature difference as:

$$(1) \quad \rho - \rho_0 = -\beta(T - T_0),$$

where β is thermal expansion coefficient, T and ρ are fluid temperature and density, and T_0 and ρ_0 are reference values for them. The fluid motion is then governed by Navier–Stokes and continuity equations for incompressible flow, and Boussinesq approximation (see [5]) is used to take into account density differences. The temperature variation in the domain is described with a heat equation. Therefore, the governing equations are

$$(2a) \quad \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \nabla \cdot \mathbf{v} = -\frac{1}{\rho_0} \nabla P + \nu \nabla^2 \mathbf{v} - \mathbf{g} \beta(T - T_0),$$

$$(2b) \quad \nabla \cdot \mathbf{v} = 0,$$

$$(2c) \quad \frac{\partial T}{\partial t} + \mathbf{v} \nabla T = \kappa \nabla^2 T,$$

where \mathbf{v} is fluid velocity, P is pressure, ν is fluid viscosity and κ is thermal diffusivity. The constants used in Eq. (2) and in other equations are given in Table 1.

Table 1. Values of the physical parameters

Constant	Value
ρ	988 kg/m ³
c_p	4185 J/(kg. °C)
λ	0.59 W/(m. °C)
V	225 m ³
I	{0; 13.5}
κ	1.4269 e-07 W.m ² /J
L	1 m
T_f	500 s
V_0	0.1963 m ³
c_m	{0; 0.0211}
T_A	25 °C
T_H	70 °C

Eq. (2) are too complex to attack directly. We assume that the domain is 1-dimensional with spacial dimension x along the vertical axis of the tank. $x = 0$ corresponds to the bottom of the tank and $x = L$ to the top part. We assume in Section 2.1 that the heat is supplied at $x = 0$. Therefore, the hot bottom plate is heating up the cold water.

To avoid solving Navier–Stokes equations in Eq. (2) for the fluid motion inside the water tank we use an estimation for the spatial velocity distribution of the fluid. Since, we are looking at a closed-flow system, the velocity on top and bottom of the heater is zero and we assume parabolic flow profile:

$$(3) \quad v(x) = \frac{4x(L-x)}{L^2}v_{\max},$$

where we estimated $v_{\max} = \sqrt{2gL\beta\Delta T}$ by equalizing the pressure difference $\Delta p = \rho\beta\Delta TgL$ to the dynamic head $\rho v^2/2$ in absence of viscous forces in the flow.

Note that this velocity choice does not satisfy the continuity equation in Eq. (2). This is a drawback of our model, however, we reason that this velocity is more realistic than simply a constant one (which satisfies the continuity equation).

2.1. Simplified model

After imposing the velocity distribution v for the fluid motion, we are left only with the heat equation (2c) from the original system Eq. (2). To close the problem, we need to impose initial and boundary conditions. The boundary conditions for this case would be Dirichlet for the lower plate

$$(4) \quad T\Big|_{x=0} = T_H$$

and Neumann (no heat loss) for the upper plate

$$(5) \quad \left. \frac{\partial T}{\partial x} \right|_{x=L} = 0,$$

where $T_H = 70^\circ\text{C}$ is the temperature of the hot water at the bottom. The initial condition is chosen to agree with boundary conditions at $t = 0$ and has a shape of piecewise linear function:

$$(6) \quad T_0(x) = \begin{cases} \frac{(T_H - T_A)(x - 0.1)}{0.1} + T_A, & 0 \leq x < 0.1, \\ T_A, & x \geq 0.1, \end{cases}$$

where $T_A = 25^\circ\text{C}$ is the ambient temperature, i.e. we are taking into account the heat loss from the heater.

We used finite difference approximation for the derivatives and thus to solve Eq. (2c). A fully implicit scheme with 100 grid points in space and 5000 grid points in time was implemented.

Fig. 2 illustrates the temperature variation for a few points inside the tank and Fig. 3 shows all the temperature waves at all times. The average speed of the temperature wave is much smaller than the maximum velocity. This suggests that for an inflow of water to heat up, there is a delay time one has to take into consideration. The delay temperature depends on the temperature of the inflow water.

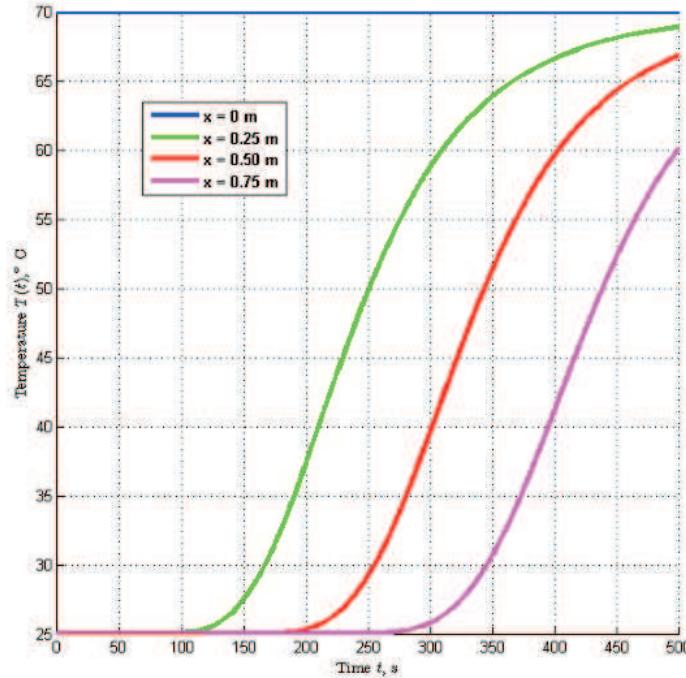


Figure 2. Temperature distribution for $v_{\max} = 0.003$ [m/s]

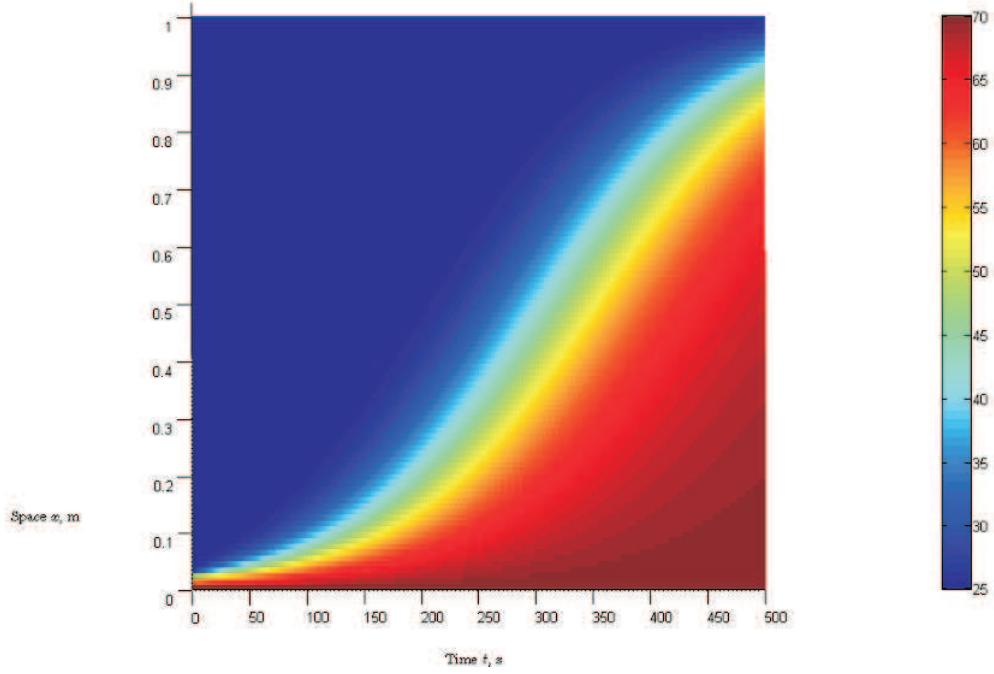


Figure 3. Temperature profile evolution of the heater

3. Water Heater Control System

Here we assume the water heater is uniform at all times. We also suppose there is no delay in water heater system. The more realistic model shall be constructed by relaxing the later assumption. The general control block of the control system is shown on Figure 4.

Here we don't take cold water variations into account therefore $T_c(t) \equiv T_c$. Also we consider the case when the desired temperature is set to a constant value $T_{set}(t) \equiv T_h$. The dynamics of the thermostat and the heater power, $P(t)$, can be described as a function of temperature and time:

$$(7) \quad P(t, T_h) = \begin{cases} VI, & T_{set} - T_h > \epsilon, \\ 0, & \text{otherwise,} \end{cases}$$

i.e. we assume either the heater works with full power or it is off.

For the flow of cold water we assume that it is a piecewise constant function, that is, the tap is either full open or full closed and also the consumption (flow) is the same whenever the tap is open. Therefore, the flow function could be

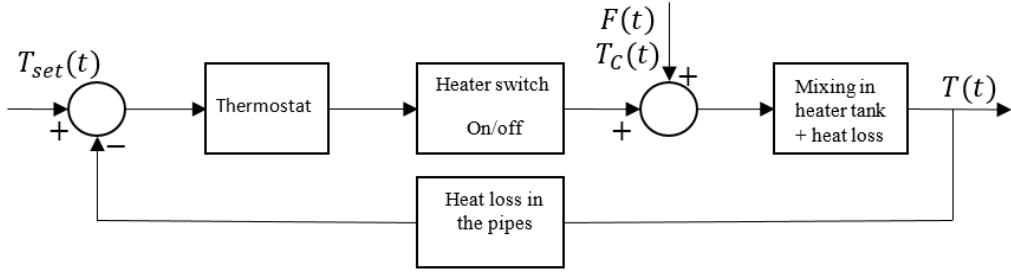


Figure 4. General control loop scheme for water heater system

described as

$$(8) \quad F(t) = \sum_i f_i \chi(t; t_i, t_{i+1}),$$

where

$$(9) \quad \chi(t; t_i, t_{i+1}) = \begin{cases} 1 & t \in [t_i, t_{i+1}] \\ 0 & \text{otherwise} \end{cases},$$

and $f_i = r_i F$ with $r_i = \{1, 0\}$ and $F = 2$ liter/min.

Thus the overall heat balance around the system reads (similarly to [1]–[4])

$$(10) \quad \frac{dE(t)}{dt} = \frac{Mc_p d(T_h(t) - T_c)}{dt} = P(t) + F(t)c_p(T_h(t) - T_c) + \frac{(T_h(t) - T_a)A}{R},$$

with $E(t)$ being the internal energy, T_a room temperature, t_d time delay due to the heating of the water and R thermal resistance of the tank.

To approach this problem, we chunk the time domain into finite small intervals. The quality of the solution clearly depends on resolution of the mesh. Due to Eqs (7) to (9), since at each time interval the coefficients in Eq. (10) are constant, Eq. (10) turns into

$$(11) \quad \frac{dy(t)}{dt} + a_i y(t) = b_i, \quad t_i \leq t < t_{i+1}$$

which has analytical solution in the form

$$(12) \quad y(t) = \xi_i e^{-a_i t} + \frac{b_i}{a_i}, \quad t_i \leq t < t_{i+1},$$

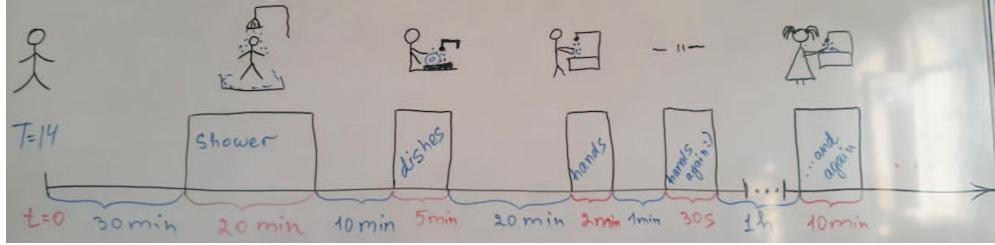


Figure 5. A sample usage of water in a day

where ξ_i depends on the solution at t_i from the previous time step (i.e., $t_i \leq t < t_{i+1}$).

As an example we assumed a consumption schedule for a person in a day time as in Fig. 5. We took the sensitivity of the thermostat to be 3°C . Fig. 6 illustrates the history of the electric power and the temperature profile for the outlet flow from $t = 30$ min onwards.

Finally, we need to take into account for the heat loss from the pipes that occurs between the heater and the sensors. According to problem description, the temperature that is measured by the sensor is on the pipe. Therefore, there is a convection heat transfer from the fluid inside the pipe to the inner wall of the pipe which is followed by a conduction heat transfer from there to the outside surface of the pipe. Finally, the outer surface is cooled down by Newton cooling law through the cold surrounding air. The heat balance for the two processes are expressed as

$$(13) \quad h_{water} A_{inner} (T_{water} - T_1) = \frac{k}{d} (T_1 - T_2),$$

$$(14) \quad h_{air} A_{outer} (T_2 - T_{air}) = \frac{k}{d} (T_1 - T_2),$$

where d is the thickness of the tube and T_1 and T_2 are the temperatures of the inner and outer surface of the pipe, respectively. Using the proper parameters for h_{water} and h_{air} and properties of typical pipes used in houses (e.g. thickness=2 mm, diameter=1 inch) we found a decrease of maximum 3°C . An improved model would be to plug Eqs (13) and (14) into Eq. (10) so that the heat loss plays its role in the dynamics of the tank.

4. Conclusions

During the week of ESGI 120 we developed and solved the model for temperature dynamics the residential electrical water heater. Two proposed approaches

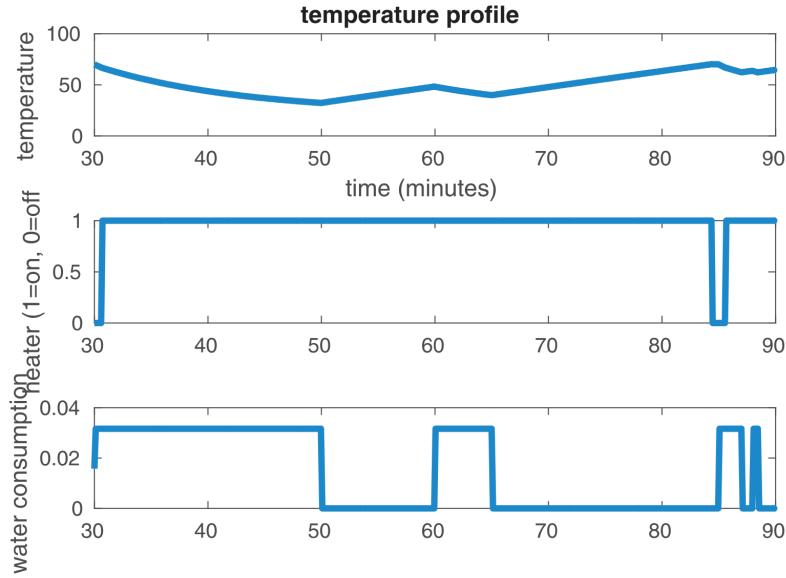


Figure 6. Hot water usage, power and temperature history for toy problem from Fig. 5

were covered. First, we constructed a 1-D spacial model of the temperature distribution inside the water tank with the heater located at the bottom of it. The 1-D approximation allowed us to avoid heavy numerical simulations, however, it has couple of limitations such as the location of the heater is strictly at the bottom of the tank and imposed fluid velocity that does not satisfy the continuity equation. The later limitation can be eliminated if one considers 2-D model. The results and parameter estimation suggest domination of the convective effect with respect to diffusion, and therefore, suitable choice of the velocity profile is crucial for the model.

The second proposed model described the temperature dynamics in the tank. This model fits better to describe the experimental data, however, also has its limitations, such as an assumption of uniform temperature inside the tank which results into neglecting the time which is required for the water on the top of the tank to heat up.

Both models can be integrated into a single complete model, and it is the proposed step for the further research.

Acknowledgements

We would like to thank all the organizers who contributed for this event and gave us the opportunity to be part of ESGI120 in Sofia. We are very grateful to problem presenter, Mr. Kristiyan Boyanov, for his support during the study group.

Last but not least, we would like to express our gratitude to Dr. Wil Schilders from Eindhoven University for his valuable comments that guided us through the problem.

References

- [1] H. A. Suthar, J. J. Gadit, Modelling and Analysis of the Simple Water Heater System, International Journal of Electrical and Computer Engineering 1 (1), 2011, 49–52.
- [2] K. I. Elamari, Using Electric Water Heaters (EWHs) For Power Balancing and Frequency Control in PV-Diesel Hybrid Mini-Grids, M.Sc. Thesis, Concordia University, 2011.
- [3] J. Vieira, A. Mota, Passing From a Gas to an Electric Water Heater System: Adaptable PID Versus Smith Predictive Control, Frontiers of Model Predictive Control (Intech open), 2012.
- [4] M. Shaad, A. Momeni, C. P. Diduch, M. Kaye, L. Chang, Parameter Identification of Thermal Models For Domestic Electric Water Heaters in a Direct Load Control Program, 25th IEEE Canadian Conference on Electrical and Computer Engineering, 2012.
- [5] Kundu, P., and I. Cohen. “Fluid Mechanics”, 4th Edition, Academic Press, 2008.