BW-CAR | SINCOM
SYMPOSIUM ON INFORMATION
AND COMMUNICATION SYSTEMS

3rd Baden-Württemberg Center of Applied Research
Symposium on
Information and Communication Systems

SInCom 2016

Franz Quint, Dirk Benyoucef (Eds.)

Karlsruhe, December 2nd, 2016

ISBN 978-3-943301-21-2

Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES
Näher dran.

ISBN: 978-3-943301-21-2

 Hochschule Offenburg
Badstraße 24
77652 Offenburg
info@hs-offenburg.de
www.hs-offenburg.de

Message from the Program Chairs

The Baden-Württemberg Center of Applied Research (BW-CAR) intends to further develop applied research at the Universities of Applied Science (UAS). The BW-CAR working group *Informations- und Kommunikationssysteme* (IKS) organizes in cooperation with the working group *Technologien für Intelligente Systeme* (iTIS) the BW-CAR Symposium on Information and Communication Systems (SInCom). This year it took place in its third edition at Karlsruhe University of Applied Sciences. IKS and its members, professors at universities of applied sciences in Baden-Württemberg, cover the whole area of communication and information systems. Retrieval, processing, transmission and storage of information are key technologies in the digital age, with impact on all areas of modern life. They are part of industry 4.0, mobile networks, smart grids, navigation systems, ambient assisted living, environmental engineering, in macroscopic or in embedded systems such as smart phones and sensor networks.

SInCom 2016 aimed at young researchers for contributions in the fields of

- Algorithms for Signal and Image Processing
- Communication Networks and Information Theory
- Pattern Recognition
- Control Theory
- Distributed Computing

The program committee thanks all authors for their valuable contributions to SInCom 2016. Furthermore, we express our great acknowledgment to the reviewers for their suggestions and help to improve the papers. Finally, we appreciate very much the support of the rectorate and the help of many employees of Karlsruhe University of Applied Sciences in organizing the symposium.

Karlsruhe, December 2nd, 2016

Franz Quint, Dirk Benyoucef

Organizing Committee

Prof. Dr. Franz Quint, Hochschule Karlsruhe

Prof. Dr. Dirk Benyoucef, Hochschule Furtwangen

Program Committee

Prof. Dr. Dirk Benyoucef, Hochschule Furtwangen
Prof. Dr.-Ing. Andreas Christ, Hochschule Offenburg
Prof. Dr. rer. nat. Thomas Eppler, Hochschule Albstadt-Sigmaringen
Prof. Dr. Matthias Franz, Hochschule Konstanz
Prof. Dr.-Ing. Jürgen Freudenberger, Hochschule Konstanz
Prof. Dr. Thomas Greiner, Hochschule Pforzheim
Prof. Dr. rer.nat. Roland Münzer, Hochschule Ulm
Prof. Dr.-Ing. Franz Quint, Hochschule Karlsruhe
Prof. Dr. Christoph Reich, Hochschule Furtwangen
Prof. Dr. Georg Umlauf, Hochschule Konstanz
Prof. Dr.-Ing. Axel Sikora, Hochschule Offenburg
Prof. Dr. Dirk Westhoff, Hochschule Offenburg

Content

Frequency Invariant Transformation of Periodic Signals (FIT-PS) for high frequency Signal Representation in NILM

Pirmin Held[1], Alaa Saleh[1], Djaffar Ould Abdeslam[2], and Dirk Benyoucef[1]

[1]Furtwangen University, Furtwangen, Germany
Email: {pirmin.held,alaa.saleh,dirk.benyoucef}@hs-furtwangen.de

[2]Université de Haute-Alsace, Mulhouse, France
Email: djaffar.ould-abdeslam@uha.fr

***Abstract*—The aim of non-intrusive load monitoring is to determine the individual energy consumption of different devices on the basis of the total energy consumption. The individual energy consumption of a device is measured at a central point without the need of individual measuring instruments on the devices themselves. In this work we present a signal representation frequency invariant transformation of periodic signals (FIT-PS) [1] on high sampled signals.**

We present a new method for signal separation, decomposition of the signal into individual states and feature extraction. Frequency invariant transformation of periodic signals (FIT-PS) is based on a signal diagram similar to the concept of trajectories [2]–[4] utilizing the periodicity of voltage and current, and their correlation. This approach breaks down the current signal into its individual periods using the voltage as a reference signal for the determination of trigger points. Thereby, the phase information between current and voltage is maintained and is inherently part of the new signal representation. In common approaches with high sample rates several signal forms must be combined to achieve good results. The advantage of this method is that the information contained in the signal is preserved entirely. Hence, this single signal representation is sufficient to create similar or even better results by using high sample rates. The efficiency of the new signal representation and signal separation is demonstrated by the example of an event detection algorithm. For testing the BLUED dataset [5] was used reaching a sensitivity in event detection of 99.45%.

I. Introduction

The disaggregation of electrical energy consumption (non-intrusive load monitoring (NILM)) determines the energy consumption of individual electrical appliances from the total energy consumption [6]. In NILM the quantities $v(t)$ and $i(t)$ are measured. They can be described as a function of the amplitude $V(t)$ or $I(t)$ and the frequency $f(t)$, respectively.

$$v(t) = V(t) \cdot \cos(2\pi f(t) \cdot t + \varphi_i(t)) \quad (1)$$

$$i(t) = I(t) \cdot \cos(2\pi f(t) \cdot t + \varphi_v(t)) \quad (2)$$

Equation (3) according to [7] describes the fundamental problem of modeling NILM.

$$y(t) = \sum_{d=1}^{D(t)} y_d(t) \qquad \text{for} \qquad t = 1, ..., T \quad (3)$$

$y(t) \in \mathbb{R}$ describes the sum of power consumption of several devices d at a time t. Where $y_d(t)$ is the power consumption of the device d at time t and $D(t)$ is the number of devices in the building. In general, neither the number of devices nor their characteristics are known and thus a distinct solution does not exist.

Most of the methods in literature use a linear process for disaggregation which are described with the following four steps: event detection, feature extraction, classification and tracking [8]. The event detection determines the corresponding switching points of the on and off times of the individual devices. The knowledge of the switching points allows a further distinction of the signal state: There are stationary states $z_i, z_{i+1}, \ldots$ and between each two stationary states there is a transient state.

In the early days of NILM [6] the differential power ΔP and ΔQ are used. Nowadays there are numerous features like harmonics, switching transient waveform, current waveform or eigenvalues [10], [11]. This features can be distinguished in steady state features and transient state features. While, steady state features utilize the signal section before and after the turning-on or switching-off procedure, transient state features only make use of the signal sequence during the turning-on or switching-off process. [12] proposes a procedure in which both features are used in combination. The classification, or distinction of different devices, is based on the results provided by the event detection and the feature extraction [13]. Therefor, each event which is not identified leads to a reduction of the recognition rate and each false positive event represents an error source for the classification.

The energy - tracking, which assigns the energy consumption to the individual devices is the last step of the NILM.

An important question for a NILM system is the choice of the sampling rate. In literature the sampling rate ranges from a few Hz [6], [14]–[17] to several kHz [18], [19]. The number of features that can be defined increases with a higher sampling rate but also the calculation effort and the costs for the whole system increase. The optimal feature for event detection and classification depends on the kind and number of devices, as well as on their switching behavior. In [10]

different combinations of features are used to produce better results but it is hard to find a combination of features that suits for all devices. The success of a NILM system depends highly on the individual condition. Therefore, there is the possibility to provide a variety of features in the hardware (HW) or to customize the features by HW adjustment for the individual equipment pool. In addition, the weighting of the features should be adjusted depending on the application environment.

Frequency invariant transformation of periodic signals (FIT-PS) is a new method of signal decomposition which is independent of the main frequency and which results in a signal representation without periodic oscillations. The current is separated into its individual periods using the voltage as a reference for the beginning of each period and the determination of the subsequent sampling points in each period.

In contrast to current waveform (CW) the voltage is used as a trigger point for the determination of the periods. Thereby, important information, such as active and reactive power for example, remains. The decomposition results in a multi-dimensional signal where all information, including low and high frequencies, and the phase angle are preserved. Furthermore, easy separation of steady states and transient states is enabled.

The generated set of periods creates the feature space. Since all information is preserved in this feature space, its solitary application is sufficient. In this paper the versatility of the introduced signal decomposition method is demonstrated at the example of the event detection.

This paper is structured as follows: Section II introduces the new signal representation FIT-PS. Section III presents an event detection method which is optimized by using the new signal representation. In section IV, the performance of an event detection exploiting the new signal representation is shown. The BLUED dataset [5] is used for the simulations. Additionally, a measurement method is presented that functions without converting data afterwards.

II. Signal Representation by FIT-PS

This section describes the new feature space based on $v(t)$ (1) and $i(t)$ (2). First, the sampling resulting in the new signal representation is described. Next, the structure of the new signal representation will be explained. Since the frequency of the power grid is not constantly 50 Hz or 60 Hz a mains frequency invariant representation of the current signal is necessary. This is realized using the permanently available voltage signal.

FIT-PS splits the current signal $i(t)$ into the individual periods whereby the voltage signal $v(t)$ is used as a synchronization signal. This is shown in the first three plots of Fig. 1. Thus, the information of the phase angle between current and voltage is preserved. This results in a $n_l \times n_k$ dimensional feature space where n_l is the number of periods and n_k is the number of sampling points in each period as it can be seen in the last plot of Fig. 1.

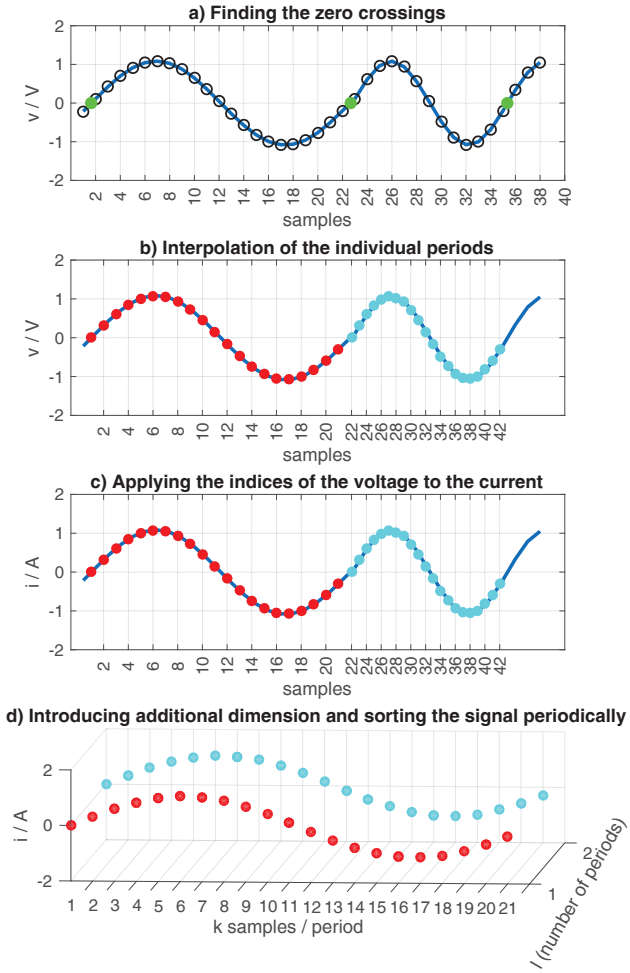


Fig. 1. Graphic illustration of FIT-PS

The value of the signal $i(t)$ is calculated by applying Eq. (4). In Eq. (4) l numbers the periods of the signal $i(t)$ and k is the sample number within one period of the signal $i(t)$.

$$\begin{aligned} &\mathbb{N}, \mathbb{N} \to \mathbb{R} \\ &l, k \mapsto i(\underbrace{(l-1) \cdot T_g}_{\text{period}} + \underbrace{T_s \cdot k}_{\text{inside one period}}) \end{aligned} \tag{4}$$

T_g is the length of one period in seconds and T_s is the time in seconds between two measurement points.

In Fig. 2 the FIT-PS representation of a $i(t)$ is shown as a colored surface. Here 600 periods are considered. Those 600 periods contain two different steady states and a transient state. Initially Fig. 2 shows the steady state z_i of device 1. After the 300th period, device 2 is connected. Following an amplitude change that is clearly apparent in all dimensions, the system is in a transient state. Subsequently, the signal again reaches a stationary state, the second steady state z_{i+1}. Both steady states have their minimum ($k = 148$) and their maximum

($k = 48$) in the same dimension, but the waveform itself has changed considerably from the steady state z_i to z_{i+1}. At steady state z_i, after the maximum in dimension $k = 48$, the amplitude decreases quickly below zero (from the third dimension on). Whereas at steady state z_{i+1}, the value does not reach a negative amplitude until the dimension $k = 100$. The figure shows that the changes in each period l within a steady state are very small.

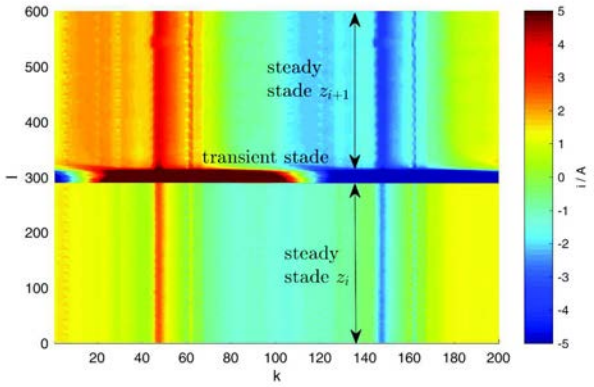


Fig. 2. FIT-PS of a current signal. Steady state z_i: device 1; Transient state: switching on procedure of device 2; Steady state z_{i+1}: device 1 and 2

A. The influence of the sampling rate

In Fig. 3 the influence of different sampling rate is shown. Each figure shows the same part of the signal, where a device is switched on and transformed with FIT-PS, but with different sampling frequencies. l is the number of periods and k is the number of dimensions which change with different sampling rates. With 1.5 kHz far fewer details can be seen. Due to the low-pass characteristic during down sampling briefly occurring peaks are smoothed. Therefore, the amplitude also changes here, which can be seen particularly clearly at 1.5kHz.

III. Event detection

We increase the sampling rate to 12 kHz (200 samples per period), compared to [1] where a sampling rate of 1.2 kHz (20 samples per period) was used. Using FIT-PS allows a simple method with low complexity for detecting switching events.

To reduce the influence of disturbances a low-pass filter (5) is applied in each dimension k of the signal $i(t)$. With the low-pass filter (5) we get I_{TP} (6)

$$h_{TP}(l) = \sum_{g=0}^{M} b_g \delta[n-g] = \begin{cases} b_n, & 0 \le n \le M \\ 0, & else \end{cases} \tag{5}$$

$$I_{TP}(l,k) = \Big(h_{TP} * I\Big)\Big(l,k\Big) \quad \forall \quad k \tag{6}$$

where M is the length of the low-pass filter and b_n are the filter coefficients. Usually, in the NILM context multiple devices need to be detected. This creates a large number of combinatorial possibilities which are displayed in the feature space. For this reason it is useful to consider only the derivative

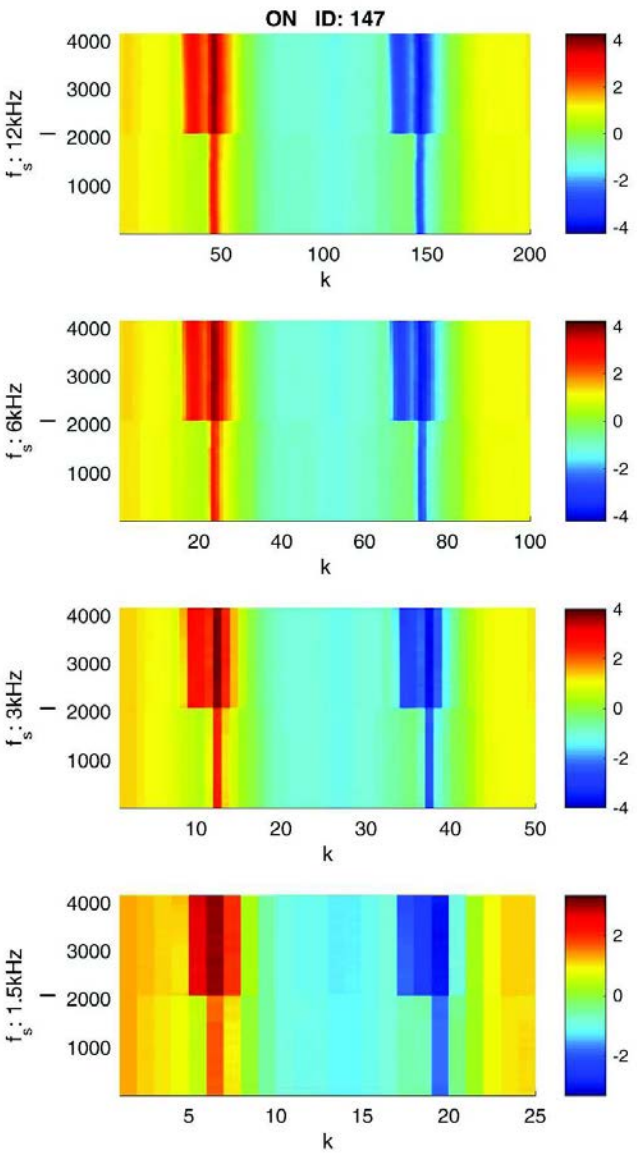


Fig. 3. FIT-PS with different sampling rate

Eq. (9) and (10) in order to utilize the feature space more effectively. The case that two devices change their state at exactly the same time is excluded.

Using this event detection algorithm directly on higher sample frequencies will produce significantly more false positives. The reason for this is device specific current peaks of some devices. These current peaks occur only in individual dimensions of the high-sampled signal marked with arrows in Fig. 4. In the case of signals which were recorded at a lower frequency (1.2 kHz), these peaks can not be seen because of the more marked low-pass characteristic. To use FIT-PS with higher sampling rates, the event detection algorithm has to be modified.

Also because of the increased number of dimensions k

caused from the higher sample rate, a dimension reduction using the principal component analysis (PCA) is included. PCA gives the k-dimensional projection matrix $(I_{PCA})_{l\times k}$ where $k << l$.

$$I_{PCA} = I_{TP}W \tag{7}$$

where W is a k by k matrix whose columns are the eigenvectors of $I_{TP}^{\top}I_{TP}$.

Because there are obviously correlations between the single dimensions (in I_{TP}) not all dimensions are needed after the PCA. We are using only the first m loading vectors, so we get only the first m principal components.

$$I_{PCAm} = I_{TP}W_m \tag{8}$$

This also reduces the problem of noise in some dimensions as discussed in Fig. 4 and 5.

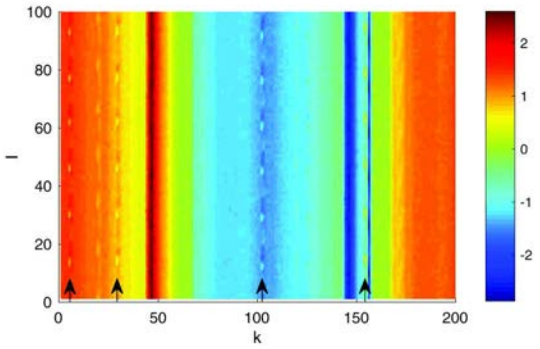


Fig. 4. current peaks at high sampled signal transformed with FIT-PS

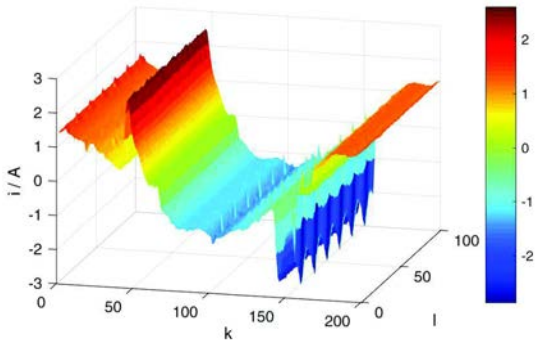


Fig. 5. 3D view of current peaks at high sampled signal transformed with FIT-PS

In order to reduce the computational effort and to decrease the complexity, the event detector uses a two-step process. In the first stage each possible event is detected with a simple threshold which is applied to the derivate of the signal. This threshold is adjusted in order to detect all true positive (TP) events. This leads to higher numbers of false positive (FP) events. The signal ranges where potential events are identified are examined more closely in the second stage. In the second stage, the difference between the stationary level before and after the event is calculated and compared with a second threshold.

The derivative of I_{PCAm} in direction of l is calculated using Eq. (9) and (10). Φ_l shows the index where the derivative has its maximum and is above the threshold Tr_1.

$$L := \left\{ l \quad \middle| \quad \left|\frac{\partial I_{PCAm}(l,m)}{\partial l}\right| \geq Tr_1, \quad \forall \quad k \right\} \tag{9}$$

$$\Phi_l = \underset{k}{\operatorname{argmax}} \left|\frac{\partial I_{PCAm}(l,m)}{\partial l}\right| \geq Tr_1 \quad \forall \quad l \in L \tag{10}$$

The advantage of using the maximum compared to the average is that devices using a phase angle control can be detected more efficiently because of the better signal-to-noise ratio (SNR). These devices only use a part of the period where the information is only conserved in a few dimensions. The problem in event detection is the suppression of multiple detections of the same event while maintaining the ability to differentiate between events which are located very closely to each other. Hence, a variable dead-time in which no additional event is detected is introduced. The disadvantage of a dead-time is that events which are too close to each other cannot be recognized. In order to avoid losing events, the time range in which no other event can be detected is set so that it starts and terminates with the beginning and ending of each transient state. For this purpose it is necessary to decompose the signal into stationary and transient sections.

In Eq. (11) and (12) the start and end index, Φ_l^s and Φ_l^e, of the transient states are calculated with respect to a fixed maximum length α. The distances $\Phi_l - \Phi_l^s$ and $\Phi_l^e - \Phi_l$, respectively, depend on the specific devices.

$$\Phi_l^s = \underset{\alpha}{\operatorname{argmin}} \operatorname{Var}(I_{PCAm}(\alpha, \Phi_l)) \tag{11}$$
$$\text{with} \quad \alpha = l, ...l + N_A \quad \forall \quad l \in L$$

$$\Phi_l^e = \underset{\alpha}{\operatorname{argmin}} \operatorname{Var}(I_{PCAm}(\alpha, \Phi_l)) \tag{12}$$
$$\text{with} \quad \alpha = l - N_B, ...l \quad \forall \quad l \in L$$

Where N_A and N_B are constants.
After determining the exact position of the transient section, a second threshold Tr_2 is used in (14) to reduce the number of false positives. Here, the information of the steady state before and after the detected event is used. As shown in (9) and (10) a threshold is applied to the dimension with

$$\tau_l = \sum_{n=\Phi_l^s-(M-1)}^{\Phi_l^s} \frac{I_{PCAm}(n,\Phi_l)}{M} - \sum_{n=\Phi_l^e}^{\Phi_l^e+(M-1)} \frac{I_{PCAm}(n,\Phi_l)}{M} \quad \forall \quad l \in L \tag{13}$$

over the mean of M values of the steady state before and after the event. In equation (14) we get the final result, the indices of the events E.

$$E = |\tau_l| \geq Tr_2 \quad \forall \quad l \in L \tag{14}$$

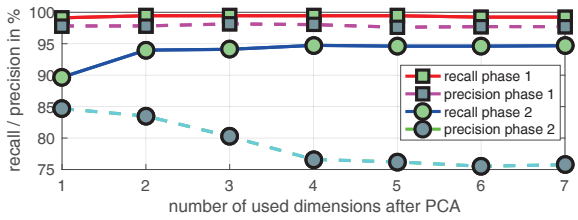


Fig. 6. Comparison of the results of the event detection with different numbers of dimensions after PCA

IV. Results

Applying the proposed feature space and the event detector on the BLUED dataset, [5] allows a better comparability. In difference to [1] where a sample frequency of 1.2 kHz was used, a 12 kHz frequency was used for this work.

Due to the varying net frequency, measurement points also have to vary during different periods. But for trajectories as well as for our new approach constant points are required. As first trigger point in a period, we decided to use the zero crossing from negative to positive. The remaining 19 trigger points were regularly distributed over this period. Signals provided by the BLUED dataset [5] were used and converted using voltage as reference.

Performance metrics (15) and (16) depend on TP, false negative (FN) and FP and were used to receive a better comparability.

$$P_{recall} = \frac{TP}{TP + FN} \tag{15}$$

$$P_{precision} = \frac{TP}{TP + FP} \tag{16}$$

Tab. I and II show the results of the event detection based on FIT-PS (sample frequency of 1.2 kHz and 57 Hz) in comparison to [20], where a modified generalize likelihood ratio detector combined with a higher sampling rate was used. The performance of all event detectors depends highly on each phase considered.

TABLE I
Detection performance BLUED Phase A

BLUED (A)	FIT-PS 12 kHz PCA	FIT-PS 1.2 kHz [1]	[20]
P_{recall}	99.45%	99.31%	98.16%
$P_{precision}$	98.15%	97.51%	97.94%

TABLE II
Detection performance BLUED Phase B

BLUED (B)	FIT-PS 12 kHz PCA	FIT-PS 1.2 kHz [1]	[20]
P_{recall}	93.98%	87.37%	70.40%
$P_{precision}$	83.50%	82.08%	87.29%

Concerning the sensitivity (P_{recall}) FIT-PS with 12 kHz at phase A leads to better results compared to the method presented in [20] without significant loss of precision. In contrast, the results of FIT-PS 1.2 Hz are slightly behind [20] and FIT-PS with 12 kHz. The biggest advantage of FIT-PS was shown when phase B was used. Even at the lower sample rate, the sensitivity of FIT-PS outperforms the event detector used in [20] significantly, with only minimal lower $P_{precision}$ than [20]. Due to the dramatically reduced amount of data and the subsequently difficult noise filtering, FIT-PS 1.2 kHz shows reduced sensitivity and precision if compared to the other FIT-PS.

Because a lot of electronic and appliances overlap at phase B [5], the results for phase A were significantly better.

V. Conclusion

We present the frequency invariant transformation of periodic signals FIT-PS for higher sample rates. We could show that the use of higher sample rate entail better results.

The voltage is used as reference signal for the determination of trigger points. With this information the current signal is interpolated and fragmented into its individual periods. Thereby the signal representation is independent from the mains frequency. The entire information of the original signal is retained.

Because of the highly increased number of dimensions (20 with 1.2 kHz to 200 with 12 kHz) the PCA is used to reduce dimensions. Depending on the used thresholds T_{r1} and T_{r2} a reduction to two or three dimensions leads to the best results. The reason for this is the following event detection method. Since the used event detection method is not complex enough to completely take account of all available information, this information has a disturbing effect on the event detector. PCA reduces the number of information to an amount which can be processed by the event detector.

For event detection this method was applied to the BLUED dataset [5] resulting in a sensitivity of up to 99.45%.

In future, we plan to apply FIT-PS for a classification in NILM. The issue of peaks in some dimensions caused by a higher sampling rate for the event detector can thereby used as additional feature.

This work was created as part of the iMon project (funding number 03FH001IX4).

REFERENCES

[1] P. Held, F. Laasch, A. Djaffar Ould, and D. Benyoucef, "Frequency invariant transformation of periodic signals (FIT-PS) for signal representation in NILM," no. 42, accepted for publication but not yet published.

[2] L. Du, D. He, R. G. Harley, and T. G. Habetler, "Electric load classification by binary voltage–current trajectory mapping," vol. 7, no. 1, pp. 358–365, 00000. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7130652

[3] T. Guzel and E. Ustunel, "Principal components null space analysis based non-intrusive load monitoring," in Electrical Power and Energy Conference (EPEC), 2015 IEEE. IEEE, pp. 420–423, 00000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7379987

[4] T. Hassan, F. Javed, and N. Arshad, "An empirical investigation of VI trajectory based load signatures for non-intrusive load monitoring," vol. 5, no. 2, pp. 870–878, 00011. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6575197

[5] K. Anderson, A. Ocneanu, D. Benitez, A. Rowe, and M. Berges, "BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research," in Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD).

[6] G. W. Hart, "Nonintrusive appliance load monitoring," vol. 80, no. 12, pp. 1870–1891, 00954. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=192069

[7] R. Dong, L. Ratliff, H. Ohlsson, and S. S. Sastry, "Fundamental limits of nonintrusive load monitoring," in Proceedings of the 3rd international conference on High confidence networked systems. ACM, pp. 11–18, 00014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2566471

[8] L. Jiang, S. Luo, and J. Li, "Automatic power load event detection and appliance classification based on power harmonic features in nonintrusive appliance load monitoring," in Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on. IEEE, pp. 1083–1088, 00004. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6566528

[9] B. Wild, K. S. Barsim, and B. Yang, "A new unsupervised event detector for non-intrusive load monitoring," in 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP). IEEE, pp. 73–77, 00000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7418159

[10] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng, "Load signature study—part II: Disaggregation framework, simulation, and applications," vol. 25, no. 2, pp. 561–569, 00083. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5337970

[11] ——, "Load signature study—part i: Basic concept, structure, and methodology," vol. 25, no. 2, pp. 551–560, 00195. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5337912

[12] H.-H. Chang, C.-L. Lin, and J.-K. Lee, "Load identification in nonintrusive load monitoring using steady-state and turn-on transient energy algorithms," in Computer supported cooperative work in design (cscwd), 2010 14th international conference on. IEEE, pp. 27–32, 00050. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5472008

[13] Y. F. Wong, Y. Ahmet Sekercioglu, T. Drummond, and V. S. Wong, "Recent approaches to non-intrusive load monitoring techniques in residential settings," in Computational Intelligence Applications In Smart Grid (CIASG), 2013 IEEE Symposium on. IEEE, pp. 73–79, 00012. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6611501

[14] Z. Zhang, J. H. Son, Y. Li, M. Trayer, Z. Pi, D. Y. Hwang, and J. K. Moon, "Training-free non-intrusive load monitoring of electric vehicle charging with low sampling rate," in Industrial Electronics Society, IECON 2014-40th Annual Conference of the IEEE. IEEE, pp. 5419–5425, 00004. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7049328

[15] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, "Smart*: An open data set and tools for enabling research in sustainable homes," vol. 111, p. 112, 00000. [Online]. Available: http://wan.poly.edu/KDD2012/forms/workshop/SustKDD12/doc/SustKDD12_3.pdf

[16] L. Mauch and B. Yang, "A new approach for supervised power disaggregation by using a deep recurrent LSTM network," in 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP). IEEE, pp. 63–67, 00000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7418157

[17] J. Liao, G. Elafoudi, L. Stankovic, and V. Stankovic, "Non-intrusive appliance load monitoring using low-resolution smart meter data," in Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on. IEEE, pp. 535–540, 00011. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7007702

[18] C. Duarte, P. Delmar, K. Barner, and K. Goossen, "A signal acquisition system for non-intrusive load monitoring of residential electrical loads based on switching transient voltages," in Power Systems Conference (PSC), 2015 Clemson University. IEEE, pp. 1–6, 00001. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7101707

[19] R. Jonetzko, M. Detzler, K.-U. Gollmer, A. Guldner, M. Huber, R. Michels, and S. Naumann, "High frequency non-intrusive electric device detection and diagnosis," in Smart Cities and Green ICT Systems (SMARTGREENS), 2015 International Conference on. IEEE, pp. 1–8, 00001. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7297979

[20] K. D. Anderson, M. E. Bergés, A. Ocneanu, D. Benitez, and J. M. Moura, "Event detection for non intrusive load monitoring," in IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society. IEEE, pp. 3312–3317, 00032. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6389367

Chase decoding for quantized reliability information with applications to flash memories

Jürgen Freudenberger, Mohammed Rajab

Institute for System Dynamics
HTWG Konstanz, University of Applied Sciences, Germany
Email: {jfreuden,mrajab}@htwg-konstanz.de
Web: www.isd.htwg-konstanz.de

***Abstract*—Chase decoding is an established soft-input decoding method for algebraic error correcting codes. This paper analyses the performance of Chase type II decoding using quantized input data, where transmission of binary data over the additive white Gaussian noise (AWGN) channel is assumed. The channel output symbols are quantized with a small number of decision thresholds. This channel model is applicable for data storage in flash memories. Simulation results demonstrate that the soft decoding performance of the Chase algorithm can be improved by optimizing the threshold values for the quantization.**

I. Introduction

Flash memories are becoming more and more important for non-volatile mass storages, where a flash memory stores the information in floating gates which can be charged and erased. These floating gates keep their electrical charge without a power supply. However, information may be read erroneously. The error probability depends on the storage density. The NAND Flash used different type of levels as single-level-cell (SLC), and currently the devices used multiple levels and are mentioned to as multiple-level cell (MLC) flash or triple-level cell (TLC) and on the number of program and write cycles [1].

In flash memories, error correction coding (ECC) is required in order to ensure integrity and reliability [2], [3]. Soft-input decoding can improve the error correcting capability compared to hard-input decoding, where the soft-input decoding is based on reliability information from the channel [4], [5]. To obtain reliability information the medium must be read several times using different read threshold voltages. Typically only a small number of reads is used resulting in reliability information with coarse quantization.

Chase decoding algorithms are reliability based decoding procedures that generate a list of candidate codeword by flipping bits in the received word [6], [7]. The test patterns for the bit flipping are based on the least reliable positions of the received word. For each test pattern, algebraic hard-input decoding is employed. Finally, the best candidate from the list is obtained by minimizing the Euclidean distance between the candidate codewords and the received word. Chase devised three different algorithms. The main difference between the algorithms is the number of test patterns. The complexity of Chase decoding depends on the size of the list of the candidates. In this paper, we investigated Chase type II decoding for quantized reliability information. In particular, we optimize the read threshold in order to improve the decoding performance with quantized reliability information.

This paper is structured as follows. Section II introduces the threshold voltage of flash memories and the channel model. In Section III the Chase decoding algorithm is described. Simulation results are presented in Section IV.

II. Threshold values

With flash memories, the cells are addressed with so-called word-lines, where a threshold voltage is required to turn on a particular transistor. The value of the threshold voltage varies from cell to cell. The probability density function of the variation of threshold voltages is usually modelled by a Gaussian distribution. Hence, the channel model for flash cells is equivalent to an additive white Gaussian noise (AWGN) channel. However, in order to obtain reliability information for the channel input values, multiple reads with different threshold values are required. The reading procedure for reliability information causes additional latency. Thus, only a small number of thresholds are applied in practice. Assuming i.i.d. Gaussian threshold voltages, the channel of a flash memory with reliability information can be considered as an AWGN channel with quantized channels values.

In the following, we assume a reading procedure with five threshold voltages. Fig. 1 shows the probability distribution of the threshold values, where the reading threshold voltages are denoted by δ_1 and δ_2. Hence, for the area between 0 and $-\delta_1$ or $-\delta_1$ corresponds to the most unreliable input values. We assume that the flash channel can be modelled as quantized AWGN channel with the following quantization function

$$\tilde{r}_i = \begin{cases} 1 & , r_i > \delta_2 \\ \delta_2 & , \delta_1 < r_i \leq \delta_2 \\ \delta_1 & , 0 \leq r_i \leq \delta_1 \\ -\delta_1 & , 0 > r_i \geq -\delta_1 \\ -\delta_2 & , -\delta_1 > r_i \geq -\delta_2 \\ -1 & , r_i < -\delta_2 \end{cases} \tag{1}$$

where r_i denotes the i-th channel value and $\tilde{r}_i$ the corresponding quantized channel value. In this work, we optimize the values δ_1 and δ_2 for Chase decoding.

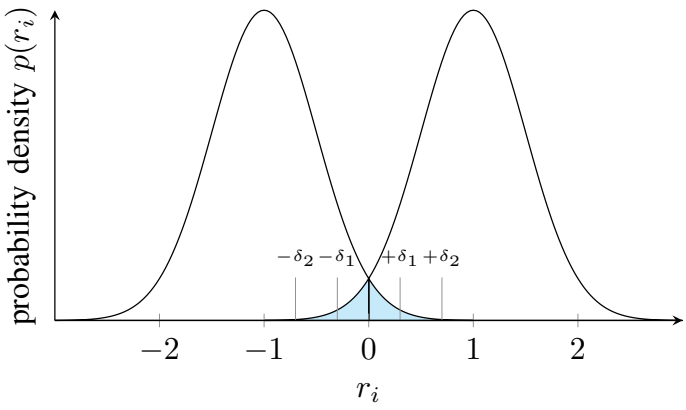


Fig. 1. Probability density function with reading thresholds δ_1 and δ_2.

III. Chase decoding algorithm

The Chase algorithm is a multi-trial procedure where bit flipping is applied to the least reliable received values. Then algebraic decoding is applied to determine a valid codeword. Using different test patterns for the bit flipping, a list of candidate codewords is obtained. Finally, the most likely codeword is selected from this list. For the AWGN channel, the most likely codeword can be determined by minimizing the Euclidean distance between the received word and the codewords in the list of candidates. With Chase type II decoding a list of 2^m candidates is obtained by systematically testing all combinations of the m least reliable positions within the received codeword. Typically, the parameter m is chosen as $m = \frac{d_min}{2}$, where d_{min} is the minimum Hamming distance of the code. Note that Chase type II decoding is a suboptimal decoding procedure, because the maximum likelihood codeword is not always in the list of candidates obtained by bit flipping.

With Chase decoding, typically unquantized channel values are assumed. In this work, we investigate Chase decoding with quantized channel values. Note that with quantized channel inputs, the set of the $\frac{d_{min}}{2}$ least reliable positions is not always unique. Moreover, minimization of the Euclidean distance among the list of candidates does not always result in a unique solution. We propose some modifications to the Chase algorithms taking the quantization into consideration.

We consider binary codes of length n. The received sequence is denoted by $\mathbf{r} = (r_1, r_2, ..., r_n)$. $\mathbf{z} = (z_1, z_2, ..., z_n)$ is the binary hard-decision sequence corresponding to the received sequence $\mathbf{r}$. If $\mathbf{z}$ is a valid codeword, then it is the the maximum likelihood codeword [5]. Thus, we calculate the syndrome value for $\mathbf{z}$ and apply the Chase decoding only for non-zero syndrome values as shown in Fig. 2. Moreover, we choose $m = \frac{d_{min}}{2} + 1$, because the set of the $\frac{d_{min}}{2}$ least reliable positions can not always be determined uniquely.

Finally, we propose a decoding procedure that may declare a decoding failure. With quantized input values, Chase decoding does not always result in a unique codeword, i.e., there might be two or more codewords in the list of candidates which have the same Euclidean distance to the received word. In this case, the probability of a decoding error is high, because the probability of selecting the correct codeword is at most a half. If the decoding procedure is used in a concatenated coding scheme, the decoder may declare a failure. Such a decoding failure can be exploited using error and erasure decoding in the next decoding stage.

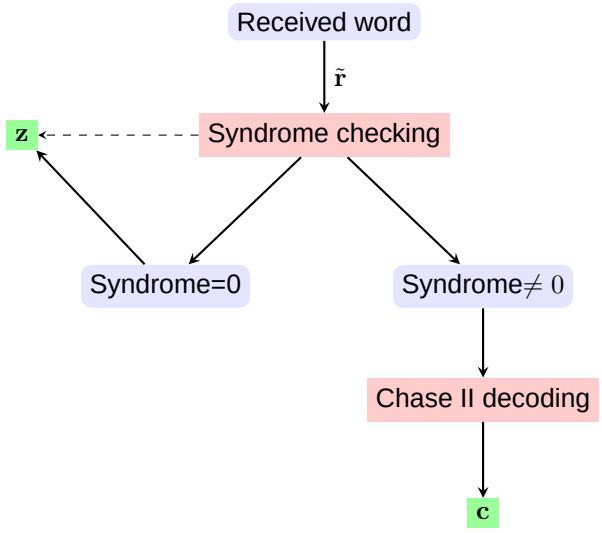


Fig. 2. Flow chart of the decoding process

IV. Simulation Result

In this section, we present simulation results that demonstrate the influence of the quantization on the decoding performance. All simulations are based on a binary Bose-Chaudhuri-Hocquenghem (BCH) code of length $n = 118$ and minimum Hamming distance $d_{min} = 4$. Hence, we choose $m = 3$. This code can be used as inner code in a concatenated code as proposed in [5], [8]. The first simulation results consider transmission over the AWGN channel without quantization. We compare the performance of Chase type II and III decoding with algebraic hard-input decoding.

In Fig. 3, Chase II has the lowest word error rate (WER) compared with Chase III and hard-input decoding. Chase II shows approximately 0.5 dB and 1.2 dB gain compared with Chase III and hard-input decoding, respectively. For the considered code, Chase II is twice as complex as Chase III, i.e., type II decoding considers 8 test patterns and type III only 4.

A. Threshold values

In the following, we consider simulations with quantization. In communication systems, typically linear quantization is used where the values of the quantization thresholds are uniformly spaced. If $B = 3$ bits represent the amplitude of a sample, linear quantization results in the threshold values $\delta_1 = 0.25$ and $\delta_2 = 0.5$ for the smallest quantization thresholds. However, these values to not lead to the best possible decoding performance. We demonstrate this in Fig. 4, where we choose $\delta_2 = 0.5$ and vary the value of δ_1. Fig. 4 shows that the decoding performance heavily depends on the

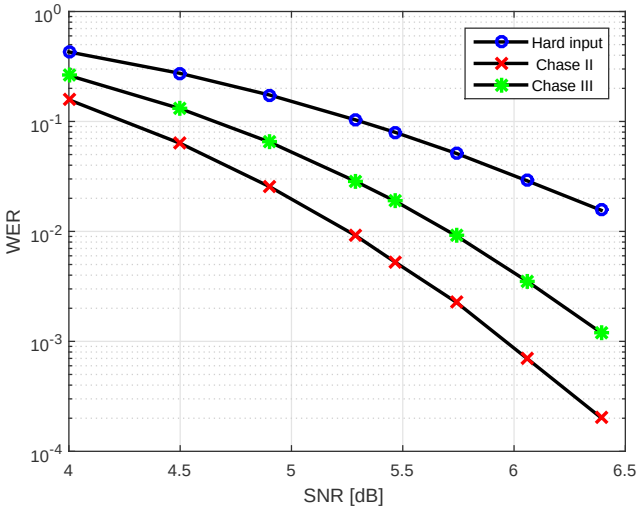


Fig. 3. A comparison between Chase II, Chase III, and hard-input decoding for the AWGN channel without quantization.

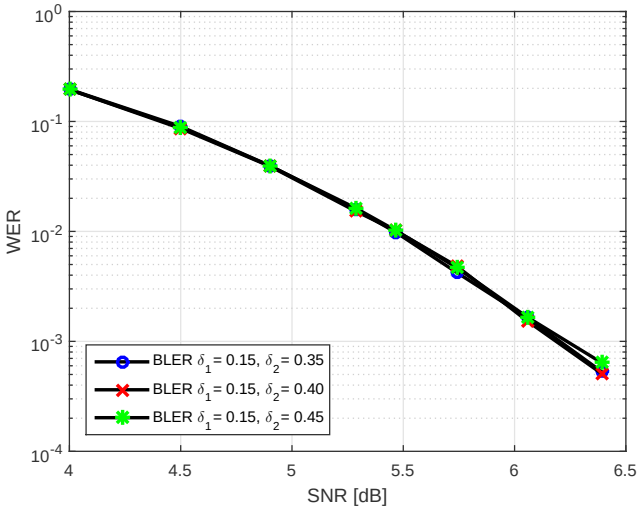


Fig. 5. A comparison for different values of the second threshold δ_2 and a fixed value of $\delta_1 = 0.15$

threshold values, where the best performance is obtained with $\delta_1 = 0.2$.

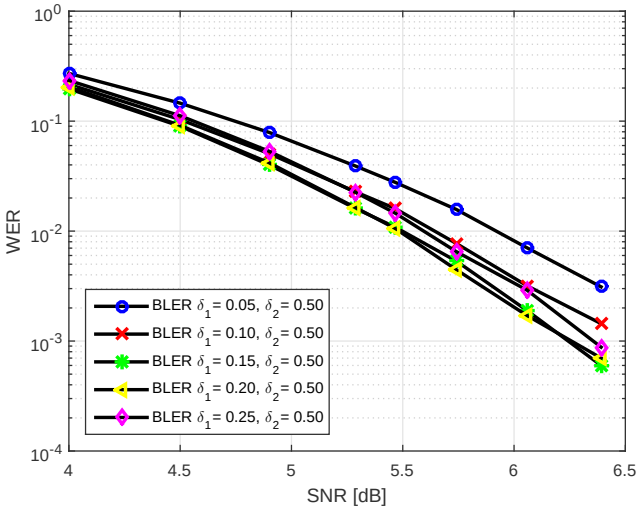


Fig. 4. A comparison of different values of δ_1 for a fixed value of $\delta_2 = 0.5$.

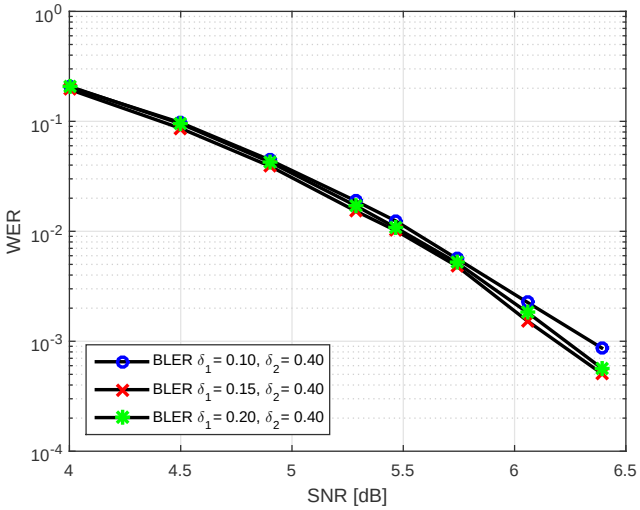


Fig. 6. A Comparison between different values of first threshold δ_1 with a fixed value of δ_2= 0.4

In Fig 4, the values δ_1 = 0.15 and 0.2 result in a similar performance with an average gain of 0.5dB compared with $\delta_1 = 0.05$. In order to optimize both threshold values, we choose δ_1 in the interval from 0.1 to 0.2 and vary the second threshold δ_2, where the best results are obtained for $\delta_1 = 0.15$.

In Fig. 5, the second threshold value δ_2 is adjusted based on $\delta_1 = 0.15$. The dependency of the decoding performance on δ_2 is smaller than on δ_1. For low SNR values the curves are close to each other. The value δ_2 = 0.4 is chosen for the second threshold.

Fig. 6 shows results for a fixed δ_2 = 0.4 and different values of δ_1. The thresholds δ_1 = 0.15 results in the best performance for the complete range of SNR values, where the differences are very small for high SNR values.

Finally, we consider a comparison of the performance with quantized versus unquantized reliability information. The corresponding simulation results are plotted in Fig. 7. The performance degradation with quantized input with the optimized threshold values is very small. For high SNR values the loss is less than 0.1dB.

B. Decoding failure declaration

Next we consider the performance of the failure declaring decoder. The quantization values of δ_1 and δ_2 are 0.15 and 0.4, respectively for this simulation. With this simulations, the decoder declares a failure when the choice of the best

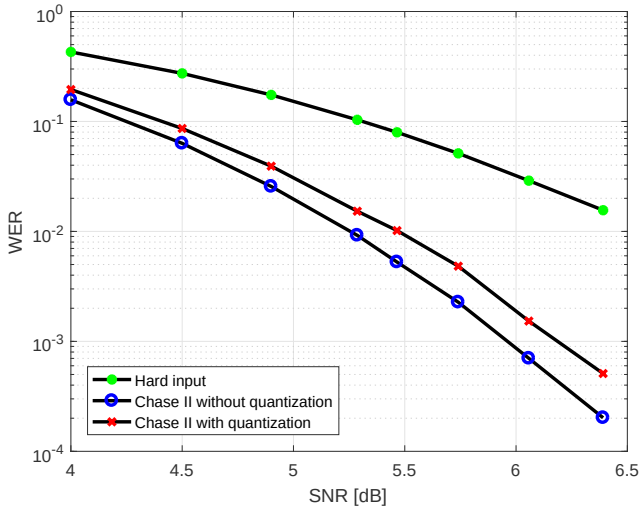


Fig. 7. Performance of Chase II decoding algorithm with quantized and unquantized inputs.

candidate is not unique. In this case, we have to error events: a decoding failure and a decoding error (word error). In a concatenated scheme, failures can be exploit using algebraic errors and erasures decoding procedures. For instance, a decoder for Reed-Solomon codes that can correct t errors can correct up to $2t$ erasures.

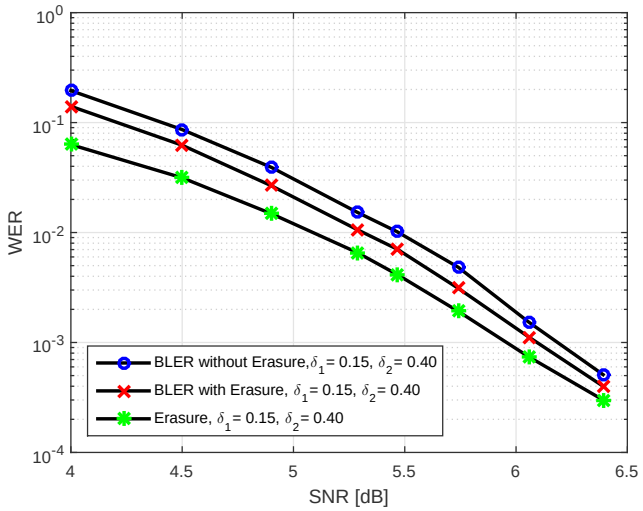


Fig. 8. Performance of Chase II algorithm with and without code block erasure.

In Fig. 8 it can be seen that the WER with erasure decoding is lower than the WER of the same code without erasure decoding. The decoding failure declaration shows an average gain of 0.3dB compared with decoding without decoding failure declaration.

V. Conclusion

In this paper we have investigated Chase II decoding with quantized input. The simulation results demonstrate that the performance loss due to quantization can be reduced by a suitable choice of threshold values. This optimization has applications to NAND flash memories, where reliability information is obtained by a multiple-read procedure with different threshold voltages.

With quantized input values, Chase decoding does not always result in a unique codeword. In this case, the decoder may declare a failure, because the probability of a decoding error is high. In a concatenated coding scheme, such decoding failures can be exploited using error and erasure decoding. The presented simulation results shown that the decoding failure declaration can improve the error and erasure performance.

Acknowledgment

We thank Hyperstone GmbH, Konstanz for supporting this project. The German Federal Ministry of Research and Education (BMBF) supported the research for this article (03FH025IX5).

References

[1] R. Micheloni, A. Marelli, and R. Ravasio, *Error Correction Codes for Non-Volatile Memories*. Springer, 2008.

[2] E. Yaakobi, J. Ma, L. Grupp, P. Siegel, S. Swanson, and J. Wolf, "Error characterization and coding schemes for flash memories," in *IEEE GLOBECOM Workshops*, Dec. 2010, pp. 1856–1860.

[3] J. Freudenberger and J. Spinner, "A configurable Bose-Chaudhuri-Hocquenghem codec architecture for flash controller applications," *Journal of Circuits, Systems, and Computers*, vol. 23, no. 2, pp. 1–15, Feb 2014.

[4] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in NAND Flash memory," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 2, pp. 429–439, Feb 2011.

[5] J. Spinner, J. Freudenberger, and S. Shavgulidze, "A soft input decoding algorithm for generalized concatenated codes," *IEEE Transactions on Communications*, vol. 64, no. 9, pp. 3585–3595, Sept 2016.

[6] D. Chase, "Class of algorithms for decoding block codes with channel measurement information," *IEEE Transactions on Information Theory*, pp. 170–182, 1972.

[7] M. P. Fossorier and S. Lin, "Chase-type and GMD coset decodings," *IEEE Transactions on Communications*, vol. 48, no. 3, pp. 345–350, 2000.

[8] J. Spinner, M. Rajab, and J. Freudenberger, "Construction of high-rate generalized concatenated codes for applications in non-volatile flash memories," in *2016 IEEE 8th International Memory Workshop (IMW)*, May 2016, pp. 1–4.

Improving gradient-based LSTM training for offline handwriting recognition by careful selection of the optimization method

Martin Schall
Institute for Optical Systems
University of Applied Sciences
Constance, Germany
Email: martin.schall@htwg-konstanz.de

Marc-Peter Schambach
Siemens Postal, Parcel &
Airport Logistics GmbH
Constance, Germany
Email: marc-peter.schambach@siemens.com

Matthias O. Franz
Institute for Optical Systems
University of Applied Sciences
Constance, Germany
Email: mfranz@htwg-konstanz.de

***Abstract*—Recent years have seen the proposal of several different gradient-based optimization methods for training artificial neural networks. Traditional methods include steepest descent with momentum, newer methods are based on per-parameter learning rates and some approximate Newton-step updates. This work contains the result of several experiments comparing different optimization methods. The experiments were targeted at offline handwriting recognition using hierarchical subsampling networks with recurrent LSTM layers. We present an overview of the used optimization methods, the results that were achieved and a discussion of why the methods lead to different results.**

***Index Terms*—offline handwriting recognition; recurrent neural network; long-short-term-memory; connectionist temporal classification; gradient-based learning; adadelta; rmsprop**

I. Introduction

Advances in the field of unconstrained and segmentation-free offline handwriting recognition using artificial neural networks have been considerable in the last years [1] and complete systems for this task have been published [2]. Offline handwriting recognition is in use in applications such as postal automation, banking and historical document analysis.

State of the art solutions for Latin script offline handwriting recognition are based on Multi-Dimensional Long-Short-Term-Memory *MDLSTM* [3] [4] recurrent neural networks organized as hierarchical subsampling networks [5]. Such networks can be trained for sequence classification using Connectionist Temporal Classification *CTC* [6]. CTC allows the training of networks for segmentation-free sequence classification without knowledge about the location of contained labels, based only on knowledge about the correct label sequence.

Newton's method can be used to determine an individual step-size for each parameter during backpropagation training of the artificial neural network [7]. Using Newton's method leads to fast convergence rates but requires the calculation of second-order derivatives of the error function. Since the calculation of second-order information is computationally expensive during backpropagation-based training, methods like AdaDelta [8] try to approximate it using only first-order information.

RProp [9] [10] provides an individual learning rate per parameter using only the changes in the sign of the partial derivative, similar to the Manhattan rule. RMSProp [11] improves on this concept by generalizing to mini-batch training variants of the backpropagation algorithm. RMSProp does so by normalizing the gradient using the rolling mean value of the previous first-order derivatives.

This work provides an overview and comparison of contemporary gradient-based optimization methods for training hierarchical subsampling MDLSTM-networks using CTC. All experiments were done using the IAM offline handwriting database [12]. It is meant as a guide for practitioners in the field of offline handwriting recognition. In addition, this work includes theoretical interpretations of the observed results.

The paper starts by describing the used network topology in section II, the investigated optimization methods in section III and the experiments executed in section IV. Section V presents the results of the experiments and section VI discusses the problems arising with the optimization methods. Section VII concludes the paper.

II. Network

The network topology was identical for all experiments and is based on the hierarchical subsampling network using MDLSTM-cells applied for Arabic handwriting recognition [2] [5]. While the network topology was unchanged, the hyperparameters and sizes of the neuron layers were modified. The exact network topology, beginning at the network input, and hyperparameters are described in table I.

The LSTM variant [13] used in the comparisons includes forget gates, peephole connections, bias values and the full gradient for backpropagation. The fully connected feedforward neurons had no bias, except for the very last neuron layer. The non-linearities are the standard logistic sigmoid $\phi(x) = \frac{1}{1+e^{-x}}$ for the LSTM gates and the hyperbolic tangent $tanh(x)$ for all other activations. The network consists of a total of 148799 parameters, all of which were initialized

TABLE I
NETWORK TOPOLOGY USED FOR THE EXPERIMENTS

Type of layer	Configuration
Input image	Grayscale; 81 pixel in height
Subsampling	2×3 (width $\times$ height)
MDLSTM	2 cells per scan direction
Subsampling	2×3 (width $\times$ height)
Fully connected feedforward	6 neurons; no bias
MDLSTM	10 cells per scan direction
Subsampling	2×3 (width $\times$ height)
Fully connected feedforward	20 neurons; no bias
MDLSTM	50 cells per scan direction
Fully connected feedforward	79 neurons; with bias
Collapse	
Softmax	
CTC	78 glyph labels; 1 blank label

drawing from a random uniform distribution in the interval $[-0.1; +0.1]$.

III. METHODS

The following paragraphs outline the gradient-based optimization methods: Steepest descent with momentum, RMSProp and AdaDelta. In all equations, g_t is the gradient at time t and δx_t the parameter updates at time t. μ is always the learning rate, α the decay rate and β the dampening factor. All variables are initialized to zero if not otherwise defined.

Algorithm 1 Steepest descent with momentum

$$\delta x_t = (\mu \times g_t) + (\alpha \times \delta x_{t-1})$$

Algorithm 1 describes the steepest descent optimizer with a simple momentum term added. It scales the first-order derivative of the error function by a constant learning rate, thus generating parameter updates that are directly proportional to the gradient. The added momentum term prevents the optimizer from following jitters in the error function along the current path. Figuratively speaking, if the optimization process is a ball moving down the error landscape, momentum changes the gradient from being a vector of movement to a vector of force applied to the ball.

Algorithm 2 RMSProp

$$E[g^2]_t = ((1-\alpha) \times g_t^2) + (\alpha \times E[g^2]_{t-1})$$
$$\delta x_t = \mu \times \frac{g_t}{\sqrt{E[g^2]_t}}$$

RMSProp, outlined in Algorithm 2, is a generalization of RProp that allows mini-batch training. Both only take the sign of the gradient into account but determine the step size of parameter updates independently from the absolute value of the gradient. RMSProp does so by using a rolling mean of the gradient for normalization. It effectively allows the user to choose the actual step size of parameter updates as a hyperparameter.

Algorithm 3 describes AdaDelta, which uses an approximation of the diagonal values of the Hessian matrix to do quasi-Newton updates. AdaDelta provides per-dimension step sizes and basically removes the need to manually choose a learning rate. The idea behind AdaDelta is outlined in the according publication [8], calculating the parameter updates based on the inverse Hessian as $\frac{\delta x}{g}$. Since both the total parameter updates δx and the total gradient g for the Newton step are unknown, they are approximated using a rolling mean of the last values. A variant of AdaDelta adds an additional learning rate μ, which should be chosen as a value near 1.0 since the unmodified AdaDelta implies a global learning rate of 1.0.

Algorithm 3 AdaDelta with additional learning rate

$$E[g^2]_t = ((1-\alpha) \times g_t^2) + (\alpha \times E[g^2]_{t-1})$$
$$u_t = g_t \times \frac{\sqrt{E[\delta x^2]_{t-1} + \beta}}{\sqrt{E[g^2]_t + \beta}}$$
$$E[\delta x^2]_t = ((1-\alpha) \times u_t^2) + (\alpha \times E[\delta x^2]_{t-1})$$
$$\delta x_t = \mu \times u_t$$

When gradient clipping was applied, only the error signal that is transported from a LSTM layer to its predecessor was truncated. Recalling the network topology defined in Table I, this concerns only the transition between the last two MDLSTM layers and their previous fully connected feedforward layers. The error signal was hard clipped to be within the interval $[-1; +1]$.

IV. EXPERIMENTS

All experiments were carried out using the IAM offline handwriting database [12] with the images being rescaled to 8-bit grayscale and fixed 81 pixel in height with a variable width. A random subset of 90% (86809) samples were used for training and 5% (4822) each for validation and evaluation. A sample of the IAM database is shown in figure 1.

Fig. 1. Example of the IAM database

The network is specified in section II and the target function of supervised training was CTC with 78 visible character classes of the IAM database. No normalization of labels was applied.

If not otherwise noted, the training was done using mini-batch updates of size 8 and the full non-clipped gradient. The gradients within a mini-batch were summed, but not normalized afterwards. The training samples were processed in a random permutation for each training epoch. The experiments used early stopping until the validation error rate did not improve for 5 epochs.

The following individual experiments were conducted in this work:

1) Steepest descent with momentum and full gradient.
2) Steepest descent with momentum and gradient clipping.

3) RMSProp.
4) AdaDelta without additional learning rate.
5) AdaDelta with additional learning rate.

The hyperparameters were chosen on basis of previous experiments with this network architecture and the IAM database. The hyperparameters have proven to be suitable for training this network for offline handwriting recognition.

Error rate was measured in terms of Character Error Rate *CER* at the end of each training epoch. CER is defined as the percentage $CER(y,z) = \frac{100 \times ED(y,z)}{|y|}$. It measures the part of the edit-distance [14] $ED(y,z)$ between the correct label string y and the decoded network output z in relation to the length $|y|$ of the correct label. The CER of these experiments are averages over all samples within the training set or validation set respectively.

V. Results

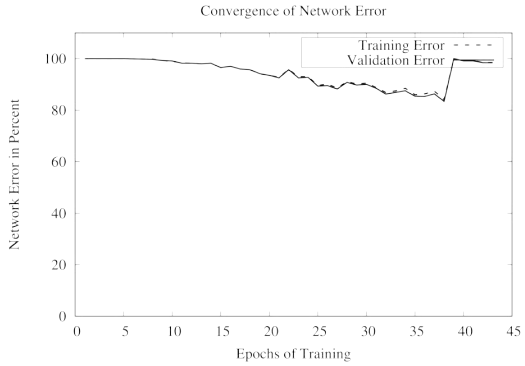


Fig. 2. Steepest descent with $\mu = 1e^{-4}$ and $\alpha = 0.9$

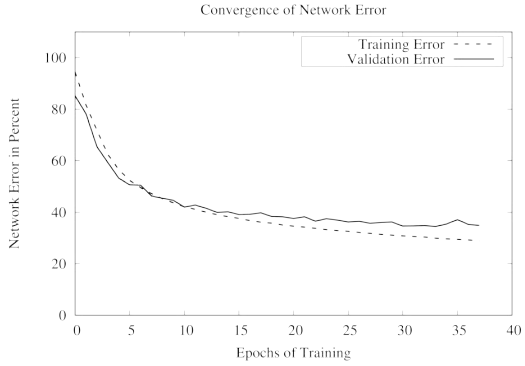


Fig. 3. Steepest descent with $\mu = 1e^{-4}$ and $\alpha = 0.9$ (gradient clipping)

Figures 2 and 3 show the convergence of the CER during training using steepest descent with momentum. Hyperparameters were $\mu = 1e^{-4}$ and $\alpha = 0.9$. The training using the full non-clipped training did not converge to acceptable error rates as can be seen in figure 2. The use of gradient clipping did improve the convergence of CER, see figure 3, during training. The convergence rate is still lower than with RMSProp or AdaDelta, however.

As can be seen in figure 2, the CER initially decreases for some epochs but then started increasing again and stabilizes at 99%.

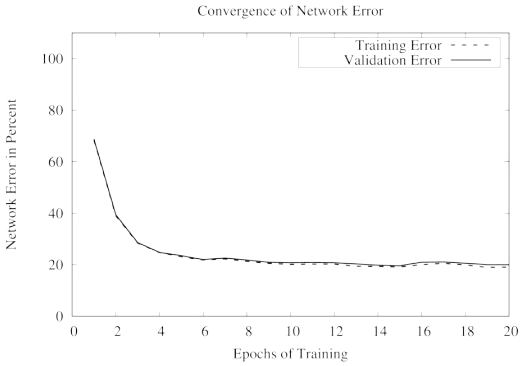


Fig. 4. RMSProp with $\mu = 1e^{-3}$ and $\alpha = 0.9$

Figure 4 shows the convergence of the error rate using RMSProp with $\mu = 1e^{-3}$ and $\alpha = 0.9$. It shows a faster convergence rate than steepest descent with gradient clipping and achieves a lower error rate.

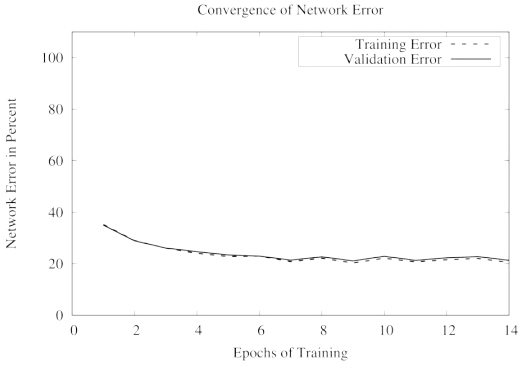


Fig. 5. AdaDelta with $\mu = 1$, $\alpha = 0.95$ and $\beta = 1e^{-6}$

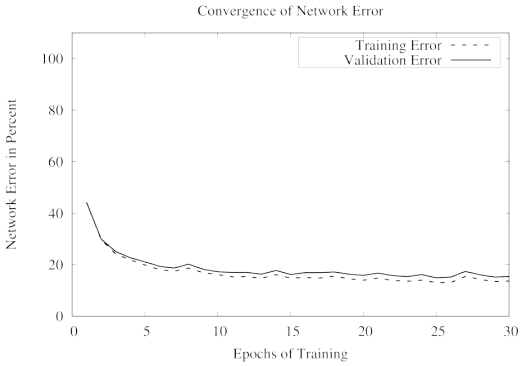


Fig. 6. AdaDelta with $\mu = 0.5$, $\alpha = 0.95$ and $\beta = 1e^{-6}$

Figures 5 and 6 contain the results using AdaDelta. Both use the hyperparameters $\alpha = 0.95$ and $\beta = 1e^{-6}$. The experiment described in figure 5 used a learning rate of $\mu = 1$, thus corresponds to the original work by the authors of AdaDelta [8]. Figure 6 uses an additional learning rate of $\mu = 0.5$, which reduces the convergence rate by the same factor. Using an additional learning rate proved to result in lower final error rates.

The fastest convergence rate in these experiments was achieved using AdaDelta with $\mu = 1$, $\alpha = 0.95$ and $\beta = 1e^{-6}$, the lowest error rate with AdaDelta and $\mu = 0.5$.

VI. Discussion

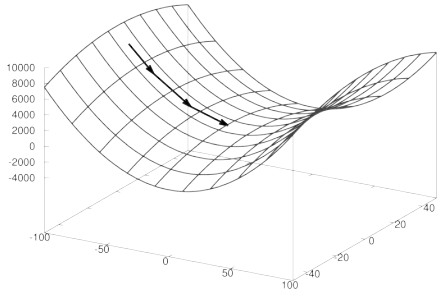


Fig. 7. Exemplary saddle point of an error function in a two-dimensional parameter space

In the following section, we discuss possible reasons for why the compared optimization methods behave differently in terms of convergence of network error. Recent work [15] [16] has shown that saddle points in the error function tend to be a major problem while training artificial neural networks. Other potential problems arise from the interaction between Backpropagation-Through-Time *BPTT* [17] and a momentum term in the optimization method. Figure 7 shows an error function with a saddle point that highlights the different behavior of the three optimization methods in this situation. Saddle points in the error function are interesting because they both consist of steep and shallow parts but the direction of any gradient descent optimization will change on a saddle point. Differences arise as soon as the gradient descent optimization moves from the steep flank of the error function to somewhere near the saddle point.

Consider Algorithm 1 (steepest descent with momentum): while descending down the steep part of the error function, the momentum will increase accordingly. The absolute value of the gradient will be very small in comparison to the gradient on the steep part, which results in only a small impact of the current gradient when updating the parameters. In this exemplary case, gradient descent will overshoot the saddle point instead of following the gradient to the decreasing error values.

RMSProp and AdaDelta, see algorithms 2 and 3, tackle this problem by normalizing the parameter updates with the expectation value of the absolute gradient. The expectation value is again large after traversing the steep flank of the error function. After normalization, the relatively small gradient near the saddle point will be even smaller. The actual per-parameter learning rate is decreased and thus the gradient descent slows down near the saddle point. An increase in the per-parameter learning rate will occur as soon as the expectation value of the gradient has adapted to the small gradient value. This behavior allows for a change of direction near saddle points without overshooting it.

Another potential problem arises in the BPTT algorithm in combination with training samples of variable sizes, e.g. different sizes of the input images. BPTT calculates the gradient by virtually unrolling the recurrent network into a feedforward network. Training samples of longer sequences will result in 'deeper' unrolled networks. Parameters of recurrent layers are shared in the unrolled network and thus their gradients need to be summed again before updating their parameters. Similar to mini-batch training, the gradients summed up to obtain the accumulated gradient for the recurrent layer.

Steepest descent with momentum and full gradient is prone to an effect similar to the 'exploding gradient': The absolute value of the gradient is directly proportional to the sequence length. For a long sequence, the momentum will be accumulated, while short sequences have little impact on gradient descent. This again leads to overshooting of minimum points or saddle points. This 'exploding gradient' explains why gradient clipping is effective for steepest descent, as can be seen in the convergence rates of figures 2 and 3.

VII. Conclusion

This work presents the results of several experiments training hierarchical subsampling networks using LSTM-cells for offline handwriting recognition. Three different gradient-based optimization methods were used: steepest descent, RMSProp and AdaDelta. Steepest descent was tested both with the full, non-clipped, gradient and with gradient clipping.

The results show a better convergence rate for RMSProp and AdaDelta than for normal steepest descent. Both RMSProp and AdaDelta are easy to implement and cause only a linear overhead in memory consumption which makes them reasonable choices for practitioners. AdaDelta with a reduced learning rate of 0.5 achieved the lowest error rate of all experiments.

Section VI rationales why steepest descent shows a worse behavior than RMSProp or AdaDelta in the presence of saddle points or when using BPTT for recurrent neural networks. Saddle points can be expected [16] in high-dimensional non-convex optimization problems such as offline handwriting recognition. BPTT is the establish method for gradient-based training of recurrent neural networks and as such, problems arising out of the interaction between BPTT and the optimization method should be considered.

Based on the observations during the experiments and the following reflections, the authors are suggesting to use

AdaDelta for training LSTM networks for offline handwriting recognition. Newer optimization methods, such as Adam [18], were not taken into consideration but may give even results.

Acknowledgment

The authors would like to thank the Siemens Postal, Parcel & Airport Logistics GmbH for funding this work. The authors would also like to thank Jörg Rottland for proof-reading this work and his valuable suggestions.

References

[1] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[2] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012.

[3] A. Graves, S. Fernandez, and J. Schmidhuber, "Multi-Dimensional Recurrent Neural Networks," IDSIA/USI-SUPSI, Tech. Rep., 2007.

[4] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM." *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[5] A. Graves and J. Schmidhuber, "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks," in *Advances in Neural Information Processing Systems 21, NIPS'21*, 2008, pp. 545–552.

[6] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification : Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proceedings of the 23rd international conference on Machine Learning*. ACM Press, 2006, pp. 369–376.

[7] T. Schaul, S. Zhang, and Y. LeCun, "No More Pesky Learning Rates," *Journal of Machine Learning Research*, vol. 28, no. 2, pp. 343–351, 2013.

[8] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," p. 6, 2012.

[9] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm," *IEEE International Conference on Neural Networks*, 1993.

[10] C. Igel and M. Hüsken, "Improving the Rprop learning algorithm," in *Proceedings of the Second International Symposium on Neural Computation*, 2000, pp. 115–121.

[11] Y. N. Dauphin, J. Chung, and Y. Bengio, "RMSProp and equilibrated adaptive learning rates for non-convex optimization," Tech. Rep., 2014.

[12] U. V. Marti and H. Bunke, "The IAM-database: An English sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2003.

[13] K. Greff, R. K. Srivastava, J. Koutník, B. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," IDSIA/USI-SUPSI, Tech. Rep., 2015.

[14] R. A. Wagner and M. J. Fischer, "The String-to-String Correction Problem," *Journal of the ACM*, vol. 21, no. 1, pp. 168–173, 1974.

[15] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, "The Loss Surfaces of Multilayer Networks," *Aistats*, vol. 38, pp. 192—-204, 2015. [Online]. Available: http://arxiv.org/abs/1412.0233

[16] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," *arXiv*, pp. 1–14, 2014. [Online]. Available: http://arxiv.org/abs/1406.2572

[17] R. Rojas, "The Backpropagation Algorithm," in *Neural Networks*. Springer, 1996, pp. 151–184.

[18] D. P. Kingma and J. L. Ba, "Adam: a Method for Stochastic Optimization," *International Conference on Learning Representations*, pp. 1–13, 2015.

Depth Estimation from Micro Images of a Plenoptic Camera

Jennifer Konz, Niclas Zeller, Franz Quint
Karlsruhe University of Applied Sciences
Moltkestr. 30, 76133 Karlsruhe, Germany
{jennifer.konz, niclas.zeller, franz.quint}@hs-karlsruhe.de

Uwe Stilla
Technische Universität München
Arcisstr. 21, 80333 Munich, Germany
stilla@tum.de

***Abstract*—This paper presents a method to calculate the depth values of 3D points by means of a plenoptic camera. Opposed to other approaches which use the totally focused image to detect points, we operate directly on the micro images taking thus advantage of the higher resolution of the plenoptic cameras raw image. Depth estimation takes place only for the points of interest, resulting in a semi-dense approach. The detected points can further be used in a subsequent simultaneous localization and mapping (SLAM) process.**

Index Terms - depth estimation, focused plenoptic camera, micro images

I. Introduction

The concept of a plenoptic camera is known for over one hundred years (Ives, 1903 [1], Lippmann, 1908 [2]) but only due to the capabilities of nowadays graphic processor units (CPUs) an evaluation of video sequences with acceptable frame rates is possible.

The main advantage of a plenoptic camera is the depth information which can be estimated from only a single image. A traditional camera, which captures a 2D image of a scene, does not provide any depth estimation from one shot. In comparison, a plenoptic camera captures a complete 4D lightfield representation, which is suitable to calculate depth information [3][4].

There have been developed two concepts of plenoptic cameras: The unfocused plenoptic camera [5] and the focused plenoptic camera [6]. In this paper we work with focused plenoptic cameras. In these, a micro lens array (MLA) which is placed between the main lens and the sensor focuses the image of the former on the later. Thus, the raw image of this type of camera consists of many micro images, which each show a portion of the main lens image. These portions are pictured from a slightly different perspective in neighboring micro images. The disparities of corresponding points in the micro images enable the estimation of depth. The procedure for depth estimation with this type of camera is described e.g in [7]. This depth is in a first instance a virtual depth, i.e. related to internal parameters of the camera. However, calibrating the camera allows to compute metrical depth values. For the calibration process of the camera as well as for the procedures to synthesize a totally focused image please refer to [8].

A. Contribution of this work

This paper focuses on simplifying depth estimation by detecting points of interest (POI) directly in the raw image and not in the synthesized totally focused image. By matching POIs in the raw image that represent the same 3D point, its depth will be estimated. Because depth estimation mainly relies on the quality of matching the POIs from different micro images, different POI detectors are used and compared. It is of particular interest whether ordinary detectors which have been developed for traditional images can also be used for the raw image provided by a focused plenoptic camera.

Correspondence for points in different micro images is sought using epipolar geometry. Since the sensor is placed in front of the main lenses image, the image coordinates of points in the micro image differ from those in the (virtual) image of the main lens. This has to be accounted for in depth estimation, which is done by a linear regression. With a subsequent consideration of the depth information and calculation of the error between the actual projections and the matched features, the results of the depth estimation can further be improved.

The paper is structured as follows. In section II.A the concept of the focused plenoptic camera which is used in our approach is described. The depth estimation is explained based on this configuration. The particularities of POI detection in raw images are presented in section III. Section IV deals with matching the points highlighted by the detectors in different micro images. In section V we formulate the depth estimation for the matched image points. Section VI presents the results of the depth estimation by using different detectors and compares them. We conclude with section VII.

II. The Focused Plenoptic Camera

To highlight the differences between the set-up of a traditional camera and a plenoptic camera, the optical path of a thin lens is displayed in Figure 1. For a traditional camera the thin lens equation as given in eq. (1) can be used to describe the relation between the object distance a_L and the image distance b_L using the focal length f_L of the main lens.

$$\frac{1}{f_L} = \frac{1}{a_L} + \frac{1}{b_L} \tag{1}$$

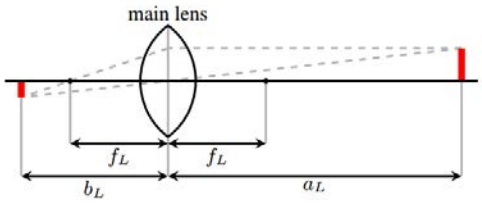


Fig. 1. Optical path of a thin lens [9]

With the configuration of traditional cameras displayed in Figure 1 the intensity of incident light is recorded on the image sensor. The 2D-image recorded by a traditional camera does not provide any information about the object distance. To gain information about the object distance a_L a plenoptic camera can be used.

A plenoptic camera consists of a micro lens array (MLA) which is placed between the main lens and the sensor. Regarding the position of the MLA and the sensor to the main lens image, two different configurations of a focused plenoptic camera are described by Lunsdaine and Georgiev [6][10].

In the Keplerian configuration the MLA and the sensor are placed behind the main lens images (s. Figure 2), whereas in the Galilean configuration MLA and sensor are in front of the main lens image (s. Figure 3). In the Galilean configuration the main lens image is only a virtual image. The plenoptic camera used in this work is with Galilean configuration.

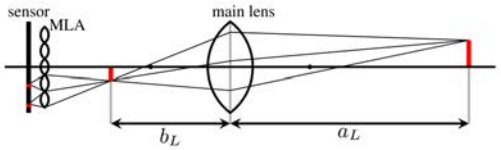


Fig. 2. Keplerian configuration [8]

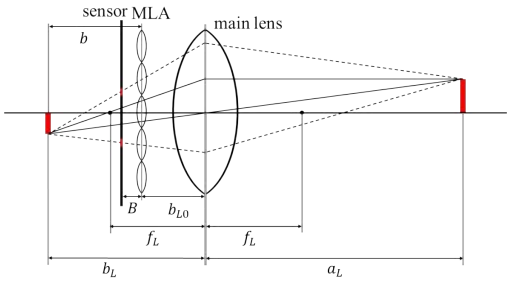


Fig. 3. Galilean configuration (based on [9])

The MLA of our camera has a hexagonal arrangement of the micro lenses (cf. Figure 4). There are three different types of micro lenses on the MLA, having different focal lengths. Thus different virtual image distances (resp. object distances) are displayed in focus on the sensor. Therefore, the effective depth of field (DOF) of the camera is increased compared to a focused plenoptic camera with a MLA consisting of micro lenses with the same focal length [8].

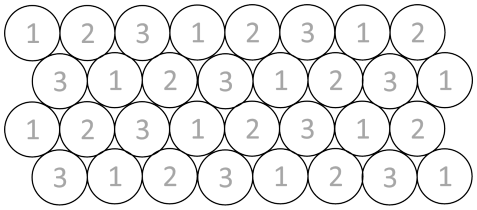


Fig. 4. Arrangement of the MLAs. Different micro lens types are marked by different numbers [11].

Each micro lens of the MLA produces a micro image on the sensor (s. Figure 5). However, depending on the focal length of the corresponding micro lens, a 3D point projected in different micro lenses will be focused in some of the micro images, while it is unfocused in others.

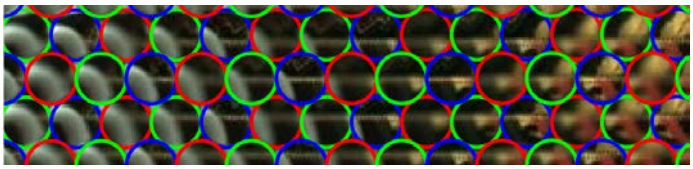


Fig. 5. Section of the micro lens images (raw image) of a Raytrix camera. Different micro lens types are marked by different colors. [8]

The thin lens equation of the main lens of a plenoptic camera in Galilean configuration can be written as in eq. (2) using the parameters of the camera. The parameter b_{L0} is the distance between the main lens and the MLA and b represents the distance between the MLA and the virtual image.

$$\frac{1}{f_L} = \frac{1}{a_L} + \frac{1}{b_L} = \frac{1}{a_L} + \frac{1}{b + b_{L0}} \tag{2}$$

The value of b_{L0} can be estimated with a previous calibration, using the method described by Zeller et al. [8]. In the same calibration process also the focal length f_L of the main lens is estimated. If one can estimate the virtual image distance b, then it is possible to calculate the depth of the object point using eq. (2). This is described in the following section.

A. Depth Estimation

To calculate the object distance of the 3D point, the virtual image distance b has to be determined. This is done using the coordinate system displayed in Figure 6, which is aligned to the MLA.

The geometrical relations of a virtual image point and the corresponding points in two micro images which result from the same 3D are displayed in Figure 7. The depth estimation is based on the method described by Zeller et al. [9] by using the disparity of a point in two micro images.

The principal points of the micro lenses (eq. (3)), the projection of a 3D point in the micro images (eq. (4)) and in the virtual image (eq. (5)) respectively are described with their three-dimensional position vectors in the coordinate system of Figure 6.

$$\vec{c}_i = \begin{bmatrix} c_{x,i} & c_{y,i} & 0 \end{bmatrix}^T \tag{3}$$

$$\vec{x}_{R,i} = \begin{bmatrix} x_{R,i} & y_{R,i} & B \end{bmatrix}^T \tag{4}$$

$$\vec{x}_V = \begin{bmatrix} x_V & y_V & v \end{bmatrix}^T \tag{5}$$

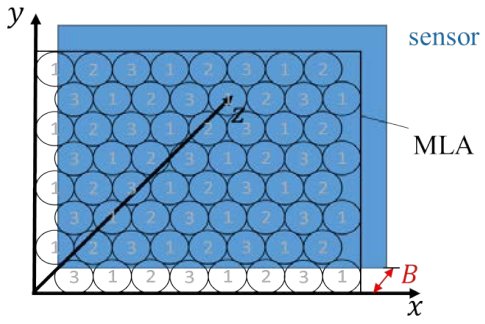


Fig. 6. Coordinate system for depth estimation

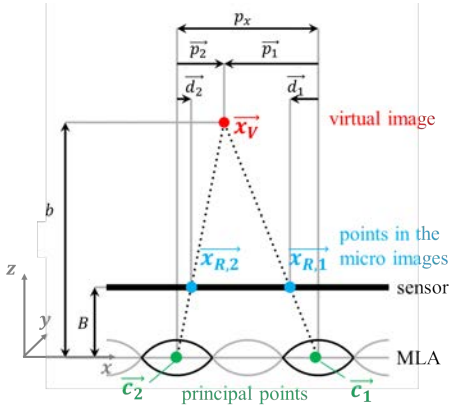


Fig. 7. Principle of the depth estimation (based on [9])

The intersection of the rays through the points in the micro images defines the virtual image in a distance b to the MLA. The triangles with a common vertex in the principal point of a micro lens and with their basis on the sensor (micro image) or in the virtual image respectively, are similar, leading to eq. (6).

$$\frac{\vec{d}_i}{B} = \frac{\vec{p}_i}{b} \tag{6}$$

The vectors $\vec{d}_i$ (eq. (7)) define the distance between the principal points of a micro lens and the point in the corresponding micro image in x- and y-coordinates. The distance between a principal point and the virtual image in x- and y-coordinates is described with $\vec{p}_i$ (s. eq. (8)). The x- and y-coordinates of both vectors are signed values and their sign is defined by the coordinate system in Figure 6.

$$\vec{d}_i = \begin{bmatrix} d_{x,i} & d_{y,i} \end{bmatrix}^T \tag{7}$$

$$\vec{p}_i = \begin{bmatrix} p_{x,i} & p_{y,i} \end{bmatrix}^T \tag{8}$$

With eq. (6) and the two-dimensional vectors, the distance b can be calculated using the x- or the y-coordinates. In the following calculations the x-coordinates are used.

The parallax p of the virtual image point, which defines the distance between the principal points of the two micro lenses, is described by eq. (9).

$$p = p_{x,2} - p_{x,1} \tag{9}$$

The disparity d is described as the difference between $d_{x,2}$ and $d_{x,1}$. With eq. (6) and (9) the definition for the disparity given in eq. (10) is received.

$$d = d_{x,2} - d_{x,1} = (p_{x,2} - p_{x,1}) \cdot \frac{B}{b} = p \cdot \frac{B}{b} \tag{10}$$

Equation (6) can be simplified using eq. (10) resulting in the distance b being defined as given in eq. (11).

$$b = \frac{p \cdot B}{d} \tag{11}$$

The disparity d and the parallax p are determined by the 3D point, respectively the distance between the micro images in which it is projected into. The fraction of the disparity with respect to the parallax is called the virtual depth v as given in eq. (12).

$$v = \frac{b}{B} \tag{12}$$

The virtual depth is proportional to the distance b between the MLA and the virtual image. This is needed to calculate the object distance a_L cf. eq. (2). The factor of proportionality is the inverse of b, i.e. the distance between MLA and sensor. This has to be estimated in a previous calibration step.

A point $\vec{x}_V$ in the virtual image is defined as the intersection of the rays through the projected points in the micro images. Its position vector $\vec{x}_V$ is given by eq. (13).

$$\vec{x}_V = (\vec{x}_{R,i} - \vec{c}_i) \cdot \frac{b}{B} + \vec{c}_i \tag{13}$$

It is straightforward to see that its x- and y-coordinates depend on the virtual depth v as indicated in the linear equations (14) and (15):

$$x_V = (x_{R,i} - c_{x,i}) \cdot v + c_{x,i} \tag{14}$$

$$y_V = (y_{R,i} - c_{y,i}) \cdot v + c_{y,i} \tag{15}$$

For two points which represent the same 3D point, the x- and y-coordinates calculated with eq. (14) and (15) have to be equal.

With this known model, the virtual image of a 3D point can be estimated if the 3D point is detected in more than two micro images. To calculate the object distance of the 3D point,

the relation between the virtual depth and the object distance is used (s. eq. (2) and (12)) and the following relation for the object distance $z_c = a_L$ holds:

$$z_c = \left(\frac{1}{f_L} - \frac{1}{v \cdot B + b_{L0}} \right)^{-1} \tag{16}$$

So to calculate the virtual depth v and the object distance z_c, points in the micro images have to be detected (s. section III) and matched (s. section IV).

III. Point Detection in Micro Images

To detect points in micro images, several detection methods like SURF, SIFT and Harris Corner Detector are applied directly to the raw image. They deliver a set of points of interest (POI), which will be used for estimation of the depth of the corresponding object points. As described in the previous section, the MLA is arranged hexagonally. By projecting the principal point of a micro lens orthogonally onto the sensor, the principal point in the corresponding micro image can be determined. With knowledge of the diameter of the micro lenses (measured in pixels on the sensor), the detected POI can be allocated to a certain micro image (s. Figure 8). To locate a point $\vec{x}_R$ in the micro image, the vector as given in eq. (18) is calculated for each micro image with principal point $\vec{c}_i$. Because of the arrangement of the micro lenses in the MLA, a lens border (of 1.5 pixels) has to be defined which separates the micro images. The lens border is not used for point detection. The POI which do not fulfill eq. (19) for any micro image are not used in subsequent processing.

$$\vec{d}_{x,ci} = \vec{c}_i - \vec{x}_R \tag{17}$$

$$\left| \vec{d}_{x,ci} \right| \leq r_{lens} - l_{border} \tag{18}$$

With eq. (18) and (19) the POI can be allocated to the micro images. The allocated points are used in the following matching as described in section IV.

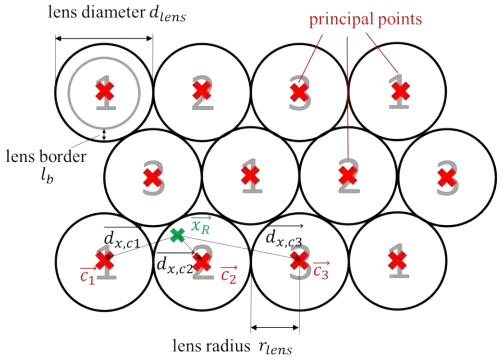


Fig. 8. Principle to determine the corresponding micro image for a detected POI

IV. Point Matching

At first, only the adjacent micro images are used to find other points that match to the POI located in the reference micro image. The approach to match them in adjacent micro images is exemplified in Figure 9. The vectors of the detected points $\vec{x}_{R,i}$ and the principal points $\vec{c}_i$ of the micro images in Figure 9 are two-dimensional. So only the x- and y-coordinates are used.

To match a POI to another POI in an adjacent micro image, the epipolar geometry can be used to restrict the search area. In Figure 10 the epipolar geometry between the virtual image and its representation in two micro images is displayed. Due to the plane-parallel arrangement of the microlenses, the epipolar lines will all be parallel to the line connecting the centers of the micro lenses (eq. (20)):

$$\vec{e}_{12} = \vec{c}_2 - \vec{c}_1 \tag{19}$$

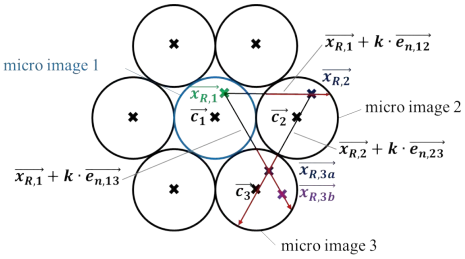


Fig. 9. Matching of the detected points using epipolar geometry

Starting from the detected POI $\vec{x}_{R,1}$, a corresponding POI would be located alongside the epipolar line. Due to manufacturing tolerances we allow a POI to be maximally one pixel away from the epipolar line to be matched to the reference POI. If several POIs are matched in the same micro image to a reference POI, those POI which do not belong to the same virtual image can be eliminated using already matched POIs from other micro images together with the epipolar geometry between those micro images (see Figure 9).

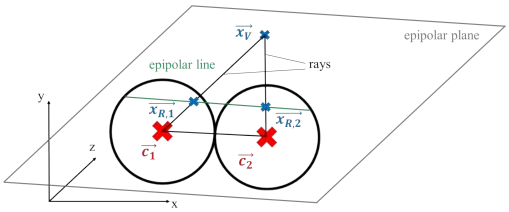


Fig. 10. Principle of the epipolar geometry with two micro images

After the matching is done in the adjacent micro images for every detected POI, the matched POIs are linked as indicated in Figure 11. This means that an uninterrupted chain of adjacent micro images is needed for POIs to be linked. However, since the micro lenses have different focal lengths and are arranged in the MLA as indicated in Figure 4, not only the directly adjacent micro lenses are considered for matching,

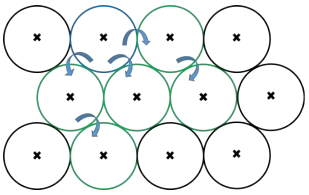


Fig. 11. Group of matched POIs with linkage

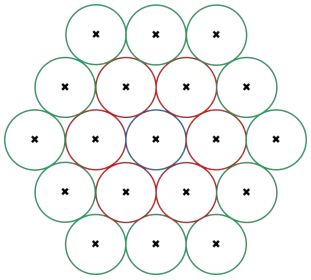


Fig. 12. Micro images used for direct POI matching

but also those with the same focal length which are just behind the directly adjacent ones (see Figure 12). This means that correspondences are searched in a total of 18 neighbouring micro images.

V. Depth Estimation using the Matched POI

Each group of matched POI should represent the same virtual image point (resp. one 3D point) in the corresponding micro images. If the rays of the matched POI do not have a common intersection (s. Figure 13), the matched POI have to be optimized.

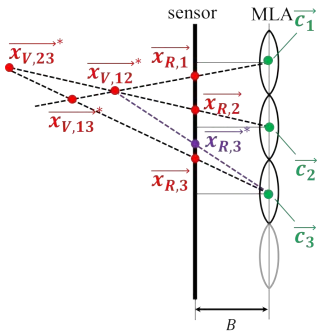


Fig. 13. Error estimation. Projecting a virtual image into the micro image

By choosing the intersection of the rays of $\vec{x}_{R,1}$ and $\vec{x}_{R,2}$, an error occurs for $\vec{x}_{R,3}$. The error for each POI can be described as given in eq. (20), where $\vec{x}^{\,*}_{R,i}$ is the virtual image point back-projected into the micro image.

$$\vec{\Delta}_i = \begin{bmatrix}\Delta_{x,i} & \Delta_{y,i}\end{bmatrix}^T = \vec{x}_{R,i} - \vec{x}^{\,*}_{R,i} \tag{20}$$

The errors are minimized using the least square method. Because of the linear relation between the known values (POI coordinates, principal points) and the unknown values (coordinates of the virtual image point) (s. eq. (14) and (15)) the error minimization is performed using linear regression.

A. Linear Regression

For linear regression eq. (14) and (15) are transformed into the following functions:

$$f_{x,i} = x_{R,i} - c_{x,i} = \frac{x_V - c_{x,i}}{v} \tag{21}$$

$$f_{y,i} = y_{R,i} - c_{y,i} = \frac{y_V - c_{y,i}}{v} \tag{22}$$

According to this, the normalized coordinates of the virtual image point cf. eq. (23) are estimated using eq. (24), whereas the residual vector is defined with eq. (25) and the Jacobi matrix is given in eq. (26). For more details on linear regression we refer to [12]. Whereas N is the number of matched POI in one group, which can maximally be 18 (cf. Figure 12).

$$\vec{a} = \begin{bmatrix}\frac{x_V}{v} & \frac{y_V}{v} & \frac{1}{v}\end{bmatrix}^T \tag{23}$$

$$\vec{a}^* = \begin{bmatrix}\frac{x_V{}^*}{v^*} & \frac{y_V{}^*}{v^*} & \frac{1}{v^*}\end{bmatrix}^T = \left(J^T \cdot J\right)^{-1} \cdot J^T \cdot \vec{r} \tag{24}$$

$$\vec{r} = \begin{bmatrix}x_{R,1} - c_{x,1} & y_{R,1} - c_{y,1} & \dots & x_{R,N} - c_{x,N} & y_{R,N} - c_{y,N}\end{bmatrix}^T \tag{25}$$

$$J = \begin{bmatrix}\frac{\partial f_{x,1}}{\partial \vec{a}} & \frac{\partial f_{y,1}}{\partial \vec{a}} & \dots & \frac{\partial f_{x,N}}{\partial \vec{a}} & \frac{\partial f_{y,N}}{\partial \vec{a}}\end{bmatrix}^T = \begin{bmatrix}1 & 0 & -c_{x,1}\\ 0 & 1 & -c_{y,1}\\ \vdots & \vdots & \vdots\\ 1 & 0 & -c_{x,N}\\ 0 & 1 & -c_{y,N}\end{bmatrix} \tag{26}$$

B. Improvement of the calculated virtual images

To ensure a certain accuracy of the estimated virtual depth (resp. virtual image), the distances between the POIs in the micro images and the virtual image of the POI back-projected in these micro images are calculated as given in eq. (27) and (28). If any POI in a group of matched POIs has a distance error (s. eq. (39)) lager than one pixel, we repeat the linear regression but without the POI with the largest distance error. This is done until the remaining POIs do not have a distance error larger than one pixel or until only two POIs remain. In the last case the whole group of matched POIs is deleted.

$$\Delta x_{R,i} = x_{R,i}{}^* - x_{R,i} \tag{27}$$

$$\Delta y_{R,i} = y_{R,i}{}^* - y_{R,i} \tag{28}$$

$$\Delta_{R,i} = \sqrt{\Delta x_{R,i}{}^2 + \Delta y_{R,i}{}^2} \tag{29}$$

After this improvement step, the virtual depth v (resp. the object distance z_c) is calculated. Due to the configuration of our plenoptic camera, only a certain range for the virtual depth is reasonable. From this, a range for the object distance can be determined. In Figure 14 the function $z_c(v)$ is displayed for the camera parameters in Table 1. Only the estimated virtual images points (resp. 3D points) inside this range are classified as valid.

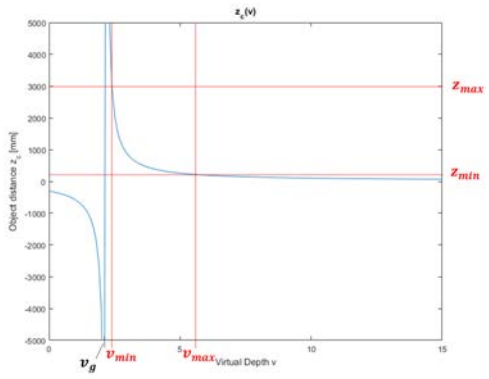


Fig. 14. Function $z_c\,(v)$ with exemplary range for v and z_c

Table 1: Camera parameters

f_L	16.279748091856455 mm
b_{L0}	15.449618357330239 mm
B	0.38300659522738911 mm
d_{lens}	23.306472861260 pixels

VI. Results

To evaluate the described methods, different POI detectors (SIFT, SURF and Harris Corner Detector (HCD)) are used from the openCV library. The estimated object distances for the image displayed in Figure 15 are compared to the values in the depth map generated with the method described in [13]. For the SIFT and SURF methods the default configuration of openCV is used and only the parameter for the amount of POIs is changed so that approximately 2000 POIs are detected. The Harris Corner Detector provides significantly less features for the captured scene (around 500 features).

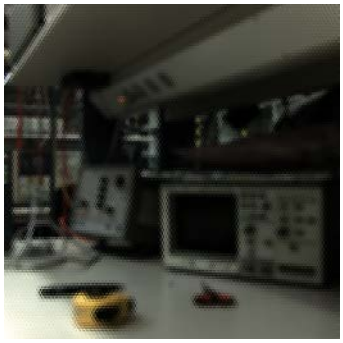


Fig. 15. Raw image used for depth estimation

In the histograms displayed in Figure 16, Figure 17 and Figure 18 the absolute differences between the calculated object distance and the object distance from the depth map of [13] are displayed for the different detectors by their frequency. The HCD provides the best results regarding the absolute error of the object distances with maximum absolute errors around 50 cm.

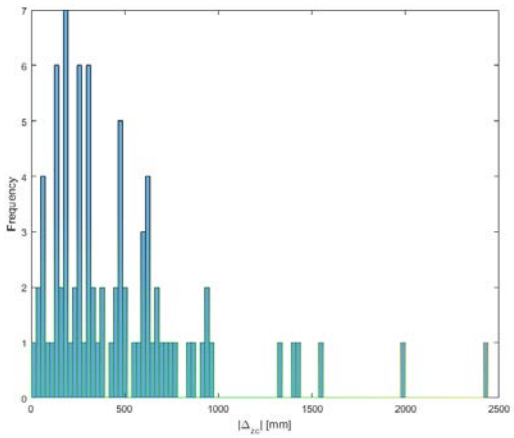


Fig. 16. Histogram of the absolute object distance errors (SIFT detector)

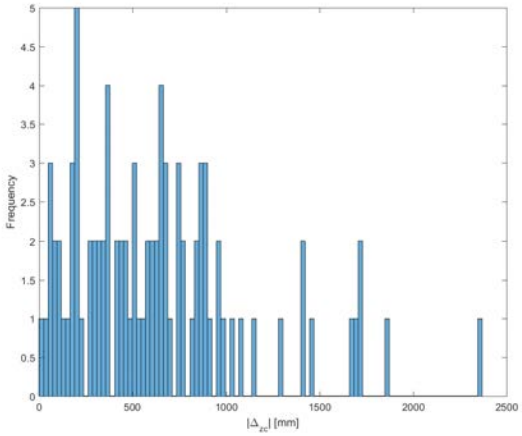


Fig. 17. Histogram of the absolute object distance errors (SURF detector)

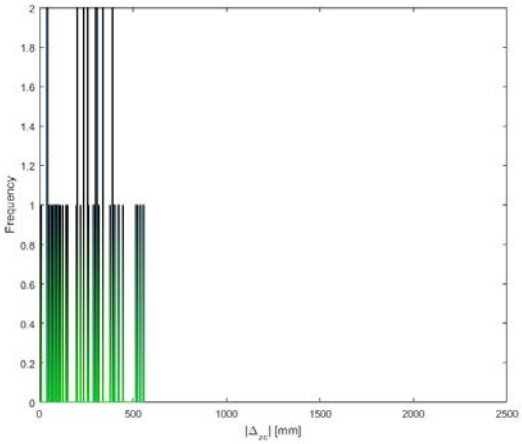


Fig. 18. Histogram of the absolute object distance errors (HCD)

VII. Conclusion

In general, the described approaches for feature matching and depth estimation provide to a certain extent acceptable results for common detectors. Although the HCD has the best results in comparison to SIFT and SURF, the number of corners which can be detected in micro images is too low to display a proper depth map.

By using only the detected POIs for matching and depth estimation, the accuracy of the estimation is limited. In further work, we want to improve the matching of POI by a correlation alongside the epipolar line to detect the corresponding points in the micro images. Furthermore, an edge detector specifically implemented for micro images could outperform SIFT and SURF, because in that case a set of edge points would be matched instead of single points.

A further application of the method described in this paper is to use it for finding matching POI in different raw images, recorded in a video. Then, this method can be used to implement simultaneous localization and mapping (SLAM) directly on raw images, without computing the totally focused image.

Acknowledgment

This research was done in the Bachelor Thesis of Jennifer Konz. It is part of the project Hypermod, which is funded by the Federal Ministry of Education and Research Germany in its program FHProfUnt. We gratefully acknowledge the support.

References

[1] F.E. Ives, "Parallax stereogram and process of making same", 1904.
[2] G. Lippmann, "preuves rversibles donnant la sensation du relief", Journal de Physique Thorique et Applique, Vol. 7, No. 1, 1908.
[3] E.H. Adelson, J.Y.A.Wang, "Single lens stereo with plenoptic camera", IEEE PAMI 14(2), pp. 99-106, 1992.
[4] S.J. Gortler, R. Grzeszczuk, R. Szeliski, M.F. Cohen, "The lumigraph", Proc. 23rd SIGGRAPH, ACM, New York, pp. 43-54, 1996.
[5] R. Ng, "Digital light field photography", PhD thesis, Stanford University, Stanford, USA, 2006.
[6] A. Lunsdaine, T. Georgiev, "Full Resolution Lightfield Rendering", Adobe Technical Report, Adobe Systems Inc., 2008.
[7] C. Perwass, L. Wietzke, "Single lens 3D-camera with extended depthof-field", Human Vision and Electronic Imaging XVII, Burlingame, California, USA, 2012.
[8] N. Zeller, C.A. Noury, F. Quint, C. Teulire, M. Dhome, "Metric calibration of a focused plenoptic camera based on a 3D calibration target", ISPRS Annals, Vol. III-3: 449-456, 2016.
[9] N. Zeller, F. Quint, U. Stilla, "Calibration and accuracy analysis of a focused plenoptic camera", ISPRS Annals, Vol. II-3, pp. 205-212, 2014.
[10] A. Lunsdaine, T. Georgiev, "The focused plenoptic camera", Proc. IEEE Int. Conf. on Computational Photography (ICCP), San Francisco, CA, USA, pp. 1-8, 2009.
[11] Raytrix GmbH, "Raytrix Lightfield Camera Demo Images", 2013.
[12] T. Strutz, *Data Fitting and Uncertainty: A practical introduction to weighted least squares and beyond*, 1st Edition, Vieweg+Teubner, 2011.
[13] N. Zeller, F. Quint, U. Stilla, "Establishing a Probabilistic Depth Map from Focused Plenoptic Cameras", Proceedings, International Conference on 3D Vision (3DV), p. 91-99, Lyon, 2015.

Feature Based RGB-D SLAM for a Plenoptic Camera

Andreas Kühefuß, Niclas Zeller, Franz Quint
Karlsruhe University of Applied Sciences
Moltkestr. 30, 76133 Karlsruhe, Germany
andreas.kuehefuss, niclas.zeller, franz.quint@hs-karlsruhe.de

Uwe Stilla
Technische Universität München
Arcisstr. 21, 80333 Munich, Germany
stilla@tum.de

***Abstract*—This paper presents a method to estimate the camera poses for images of a plenoptic camera. For this, a feature based RGB-D SLAM is used. A new method for matching the features between two images will be presented. Finally the result of the algorithm is compared with the trajectory from the Google Project Tango.**

Index Terms - RGB-D; SLAM; feature based; bundle adjustment; focused plenoptic camera

I. Introduction

With rising computing power SLAM (simultaneous localization and mapping) algorithms will gain more and more importance. In this paper a feature based RGB-D SLAM (red-green-blue-distance SLAM) algorithm for a plenoptic camera is presented. Plenoptic cameras are able to deliver (with a certain accuracy) depth information from a single image, which in turn allows the SLAM algorithm to generate a scaled 3-D map and a scaled trajectory. In our approach the SLAM works on RGB images and corresponding depth maps generated with a Raytrix R5 plenoptic camera.

Several SLAM algorithms are known from literature. A. Davison presented in [1] for example an Extended Kalman filter (EKF) based monocular SLAM that is able to recover a 3D trajectory for a uncontrolled camera with a frame-rate of 30 Hz. A keyframe-based SLAM algorithm has been presented by G. Klein in [2] to recover a 3D trajectory.

II. The Plenoptic Camera

The model for a plenoptic camera described in [3] is used in this paper. A plenoptic camera has, in contrast to a normal pinhole camera, a micro lens array (MLA) between the main lens and the image sensor. This leads to a point from object space being projected, as shown in Fig. 1 not only to a single image point but to several, which are located in different micro images. By finding the points in the micro image which correspond to the same object point, the so called virtual image point can be calculated for each object. Each virtual image point is characterized by an associated virtual depth which is defined as the distance between the virtual image point and the MLA in relation to the distance between the sensor and MLA, cf. eq. (1). Please refer also to Fig. 1 for the definition of the variables.

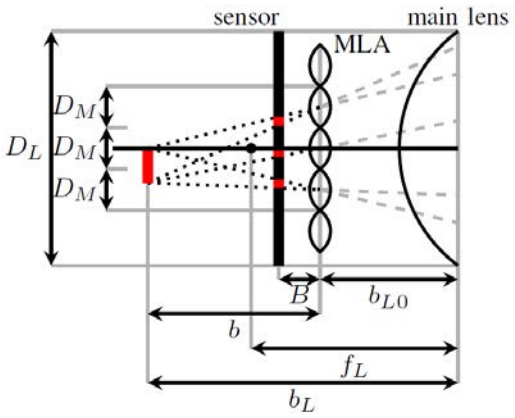


Fig. 1. Optical path inside a plenoptic camera. The MLA projects a single object to different points on the sensor. [3]

$$v = \frac{b}{B} \tag{1}$$

Substituting eq. (1) in the Fresnel-equation for pinhole cameras, the object distance z_C can be calculated from the virtual depth

$$z_C = \left(\frac{1}{f_L} - \frac{1}{B \cdot v - b_{L0}} \right)^{-1}. \tag{2}$$

The object distance is used to project an image coordinate $\boldsymbol{x}_i$ into camera coordinate $\boldsymbol{x}_C$ with

$$\boldsymbol{x}_C = z_C \cdot (K_S \cdot K)^{-1} \cdot \boldsymbol{x}_i. \tag{3}$$

The matrices K_S and K are defined in [3] and contain the intrinsic camera parameters, as shown in (4).

$$\boldsymbol{x}_C = z_C \cdot \left(\begin{pmatrix} s_p^{-1} & 0 & 0 & c_x \\ 0 & s_p^{-1} & 0 & c_y \\ 0 & 0 & B^{-1} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} b_L & 0 & 0 & 0 \\ 0 & b_L & 0 & 0 \\ 0 & 0 & b & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \right)^{-1} \cdot \boldsymbol{x}_i \tag{4}$$

The parameter b_L is the distance of the virtual image to the optical center of the main lens and is calculated using the Fresnel equation for the main lens

$$b_L = \left(\frac{1}{f_L} - \frac{1}{a_L} \right)^{-1}. \tag{5}$$

III. Feature Acquisition

This section describes which feature types are used and how features are matched between images. First the raw images of a plenoptic camera have to be converted into RGB and depth map images. For this, the method presented in [4] is used. After feature extraction, a matching between features in two consecutive images is established.

A. Image acquisition

The scene is recorded with a plenoptic camera. After recording, all raw images are converted to RGB and depth-map images, which in our case have the size 1024 x 1024 pixels. To do this, we use the method presented in [4]. The depth map contains for each pixel a virtual depth, which is coded as a 16-bit grayscale value. Conversion to the virtual depth v cf. eq. (1) is performed as suggested in [5] with

$$v = \frac{1}{1 - \frac{grayscale_{16-bit}}{65535}}. \tag{6}$$

This virtual depth is used together with the internal camera parameters f_L, B and b_{L0}, which have been determined in a previous calibration step, to calculate the object distance z_C with eq. (2). Finally, the camera coordinates of a point can be calculated with eq. (3) out of the image coordinates.

Due to the small baseline between the micro images projected by the MLA, estimation of the virtual depth from a single image of a plenoptic camera is possible within a limited accuracy. Depth estimation can be improved using the larger baseline between successive images of a video. For this, corresponding points have to be matched. This is done with a new method, which we call Slope Matching and which is presented in the following.

B. Slope Matching

Interesting points in the images of the video sequence are detected using the SURF feature detector [6]. After detecting all SURF features in two consecutive images, a match for each feature from the first image is searched in the second. This is done by looking for nearest neighbour with the method presented in [7] and considering the feature quality attributes delivered by the SURF detector. The resulting matches for two consecutive images are shown in Fig. 2.

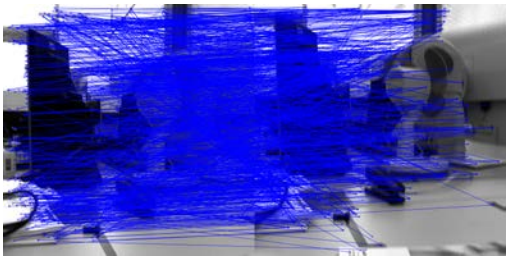


Fig. 2. Resulting matches for two consecutive images, after next neighbor search.

As one can see in Fig. 2 there are many wrong matches which have to be removed. To remove the wrong matches, we just stick the two images side by side and calculate the slope of the line connecting the matched points (blue lines in Fig. 2) with

$$m_i = \frac{y_{i,2} - y_{i,1}}{(x_{i,2} + 1024) - x_{i,1}}. \tag{7}$$

After calculating the slope for each match, the median of all slopes is computed. Each match will be marked as wrong if it doesnt satisfy condition (8), where ϵ is a suitably chosen threshold.

$$m_{median} - \epsilon \leq m_i \leq m_{median} + \epsilon \tag{8}$$

Since the rotation and the translation between two consecutive images of a video is small, the slope for each match should be within the limits given by ϵ. Inequation (8) removes all matches with slopes that differ significantly from the median.

After this operation there still might remain wrong matches, e.g. between a feature from the very left of the first image to a feature of the very right in the second image, but having the slope similar to the median. To remove these wrong matches, the two pictures are stacked one above the other and the previous step is repeated. Mathematically this is done by calculating the inverse slope $m_i^{'}$ of the matches

$$m_i^{'} = \frac{x_{i,2} - x_{i,1}}{(y_{i,2} + 1024) - y_{i,1}}. \tag{9}$$

Again the median will be determined and a constraint similar to the one presented in ineq. (8) removes all matches with a wrong slope. The result of both slope filters is shown in Fig. 3.

Fig. 3. Resulting matches for two consecutive images, after next neighbor search and slope filters.

The resulting matches will be used for pose estimation between two consecutive images.

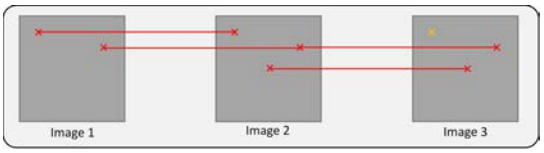


Fig. 4. Feature chain between three images.

The baseline can be further extended, if matches are not restricted to consecutive images. For this we use a feature chain as shown in Fig. 4. If a feature is matched between the first and the second and also between the second and the third image, the feature from the first image is linked to the feature from the third image. The yellow x in the third image shows an feature that could not be matched. The feature chain is broken for this feature.

IV. Pose Estimation

Pose estimation is divided into two parts: First initial poses are estimated between two consecutive images. After estimating enough poses, a local bundle adjustment will be performed.

A. Transformation Matrix

To estimate the pose for two consecutive images, the transformation matrix has to be estimated. The transformation matrix is defined by three rotations and three translations. These six parameters form the camera vector $\boldsymbol{a}$ as shown in eq. (10) and need to be estimated for every new image.

$$\boldsymbol{a} = \begin{pmatrix} \alpha & \beta & \gamma & t_x & t_y & t_z \end{pmatrix}^T \tag{10}$$

B. Initial pose estimation

Instead of estimating the twelve (mutually dependent) elements of the transformation matrix in homogeneous coordinates between the camera coordinate systems of two images, we estimate directly the six elements of $\boldsymbol{a}$ with an iterative procedure.

The first pose is estimated using the results of the slope matching shown in Fig. 3. First the features are projected from the image coordinate system in the camera coordinate system using eq. (4). Now they can be transformed from the camera coordinates corresponding to the first image in the camera coordinates corresponding to the second image. This is done by multiplying the coordinates with the transformation matrix T as shown in (11).

$$\boldsymbol{x}_C^{'} = T \cdot \boldsymbol{x}_C \tag{11}$$

The transformation matrix T is the one which is determined by the vector $\boldsymbol{a}$ to be estimated. At the beginning of the iterative process, the matrix is initialized with the identity matrix.

The coordinates of the features from the first image in the camera coordinate system of the second image can be back-projected in the image coordinate system if the second image by using the inverse of eq. (3), as shown in (12).

$$\boldsymbol{x}_i^{'} = \frac{1}{z_C} \cdot (K_S \cdot K) \cdot \boldsymbol{x}_C^{'} \tag{12}$$

Note that the matrix K_s is the same for both images since the camera remains unchanged. The matrix K however needs to be adapted, since it depends on b_L, which is a function of the virtual depth (see eq. (5)). After transforming every feature of the first image into the second image (initially with the identity matrix), there will remain a reprojection error for each match, expressed as the difference between the position vector of a feature from the first image transformed to the second image and its match in the second image (eq. (13)). The position vectors are three-dimensional since they describe points in the virtual image. The third dimension in the virtual image is the virtual depth.

$$\boldsymbol{r}_j(\boldsymbol{a}) = \boldsymbol{x}_{i_j}^{'} - \boldsymbol{x}_{i_j} = \begin{pmatrix} x_{i_j}^{'} \\ y_{i_j}^{'} \\ v_{i_j}^{'} \\ 1 \end{pmatrix} - \begin{pmatrix} x_{i_j} \\ y_{i_j} \\ v_{i_j} \\ 1 \end{pmatrix} \tag{13}$$

The goal is to find the camera vector $\boldsymbol{a}$ (and thus implicitly the transformation matrix T) which minimizes the mean reprojection error over all matches:

$$\boldsymbol{a}^{'} = \min_{\boldsymbol{a}} \sum_{j=1}^{n} ||\boldsymbol{r}_j(\boldsymbol{a})|| \tag{14}$$

To solve the minimization problem, the Gauss-Newton algorithm is applied. To minimize the impact of outliers, Tukeys biweight cost function is used [8]. This function suppresses outliers by weighting them with zero, as shown in (15).

$$w_{Tb} = \begin{cases} \left(1 - \dfrac{\boldsymbol{r}_j^2}{\sigma^2}\right)^2 & |\boldsymbol{r}_j| \leq \sigma \\ 0 & |\boldsymbol{r}_j| > \sigma \end{cases} \tag{15}$$

By weighting large outliers with zero, wrong matches will not influence the estimation. For calculating the weight w_{Tb}, the empirical standard deviation σ of the errors is used.

C. Bundle adjustment

To optimize the camera position for more than just two images a local bundle adjustment [9] is performed. The bundle adjustment is called local since is not carried out over all images (frames) of the sequence, but only from a so called key-frame to the next key-frame. A new key-frame is set, if the number of matches in a feature chain (see Figure 4) falls below a threshold. In our experiments we have set a new key-frame when the number of matches has fallen below a threshold. However, even if the number of matches exceeds the threshold, every sixth frame is set as a key-frame to ensure that

the bundle adjustment is executed with a sufficient frequency to prevent a drift.

The m images contained between the two key-frames participate in the local bundle adjustment with their camera vectors $\boldsymbol{a}_k$. Consider that for a total of n world points $\boldsymbol{b}_j$ matches have been found. The world points are given by their homogeneous coordinate vector

$$\boldsymbol{b}_j = \begin{pmatrix} x_{w_j} & y_{w_j} & z_{w_j} & 1 \end{pmatrix}^T. \tag{16}$$

The residual error of the world points, projected in the corresponding frames depends of course on the camera vectors $\boldsymbol{a}_k$ as shown in eq. (17)

$$\boldsymbol{r}_{jk}\left(\boldsymbol{a}_k, \boldsymbol{b}_j\right) = \boldsymbol{x}^{'}_{i_{jk}}\left(\boldsymbol{a}_k, \boldsymbol{b}_j\right) - \boldsymbol{x}_{i_{jk}}. \tag{17}$$

The image coordinates used to calculate the error are obtained by projecting the world points in the corresponding frames using eq. (18), which is similar to eq. (12)

$$\boldsymbol{x}^{'}_{i_{jk}}\left(\boldsymbol{a}_k, \boldsymbol{b}_j\right) = T_{a_k} \cdot \frac{1}{z_{w_k}} \cdot \left(K_S \cdot K\right) \cdot \boldsymbol{b}_j. \tag{18}$$

The task of the bundle adjustment is to find the parameters which minimize the mean error over the n world points and m camera positions with

$$\left\{\boldsymbol{a}^{'}_k, \boldsymbol{b}^{'}_j\right\} = \min_{\boldsymbol{a}_k, \boldsymbol{b}_j} \sum_{k=1}^{m} \sum_{j=1}^{n} \left|\left|\boldsymbol{r}_{jk}\left(\boldsymbol{a}_k, \boldsymbol{b}_j\right)\right|\right|. \tag{19}$$

To minimize the effect of outliers, we use in this step also Tukeys biweighted function (15).

V. Evaluation

To evaluate the presented RGB-D SLAM algorithm, two experiment setups have been made. The first is a set of 40 images with known ground truth and the second is a set of 1974 images where the trajectory is compared with the one estimated by the Project Tango of Google. Example images for both experiments are shown in Fig. 5 and Fig. 6.

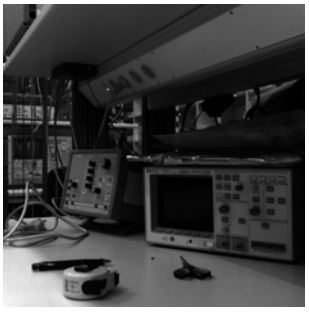


Fig. 5. Image 20 / 40 of the first experiment

Fig. 6. Image 800 / 1974 of the second experiment

A. Comparison with ground truth

The first experiment contains a small number of 40 images which have been recorded in a known grid. First, the camera was moved 19 cm to the right in one cm steps. After that the camera was moved 10 cm back, again in steps of 1 cm each. At last, the camera was moved 19 cm back to the left, in 2 cm steps. The last step amounted again to one cm. Thus the camera was at the end of the movement 10 cm behind its start position. The result of the RGB-D SLAM without a bundle adjustment is shown in Fig. 7. The blue cones mark the position and the orientation of the camera as estimated by our SLAM algorithm without bundle adjustment. The white cones show the ground truth.

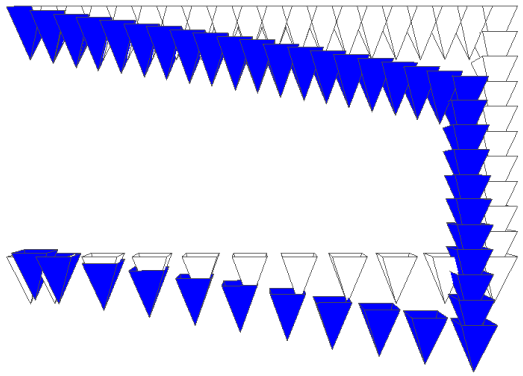


Fig. 7. Result for the first experiment, without bundle adjusment. Blue cones mark the estimated trajectory, the white ones the ground truth.

As one can see in Fig. 7 there is a big drift when not using local bundle adjustment. The trajectory for the same images with local bundle adjustment is shown in Fig. 8.

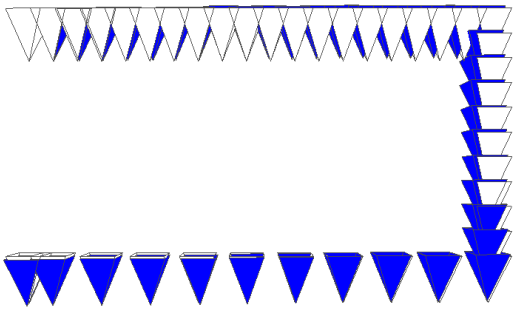


Fig. 8. Result for the first experiment, with bundle adjusment. Blue cones mark the estimated trajectory, the white ones the ground truth.

As one can see in Fig. 8 the estimated trajectory fits well to the known ground truth. The drift error is compensated by the local bundle adjustment.

B. Comparison with Google Project Tango

The second experiment was done on a longer trajectory, using 1974 images. To evaluate the result, the experiment was recorded in parallel with the plenoptic camera and with a tablet

fixed to the plenoptic camera. On the tablet Google Project Tango was running. The software of Project Tango estimates a 3D trajectory using a accelerometer and a monocular camera [10]. The trajectories resulting from our RGB-D SLAM and from Project Tango are compared qualitatively. Of course the result of Google Project Tango is not the actual ground truth, but also subject to errors. However, it is a quite good estimate.

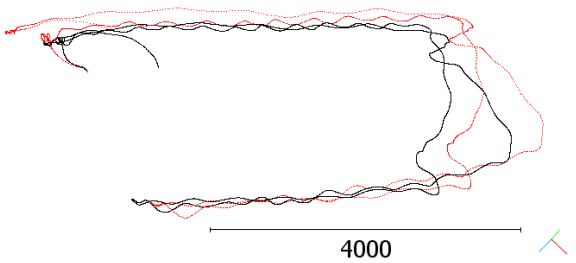


Fig. 9. Result for the second experiment. Red trajectory is from RGB-D SLAM and the black one is from Google Project Tango. Scale is in mm.

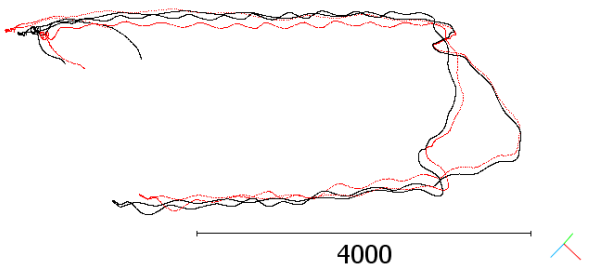


Fig. 10. Result for the second experiment. Red trajectory is from RGB-D SLAM with corrected scale and the black one is from Google Project Tango. Scale is in mm.

The result of the second experiment is shown in Fig. 9. The estimated trajectory matches qualitatively with the trajectory from Google Project Tango, but as one can see there is a scaling error in the data. For better comparison we estimated the scale error using CloudCompare [11]. After correcting our trajectory with the estimated scale, which amounted to 1.125, the comparison can be performed easier. It is shown in Fig. 10.

As one can see in Fig. 10, the two trajectories fit quite well. There is still a very small offset, observable in the top left of the figure. To minimize also this offset a global bundle adjustment could be performed.

VI. Conclusion

The presented method works quite well for typical videos recorded with the plenoptic camera. The matching method supplies good and enough matches for the trajectory estimation to work stable, even for a long scene as shown in the second experiment. Nevertheless there remain some unsolved problems like the scaling error shown in Figure 9. This error has been corrected up to now only interactively and an automated procedure has to be implemented. Furthermore would a global bundle adjustment using the key-frames only help to achieve higher stability against drift.

VII. Acknowledgment

This research was done in the Bachelor Thesis of Andreas Kühefuß. It is part of the project Mosyko3D, which is funded by the Baden-Wrttemberg-Stiftung in its program Photonics, Microelectronics, Information Technology. We gratefully acknowledge the support.

References

[1] A. Davison, I. Reid, N. Molton and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM" IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 29, nr. 6, p. 1-16, 2007.
[2] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces", Proc. International Symposium on Mixed and Augmented Reality, ISMAR'07, Nara, 2007.
[3] N. Zeller, F. Quint and M. Stterlin, "Investigating Mathematical Models for Focused Plenoptic Cameras", Proc. of International Symposium on Electronics and Telecommunications ISETC2016, p. 285-288, Timisoara, 2016.
[4] N. Zeller, F. Quint, U. Stilla, "Establishing a Probabilisitc Depth Map from Focused Plenoptic Cameras", Proc. International Conference on 3D Vision (3DV), p. 91-99, Lyon, 2015.
[5] "Converting Depth Data," *www.raytrix.de*, Raytrix GmbH, 2013.
[6] H. Bay, T. Tuytelaars, L. Van Gool, "SURF: Speeded Up Robust Features", Journal on Computer vision and image understanding, vol. 110, nr. 3, p.346-359.
[7] M. Muja, D. Lowe, "Fast Approximate Nearest Neighbors With Automatic Algorithm Configuration", International Conference on Computer Vision Theory and Application VISSAPP'09, INSTICC Press, p. 331-340, 2009.
[8] R. Maronna, D. Martin, V. Yohai: Robust statistics: Theory and methods, Wiley Series in Probability and Statistics, Chichester: John Wiley & Sons, 2006.
[9] R. Hartley, A. Zisserman: Multiple View Geometry in Computer Vision. Cambridge University press, 2004.
[10] "Google Project Tango", *www.google.com/tango/*, 2016.
[11] "CloudCompare", *www.cloudcompare.org*, 2016.

A Comparative Study of Data Clustering Algorithms

Ankita Agrawal and Artur Schmidt
Institue for Artificial Intelligence
University of Applied Sciences, Ravensburg-Weingarten, Germany
Email: agrawala@hs-weingarten.de

***Abstract*—Cluster analysis is important for understanding the structure of a particular data set. This paper compares some known clustering algorithms. The criteria for comparison are usability (choice of initial input parameters), simplicity, accuracy and computational complexity. The aim is to search for an optimal clustering algorithm by evaluating its performance on data sets from different fields of application. Other related topics, scaling and cluster validation are also discussed.**

I. Introduction

The goal of this paper is to understand various clustering algorithms and to provide an unbiased evaluation based on comparison criteria. Clustering is the grouping of data into clusters such that the data vectors in one cluster are more similar to each other than those in another cluster. A data set $X = (x_1, \ldots, x_n)$ consists of n d-dimensional vectors, $x_i \in \mathcal{X} \subseteq \mathbb{R}^d$.

Clustering is the unsupervised learning problem of finding a structure in a set of unlabeled data, where the correct clusters are not previously defined. It is used for data mining techniques in many fields, such as, medicine [1], psychology [2], marketing [3], biology [4], linguistics [5]. Clustering is used to find internal structures in the data, such as, reoccurring patterns and anomalies. Also, it is usually the pre-processing step for many other algorithms, such as artificial intelligence techniques. For comparison, we chose popular clustering algorithms based on different working principles as given below.

a) Centroid-based (K-means, Kernel k-means, Spectral)
b) Distribution-based (EM Algorithm)
c) Connectivity-based (Hierarchical)
d) Density-based (DBSCAN, OPTICS)

II. Metrics

A. Distance

The performance of a clustering algorithm depends critically on a good distance metric in the input space that reflects the relationship between a given data. A metric is a function defining the distance between each pair of elements or vectors in a data set. For example, the Minkowski distance given by, $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0^+$, where $\mathbb{R}_0^+ = \{t \in \mathbb{R} \mid t \geq 0\}$.

$$d(x, y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}$$

is a generalized form. The widely used distances are Manhattan-distance ($p = 1$), Euclidean-distance ($p = 2$) and Chebyshev-distance ($p = \infty$). We used primarily the Euclidean distance as it is translation and rotation invariant.

B. Scaling

Scaling or data normalization is a pre-processing step for clustering. The range of data values for different dimensions may vary greatly, thereby causing one or more dimensions to strongly influence the result. Scaling ensures the proportional contribution of each dimension.

We used Min-Max Scaling (rescaling the values of all dimensions to a target range), defined as,

$$x_{i,norm}^l = \frac{x_i^l - \min\limits_{j \in [1,n]} (x_j^l)}{\max\limits_{j \in [1,n]} (x_j^l) - \min\limits_{j \in [1,n]} (x_j^l)}\ , \quad \text{where } l \in [1, d] \subseteq \mathbb{N}$$

and Standardization (values of each dimension have zero-mean and unit-variance), defined as,

$$x_{i,norm} = \frac{x_i - \text{mean}(x)}{\text{var}(x)}$$

For comparison, we also used Multi-Dimensional Scaling (MDS) [6] to detect relevant dimensions that represent the similarities within data.

III. Cluster Validation

Every clustering algorithm can find a set of clusters, whether the structure exists in the data or not. Even the order of data or choice of different input parameters for the same algorithm can lead to very different clustering results. Therefore, validation of the cluster quality becomes very important. The silhouette criterion for evaluating the clustering quality is defined as follows.

A. Silhouette Criterion

The Silhouette criterion [7] takes as input the final set of clusters C obtained from a clustering algorithm. It measures how closely the data matches to other data within its own cluster and how loosely it matches to data in other clusters. It returns the value S on the interval $[-1, +1]$ where $+1$ indicates that the data is appropriately clustered and vice-versa.

$$s(x_i) = \frac{b(x_i) - a(x_i)}{max\{a(x_i), b(x_i)\}} \text{ and } S = \frac{1}{n} \sum_{i=1}^{n} s(x_i)$$

Here, $a(x_i)$ is the average distance of x_i to other data within the same cluster and $b(x_i)$ is the lowest average distance of x_i to any other cluster.

IV. Clustering Algorithms

The algorithms take as input the data set $X \in \mathcal{X}^n$. The output is $1 \leq k \leq n$ clusters $C = \{C_1, \ldots, C_k\}$, where C_i is a mutually exclusive subset of X.

A. K-Means

We implement the k-means algorithm as proposed in [8], which partitions the data into k clusters by iteratively minimizing the within-cluster sum of squares with k chosen manually a priori. The algorithm alternates between two steps, firstly assigning each data vector to the cluster with the nearest cluster mean and secondly, re-calculating the new mean for the updated clusters. The objective is to find the clusters such that, $\underset{C}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$, where $\mu_i \in \mathcal{X}$ is the mean of vectors in C_i.
We analyzed the following two methods to select the initial cluster centers. First, randomly choose k vectors from the data set X as initial means. Second, k-means++ [9] algorithm is used, where a subsequent cluster center in an iteration is chosen with probability proportional to its squared distance from the existing cluster centers.

B. Kernel K-Means

Kernel k-means applies k-means algorithm in kernel space as proposed in [10]. The data points are mapped from input space to a higher dimensional space with ϕ: $\mathcal{X} \rightarrow \mathcal{H}$ and then k-means is performed in this transformed space. The k-means equation changes to $\underset{C}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in C_i} \|\phi(x) - \tilde{\mu}_i\|^2$, where $\tilde{\mu}_i \in \mathcal{H}$ is mean of vectors in C_i in the transformed space.

C. Spectral Clustering

Spectral clustering [11] uses eigenvectors of similarity matrix (also known as kernel or Gram matrix) calculated from the input data. A dimensionality reduction is performed by taking the k largest eigenvectors as columns to obtain a $n \times k$ matrix. Clustering is performed on this reduced matrix with new input vectors $Y = (y_1, \ldots, y_n)$, where y_i is the reduced transformed vector of x_i in the higher dimensional space $\mathcal{H}$.

D. Expectation-Maximization Algorithm

EM algorithm [12], assumes that the data set can be modeled as a combination of multivariate normal distributions. It finds the maximum-likelihood estimate of the parameters of an underlying distribution in a statistical model for a data set X. Here, besides initial cluster centroids, initial co-variance matrices and probability distribution of each cluster are also required as input parameters.

E. Hierarchical Clustering

We use the single-linkage variant of the agglomerative hierarchical clustering suggested independently by McQuitty [13] and Sneath [14]. Initially each input vector is in its own cluster. At each step, the two clusters separated by the least distance are merged. The distance d between two clusters is defined as $d(C_i, C_j) = \min_{p \in C_i, q \in C_j} d(p, q)$.
The process goes on until the termination criterion is reached (in our case, number of desired clusters).

F. DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [15] is an algorithm which forms clusters based on the local density of data points. It requires two input parameters, $MinPts$ (the number of points required to form a cluster) and ϵ (neighborhood size). The data points are assigned to a cluster if the minimum number of neighbouring points specified at the beginning are present, otherwise it is classified as noise.

G. OPTICS

OPTICS (Ordering Points To Identify the Clustering Structure) [16] builds upon DBSCAN so that only one parameter is needed for clustering. It addresses the major drawback of DBSCAN clustering, which is, not being able to find clusters with different densities. The input parameters are $MinPts$ and γ (maximum distance to consider for a cluster). The value of γ influences the time complexity of the algorithm and the cluster quality. However, in OPTICS, the output is difficult to interpret, especially with large data count. The output is a particular ordered list of data from the input data set according to their smallest reachability distance, which is the distance of a data point x_i to it's $MinPts$-th closest point.

V. Experiments

A. Datasets

For evaluating performance of the clustering algorithms, we used pre-labeled data sets available in [17] and 2-dimensional synthetic data from [18]. The data sets, summarized in Table I, were chosen from various fields of application, so as to find a general plausible algorithm.

B. Comparison of Algorithms

This section details how the experiments were carried out. Four variants of data are used, unscaled and scaled data using the three techniques mentioned in Section II-B. The clustering algorithms are then executed with the initial values required by the individual algorithms. Below are the additional details about the algorithms in our implementation.
The initial centroids in k-means algorithm are selected from the data using k-means++ algorithm, as it distributes the centroids in the data space, increasing the probability for better results. For kernel k-means and spectral clustering, Gaussian and polynomial kernels are used to form the similarity matrix. In EM algorithm, k-means with k-means++ initialization is applied to find the initial clusters and then EM algorithm

TABLE I: Details of the Data Sets

Data set	Data Count	Dimensions	Num Class	Description
Appendicitis	473	15	2	Medicine, diagnosis of appendicitis [20]
Iris	150	4	3	Botany, iris plant types
Movement Libra	360	90	15	Video, hand movement in Brazilian sign language
Seeds	210	7	3	Agriculture, three different varieties of wheat
Thyroid	215	5	3	Medicine, analysis of thyroid disease
User Knowledge Modeling	403	5	4	Knowledge, student knowledge level [21]
Wholesale Customers	440	7	2	Business, clients of a wholesale distributor [22]
Wine	178	13	3	Food, wine origin/type
Yeast	1484	8	10	Food, cellular localization sites of proteins
Aggregation	788	2	7	Synthetic data
Compound	399	2	6	Synthetic data
Pathbased	300	2	3	Synthetic data
R15	600	2	15	Synthetic data

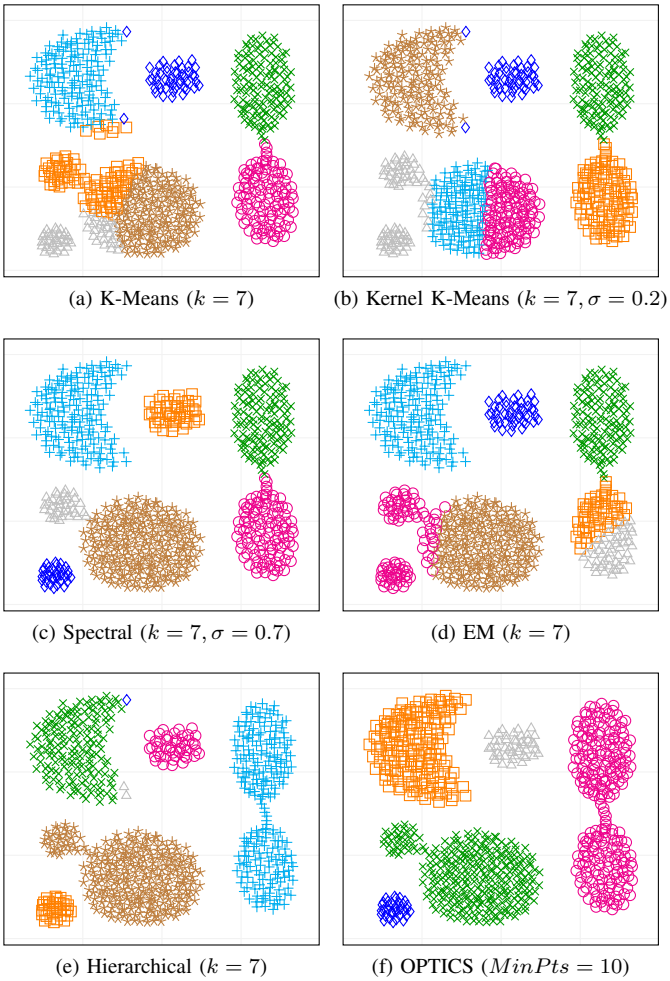
is carried out to optimize the log likelihood function. To overcome the problem of an infinite log value resulting from a singular covariance matrix caused by numerical computation of matrix determinant, a random value between 0 and 0.01 is added repeatedly to the diagonal elements in the covariance matrix till a positive determinant value is obtained.

In OPTICS, the value of γ is set to $\max_{x_i,x_j} d(x_i, x_j)$, eliminating the need for a second input parameter and the clusters adjust themselves according to the density of data points in a particular region. In the output list, the data having low reachability are aligned together, showing that they belong to the same cluster. When the reachability distance shows a jump, it indicates the start of another cluster. Our implementation looks for this sudden jump by comparing the relative difference in distances of current datum in the list to its previous two data, thus selecting the resulting clusters.

Each algorithm is executed five times for each data set and the average of all results (clustering accuracy) along with the variance is taken as the final result for an algorithm on a dataset. The optimal values for some initial parameters are found using cross-validation, such as σ in Gaussian kernel for kernel k-means and spectral clustering. The details about the clustering algorithms are listed in Table II.

VI. Results

In this section, the results obtained for the algorithms are discussed. Figure 1 shows the clustering results on the Aggregation data set. We can see that k-means linearly separates the data whereas spectral is able to cluster the data perfectly. Hierarchical clustering and OPTICS give similar result. The two connected clusters on the right side and bottom left corner are combined as one cluster. As mentioned in Table II, it can be seen that a large k does not affect the result of hierarchical clustering. Since, the cluster number is given as 7 initially, only one (in blue) and two (in gray) data points are assigned to the remaining two clusters, not much affecting the error rate.



(a) K-Means ($k = 7$) (b) Kernel K-Means ($k = 7, \sigma = 0.2$)

(c) Spectral ($k = 7, \sigma = 0.7$) (d) EM ($k = 7$)

(e) Hierarchical ($k = 7$) (f) OPTICS ($MinPts = 10$)

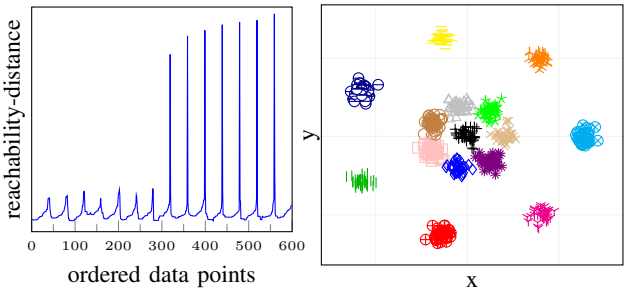
Fig. 1: Clustering results on the Aggregation data set. Each cluster is depicted by a different color. The input parameter values for each algorithm are also mentioned. For Kernel k-means and Spectral algorithms, a Gaussian kernel is used.

Figure 2 and 3 represent the output of OPTICS algorithm and its interpretation as clusters. In Figure 2a, it can be easily seen that there are 15 clusters. Here, it is also not difficult to convert the list to clusters, as can be seen in Figure 2b. The R15 data set is correctly clustered by all the algorithms. On the other hand, Figure 3a shows that the first cluster can end either around 50 or 140 data points. It is not a trivial task to write a generalized algorithm that could convert the list to clusters appropriately. Even if the list is ordered correctly, the conversion may not be accurate. The results of DBSCAN are worse than OPTICS and hence, are not discussed here.

The clustering results for the Compound data set are shown in Figure 4. Here, spectral clustering does not cluster the data in the lower left corner correctly because the clusters consist of

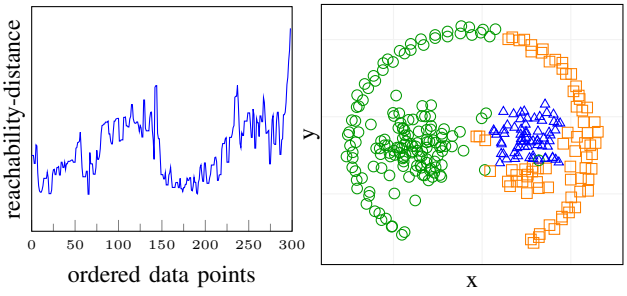
TABLE II: Properties of Clustering Algorithms and their Complexity (n=data count, k=number of clusters, d=dimensions, i=number of iterations)

Algorithm	Input Parameters	Constraints	Positives	Complexity
K-means	# of clusters, initial k centroids	linear separation, local minima, no outlier detection, very sensitive to initial centroids	easy to understand and implement, fast if k is small, even with large data	$O(nkdi)$
Kernel k-means	# of clusters, k centroids, standard deviation σ (gaussian), degree d and constant c (polynomial kernel)	slow due to computation of kernel matrix, esp. for large data, requires tuning of kernel parameters	k-means applied in kernel space, minimizes clustering error in the transformed space	$O(n^2 + nkdi)$
Spectral	# of clusters, k centroids, standard deviation σ (gaussian), degree d and constant c (polynomial kernel)	computationally expensive due to computation of kernel matrix and eigen values	kernel PCA to reduce dimensionality, effective for both low- and high-dimensional data	$O(n^3)$
EM	# of clusters, k means, k variance, k probability distribution	numerical problems	fast given good initialization, effective for high-dimensional data	$O(n^2 + nkdi)$
Hierarchical	# of clusters	sensitive to order of points	fast, simple, large k does not affect the result	$O(n^2)$
DBSCAN	neighbors $MinPts$, distance ϵ	can not identify varied density clusters	insensitive to order of points and noise	$O(n^2)$
OPTICS	neighbors $MinPts$, max. distance γ	output as ordered list and not clusters	filter noise, can detect varied density clusters.	$O(n^2)$



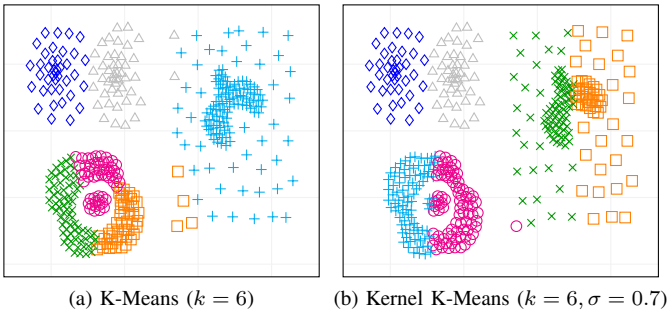
(a) Ordered Output List (b) Interpreted Clusters

Fig. 2: The clustering output of OPTICS for the R15 data set with $MinPts = 10$ is shown in (a). The $reachability\ distance$ for each data point is plotted on the $y-axis$. The interpretation of the list into clusters is shown in (b).



(a) Ordered Output List (b) Interpreted Clusters

Fig. 3: The clustering output of OPTICS for the Path-based data set with $MinPts = 3$ is shown in (a). The $reachability\ distance$ for each data point is plotted on the $y-axis$. The interpretation of the list into clusters is shown in (b).



(a) K-Means ($k = 6$) (b) Kernel K-Means ($k = 6, \sigma = 0.7$)

(c) Spectral ($k = 6, \sigma = 0.7$) (d) EM ($k = 6$)

(e) Hierarchical ($k = 6$) (f) OPTICS ($MinPts = 4$)

Fig. 4: Clustering results on the Compound data set.

different structures. Hierarchical clustering forms appropriate clusters based on the distance function but not according to the correct labels. OPTICS algorithm identifies the clusters accurately except the ones in the upper left corner. The other algorithms do not give satisfying results.

The results for the multi-dimensional data sets are presented in Table III. It contains sub-parts for the clustering performed on data with different scaling methods. Each element of the table represents average percentage of data points that are correctly clustered over multiple executions of the algorithms on each data set and variance in the results. The results for hierarchical clustering and OPTICS algorithm remain same for each iteration, and hence, the variance is not mentioned. For other algorithms, the results in each iteration vary with different initial parameters, so the variance is relevant. The combination of higher percentages and lower variances for each data set is highlighted in the table. The average for unscaled data is $65.1 \pm 2.4\%$, for MinMax scaling $67.4 \pm 1.9\%$, $57.0 \pm 2.5\%$ for Standardization and $63.8 \pm 2.1\%$ for Multi-Dimensional scaling. It can be seen that for almost all data sets, there is $6 - 7\%$ drop in the clustering accuracy with Standardization. On the other hand, with MinMax scaling, the clustering result either improves or stays the same. We can also see that the k-means, kernel k-means, hierarchical clustering and OPTICS algorithm either have no best results or only for one or two data sets. K-means works only good for linearly separable data. Kernel k-means performs only slightly better than k-means. Hierarchical clustering yields good results for 2-dimensional data but does not handle high-dimensional data well. It is found that silhouette criterion does not provide the optimal cluster count. This is because it uses Euclidean distance, which prefers linearly-separable spherical structures. We executed silhouette on the top of EM algorithm for the data sets and the results got worse. E.g. for the Iris data set, silhouette returned the optimal cluster count as 2 instead of 3 and the accuracy reduced from 98% to 66.7%. Due to the lack of an efficient algorithm to convert the output ordered list to clusters, OPTICS often produces high error rate in final clustering result. In general, spectral clustering gives best results as it applies PCA to data in the transformed high-dimensional space to extract those dimensions having the most important information about the data set. EM algorithm, which does not give good results for two-dimensional data and does not converge for Movement Libra data set, gives the best results for high-dimensional data. It optimizes the Gaussian distribution for each dimension to find clusters with different shapes in a data set, thereby improving the result.

VII. Conclusion

The results show that overall Spectral clustering and EM algorithm give the best results for most real data sets. However, Spectral clustering is computationally expensive due to kernel matrix and eigenvalue calculation and is very slow for large data sets. Therefore, it is recommended to use EM algorithm where the initial clusters are given by k-means with k-means++ algorithm.

MinMax Scaling is fast and also has the highest average percentage for all datasets, making it the favorite choice among the scaling methods.

References

[1] R. Xu and D. C. Wunsch, *Clustering Algorithms in Biomedical Research: A Review*, Biomedical Engineering, IEEE Reviews, 3, 120-154, 2010.

[2] J. Clatworthy, D. Buick, M. Hankins, J. Weinman and R. Horne, *The use and reporting of cluster analysis in health psychology: A review*, British Journal of Health Psychology, 10, 329-358, 2005.

[3] H. Lai and T.-C. Yang, *A group-based inference approach to customized marketing on the Web integrating clustering and association rules techniques*, System Sciences, Hawaii International Conference, 10, 2000.

[4] A. C. Tan and D. Gilbert, *An Empirical Comparison of Supervised Machine Learning Techniques in Bioinformatics*, APBC 2003.

[5] V. Hatzivassiloglou, L. Gravano and A. Maganti *An investigation of linguistic features and clustering algorithms for topical document clustering*, SIGIR, 224-231, 2000.

[6] J. B. Kruskal and M. Wish, *Multidimensional Scaling, Sage University Paper series on Quantitative Application in the Social Sciences*, 7-11, 1978.

[7] P. J. Rousseuw, *Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis*, Computational and Applied Mathematics, 20, 53-65, 1987.

[8] S. P. Lloyd, *Least Squares Quantization in PCM*, IEEE Transactions on Information Theory, 28(2), 129-137, 1982.

[9] D. Arthur and S. Vassilvitskii, *k-means++: the advantages of careful seeding*, ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics Philadelphia, USA, 1027-1035, 2007.

[10] B. Schlkopf, A. Smola and K.-R. Mller, *Nonlinear Component Analysis as a Kernel Eigenvalue Problem*, Neural Computation, 10, 1299-1319, 1998.

[11] A. Y. Ng, M. I. Jordan and Y. Weiss, *On spectral clustering: Analysis and an algorithm*, Advances in Neural Information Processing Systems, Cambridge, MA, 14, 849-856, 2002.

[12] A. P. Dempster, N. M.Laird and D. B. Rubin, *Maximum Likelihood from Incomplete Data via the EM Algorithm*, Journal of the Royal Statistical Society, Series B (Methodological), 39(1), 1-38, 1977.

[13] L. L. McQuitty, *Elementary Linkage Analysis for Isolating Orthogonal and Oblique Types and Typal Relevancies*, Educational and Psychological Measurement, 17, 207-229, 1957.

[14] P. H. A. Sneath, *The Application of Computers to Taxonomy*, Journal of General Microbiology, 17, 201-226, 1957.

[15] M. Ester, H. Kriegel, J. Sander and X. Xu, *A density-based algorithm for discovering clusters in large spatial databases with noise*, KDD, 226-231, 1996.

[16] M. Ankerst, M. M. Breunig, H. Kriegel and J. Sander, *OPTICS: Ordering Points To Identify the Clustering Structure*, ACM SIGMOD international conference on Management of data, 49-60, 1999.

[17] M. Lichman, *UCI Machine Learning Repository* http://archive.ics.uci.edu/ml, Irvine, CA: University of California, 2013.

[18] School of Computing, University of Eastern Finland, *Clustering datasets*, Retrieved from http://cs.joensuu.fi/sipu/datasets/.

[19] W. Ertel and M. Schramm, *Combining Expert Knowledge and Data via Probabilistic Rules with an Application to Medical Diagnosis*, In UAI-2000 Workshop on Fusion of Domain Knowledge with Data for Decision Support, Stanford CA, 2000.

[20] W. Ertel and M. Schramm, *Combining Data and Knowledge by MaxEnt-Optimization of Probability Distributions*, In PKDD'99, Prague, 1704 of LNCS, 323-328, 1999.

[21] H. T. Kahraman, S. Sagiroglu and I. Colak, *Developing intuitive knowledge classifier and modeling of users' domain dependent data in web*, Knowledge Based Systems, 37, 283-295, 2013.

[22] N. Abreu, *Analise do perfil do cliente Recheio e desenvolvimento de um sistema promocional*, ISCTE-IUL, Lisbon, 2011.

[23] W. Jiang and F. L. Chung, *Transfer Spectral Clustering*, ECML PKDD, 789-803, 2012.

[24] B. Boden, M. Ester and T. Seidl, *Density-Based Subspace Clustering in Heterogeneous Networks*, ECML PKDD (1), 149-164, 2014.

[25] G. Tzortzis and A. Likas, *Ratio-based Multiple Kernel Clustering*, ECML PKDD (3), 241-257, 2014.

[26] Y. Ren, U. Kamath, C. Domeniconi and G. Zhang, *Boosted Mean Shift Clustering*, ECML PKDD (2), 646-661, 2014.

TABLE III: Clustering Results - Correctly classified data along with variance over multiple runs of an algorithm for a data set (in %). The last column gives the mean percentage accuracy for an algorithm (average of the columns in a row).

	Appendicitis	**Iris**	**Libra**	**Seeds**	**Thyroid**	**User K.M.**	**Wholesale**	**Wine**	**Yeast**	
Scaling - None										
K-means	**64.6 ± 0.6**	88.7 ± 0	47.6 ± 2.4	89.1 ± 0	80.7 ± 5.4	56.4 ± 8.3	74.9 ± 2.2	68.8 ± 1.4	50.6 ± 2.7	69.0 ± 2.5
K. k-means (gaussian)	**64.4 ± 0.8**	87.8 ± 1.5	46.1 ± 3.1	**89.3 ± 0.2**	81.3 ± 4.8	59.1 ± 5.7	73.5 ± 6.9	69.7 ± 0.5	**52.2 ± 2**	69.3 ± 2.8
K. k-means (polynomial)	**64.5 ± 0.7**	86.3 ± 3	46.8 ± 2.1	**89.3 ± 0.2**	81.1 ± 5	56.9 ± 7.9	71.9 ± 8.5	69.7 ± 0.5	50.8 ± 2.6	68.6 ± 3.4
Spectral (gaussian)	57 ± 0	89.6 ± 2.4	48.3 ± 4.5	88.9 ± 1.6	75.4 ± 4.6	**59.1 ± 3.2**	67.7 ± 0	66.3 ± 1.7	51.1 ± 2.5	67.0 ± 2.3
Spectral (polynomial)	57 ± 0	96.7 ± 1.3	**49.3 ± 1.8**	82.5 ± 0.4	85.9 ± 1.1	56 ± 4.5	75.1 ± 0.6	41.3 ± 0.9	51.1 ± 1.6	66.1 ± 1.3
EM	59 ± 0.6	**98 ± 0**	-	**89.3 ± 2.1**	**94.4 ± 0.5**	**61 ± 5.6**	**78.8 ± 2.3**	**77.3 ± 7.6**	44.3 ± 0.5	**77.9 ± 2.1**
Hierarchical	57.5	68.0	12.5	37.1	70.7	32.8	67.7	43.3	32.4	46.9
OPTICS	38.9	68.7	36.9	62.4	83.7	50.4	67.7	54.5	34.8	55.3
Scaling - MinMax										
K-means	71.2 ± 0	88.4 ± 0.3	46.4 ± 1.9	88.8 ± 0.2	88.8 ± 0	57.4 ± 4.7	67.7 ± 0	94.9 ± 0	53 ± 2	72.9 ± 1.0
K. k-means (gaussian)	71.2 ± 0	84.8 ± 3.8	46.2 ± 4.6	88.8 ± 0.2	88.2 ± 0.7	55.7 ± 7.6	67.7 ± 0	94.8 ± 0.7	**53.1 ± 2.5**	72.3 ± 2.1
K. k-means (polynomial)	71.2 ± 0	78.9 ± 9.8	47 ± 3.3	88.8 ± 0.2	87.8 ± 1	53.6 ± 8.2	67.7 ± 0	94.6 ± 0.4	52.3 ± 2	71.3 ± 2.8
Spectral (gaussian)	69.5 ± 2	87.3 ± 2.7	**49.2 ± 4.4**	**91.6 ± 2.7**	94.2 ± 1.6	57.4 ± 3.6	**71.2 ± 4.2**	**96.6 ± 1.2**	**53.2 ± 0.8**	74.5 ± 2.6
Spectral (polynomial)	**71.3 ± 0.9**	81.2 ± 3.4	**49.7 ± 3.1**	89.4 ± 2.6	90.3 ± 1.3	53 ± 6.3	67.7 ± 0	93.4 ± 2.7	51.4 ± 1.9	72.0 ± 2.5
EM	66.3 ± 0.2	**98 ± 0**	-	88.1 ± 0	**95.8 ± 0**	**63.1 ± 3.2**	**68.5 ± 0.3**	**96.4 ± 0.3**	51.8 ± 0.2	**77.9 ± 1.0**
Hierarchical	57.5	66.7	12.22	34.8	72.1	32.8	67.7	39.9	32.4	46.2
OPTICS	36.4	66.7	39.4	64.3	75.8	45.9	67.1	39.9	35.7	52.3
Scaling - Standardization										
K-means	57.9 ± 0.6	75.3 ± 0	45.9 ± 2.7	51.2 ± 0.7	78.8 ± 2.1	40.5 ± 3.9	67.7 ± 0	**61 ± 0.8**	51.1 ± 0.8	58.8 ± 1.3
K. k-means (gaussian)	57.5 ± 1.1	73 ± 7.7	**46.8 ± 3.4**	51.7 ± 0.7	77.4 ± 3.5	45.5 ± 5.6	67.7 ± 0	58.3 ± 7	50.3 ± 2.5	58.7 ± 3.5
K. k-means (polynomial)	57.5 ± 1	57.8 ± 4.2	46.4 ± 5.8	51.6 ± 0.7	77.4 ± 3.5	39.2 ± 7.4	67.7 ± 0	57.9 ± 5.1	49.7 ± 2.3	56.1 ± 3.3
Spectral (gaussian)	57.3 ± 1.2	79.3 ± 2.1	**47.3 ± 4.9**	52 ± 1.8	73.9 ± 4.7	**47.1 ± 5**	67.7 ± 0	57.5 ± 7.1	51.7 ± 2.6	59.0 ± 3.3
Spectral (polynomial)	**59.4 ± 3**	81.4 ± 1.9	42.2 ± 2.8	55.5 ± 0.2	78.7 ± 1.4	36.6 ± 2.9	67.7 ± 0	58.4 ± 5.1	**53.4 ± 0.4**	59.2 ± 2.0
EM	**58.7 ± 0.3**	**98 ± 1.3**	-	**86.4 ± 1.2**	**89.8 ± 6.1**	41.2 ± 1.1	67.7 ± 0	59.8 ± 3.7	-	**70.5 ± 1.5**
Hierarchical	58.5	68.0	12.2	36.2	72.6	32.5	67.7	39.9	32.2	46.6
OPTICS	24.1	60.7	33.1	43.8	75.8	38.7	61.1	57.3	35.2	47.7
Scaling - MDS										
K-means	**65.2 ± 0**	91.3 ± 0	43.6 ± 1.2	**87.1 ± 0**	85.1 ± 0	57.5 ± 7	67.7 ± 0	68.3 ± 0.8	49.4 ± 0.8	68.3 ± 1.1
K. k-means (gaussian)	64.7 ± 0.5	91.3 ± 0	45.8 ± 4.2	85.7 ± 1.4	83.4 ± 1.8	58 ± 6.5	**75.4 ± 5**	69.2 ± 1	50.9 ± 2.5	69.4 ± 2.5
K. k-means (polynomial)	64.6 ± 0.6	91.3 ± 0	45.4 ± 3.8	86.3 ± 0.9	83.6 ± 1.6	56.4 ± 8.1	74.7 ± 5.8	70 ± 0.2	51 ± 1.4	69.2 ± 2.5
Spectral (gaussian)	57 ± 0	91 ± 0.3	43.6 ± 7	85.8 ± 2.3	72.3 ± 5.9	60.4 ± 2.2	-	67.4 ± 0	**51.9 ± 1.6**	66.2 ± 2.1
Spectral (polynomial)	57 ± 0	**92 ± 2.7**	**48.6 ± 3.4**	84.3 ± 3.3	69.8 ± 0	51.7 ± 8.4	67.7 ± 0	41.4 ± 1.3	50 ± 4.6	62.2 ± 3.0
EM	63.8 ± 0.2	90 ± 0	-	85.2 ± 1	**95.4 ± 1**	**61.6 ± 4.9**	67.7 ± 0	**71.2 ± 0.7**	49.5 ± 3.7	**72.8 ± 1.5**
Hierarchical	57.5	67.3	18.3	36.7	70.7	33.3	67.7	43.3	31.7	47.4
OPTICS	39.5	66.7	38.1	63.3	83.7	46.2	67.7	56.7	34.8	55.2

A short survey on recent methods for cage computation

Pascal Laube and Georg Umlauf

Institute for Optical Systems, University of Applied Sciences Constance, Germany

***Abstract*—Creating cages that enclose a 3D-model of some sort is part of many preprocessing pipelines in computational geometry. Creating a cage of preferably lower resolution than the original model is of special interest when performing an operation on the original model might be to costly. The desired operation can be applied to the cage first and then transferred to the enclosed model. With this paper the authors present a short survey of recent and well known methods for cage computation. The authors would like to give the reader an insight in common methods and their differences.**

I. INTRODUCTION

Due to the ever increasing amount of highly detailed 3D models algorithms that can handle large scale models and perform operations in acceptable time are a necessity. Since these models often consist of meshes with many thousand vertices, algorithms need to be highly optimized. Aside the possibility of optimizing each algorithm for speed it is also possible to simplify the problem itself. Many applications today use lower resolution versions of the respective models for evaluation and then apply the result to the high resolution model. These low resolution approximations that enclose the original model are called cages or envelopes. Fig. 1 shows an example model of an elephant and its cage.

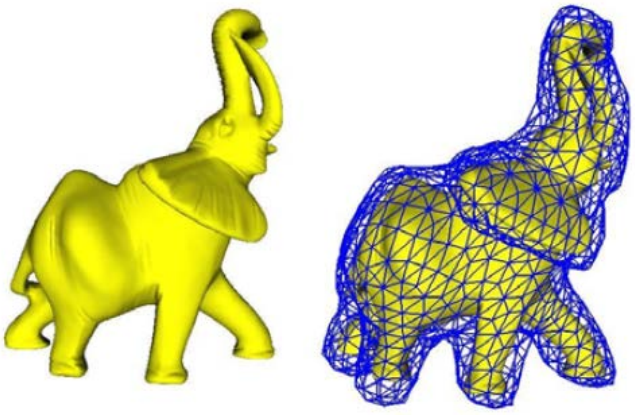


Fig. 1: Model of an elephant on the left and the model enclosed in a cage on the right [1].

There exist many different applications for cages and envelopes. The most prominent application is model deformation. When deforming a model, the deformation functional has to be applied to the whole model domain. In this case computational complexity depends on surface complexity as e.g. number of vertices or faces in a mesh. Applying the deformation to a simpler cage that encloses the original model and projecting the deformation onto the fine mesh, after applied to the cage, reduces the computational cost significantly. In [2] Lipman et al. propose the usage of Green Coordinates for cage-based space deformation. Debunne et al. [3] use multiresolution tetrahedral meshes to guarantee a certain framerate for the deformation of visco-elastic deformable objects. A fine mesh with physical properties is embedded in a tetrahedral grid to simplify deformation computation by the finite element method in [4].

Another important application that is directly related to deformation is contact and collision detection. A very common bounding structure for collision detection are spheres. James and Pai [5] use a Bounded Deformation Tree to perform collision detection on large amounts of objects using spheres. Dingliana and O'Sullivan [6] propose a multiresolution scheme detecting collisions based on level of detail when using spherical cages.

Other applications of cages and envelopes include e.g. projections of complex functions onto bounded models [7] or fast realistic rendering of objects based on high resolution texture but low resolution meshes [8].

In each area where cages are applied there exist task-specific requirements. Aside these task-specific properties there exist properties that are generally beneficial when applying cages. Cages should

- not self-intersect,
- not intersect the original model,
- be homeomorphic to the model enclosed,
- follow the model as close as possible while being a simplified version of the model.

This survey paper of methods for caging and enveloping will give the reader a quick overview of the topic and recent methods. The structure of this paper is as follows. In section two, short summaries of related methods are presented. Section three will compare the different methods while we conclude in section four.

II. RELATED METHODS

This section will give short summaries of caging methods from the areas of "Simplification and Flow", "Voxelization and Multigrid-Methods" and "Offset Surfaces".

A. Simplification and Flow

An approach for nesting multiresolution meshes is proposed by Sacht et al. [9]. The proposed algorithm does not present new results on surface simplification since the approach is independent of the used simplification. As input a number of polyhedra with varying resolution and a fitness function are needed. The polyhedra can be overlapping but need to be watertight. Taking a mesh of high resolution $\hat{M}_0$ and k decimated meshes $\hat{M}_1, \ldots, \hat{M}_k$ the method will output a sequence $M_1, \ldots, M_k$ of nested meshes, where M_{i-1} is strictly contained in M_i. $M_1, \ldots, M_k$ will be created minimizing a user-defined energy function E. Nesting is ensured by only operating on two meshes at a time starting with the finest and second finest mesh. In each step a finer mesh F is embedded inside a mesh C that was derived from an input mesh $\hat{C}$. The method consists of two main steps: In the first step the vertices of the finer mesh F are moved along a flow inwards to minimize the total signed distance to $\hat{C}$ and in a second step this mesh $\bar{F}$ will then be re-inflated back to F pushing $\hat{C}$ out of the way to become C. See Fig. 2 for a 2D example of the process. Flowing $\bar{F}$ inside $\hat{C}$ is done by minimization of

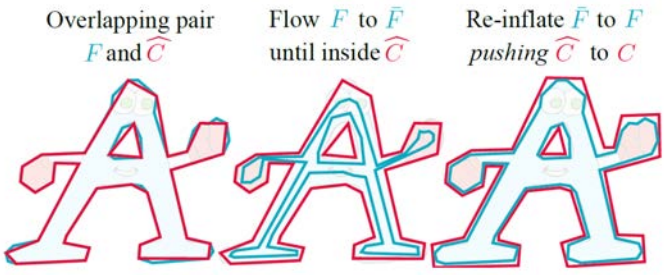


Fig. 2: Main steps of the algorithm by Sacht et al. [9]

the unsigned distance function $d(\mathbf{p})$ integrated over all points $\mathbf{p}$ of the deforming surface

$$\Phi(F) = \int_F s(\mathbf{p}) d(\mathbf{p})\, dA \tag{1}$$

where $s(\mathbf{p})$ is the sign modulation. Minimization is performed by gradient descent with fictitious time t:

$$\frac{\partial \bar{\mathbf{f}}}{\partial t} = -\nabla_f \Phi(\bar{F}).$$

Vertex positions $\bar{\mathbf{f}}$ in $\bar{F}(t)$ will flow inside the coarse mesh by taking small steps in the opposite gradient direction. Since it is not always guaranteed that there will be no intersections, the possibility of first expanding the coarse mesh is proposed. The coarse mesh will then flow outwards creating some distance between the finer mesh and itself. Contact forces are introduced to prevent self-intersections. The problem of self-intersection is a common problem in mesh expansion. This is why inward flow or shrinkage of the fine mesh is preferred.

In a next step $\bar{F}$ needs to be re-inflated to recover the original position F. While re-inflating, $\hat{C}$ needs to be pushed outwards so that there are no collisions of $\bar{F}$ and $\hat{C}$. The previous steps of flowing F inwards can now be reversed to gradually inflate back to F. Since each step back to F is a positional change in some time step Δt this can be expressed in terms of velocity. Defining the re-inflation in terms of velocity makes the use of physical simulations for contact detection possible. The method [10] can be used out of the box since it takes mesh vertices as well as desired velocities and outputs adjusted velocities. By assigning infinite mass to $\bar{F}$ one can make sure that the fine mesh will return to its original position F. If [10] fails the slower but more robust method [11] can be used.

Much like Sacht et al. [9], Sander et al. [8] use the concept of simplifying the original model and flowing this simplification away from the original to compute cages. Sander et al. render models as coarse cages to reduce rendering complexity. Their approach is based on the concept of progressive nested hulls. They start by first defining the interior volume of the model. The decision if a point p lies inside the volume $\mathcal{V}(M)$ of some model M is based on the winding number. If one takes a ray from p to infinity and tracks its intersections with M one can decide if p lies inside M. For an intersection of the ray with an inner side of a face the winding number is increased by one while for intersections with an outer side it is decreased by one. Points with positive winding number will lie inside the model.

The progressive hull algorithm is strongly based on the original progressive mesh by Hoppe [12]. In the context of simplifying a model to create a cage the original mesh M^{i+1} has to be fully enclosed by M^i so that $M^i \subseteq M^{i+1}$. This relation can be ensured by introducing inequality constraints for the position of the unified vertex in an edge collapse. This unified vertex V is constrained to lie outside the model volume $\mathcal{V}(M^{i+1})$ after an edge collapse (refer to Fig. 3 for the setup). The position of V after the collapse can be found

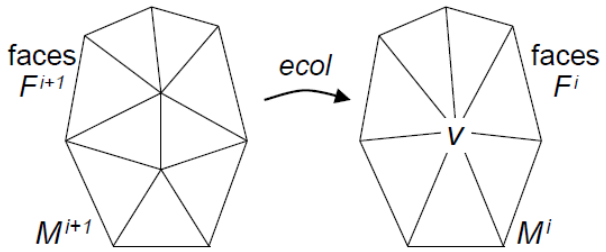


Fig. 3: Edge collapse [8].

by using linear programming to solve the resulting inequality constraints and minimizing the resulting volume enclosed by M^{i+1} and M^i. This ensures that the cage will enclose the model tightly. Additionally cost metrics can be applied when collapsing edges to ensure mesh quality.

B. Voxelization and Multigrid-Methods

Xian et al. [13] use an improved Oriented Bounding Box (OBB) tree [14] to create coarse cages. OBB structures are often applied to the problem of collision detection. Xian et

al. start by computing an initial OBB O for the model at hand. O is then subdivided by voxelization. In a classification step the voxels are classified as inner voxels, outer voxels and feature voxels. Voxels that contain part of the model mesh are feature voxels while inner voxels lie within the model and outer voxels lie outside of the model. A point set P is created that contains the mesh vertices as well as the barycenters of inner voxels. Based on P the initial OBB can be recomputed by Principal Component Analysis to get a tighter fit on the model. To further divide the OBB information about the object shape is considered. The division takes place in regions of largest change in shape (see figure 4). Xian et al. define the

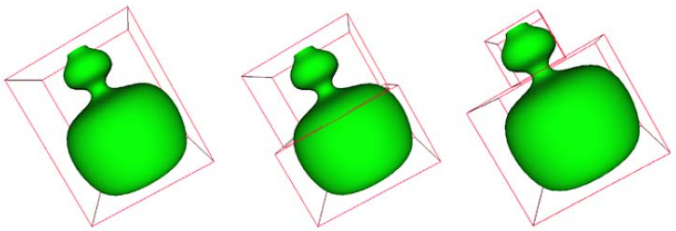


Fig. 4: Initial OBB on the right, OBB simply split in half in the middle and OBB split at locations of largest shape change on the right [13].

change in shape as cross section area function differences. The cross section area is defined as the area enclosed by the intersection of the model and a splitting plane parallel to the OBB faces composed of the smaller two OBB dimensions. The function of cross section area change is built by moving the splitting plane through the model. The location of largest change in shape is then defined as the biggest jump in the cross section area change function. A split of the initial OBB at this location results in further sub-OBBs for which the procedure is repeated until certain termination conditions are satisfied. The so created OBBs represent a first coarse cage of the model. This cage is refined by registration and merging of adjacent OBBs. Two adjacent OBBs are registered by first projecting the corner points A and B of the two adjacent faces onto an intermediate plane between them. A 2D-OBB is computed that encloses the resulting projection points A' and B' on the intermediate plane. Projection points A' and B' are then registered onto the corner points of the 2D-OBB. As a final step the two adjacent OBB are linked considering the registration of A' and B' in the intermediate plane. A triangulation can be created by simply splitting quads at diagonals. The resulting triangular mesh is then re-meshed testing for intersections of the coarse cage and the model in every step.

In [1] Xian et al. propose cage computation by first voxelizing the model to enclose. Like in [13] voxels are categorized as outer, inner and feature voxels. The resolution of the voxel grid will later on define the number of vertices of the coarse cage. The faces of the feature voxels that coincide with faces of outer voxels build a first rough approximate coarse cage. This initial cage might not be 2-manifold. At non-2-manifold edges voxels are attached while at non-2-manifold vertices a vertex-split operation is employed. For triangulation the surface quads are split at their diagonal. For further smoothing of the cage Xian et al. use an adapted mean curvature flow method [15]. Movement of vertices in the smoothing process is based on the curvature vector $H\mathbf{n}$ where H is the curvature and $\mathbf{n}$ the normal at a vertex. This vector points outwards on convex vertices while pointing inwards on concave vertices. Based on the information of outer and inner voxels Xian et al. computer an additional vector ∇d that always points away from the mesh. If the angle θ between $-H\mathbf{n}$ and ∇d lies between 0 and $\frac{\pi}{2}$ the vertex is moved along ∇d. If θ lies between $\frac{\pi}{2}$ and π it is moved along the normal $\mathbf{n}$. Additionally, the distance of each moved vertex to the model is tested at each step. If the vertex falls inside the model it is moved in the direction of ∇d. To create homeomorphic cages, the resolutions of the initial voxelization is iteratively increased until homeomorphism is given.

In [4] a very simple grid based approach is used for deformation simulation. The mesh is embedded in a hexahedral grid. A first, very coarse grid is further decomposed by an octree structure. The octree depth is defined by the user. Each of the so created voxels will inherit the mechanical properties of the enclosed polygons. This fine voxelization is then transformed back to a coarse approximation. By connecting eight voxels on each level one receives the next lager voxel of the octree. Deformation properties of the fine voxelization can be applied to each coarser level by recursive calculation. By using this approach deformation properties like e.g. stiffness of different materials can be merged on a coarser level.

C. Offset Surfaces

Ben-Chen et al. [7] create a bounding cages for the purpose of deformation transfer. In a first step they create a set of points along the surface. This can be any kind of surface as long as it is possible to assign normals to the created surface points. Next the points will be enveloped by using the Poisson reconstruction algorithm of Kazhdan et al. [16]. The resulting mesh or envelope E is simplified. This is done by using progressive mesh [12]. The surface is simplified until a user defined threshold is reached. Then for each remaining vertex a new offset position is computed by flowing each vertex of E along its normal direction outwards with predefined step size s. Ben-Chen et al. compute the vertex normal as the area-weighted average of normals of the vertices adjacent faces. This process is then repeated until the desired number of faces is reached. Ben-Chen et al. also state that the same step size s at all points can lead to self-intersections in regions that are close to each other like legs of a human model. For this case multiple user defined step sizes at different locations should be applied.

In [17] Shen et al. present an algorithm to create envelopes of polygonal soups based on approximation by moving least-

squares. For a normal least squares fit one would have

$$\begin{bmatrix} \mathbf{b}^T(\mathbf{p}_1) \\ \vdots \\ \mathbf{b}^T(\mathbf{p}_N) \end{bmatrix} \mathbf{c} = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_N \end{bmatrix},$$

with points $\mathbf{p}_i, i \in [1, \ldots, N]$, basis functions $b(x)$, the values ϕ_i at points $\mathbf{p}_i$ and the unknown coefficients c. The authors introduce a weight function $w(x)$ into the normal equation of the standard least-squares formulation. $w(x)$ is a distance function by which one can regulate approximation behavior up to interpolation. The least-square fit then becomes

$$\begin{bmatrix} w(\mathbf{x}, \mathbf{p}_1) & & \\ & \ddots & \\ & & w(\mathbf{x}, \mathbf{p}_N) \end{bmatrix} \begin{bmatrix} \mathbf{b}^T(\mathbf{p}_1) \\ \vdots \\ \mathbf{b}^T(\mathbf{p}_N) \end{bmatrix} \mathbf{c} = \begin{bmatrix} w(\mathbf{x}, \mathbf{p}_1) & & \\ & \ddots & \\ & & w(\mathbf{x}, \mathbf{p}_N) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_N \end{bmatrix},$$

with $w(x, p_i) = w(||x - p_i||)$. Since the moving-least squares method is a point based method one needs to adapt the concept for polygons. Taking points along the polygons and performing a point-based approximation (especially very tight approximations) leads to bumps and dimples along the surface. This even is the case when using quadrature points. To handle these problems Shen et al. propose using the least-squares method not to blend points of each polygon but to blend functions associated to the polygons. The standard normal equation can be built based on these functions and becomes

$$\begin{bmatrix} w(\mathbf{x}, \mathbf{p}_1) & & \\ & \ddots & \\ & & w(\mathbf{x}, \mathbf{p}_N) \end{bmatrix} \mathbf{c} = \begin{bmatrix} w(\mathbf{x}, \mathbf{p}_1) & & \\ & \ddots & \\ & & w(\mathbf{x}, \mathbf{p}_N) \end{bmatrix} \begin{bmatrix} S_1(\mathbf{x}) \\ \vdots \\ S_N(\mathbf{x}) \end{bmatrix},$$

where $S_i(\mathbf{x})$ is the polygonal function. Figure 5 shows some point-based and function-based approximations. For a more detailed explanation please consider reading [17].

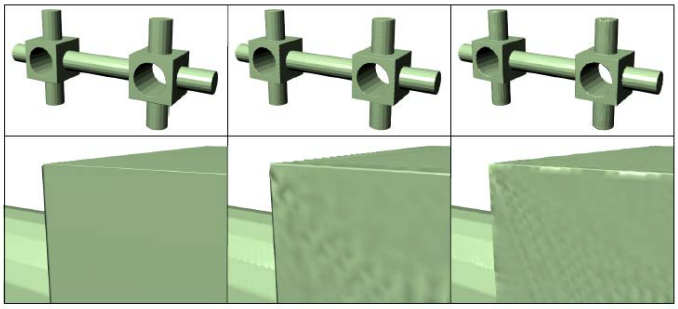


Fig. 5: The left column shows an exemplar approximation with polygonal constraints while the examples in the middle and on the right show results of point constraint examples with different densities of scattered points [17].

III. Method comparison

This section compares the different methods explained in the previous chapter. The methods will be evaluated based on the properties influencing quality defined in section one.

A problem of high priority when computing cages is the prevention of intersections between the model and the cage. While all caging methods aim to prevent these model-cage-intersections, ensuring absence of intersections (AOI) results in significant increases of computational costs. For Sacht et al. [9] AOI comes built in by the methods for physical simulation [10] and [11]. For these methods to work Sacht et al. need to ensure that the finer mesh does not intersect the coarser mesh after the inward flowing process. The methods [1] and [13] need to make sure that the cage does not intersect the model at different steps in the caging process. In [13] AOI is obtained by accepting a smaller cage resolution while in [1] vertices are pulled out of the model in the smoothing process which leads to other problems like larger cage-model-distance and possible self-intersection. For grid-based methods it is very easy to ensure AOI since they are based on the bounding box containing the model. Using the outer surface of a bounding grid or voxelization makes getting AOI simple but lacks a close resemblance of the model. The methods of [7] and [17] for offset surfaces do not include proper handling of model-cage-intersections. Deciding on intersections of the model and the cage is left to the user. For [17] it is clear that the focus of the presented work did not lie on creating cages that strictly enclose a model which is not the case in [7] where the cage generation method simply seems sufficient for the task at hand. In [8] ensuring AOI is a vital part of the method since it is introduced as a constraint into linear programming.

There exists a multitude of problems arising from cage self-intersections (SI) like unintended deformations in applications of contact detection and deformation. Preventing SI in the process of cage computations is very difficult. Since Sacht et al. [9] use a collision detection methods from physical simulation the re-inflation of the fine mesh leads to SI-free meshes. For the case where they first inflate the coarse mesh they include contact forces to prevent SI. In [1] the first approximate coarse cage (the triangulated voxelization) leads to a SI-free bounding cage. In the smoothing step the cage is forced to reside strictly outside of the enclosed model. Since in this step the cage is not checked for SIs this might lead to non-SI-free bounding cages. In [13] as well as in [1] there is no explicit SI-test or condition included in the cage computation process. Since both methods depend on user-defined thresholds for the resolution of the cages SIs might be resolved by larger resolution. This of course leads to higher cage resolution which might not be desired. When using a grid or voxelization SIs are excluded by the grid-structure itself. For methods like [4] or [1] (without the smoothing step) the problem would not be SI but merging of the bounding cage in regions where the finer model is separated like between the legs of a humanoid model. This merging can lead to problems in deformation or collision detection. The same is true for methods computing offset surfaces like [17] or [7]. Here the cage merges in areas that are very close rather than creating self-intersections. Every method that is able to create non-SI-cages like [9] and [8] needs to introduce a specific step in the computation process to handle SIs. To the authors knowledge

there currently exists no method that implicitly creates non-SI-cages.

Creating cages that are homeomorphic to the enclosed model is an important feature in cage computation. Nearly all of the mentioned methods are able to keep homeomorphism. Since none of the presented methods explicitly ensure homeomorphism it is intrinsic to the methods themselves. The only presented method that is unable to create homeomorphic cages is [4] since it keeps empty voxels in the grid-structure and handles them later in the propagation of deformation information through the grid. For all other methods homeomorphism is a question of cage resolution. With smaller cage resolution holes may be filled and by that topological information about the enclosed model will be lost. Figure 6 shows an example from [7] where the homeomorphism is lost. Homeomorphism can

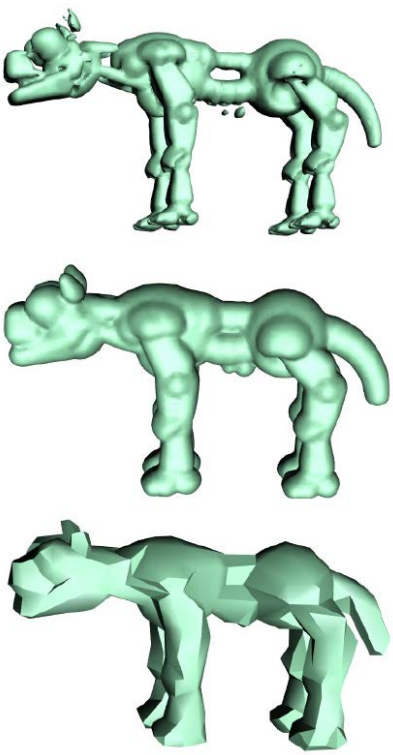


Fig. 6: Top: Initial triangular mesh; middle: Offset surface; bottom: Simplified offset surface. [8].

only be guaranteed by iteratively decreasing cage resolution toward the desired threshold and checking for homeomorphism at each step of the iteration.

Staying as close as possible to the original model while resembling the models shape is important to many applications like projection of functionals from cage to model. Sacht et al. [9] propose using an energy term that penalizes total volume between cage and model. This way a very tightly fitting cage can be computed. For methods that depend on voxelization or a grid the tightness of the fit can be controlled by selecting a proper resolution. In contrast to [9] the tightness of the fit and the resolution of the bounding cage are directly correlated which is an undesirable property. A comparable method would be [8] where the tightness of the fit and cage resolution are coupled in terms of that the tightness will vary between different levels of resolution. In [17] the tightness of the resulting offset surface can be controlled by a user defined parameter. In case that the offset surface is then triangulated like in [7] this control is lost.

IV. Conclusion

Which method to use strongly depends on the task at hand. While voxel- and grid-based methods produce bounding cages that more loosely enclose a model they may be sufficient for some collision detection or deformation tasks. The main advantage of these methods is their small computational complexity. Methods to create offset surfaces like [7] and [17] are able to produce very tight cages but to the cost of higher computational complexity resulting from the surface approximation itself. Using them as a preliminary stage to receive a triangulated cage only seems feasible if the computational cost is of no concern. Methods that use mesh simplification and flow like [9] and [8] are able to produce tight cages but also at the price of high computational cost. For [9] costly step in the process is the collision detection while for [8] linear programming seems to be the bottleneck.

Since for many cases cage computation is part of the preprocessing pipeline the impact of computational cost might be neglected.

References

[1] C. Xian, H. Lin, and S. Gao, "Automatic generation of coarse bounding cages from dense meshes," in *Shape Modeling and Applications, 2009. SMI 2009. IEEE International Conference on*. IEEE, 2009, pp. 21–27.

[2] Y. Lipman, D. Levin, and D. Cohen-Or, "Green coordinates," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 78.

[3] G. Debunne, M. Desbrun, M.-P. Cani, and A. H. Barr, "Dynamic real-time deformations using space & time adaptive sampling," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 31–36.

[4] M. Nesme, P. G. Kry, L. Jeřábková, and F. Faure, "Preserving topology and elasticity for embedded deformable models," in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3. ACM, 2009, p. 52.

[5] D. L. James and D. K. Pai, "Bd-tree: output-sensitive collision detection for reduced deformable models," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 393–398, 2004.

[6] J. Dingliana and C. O'Sullivan, "Graceful degradation of collision handling in physically based animation," in *Computer Graphics Forum*, vol. 19, no. 3. Wiley Online Library, 2000, pp. 239–248.

[7] M. Ben-Chen, O. Weber, and C. Gotsman, "Spatial deformation transfer," in *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2009, pp. 67–74.

[8] P. V. Sander, X. Gu, S. J. Gortler, H. Hoppe, and J. Snyder, "Silhouette clipping," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 327–334.

[9] L. Sacht, E. Vouga, and A. Jacobson, "Nested cages," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 170, 2015.

[10] T. Brochu and R. Bridson, "Robust topological operations for dynamic explicit surfaces," *SIAM Journal on Scientific Computing*, vol. 31, no. 4, pp. 2472–2493, 2009.

[11] S. Ainsley, E. Vouga, E. Grinspun, and R. Tamstorf, "Speculative parallel asynchronous contact mechanics," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 151, 2012.

[12] H. Hoppe, "Progressive meshes," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 99–108.

[13] C. Xian, H. Lin, and S. Gao, "Automatic cage generation by improved obbs for mesh deformation," *The Visual Computer*, vol. 28, no. 1, pp. 21–33, 2012.

[14] S. Gottschalk, M. C. Lin, and D. Manocha, "Obbtree: A hierarchical structure for rapid interference detection," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 171–180.

[15] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 317–324.

[16] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006.

[17] C. Shen, J. F. O'Brien, and J. R. Shewchuk, "Interpolating and approximating implicit surfaces from polygon soup," in *ACM Siggraph 2005 Courses*. ACM, 2005, p. 204.

Character Recognition in Satellite Images

Ankita Agrawal and Wolfgang Ertel
Institue for Artificial Intelligence
University of Applied Sciences, Ravensburg-Weingarten, Germany
Email: agrawala@hs-weingarten.de

***Abstract*—Object classification in images is a challenging task in computer vision and machine learning due to the heterogeneity of images. This paper explores hand-engineered features and popular classification algorithms along with more modern approaches such as convolutional neural networks for detecting and recognizing letters and digits in satellite images. For this purpose, a new image dataset is created, containing alphabets and numbers from the aerial view of earth.**

I. Introduction

The planet Earth is full of interesting patterns many of which are similar to each other in some or the other way. The satellites revolving around the planet come across thousands of unique patterns every day, of which we have scant knowledge. However, if we had to scan the whole planet in order to find these interesting patterns, doing so manually would be both exhaustively time consuming and tedious. In recent years, significant improvement in processing speed and storage capacity along with advancements in machine learning have enabled machines to reach or even outperform humans in special pattern recognition tasks on images, such as face or optical character recognition [1]. This paper explores the opportunity of working in the field of machine learning with the project, "Aerial Bold: A Planetary Search for Letterforms" [2], funded through Kickstarter[1]. The aim of this project is to find alphabet and number shapes that are present in earth's satellite imagery. The aerial view of the earth gives life to the patterns formed by buildings, lakes, streets, trees, parking spaces, rivers, and many such topologies. The idea is to scan these satellite images and find the structures that are shaped as letters and digits in order to create a new font based on these alphabets, the first typeface of the earth. The contributions of this paper include evaluation of image feature extraction methods and classification algorithms. Also, the creation of a processing pipeline for character recognition in satellite images based on the available techniques, formation of meaningful conclusions from the results and identification of the possible areas where further research could be carried out for improving the results.

II. Aerial Bold Dataset

To generate the labeled training data, the crowd sourcing application called "The Letter Finder App"[2] was developed by the project founders, Benedikt Groß and Joseph Lee. The application enabled all people across the globe to find the letters/digits in the satellite imagery, put a rectangular bounding box around them, assign a label and save them. These images could have different sizes. In order to have consistency, 256×256 pixel images are generated by the application, each containing letterforms saved by the people as center part of the image. Some examples of the training data images are shown in Figure 1. The distribution of the letterforms found using the application is shown in Figure 2.

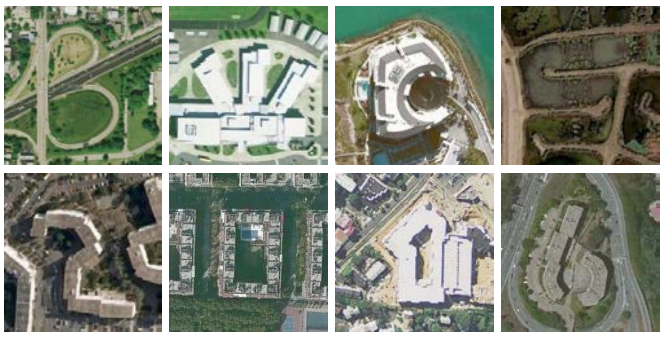


Fig. 1: Example images in Aerial Bold dataset

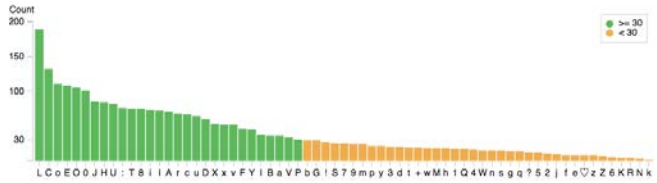


Fig. 2: Distribution of the letterforms found using Letter Finder App. Some letters are in abundance and some not. Letters like L, C, 0 occur at a relatively high frequency in sport stadiums, buildings, while in comparison occurrences of buildings forming the letters R, N, G are infrequent.

The satellite view of a random area on earth is manually scanned. When a letter is found, the latitude, longitude, label of that letter and other relevant information is saved in a database. Later, the image of each letter or digit found through the application is retrieved and rotated at all angles from -20 to 20 degrees in steps of 5° to achieve rotation invariance during classification. The range of rotation angle is chosen to keep a letter non-ambiguous. For example, letter 'C' turns into letter 'U' when rotated 90° anti-clockwise or letter 'n' when rotated clockwise. The data is then randomly divided into 90% training and 10% test data. The training images for letters which look similar are given the same label. For example, c/C

[1]https://www.kickstarter.com/
[2]http://letterhunt.aerial-bold.com/

and x/X are given the labels 'C' and 'X' respectively. Some letters are represented by only 30 instances whereas some other classes contain up to 800 instances. Since, unbalanced data is often a cause for performance deterioration in classification algorithms [3], the images of those letters/digits having less data are supersampled to balance the data. Synthetic data is also added to the training data to additionally enhance the training accuracy of the classifier. Synthetic data is created by taking standard fonts that are available for public use. They are also rotated from -20 to 20 degrees. The final training dataset consists of 48 classes, namely, 0-9,a,b,d,e,g,h,i,j,l,n,r,t,A-Z and the entire dataset comprises of about 140000 images.

III. Feature Extraction and Classification

The interpretation of a feature is generally application-dependent. According to Castleman [4], a feature can also be defined as, "a function of one or more measurements, computed so that it quantifies some significant characteristic of the object". At first, meaningful features need to be extracted from the labeled satellite images. These features should contain necessary information which allows us to identify the alphabets/numbers with sufficient accuracy. Any irrelevant detail that could interfere in the detection task is not desirable. The dataset obtained after the feature extraction is defined as $X = (x_1, \ldots, x_n)$, which consists of n vectors, with each $x_i \in \mathbb{R}^d$. Each data point is represented as a d-dimensional vector of features and the number of dimensions is based on the parameters used for extracting the feature vector. The class labels are defined as $Y = (y_1, \ldots, y_m)$, with m classes.
Thereafter, supervised classification is carried out. In the training phase, a unique description of each class is learned from the extracted features to be able to later classify new data in the testing phase. We compare Convolutional Neural Networks (CNN) [5], [6], [7] to the classification variant of K-Means [8] with k-means++ [9] initialization and Support Vector Machines (SVM) [10] using Histogram of Oriented Gradients (HOG) [11] and Overfeat [12] features. CNN is used as a feature extractor as well as a classifier. The classification algorithm takes labeled data X as input which is divided further into training data U and and validation data V. The classifier is trained with training data and tested on validation data to find the set of parameters that gives the best result. Once the best set of parameters for a classifier are found, the trained classifier object is saved and utilized later to classify the new test data, W.
The following parameters were optimized in the given range for various classifiers:

- K-Means
  - Number of clusters, k - 10 to 25
- SVM
  - Kernel type, $kernel$ - linear, rbf and polynomial kernels
  - Regularization parameter for error, C - 0.01 to 10
  - Kernel coefficient for rbf and polynomial kernels, γ - 0.01 to 1
  - Coefficient for polynomial kernel, r
  - Degree of polynomial kernel, deg - 2 to 6
- CNN
  - Number of convolution layers, n_{CL} - 3 to 8
  - Number of filters in a convolution layer, n_F - 16 to 128
  - Filter size for a convolution layer, $size_F$ - 2×2 to 11×11
  - Size of pooling layer, $size_P$ - 2×2
  - Number of hidden layers, n_{HL} - 1 to 3
  - Number of units in hidden layer, n_{HU} - 500 to 1000
  - Learning rate, η - 0.01 initially, updated to 0.001 if the training error reduces to 15%.
  - Batch size, $size_B$ - 100 and 200
  - Number of epochs, $epochs$ - 1500, 2000 and 3000
  - Image size, $size_I$ - 96×96 and 256×256

K-means is a straightforward and fast algorithm to classify the images. It aims at forming k clusters by minimizing the within-cluster sum of squares. The cluster centroids can then be used to classify a test image by assigning the class of the closest centroid to the test vector. Since the same letter can have different forms, such as letter 'A' can be written in many different fonts, the number of clusters is generally chosen to be higher than 1.

SVMs aim to find a hyperplane that splits the training data as clearly as possible, while maximizing the distance between clearly split data. The parameter C specifies the trade-off between training data misclassification and the simplicity of decision surface. A lower C value gives a smooth decision surface, while a higher value tries to classify all training samples correctly which can result in overfitting. The parameter γ has a much stronger impact on the classification results. It defines the distance which a single training sample can reach. The larger the value of γ, the closer other samples must be to be affected. Hence, it results in over-fitting. A very small value means that the distance is greater, resulting in underfitting.

In this work, the CNN architecture is adapted from LeNet [13], which is designed for the recognition of handwritten and machine-printed characters. Multiple convolution layers are defined with a built-in max-pooling layer. The output of one convolution layer is the input of the next convolution layer. One or more hidden layers are defined after the convolution layers. Output of the last hidden layer is then fed into the softmax layer for computing the class-membership probabilities.

Pre-Attempts: In the beginning, we applied image preprocessing techniques such as Gaussian blur on the input images to extract better features from the image. However, due to vast differences in the nature of satellite images, a common set of image parameters could not be found. Also, it was attempted to build one multiclass CNN

classifier to train all the letters together. But the training data for all the letters together exceeded the GPU memory limit. Thus, this approach had to be discarded.

Final approach: There are a lot of parameters to be optimized over a large range of values, CNN requires a lot of computational resources and time for training. Hence, due to resource limitations, instead of building one multiclass CNN classifier model to train all the letterforms together, each class is trained separately giving rise to 48 binary CNN classifiers. Each letter is trained individually against negative samples as a binary classifier. The training data for one classifier consists of 12% positive data for a letter, and 88% negative data formed by taking data samples for all other letters. During the training of a single CNN model, the negative log likelihood function is minimized at each epoch to optimize the filter values at each convolution layer in that model. Since, there are 48 classes, this process results in 48 learned models, one for each letter. An unlabeled satellite image could contain multiple letterforms. Hence, during the testing phase, each image is classified by all the 48 models individually to obtain the possible target classes.

This paper aims to classify images accurately with the requirement of minimizing the number of false positives. This is because the amount of satellite imagery to be classified is abundantly high and the results should contain the maximum amount of correct letters. Thus, to measure the quality of parameter combinations, *fpr (false positives rate)*, *ras (roc auc score)* [14] and *fs (f1 score)* [15] are used.

$$fpr = \frac{number\ of\ false\ positives}{total\ negative\ data\ samples}$$

$$ras = area\ under\ roc\ curve$$

$$fs = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

To define a loss function, each of these scores are given a weight value. Hence, the loss function is defined as,

$$loss = 1 - \left(\frac{1 \cdot (1 - fpr) + 2 \cdot ras + 1 \cdot fs}{4} \right)$$

where $loss \in [0.1]$.

The measurements *ras* and *fs* have their maximum value at 1 and *fpr* at 0. It is intended that a loss value of 0 is the best score, hence, the subtraction from 1 in the above equation.
The following pipeline is carried out for convolution neural network until the maximum number of trials are executed. First, the input images are separated per class. Hyperparameters are defined based on the search space formed by combination of variable parameter ranges. Then using one set of hyperparameter, a CNN model is created and feature vector set X is extracted. The features could also be as simple as the input images flattened into a vector set. Now, classification is performed on validation set $V \subset X$ and $loss$ value is obtained. Based on $loss$ value, next set of hyperparameters is chosen and the process of creating and testing the new model is repeated. After the best parameter set for the problem is found for one letter, the same parameter combination is used to train the classifier for remaining 47 classes. The trained classifier object is loaded and the classes for unlabeled test data from the satellite imagery are predicted.

Challenges: As mentioned in section II, the data containing letters and digits for the Aerial Bold dataset has been collected by people all over the globe. However, many of them were not suitable for a machine learning task. Also, the data for some letters such as i, j, etc. is not easily available in the satellite images and if found, not very clear. Therefore, each image had to be checked and labeled again manually. Secondly, even after applying the data augmentation techniques, the dataset is rather small (on an average about 2000 images for each letter) in comparison to the number of classes and complexity of the problem. Another major challenge was the computing power available for the task. In the beginning, a system with 16GB RAM was available which took about 10 to 48 hours to compute a single SVM model for the entire training data. Later a system with 64 GB RAM and four 4GB graphic cards was used. The code is organized such that each GPU executes a different set of hyperparameters for the convolutional neural network in parallel. The CNN classifier executes 10 times faster on a GPU than a CPU. However the graphic card memory was not sufficient to train large networks, especially with larger filter sizes. Even a 5 convolution layer network with filter size 11 and 512 neurons could not be evaluated due to the memory error. In the existing works, for a complex problem, larger networks having more than 7 convolution layers are found to produce better results which could not be evaluated due to hardware limitations.

IV. Results

Table I shows an overview of the performance of various feature extraction and classification method combinations.

TABLE I: Minimum classification error (%) on Aerial Bold dataset

K-Means		SVM		CNN
HOG	Overfeat	HOG	Overfeat	
66.8	79.3	49.8	64.5	**14.3**

The HOG features give the best result with 128×128 image size, 6×6 cell size, 2×2 block size and 12 orientations along with 15 clusters for the k-means algorithm. SVM provides the highest accuracy using a polynomial kernel with $\gamma = 0.02$ for a 96×96 image. However, CNN, that has been known to significantly improve the classification performance on various image datasets, including MNIST [13], CIFAR10 [16] and ImageNet [17], provides the lowest error among the three classifiers on the Aerial Bold dataset.

Figure 3 represents nine different rotations of a satellite

image containing letter A and their respective outputs from the convolution layer. The first row contains the original images, whose gray scale variant is fed to the network initially. Row two depicts one of the output images of the first convolution layer, row three shows output of the second convolution layer and so on. The model used to generate these images has five convolution layers.The filter sizes for the convolution layers are 3, 4, 4, 8, 4 and the number of kernels for each layer are 20, 52, 42, 43, 44 respectively. It has a single hidden layer with 907 hidden units.

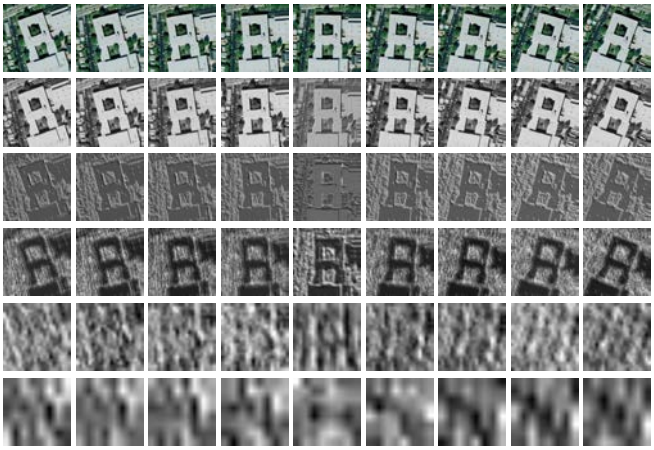


Fig. 3: Original images and the output of each convolution layer for the respective images in the previous row.

As can be seen in figure 3, the first three layers remove irrelevant information and background noise, and also improve the image contrast. The learned filters are able to distinguish letter A independent of its rotation angle. This is important to recognize the letterforms in different orientations.

Figure 4 depicts the classification accuracy on the validation data with respect to the number of convolution layers for different set of hyperparameters. The graph shows that the result improves with the increase in number of convolution layers. This can be seen by the mean classification accuracy for each layer indicated by the red-colored circle. Unfortunately it was not possible to evaluate a sufficient number of large filters with six and more convolution layers, as they demand high machine memory. Hence, the figure shows a drop in the accuracy from 5 to 6 convolution layers.

The classification accuracy on validation data during training phase is shown in figure 5. The filter size at convolution layer 3 is extracted and the results for small filter size are represented in the figure 5a. Figure 5b represents the results for filter size larger than 9x9. Both graphs show a positive correlation between the correctly identified A's and non-A's. However, the large filter size shows a much clearer correlation, that is, they tend to perform better than the small ones.

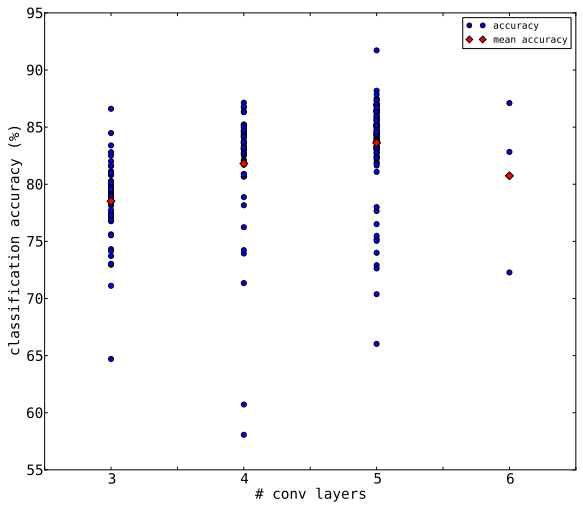


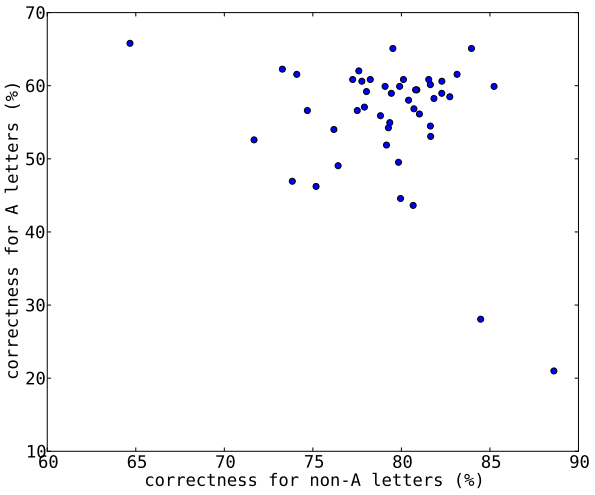
Fig. 4: Results on test data depending upon number of convolution layers. The red circle depicts the mean accuracy for each layer.

Each row of figure 6 shows an input image, one of the many filters (learned by the final network after training) that is applied to it at a particular layer and the output image of that layer respectively, for first three convolution layer. We can see that at higher convolution layers, the background noise dissolves and the letterform becomes more apparent.

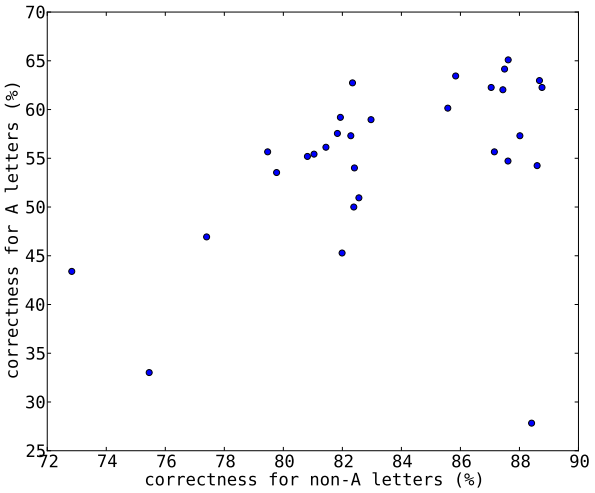
Although the model is only trained for letter 'A', if the same model or set of parameters is applied for other letters, it is able to separate the letterform from the background noise. This is shown in figure 7 for a letter 'X' sample. where the input, output and applied filter (learned by final network trained on letter 'A') for first three convolution layers are depicted. Therefore, the same model is used for identification of other letterforms.

The other widely used filters for gradient detection such as Sobel and Canny operators do not work well on the Aerial Bold dataset. However, the convolutional neural networks have been able to learn filters that produce a preprocessed image. As we can see in the above figures, the filters are complex and would be close to impossible to be produced by a human expert, especially that it can be used for different letterforms in a generalized way. This represents that CNN is capable in solving complex image recognition tasks.

Figure 8 presents some of the letterforms classified by the CNN models in the new unlabeled data. Although the classifier does not specify the position or rotation of the letterform in the image, it can find translational and rotation invariant letters and digits. Examples of the images which could not be identified by the CNN due to excessive noise or



(a) Small Filter Size



(b) Large Filter Size

Fig. 5: Classification results on validation and test data for letter A for different filter sizes at convolution layer 3

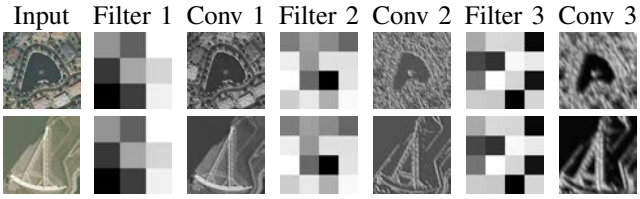


Fig. 6: Input, output and filter for the first three convolution layers

their similarity with other letterforms are shown in the fourth row of the figure.

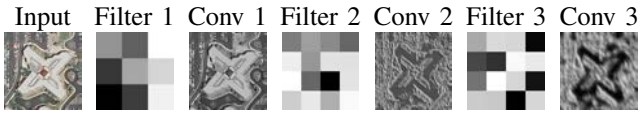


Fig. 7: Input, output and filter for the first three convolution layers for letter X

Fig. 8: New unlabeled images from the satellite imagery classified with the trained CNN models for each letterform along with their correct labels below each image. The first three rows show the correctly classified letterforms and in the fourth row we display the instances which were incorrectly classified.

Despite the challenges stated in section III and the high problem complexity, convolutional neural network classifies the letterforms with high accuracy.

V. Conclusion

Several feature extraction methods and classifiers have been investigated for letter and digit recognition within landscapes and buildings in satellite images. The hand-engineered features do not perform well on the Aerial Bold dataset. Convolutional neural networks showed promising results outperforming other classifiers.

In this paper, training of CNN was carried out using stochastic gradient descent optimization. The current research on artificial neural networks indicate that the optimization algorithms with second order approximation perform better. Methods such as AdaDelta [18] would be applied in future to introduce dynamic learning rate for training phase. To avoid

co-adaptation of features, dropout learning [19], [20] will be introduced.

REFERENCES

[1] R. Benenson, "Classification datasets results." [Online]. Available: http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html

[2] B. Groß and J. Lee, "Aerial Bold: Kickstart the Planetary Search for Letterforms!" [Online]. Available: https://www.kickstarter.com/projects/357538735/aerial-bold-kickstart-the-planetary-search-for-let

[3] R. Barandela, R. M. Valdovinos, J. S. Sánchez, and F. J. Ferri, "The imbalanced training sample problem: Under or over sampling?" in *Structural, Syntactic, and Statistical Pattern Recognition*. Springer, 2004, pp. 806–814.

[4] K. R. Castleman, Ed., *Digital Image Processing*. Prentice Hall, 1996.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[6] S. Behnke, "Hierarchical Neural Networks for Image Interpretation," *Lecture Notes in Computer Science, Springer*, vol. 2766, 2003.

[7] P. Simard, D. Steinkraus, and J. C. Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," *ICDAR*, vol. 3, pp. 958–962, 2003.

[8] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[9] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," *Proceedings of the 18th annual ACM-SIAM Symposium on Discrete algorithms, Society for Industrial and Applied Mathematics Philadelphia, PA, USA*, pp. 1027–1035, 2007.

[10] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[11] N. Dadal and B. Triggs, "Histogram of Oriented Gradients for Human Detection," *Proceedings of IEEE Conference Computer Vision and Pattern Recognition, San Diego, USA*, pp. 886–893, 2005.

[12] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated Recognition,Localization and Detection using Convolutional Networks," *International Conference on Learning Representations*, 2014.

[13] L. Lab, "Convolutional Neural Networks (LeNet) - DeepLearning 0.1 documentation, DeepLearning 0.1." [Online]. Available: http://deeplearning.net/tutorial/lenet.html

[14] J. A. Hanley and B. J. McNeil, "The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[15] C. J. Van Rijsbergen, *Information Retrieval*, 2nd ed. Butterworth-Heinemann, 1979.

[16] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *IEEE Conference on Computer Vision and Pattern Recognition, New York*, pp. 3642–3649, 2012.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A Large-Scale Hierarchical Image Database," *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[18] M. Zeiler, "ADADELTA: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012. [Online]. Available: http://arxiv.org/abs/1212.5701

[19] G. Hinton, N. Srivastava, and A. Krizhevsky, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:*, 2012. [Online]. Available: http://arxiv.org/abs/1207.0580

[20] S. Wang and C. Manning, "Fast dropout training." *ICML (2)*, 2013.

Towards Sensorless Control for Softlanding of Fast-Switching Electromagnetic Actuators

Tristan Braun, Johannes Reuter
Institute of System Dynamics,
Konstanz University of Applied Sciences, Germany
Email: {tbraun, jreuter}@htwg-konstanz.de

***Abstract*— An overview of current research regarding the sensorless control of digital (on/off) single coil electromagnetic actuators (EMAs) is given. This includes the discussion of a possible control strategy as well as the design of suitable nonlinear observers to obtain the position and velocity information of the moving plunger which is necessary for feedback. Self-sensing in this context means that solely the available energizing signals, i.e., coil current and driving voltage are used to estimate the position and velocity trajectories. Experimental results for the considered control approach as well as for the estimation approach will be shown.**

I. Introduction

Electromagnetic actuators (EMAs) are widely used in industrial, automotive, and other mechatronic applications. Fast switching or digital (on/off) solenoid actuators can be found for instance in internal combustion engines as gas exchange valves [1], as fuel injection valves [2], as actuation valves for hydraulic systems [3], or as antilock braking system valves [4]. Physical sensors are normally undesired, mostly due to economic and fabrication aspects, but also due to reliability issues. On the contrary, information of the actual position of the plunger is highly useful due to a variety of reasons, as for instance, to control the motion trajectory of the plunger or for monitoring tasks. Especially, softlanding control is highly desired where a movement of the plunger with zero impact velocity is enforced to reduce noise emission and waste of material. As investigated by several authors [2], [5]–[8], feedforward control strategies turned out to be well suited for such problems of motion planning. Single feedforward control, in general, is not robust against parameter variations or disturbances. Consequently, feedback of the position and velocity is needed for stabilization of the desired plunger motions. In this regard, model based algorithms are useful to reconstruct the state information from the driving signals, i.e., coil current and driving voltage. Different versions of self-sensing approaches for solenoid valves can be found in the literature. Beside the methods which are based on the estimation of position-dependent parameters [9]–[12], that are particularly suitable for positioning actuators, observer-based methods [13]–[15] are promising for the trajectory estimation of fast switching EMAs.

This paper is structured as follows. In Section II, the dynamic model of EMAs is constituted. Then, in Section III, a flatness based control approach is discussed. Subsequently, the design of nonlinear observers for this task is thoroughly discussed in Section IV. In Section V, the need for softlanding control methods is further motivated, and the control approach as well as the estimation approach will be illuminated by practical experiments for a commercial digital EMA. The paper concludes with an outlook for future work in Section VI.

II. Dynamic Model of a Digital Solenoid Valve

The voltage drop V over the coil of an EMA can be described by

$$V = R_{\mathrm{Cu}}\, i + \frac{\partial \Psi(i,z)}{\partial i}\frac{\mathrm{d}i}{\mathrm{d}t} + \frac{\partial \Psi(i,z)}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}t}, \tag{1}$$

with driving current i, copper resistance R_{Cu}, and the flux linkage $\Psi(\cdot)$, with nonlinear dependence on current and position z of the iron portion in the magnetic field. The second term on the right hand side of (1) accounts for the induction due to a change in the current i, and the third term is the back-electromotive force (back-EMF) that is proportional to the velocity v of the plunger moving in the magnetic field. This model must be enhanced in order to include eddy current effects which have significant impact on the valve dynamics. It can be shown (see [16]) that in the presence of eddy currents, it holds for the magnetomotive force

$$\theta(t) = \oint_l H\,\mathrm{d}l + R_{\mathrm{ed}}\dot{\Psi}, \tag{2}$$

where H is the magnetic field strength with path l of magnetic field lines, and R_{ed} is called the eddy current resistance. Taking into account (1), the equivalent circuit results as shown in Fig. 1. To achieve better numerical stability, an additional constant inductance L_{ed} is used in series to R_{ed}. This is, moreover, physically justified since a voltage step at the coil cannot lead to a discontinuous change in the current. The resulting system of differential equations reads:

$$\frac{\mathrm{d}i_{\mathrm{ed}}}{\mathrm{d}t} = -\frac{R_{\mathrm{ed}}}{L_{\mathrm{ed}}}\, i_{\mathrm{ed}} - \frac{R_{\mathrm{Cu}}}{L_{\mathrm{ed}}}\, i + \frac{V_{\mathrm{drive}}}{L_{\mathrm{ed}}} \tag{3a}$$

$$\frac{\mathrm{d}i}{\mathrm{d}t} = -\frac{R_{\mathrm{ed}}}{L_{\mathrm{ed}}}\, i_{\mathrm{ed}} - \alpha(i_{\mathrm{L}},z)(R_{\mathrm{Cu}}\, i - V_{\mathrm{drive}}) + \beta(i_{\mathrm{L}},z)\, v \tag{3b}$$

where $i_{\mathrm{L}} = i - i_{\mathrm{ed}}$, $\alpha(i_{\mathrm{L}},z) := 1/\Psi_i(i_{\mathrm{L}},z) + 1/L_{\mathrm{ed}}$, $\beta(i_{\mathrm{L}},z) := -\Psi_z(i_{\mathrm{L}},z)/\Psi_i(i_{\mathrm{L}},z)$, and $\Psi_i(i_{\mathrm{L}},z)$ and $\Psi_z(i_{\mathrm{L}},z)$ are the partial derivatives of $\Psi(i_{\mathrm{L}},z)$.

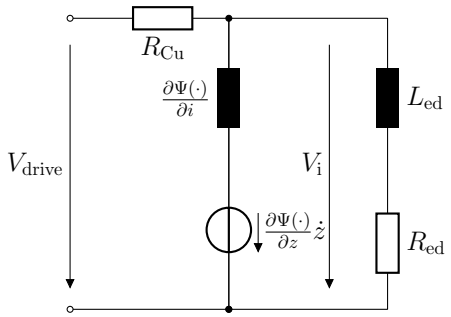


Fig. 1. Lumped electromagnetic equivalent circuit of the solenoid.

A. Mechanical Subsystem

The mechanical subsystem can be obtained by Newton's second law of motion. It follows

$$\dot{z} = v, \qquad z(0) = z_0 \tag{4a}$$
$$m\,\dot{v} = F_{\mathrm{m}}(i,z) - d\,v - c\,z - F_{\mathrm{s}}, \quad v(0) = v_0 = 0, \tag{4b}$$

where

$$F_{\mathrm{m}}(i,z) = -\int_{i_0}^{i} \Psi_z(\iota,z)\mathrm{d}\iota \tag{5}$$

is the magnetic force acting against the spring force with suspension rate c. The spring preload is taken into account by the constant F_{s}. Furthermore, viscous friction is modeled by the term dv.

III. Control Approach

The objective of a softlanding control strategy for switching solenoid actuators is the realization of transitions of stationary setpoints. The desired trajectory between the two extremals of the air gap is planned in such a way that a smooth stop with zero velocity can be achieved. A proper trajectory control strategy for nonlinear systems is based on flatness based control [17], [18].

A. Feedforward Control

The planning of a desired trajectory for the transition of stationary set-points in a finite time interval can be performed in a convenient way if the system is differentially flat. The flatness property is defined below [18], [19]:

Definition 1: A nonlinear system $S(\omega_i, \dot{\omega}_i, \ddot{\omega}_i, \ldots, \omega_i^{(\sigma)}) = 0$, with system variables ω_i, $i = 1,\ldots,s$, is said to be (differentially) flat if there exists a function $y = \phi(\omega_i, \dot{\omega}_i, \ddot{\omega}_i, \ldots, \omega_i^{(\alpha)})$ such that the derivatives of y are differentially independent, i.e., $R(y,\dot{y},\ldots,y^{(\beta)}) \neq 0$, and $\omega = \psi(y,\dot{y},\ldots,y^{(\gamma)})$. The variable y is called the flat or basic output.

Roughly speaking, if the system is differentially flat the input $t \to u(t)$, $t \in [0,T]$ that steers the system along a desired trajectory y can be calculated by y and its time derivatives up to the order n of the system, if the system is of finite dimension. Flatness-based feedforward control for electromechanical systems has been extensively studied in the last decades. From [20] it is known that the system of an EMA is differentially flat, and the variable z is a flat output for the system. Considering the state representation of the mechanical system (4a) the necessary current that steers the position along a desired path can be easily obtained by

$$i_{\mathrm{d}} = F_{\mathrm{m}}^{-1}(z_{\mathrm{d}})[m\,\ddot{z}_{\mathrm{d}} + c_\mu \dot{z}_{\mathrm{d}} + c_{\mathrm{s}}\,z_{\mathrm{d}} + F_{\mathrm{s}}], \tag{6}$$

where the subscript d indicates the desired trajectories. Furthermore, if the current is controlled by a fast current controller such that the dynamics of the electromagnetic subsystem can be neglected, the variable i can be regarded as the systems input. The control law requires an inversion of the magnetic force, which can also be performed numerically if its characteristic is given in the form of look-up tables. Taking into account magnetic diffusion effects due to eddy currents which have a significant impact on the force dynamics, the control law can be enhanced by considering the spatial distribution of the magnetic field in the iron core of the solenoid [2], [7], [8].

The feedforward control law depends on the system model, consequently, it is not robust against parameter variations or disturbances. Therefore, feedback is necessary to stabilize the position along the desired trajectory. Due to cost and fabrication reasons, solenoid valves are normally not equipped with position sensors. Accordingly, it is suggested to estimate the needed quantities by an observer approach.

IV. Observer Design for the Solenoid Actuator

Let the state vector be defined as $\boldsymbol{x} = (z,\, v,\, i)$, the input $u = \bar{V}_{\mathrm{drive}}$, i.e., the filtered value of the driving voltage, and the output $y = x_3$. For the sake of simplicity, eddy currents are neglected in the observer design, however, the incorporation is feasible [21]. A possible state representation for (3) and (4) reads

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},u), \quad \boldsymbol{x}(0) = \boldsymbol{x_0} \tag{7a}$$
$$y = \boldsymbol{c}^T\,\boldsymbol{x}. \tag{7b}$$

A. Simple Nonlinear Observer Method and Motivation

A straightforward procedure for the design of a nonlinear observer is the method of extended linearization [22]–[24]. However, beside the observability property also the structure of the dynamic model has to be taken into account to satisfy the conditions for a stable error dynamics. The procedure will be briefly reviewed. Therefore, system (7) is expressed by Taylor series expansion around the trajectory $\boldsymbol{x} = \hat{\boldsymbol{x}}$. With the error defined as $\boldsymbol{e} := \boldsymbol{x} - \hat{\boldsymbol{x}}$ it yields

$$\boldsymbol{f}(\boldsymbol{x},u) = \boldsymbol{f}(\hat{\boldsymbol{x}},u) + \frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x},u)\bigg|_{\hat{\boldsymbol{x}}}\boldsymbol{e} + R(\boldsymbol{e},u) \tag{8}$$
$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\hat{\boldsymbol{x}},u) + \boldsymbol{A}(\hat{\boldsymbol{x}},u)\boldsymbol{e} + \boldsymbol{R}(\boldsymbol{e},u),$$

where $\boldsymbol{R}(\boldsymbol{e},u)$ represents terms of higher order, and $\boldsymbol{A}(\hat{\boldsymbol{x}},u)$, is the Jacobian of $\boldsymbol{f}(\boldsymbol{x},u)$. The observer is stated by

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{f}(\hat{\boldsymbol{x}},u) + \boldsymbol{L}(\hat{\boldsymbol{x}},u)(y - c^T\hat{\boldsymbol{x}}). \tag{9}$$

The term $\boldsymbol{R}(\boldsymbol{e},u)$ might be neglected for sufficiently small initial errors $\boldsymbol{e}(0) = \boldsymbol{x}(0) - \hat{\boldsymbol{x}}(0)$ [24]. Writing the derivative

w.r.t. time of the error $\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}} - \dot{\hat{\boldsymbol{x}}}$, the linear error dynamics can be obtained as

$$\dot{\boldsymbol{e}} = \boldsymbol{A}_o(\hat{\boldsymbol{x}}, u)\, \boldsymbol{e}, \tag{10}$$

where

$$\boldsymbol{A}_o(\hat{\boldsymbol{x}}, u) := \left(\boldsymbol{A}(\hat{\boldsymbol{x}}, u) - \boldsymbol{L}(\hat{\boldsymbol{x}}, u)\, \boldsymbol{c}^T\right). \tag{11}$$

It is clear that if $\boldsymbol{A}_o(\hat{\boldsymbol{x}}, u)$ can be made constant by suitable choice of $\boldsymbol{L}(\hat{\boldsymbol{x}}, u)$, then the error dynamics (10) is time invariant, and can be made exponentially stable if $\boldsymbol{A}_o \equiv const$ is Hurwitz. Since here $\boldsymbol{c}^T = (0,\, 0,\, 1)$, the error matrix has the form

$$\boldsymbol{A}_o(\cdot) = \begin{pmatrix} 0 & 1 & -L_1(\cdot) \\ a_{21}(\cdot) & a_{22} & a_{23}(\cdot) - L_2(\cdot) \\ a_{31}(\cdot) & a_{22}(\cdot) & a_{21}(\cdot) - L_3(\cdot) \end{pmatrix}, \tag{12}$$

where $a_{ij}(\cdot)$ are the entries of the Jacobian which are all time-dependent except of a_{22}. Obviously, the condition that $\boldsymbol{A}_o(\cdot)$ being constant can not be fulfilled irrespective of the choice of the observer injections $\boldsymbol{L}(\cdot)$. Nevertheless, desired convergence properties of the observer can be examined in simulations [22]–[24], however, stability can not be guaranteed.[1]

B. Tracking Observer

In the realm of tracking control, in general, it is beneficial to design the observer together with the knowledge of the desired state trajectories. To this end, the foregoing approach can be rendered to the design of a tracking observer as proposed in [26]. Instead of linearizing the system along the estimated state $\hat{\boldsymbol{x}}$, it is linearized along the desired state vector $\boldsymbol{x}_\mathrm{d}$. Neglecting terms of higher order, the error dynamics yields the form

$$\dot{\boldsymbol{e}} = \left(\boldsymbol{A}(\boldsymbol{x}_\mathrm{d}, u_\mathrm{d}) - \boldsymbol{L}(t)\right)\, \boldsymbol{e}, \tag{13}$$

where $\boldsymbol{L}(t)$ is a time-varying observer gain. Since $\boldsymbol{A}(\boldsymbol{x}_\mathrm{d}, u_\mathrm{d})$, depends on known time-varying functions, the observer problem can be solved using the generalized Ackermann's formula for time-variable systems. With this procedure, the time-varying gain $\boldsymbol{L}(t)$ can be determined such that in invariant coordinates the transformed matrix in observer normal form is Hurwitz. It reads [27]–[29]

$$\boldsymbol{L}(t) = \left(p_0 + p_1\mathcal{N} + p_2\mathcal{N}^2 + \cdots + p_{n-1}\mathcal{N}^{n-1}\right)\, \boldsymbol{v}(t), \tag{14}$$

where $p_0 \ldots p_{n-1}$ are the coefficients of the characteristic polynomial $\rho(s) = s^n + p_{n-1}s^{n-1} + \cdots + p_1 s + p_0$ of the associated error system in observer normal form that governs the error dynamics. The linear differential operator is defined as $\mathcal{N}\boldsymbol{v} = -\dot{\boldsymbol{v}} + \boldsymbol{A}\boldsymbol{v}$. The vector $\boldsymbol{v}(t)$ can be obtained from the system of linear equations

$$\boldsymbol{Q}(t)\, \boldsymbol{v}(t) = \boldsymbol{e}_n, \tag{15}$$

where $\boldsymbol{e}_n$ is the n-th unit vector, and $\boldsymbol{Q}(t)$ is the observability matrix. Consequently, $\boldsymbol{v}(t)$ is the last column of the inverse observability matrix. Obviously, at points where the system is not observable, $\boldsymbol{v}(t)$ is not defined due to a singularity of $\boldsymbol{Q}^{-1}(t)$. For a linear time-variant system

$$\boldsymbol{Q}(t) = \begin{pmatrix} \boldsymbol{c}(t) & \mathcal{L}\boldsymbol{c}(t) & \mathcal{L}^2\boldsymbol{c}(t) & \ldots & \mathcal{L}^{n-1}\boldsymbol{c}(t) \end{pmatrix}^T, \tag{16}$$

where $\mathcal{L}\boldsymbol{c}(t) = \dot{\boldsymbol{c}}(t) + \boldsymbol{c}(t)\boldsymbol{A}(t)$. This method has been tested for the EMA-system described by the state representation (3) and (4) for smooth polynomial reference trajectories $t \to x_\mathrm{d}$, $t \to \dot{x}_\mathrm{d}$, $t \in [0, T]$. However, the inverse of the observability matrix $\boldsymbol{Q}(t)$ which enters directly in the calculation of the observer gains comes along with singular points due to zero crossings of $\boldsymbol{Q}(t)$ at a few time points. Although the observability condition is lost merely at a few time instances, this might lead to numerical problems. Such singularities depend on the selected reference trajectories and also on the system parameters. The following approach shows satisfying performance in simulation and experiments.

C. Nonlinear Sliding Mode Observer

The design of a nonlinear sliding mode observer (SMO) as proposed by [30] is convenient for the problem considered here. The SMO is given as

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{f}(\hat{\boldsymbol{x}}, u) + \boldsymbol{s}(e_y), \quad \hat{\boldsymbol{x}}(0) = \hat{\boldsymbol{x}}_0 \tag{17a}$$

$$y = \boldsymbol{c}^T\, \hat{\boldsymbol{x}}, \tag{17b}$$

where $e_y = x_3 - \hat{x}_3$, and $\boldsymbol{s}(e_y)$ is the vector of discontinuous observer injections

$$s_i(e_y) = h_i\, \mathrm{sign}(e_y) + k_i\, e_y, \quad i = 1 \ldots 3, \tag{18}$$

where

$$\mathrm{sign}(e_y) = \begin{cases} 1, & e_y > 0, \\ -1, & e_y < 0 \end{cases}, \tag{19}$$

with the observer gains h_i and k_i, such that the error dynamics is stable. The detailed structure of the SMO is established by

$$\dot{\hat{x}}_1 = \hat{x}_2 + s_1(e_y) \tag{20a}$$

$$\begin{aligned} \dot{\hat{x}}_2 =& \frac{1}{m} F_\mathrm{m}(\hat{x}_1, y) - \frac{c}{m}\, \hat{x}_2 - \frac{d}{m}\, \hat{x}_2 + s_2(e_y) \\ \dot{\hat{x}}_3 =& \alpha(\hat{x}_1, y)(u - R_\mathrm{Cu}\hat{x}_3) + \beta(\hat{x}_1, y)\, \hat{x}_2 \\ &+ s_3(e_y), \quad \hat{\boldsymbol{x}}(0) = \hat{\boldsymbol{x}}_0. \end{aligned} \tag{20b}$$

The stability of the observer has to be examined in the sliding regime $\dot{e}_y = e_y = 0$ by considering the underlying equivalent error dynamics [30], [31]. A proof of stability is skipped here for the sake of brevity, but can be found in [21].

V. Experimental Results

Experimental validations were performed with a commercial fast-switching hydraulic valve under dry conditions. Typical switching times are between 1 ms and 5 ms dependent on the driving current. The estimation results are validated by a high resolution optical position sensor. In Fig. 2 and Fig. 3 the difference of a soft-landed trajectory by feedforward control, and a trajectory with hard impact (blue curves) is illustrated.

[1] Note that, even if the real parts of all eigenvalues are negative and constant, stability can not be guaranteed [25], page 158.

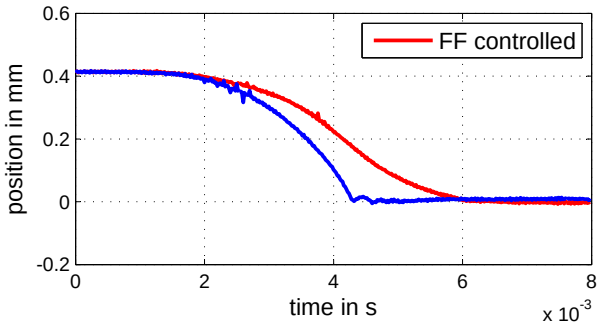


Fig. 2. Softlanding trajectory by feed forward control, and hard switching trajectory by constant driving current input.

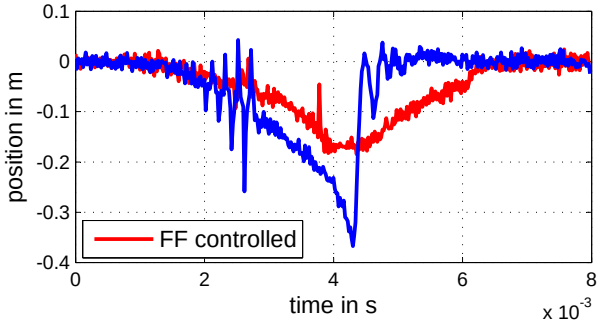


Fig. 3. Softlanding velocity by feed forward control, and velocity from hard switching by constant driving current input.

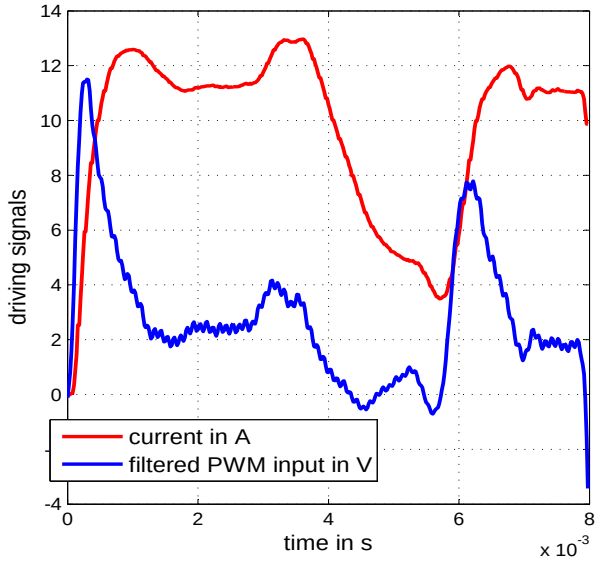


Fig. 4. Feedforward control signals. The voltage input is the filtered PWM-voltage.

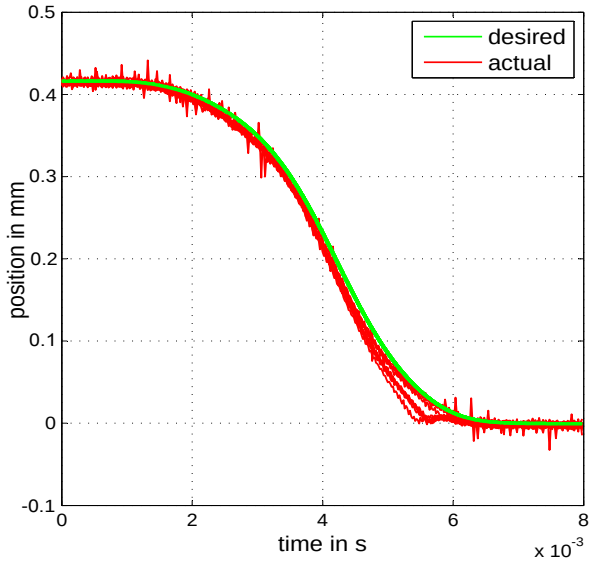


Fig. 5. Steered trajectory from consecutive switching operations.

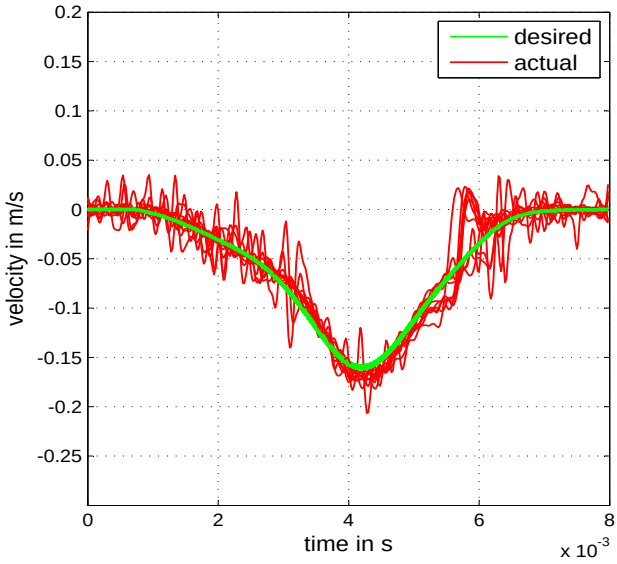


Fig. 6. Velocity from consecutive switching operations.

The feedforward control signals that steer the plunger along the desired path are shown in Fig. 4.

It is worth the mention that the control works repeatedly, however, small disturbances or model inaccuracies can lead to errors regarding the desired trajectory. This is demonstrated by Fig. 5 and Fig. 6, where the position and velocity from 10 consecutive feedforward controlled switching operations is shown. Note that solely dry operation is investigated. In the case of varying forces associated with changing fluid pressure levels, feedback would be absolutely necessary. The observer position estimate is rather close to the actual position, as shown in Fig. 7. In Fig. 8 the estimate of velocity

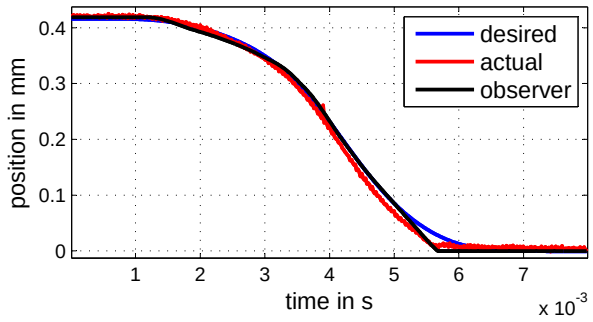


Fig. 7. One open-loop switching operation with estimated position.

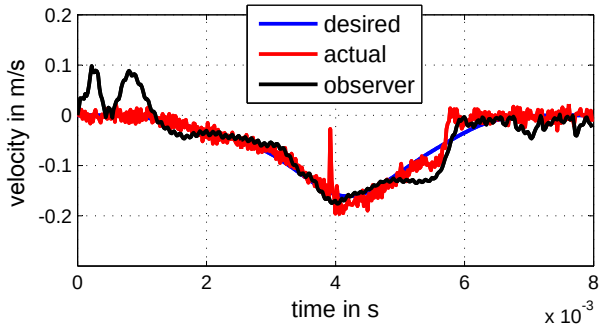


Fig. 8. One open-loop switching operation with estimated velocity.

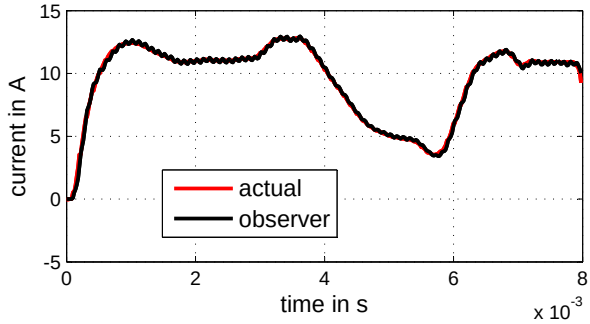


Fig. 9. One open-loop switching operation with estimated current.

versus the actual and desired trajectory, and in Fig. 9 the measured and estimated energizing current is shown. The observer can in principle correct the deviations from the normal operation. However, one has to be careful with the dimension of the feedback gains since the observer estimates are slightly slower than the plant quantities, as it can be seen by the Figs. 7 and Fig. 8.

VI. Conclusion

The problem of sensorless softlanding control has been investigated, where a possible feedforward control scheme as well as suitable nonlinear observer approaches were discussed. Preliminary experimental results have been presented for the proposed sensorless control strategy. In future work, further tests regarding the performance and robustness of the observer within the feedback control scheme will be conducted. To this end, the effect of varying fluid pressure levels has to be investigated.

References

[1] L. Behre, P. Mercorelli, U. Becker, and T. van Niekerk, "Rapid prototyping of a mechatronic engine valve controller for ic engines," in *Proc. of the 7th IFAC Symposium on Mechatronic Systems*, Loughborough University, UK, Sep 2016, pp. 54–58.

[2] R. Rothfuss, U. Becker, and J. Rudolph, "Controlling a solenoid valve: A distributed parameter approach," in *Proc. Mathematical Theory of Networks and Systems (MTNS)*, 2000.

[3] C. Stauch, J. Rudolph, and F. Schulz, "Some aspects of modelling, dimensioning, and control of digital flow control units," in *Proc. 7th Workshop on Digital Fluid Power, DFP15, Linz, Austria*, 2015, pp. 101–113.

[4] K. Lolenko and A. A. R. Fehn, "Modellbasierter Steuerungsentwurf für ein hydraulisches Bremssystem mit magnetischen Schaltventilen (Model-based Open-loop Control Design for a Hydraulic Brake System with Switching Solenoid Valves)," in *at - Automatisierungstechnik*, vol. 55, feb 2007, pp. 86–95.

[5] J. Reuter, "Flatness based control of a dual coil solenoid valve," in *4th IFAC Symposium on Mechatronic Systems*, Ruprecht-Karls-University, Germany, 2006, pp. 48–54.

[6] T. Glück, *Soft landing and self-sensing strategies for electromagnetic actuators*, ser. Modellierung und Regelung komplexer dynamischer Systeme, A. Kugi and K. Schlacher, Eds. Shaker, 2013.

[7] T. Braun and J. Reuter, "A distributed parameter approach for dual coil valve control with experimental validation," in *18th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR13)*, Miedzyzdroje, Poland, 2013, pp. 235 – 240.

[8] T. Braun, F. Straußberger, J. Reuter, and G. Preissler, "A semilinear distributed parameter approach for solenoid valve control including saturation effects," in *American Control Conference (ACC15)*, Chicago, IL, 2015, pp. 2600–2605.

[9] M. Rahman, N. Cheung, and K. W. Lim, "Position estimation in solenoid actuators," *IEEE Transactions on Industry Applications*, vol. 32, no. 3, pp. 552–559, May 1996.

[10] T. Glück, W. Kemmetmüller, C. Tump, and A. Kugi, "A novel robust position estimator for self-sensing magnetic levitation systems based on least squares identification," *Control Engineering Practice*, vol. 19, no. 2, pp. 146 – 157, 2011.

[11] F. Straußberger, M. Schwab, T. Braun, and J. Reuter, "Position estimation in electro-magnetic actuators using a modified discrete time class a/b model reference approach," in *American Control Conference (ACC14)*, Portland, OR, 2014, pp. 3686–3691.

[12] T. Braun, J. Reuter, and J. Rudolph, "Position observation for proportional solenoid valves by signal injection," in *7th IFAC Symposium on Mechatronic Systems (Mechatronics 2016), Loughborough, UK*, 2016.

[13] Q. Yuan and P. Li, "Self-sensing actuators in electrohydraulic valves," in *ASME International Mechanical Engineering Congress and Exposition*, Anaheim, CA, USA, Nov 2004, pp. 3686–3691.

[14] P. Mercorelli, "A two-stage sliding-mode high-gain observer to reduce uncertainties and disturbances effects for sensorless control in automotive applications," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 9, pp. 5929–5940, Sept 2015.

[15] T. Braun and J. Reuter, "Sliding-Mode Observation with Iterative Parameter Adaption for Fast-Switching Solenoid Valves," in *Variable-Structure Approaches for Analysis, Simulation, Robust Control and Estimation of Uncertain Dynamic Processes*, ser. Mathematical Engineering, (Series Editors: C. Hillermeier, J. Schröder, B. Weigand), A. Rauh and L. Senkel, Eds. Springer Int. Publish., 2016.

[16] E. Kallenbach, *Elektromagnets: basics, dimensioning, design and application (in German)*, 3rd ed. ViewegTeubner [GWV Fachverlage GmbH], Wiesbaden, 2008.

[17] R. Rothfuss, *Anwendung der flachheitsbasierten Analyse und Regelung nichtlinearer Mehrgrößensysteme*. Fortschrittberichte VDI, 1997, Reihe 8, Nr. 664.

[18] J. Rudolph, *Beiträge zur flachheitsbasierten Folgeregelung linearer und nichtlinearer Systeme endlicher und unendlicher Dimension*. Aachen: Shaker Verlag, 2003.

[19] ——, *Flatness based control of distributed parameter systems*. Aachen: Shaker Verlag, 2003.

[20] von Löwis J., *Flachheitsbasierte Trajektorienfolgeregelung elektromechanischer Systeme*. Aachen: Shaker Verlag, 2002.

[21] T. Braun and J. Reuter, "A position and velocity observer for single-coil digital solenoid valves including stability analysis," in *Submitted to American Control Conference 2017 (ACC17), Seattle, WA*, 2017.
[22] J. Adamy, *Nichtlineare Systeme und Regelungen*, 2nd ed. Springer Vieweg, 2014.
[23] O. Föllinger, *Nichtlineare Regelungen II*. Oldenbourg, 1993.
[24] M. Zeitz, *Nichtlineare Beobachter für chemische Reaktoren*, ser. Fortschrittberichte der VDI-Reihe 8; Nr. 27. VDI-Verlag, Düsseldorf, 1977.
[25] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.
[26] M. Fliess and J. Rudolph, "Local "tracking observers" for flat systems," in *Proc. Symposium on Control, Optimization and Supervision Computational Engineering in Systems Applications*, 1996, p. 213217.
[27] O. Föllinger, "Entwurf zeitvarianter Systeme durch Polvorgabe," *Automatisierungstechnik at*, vol. 26, no. 6, pp. 189–195, 1978.
[28] J. Winkler, S.-O. Lindert, K. Röbenack, and J. Rudolph, "Design of a nonlinear observer using automatic differentiation," *PAMM*, vol. 4, no. 1, pp. 147–148, 2004.
[29] K. Röbenack, *Beobachterentwurf für nichtlineare Zustandssysteme mit Hilfe des Automatischen Differenzierens*, ser. Berichte aus der Steuerungs- und Regelungstechnik. Shaker Verlag, Aachen, 2003.
[30] J.-J. E. Slotine, J. Hedrick, and E. Misawa, "On sliding observers for nonlinear systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, no. 3, pp. 245–252, 1987.
[31] V. Utkin, J. Guldner, and J. Shi, *Sliding Mode Control in Electro-Mechanical Systems*, 2nd ed., ser. Automation and Control Engineering. CRC Press, 2009.

Intelligent Fault Detection and Prognostics in Linear Electromagnetic Actuators – A Concept Description

Christian Knöbel *, Hanna Wenzl * and Johannes Reuter *
*Institute of System Dynamics, University of Applied Sciences Konstanz, Konstanz, Germany
Email: {cknoebel, hwenzl, jreuter}@htwg-konstanz.de

***Abstract*—In this paper two approaches for diagnostics and prognostics in linear electromagnetic actuators (*LEA*) are outlined. The introductory and problem formulation sections show the need for advanced diagnostic/prognostic methods compared to existing ones. Modelling and identifiability aspects are covered resulting in a mathematical model that is capable of accurately simulating the dynamic behaviour of a *LEA* and even depict deterioration phenomena. Data-driven approaches are presented as an interesting alternative if no model can be derived. It is shown how their output, e.g. with respect to wearout indicators, correlate. They are even suitable for fault detection and classification. Possible application scenarios are discussed and an outlook on future work is given.**

I. Introduction

With machinery becoming more complex and with a further increasing degree of automation, traditional maintenance strategies are no longer feasible, as industry and end consumers demand higher levels of security, reliability and cost-effectiveness from their assets or products. I.e. maintenance approaches like breakdown maintenance or preventive maintenance are rendered obsolete. This is on the one hand due to the accompanied high machine down time, man hour intensiveness as well as reduced machine availability and on the other hand due to incalculable costs.

To elude these drawbacks, a paradigm shift towards strategies that allow planning and scheduling of maintenance tasks has set in. As umbrella term predictive maintenance (*PM*) is widely used. *PM* can be separated in several approaches: Prognostics and Health Management (*PHM*, [1]), Integrated Systems Health Management (*ISHM*, [2]), Abnormal Event Management (*AEM*, [3]), Condition Based Maintenance (*CBM*, [4]) and Reliability Centered Maintenance (*RCM*, [5]). *PHM*, *AEM* and *ISHM* are per definition designated to very complex systems (like air- and spacecraft or whole chemical plants/processes) and include (besides the pure diagnosis/prognosis) also supply management, logistics and other business issues. Main aspects of *RCM* are influence assessment of maintenance actions on the system reliability and analysis of failure modes. *CBM* is often used as umbrella term for *PHM*, *ISHM* and *AEM*, although it focuses solely on pure diagnostic and prognostic tasks. Nevertheless all approaches have in common, that they take the actual asset state into account and base their maintenance decisions on it. As reliable diagnostic and prognostic tasks require an in-depth knowledge of the actual system state, research has focused on determining the health state of low-level components like bearings or electronic components. Therefore one does not only monitor the output signals of system components, but as well gathers additional information from e.g. vibration sensors.

II. Problem Formulation

A lot of work has been going on regarding diagnostics and prognostics of electrical machines and especially rotating machinery [6], [7], but for *LEA*s only some rudimentary approaches exist. They can mainly be found in patent literature and describe simple limit and threshold checking. But as application fields of *LEA*s constantly grow and they are used in safety relevant and critical applications (elevators, safety switches, etc.) there is a need for more sophisticated diagnostic and prognostic procedures that can detect various faults and estimate the remaining useful life (*RUL*) of the actuator. Figure 1 shows a schematic drawing of a *LEA*. It is used to transform electric energy via magnetic energy into mechanical energy. A constant voltage applied to the coil results in an electromagnetic force (current and magnetic flux build up) moving the plunger from its initial position to its end position (once reached, the coil current increases to its final value).

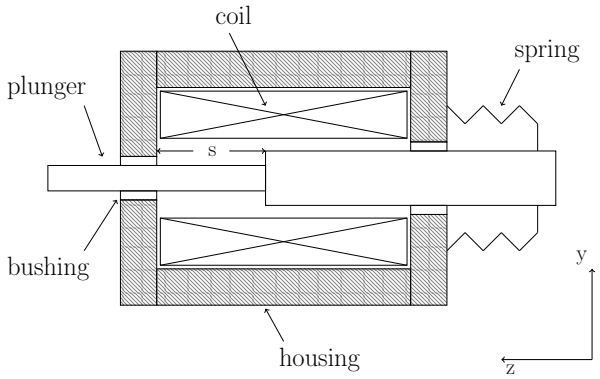


Fig. 1. Schematic drawing of a *LEA*. The assembly mainly consists of: housing, bushings, coil, plunger and spring.

When the voltage is switched off, current and hence magnetic flux reduce and the spring retracts the plunger to its initial position (spring force exceeds magnetic force). The direction of motion is in z-direction. One operating cycle is completed when the actuator has moved from its initial position to its end position and back to its initial position and current decayed to a stationary value.

Besides the aforementioned simple diagnostic procedures, one approach presented in [8] allows an in-depth examination of the actuator health state by measuring its magnetic characteristics. Major drawback of this method is, that the actuator itself is used as sensor and can't perform its intended task during a measurement. This is where our approach sets in and equips an allegedly simple actuator with diagnostic and prognostic capabilities without the use of additional sensors or interruption of normal operation. To accomplish this task, two approaches can be used: (a) model-based, where a mathematical model is derived from physical background [9], (b) data-driven, where the input and output data is used to extract information [10], [11]. Both approaches rely on extracted indicators that picture the current health state of the actuator under observation. These indicators are called features and can either be identified model parameters or abstract parameters calculated from e.g. time-frequency representations of the time-series signal [12].

A. Model-based Approach

For modelling a *LEA*, three coupled sub-systems are necessary to describe its dynamic behaviour: electrical, magnetic and mechanical. Usually the electrical and magnetic sub-models are merged leaving two sub-systems (mechanical and electro-magnetic) that interact through a coupling term $\Psi_z \dot{z}$. The model we use in our research can be seen in fig. 2 and equations (1)-(5). Model parameters are $\Theta = [f_1\ L_s\ h_z\ K]$. L_s describes flux leakage, h_z is used to calculate the eddy current resistance, K scales the influence of the back electromotive force (coupling term $\Psi_z \dot{z}$ and f_1 is a friction parameter. R_{Cu} is the copper resistance and calculated from stationary end values of current and voltage.

$$U_0 = iR_{Cu} + L_s\dot{i} + L_d\dot{i}_L + \Psi_z\dot{z} \tag{1}$$

$$U_i = i_w R_w = L_d\dot{i}_L + \Psi_z\dot{z} = U_0 - iR_{Cu} - L_s\dot{i} \tag{2}$$

$$i_w = i - i_L \tag{3}$$

$$\dot{i} = \frac{1}{L_s}(U_0 - iR_{Cu} - i_w R_w) \tag{4}$$

$$\dot{i}_L = \frac{1}{L_d}(U_0 - iR_{Cu} - L_s\dot{i} - \Psi_z\dot{z}) \tag{5}$$

The differential inductance L_d and Ψ_z are characteristic maps derived from static $\Psi(i)$ measurements of the specific actuator (plunger is fixed at different positions and $\Psi(i)$ measurements are performed). The mechanical sub-model is described in detail in [13] (please note that in [13] Ψ is used as state whereas here i and i_L are used). In contrast to classical friction models like LuGre [14] or Stribeck, the approach presented in [13] uses a position dependent adaptive friction characteristic (see fig. 3). This is due to the long stroke movement some actuators have (friction changes over stroke) and due to the fact that actuators with long stroke have often modified characteristics (force over stroke changes). For diagnostic purposes it is crucial to ensure that all model parameters are identifiable and have low deviations [15]. Many model-based diagnostic approaches neglect the identifiability aspect, but in our case identifiability is can be shown, equipping us with a reliable

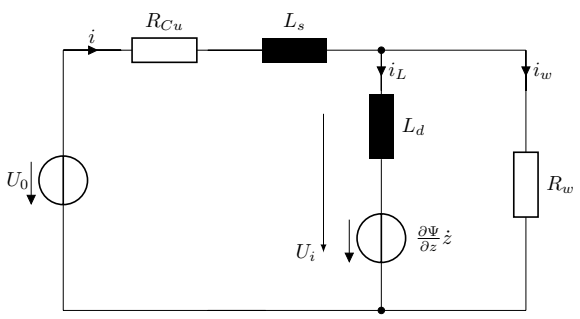


Fig. 2. Equivalent circuit of the electromagnetic system with main inductance, leakage inductance, eddy current resistance, coil resistance and back electromotive force.

model. The free parameters are: L_s - leakage inductance, f_x - friction parameter, h_z - eddy current resistance parameter and K - scaling parameter for the coupling term.

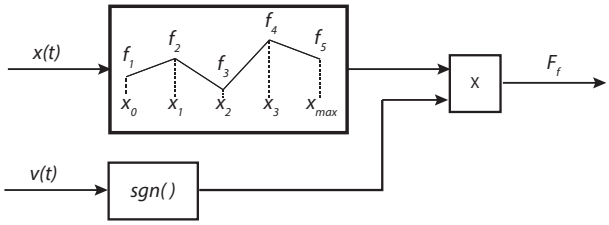


Fig. 3. Working principle of the adaptive position-dependent friction estimation.

B. Data-based Approach

Data-bases approaches are mainly used when no mathematical model can be derived or when the model becomes too complex. The realm of those approaches is to extract wear and fault indicators (features) solely based on measured input/output signals (no detailed system knowledge is necessary). In our case we can rely on coil current i_c and coil voltage U_c. To test the applicability of the approach, run-to-failure and emulated fault data was collected on test benches.

a) Run-to-Failure Data: As deterioration happens over time, accelerated test scenarios are used to generate wearout data within reasonable time. A test bench capable of driving ten *LEA*s in parallel is used to measure coil current and supply voltage of the operating actuators. The actuators are switched with a frequency of 1Hz, i.e. one complete cycle per second. Figure 4 shows current profiles of one actuator at different ageing stages. It is directly visible that information about wear and tear phenomena can be extracted from coil current. I.e. features have to be found that allow a discrimination between a new and a worn out actuator. These features can be identified by extracting information from healthy and deteriorated states. The current health-state is assessed by calculating distance metrics in feature space to either of the known states (*new* or *deteriorated*). Drawback of this approach is, that run-to-failure data must be readily available. A second method of generating

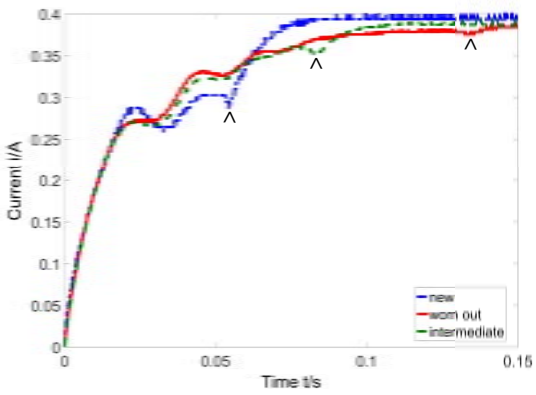


Fig. 4. Coil current at different ageing stages. The end of motion shifts gradually to the right (longer switching time) and is indicated with an arrow. The current ripple indicates acceleration and deceleration of the plunger during its movement.

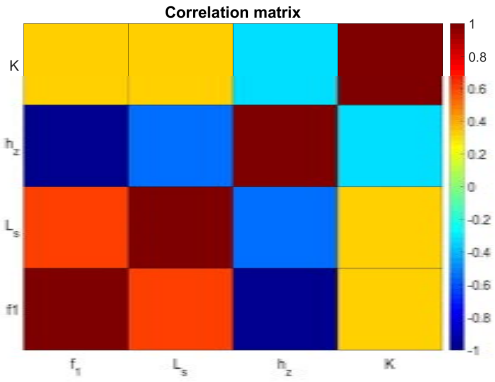


Fig. 5. Parameter correlations of the model given in eq. (1)-(X).

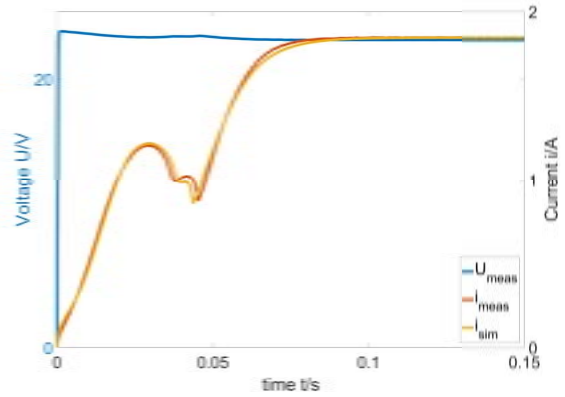


Fig. 6. Simulated current i_c using identified parameters.

wearout features is by learning the healthy state and using it as reference for residual calculation.

b) Emulated Fault Data and Classification: For emulating faults, a test bench is used, that allows manipulating stroke movement, supply voltage and initial position. Time and time-frequency-domain methods are used to generate features from measured coil current and voltage. After transformation and rating of the features, an optimised feature set is obtained representing different fault classes. By training a classifier, unseen data can be classified and hence faults detected. Detailed results and methods can be found in [16].

III. Results

The following chapter gives an overview of ongoing research. First the focus lies on model-based and then on data-based approaches.

A. Model Identification

The model parameter set was identified using measurement data i_c, U_c and verified with measured plunger position z. Figure 5 shows results of the identifiability analysis. Looking at the confusion matrix one can see that friction parameter f_1 and eddy current resistance parameter h_z show high negative correlation (strong interdependence) whereas the rest of the parameters show low to medium correlation. The parameter deviations are well below 15%. High deviations indicate poor identifiability of the respective parameters. Fig. 6 shows a simulated current profile which is in good accordance with the measured data. U_c is the input signal used to excite the actuator as well as the system model.

B. Data-driven Wear Estimation

Figure 7 shows different distance metrics calculated for ten *LEA*s over approximately 4.5 million switching cycles. In both graphs one can see how the distance to the reference states changes over time in feature space. Actuators that deteriorated much faster than the others are marked in red. These actuators show as well a different starting point compared to the rest, i.e. their initial distance to the healthy state is bigger at the beginning of the run-to-failure test. In fig. 8 switching times and the calculated reconstruction error for three actuators are shown. The progression of both features over time shows good accordance.

IV. Discussion

The identifiability analysis indicates that a model was found that can be used for diagnostics and prognostics on *LEA*s. But as for each actuator type a specific $\Psi(i)$ measurement has to be derived to capture the very own characteristics, the whole process of identification depends on the quality of the original $\Psi(i)$ measurement data. Furthermore, the number of friction parameters used in the mechanical model depend on the actuator under consideration. By sorting out the described problems, an automated model parametrisation and adaptation is possible leading to a self contained approach that only needs $\Psi(i)$ measurements as input. When no model can be derived, e.g. when no $\Psi(i)$ data is available, or the model is not identifiable (or does not reproduce measurement data well), data-based approaches are a good alternative to assess the health state. Although they do not provide physical features

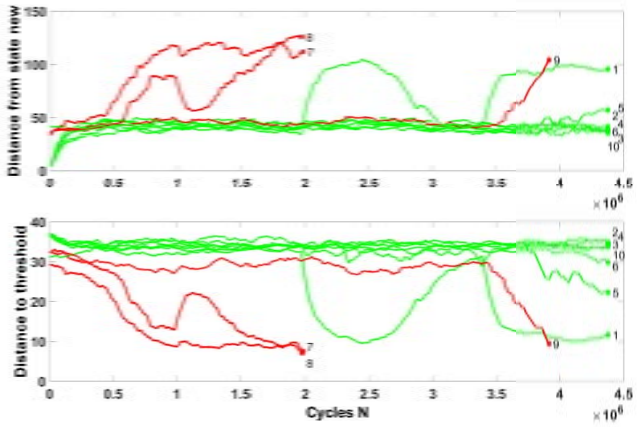


Fig. 7. Distance metrics in feature space. *N* are the complete switching cycles that were performed during the test. The upper graph shows the distance calculated with respect to the healthy/new state and its progression over lifetime. The lower graph shows the distance to a wearout threshold (defined in feature space as well).

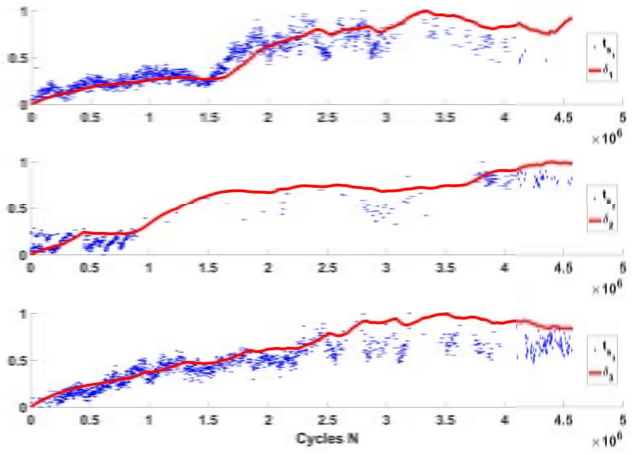


Fig. 8. Switching times and reconstruction error over lifetime. *N* are the complete switching cycles that were performed during the test. Missing switching times indicate that due to noise or bad signal quality no determination of the switching time was possible for the particular measurement.

(like friction or inductance), it is possible to extract fault and wearout related features that allow classification or monitoring of deterioration.

V. Conclusion and Future Work

A concept for linear electromagnetic actuator diagnostics and prognostics was outlined in this paper. Based on an identifiable mathematical model and data-driven approaches, fault detection and deterioration estimation is possible. A possible application scenario could be to spec *LEA*s in critical tasks with the ability of self diagnosis and remaining useful life prediction. Data-driven approaches are suitable for large assets where many identically constructed *LEA*s are in operation. Data acquisition and diagnostics/prognostics is performed centrally learning from the collected data. A combination of model and data-driven approaches might even improve the capabilities and enhance diagnostic depth.

Acknowledgment

The support of this work by the German Federal Ministry of Education and Research under grant 03FH034PX3 is gratefully acknowledged.

References

[1] A. Hess, G. Calvello, and P. Frith, "Challenges, issues, and lessons learned chasing the big p. real predictive prognostics - part 1," in *Aerospace Conference, 2005 IEEE*, 2005, pp. 3610–3619.

[2] M. Schwabacher and K. Goebel, "A survey of artificial intelligence for prognostics," in *Association for the Advancement of Artificial Intelligence AAAI Fall Symposium 2007*, 2007, pp. 107–114.

[3] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis part i: Quantitative model-based methods," *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 293–311, 2003.

[4] R. Kothamasu, S. H. Huang, and W. H. VerDuin, "System health monitoring and prognostics a review of current paradigms and practices," in *Handbook of Maintenance Management and Engineering*, M. Ben-Daya, S. O. Duffuaa, A. Raouf, J. Knezevic, and D. Ait-Kadi, Eds. Springer London, 2009, pp. 337–362.

[5] J. Moubray, *Reliability-centered maintenance*, 2nd ed. New York: Industrial Press, 2001.

[6] Yao Da, Xiaodong Shi, and M. Krishnamurthy, "Health monitoring, fault diagnosis and failure prognosis techniques for brushless permanent magnet machines," in *Vehicle Power and Propulsion Conference (VPPC), 2011 IEEE*, 2011, pp. 1–7.

[7] S. Nandi, H. A. Toliyat, and Xiaodong Li, "Condition monitoring and fault diagnosis of electrical motors-a review," *Energy Conversion, IEEE Transactions on*, vol. 20, no. 4, pp. 719–729, 2005.

[8] A. Gadyuchko and E. Kallenbach, "Magnetische messung-neue wege der funktionsprfung bei der herstellung von magnetaktoren," *ETG-Fachbericht-Innovative Klein-und Mikroantriebstechnik*, 2010.

[9] R. Isermann, "Model-based fault-detection and diagnosis status and applications," *Annual Reviews in Control*, vol. 29, no. 1, pp. 71–85, 2005.

[10] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Remaining useful life estimation a review on the statistical data driven approaches," *European Journal of Operational Research*, vol. 213, no. 1, pp. 1–14, 2011.

[11] P. Esling and C. Agon, "Time-series data mining," *ACM Computing Surveys*, vol. 45, no. 1, pp. 1–34, 2012.

[12] E. G. Strangas, S. Aviyente, and S. Zaidi, "Timefrequency analysis for efficient fault diagnosis and failure prognosis for interior permanent-magnet ac motors," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4191–4199, 2008.

[13] H. Wenzl, C. Knoebel, J. Reuter, and H. Aschemann, Eds., *Adaptive Position-Dependent Friction Characteristics for Electromagnetic Actuators*, 2016.

[14] Canudas de Wit, C., H. Olsson, K. J. Astrom, and P. Lischinsky, "A new model for control of systems with friction," *IEEE Transactions on Automatic Control*, vol. 40, no. 3, pp. 419–425, 1995.

[15] S. Wirtensohn, H. Wenzl, T. Tietz, and J. Reuter, "Parameter identification and validation analysis for a small usv," in *Methods and Models in Automation and Robotics (MMAR), 2015 20th International Conference on*, 2015, pp. 701–706.

[16] C. Knbel, Z. Marsil, M. Rekla, J. Reuter, and C. Ghmann, "Fault detection in linear electromagnetic actuators using time and time-frequency-domain features based on current and voltage measurements," in *Methods and Models in Automation and Robotics (MMAR), 2015 20th International Conference On Methods and Models in Automation and Robotics*, IEEE, Ed., 2015.

Secure Zero Configuration of IoT Devices - A Survey

Kevin Wallis and Christoph Reich

***Abstract*— It is forcasted that with the Internet of Things (IoT) the number of devices is increasing enormously. Up to 34 billion devices will be connected to the Internet by 2020. All of them have to be configured. For example adding a new device, replacing an old one, renew the expired certificates, patch the device software, etc. Zero Configuration (ZeroConf) is the approach to tackle the immense administrative effort. The configuration consums time, resources and is error-prone, which often leads to security and compliance violations. Another important aspect is securing the communication between these devices otherwise a malicious attack could change the configuration of a device and harm the whole system. Many of the actual ZeroConfig systems do not take the accompanying security risks into account, so a SecZeroConf approach is required.**

This paper gives a survey about Zero Configuration (ZeroConf) and Secure Zero Configuration (SecZeroConf) and discusses the state of the art and potential missing features.

I. INTRODUCTION

An application domain that integrates different technological and social fields is called Internet of Things (IoT)[1]. Up to 34 billion devices will be connected to the Internet by 2020.[2] So, a trouble-free communication and interaction between those devices is necessary. One possibility to achive this is Zero Configuration. *"The goal of the Zero Configuration Networking (ZEROCONF) Working Group is to enable networking in the absence of configuration and administration. Zero configuration networking is required for environments where administration is impractical or impossible, such as in the home or small office, embedded systems 'plugged together' as in an automobile, or to allow impromptu networks as between the devices of strangers on a train."*[3] In a nutshell, it is the process of automatically configuring a device without additional user intervention. Often the term is only used in combination with the Network layer than it is called Zero Configuration Networking. Examples for ZeroConf frameworks are Apple's Bonjour[4], Avahi[5] and Mono.Zeroconf[6]. Xiaolong et al. [7] show security risks of Apple's Bonjour and Farhan et al. [8] investigated implementation, performance and security of Avahi and Mono.ZeroConf. The term SecZeroConf describes a process where in addition to automatically configuring devices also security is taken into account. Some necessary security objectives for ZeroConf are described in III.

II. PROBLEM DESCRIPTION

A variety of applications exists, where ZeroConf in combination with security make sense. In the following two examples for different scopes are given: II-A - *Industrie 4.0* and II-B - *Smart Home*.

A. INDUSTRY 4.0

A machine has vision sensors which are responsible for measuring the quality and counting the amount of produced products. The measurements from the vision sensors are sent to a server. Each of those vision sensors has an own configuration this configuration depends on the position. For example the input material has a different shape in comparison with the output material and there are two sensors, one for the input and one for the ouput each of these sensors needs another configuration for measuring the quality. When one of those vision sensors is broken it is necessary to replace it with a new one. This new sensor should automatically know the position and also depending on this position automatically configure. Different approaches for the automatic configuration are shown in VII. The measured amounts should only be available for the management so that no competitor can take market specific benefits of them.

B. SMART HOME

A smart home with light sensors and presence sensors in each room is given. The measured values from the light sensors in combination with the presence measurements are used for automatically controlling the lights. When a light sensor is replaced, the new sensor should automatically know where it is placed. Also a default configuration is needed. This configuration is responsible for defining the measuring times. For example during nighttime the number of measurings should be less as during daytime. The presence sensor configuration should only be defined once and automatically configured on any suitable sensor.

III. SECURITY OBJECTIVES

The security objectives differ depending on the scenario. In this paper we use the five defined security objectives for Smart Grid and Smart Home Security from Komninos et al. [9] and apply them on our two different problem scenarios from II.

- **Confidentiality:** the assurance that data will be disclosed only to authorized individuals or systems.
- **Authenticity:** the validation that communicating parties are who they claim they are, and that messages supposedly sent by them are indeed sent by them.
- **Integrity:** the assurance that the accuracy and consistency of data will be maintained. No unauthorized modifications, destruction or losses of data will go undetected.
- **Availability:** the assurance that any network resource (data/bandwidth/equipment) will always be available for

any authorized entity. Such resources are also protected against any incident that threatens their availability.
- **Authorization:** the assurance that the access rights of every entity in the system are defined for the purposes of access control.

The two security objectives authenticity and authorization are strongly related because a not authenticated user does also not have any authorization.

A. INDUSTRY 4.0

Confidentiality is important for the given scenario because no competitor should get any information about the produced product amount. It should not be possible for competitors to introduce fake amount measurings into the system, so authenticity of measurements is also needed. The third security objective - integrity - ensures that a received measured amount is not manipulated. Availability of the sensors is also necessary otherwise the management cannot take any further steps depending on the current production. No external organisation should have access to the system, so authorization is important. It is also used for securing the access only to the management and no other employee. Michael et al. [10] gives a more detailed insight into industry 4.0 security objectives.

B. SMART HOME

Using a smart home contains a lot of risks, which are not always obvious at the first sight. For example when burglars have access to the smart measurements they are able to figure out when no one is at home. To disable this possibility confidentiality and authenticity are needed. Also integrity should be considered because otherwise the burglars could actively manage the smart home. The other two security objectives: authorization and availability are not as important as they are for example III-A. A detailed investigation of smart home objectives are done by Komninos et al. [9]

The conclusion from the security objectives of the given problem scenarios shows that the importance of the specific security objectives depends on the scenario itself. After all the three corresponding objectives of both examples are authenticity, integrity and confidentiality.

IV. ZERO CONFIGURATION

Aidan Williams defined the four requirements[11] for ZeroConf:

- IP interface configuration
- Translation between host name and IP address
- IP multicast address allocation
- Service discovery

Each of these requirements will be described in the following.

A. IP INTERFACE CONFIGURATION

The general IP interface configuration can be done in two different ways. If a Dynamic Host Configuration Protocol (DHCP) server is present, the server provides an IP address for the device and no further IP interface configuration is needed. The second approch is a manual assignment, which is usually done by an administrator. Using a DHCP server or a manual configuration needs a central authority for policing the IP address allocation. ZeroConf uses a distributed approach, where each device is responsible for choosing and verifiyng its own address. The IPv4 address range for ZeroConf is between 169.254.0.0 to 169.254.255.255 and described in the RFC 3927 as link-local address range. For generating ZeroConf addresses the Address Resolution Protocol (ARP, IETF RFC 826) is used. First of all you have to chose a link-local address, this is done by random. If the particular address is already in use, an new random address has to be generated and verified. The verification is done by ARP requests. In general ARP requests are used for discovering the MAC address of a machine to a given IP address. For ZeroConf the request uses the generated IP address and checks that none responses to the request, these requests are called ARP probes. *"RFC 3927 recommends that the host send three probe packets spaced one to two seconds apart [...]."*[12, p. 24] If an ARP probe receives a response, a new IP address has to be generated and verified again. After successfully verifiying an IP address, the address has to be announced to the other hosts on the network by sending two ARP announcement-requests.

B. HOST NAME AND IP ADDRESS TRANSLATION

For a locally unique name a Domain Name System (DNS) is necessary. Setting up and running a DNS server needs an administrator, which is impractical and not the aim of ZeroConf, so another approach Multicast DNS (mDNS, RFC6762), was developed. The reasons for the translation between host name and IP address are the change of a given address over time and the not human-friendly form of IP especially IPv6 with 32 hexadecimal characters.[12, p. 32] When a client uses mDNS and wants to do a query, the query is not send to a centralized authority, instead a IP Multicast is done. Each device on the local network listens to the multicast. A device answers the query when the query is addressed to it. The responsible software for listening and answering the queries is called mdnsd on Unix, mDNSResponder on OS X (macOS) and mDNSResponder.exe on Windows. For sending a mDNS query the IP address 224.0.0.251[13] is reserved. The procedure for verifiying and announcing the host name uses the same approach as the IP interface configuration. After choosing a hostname a probing to check for uniqueness is needed. This is done by creating an address record of type A - the different record types are documented in the RFC 1035. The created record is send to the multicast address three times, with a 250ms waiting time between each query and a query type of T_ANY. So all records, which match the given record are returned to this query. If some device already uses the selected hostname, a new

name will be selected and the verification starts again. After successfully verifying the hostname an announcing must be performed. The verified hostname is send via multicast to each device in the network and the mDNS responders send a mDNS response with all of there mDNS records. Because every device of the network listens to this messages, they are able to update their record caches.

C. IP MULTICAST ADDRESS ALLOCATION

For the multicast address allocation the ZeroConf Multicast Address Allocation Protocol[14] (ZMAAP) could be used. This protocol was defined by the IETF but never published. ZMAAP uses an Address In Use (AIU) message in combination with a Mini- Multicast Address Allocation Server (mini-MAAS) to coordinate multicast address allocations. An application sends a request for a specified amount of multicast addresses. The local mini-MAAS creates multicast address proposals for the request and send them to other mini-MAAS. If after a specified time an AIU message from another mini-MAAS is received new multicast address proposals will be created otherwise a successful allocation is assumed.

D. SERVICE DISCOVERY

If the IP address or the hostname of a service is known, a user is able to execute it. On a network where many services are provided users do not know the explicit name of the service but they know the kind of the service for example a printing service. So it should be possible to get or browse all available services in the network. If ZeroConf is used, a DNS Service Discovery (DNS-SD) is supported. The DNS-SD builds on the defined DNS protocol family, especially on the service discovery (SRV) record type, which is specified in RFC 2782 - *"A DNS RR for specifying the location of services (DNS SRV)"*. In table I some reasons for using the existing DNS technology for service discovery described by Stuart Cheshire et al. [12] are shown.

TABLE I
OVERVIEW OF PROVIDED PROPERTIES BY DNS

Needed property	Existing
central aggregation server	DNS server
service registration protocol	DNS dynamic update
query protocol	DNS
security mechanism	DNSSEC
multicast mode for ad-hoc networks	ZeroConf already requires a multicast-based protocol

For a name resolution a client sends a DNS request for the SRV record of the name. The result is the SRV record with the target port and host. A more detailed explanation is given by Cheshire - *Discovering Named Instances of Abstract Services using DNS*[15].

V. ZEROCONF TECHNOLOGIES

In the following the three ZeroConf frameworks Apple's Bonjour, Avahi and Mono.Zeroconf are short described.

- **Apple's Bonjour:** was formerly called Rendezvous and is the ZeroConf framework implementation from Apple. The framework is available on two operating systems: macOS and Microsoft Windows. It used for discovering services like printing, file sharing and collaborative document editing. Bonjour can only be used in a single broadcast domain and does not support advertising services to the public Internet.
- **Avahi:** is originally implemented for Linux systems but could be used on other systems as well because it is implemented in C. The framework passes all Bonjour conformance tests.
- **Mono.ZeroConf:** is a cross platform for Mono and .NET, which provides a unified API for the most common ZeroConf operations. So it is possible to abstract the differences between different providers like mDNSResponder and Avahi.

Original ZeroConf was not implemented to be secure but the given examples in II and III are showing the importance of investigating possible security enhancements for ZeroConf. The identified security weaknesses by Siddiqui et al. [8] are used as base for vulnerabilities overview shown in table II.

TABLE II
OVERVIEW OF VULNERABILITIES IN ZEROCONF NETWORKS

Vulnerability	Description
Configuration of addresses	The automatic ip address assignment makes ARP poisining (also called ARP spoofing) possible. A unauthorized user can impersonate a service and get all the messages for the origin service.
Hostname to IP address translation	The automatic translation between hostname and address can be attacked by DNS poisoning. This means the existing hostname - IP address table gets altered and the hostname resolving returns a malicious service address.
Allocation of multicast addresses	If a malicious host always responses to multicast address request the mini-MAAS starts to reuse already assigned addresses, this is called address hijacking.
Service discovery	The service discovery is based on DNS and allows DNS poisoning. This leads to the same result as in the case of the DNS poisoning from the hostname to IP address translation, the attacker can inject malicous service addresses.

Denial of service (DoS) attacks could be produced from each of the listed vulnerabilities. For example an attacker responses to every ARP request, which are used for testing the availability of an IP address - so each IP address is marked as used. Another possible DoS attack is sending recursively DNS requests.[16]

VI. STATE OF THE ART - SECURITY

ARP and DNS protocols are already investigated in connection with security. In the following some of the latest investigation results and some state of the art security technologies like IPSec are explained.

A. ARP - SECURITY

The latest paper from Cox et al. [17] describes a Network Flow Guard for ARP (NFGA), a Sofware Defined Networking (SDN) module. The Network Flow Guard augments an SDN controller with the ability to detect and prevent ARP replies from unauthorized hosts. This is done by monitoring the DHCP messages (offers, requests and acknowledgements) and constructing a dynamic table with an entry consisting of MAC:IP:port:fixed:state associations for each device in the network. By using this entries ARP spoofs can be blocked after NFG detects the first spoofed packet. For using this security technology a OpenFlow[18] switch is required.

A second approach for securing ARP is arpsec[19]. arpsec uses a Trusted Platform Module (TPM)[20]. The TPM supports cryptographic functions, unique identity, random number generation and secure storage. In a nutshell, it covers inegrity protection, isolation and confidentiality. The arpsec approach does not alter the existing ARP, instead it formalizes the ARP system binding using logic and uses a logic prover for verifying an ARP reply against defined logic rules in combination with the previously stored binding history. If the logic layer fails the implemented TPM attestation protocol will be used. This protocol is used to determine the trustworthiness af a network device.

Oh et al. [21] describes a security improvement by installing an anti-ARP spoofing agent (ASA). This agent blocks potentially insecure communications and intercepts unauthenticated exchanging of ARP packets. In system, where ASA is used only the ASA agent has the authority to update the ARP cache table. The big advantages of this approach are: no secure server is required and the existing protocol implementation does not need to be modified.

B. DNS - SECURITY

Based on the first published RFC (2065)[22] on securing DNS in 1997 several additional investigations were done. The basic concept behind them is the usage of digital signatures by public-key cryptography. Every DNS server is gets a key-pair (private and public). If a DNS server sends a message the message will be signed with the private key and can be verified with the public from the sender. In general one or more authenticated DNS root public keys are known within the network. The public root keys are used for creating certificates otherwise the public key of each DNS server has to be stored on the receiver of signed messages. A best practice approach for using DNSSec in combination with the Berkeley Internet Name Domain (BIND) was described by Jalalzai et al. [16] *BIND is open source software that implements the Domain Name System (DNS) protocols for the Internet. It is a reference implementation of those protocols, but it is also production-grade software, suitable for use in high-volume and high-reliability applications.*[23]

Another approach for DNSSec is proposed by Ateniese et al. [24] The approach uses primarly symmetric instead of asymmetric cryptography.

Zhu et al. [25] suggests a *Connection-Oriented DNS to Improve Privacy and Security*. Normally DNS requests are send via UDP, which means connectionless. The connection-oriented approach suggests instead using TCP in combination with the transport layer security (TLS) - this is called T-DNS. Using TLS provides privacy between a user and a DNS server. The stateful protocol has a performance disadvantage in comparison with UDP of about 22%.

C. OTHER SECURITY APPROACHES

In the year 2000, when ZeroConf requirements were investigated also the security of ZeroConf was taken into account.[26] One of the most promising security technologies was the Internet Protocol Security (IPSec, RFC 4301)[27]. IPSec adds an additional layer between the IP and the TCP layer. The attached layer is responsible for authenticity, integrity and confidentiallity. A virtual private network (VPN) could be established by using IPSec in tunnel mode.

Trusted Neighborhood Discovery (TND)[28] is a decentralized security approach for critical infrastructures, which uses a TPM for each device in the system. In general every device has a neighbourhood with other devices. These neighbours are constantly monitoring each other and when some malicous/suspicious behaviour is detected a message to a monitoring server will be send. The monitoring server can raise alerts and inform a administrator, correlate reports and inudce reactions for example removing the malicious neighbour from the system.

VII. SECURE ZERO CONFIGURATION

For a SecZeroConf approach two key points should be considered. The first point is a device identifier, which should be unique and tamper-proof, so that authenticity is secured. This is also part of the ZeroConf requirements. *Note that in general, devices running zeroconf protocols must trust the other devices in the group because any device may claim to be using an address or name, or advertising a service.*[p. 16][11] For the second point keys (private and public) are considered, they are used for encryption, decryption and also for signing. Using the explained keys would lead to confidentiallity and integrity. So the three corresponding security objectives: authenticity, confidentiallity and integrity from III could be supported. Taking these three security objectives into account leads in combination with the state of the art - security from VI to the usage of a TPM in combination with signing. Every device in the system gets a TPM. The TPM

has a unique identification id, a public key pub_d, a private key pri_d, the public key pub_{prod} from the producer and a signed identification id_s- this is the identification encrypted by the private key pri_{prod} from the producer. If a device tries to connect to the network via ZeroConf the pub_{prod} will be compared. Should the keys differ the device is not able to connect, a reason could be another producer or a malicious device. If the keys are equal the id_s is decrypted by using pub_{prod} and verified against the id. Is the verification successful authentification is secured. For communication integrity and confidentiallity between two devices $d1$ and $d2$ the keys pub_{d1}, pri_{d1} and pub_{d2}, pri_{d2} are used. The pub_d can be shared between the devices. For sending a message m_{d1} from $d1$ to $d2$ the m_{d1} is encrypted with pub_{d2} then send to $d2$ and there decrypted with pri_{d2}. The explained approach prevents the identified problem from Williams: *"The usual approach taken to secure radio and powerline networks is to rely on some form of Layer-2 encryption. Unfortunately, this approach would prevent new devices from using zeroconf protocols at all until they are configured with some kind of key which allows them to access the network medium. Zeroconf protocols in the home are attractive because they don't require the users to be network engineers in order to plug devices in and have them work properly."*[26, p. 3] because no additional user intervention is needed.

VIII. CONCLUSION AND OUTLOOK

ZeroConf is not necessary unsecure. Like the given approach in VII by using a TPM the three security objectives authenticity, integrity and confidentiallity could be achieved. After all further investigation are needed: taking other security objectives into account, a signing authority (more than one producer should be able to sign devices), the usage of existing protocols in combination with the SecZeroConf-TMP approach and a prototyp implementation.

REFERENCES

[1] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the internet of things (iot)", IEEE Internet Initiative, Tech. Rep., 2015.

[2] P. Middleton, J. Tully, K. F. Brant, *et al.*, "Forecast: Internet of things, endpoints and associated services, worldwide, 2014", Gartner, Tech. Rep., 2014.

[3] Z. W. Group. (2016). Zero configuration networking (zeroconf), Internet Engineering Task Force (IETF), [Online]. Available: http://www.zeroconf.org/zeroconf-charter.html (visited on 10/29/2016).

[4] Apple. (2016). Bonjour, Apple, [Online]. Available: https://developer.apple.com/bonjour/ (visited on 10/21/2016).

[5] Avahi. (2016). Avahi, Avahi, [Online]. Available: http://avahi.org/ (visited on 10/28/2016).

[6] M. Project. (2016). Mono.zeroconf, Mono Project, [Online]. Available: http://www.mono-project.com/archived/monozeroconf/ (visited on 10/22/2016).

[7] X. Bai, L. Xing, N. Zhang, *et al.*, "Staying secure and unprepared: Understanding and mitigating the security risks of apple zeroconf", in *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016, pp. 655–674. DOI: 10.1109/SP.2016.45.

[8] F. Siddiqui, S. Zeadally, T. Kacem, and S. Fowler, "Zero configuration networking: Implementation, performance, and security", *Computers & Electrical Engineering*, vol. 38, no. 5, pp. 1129–1145, 2012.

[9] N. Komninos, E. Philippou, and A. Pitsillides, "Survey in smart grid and smart home security: Issues, challenges and countermeasures", *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1933–1954, Fourthquarter 2014, ISSN: 1553-877X. DOI: 10.1109/COMST.2014.2320093.

[10] M. Waidner and M. Kasper, "Security in industrie 4.0 - challenges and solutions for the fourth industrial revolution", in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, Mar. 2016, pp. 1303–1308.

[11] A. Williams. (2016). Requirements for automatic configuration of ip hosts, Internet Engineering Task Force (IETF), [Online]. Available: http://files.zeroconf.org/draft-ietf-zeroconf-reqts-12.txt (visited on 10/28/2016).

[12] S. Cheshire and D. H. Steinberg, *Zero configuration networking: The definitive guide - the definitive guide*.

Sebastopol: "O'Reilly Media, Inc.", 2005, ISBN: 978-1-449-39079-2.

[13] S. Venaas. (2016). Ipv4 multicast address space registry, IANA, [Online]. Available: http://www.iana.org/assignments/multicast-addresses/multicast-addresses.xhtml (visited on 10/29/2016).

[14] IETF. (2016). Zeroconf multicast address allocation protocol (zmaap), IETF, [Online]. Available: http://files.zeroconf.org/draft-ietf-zeroconf-zmaap-02.txt (visited on 11/03/2016).

[15] S. Cheshire. (2016). Discovering named instances of abstract services using dns, Apple Computer, [Online]. Available: http://quimby.gnus.org/internet-drafts/draft-cheshire-dnsext-nias-00.txt (visited on 11/03/2016).

[16] M. H. Jalalzai, W. B. Shahid, and M. M. W. Iqbal, "Dns security challenges and best practices to deploy secure dns with digital signatures", in *2015 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, Jan. 2015, pp. 280–285. DOI: 10.1109/IBCAST.2015.7058517.

[17] J. H. Cox, R. J. Clark, and H. L. Owen, "Leveraging sdn for arp security", in *SoutheastCon 2016*, Mar. 2016, pp. 1–8. DOI: 10.1109/SECON.2016.7506644.

[18] N. McKeown, T. Anderson, H. Balakrishnan, *et al.*, "Openflow: Enabling innovation in campus networks", *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008, ISSN: 0146-4833. DOI: 10.1145/1355734.1355746. [Online]. Available: http://doi.acm.org/10.1145/1355734.1355746.

[19] J. (Tian, K. R. Butler, P. D. McDaniel, and P. Krishnaswamy, "Securing arp from the ground up", in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '15, San Antonio, Texas, USA: ACM, 2015, pp. 305–312, ISBN: 978-1-4503-3191-3. DOI: 10.1145/2699026.2699123. [Online]. Available: http://doi.acm.org/10.1145/2699026.2699123.

[20] ISO(IEC 11889-1:2015, *Information technology - trusted platform module library part 1: Architecture*. ISO, Geneva, Switzerland.

[21] M. Oh, Y. G. Kim, S. Hong, and S. Cha, "Asa: Agent-based secure arp cache management", *IET Communications*, vol. 6, no. 7, pp. 685–693, May 2012, ISSN: 1751-8628. DOI: 10.1049/iet-com.2011.0566.

[22] D. Eastlake and C. Kaufman. (2016). Rfc 2065: Domain name system security extensions, CyberCash, [Online]. Available: https://tools.ietf.org/html/rfc2065 (visited on 11/04/2016).

[23] I. Internet Systems Consortium. (2016), Internet Systems Consortium, Inc., [Online]. Available: https://www.isc.org/downloads/bind/ (visited on 11/04/2016).

[24] G. Ateniese and S. Mangard, "A new approach to dns security (dnssec)", in *Proceedings of the 8th ACM Conference on Computer and Communications Security*, ser. CCS '01, Philadelphia, PA, USA: ACM, 2001, pp. 86–95, ISBN: 1-58113-385-5. DOI: 10.1145/501983.501996. [Online]. Available: http://doi.acm.org/10.1145/501983.501996.

[25] L. Zhu, Z. Hu, J. Heidemann, *et al.*, "Connection-oriented dns to improve privacy and security", in *2015 IEEE Symposium on Security and Privacy*, May 2015, pp. 171–186. DOI: 10.1109/SP.2015.18.

[26] A. Williams. (2016). Securing zeroconf networks, Internet Engineering Task Force, [Online]. Available: https://tools.ietf.org/html/draft-

williams-zeroconf-security-00 (visited on 11/03/2016).

[27] S. Kent and K. Seo. (2016), BBN Technologies, [Online]. Available: https://tools.ietf.org/html/rfc4301 (visited on 11/04/2016).

[28] N. Göttert, N. Kuntze, C. Rudolph, and K. F. Wahid, "Trusted neighborhood discovery in critical infrastructures", in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*, Nov. 2014, pp. 976–981. DOI: 10.1109/SmartGridComm.2014.7007775.