EMERGING 2020

The Twelfth International Conference on Emerging Networks and Systems Intelligence

ISBN: 978-1-61208-815-0

October 25 - 29, 2020

EMERGING 2020 Editors

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

Claus-Peter Rückemann, Westfälische Wilhelms-Universität Münster (WWU) / DIMF / Leibniz Universität Hannover, Germany

EMERGING 2020

Forward

The Twelfth International Conference on Emerging Networks and Systems Intelligence (EMERGING 2020), held on October 22-29, 2020, constituted a stage to present and evaluate the advances in emerging solutions for next-generation architectures, devices, and communications protocols. Particular focus was aimed at optimization, quality, discovery, protection, and user profile requirements supported by special approaches such as network coding, configurable protocols, context-aware optimization, ambient systems, anomaly discovery, and adaptive mechanisms.

Next-generation large distributed networks and systems require substantial reconsideration of exiting 'de facto' approaches and mechanisms to sustain an increasing demand on speed, scale, bandwidth, topology and flow changes, user complex behavior, security threats, and service and user ubiquity. As a result, growing research and industrial forces are focusing on new approaches for advanced communications considering new devices and protocols, advanced discovery mechanisms, and programmability techniques to express, measure and control the service quality, security, environmental and user requirements.

The conference had the following tracks:

- Technology and networking trends
- Quality and optimization

We take here the opportunity to warmly thank all the members of the EMERGING 2020 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to EMERGING 2020. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also gratefully thank the members of the EMERGING 2020 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that EMERGING 2020 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the field of emerging networks and systems intelligence.

EMERGING 2020 General Chair

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

EMERGING 2020 Steering Committee

Raj Jain, Washington University in St. Louis, USA
Robert Bestak, Czech Technical University in Prague, Czech Republic
Emilio Insfran, Universitat Politecnica de Valencia, Spain
Euthimios (Thimios) Panagos, Perspecta Labs, USA
Dimitris Kanellopoulos, University of Patras, Greece
Jason Zurawski, Lawrence Berkeley National Laboratory / Energy Sciences Network (ESnet), USA
Linda R. Elliott, Army Research Laboratory, USA
Theodor D. Popescu, National Institute for Research and Development in Informatics Bucharest, Romania

EMERGING 2020 Publicity Chair

Daniel Andoni Basterrechea, Universitat Politecnica de Valencia, Spain

EMERGING 2020

COMMITTEE

EMERGING 2020 General Chair

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

EMERGING Steering Committee

Euthimios (Thimios) Panagos, Perspecta Labs, USA
Dimitris Kanellopoulos, University of Patras, Greece
Jason Zurawski, Lawrence Berkeley National Laboratory / Energy Sciences Network (ESnet), USA
Linda R. Elliott, Army Research Laboratory, USA
Raj Jain, Washington University in St. Louis, USA
Theodor D. Popescu, National Institute for Research and Development in Informatics Bucharest, Romania
Robert Bestak, Czech Technical University in Prague, Czech Republic
Emilio Insfran, Universitat Politècnica de València, Spain

EMERGING 2020 Publicity Chair

Daniel Andoni Basterrechea, Universitat Politecnica de Valencia, Spain

EMERGING 2020 Technical Program Committee

Andrea F. Abate, University of Salerno, Italy
Dorel Aiordachioaie, Dunarea de Jos University of Galati, Romania
Mohammed GH. AL Zamil, Yarmouk University, Jordan
Firkhan Ali Bin Hamid Ali, Universiti Tun Hussein Onn Malaysia, Malaysia
Nik Bessis, Edge Hill University, UK
Robert Bestak, Czech Technical University in Prague, Czech Republic
Graziana Cavone, Polytechnic of Bari, Italy
Chin-Chen Chang, Feng Chia University, Taiwan
Hyung Jae (Chris) Chang, Troy University, USA
Hongxu Chen, University of Technology, Sydney
Arun Das, Visa Inc., USA
David de Andrés, Universitat Politècnica de València, Spain
Ramadan Elaiess, University of Benghazi, Libya
Linda R. Elliott, Army Research Laboratory, USA
Nuno Gonçalves Rodrigues, Polytechnic Institute of Bragança, Portugal
Chunhui Guo, San Diego State University, USA
William Hurst, Liverpool John Moores University, UK
Sergio Ilarri, University of Zaragoza, Spain
Emilio Insfran, Universitat Politecnica de Valencia, Spain
Raj Jain, Washington University in St. Louis, USA
Jin-Hwan Jeong, SK telecom, Republic of Korea

Copyright Information

Table of Contents

Dynamic Content Adaptation for Timely Delivery of Critical Data under Network Congestion

Jia Lei Du, Stefan Linecker, Rodrigo Pommot
Advanced Networking Center, Salzburg Research
Salzburg, Austria
Email: {jia.du, stefan.linecker}@salzburgresearch.at

Reinhard Mayr
COPA-DATA GmbH
Salzburg, Austria
Email: reinhardm@copa-data.com

***Abstract*—The ever increasing efficiency in sensor-actuator electronics and data transmission technologies is enabling another phase in the digital transformation in the form of billions of connected Cyber-Physical Systems (CPS). As CPS are grounded in the real-world, (near) real-time communication will be essential for many use cases. And this communication will take place over networks that will typically not be under full control of the CPS. For this reason, a solution for robust and interoperable communication of critical data between CPS in a global network is essential. In this paper a general framework for dynamic and interoperable content adaptation based on underlying network conditions is presented. First validation results show that the framework can be used to dynamically scale the quality and amount of transmitted data and thus maintain timely delivery of critical data.**

***Keywords–Dynamic Content Adaptation; Adaptive Streaming; Congestion Control; Real-time Communication; Cyber-physical Systems*.**

I. INTRODUCTION

A Cyber-Physical system (CPS) is a system that integrates computation with physical processes. Transcending traditional, standalone, embedded systems, a CPS is a network of interacting entities [1]. CPS is strongly related to another concept called the Internet of Things (IoT). IoT is a global infrastructure of networked everyday objects ("things"), connected through interoperable communication technologies [2].

The ongoing TriCePS project investigates barriers and possible solutions concerning such interoperable communication technologies. With a myriad of network-connected entities, there will also be an immense number of different communication connections with varying qualities of connectivity. Caused by the stochastic behaviour of the underlying wired or wireless (usually IP) networks and applications that compete for bandwidth, the experienced Quality of Service (QoS) can be volatile. The size of the experienced variations depends on the ratio between the application requirements and the granted network resources. Managing this interdependence successfully without relying on exclusive and over-provisioned communication infrastructures is a main challenge in comparison to existing systems. In particular when time-critical data has to be transmitted strategies that maximize the robustness of throughput for the most relevant pieces of information are needed.

A. Adaptation strategies

Generally, there is a variety of possible adaptation mechanisms for dealing with network congestion.

1) Network adaptation: One possibility would be to adapt the network environment, for example by using some form of prioritization based on Weighted Fair Queueing (WFQ) [3]. However, this typically requires support from and proper configuration of the underlying network and is usually not well supported over networks that are not under own control such as the public Internet.

2) Time adaptation: Self-limiting the data rate which increases the duration a data transfer takes may also help to ease overall network congestion. This approach can be useful for communication that is not time-sensitive. A background download of a software update might be a good example where mainly data integrity and not timing is important.

For this adaptation mechanism we have successfully experimented with the use of different congestion control algorithms for data flows with different priorities. The approach uses an aggressive congestion control algorithm for high-priority data flows (e.g., CUBIC [4]) and a conservative congestion control algorithm for low-priority ones (e.g., TCP Vegas [5] or LEDBAT [6]). Such an approach can be helpful if there is no control over the underlying network and some data has loose timing constraints.

3) Data adaptation: The focus of this paper will be on the third mechanism which dynamically adapts the content to be send to the underlying network situation. For demonstration purposes we have chosen the use case of transmitting a live video from a surveillance camera. As network conditions become worse, the camera can switch from high-definition video to low-quality video (as in HTTP adaptive streaming), then gradually reduces frame rates down to 1 frame per second, applies grayscale filters and in the most extreme cases only transmits textual representations of detected objects and their movements or just signals if there has been any movement in the observed area or not. The surveillance camera use case has mainly been chosen for its visual attractiveness and easy understandability. The same framework developed in this project could be similarly applied to other use cases in which timeliness of data is essential and the normal amount of data transmitted can be temporarily significantly reduced (for example by only sending alarms instead of all sensor values).

B. Network model

TriCePS assumes a very general network model. Two communicating nodes A and B interchange information via a communication channel. This communication channel can span multiple network devices. Every route can be thought of as having a variety of properties including current available bandwidth, round-trip time and number of hops. Some of these properties might be mostly static (like the number of hops between two devices in a wired, local network), whilst others may change continuously (like the available bandwidth between two devices when other network flows are competing for resources).

Additionally, some further general assumptions are:

- There is no control over the network devices, features or configurations between two nodes.
- The route between two nodes can be used by potentially many other devices, too.
- There is no general means to control or modify network stack behaviour.
- It is possible to influence user space application behavior on nodes.

II. Related Work

A summary of various approaches for application layer congestion control is presented in [7]. The discussed approaches include message bundling (combining messages to the same recipient into a single message), the use of a message dispatcher which sends/receives messages on behalf of other systems, conditional messaging where (meta-)data or context information in messages is used to reduce the overall amount of data or messages exchanged, the use of persistent connections so that connections are not torn down after usage but instead kept open for reuse, piggybacking of data unrelated to the main exchange, self-throttling where nodes adapt the frequency of their messages based on their importance, data compression, and finally delta encoding where only differences to previous data is exchanged. The authors also give real-world usage examples for each of these approaches and analyse the approaches' effects on protocol efficiency, message sequencing, latency, performance and code complexity.

In general, with application level adaptation, the application layer is typically informed about current network metrics and adapts accordingly. This is not a novel approach; adaptive codecs have been around for quite a while. The Adaptive Multi-Rate (AMR) audio codec, which is widely used in GSM and UMTS, is a prominent example. With AMR, different coding schemes can be used depending on link quality measurements performed on the receiver side [8]. If conditions are good, AMR strives for speech quality (high speech bandwidth, low error protection). If conditions are bad, AMR strives for robustness (compromising on speech quality but boosting error correction and limiting bandwidth needs).

Another prominent example for existing application-level optimization used by many video stream providers is HTTP Adaptive Streaming (HAS). HAS estimates the current available bandwidth and adjusts the quality of the video stream accordingly. Segments of content (chunks that comprise short periods of video material) are encoded with different quality levels and sender/receiver choose segments according to current bandwidth estimates. The Motion Picture Expert Group (MPEG) proposed a standard called Dynamic Adaptive Streaming over HTTP (DASH) [9].

Minerva [10] is a solution for achieving video Quality-of-Experience (QoE) fairness based on TCP congestion control-like algorithms. In case of network congestion, Minerva video clients' bit rates converge towards QoE fairness while also ensuring fairness to other TCP flows on average. Compared to the use of other congestion control algorithms such as CUBIC or BBR (which do not optimize for QoE), a QoE increase of up to 32% could be achieved.

NADA (Network-Assisted Dynamic Adaptation) [11] also suggests adaptive real-time media applications that adapt their video target rate and thus their sending rate based on both Explicit Congestion Notifications (ECN) from network devices and implicit congestion signals (delay, packet loss).

In difference to the solutions above, TriCePS follows a more advanced approach by allowing the type of content being transported to change. A general framework and example implementation is presented that not only allows to adapt the sending rate but actually switches between very different types of application data to be transmitted ranging from video to high quality images to extracted features in form of text files.

III. Architecture

In this section, the architecture of the solution is presented. From a granularity point of view, all adaptation mechanisms are performed at flow-level (as opposed, e.g., to application-level or device-level). This is a comparatively non-invasive approach as each data flow can be treated separately and gives CPS users the means to prioritize individual data flows as necessary. Each application will also create and control its communication sockets (as opposed to, e.g., let an intermediary create and control all sockets and passing the data back and forth to the applications).

1) Components: Figure 1 shows two nodes, A and B, that represent communicating applications [12]. Both nodes A and B use the TriCePS software library. The library consists of several components.

The network monitoring module manages a modular set of measurement methods that continuously monitor the network flows between A and B. It then supplies a set of network metrics for both library-internal use and for use by the application business logic. The main network information required by TriCePS is whether there is congestion on the used network paths and the two main network-related metrics that are of importance for the purposes of TriCePS are latency and bandwidth. Changes in one or both of these two main metrics will serve as potential triggers for adaptations in the communication behavior of TriCePS-enabled CPS.

The pipeline, with its handlers, is an idea that we borrowed from Facebook's Wangle [13] project, which itself adapted this from Netty [14]. With pipelines, "the basic idea is to conceptualize a networked application as a series of handlers that sit in a pipeline between a socket and the application logic". Each handler has a specific duty. For example encryption, framing, compression, encoding, conversion, etc. The aim of this level of modularity is to keep the complexity associated with handling the different communication mechanisms of a wide range of entities manageable. The pipeline module glues together the individual handlers. Data flows from the business logic of node A through all handlers in the pipeline through the network to the opposite node, where all handlers are traversed in reverse order. It is noteworthy that Figure 1 shows one specific, simplified example of a pipeline setup. The two pipelines in the example hold the same handlers (h1, h2, h3,

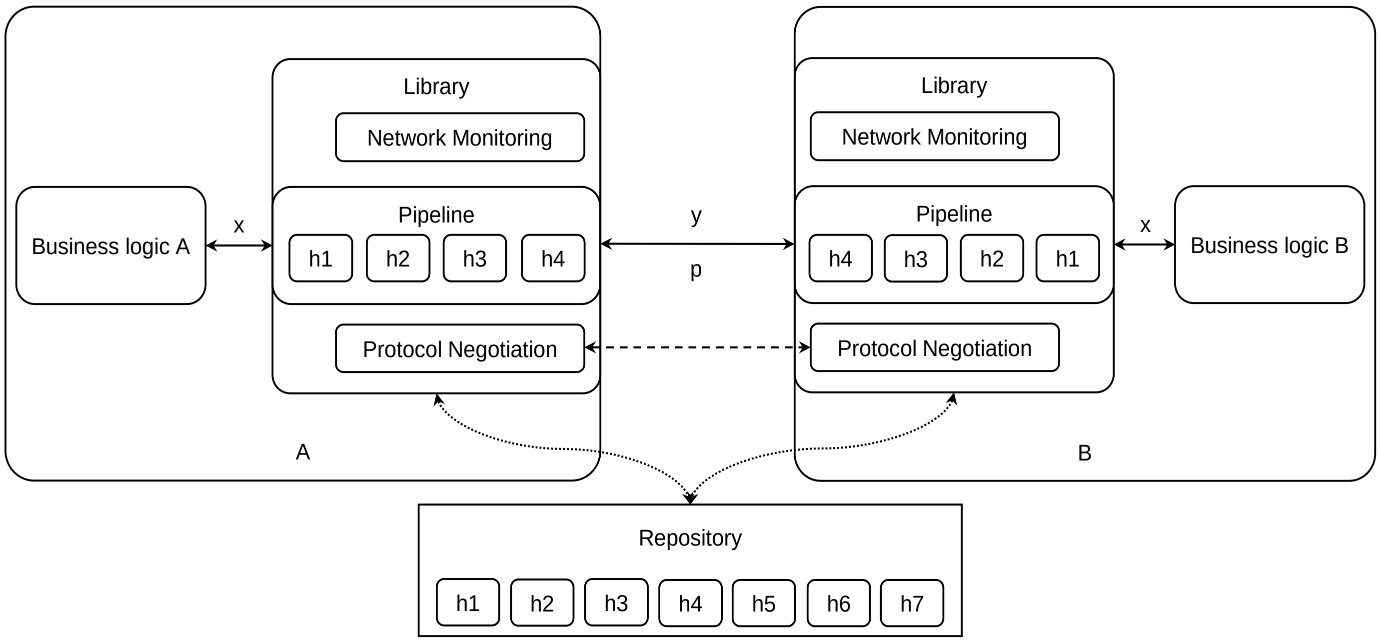


Figure 1. Overview of the reference implementation and its use by two nodes

h4) for each node. That is not a requirement, the two pipelines could each hold a different set and/or number of handlers if necessary.

Finally, the protocol negotiation module makes sure that the applications use pipeline setups that are compatible with each other. The mechanism has to be lightweight and fast to enable recurring, short-term (think seconds) pipeline reconfigurations due to network congestion. A repository acts as a library for new and updated handlers. The protocol negotiation module also takes care of the process of retrieving new or updated handlers from the repository.

It should be noted that the TriCePS concept assumes that all involved nodes use the TriCePS library and that the main constraint is limited network resources. The involved CPS need to have sufficient additional other resources (e.g, computing or memory) to execute the necessary TriCePS functions. As a workaround, the use gateway devices could also be considered.

2) In-band vs out-of-band communication of meta-data: Note that both the negotiation and the actual coordinated switching of handlers between TriCePS nodes require the exchange of metadata. This can happen in-band or out-of-band. Using an in-bad channel implies that the application stream has to be modified (e.g., negotiation packets inserted) and/or potentially even be (temporarily) paused for the negotiation/switching to occur. This can be comparatively intrusive from an application point of view. The advantage of using an in-band channel is that no second communication channel is needed as there might be scenarios in which the creation of such an additional channel may not be possible. An out-of-band negotiation/switching process can occur concurrently to the transmission of main data, avoiding any interruptions. Both approaches are supported by TriCePS, however, the latter is the preferred choice to allow for a more seamless operation.

3) Seamless Switching: To avoid any interruptions when performing a switch, the sender announces in advance after how many additional packets it will switch to a new pipeline. This switch is only performed upon confirmation from the receiver. Figure 2 shows an example of a switch after 4 additional packets. In this figure at first data is sent using pipeline “A” (black arrows). To start switching to pipeline “B” (blue arrows), the sender sends its current “send_counter” and the “switch_delay” (4 in this example) through the out-of-band channel. The sender keeps using pipeline “A” for 4 more times (black half-arrows) and only then it switches pipeline “B”. The receiver, upon receiving “send_counter” and “switch_delay”, will send a confirmation and then wait for its “receive_counter” to match the “send_counter”. Thus the receiver will process 4 more packages using the old pipeline and then switch to the new pipeline. This way a seamless transition can be achieved without any interruptions or modifications of the main data flow.

4) Stability and Fairness: In the case of multiple competing network flows, questions of stability (the desire to not switch between the modes too often) and fairness (on average every node gets about the same amount of resources) become relevant.

Two thresholds are defined for each send mode/pipeline: If the bandwidth (or frame rate) falls below the low threshold for some time, the mode changes to one that requires less network resources. If the bandwidth (or frame rate) rises above the upper threshold, the mode changes to one that requires more network resources.

To avoid that a node keeps switching between different modes too quickly and to avoid that multiple nodes switch to a higher sending rate and immediately back to a lower sending rate in a synchronized way, both a hysteresis and a probabilistic element are added to the switching component.

Upon hitting one of the two thresholds, a timer is started which serves as input to a sigmoid function which defines the probability p that a switch to a different mode is attempted.

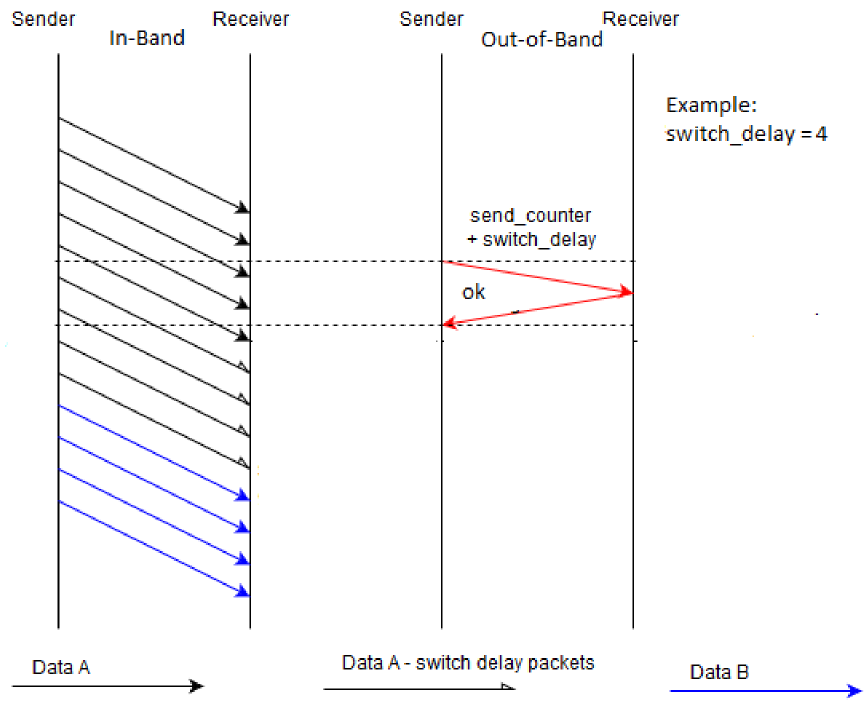


Figure 2. Implemented seamless switching mechanism

The timer is reset to 0 if the threshold criteria is not met anymore. The result is the longer a node remains below/above the lower/upper threshold, the higher the probability it attempts to switch to a different mode.

The specific thresholds and parameters for reducing or increasing the send rate can be chosen differently. In the example application, a down switch was triggered faster to avoid a reduced frame rate and an up switch was triggered more slowly as probing for a higher bandwidth too frequently can reduce overall system efficiency. This approach is quite generic and can be applied to any kind of transmission, however, the values for thresholds and the parameters may need to be adjusted for specific use cases.

IV. Implementation

A reference implementation of the TriCePS architecture was realized. We use a surveillance camera as example scenario and assume that the camera needs to transmit live digital footage over a communication network. The bandwidth requirement can be significant while the amount of available bandwidth fluctuates. Tackling this problem through (potentially extensive) buffering is not sufficient as the footage needs to be transmitted and processed in real-time.

To ensure liveliness of data, it can be better to compromise on video quality than to look at a stuttering and delayed high quality video. The amount of transmitted data is adopted according to the following hierarchy (from good to bad network quality):

- HD video, normal compression, normal frame rate
- Individual images at low frame rate, reduced quality
- Individual image features (using image feature extraction and sent through text descriptions)

This way relevant live information can be provided to the receiver even in case the available bandwidth is reduced by up to a factor of 100. The bandwidth requirements range from several Mbit/s for HD video to only several Kbit/s for textual descriptions of image features (see Table I).

Fig. 3 shows this mechanism as a state machine [12]. When network conditions are good, high quality video (as provided by [15]) will be emitted. When congestion is detected, lower quality still images will be emitted and when conditions get even worse, a text representation of the moving parts of the image will be emitted (the background image is only transmitted once and remains static). With improving conditions, the same process will happen in reverse order.

Note that the video data can be considered most valuable as this is the original data and the data reduction steps (conversion to single images, feature extraction) could still be performed at the receiving node if desired. The other way round, information that has already been discarded in the text representation cannot be reconstructed anymore (e.g., the arm and hand movements of the persons which have not been extracted).

The state where video is emitted uses a pipeline with two handlers (transcoding video to target bandwidth, framing), the state where still images are emitted uses a pipeline with three handlers (image extraction, image compression, framing) and the state where text is emitted uses a pipeline with four handlers (feature extraction, filtering of irrelevant features, encoding, framing).

1) Custom metrics: The network monitoring module manages a modular set of measurement methods that continuously monitor the network conditions between two TriCePS nodes (so, in this example case between the camera and the remote monitor). While the TriCePS software library provides the

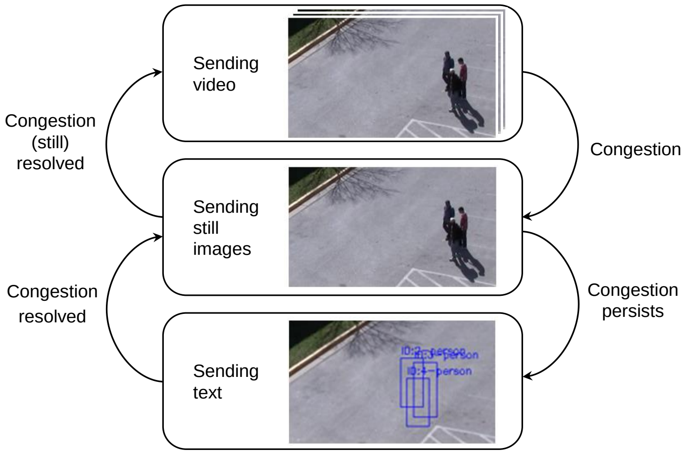


Figure 3. State machine for an adaptive camera

TABLE I. SENDING STATES AND THEIR PROPERTIES

State	*Updates/s*	*Bandwidth (kbit/s)*	*Bandwidth (%)*
Video	25	≈ 2000	100
Images	1	≈ 200	≈ 10
Text	25	≈ 20	≈ 1

software interfaces to handle network measurements in a generalized way and already includes some standard measurements (e.g., round-trip delay or current sending rate), more application specific measurement methods can also be integrated. Since the camera use case is focused on image frames, we implemented a simple measurement method that measures the rate at which individual images can be emitted (frames per second).

This demonstrator serves as a visually attractive example application for the developed framework as it makes use of most TriCePS adaptations mechanisms and components at once. It will be subsequently adapted by industrial partner COPA-DATA for industrial use cases where for example near real-time availability of critical SCADA data is crucial.

V. TESTING AND VALIDATION

Figure 4 shows a basic test setup. A camera sends live images towards the display, and a traffic generator is used to create various degrees of network use/congestion.

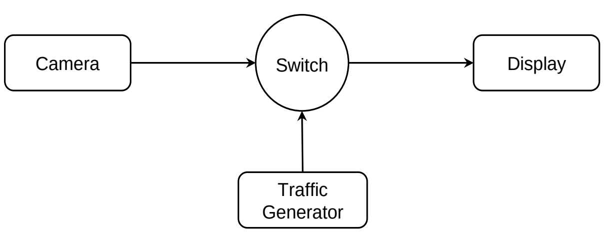


Figure 4. Schematic layout of the test setup

Figure 5 shows the measurement result of a test. Network congestion is generated roughly between seconds 30 to 60 and between seconds 100 to 160. The figure shows six curves, three of them depicting the frame rate in frames per second (fps_source, fps_control and fps_test) and the other three depicting the total frame count (frame_source, frame_control and frame_test). The curves with the "source" suffix represent the count/frame rate of the camera. The curves with the "test" suffix show the count/frame rate of the receiver in a TriCePS system. And finally, the curves with the "control" suffix show what would happen without TriCePS. It can be seen that by prioritizing the timeliness of data (at the cost of quality), the frames/updates per second could be almost hold stable even in times of network congestion (fps_test recovers to stay close to fps_source after congestion sets in while fps_control remains low). Consequently, frame_test also follows frame_source much more closely than frame_control, showing a significantly smaller lag between between the camera output and the received data when using the TriCePS system. Overall, we observed significant improvements concerning number of successfully received frames/updates, delay of frames/updates and round-trip time (as a measure of network conditions) as summarized in Table II.

TABLE II. MEASUREMENTS WITHOUT AND WITH TRICEPS. MEAN VALUES AND STANDARD DEVIATIONS AVERAGED OVER FIVE RUNS

Value	*Without TriCePS*	*Using TriCePS*
Sent frames	2566 (72)	3995 (34)
Average delay	871 ms (146 ms)	108 ms (12 ms)
Average RTT	53 ms (4.5 ms)	33 ms (2.4 ms)

For larger-scale testing with up to 20 nodes, simulations with real TriCePS code in loop with a simple network simulation have been performed. When running multiple TriCePS nodes, on average flows get roughly an equal share of bandwidth (see Table III).

TABLE III. FAIRNESS MEASUREMENTS

# of flows	*Fair share of total BW*	*Min. share*	*Max. share*
2	50%	47.3%	52.6%
4	25%	22.3%	25%
8	12.5%	11.2%	13.4%
20	5%	4.16%	5.43%

VI. CONCLUSION

We have presented a framework for dynamic content adaptation for the timely delivery of critical data under network congestion. A pipeline-based data processing architecture with support for live negotiation and switching of pipelines forms the core of the solution. Using this approach a demo application that can scale the amount of sent data by orders of magnitude has been implemented. We have shown that average delay of critical data can be reduced by almost up to 90% (at the cost of data quality/amount). When multiple network nodes using the developed solution compete for resources, on average fairness of bandwidth allocation between the nodes is achieved.

ACKNOWLEDGMENT

This project is partially funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) under the program "ICT of the Future" (https://iktderzukunft.at/en/).

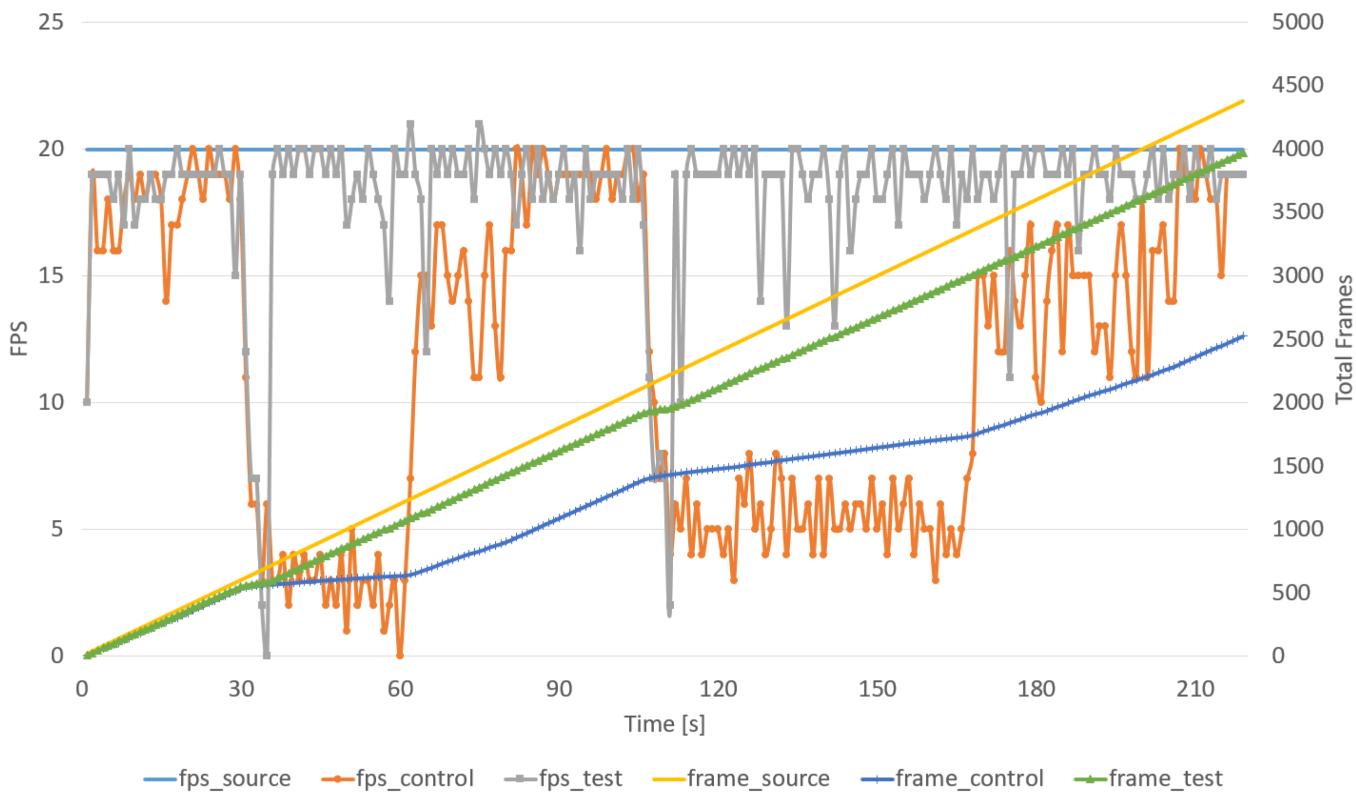


Figure 5. Frame rate and total frame count with and without TriCePS

REFERENCES

[1] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the Internet of Things (IoT)," IEEE Internet Initiative, vol. 1, 2015, pp. 1–86.

[2] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of Things," International Journal of Communication Systems, vol. 25, no. 9, 2012, pp. 1101–1102.

[3] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," IEEE/ACM Transactions on Networking, vol. 1, no. 3, 1993.

[4] S. Ha, I. Rhee, and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," SIGOPS Oper. Syst. Rev., vol. 42, no. 5, 2008.

[5] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP vegas: New techniques for congestion detection and avoidance," SIGCOMM Comput. Commun. Rev., vol. 24, no. 4, 1994.

[6] S. Shalunov, G. Hazel, J. Iyengar, and M. Kühlewind, "Low Extra Delay Background Transport (LEDBAT)," RFC 6817, 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6817.txt

[7] P. Adamczyk, M. Hafiz, F. Balaguer, and C. L. Robinson, "Network congestion control at the application layer," in Proceedings of the 14th Conference on Pattern Languages of Programs, 2007, pp. 1–15.

[8] E. Ekudden, R. Hagen, I. Johansson, and J. Svedberg, "The adaptive multi-rate speech coder," in 1999 IEEE Workshop on Speech Coding Proceedings. Model, Coders, and Error Criteria (Cat. No. 99EX351). IEEE, 1999, pp. 117–119.

[9] I. Sodagar, "The MPEG-Dash standard for multimedia streaming over the internet," IEEE multimedia, vol. 18, no. 4, 2011, pp. 62–67.

[10] V. Nathan, V. Sivaraman, R. Addanki, M. Khani, P. Goyal, and M. Alizadeh, "End-to-end transport for video QoE fairness," in Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM '19), 2019.

[11] X. Zhu, R. Pan, M. Ramalho, S. de la Cruz, C. Ganzhorn, P. Jones, and S. D'Aronco, "NADA: A unified congestion control scheme for real-time media," Draft IETF, Mar, 2013.

[12] S. Linecker, J. L. Du, A. V. Silva, and R. Mayr, "The pipeline concept as key ingredient for modular, adaptive communication for cyber-physical systems," in Proceedings of the 17th International Conference on Informatics in Control, Automation and Robotics (ICINCO), 2020.

[13] Wangle Tutorial, (accessed January 20, 2020). [Online]. Available: https://github.com/facebook/wangle/blob/master/tutorial.md

[14] Netty: Home, (accessed January 20, 2020). [Online]. Available: https://netty.io/

[15] S. O. et al., "A large-scale benchmark dataset for event recognition in surveillance video," in Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR), 2011.

Semiconductor Defect Classification

Terence Sweeney, Sonya Coleman, Dermot Kerr
School of Computing, Engineering and Intelligent Systems,
Ulster University
Londonderry, Northern Ireland
Email: Sweeney-t4@ulster.ac.uk, sa.coleman@ulster.ac.uk, d.kerr@ulster.ac.uk

***Abstract*—Automated inspection has become a vital part of quality control during semiconductor wafer production. Current processes are focussed on finding defects via variation from a 'golden' image using pixel to pixel comparisons or utilization of opaque neural network-based approaches. We present a novel approach, which uses the Bag of Visual Words technique to determine local features that correspond to specific defects within a wafer image, known as a custom vocabulary, as a way to begin creation of a more transparent system for automated defect detection and classification. We demonstrate that the custom vocabularies, combined with machine learning algorithms, result in high performance accuracies with efficient computational run-times.**

***Keywords*— Defect Detection; Defect Classification; Bag of Visual Words; Local Features; Semiconductor wafers; Image Processing.**

I. INTRODUCTION

Semiconductor wafers are a component used in products such as processors and hard drive media. Inspection is vital during the manufacturing process in order to detect defects and ensure quality control. Several methods have been proposed for defect detection on semiconductor wafers, however the majority of techniques focus on defect detection across the wafer as a whole. When defects are detected they are marked on the wafer bin map in order to identify the total number found. This is a useful approach when looking for systematic defects across a product line and removing a defective product earlier in the production line. However, it is sometimes desirable to detect not only the location of a defect but also the type of defect as some of the product may still be commercially viable. The goal of this research is to use images of a single chip on the wafer, known as die images, to detect and classify defects. An example of a wafer bin map and a die image is presented in Figure 1.

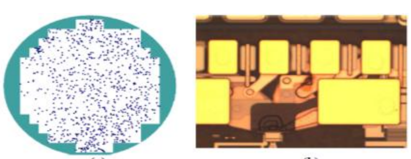


Figure 1. (a) Wafer bin map with detected defects coloured blue and (b) Single die image

Production of semiconductor wafers involves multiple stages and many different components are used during this process. Due to the varying size and criticality of these components, many different inspection techniques are used throughout manufacturing to ensure quality control. Inspection techniques include using electrical input and microwave testing along with optical cameras that can inspect to pico-meter level. The difference in these types of inspection systems has resulted in many interpretations of how to best detect and classify defects [10][21]. One widely used approach is to observe the overall frequency and location of defects using the wafer map in order to detect systematic or widespread damage over the complete wafer, such as a scratch or tear, as shown in Figure 2. Whilst this solution [9][11]has been proven to be useful for finding systematic or clustered defects across the whole wafer, it does not consider the type of defect, and consequently whether the product is still viable.

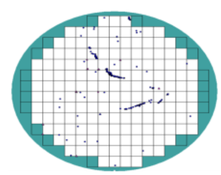


Figure 2. Example of a Scratch defect using the wafer bin map

When considering automated visual inspection of semiconductor wafers, die images are used and examples of defects upon these die images are given in Figure 3. Most previous work is based on the use of global features with Tobin's content-based image retrieval golden image comparison method [25] being the most popular. This inspection representation is commonly found in most Automated Defect Classification (ADC) machines [1][21]. However, other methods have been used to detect specific types of defects across the industry. Chou [6] uses the Hough transformation to detect scratches or gouges on a wafer surface while Park [23] detailed an approach using the Histogram of Gradient (HOG) operators to great effect. However, there has been little work on the use of local image features for automated inspection using techniques, such as Sobel [14], Scale Invariant Feature Transform (SIFT) [8], Oriented FAST and Rotated BREIF (ORB) [13], and SURF [2]. In [16], we proposed the use of SIFT and SURF local image features for wafer defect detection and concluded that whilst both techniques could identify wafer defects, the use of SURF resulted in improved detection accuracy.

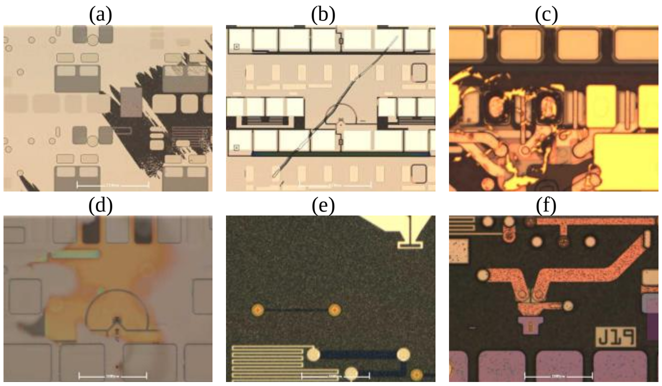


Figure 3. Examples of defects present on semiconductor die images: (a) Rip, (b) Scratch, (c) Warp, (d) Delamination, (e) Incomplete liftoff, (f) Corrosion

In addition to detecting defects, defect classification is also necessary. In some cases, products that have been identified with non-critical defects, which would otherwise be removed due to the presence of a defect, can continue on the production line. Additionally, identifying the type of defect and the stage at which it occurs in the production process can help in improving overall quality, yield and production processes. The majority of defect classification approaches to date focus on the use of neural networks and deep learning. For example, Reza [12] used an artificial neural network with a back-propagation algorithm to observe contamination defects on wafers, a method also applied by Chou [6]. The work in [11][19][20][21] uses Convolutional neural networks for classification. While these deep learning methods return good results, the black box nature of neural networks can be a problem in industry, such as semiconductor manufacturers, since the designs are frequently updated and changed and although the neural network could be trained to work well with current designs, new designs could cause system failure as we are currently seeing in domains such as self-driving cars [26] and image recognition [27]. Thus, we have developed a novel approach based on transparent local features to create a more understandable system.

A well-known feature extraction technique is the Bag of Visual Words (BoVW) method, which extends the Bag of Words (BoW) method from the text retrieval domain to the visual classification domain and can be used as an alternative to global image features. When using the BoW technique on a text document, a normalized histogram of word counts is computed as well as a sparse term vector where each bin corresponds to a term in the vocabulary. The BoVW technique [29] enables the generalisation of local image feature descriptors in a similar manner and has been used for image classification [3][15][17]. Improving further on BoVW, a custom vocabulary [20] or codebook is a concept in which specific subsets of visual words are selected, which represent the most important features of the images, rather than using the complete vocabulary created from a set of training images. One example of this approach is the dual vocabulary approach [17] where two vocabularies are trained on different training set classes before being run on its testing data in order to observe, which returns the highest accuracy for each testing class and therefore which features are most important for detection and classification of these classes. Custom Vocabularies take this a step further by observing which visual words contain the most important information for a given task and utilize only these visual words in order to increase overall accuracy and also reduce overall computation time.

It is also possible to combine BoVW with machine learning classifiers. For example, Hentschel [4] evaluated several different classification methods such as AdaBoost [5], Support Vector Machines (SVM) and decision trees on an image classification problem utilizing BoVW and found several methods that achieve high accuracy when combined with local feature methods. Two popular image classification approaches that are widely used across many different fields of automated visual inspection are multi-class SVM [7][24] and Random forest [22].

Building on previous work [16], this paper proposes a novel approach to defect detection and classification in semiconductor wafers. We identify specific visual words that correspond to a defect descriptor, *a custom vocabulary*, and use these for classifying a defect within an image as close to real time speed as possible, whilst still retaining high levels of accuracy. The remainder of this paper is organized as follows: Section 2 introduces the current industry inspection process used by our industry partner and its problems. Section 3 covers the proposed *custom vocabulary* and Section 4 discusses the performance evaluation of the approach. Finally, Section 5 details the conclusion and further work.

II. CURRENT INDUSTRY INSPECTION PROCESS

There are around 600 stages in the production of a single semiconductor wafer. In order for the wafer to fully function it needs to be kept free of defects which can be caused in many ways, including particle damage, atmospheric changes as well as human- and machine-error. Thus, a typical semiconductor production line will have many in-line inspection tools at various manufacturing stages in order to ensure quality control. Due to the size of critical parts on the wafer, some as small as 7nm, specialised inspection equipment must be used. The inspection process can be conducted in various ways, for example using electrical fault detection and x-rays, however the most time-affordable systems are visual inspection systems.

Current industry practice for defect detection and classification is a global image matching approach where a direct pixel-to-pixel comparison is performed using a database of control images which are directly compared with the current product passing through the inspection system. This is commonly known in the industry as a 'golden image' approach. In order to prevent false detection of defects, a defect reduction factor is used where pixel intensities within a 3x3 pixel neighborhood are compared before any area is regarded as a defect.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-815-0

The Rudolph NSX105 [1] is a commonly used industry standard inspection device which uses the golden image approach. The NSX105 inspection system uses its initial stage camera to strobe over the wafer comparing captured images with the corresponding database of golden images. The golden images in the database are initially manually pre-programmed. Hence, when a product is developed or updated, a new set of golden images must be created. If a defect is detected, its coordinates are saved into a reference file and then additional high-resolution images of the defect on the die are captured using a second inspection camera for subsequent manual inspection. The number of defects for which high resolution images are captured is capped at a level according to parameters set manually, typically 80 images per wafer. A critical problem with the NSX105's inspection detection is that while it can determine a problem at a specific location, it cannot determine the type of defect that has been found on the die. Hence the severity of the defect is unknown, and this may result in more serious defect types, such as corrosion damage on critical parts, going unnoticed until later in production.

We seek to improve on this by developing an automated inspection system, which uses the existing inspection equipment output, and is focussed specifically on classifying high resolution defect images from the die rather than the defect identification stage which creates the wafer bin map.

III. CUSTOM VOCABULARY

In the proposed methodology, the SURF interest point detector is used to obtain key-points k_n and corresponding SURF descriptors d_n where $i = 1 \dots n$ such that a keypoint is represented as:

$$k_i = (x_i, y_i, d_i) \quad (1)$$

where x and y are the coordinates of a point in an image. The SURF keypoint descriptors are of 64 dimensions. An image feature set S can be represented by the set of local keypoint descriptors such that

$$S_I = \{k_1, k_2, \dots, k_n\} \quad (2)$$

where $I = 1 \dots m$ and m is the number of images in the image set. The BoVW algorithm B is considered to quantize the descriptor $d \in R^I$

$$B: R^I \to [1, K] d \to B(d). \quad (3)$$

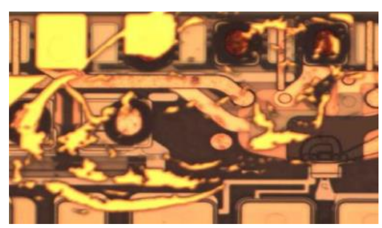
The B assigns descriptor $d \in R^I$ to the appropriate cluster K, where each cluster represents a visual word and the set of visual words is the initial defect vocabulary.

We can further refine the initial vocabulary to form a *custom vocabulary* through manual inspection of the defect images where only visual words that represent wafer defect features are retained and that is the approach used here.

IV. PERFORMANCE EVALUATION

There are various defects that can occur in semi-conductor wafers, such as splatter, warp, scratch, rip, delamination and corrosion. To evaluate the proposed approach for defect detection, we focus on the warp defect. The warp defect occurs for various reasons including temperature changes, rise in atmospheric pressure, or human and machine error. Its main feature is that parts of the golden resist (or paint), also called the gold pad, are removed or warped in some way. Examples of warp die images are presented in Figure 4 where Figure 4(a) illustrates the gold resist in various stages of damage from the warp defect and Figure 4(b) illustrates complete removal of the resist. All images are captured by the Rudolph NSX105 from one layer of one product, and all images are 648x494 pixels. All experiments are run using Python OpenCV and Sklearn on an Intel Xeon CPU E5-120 0@ 3.60 GHZ with 16 GB of RAM.

In the initial experiment, we evaluate the proposed approach using a vocabulary of 1000 visual words and various well-known machine learning algorithms including AdaBoost, Random Forest, Support Vector Machines (SVM) with a range of kernel functions (Linear, Polynomial and Radial). We use sets of warp images (Figure 4) and control images (Figure 5). Both the warp and control classes contain 100 images each (200 in total), split 80/20 for training and testing. The machine learning algorithms have been optimised via a grid search and a summary of the results is displayed in Table I. Using 1000 visual words, the results vary across the different machine learning approaches with the SVM using a Radial Basis Function (RBF) and C=10 providing the highest accuracy.



(a) Gold resist in various stages of damage

(b) Complete removal of the resist

Figure 4. Examples of the warp defect

 ISBN: 978-1-61208-815-0

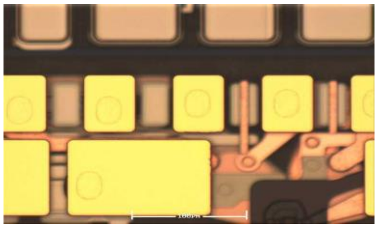


Figure 5. Control image

While the accuracy results are promising, the system takes significant time to process 1000 visual words, with the training alone taking 7 minutes and 37 seconds. Additionally, although we can determine from the BoVW histogram which visual words occur most often for each class, it is not possible to determine what initial local features make up each visual word. From the 1000 visual words, it was possible to isolate 106 visual words that corresponded solely to the warp defect, with no key points detected on the image background. The experiments were conducted again using this refined set of visual words, a *custom vocabulary*, and the results are presented in the last column of Table I. In this scenario, several machine learning approaches, combined with the *custom vocabulary*, provide an accuracy of 97%, hence the use of a custom vocabulary is more consistent and less dependent on the machine learning algorithm it is combined with, and the training time using this vocabulary is also much closer to a real time system, taking only 28 seconds, approximately 15x faster than using 1000 words.

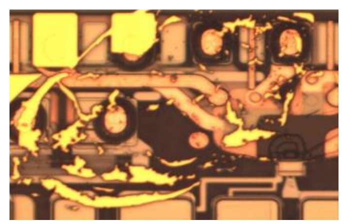
TABLE I– EXPERIMENTAL RESULT

	Accuracy **1000 visual words**	**Accuracy** **106 visual words (custom vocabulary)**
AdaBoost	**95%**	87%
Random Forest	**79%**	75%
SVM - Linear C=1	51%	**51%**
SVM – Linear C=10	53%	**75%**
SVM – Linear C=100	90%	**97%**
SVM – Poly C=1	**51%**	50%
SVM – Poly C=10	51%	**80%**
SVM – Poly C=100	56%	**97%**
SVM – RBF C=1	56%	**65%**
SVM – RBF C=10	**100%**	97%
SVM – RBF C=100	97%	**97%**

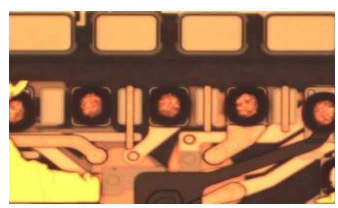
The ability to identify a defect in a die image is important in automated inspection, and it is possible to further define the warp defect into 3 sub-classes. This has important consequences as some sub-classes of warp defect have more impact on the wafer production than others. The first sub-class denoted as Warp 1 contains erratic shapes and sharp-edged resist pieces that appear across the wafer image. The second sub-class, denoted Warp 2, focusses on the circles that appear as the resist is wiped away from the wafer. The third sub-class, denoted as Warp 3, has circular blobs or scratches through the resist. An example of each warp sub-class is illustrated in Figure 6.

Using the custom vocabulary of 106 visual words, we create a new custom vocabulary for each sub-class where Warp 1 requires 63 visual words, Warp 2 requires 48 visual words and Warp 3 requires 29 visual words. The custom vocabulary for Warp 1 contains the most unique visual words whereas the custom vocabulary for Warp 2 has overlap with both Warp 1 and Warp 3. The experiments were conducted again using the custom vocabularies. As the results in Table I demonstrated that AdaBoost and Random Forest do not perform as well as SVM, we present results only for SVM in Table II.

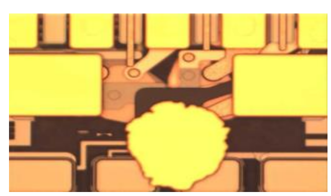
As shown in Table II, the linear SVM performs similar to the results presented in Table I, and hence it remains the worst performing SVM. The polynomial kernel SVM has increased accuracy compared with the linear SVM, however the RBF kernel SVM retains the highest accuracy for all SVMs across the three sub-classes. The key significance of the results in Table II is that the classification accuracy is high with an improvement in computational efficiency due to the reduced feature set, *the custom vocabulary*.



(a) Example of Warp 1 image



(b) Example of Warp 2 image



(c) Example of Warp 3 image

Figure 6. Examples of warp sub-classes

TABLE II - SVM ACCURACY RESULTS

SVM			
	Warp 1	**Warp 2**	**Warp 3**
SVM Linear C-1	51%	51%	51%
SVM Linear C-10	92%	85%	68%
SVM Linear C-100	97%	97%	87%
SVM Poly C-1	70%	73%	78%
SVM Poly C-10	85%	95%	87%
SVM PolyC-100	**100%**	97%	70%
SVM RBF C-1	87%	90%	70%
SVM RBF C-10	97%	**100%**	92%
SVM RBF C-100	**100%**	97%	95%

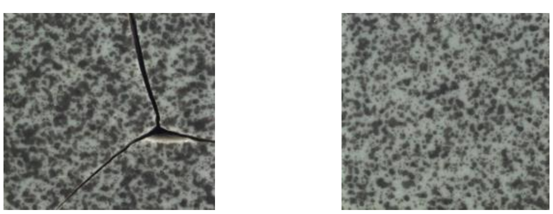
TABLE III - SPEED TEST RESULTS

Computational speed Test	Training Time	Prediction Time	Highest Classification Accuracy
1000 Words	7m 37s	13s	100%
106 words	28s	9s	97%
Subclass	**25s**	**6s**	**100%**

Another important consideration is the speed of this system, as it is required to operate with in-line inspection tools and should therefore be as close to real time as possible whilst still retaining a high degree of accuracy. Table III shows that the proposed approach, based on the custom vocabulary, achieves the fastest run-time compared with the use of a larger vocabulary, as well as high accuracy.

V. MVTEC EVALUATION

The results presentenced in the previous section demonstrate that the custom vocabulary that corresponds to a specific defect provides high classification accuracies. In order to further validate this system, we use the MVTEC anomaly detection dataset [28]. From the dataset, we selected the Tile Crack image set which contains 20 images, 10 for training and 10 for testing, along with a control class, again using 10 for training and 10 for testing. Examples of these images are given in Figure 7 .



(a) (b)

Figure 7. Examples of (a) Tile Crack Defect image and (b) Tile Control Image

In line with the previous experiment, as the SVM performed best, we use only an SVM with all 1000 visual words and the defect only visual words, for which 69 were detected for this dataset.

TABLE IV – TILE CRACK RESULTS

SVM		
	Full 1000 Words	**69 Defect only Words (Custom Vocabulary)**
SVM Linear C-1	72%	80%
SVM Linear C-10	72%	80%
SVM Linear C-100	72%	80%
SVM Poly C-1	72%	80%
SVM Poly C-10	72%	80%
SVM Poly C-100	72%	**97%**
SVM RBF C-1	72%	80%
SVM RBF C-10	72%	80%
SVM RBF C-100	82%	**97%**

As illustrated in Table IV, a maximum accuracy for this dataset, when using 1000 visual words was 82% using an SVM, with the RBF kernel and C=100. However, this is reproved significantly by using a custom vocabulary that corresponds to the defect only features present in the images. We can see an increase to 97% using both the polynomial and RBF kernels with C=100. This is excellent performance accuracy given the small dataset used and would be difficult to achieve using deep learning which requires a significant volume of data. This demonstrates the robustness of the proposed approach across industrial datasets.

VI. CONCLUSION AND FURTHER WORK

We have presented an approach to semi-conductor wafer defect classification by utilizing the bag of visual words method with a *custom vocabulary* formed from a reduced set of visual words. We have demonstrated that this novel approach achieves competitive accuracies when compared with the use of a larger set of visual words (1000) but is much more computationally efficient as demonstrated by the presented run-times.

As the proposed approach works well, both on our industrial dataset and the MVTEC anomaly dataset, future work will investigate the design of custom vocabularies for other defect types, namely splatter, scratch, rip, delamination, and corrosion. Additionally, we will explore the ability to accurately characterise and hence classify the warp defect images using only the custom vocabulary without additional machine learning. The motivation for this is that, within the production line, if there is a design change then a neural network focused automated inspection system will require retraining. However, if we can accurately classify defects without the use of deep learning and by using the custom vocabulary approach, this will enable the system to be readily adaptable to product changes and developments creating a more open and understandable system.

ACKNOWLEGMENT

This work was funded by a DfE CAST scholarship in collaboration with Seagate Technology. We would also like to thank Seagate Technology for providing the image dataset used in the research.

REFERENCES

[1] Tarasemi, "Rudolph August NSX 105 Automated Defect Inspection," 2019.[Online].Available: https://www.tarasemi.com/product/rudolph-august-nsx-105-automated-defect-inspection/. [Accessed 2 September 2020].

[2] H. Bay, "Surf: Speeded up robust features," in *European Conference on computer vision, pp. 404-417*, Berlin, 2006.

[3] G. Csurka, "Visual categorization with bags of keypoints," in *Workshop on Statistical learning in computer vision, pp. 1-2*, 2004.

[4] C. Hentschel and H. Sack, "Does one size really fit all? Evaluating classifiers in Bag-of-Visual-Words classification," in *14th International Conference on Knowledge Technologies and Data-driven Business.*, pp. 1-8, Graz, Austria, 2014.

[5] T. Hastie, S. Rosset, J. Zhu and H. Zou, "Multi-class AdaBoost" *Statistics and its Interface,* pp. 349-360, 2009.

[6] C. J. Huang, C. F. Wu and C. C. Wang, "Image processing techniques for wafer defect cluster identification," *IEEE Design & Test of Computers,* pp. 44-48, 2002.

[7] C. Li-Chang and T. Lee-Ing, "Wafer defect pattern recognition by multi-class support vector machines," *Expert Systems with Applications,* pp. 10158-10167, 2009.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision,* pp. 91-110, 2004.

[9] D. P. Mital and E. K. Teoh, "Computer based wafer inspection system," in *Proceeding of international conference on industrial electronics, control and instrumentation*, Kobe, 1991.

[10] N.G. Shankar and Z.W. Zhong, "Defect detection on semiconductor wafer surfaces," *Microelectronic Engineering,* vol. 77, no. 3, pp. 337-346, 2005.

[11] M. P.-l. Ooi, "Defect cluster recognition system for fabricated semiconductor wafers," *Engineering Applications of Artificial Intelligence,* pp. 1029-1043, 2013.

[12] A. Z. Reza, T. Hisashi, T. Y. S. Shinichi , and S. Kazuma, "Automated inspection of IC wafer contamination," *Pattern Recognition,* pp. 1307-1317, 2001.

[13] E. Rublee, V.Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *ICCV,* p. 8, 2011.

[14] I. Sobel and G. Feldman, "A 3x3 Isotropic Gradient Operator for Image Processing"," in *Stanford Artificial Intelligence Project*, Stanford, 1968.

[15] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Nice, 2003.

[16] T. Sweeney, S. Coleman, and D. Kerr, "A Machine Learning Approach to Wafer Defect Classification using Bag of Visual Words," in *Irish Machine Vision and Image Processing Conference. 2019*, Dublin, 2019.

[17] K. S. Sujatha, P. Keerthana, S. Suga Priya, and E. Kaavya "Fuzzy based multiple dictionary bag of words for image classification," *Procedia Engineering,* pp. 2196-2206, 2012.

[18] F.-C. Tien, "Reclaim wafer defect classification using large scale BPNS with Tensorflow," *National Academy of Managerial Staff of Culture and Arts Herald,* pp. 641-644, 2018.

[19] S. Varsha Devi et al., "Better object recognition using bag of visual word model with compact vocabulary," in *13th International Conference on Emerging Technologies (ICET),* Islamabad, 2017.

[20] S. Zeng, S. Dai, and P. Mu, "Wafer Defects Detecting and Classifying System Based on Machine Vision," in *8th International Conference on Electronic Measurement and Instruments*, Xi'an, 2007.

[21] P. Chou et al., "Automatic defect classification for semiconductor manufacturing," *Machine Vision and Applications,* vol. 9, no. 4, pp. 201-214, 1997.

[22] L. Puggini, J. Doyle, and S. McLoone, "Fault Detection using Random Forest Similarity Distance," in *9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes,* Paris, 2015.

[23] J.-K. Park et al., "Machine Learning-Based Imaging System for Surface Defect Inspection," *International Journal of Precision Engineering and Manufacturing Green Technology,* vol. 3, no. 3, pp. 303-310, 2016.

[24] C. Pu et al., "Recognition and classification of coating film defects on automobile body based on image processing," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*, Shanghai, 2017.

[25] K. Tobin, T. Karnowski, and F. Lakhani, "Integrated applications of inspection data in the semiconductor manufacturing environment," *Metrology-based Control for Micro-Manufacturing,* pp. 31-41, 2001.

[26] K. Eykholt, et al. "Robust physical-world attacks on deep learning visual classification." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[27] N. Anh, J. Yosinski, and J. Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[28] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. MVTec AD - A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection; in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[29] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* San Diego, 2005.