Area & Perimeter Surveillance in SAFEST using Sensors and the Internet of Things

Emmanuel Baccelli, Gabriel Bartl, Alexandra Danilkina, Veronika Ebner, François Gendry, Christophe Guettier, Oliver Hahm, Ulrich Kriegel, Gabriel Hege, Mark Palkow, et al.

To cite this version:

Emmanuel Baccelli, Gabriel Bartl, Alexandra Danilkina, Veronika Ebner, François Gendry, et al.. Area & Perimeter Surveillance in SAFEST using Sensors and the Internet of Things. WISG 2014 - Workshop Interdisciplinaire sur la Sécurité Globale, Jan 2014, Troyes, France. hal-00944907

HAL Id: hal-00944907
https://hal.inria.fr/hal-00944907

Submitted on 11 Feb 2014

Area & Perimeter Surveillance in SAFEST using Sensors and the Internet of Things

E. Baccelli[1], G. Bartl[7], A. Danilkina[3], V. Ebner[6], F. Gendry[4], C. Guettier[4], O. Hahm[1], U. Kriegel[5], G. Hege[7], M. Palkow[7]
H. Petersen[3], T. C. Schmidt[2], A. Voisard[3,5], M. Wählisch[3], H. Ziegler[5]

[1]INRIA Saclay Île-de-France, 91120 Palaiseau (CEDEX), France

[2]Hamburg University of Applied Sciences, Department of Computer Science, Berliner Tor 7, 20099 Hamburg, Germany

[3]Freie Universität Berlin, Institut für Informatik, Takustraße 9, 14195 Berlin, Germany

[4] SAGEM Défense Sécurité, 100 Avenue de Paris 91300 Massy, France

[5] Fraunhofer Institute for Open Communication Systems (FOKUS), Kaiserin Augusta-Allee 31, 10589 Berlin, Germany

[6] Fraunhofer Institute for Material Flow and Logistics (IML), Joseph-von-Fraunhofer-Str. 9,83209 Prien am Chiemsee, Germany

[7]Research Forum on Public Safety and Security, Fabeckstr. 15, 14195 Berlin, Germany

[7]Daviko gmbh, Berlin, Germany

Contact email: emmanuel.baccelli@inria.fr

Abstract – SAFEST is a project aiming to provide a comprehensive solution to ensure the safety and security of the general public and critical infrastructures. The approach of the project is to design a lightweight, distributed system using heterogeneous, networked sensors, able to aggregate the input of a wide variety of signals (e.g. camera, PIR, radar, magnetic, seismic, acoustic). The project aims for a proof-of-concept demonstration focusing on a concrete scenario: crowd monitoring, area and perimeter surveillance in an airport, realized with a prototype of the system, which must be deployable and foldable overnight, and leverage autoconfiguration based on wireless communications and Internet of Things. This paper reviews the progress towards reaching this goal, which is planned for 2015.

1. Introduction

This paper describes the current status of SAFEST [1], a project aiming at providing a comprehensive solution to ensure the safety and security of the general public and critical infrastructures. Specifically, SAFEST addresses the problems of *crowd analysis and monitoring*, as well as *intrusion detection*, using heterogeneous, networked sensors. As various distributed embedded systems have emerged recently (e.g., in home automation, building automation, healthcare automation, and intelligent transport systems) power-line communications and spontaneous wireless networks are indeed expected to connect heterogeneous devices. These include sensors, home appliances, handhelds, and vehicles, giving birth to the Internet of Things (IoT) [2], a concept heavily leveraged in SAFEST.

The approach taken in the SAFEST project is interdisciplinary in the sense that it complements a technical part (i.e., designing a distributed system for sensing and alerting) with a part focusing on social science aspects (i.e., analyzing the public's awareness and acceptance of such sensing and alerting systems). Scientific and technical challenges in this endeavor are thus very diverse. The key challenges that were identified concern the following fields: sensor hardware design, sensor software platform and techniques for local even detection (such as video processing), communication protocols and knowledge fusion techniques for complex distributed event detection, and methods to evaluate and analyze social acceptance. Figure 1 depicts the overall architecture of SAFEST.

In order to evaluate SAFEST's project realizations, we tightly involve potential end-users of such a sensing and alerting system, over the complete duration of the project. To that end, the consortium includes FBB, which manages Berlin airports. FBB provided continuous feedback and input on the system's requirement analysis and validation, and provides applicable venues to conduct interviews for the socio-cultural analysis. Applicability of the designed system will be further verified by the means of a demonstrator deployed at Berlin's international airport. This setting is particularly adequate since airports present a very challenging and diverse use case with the highest security requirements [31]: operational challenges include the protection of passengers, staff, and critical infrastructure from serious risks such as mass panic, criminal or terrorist activities in a busy, crowded environment. It is furthermore well conceivable that the SAFEST approach will be adaptable to other types of public infrastructures, such as railway stations or stadiums.

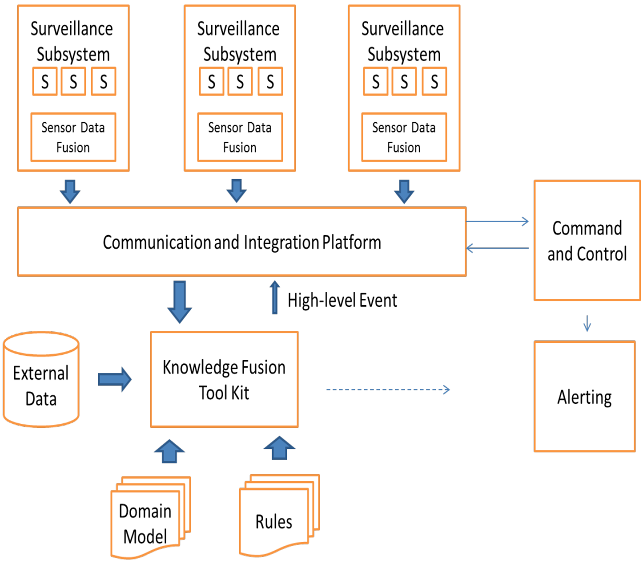

Figure 1: Overall Architecture of SAFEST

The remainder of this paper reviews the current progress of SAFEST in the key domains of sensor hardware design (Section 2), embedded software platform design (Section 3), video processing techniques (Section 4), distributed cooperation platform design (Section 5), knowledge fusion techniques (Section 6), social science studies (Section 7), and simulation and demonstrations (Section 8). Finally, Section 9 presents the next steps of the project.

2. Sensor Node Hardware Design

Design requirements of hardware systems must meet ad hoc deployments of security and crowd monitoring perimeters. Communications can be wireless and must support various types of traffics (unicast, multicast, broadcast, image, video and other perception data). Processing must support detection, as well as knowledge fusion and maintenance algorithms. The hardware design must trade off performance with low power consumptions goals. Its multi-core processing must dynamically adjust to application needs, while supporting distributed processing.

The hardware specifically designed for SAFEST includes (i) an uncooled infrared camera and (ii) a processing node called Smartnode.

Figure 2: Prototype of IR Camera

The infrared (IR) camera architecture follows a compact design and has been optimized to tackle a large variety of public areas like airports or commercial centers with many windows. These buildings are subject to a large range of light and temperature conditions. The camera must indeed support large fields of view; but the IR detection is also very sensitive to the scene, the object to visualize and the environment in terms of light, contrast and temperature. Therefore, dedicated (remote) calibration tools and procedures are required for efficient operation in various deployment environments. The design of electronic boards and video processor must also meet both low power consumption goals and low response times. The IR camera developed and used in SAFEST satisfies these constraints, and integrates a data control layer for video broadcast / multicast in surveillance applications.

The Smartnode architecture must constantly cope with both (i) application-driven distributed processing involved in target detection and tracking, data/knowledge fusion and (ii) sporadic or periodic system-level tasks for ad hoc routing and transport layers, synchronization of processes, shared data updates, monitoring of remote sensors etc. We thus proposed a hybrid processing architecture where a low power consumption processor achieves continuous minimal activity to maintain system synchronization and consistency network wise, while a more powerful processor, dedicated to numerical processing, is woken-up on demand to satisfy system-wide distributed applications. Alternatively, load balancing can also be achieved over the processors if needed.

A prototype of this hybrid architecture designed for the Smartnode was built and is used in SAFEST. This prototype leverages state-of-the-art hardware:

- A Dual-core Cortex-A9 800MHz Nvidia processor to perform sporadic/periodic system-level tasks.
- A 1.6GHz Atom processor to support heavier processing loads.

To ease sensor control and network connectivity, the prototype features a variety of I/O ports types: USB 2.0 (Type A), UART TTL, multi-standard serial port RS422/RS485 half and full-duplex, Ethernet 10/100 ports (RJ45) and PAL video input (BNC).

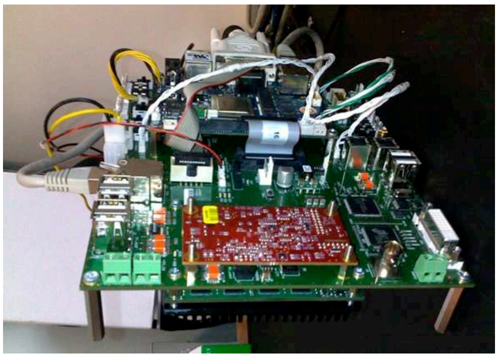

Figure 3: Prototype of Smartnode

3. Embedded Software Platform

The SAFEST project couples two categories of systems: (*i*) a visual and audio surveillance system that monitors large crowds in order to provide guidance in case of unexpected events that trigger mass panic, and (*ii*) a perimeter protection system that uses distributed event detection algorithms to detect unauthorized intrusions.

For the first task, rather powerful hardware is required: The system must be capable of audio-video processing and the amount of data that has to be transferred can be substantial. For the second task, on the other hand, hardware requirements are quite different: light-weight nodes should be scattered over a large area, in which wired power supply may not be feasible and the amount of data that has to be transferred is much less substantial. In order to fit these diverging requirements, very heterogeneous hardware platforms are used. Our analysis [6] concluded that neither conventional operating systems, nor existing compact operating systems (e.g., designed for sensor networks) could meet the diversity of needs in terms of heterogeneous hardware and software integration for SAFEST in particular, and for the Internet of Things in general. We have thus designed an novel IoT middleware: RIOT [3], applicable both in the context of SAFEST and in a wide range of Internet of Things scenarios, which aims at networking together heterogeneous hardware, from nodes based on low-power 16-bit microcontrollers, to nodes powered by new generations of energy-efficient 32-bit processors.

RIOT provides a uniform programming interface across this wide range of devices, allowing multi-threading with standard API with very small memory footprint, starting from 1,5kB RAM and 5kB ROM (without network stack). By design it provides energy-efficiency, reliability, and real-time capabilities, based on a modular, microkernel architecture.

On the high end in terms of hardware CPU/memory capacities, RIOT compares mainly with Linux and FreeRTOS [28]. Compared to Linux, RIOT can scale down to orders of magnitude less memory requirements and offers native real-time capabilities as well as built-in energy efficiency. Compared to FreeRTOS, RIOT offers built-in energy efficiency and a full-featured OS including cutting-edge, free, open-source interoperable network stacks (6LoWPAN, IPv6, and CCN stacks), instead of just a kernel.

On the low end in terms of hardware CPU/memory capacities, RIOT compares mainly with Contiki [26] and TinyOS [27]. Compared to these operating systems, RIOT offers real-time capabilities and multi-threading. RIOT also offers standard POSIX APIs and the ability to code in standard programming languages (i.e., C and C++) using standard debugging tools, thus drastically reduces the learning curve of developers and the software development lifecycle process.

4. Video Processing

SAFEST aims at the protection of people, the goal being to detect unexpected and unusual situations. For monitoring crowds in the context of SAFEST we use infrared cameras (QVGA, λ=8-12μm).

A grid-based approach allows us to model high-level data; the alerting system defines and applies rules on density information. Therefore we divide each scene into non-overlapping regions of the same size and approximate the density by number of persons in each cell.

The raw video data is processed directly on camera nodes in order to reduce the amount of data, which needs to be sent to a central complex event-processing unit. We do not store or transmit any original video frames – the camera node preprocesses the images to anonymized time-stamped objects.

The architecture of the system is designed as follows: first the camera node processes raw video data and produces a stream of objects, which contain the size and the position of detected persons. This information is transformed by a gateway node into a grid-based density map, which is sent to the central event processing unit producing alarms in case of detecting critical density and other predefined dangerous situations. The gateway node does not only aggregate data flows, but also merges information from different nodes and sends information to a knowledge fusion component for further processing. Thus, we call the gateway a merging component.

Crowd detection and counting is a challenging task. Since our approach should be privacy-friendly, we do not focus on systems using analysis of special parts of bodies such as eyes, faces, or silhouettes. We also do not focus on texture analysis such as [21] due to their inability to detect single persons but only roughly estimating number of people or [23] since they are using high-weighted learning algorithms on texture and are not sensor network and real-time capable. Likewise we do not focus on systems using motion analysis [22]. We apply the idea of a multistage approach leading to object-level analysis.

Since the camera is placed on the ceiling and looks vertically down to the scene, people appear as moving “blobs” in the scene. Therefore the main aim of the crowd analysis is to first detect and then count individual persons. The foreground objects – people – need to be extracted from the background. To achieve this, we apply a mixture of Gauss functions for statistical background modeling [18]. Each pixel's intensity is evaluated frame-wise to parameterize several Gauss functions. Adaptive Gauss functions are able to describe different surfaces, lighting conditions, and their changes and as a weighted mixture model the background.

Having a background model we compare it to the image of the current scene. The resulting differences are the foreground pixels, which are then merged into connected regions using connected component analysis [19]. Connected regions represent parts of detected people such as legs or arms. These are then clustered in order to find

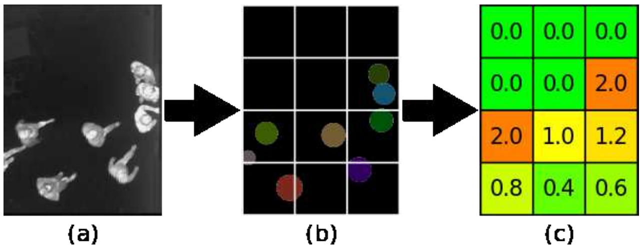

Figure 4: Processing of IR-Video (a), derived people coordinates and radii (b), derived density (c)

the number of individuals in the scene – each cluster represents one or multiple persons. Our clustering technique is based on the principles of density-based reachability [20], which also allows to find non-linearly separable clusters.

While moving, people can come very close to each other or even enter the scene holding hands. In this case, the described approach will find connected regions, which actually represent several people. In such cases, we apply different heuristics to split too large or wrongly merged clusters.

Looking down to people, they can be approximated by circles. Therefore we compute the position and the radius of the approximating circle for each cluster. The result of the video analysis is a list of detected people, their positions and radii. We smooth the data by averaging these parameters over 25 frames per second. This list is then sent by each camera to the merging component. Our approach provides suitable data for further analysis of crowd information, its shape, density, position and behavior without sending whole images throughout the network. Another advantage is that it is impossible to identify or track individual people, reducing privacy concerns.

In the merging component, we compute the crowd density for each grid separately. The discretized density is represented by the number of people per grid cell (see Figure 4). Some people may belong to several grid cells, in case they are positioned on the lines separating the cells. We assign them to the overlapped cells proportional to the area occupied in each cell. We developed geometrical solutions for each of the 16 cases how the grid can divide a person into areas. Furthermore, the merging component has to correct geometrical distortion of the grid at the image edges. It is planned to eventually use more than a single camera to monitor wider areas. Therefore, the merging component computes a single grid-based density map from multiple lists from different cameras. The output of the merging component is one density map per second which is sent to the event processing component described in Section 6 and to the data-mining component for further offline data analysis.

We evaluated our approach using sequences recorded during testing of a live demo [4], which we annotated manually. The results are satisfactory – with a maximum of 11 people in the frame, the number of people counted per frame deviated from the expected value on average by 0.839 persons. The variance of the error equals 0.567. The actual number of people in the scene was determined by manual counting for 320 frames chosen at constant intervals from recorded test sequences. We expect decreasing results for counting by evaluating our approach with default parameters and random sequences. Our current evaluation shows that the algorithm we developed for people counting is able to deal with video scenes showing single people, sparse groups of people, and dense crowds (assuming that the background is still present in the scene, i.e. people are not too densely distributed, and don't form a non-separable "blob" covering the whole image).

5. Distributed Cooperation Platform

Communication in the IoT is required to enfold spontaneously and organize autonomously without infrastructure provisioning. To meet this objective in *low-power and lossy networks* (LLN), the standard routing protocol RPL [5] has been designed and implemented on the RIOT operating system platform [6] of SAFEST.

We have furthermore designed and standardized an extension to the RPL protocol, P2P-RPL [29] [30], which improves the performance of RPL in scenarios where application data traffic between arbitrary nodes in the network would benefit from not flowing systematically through a central node (the root of the RPL tree). For this purpose, based on a reactive approach, P2P-RPL enables on-demand establishment of shorter sensor-to-sensor paths that do not necessarily follow the basic RPL tree structure. This type of traffic is necessary in SAFEST to enable nodes to interact and take decisions locally without central control.

5.1 Enabling Spontaneous, Secure Communication in the IoT

RPL in its current state of standardization is however vulnerable to a variety of severe attacks that build on topological infringements. To cure the deficits, we designed and evaluated TRAIL (*Trust Anchor Interconnection Loop*) [7], which can discover and isolate bogus nodes while they attack the RPL routing hierarchy. TRAIL is derived from first hand principles and shall resolve the issues of topological infringements. Using proper reachability tests, TRAIL reliably identifies any topological attacker without strong cryptographic efforts in a scalable manner. It has been implemented and made openly available on the RIOT platform.

5.2 Programming the IoT

In the IoT, a large number of constraint devices typically cooperate to perform common tasks. This dedication requires a new approach to (highly) distributed programming, which complies to the following environment-specific requirements:

1. High scalability up to thousands of nodes

2. Loose coupling of components to allow for a high degree of independent tasks
3. Enhanced fault tolerance to keep the overall system working even at repeated node failures
4. Low overheads compliant to the constraint environment
5. Native implementations to avoid additional software overheads

The SAFEST consortium has followed the concept of Actor programming as a lightweight approach to highly scalable distributed programming. The actor model is a formalism describing concurrent entities - “actors" - that communicate by asynchronous message passing. An actor can send messages to addresses of other actors and can create new actors. Actors do not share state and are executed concurrently. Because Actors are self-contained and do not rely on shared resources, race conditions are avoided by design. The message passing communication style also allows network transparency and thus applies to both concurrency, if actors run on the same host on different processors, and distribution, if actors run on different nodes connected via the network.

We have contributed libcppa [8], a native actor library written in C++ that adapts the original model to heterogeneous and constraint environments. libcppa enables lightweight distributed programming on embedded devices without introducing interdependencies and faulty conditions in distribution. This new distributed programming environment is extremely light-weight and demonstrates its high scalability in thorough evaluations.

6. Knowledge Fusion Techniques

A knowledge fusion subsystem receives a continuous stream of information from attached sensors, augments and correlates them to extract higher level information, which is beyond the scope of a single sensor. The basis of the knowledge processing subsystem in SAFEST is a distributable implicit middleware for *knowledge processing components* (KPC), developed at Fraunhofer FOKUS [10]. A KPC is a worker component in an event-driven architecture [9]; it is activated on arrival of new events on a given event input stream and it publishes its processing results on an event output stream. In the context of SAFEST, two types of KPC are used: (i) KPCs programmed according to the KPC component model to preprocess and filter the incoming event stream, and (ii) rule-based KPCs that encapsulate the industry-standard of-the-shelf open source rule-based software Drools Fusion [11] to be able to easily adapt fusion logic to new and changing situations. One or more KPC controllers (which can be distributed over different execution environments) load their KPCs, provide a runtime environment for them, and connect them to dedicated event input and output channels. As a communication platform, Redis [13] was chosen. Redis is a distributed and scalable open source key-value store with publish/subscribe features, for which a client implementation exist for almost all mainstream programming languages. This makes it an ideal platform for asynchronous inter-platform communication in SAFEST.

In the context of SAFEST, the knowledge fusion subsystem is used to detect unusual and critical crowd behavior. It receives a stream of density maps from the video processing subsystem and issues alerts when a critical situation is detected according to one of the two implemented mechanisms.

The first mechanism is based on static patterns, allowing the definition of observation regions around critical areas, such as exits, security areas, or evacuation tunnels. Hereby an average person density value will be calculated and observed via a rule engine, which initiates actions (e.g., alert notifications) on fulfillment of certain conditions, such as exceeding density thresholds. This is particularly useful for identification of bottleneck characteristics based on density/velocity ratios. Furthermore, the uncomplicated setup is quite advantageous.

The second mechanism is built on dynamic detection. Opposed to the static patterns, it can be applied to the full camera-observed area and is able to locate and track crowds and to observe size changes. With regards to the implementation on the one hand data clustering algorithms were considered, on the other hand a contour detection mechanism for two-dimensional scalar fields (i.e., marching squares) was investigated. The clustering algorithms included centroid models, distribution models, density models, hierarchical density models and subspace models. The comparison was performed by application to different crowding scenarios and crowd types. The results demonstrate that data clustering techniques have limitations regarding special crowd types, such as low density crowds and non-centric crowds, but also outliers caused difficulties. In contrast, the results of the marching squares algorithm were very satisfying, as even unusual crowd types were analyzed correctly.

As consequence the latter was chosen and integrated into the system. The algorithm was adapted in order to support the specification of density ranges, but also to support recursive analysis, which is required for crowd decomposition and inner structure analysis.

7. Social Science Studies

The goal of the social study is to find out whether surveillance techniques used for crowd monitoring affect privacy issues and thus influence the users’ acceptance of security measures at airports [14]. Besides, the study aims at evaluating how privacy relates to other factors of acceptance such as transparency, health, time and effort, or discrimination.

On the one hand, the process of technical development was reflected by making use of expert interviews with security experts from BER airport. On the other hand, an interview study with airport passengers consisting of a qualitative and a quantitative part aims at identifying relevant dimensions of acceptance of security measures at airports. Thus, the technical development process can be oriented on the passengers' needs and preferences what is seen as a precondition for an efficient use of the SAFEST technology.

The interview guideline of the expert interviews focused on two core topics: technical attributes (e.g., technical edge conditions, and technical product attributes) and social aspects (e.g., social impacts of video surveillance and security culture at the airport). Concerning the technical requirements attributes such as compatibility, adaptability, usability, reduction of complexity, or the exchangeability of technical components were brought up by the experts. Besides, spatial constraints and access to electricity resulting from the architecture of the airport were seen as important aspects. The changing of organizational structures through the implementation of new technical systems was another interesting point that revealed the inseparability of technical and social issues in the context of security measures at the airport.

The first part of the study of acceptance with airport passengers has also been completed. 18 problem-centered interviews could be conducted in September 2013 at the Berlin Airport of Schönefeld [15]. These interviews were analyzed by making use of the method of qualitative content analysis [16]. The results show that acceptance is a multi-factorial construct that varies significantly on the individual and socio-cultural level. This finding will be explored in greater detail in the quantitative survey with flight passengers. Additionally, a preliminary model of acceptance was derived from the interview material that structures various factors of acceptance on different levels like types of acceptance, dimensions of acceptance, subjective perception, personal experience, values and security culture [17] (see Figure 5).

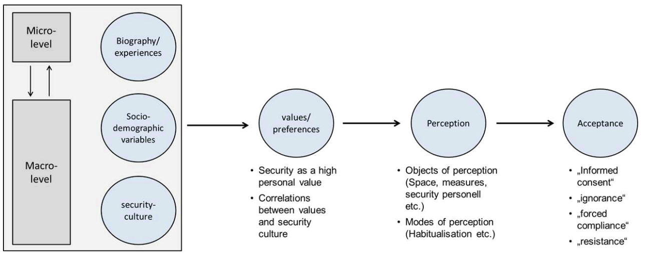
Micro-level
Macro-level
Biography/ experiences
Socio-demographic variables
security-culture
values/ preferences
Perception
Acceptance
• Security as a high personal value
• Correlations between values and security culture
• Objects of perception (Space, measures, security personell etc.)
• Modes of perception (Habitualisation etc.)
• „Informed consent"
• „ignorance"
• „forced compliance"
• „resistance"

Figure 5: Preliminary model of acceptance

8. Simulations and Demonstrations

One of the main tasks of a SAFEST infrastructure is to detect unexpected situations. Therefore, a dedicated simulation environment has been developed, to be able to simulate various scenarios. The simulation output provides series of density maps that can be used to describe the expected course of events, e.g., in case of an incoming flight, an incoming train or an emergency evacuation situation. The results can be used to (1) specify the relevant event patterns and to parameterize the event detection rules, and (2) to test and validate alert conditions. The scenarios can be parameterized by, e.g., the modal split of inbound and outbound passengers (coming or going by car, taxi, bus, train, and so on), by gender aspects, or by the ratio of Schengen to Non-Schengen visitors. The solution is based on an industry-strength micro-simulation for pedestrian streams in 3D space (PTV Viswalk [PTV]) that has been customized by SAFEST specific output modules. Scenarios have been tested using different types of pedestrians featuring different parameters like size and behavior representing the spectrum of airport visitors (business travelers, tourists, or couples and roll chair users). The result is a generic process model combined with best practice approaches describing how to develop an evacuation monitoring solution for an airport based on a SAFEST infrastructure. The process model has been validated using the BER airport as an example.

We have furthermore recently demonstrated an early version of the full system during a live demo at FU Berlin [4], which included a demonstration of (i) the distributed spontaneous wireless network of nodes, using our proposed improvements of RPL, (ii) distributed, simple event detection for perimeter surveillance, (iii) distributed, simple event detection for crowd monitoring using the prototype IR camera and the developed video analysis tools, and (iv) a prototype of the knowledge fusion module and an event visualization tool usable on desktops and/or handheld devices.

9. Next Steps

The next steps in the video processing component will be to optimize the algorithms further to be able to better handle dense crowds. Especially when crowds form very densely packed areas counting of individual people becomes a very challenging task. Therefore detailed sensor data from the camera, such as the applied gain and the observed temperature range in the scene, will be fed into the algorithm in real time. Also, the aggregation and correlation of data from multiple cameras still needs to be tested.

Moreover, we are aiming at integrating "human sensors" or "validators" into the knowledge fusion system, by providing airport personnel with mobile devices and easy-to-use interfaces.

Further hardware developments are currently in the works such as a rugged version and a low-cost version of the IR camera, as well as an improved prototype of the Smartnode based on an ARM Cortex-A15.

We also plan to conduct a comparative analysis and experiments of different IoT network stack elements, using RIOT as common software platform and cyber-physical systems testbeds (such as Senslab [24], DES, FIT [25]). These large-scale testbeds (hundreds of nodes each) will help validate and refine our approach towards tailoring an appropriate network stack for SAFEST. More field tests and fuller software/hardware integration will have to take place in order to prepare for the final demonstration planned in 2015.

In the mean time, we plan to continue our in-depth social study. The results from the problem-centered interviews with flight passengers were used for the construction of the standardized questionnaire. This questionnaire contains items that relate to subjective perception (e.g., cameras and space), acceptance (i.e., privacy in relation to other dimensions of acceptance), security culture at the airport, flight habits (e.g., profession vs. holiday, flights per year), and personal data (i.e., age, gender, education, and occupational field). The survey is planned for early 2014.

References

[1] The SAFEST project: Social-Area Framework for Early Security Triggers at Airports, 2012. [online: http://safest.realmv6.org]

[2] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, Eds., Vision and challenges for realising the Internet of Things. Cluster of European Research Projects on the Internet of Things, European Commision, 2010.

[3] RIOT OS - An operating system for the Internet of Things, 2013. [online: http://www.riot-os.org]

[4] The SAFEST project mid-term demonstration, Berlin, Germany, November 25th, 2013. Video summary [online: http://www.youtube.com/watch?v=WRfZwO-ahrg]

[5] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” IETF, RFC 6550, March 2012.

[6] E. Baccelli, O. Hahm, M. G¨unes, M. Wählisch, and T. C. Schmidt, “RIOT OS: Towards an OS for the Internet of Things,” in Proc. of the 32nd IEEE INFOCOM, 2013.

[7] H. Perrey, M. Landsmann, O. Ugus, M. Wählisch, and T. C. Schmidt, “Topology Authentication in RPL,” Technical Report, No. arXiv:1312.0984, December 2013 [online: http://arxiv.org/abs/1312.0984].

[8] D. Charousset, T.C. Schmidt, R. Hiesgen, M. Wählisch, “Native Actors -- A Scalable Software Platform for Distributed Heterogeneous Environments,” In: Proc. of the 4rd ACM SIGPLAN Conference on Systems Programming and Applications (SPLASH '13) Workshop AGERE!, New York, NY, USA:ACM, Oct. 2013.

[9] T. Everding and T. Foerster, “An Event Driven Architecture for Decision Support,” in An Event Driven Architecture for Decision Support, Münster, 2011.

[10] E. Ulrich Kriegel, Stefan Pfennigschmidt, Hans. G. Ziegler, “Practical Aspects of the Use of a Knowledge Fusion Toolkit in Safety Applications,” presented at the ISADS-2013, Mexico City, 2013.

[11] Drools Fusion. [online: http://www.jboss.org/drools/drools-fusion.html]

[12] PTV Viswalk. [online: http://vision-traffic.ptvgroup.com/en-uk/products/ptv-viswalk/]

[13] J. L. Carlson, “Redis in Action,” Manning Publications, ISBN: 9781617290855, 2013.

[14] Bartl, G. & Gerhold, L., “Soziale Dimensionen der Flughafensicherheit,” Crisis Prevention 1/13. Bonn, 14-15.

[15] A. Witzel, H. Reiter, “The problem-centred interview,” London, Sage Publications, 2012.

[16] P. Mayring, “Qualitative Inhaltsanalyse: Grundlagen und Techniken,” Weinheim, Beltz, 2010.

[17] Bartl, G. & Gerhold, L., “Die Bevölkerung als Adressat der Sicherheitsforschung,” Innosecure, Tagungsband 2013. Berlin/Offenbach, 41-48.

[18] Z. Zivkovic, “Improved adaptive Gaussian mixture model for background subtraction,” Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on , vol.2, pp.28,31 Vol.2, 23-26 Aug. 2004

[19] M. Dillencourt, H. Samet, and M. Tamminen. “A general approach to connected-component labeling for arbitrary image representations,” J. ACM, 39(2):253–280, April 1992.

[20] H.-P. Kriegel, P. Kröger, J. Sander, A. Zimek. “Density-based Clustering,” WIREs Data Mining and Knowledge Discovery 1 (3): 231–240, 2011.

[21] A. N. Marana, S. A. Velastin, L. F. Costa, and R. A. Lotufo. “Automatic estimation of crowd density using texture,” In International Workshop on Systems and Image Processing, IWSIP97, Poland, May 1997.

[22] V. Rabaud., S. Belongie, “Counting Crowded Moving Objects,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.1, pp.705-711, 17-22 June 2006.

[23] X. Wu, G. Liang, K. K. Lee, and Y. Xu, “Crowd density estimation using texture analysis and learning,” in Proc. IEEE Int. Conf. Robotics and Biomimetics, 2006, pp. 214–219.

[24] Senslab: Very Large Scale Open Wireless Sensor Network Tesbed, [online: http://www.senslab.info/]

[25] FIT: The Future Internet (of Things) Platform, [online: http://fit-equipex.fr/]

[26] A. Dunkels, B. Gronvall, T Voigt, ”Contiki - a lightweight and flexible operating system for tiny networked sensors,” in Proceedings of the First IEEE Workshop on Embedded Networked Sensors (Emnets-I), Tampa, Florida, USA, November 2004.

[27] The TinyOS Operating System, [online: http://www.tinyos.net/]

[28] The FreeRTOS Operating System, [online: http://www.freertos.org]

[29] M. Goyal, E. Baccelli, M. Philipp, J. Martocci, A. Brandt, “Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks,” IETF Request For Comments RFC 6997, August 2013.

[30] M. Goyal, E. Baccelli, J. Martocci, A. Brandt “A Mechanism to Measure the Quality of a Point-to-point Route in a Low Power and Lossy Network,” IETF Request For Comments RFC 6998, August 2013.

[31] M. Wählisch, E. Baccelli, J. Schiller, A. Voisard, T. C. Schmidt, S. Pfennigschmidt, M. Palkow, U. Weigmann, U. Hanewald, “Technische Dimensionen der Flughafensicherheit,” Crisis Prevention, No. 1, pp. 15--16, Jan. 2013.