Towards cognitive and perceptive video systems

Toygar Akgun, Charles Attwood, Andrea Cavallaro, Christian Fabre, Fabio Poiesi, Piotr Szczuko

Abstract In this chapter we cover research and development issues related to smart cameras. We discuss challenges, new technologies and algorithms, applications and the evaluation of today's technologies. We will cover problems related to software, hardware, communication, embedded and distributed systems, multi-modal sensors, privacy and security. We also discuss future trends and market expectations from the customer's point of view.

1 Introduction

A smart camera is an image capturing optical device with additional embedded hardware. The device is capable of extracting, processing and communicating information. The data processing and communication capabilities of these embedded de-

Toygar Akgun
ASELSAN Elektronik Sanayi ve Ticaret A.S., Mehmet Akif Ersoy Mahallesi 296 Cadde 16, Yenimahalle Ankara, 06172, Turkey, e-mail: `takgun@aselsan.com.tr`

Charles I. Attwood
THALES UK Limited, Research & Technology, Worton Drive, Worton Grange Business Park, Reading, RG2 0SB, United Kingdom, e-mail: `charles.attwood@uk.thalesgroup.com`

Andrea Cavallaro and Fabio Poiesi
Queen Mary University of London, Mile End Road, E1 4NS, London, United Kingdom, e-mail: `{a.cavallaro,fabio.poiesi}@qmul.ac.uk`

Christian Fabre
CEA, LETI, MINATEC Campus, F-38054 Grenoble, France. Univ. Grenoble Alpes, F-38000 Grenoble, France, e-mail: `christian.fabre1@cea.fr`

Piotr Szczuko
Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Ul. Gabriela Narutowicza 11/12, Gdansk, 80-233, Poland, e-mail: `szczuko@ssound.eti.pg.gda.pl`

vices enable a variety of video-based applications, which include surveillance [28], quality assessment in industrial production [41] and object tracking for security applications [56]. Smart cameras are employed to lower the amount of visual information shared in a network and analysed by an operator, and also to increase situational awareness.

A smart camera system must react to events that a human operator would find of interest with the implicit assumption that some of the events reported by the system might not be of interest to the operator. The goal is therefore to have no missed detections with the minimum false alarm rate. Events of interest are defined within the context of the application. In perimeter-control applications where the activity within the coverage area is expected to be very limited, a typical event of interest is meaningful or intentional motion within the coverage area [17, 40]. Not all changes in pixel values are considered meaningful or intentional. Moving shadows due to changing illumination conditions or terrain specific motion due to wind or water are considered to be part of the background and should not normally be reported to the operator. In the case of airport terminal security, where structured or intentional activities are expected within the coverage area, a typical event of interest is detecting unattended luggage [20, 55, 58]. For monitoring in industrial production, smart cameras must be flexible enough to adapt to various kinds of tasks, such as health and safety or quality control. Video analytics algorithms for quality control vary with the object being manufactured [22].

Market expectations for smart cameras (e.g. in the CCTV market) have traditionally been over-optimistic, compared to the current performance levels of video analytics algorithms and systems. By employing new low cost, low size, low weight and low power processing developments, significant performance improvements are possible. A combination of improved robustness in algorithms with clearly understood user needs coupled with new business models that fulfil both security requirements and offer a return on investment should generate increased user confidence and a cycle of good market growth.

The chapter is organised as follows. Section 2 describes recent advances of processing units and algorithms for smart cameras. Section 3 defines standards for interoperability of smart camera networks and cooperation of different sensor types. Section 4 discusses the communication among cameras while considering privacy. Section 5 covers market expectations and Section 6 draws the conclusions.

2 Processing units and algorithms for smart cameras

Cognitive and perceptive video systems involve largely distributed smart cameras that have limited power/thermal budgets and can communicate with each other to achieve shared goals. Smart cameras can, for example, be mounted on mobile platforms that introduce constraints on the supportable algorithmic complexity and the need for increased power efficiency.

Table 1 Power/performance figures for selected mobile/embedded GPUs. Giga FLoating-point Operations Per Second (GFLOPs) represent the theoretical peak, single precision, combination of fused multiply-add operations (floating-point multiply-add operations performed in one step).

	Release date	GFLOPs	Architecture	Thermal Designing Power (max/min watts)	Process node (nm)
NVIDIA-GT640	Q2 2012	691	GK104	65/15	28
AMD-E6760	Q2 2011	576	VLIW5	35/5	40
Intel-HD5200	Q3 2013	640	GT3e	47/2.4	22

Video data can be processed on smart cameras using Graphics Processing Unit (GPU) platforms [16]. GPUs can achieve high performance in terms of FLoating-point Operations Per Second ($\sim$ 8 TeraFLOPs or TFLOPs) but have several drawbacks. The power consumption of such devices is high ($\sim$250W) and great effort is required to maintain low working temperatures. In turn, devices get physically heavier and are not employable for mobile platforms (e.g. unmanned aerial vehicles). However, embedded GPUs can offer an interesting trade-off among processing power, power consumption and operating temperatures. Examples of embedded GPUs are from NVIDIA (Tegra K1), AMD (E6760, E8860) and Intel (HD5200) (Table 1). These embedded GPUs offer substantial floating point compute power for inherently massively data parallel processing tasks, which are quite common in image and video processing.

As embedded platforms are being used for computing tasks with ever increasing computational complexity, the level of performance expected from embedded processors is also increasing. The main trend in both embedded and desktop computing is shifting from highest performance to highest performance per watt [3].

Engineers are moving towards many-core platforms with much lower power consumption ($\sim$2W) at the cost of lower performance ($\sim$80 GigaFLOPs or GFLOPs). By implementing many-core template architectures [49] on advanced silicon technologies like FD-SOI (Fully Depleted Silicon on Insulator), the GFLOPs/W ratio can be improved [61]. For example, a ratio of 20 GFLOPs for 105 mW, or 380 GFLOPs for 2W, could be achieved in the next few years.

In order to have flexible platforms that simultaneously meet performance and power efficiency targets, fixed function hardware blocks can be combined with specialised accelerators (e.g. DSP, FPGA, GPU) and/or general purpose pre-processors (e.g. ARM, x86). This approach is already in use in modern Systems on a Chip (SoC) with successful results. Specialised accelerators and general-purpose processors have fixed function hardware blocks that implement frequently used mathematical operations (e.g. transcendental functions) or application-specific tasks (e.g. block sum of absolute values computation for motion estimation). Even larger fixed function hardware blocks with limited control parameter programming functionality are used for tasks such as video encoding and image/video filtering.

Heterogeneous platform architectures for smart cameras are available and we can identify two broad processing architectures, namely General-Propose Symmetric Multi-Processors (GPSMP), and General-Purpose computing on Graphics Processing Units (GPGPU) or programmable graphics units. GPSMPs have strong memory

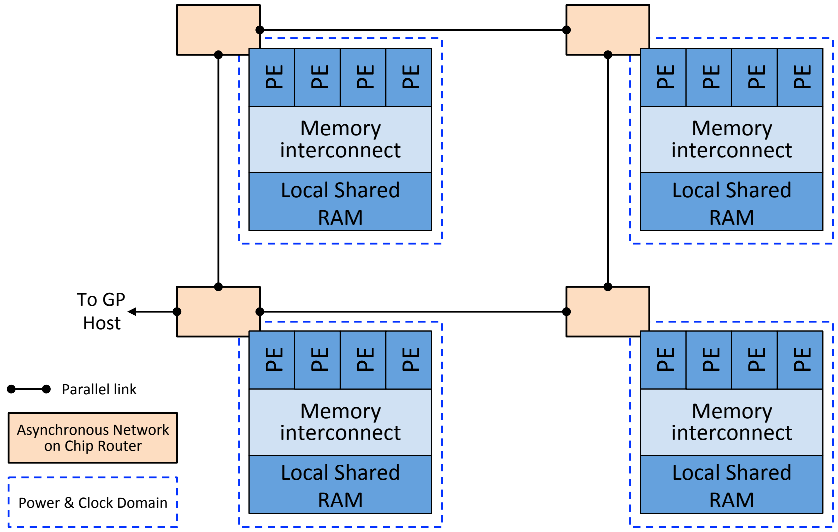
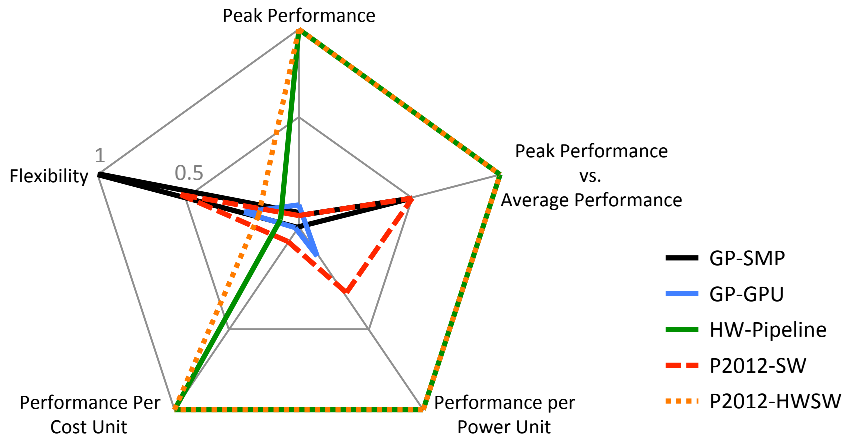


Fig. 1 NoC-based GALS architecture with four many-core clusters (power+clock) with four Processing Elements (PE) per cluster. Key: GP: General Purpose.

consistency models that limit them to (about) a dozen cores, while GPGPUs exhibit much more parallelism with thousands of cores and the same program/instruction executed on several datasets simultaneously.

For example, many-core architectures are based on a Globally Asynchronous Locally Synchronous (GALS) architecture [48] where a number of synchronous tiles are connected through an asynchronous NoC (Network on Chip). Each tile can have its own clock and power domain and is made of up to 16 Processing Elements (PE), each with their own flow of instructions, connected in an SMP fashion around local memory [49]. Because each cluster is controllable in frequency and voltage, the overall available computing power can be adjusted to the computing demand, therefore also controlling electrical power (Fig. 1). This kind of architecture can be further tuned for video analytics by means of hardware accelerating blocks in clusters (i.e. a mixed hardware-software many-core) or by dedicated instructions (e.g. 16 bit floating point arithmetic or specific instructions to compute the sum of absolute difference on arrays).

Solutions from low-power embedded systems, such as GALS many-cores on FD-SOI, can bring significant benefits faster than the downscaling of desktop technologies. For instance, many-core template architectures can be specialised by class of applications. This can be done either through dedicated hardware blocks or by means of additional instructions, and can also provide fine-grain power control, e.g. at the cluster level. These architectures are more programmable, especially when data-dependent algorithms are being used, such as in video analytics and data fusion.



	GP	GPGPU	16 Processing Element Cluster	Hardware blocks
GFLOPs/mm^2	1	3	6	>100
GFLOPs/mm^2/W	1	2.3	25	>100

Fig. 2 Positioning of many-core with respect to General-Propose Symmetric Multi-Processors (GPSMP) and General-Purpose Computing on Graphics Processing Units (GPGPU) [49].

Such a variety of platforms leaves room for more flexible, fully programmable, and possibly heterogeneous many-core platforms that can serve data-dependent classes of algorithms. Fig. 2 summarises the positioning of many-core with respect to traditional architectures.

Real-time algorithms currently running on smart cameras include motion detection, block-based motion estimation, adaptive histogram equalisation, bounding-box drawing, denoising, sharpening along with basic raw sensor data processing such as demosaicing, normalisation and colour processing [12, 19]. Most of these algorithms achieve real-time performance for standard definition frame resolution (PAL, NTSC or VGA) and typically with the help of platform (ARM+ASIC or ARM+ASIC+DSP/FPGA) optimised code. However, increasing frame rates, bit depths and frame resolutions are affecting current implementations. The need for hardware improvements providing higher memory bandwidth and processing is thus increasing.

The adoption of compute-capable GPUs in embedded systems can enable multi-threaded data parallelism [4]. OpenCL [8] can be used for programming heterogeneous multi/many-core platforms to achieve data and task-level parallelism. It is possible to off-load compute and memory bandwidth intensive frame processing to GPUs, or similar multi-core co-processors, leaving the remaining embedded processors available for additional functionality that best fits their architecture [30, 51, 54]. However, it is possible that the porting of some algorithms on smart cameras, which integrate parallel devices such as GPUs or multi-core accelerators, may not be able to achieve user-expected performance levels. This concern is mainly due to algorithm specifics, limited memory and core clock speeds, limited power consumption requirements or strict thermal budgets. Importantly, the nature of some algorithms may also not allow parallelisation [36, 39, 52].

Engineers are hence coping with the portability of algorithms from non-integrated to embedded processors. The major issue is achieving the same performance on both device types. This target motivates the integration of programmable accelerators inside the board to speed-up basic operations such as pixel-wise frame difference. Another key aspect is the programmability of the various kinds of architectures and portability of the application code [29]. This will be achieved by using industrial frameworks that bring together extensions to existing languages with their runtime systems [8, 10, 13], or by using emerging frameworks that address the specific needs of image processing algorithms and video analytics [44, 45].

3 Networks of smart cameras

Modern large area surveillance networks support multiple high-resolution cameras with high frame-rate video streams. Centralised data processing requirements can be addressed by increasing the memory and compute bandwidth, for example via processing units that employ high memory bandwidth and hardware accelerators such as GPUs. Such hardware accelerators can provide substantial performance gains and subsequently achieve much larger camera-feed per processor ratios [31]. Alternatively, a mixed centralised and distributed processing model can reduce both the transmission bandwidth between edge nodes (e.g. smart cameras) and a central control station and, possibly, the amount of processing that needs to be done at the central station.

Manual configuration of cameras in large area surveillance networks is costly, thus making autonomous and self-adaptive smart cameras desirable. Self-adaptive smart cameras must be capable of self-calibration, cooperation with heterogeneous hardware to identify neighbouring cameras and satisfy task requirements [53]. The advances in low-power computing discussed in the previous section bring computing power near to the sensors, which is key to reducing communication overheads.

Several solutions exist that attempt to interface heterogeneous cameras in a network, although none claim outright universality. An example is the Camera Link standard, version 2.0 managed by the Automated Imaging Association (AIA) [1]. Camera Link specifies camera connectors and a real-time communications protocol, provides standardised connections to programmable circuit hardware (e.g. Frame grabbers, FPGAs), and has enjoyed some market adoption. Hybrid analog and digital cameras are currently necessary to allow heterogeneous network devices to inter-connect. Analog to IP converters help integrate legacy analog systems into IP-networks, though with very limited sensor control. Since components within smart cameras, or smart cameras themselves, are produced by multiple vendors, the collaboration among cameras has to be achieved among bespoke systems.

Two standards for IP cameras are emerging as dominant players in the market, the Physical Security Interoperability Alliance (PSIA) standard [9] and the Open Network Video Interface Forum (ONVIF) specification [15].

PSIA covers a range of products, not limited to IP cameras. The group behind PSIA formed from smaller companies and has seen reasonable adoption of its standard for IP video. The PSIA Recording and Content Management (RaCM) specification, combined with the PSIA IP Media Device specification, enables Digital Video Recorders, Network Video Recorders and Video Management Systems from different manufacturers to interoperate and to control different devices (e.g. cameras and encoders) in a video surveillance network.

ONVIF (started in 2008 by Axis communications) defines a common protocol for the exchange of information between network video devices, including automatic device discovery, video streaming and metadata. The standard is aimed at the surveillance market for intelligent cameras and analytics. In September 2013 more than 3,700 products were ONVIF conformant and more than 460 manufacturers, distributors and others were ONVIF members.

Software interoperability solutions have also been emerging. Genetec, for example, provides a software solution by the name of Omnicast [7], which claims to integrate a large number of IP cameras in such a way that an existing infrastructure should be interoperable with any new hardware selected by a customer.

A smart platform may incorporate multiple sensors, and blend them to boost event detection and classification performance. Night vision modules [24] and stereo pairs [35] can be embedded in smart cameras to boost perceptive capabilities. 3D object measurements can be used to improve type classifications [60] and to increase robustness in the case of occlusions [50, 59]. Smart cameras can perceive the surroundings also by integrating other sensors, such as microphones for gunshot detection and localisation, infrared motion sensors and radio-frequency identification for staff authorisation [32]. A video stream associated to an audio stream to enhance detection, identification and classification of events, and can broaden the type of applications that can be addressed, such as automatic camera re-orientation when an audio source location of interest is detected [46], or multi-modal object tracking [18, 59]. Some classes of sound events have been accurately detected and classified in research [42, 43] and in market products [2, 14]. Further enhancements involve robustness against difficult noise conditions and source localisation [47]. For the latter, dedicated acoustic sensors can be based on direction of air particles flow [5,62] or beam-forming with time difference of arrival [38]. In order to increase the detection rate, one can employ an integrated solution of an acoustic-enabled Pan-Tilt-Zoom camera with sound source classification and localisation [46] (Fig. 3).

4 Privacy

Cooperation between smart cameras may lead to potential security problems as data communications may be accessible to unauthorised third-parties [23]. Moreover, be-

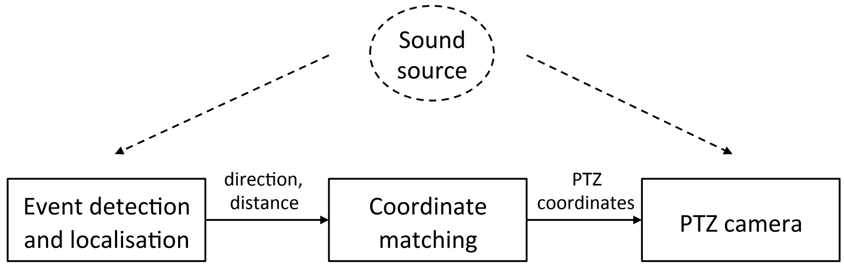


Fig. 3 General scheme of sound source detection, localisation and tracking, with automatic camera positioning. Key. PTZ: Pan-Tilt-Zoom.

cause of the use of images of individuals, privacy has to be considered in the design of cognitive and perceptive video systems. Since embedded image processing supports detection of objects of interest using raw sensor data, object masking, as the simplest mean of privacy protection, can be performed before stream encoding for transmission [34]. The original images could be encrypted and stored in the camera itself for a short period of time. Privacy in surveillance can be addressed in numerous ways [21]. Generally, people's faces and license plates are the most prominent personal information readily extracted. A range of privacy-protection algorithms can be directly implemented in smart cameras in order to achieve a sufficient level of privacy in secure data communications. One simple approach is streaming-on-demand, in which the camera notifies an operator of detected events, and the operator engages the video stream if required. In such an approach the processing and archiving could be done within the camera itself [34].

Real-time anonymisation can also be a solution to the privacy problem [23]. People in a scene can be made anonymous by blurring the portion of the images representing their face and by storing the original non-blurred face for future use, e.g. inside an encrypted watermark. To reduce privacy-loss due to low face detection accuracy the whole top part of the detected object's silhouette can be masked. This reduces false-negatives as moving object detection can be considered reliable in most cases. Face blurring ensures that identities of people are safely managed, but at the same time the process of anonymisation can make other tasks difficult to perform, such as person re-identification. This can be dealt with by parameterisation of the original face images. Instead of distributing an image, the re-identification among numerous cameras could be based on numerical description - image features are not directly associated with the person identity, but are useful in confirming a visual similarity of objects seen by various cameras [25, 33].

Selective data erasure for privacy protection assumes that all footage, except identifiably important videos, should be deleted after predefined short periods of time. Important videos could be defined by event detection and re-identification of individuals [27] or with abandoned luggage, counter-flow and barrier crossing detection [26].

Smart camera networks could generate a fully symbolic representation of the monitored scene state (e.g. number of people entering through a particular door or in a highly sensitive zone or events of barrier crossing per hour). These could be the only data transmitted to an end-user application by re-creating a virtual reality representation of the real scene - anonymous avatars walking in a 3D environment, mimicking a person's behaviour, providing an operator with a comparable situational awareness to that of a traditional video feed [57].

5 Market expectations

Typical CCTV (smart cameras for surveillance) end-user budgets allow them to spend only a small amount per camera as CCTV procurement is often part of a small physical security budget which tends to be seen as a drain on an organisation's resources. CCTV installation is traditionally a highly competitive, price-sensitive market with slow-growth. The addition of smart cameras has allowed a small increase in prices to be introduced, but in general CCTV users commonly expect the same or only marginal increases in price per camera. This effective cap on unit cost may have tended to force manufacturers, and their algorithm developers, to constrain their approaches to low computational cost processes that will run on standard Personal Computer (PC) architectures, thus tending to force smart camera system developers to limit themselves to low-computational complexity approaches to basic visual processing steps such as object detection, classification and tracking. Besides the constant search for better algorithmic approaches, huge increases in processing power are needed to ultimately allow high frame-rates, image resolutions and motion, texture and colour modelling to be utilised. For example, global minimisation techniques such as Simulated Annealing or Gibbs Markov Random Fields [37] can require millions of operations per pixel, and these might be a small part of a much larger set of complex processing stages. Whilst optimisations and cuts often help, they do not allow the developer free reign in combining the ideal combination of processing steps. In a recent thorough review of pedestrian classifiers [28], the best performing classifier was running at roughly five minutes per frame on standard NTSC images (although the authors note that many speed-ups are possible).

The introduction of cameras, with at-the-edge processing capabilities, is beginning to improve the situation, but cannot currently offer the level of processing that the most robust visual processing would consume.

CCTV end users tend to either have little or no conception of the potential offered by the current breed of commercial systems. Or have wildly over-optimistic expectations, perhaps fuelled by film and television (and also over-selling by some systems salespersons), of what is possible using smart cameras. Worse still, developers seldom get to understand the true needs or requirements of CCTV end-users. This lack of dialogue leads to a technology driven set of applications that seldom meet a genuine end-user need. A case in point might be the smart camera systems

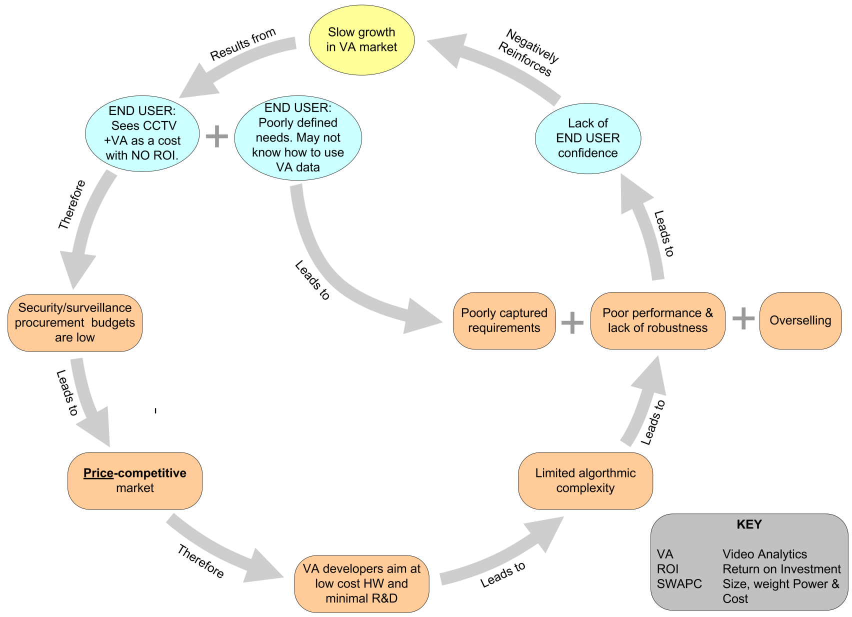


Fig. 4 Perceived market failure due to poor requirements definition and low smart CCTV user confidence.

offering 'loitering' detection, often marketed into the public transport sector. 'Loitering' detection is a natural spin-off of people tracking technologies. However, in the transport domain, the incident of 'loitering' passengers is so high (i.e. waiting for connections is a natural part of travel) as to have little or no value to the end user. At the time of marketing, the purchaser would be presented with the scenario that loitering individuals up to no good would be detected, which sounds attractive, due to the semantic overloading of the term 'loitering'. Unfortunately, after a system is in operational use they then discover it cannot distinguish between 'good' and 'bad' loitering. This poor experience then feeds into the end user's negative perceptions of the benefits of smart cameras, and reduces their confidence in the concept, and reinforces the resistance to pay a significant mark-up on top of the CCTV cost for smart camera capabilities. High false alarm rates and the mismatching of capability to user needs therefore holds up both commercial progress and application performance, in a negative feedback loop (Fig. 4).

The smart camera market has seen a plethora of relatively low-cost systems appear in the last few years. Typical systems use a standard PC-based architecture and most use standard CPU processing, although smart cameras with embedded systems have improved the computational power available per camera.

The preceding comments (summarised in Fig. 4) might seem to paint a rather bleak future for smart cameras; however, there are ways forward. The push for business innovation over the last decade has shown that by merging previously disparate information sources new and valuable knowledge can be gained. Such gains can be the basis of developing new business cases that change the smart camera system from being seen as a drain on resources to being a business opportunity and a revenue generator in itself. In essence, we should try and offer a surveillance capability that pays for itself. For example (i) reduce human work load or increase productivity (e.g. monitor more cameras with same number of staff) by using analytics to pre-filter large video archives for relevant information; (ii) preventative security (e.g. reduce vandalism repair costs in the Rail industry via detection of intruders at Rail stock yards and preventing graffiti tagging; (iii) create metadata that can be sold or aid the business and show a customer how to do it (e.g. counting the number of people passing specific locations in transport hubs to set advertising rates, and subsequently verify to advertisers that "hit rate" targets were met).

Another key aspect to managing user expectations and the acceptance of smart cameras is that the installation and maintenance of such systems should be as easy, robust and self-adaptive as possible. Additionally, honest collaboration with the end users is important to meet their expectations as opposed to proposing products that are a poor fit or unsuitable for their purposes. A recent trend in this area has been for sellers to appoint a "Customer's friend" (an extension of the Account Manager role). This individual works with the user organisation and challenges their own company where they see any sign of a mismatch or over-selling.

The goal is to create a beneficial circle (Fig. 5) so that as end users start to see a smart CCTV system as a net benefit, there is a potential for procurement budgets to be raised because they can now pay for themselves. That could (or should) lead to a more performance-based competitive market as opposed to a cost-competitive one, and it would enable the advancement of video analytics work using superior hardware and heavier investment in research and development costs.

New capabilities can therefore be generated, new products can be offered to customers and new applications can be better matched to real customer requirements. Finally, we should achieve the removal of existing negative feedback loops and the closure of a positive beneficial circle with a new level of end user confidence and approval.

6 Conclusions

Today's mobile and embedded processors are capable of what desktop computers could do ten years ago at substantially lower power consumption levels [6, 11]. As embedded platforms are improving in terms of raw computational power and available memory bandwidth, it is becoming possible to implement computationally complex processing tasks on these platforms. This trend is likely to continue thanks to developments in silicon process technology and architectural breakthroughs. Smart cameras are already benefiting from these developments by incorporating the newly available processing power to tackle more and more complex processing tasks locally leading to improved response times, more capable algorithms, lowered network traffic and enhanced overall system performance.

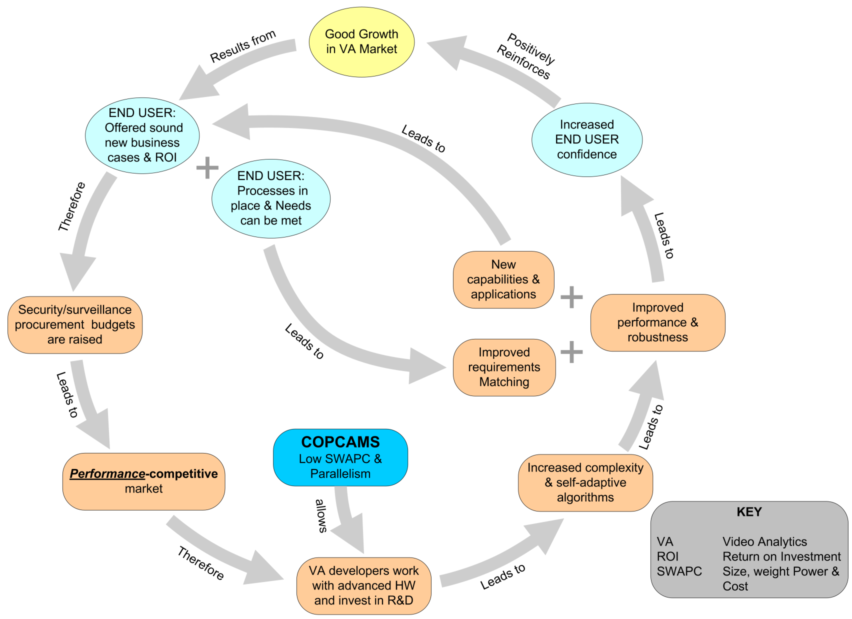


Fig. 5 Market success model that would allow end users to see smart CCTV systems as a benefit. COPCAMS: Cognitive and perceptive cameras (http://copcams.eu).

Nowadays heterogeneity is key given the diversity of cameras that can be found in many large legacy networks. This capability will be needed for some time to come, as the renewal cycle for large camera networks is typically long due to the large costs involved in network, camera and analytics replacement.

Equipping smart cameras with a variety of sensors can result in a broader range of business models, for example an increased number of possible events to be detected (e.g. gunfire, screaming), and robustness in extreme conditions (e.g. object detection in vision versus thermal vision in low light). Advanced technologies can be applied for privacy protection, increasing the societal acceptance of surveillance or to broaden the range of surveillance applications, for example by enabling the collaboration amongst aerial and terrain vehicles.

Acknowledgements

This work has been partially funded by the Artemis JU and partially by TUBITAK - The Scientific and Technological Research Council of Turkey (Toygar Akgun), the UK Technology Strategy Board (Charles Attwood, Andrea Cavallaro, Fabio Poiesi), French *Ministère de l'économie, du redressement productif et du numérique* (Christian Fabre) and Polish National Centre for Research and Development (Piotr

Szczuko) as part of the COPCAMS project (http://copcams.eu) under Grant Agreement number 332913.

References

1. Camera Link Standard: `http://www.visiononline.org/vision-standards-details.cfm?type=6`. Last accessed: June 2014
2. Cisco Gunshot Location Surveillance: `http://www.cisco.com/web/strategy/government/solution_GunshotLocationSurveillance.html`. Last accessed: June 2014
3. DARPA PERFECT: `http://www.darpa.mil/Our_Work/MTO/Programs/Power_Efficiency_Revolution_for_Embedded_Computing_Technologies_(PERFECT).aspx`. Last accessed: June 2014
4. MALI OpenCL SDK: `http://malideveloper.arm.com/develop-for-mali/sdks/mali-opencl-sdk/`. Last accessed: June 2014
5. Microflown: `http://www.microflown.com/`. Last accessed: June 2014
6. NVIDIA Whitepaper: `http://www.nvidia.com/content/PDF/tegra_white_papers/tegra-K1-whitepaper.pdf`. Last accessed: June 2014
7. OmniCast: `http://www.genetec.com/solutions/all-products/omnicast`. Last accessed: June 2014
8. Open Computing Language: `https://www.khronos.org/opencl`. Last accessed: June 2014
9. Open Network Video Interface Forum: `http://www.onvif.org/`. Last accessed: June 2014
10. A parallel computing platform and programming model: `www.nvidia.com/object/cuda_home_new.html`. Last accessed: June 2014
11. SAMSUNG Whitepaper: `http://www.samsung.com/global/business/semiconductor/minisite/Exynos/data/Enjoy_the_Ultimate_WQXGA_Solution_with_Exynos_5_Dual_WP.pdf`. Last accessed: June 2014.
12. Smart camera NI: `http://sine.ni.com/nips/cds/view/p/lang/en/nid/210036`. Last accessed: June 2014
13. A specification for parallel programming: `http://openmp.org/wp/`. Last accessed: June 2014
14. SST, Shotspotter Flex: `http://www.shotspotter.com/solutions`. Last accessed: June 2014
15. The Physical Security Interoperability Alliance: `http://www.psialliance.org`. Last accessed: June 2014
16. What is GPU accelerated computing?: `http://www.nvidia.com/object/what-is-gpu-computing.html`. Last accessed: June 2014
17. Abdelkader, M., Chellappa, R., Zheng, Q., Chan, A.: Integrated motion detection and tracking for visual surveillance. In: Proc. of Computer Vision Systems, pp. 28–34. New York, USA (2006)
18. Arguedas, V., Zhang, Q., Izquierdo, E.: Multimodal fusion in surveillance applications. In: B. Ionescu, J. Benois-Pineau, T. Piatrik, G. Qunot (eds.) Fusion in Computer Vision, pp. 161–184. Springer (2014)
19. Belbachir, A., Nabil, A. (eds.): Smart Cameras. Springer (2010)
20. Bhargava, M., Chen, C., Ryoo, M., Aggarwal, J.: Detection of object abandonment using temporal logic. Machine Vision and Applications **20**(5), 271–281 (2009)
21. Cavallaro, A.: Privacy in video surveillance. IEEE Signal Processing Magazine **24**(2), 168–166 (2007)
22. Chen, X., Yang, S.: A practical solution for ripe tomato recognition and localization. Journal of Real-Time Image Processing **8**(1), 35–51 (2013)

23. Cichowski, J., Czyzewski, A.: Reversible video stream anonymization for video surveillance systems based on pixels relocation and watermarking. In: Proc. of International Conference on Computer Vision Workshops, pp. 1971–1977. Barcelona, Spain (2011)
24. Corsi, C.: Infrared: A key technology for security systems. In: F. Baldini, et al. (eds.) Sensors, Lecture Notes in Electrical Engineering, vol. 162, pp. 37–42. Springer (2014)
25. Dalka, P.: Multi-camera vehicle tracking using local image features and neural networks. In: D. Andrzej, C. Andrzej (eds.) Multimedia Communications, Services and Security Communications in Computer and Information Science, vol. 287, pp. 58–67. Springer (2012)
26. Dalka, P., Bratoszewski, P.: Visual detection of people movement rules violation in crowded indoor scenes. In: D. Andrzej, C. Andrzej (eds.) Multimedia Communications, Services and Security Communications in Computer and Information Science, vol. 368, pp. 48–58. Springer (2013)
27. D'Arminio, P., et al.: Technologies for granting balance between security and privacy in video-surveillance. In: Proc. of Intelligence and Security Informatics, pp. 278–283. Arlington, VA, USA (2012)
28. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. IEEE Trans. on Pattern Analysis and Machine Intelligence **34**(4), 743–761 (2012)
29. Fabre, C., et al.: PRO3D, Programming for Future 3D Manycore Architectures: Project Interim Status. Formal Methods for Components and Objects Lecture Notes in Computer Science **7542**, 277–293 (2013)
30. Gaster, B., Howes, L., Kaeli, D., Mistry, P., Schaa, D. (eds.): Heterogeneous Computing with OpenCL. ELSEVIER (2011)
31. Guler, P., Emeksiz, D., Temizel, A., Teke, M., Temizel, T.: Real-time multi-camera video analytics system on GPU. Journal of Real-Time Image Processing (2013)
32. Haering, N., Venetianer, P., Lipton, A.: The evolution of video surveillance: an overview. Machine Vision and Applications **19**(5-6), 279–290 (2008)
33. Hamdoun, O., Moutarde, F., Stanciulescu, B., Steux, B.: Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences. In: Proc. of Distributed Smart Cameras, pp. 1–6. Stanford, CA, USA (2008)
34. Helten, F., Fischer, B.: Reactive attention: Video surveillance in Berlin shopping malls. Surveillance & Society **2**(2/3) (2004)
35. Houben, Q., Czyz, J., Tocino, D., Debeir, O., Warzee, N.: Feature-based stereo vision using smart cameras for traffic surveillance. In: M. Fritz, B. Schiele, J. Piater (eds.) Computer Vision Systems, vol. 5815, pp. 144–153. Springer (2009)
36. Jaja, J. (ed.): Introduction to Parallel Algorithms. Addison-Wesley Professional (1992)
37. Kato, Z., Zerubia, J.: Markov Random Fields in image segmentation. Foundations and Trends in Signal Processing **5**(1-2) (2012)
38. Kellermann, W.: Beamforming for speech and audio signals. In: D. Havelock, S. Kuwano, M. Vorlander (eds.) Handbook of Signal Processing in Acoustics, 691-702. Springer (2008)
39. Kirk, D., Hwu, W. (eds.): Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann (2012)
40. Kiryati, N., Raviv, T., Ivanchenko, Y., Rochel, S.: Real-time abnormal motion detection in surveillance video. In: Proc. of International Conference on Pattern Recognition, pp. 1–4. Tampa, FL, USA (2008)
41. Koblar, V., Filipic, B.: Designing a quality-control procedure for commutator manufacturing. In: Proc. of Multiconference Information Society, pp. 55–58. Ljubljana, Slovenia (2013)
42. Kotus, J.: Application of passive acoustic radar to automatic localization, tracking and classification of sound sources. In: Proc. of Information Technology, pp. 67–70. Gdansk, PL (2010)
43. Kotus, J., Lopatka, K., Czyzewski, A.: Detection and localization of selected acoustic events in acoustic field for smart surveillance applications. Multimedia Tools and Applications **68**(1), 5–21 (2014)
44. Lepley, T., Paulin, P., Flamand, E.: A novel compilation approach for image processing graphs on a many-core platform with explicitly managed memory. In: Proc of Compilers, Architecture and Synthesis for Embedded Systems, pp. 1–10. Montreal, Canada (2013)

45. Llopard, I., Cohen, A., Fabre, C., Hili, N.: A parallel action language for embedded applications and its compilation flow. In: Proc. of Software and Compilers for Embedded Systems, pp. 118–127. St. Goar, Germany (2014)
46. Lopatka, K., Kotus, J., Czyzewski, A.: Application of vector sensors to acoustic surveillance of a public interior space. Archives of Acoustics **36**(4), 851–860 (2011)
47. Lopatka, K., Kotus, J., Czyzewski, A.: Evaluation of sound event detection, classification and localization in the presence of background noise for acoustic surveillance of hazardous situations. In: D. Andrzej, C. Andrzej (eds.) Multimedia Communications, Services and Security Communications in Computer and Information Science, vol. 429, pp. 96–110. Springer (2014)
48. Meincke, T., et al.: Globally asynchronous locally synchronous architecture for large high-performance ASICs. In: Proc. of Circuits and Systems, pp. 512–515. Orlando, FL, USA (1999)
49. Melpignano, D., et al.: Platform 2012, a many-core computing accelerator for embedded SoCs: Performance evaluation of visual analytics applications. In: Proc. of Design Automation Conference, pp. 1137–1142. San Francisco, CA, USA (2012)
50. Mittal, A., Davis, L.: M2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In: Proc. of European Conference on Computer Vision, pp. 18–22. Copenhagen, Denmark (2002)
51. Munshi, A., Gaster, B., Mattson, T., Fung, J., Ginsburg, D. (eds.): OpenCL Programming Guide. Addison-Wesley Professional (2011)
52. Roosta, S. (ed.): Parallel Processing and Parallel Algorithms: Theory and Computation. Springer (2000)
53. SanMiguel, J., Micheloni, C., Shoop, K., Foresti, G., Cavallaro, A.: Self-reconfigurable smart camera networks. IEEE Computer **47**(5), 67–73 (2014)
54. Scarpino, M. (ed.): OpenCL in Action: How to Accelerate Graphics and Computations. Manning Publications (2011)
55. Smith, K., Quelhas, P., Gatica-Perez, D.: Detecting abandoned luggage items in a public space. In: Proc. of Computer Vision and Pattern Recognition, Workshop on Performance Evaluation of Tracking and Surveillance, pp. 75–82. New York, USA (2006)
56. Stalder, S., Grabner, H., Gool, L.V.: Cascaded confidence filtering for improved tracking-by-detection. In: Proc. of European Conference on Computer Vision, pp. 369–382. Crete, Greece (2010)
57. Szczuko, P.: Augmented reality for privacy-sensitive visual monitoring. In: A. Dziech, A. Czyzewski (eds.) Multimedia Communications, Services and Security, Communications in Computer and Information Science, vol. 429, pp. 229–241. Springer (2014)
58. Szwoch, G., Dalka, P., Czyzewski, A.: A framework for automatic detection of abandoned luggage in airport terminal. In: G. Tsihrintzis, E. Damiani, M. Virvou, R. Howlett, L. Jain (eds.) Smart Innovation, Systems and Technologies, Intelligent Interactive Multimedia Systems and Services, vol. 6, pp. 13–22. Springer (2010)
59. Szwoch, G., Dalka, P., Czyzewski, A.: Spatial calibration of a dual PTZ-fixed camera system for tracking moving objects in video. Journal of Imaging Science and Technology **57**(2), 1–10 (2013)
60. Wang, Y., Kato, J.: Integrated pedestrian detection and localization using stereo cameras. In: J. Hansen, P. Boyraz, K. Takeda, H. Abut (eds.) Signal Processing for In-Vehicle Systems and Safety, pp. 229–238. Springer (2012)
61. Wilson, R., et al.: A 460mhz at 397mv, 2.6ghz at 1.3v, 32b vliw dsp, embedding fmax tracking. In: Proc. of Solid-State Circuits Conference Digest of Technical Papers, pp. 452–453. San Francisco, CA, USA (2014)
62. Wind, J., de Bree, H.E., Buye, X.: 3D sound source localization and sound mapping using a PU sensor array. In: Proc. of AIAA/CEAS Aeroacoustics. Stockholm, SE (2010)