

Image Recognition Traffic Patterns for Wireless Multimedia Sensor Networks

Ruken Zilan¹, José M. Barceló-Ordinas¹, and Bülent Tavli²

¹ Universitat Politècnica de Catalunya (UPC)
Computer Architecture Department
Jordi Girona 1-3, E-08034 Barcelona, Spain
{rzilan,joseb}@ac.upc.edu

² TOBB Univ. of Economics and Technology
Computer Engineering Department
Ankara, Turkey
btavli@etu.edu.tr

Abstract. The objective of this work is to identify some of the traffic characteristics of Wireless Multimedia Sensor Networks (WMSN). Applications such as video surveillance sensor networks make use of new paradigms related with computer vision and image processing techniques. These sensors do not send whole video sequences to the wireless sensor network, but objects of interest detected by the camera. In order to be able to design appropriate networking protocols, a better understanding of the traffic characteristics of these multimedia sensors is needed. In this work¹, we analyze the traffic differences between cameras that send whole coded images and those that first process and recognize objects of interest using Object Recognition techniques.

Keywords: Wireless Multimedia Sensor Networks, Object and Image Recognition, Traffic Patterns, Imagers, Image Compression, Video Coding.

1 Introduction

Wireless Multimedia Sensor Networks (WMSN) are gaining research interest due to the availability of low-cost cameras and CMOS image sensors, also due to the broad application requirements. Applications of such networks can be listed as: Multimedia surveillance sensor networks, advance health care delivery, personal locator services, traffic avoidance, enforcement and control systems, etc, [1].

WMSN may include the transmission of *snapshots* in which event triggered observations are obtained and *multimedia streaming* in which long multimedia data is sent requiring high data rates and low-power consumption transmission techniques. For example, we can imagine a surveillance application in which several cameras are deployed to control habitat monitoring. When a motion

¹ This work has been supported by Spanish Ministry of Science and Technology under grant TSI2007-66869-C02-01 and EuroFGI Network of Excellence.

sensor detects an animal, the camera sensor takes images and send the data to a sink using a multihop network. In order to minimize the power consumption and network lifetime, it would be interesting to first detect whether the image captures a phenomena of interest (e.g. a particular animal) and second send the minimum data to represent that object. Furthermore, it would be interesting to describe the phenomena from multiple views and on multiple resolutions.

In Wireless Sensor Networks, the traffic sent into the network consists of a few bytes of data. Most of the works consider that sensors react to events or to queries. As an example of typical parameters, in Directed Diffusion [2], the authors consider that each source generates two events per second and each event was modeled as a 64-byte packet while interests were modeled as 36-byte packets at a periodic rate of one interest every 5 seconds with a interest duration of 15 seconds. Another example is T-MAC [3], in which the authors use as a data model based on periodic packets of 50 bytes every second to the sink, packets of 30 bytes every 4 seconds in the neighborhood of the event (and event is produced every 10 seconds) and local packets every 20 seconds. As can be seen, in general traffic in sensor networks is modeled as periodic sources.

In Wireless Multimedia Sensor Networks traffic sources produce *snapshots* or *multimedia streaming*. In any case, if we take a camera and produce frames at a rate that may range between 5 and 30 fps (frame per second) the traffic produced will not be periodic. For instance, it is well known that video streams (e.g. MPEG coded) exhibit heavy-tailed probability distributions and autocorrelation functions with a mixture of Short Range Dependence (SRD) and Long Range Dependence (LRD). This kind of traffic in a sensor network may rapidly consume the sensor batteries and also will fill the buffers of the sensors. Object Recognition is a scientific discipline that focuses on obtaining specific information from images. This research area includes scene reconstruction, event detection, tracking, object recognition, etc. Given most of the applications present in WMSN, we believe that Object Recognition is a discipline that has to be taken into account in the design of most multimedia sensor networks. There are applications that instead of sending the whole set of frames, send a stream of bytes representing part of the scene, such as the edge of an object of interest in the scene (e.g. a person, a car, etc). That means the sensor will produce traffic based on the output of software that manipulates the frames taken from the scene. Many works related to traffic characterization on different network architectures may be found in the literature. O. Rose et al [4], studied the impact of MPEG1 video traffic in ATM networks with detailed statistical analysis. F. Fitzek et.al. [5], also presents statistical results from MPEG4 and H.263 encoded video streams for wire-line and wireless networks. Related to WMSN, C.F. Chiasserini et.al. [6], investigate the possible trade-offs between energy consumption and image quality. The authors mention the high complexity in terms of time and power that may result in performing motion estimation when coding frames. After that, the authors study JPEG performance in still images in which only intra-frame (Discrete Cosine Transform mainly) is performed. Power consumption in video sensor technologies are also studied in several video platforms proposed. Examples are Panoptes,

Meerkats and Cyclops [7], [8] and [9]. Meerkats [8], shows that energy dissipation on computation is no longer negligible and is comparable to the energy dissipation on communications. Meerkats utilizes and object detect based on background subtraction that simply detects motion taken between two snapshots at different short time lags. Cyclops [9], performs object recognition using background subtraction on images taken periodically. AER [10], takes another approach. AER (Address Event Representation) outputs only a few features of interest of the visual scene by detecting intensity differences (motion) information.

However, there is no works able to identify traffic characteristics that these sensor devices may produce. Even more, Meerkats chose DSR (Dynamic Source Routing) as a routing protocol for tests. Although DSR is not a suitable protocol for sensor networks, we think that in order to design appropriate sensor network protocols for video sensing, it is necessary to identify the traffic that these devices produce. Therefore, the objective of this work is to identify some of the traffic characteristics that may be found in Wireless Multimedia Sensor Networks (WMSN) that make use of new paradigms related with computer vision and image processing techniques. These sensors do not send whole video sequences to the wireless sensor network, but objects of interest detected by the camera. In order to, may be in further studies, able to design appropriate networking protocols, a better understanding of the traffic characteristics of these multimedia sensors is needed. In this work, we analyze the traffic differences between cameras that send whole coding images and those that first process and recognize objects of interest using Object Recognition techniques.

The paper is organized as follows, in section 2, Traffic Generation of Framework is explained under three subsections which are General Framework, Video Surveillance in WSN and Experimental Study. In section 3, Statistical Analysis are discussed. Finally in section 4, the paper is concluded.

2 Traffic Generation Framework

2.1 Object Recognition Basics

There is a wide application area for object recognition fields like industry (quality control etc.), medical image possessing, military applications and in the explorations of field and autonomous processing vehicles (cars, robots, etc.). Mainly, identification of an object and determination of its parameters are called Object Recognition. Although there are huge amount of techniques for object recognition depending on the applications, general object recognition methods can be categorized as in Figure 1.

There are typical functions that can be found in many object recognition techniques: *Image acquisition* (either 2D image, 3D volume or an image sequence), *pre-processing* (noise reduction, contrast enhancement and scale-space representation), *detection/segmentation* (one or multiple image regions), *feature extraction* (lines, edges and ridges, or more complex features such as texture, shape or motion related requirements), *high level processing*. Further information on object recognition basics can be found in [11].

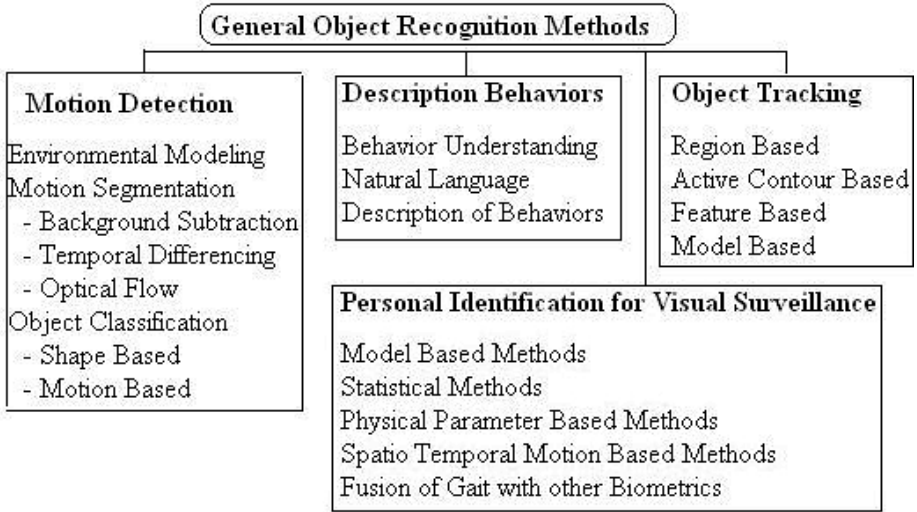


Fig. 1. Generation Object Recognition Methods

2.2 General Framework

The experimental set-up, see Figure 2, consists of a camera that takes video at a configurable rate of M fps (frames/s). Here, we consider two cases: frames f_1 ,

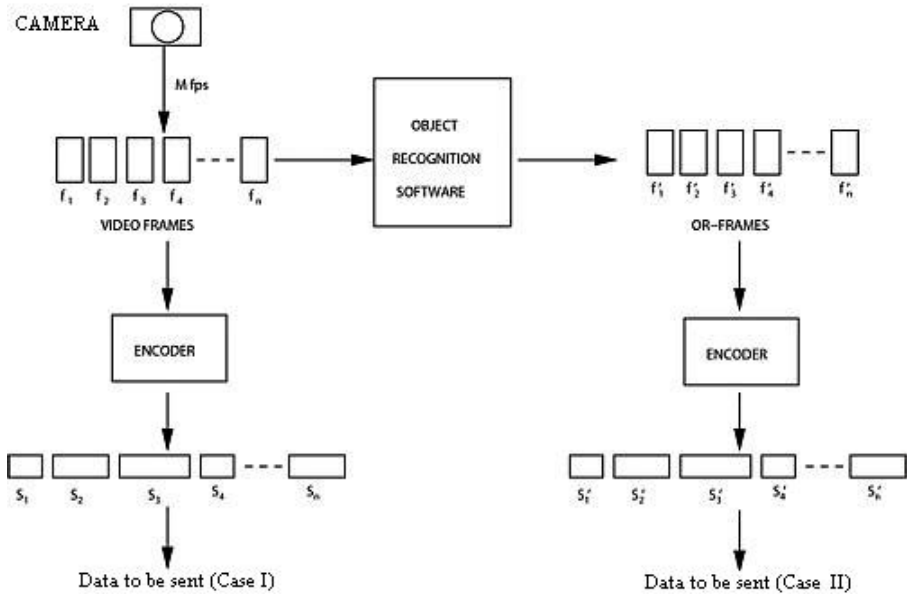


Fig. 2. Traffic Generation Framework

f_2, \dots, f_n may be encoded and be sent to the network (Case I). After encoding, frames have a size (in bytes) S_1, S_2, \dots, S_n that depends on the chosen coding technique. This is the data that should have to be sent to the network if the camera does not implement any detection technique. In the second case frames f_1, f_2, \dots, f_n are fed to one of the OR tools. This output consists of a set of non-encoded OR-frames f'_1, f'_2, \dots, f'_m . In general, and depending on what does the Object Recognition (OR) tool, the number of OR-frames may be different from the original set ($m \neq n$). For instance, some tools represent changes in light intensity with metadata. On the reception side the tools are able to represent detected objects from these metadata packets received. Other tools, subtract the background and outputs frames with the envelope of the object detected. Further compression may be obtained using any coding technique. Therefore, after encoding, OR-frames have a size (in bytes) S'_1, S'_2, \dots, S'_m .

2.3 Experimental Framework

In this study the chosen OR method is *Edge Detection*. It is one of the main steps in feature extraction, detection and segmentation. Edges could be considered as a boundary between two dissimilar regions in an image. Computation of edges are fairly cheap and recognition of an object is easy since it provides strong visual clues however edges can be affected by the noise in an image. Although there are several different methods to perform edge detection, generally they can be categorized under two subtitles; Gradient and Laplacian. While the first one detects the edges by looking for the maximum and minimum in the first derivative of the image, the second one searches for zero crossings in the second derivative of the image to find edges. For this study, *Sobel Edge Detection method* which can be categorized as gradient, is chosen. We have used a software called Video OCX [12] to capture live video as a gradient method and perform image processing. Although Video OCX and its tools are not designed for sensor networks, they are practical tools for image processing that allows capturing live video, displaying and saving AVI files. It also helps for applying image processing functions like edge detection based on Sobel filtering, or motion detection of images.

AER (Address Event Representation), developed by YALE University [10], is an address event sensor that extracts and outputs only a few features of interest from the visual scene. These sensors are sensitive to light and motion so that only pixels that include the brightest one generate the events first and more periodically. Thus, only pixels which experience a high enough difference in the light intensity generate the events. ENALAB at YALE University provides AER emulator for research purposes. AER emulator takes an image from the camera, after some low-level feature detection it produces a frame indicating the importance of the detected feature with the pixel value. After that, an algorithm converts each pixel's magnitude into frequency coding, by connecting it to the other pixels' own streams to produce a stream of address-events. AER uses the maximum event rate value to produce as many as events per pixels. Since AER is designed for sensor networks, we have used AER Emulator to produce traffic



Fig. 3. Original, Edge Detected and AER snapshots

representative of sensor imagers. Figure 3 presents a snapshot of the original sequence, edge detected sequence, and AER sequence.

After Object Recognition, AER processed frames are not optimally encoded in terms of spatial or temporal redundancies. Since, at the moment, no specific encoders are designed for sensor networks, we have used MPEG4. MPEG4 may not be suitable for sensor networks in terms of computation. In Wireless Sensor Networks, the transmission/reception module consumes more power than computation of data. In Wireless Multimedia Sensor Networks, computation of data (e.g. encoding) may be higher than in data-centric sensors. We have not found studies on suitable encoder for WMSN. Nevertheless MPEG encoding requires high computation (e.g. Discrete Cosine Transform, etc), we think that an encoder which includes spatial redundancy compression techniques may reduce the amount of data to be sent in a sensor network. Applying temporal redundancy compression is more difficult due to the need of higher computation capabilities and the need of buffering past/future frames (P and B frames).

The MPEG (Motion Picture Experts group) coding algorithm was developed mainly for the storage of compressed video on digital storage media. These standards are based on coding technologies was developed mainly for digital video compression for storage on digital media. The MPEG video compression algorithm uses two techniques [13]: block-based motion compensation for the reduction of the temporal redundancy and transform domain-(DCT) based on compression for the reduction of spatial redundancy. Motion compensated techniques are applied with both causal (pure predictive coding) and non-causal predictors (interpolative coding). The remaining signal (prediction error) is further compressed with spatial redundancy reduction (DCT). To encode frames, three modes can be used; intraframe (I), predictive (P) and interpolative (B). An I-frame is encoded as a single image, with no reference to any past or future frames. A P-frame is encoded relative to the past references frame. A reference frame is a P- or I-frame. A B-frame is encoded relative to the past reference frame, the future reference frame, or both frames (I or P).

3 Experimental Study

To study the traffic generated by the OR tools we characterize the frame sizes. For this purpose, we analyze three different sets of data: A live captured-video file, a captured and features detected by Edge Detection video file and a video file which is progressed with AER emulator. The scenario can be considered as a surveillance environment processed by the AER emulator in which a person walks in a room in front of the camera. The video sequence lasts around one minute. The experiment emulates a sensor camera that detects an object of interest (a person) and sends the captured images to a sink. The objective is to analyze the different kind of traffic this sensor could generate before sending the data to the sink.

The uncompressed 16-bit IYUV video is captured at different frame rates in the QCIF format (176x144 pxl). When using Edge Detection, files are captured in gray scale. Captured files, first encoded by MPEG4 encoder (IMToo Mpeg Encoder) and then frame sizes are extracted by the usage of MPEG-4 Parser [14], and statistics are taken from these trace files.

Table 1 shows the data sizes generated before and after encoding. At the moment of this work, the capturing software can only work with live video. Therefore, the input of the three methods are different even though they are of similar sequence and time and can not be compared directly. However, the results still are interesting, since they show the different traffic patterns produced by the three methods. In Table 1, we see that the results from different Frame Rates behave in a similar way. When we observe the 25 fps capture, we can see that the Original data to be sent without any compression is 89873 KB and using MPEG4 compression reduces to 4592 KB. Using Edge Detection the data to be sent without compression reduces to 21139 KB, due to the fact that Edge Detection is coding in gray scale using 2 bits/pixel. Further compression using an encoder such as MPEG4 reduces the amount of data to 4585 KB. Although the compression ratio of the original file is higher than the other, the amount of data to be sent is approximately the same. Note that we can not directly compare both outputs since they correspond to different input files (all around 1 minute) and we only consider the magnitude of the results as a first approximation.

However, traffic characteristics, see Table 2, are quite different. Original frames have a higher Peak to Mean Ratio, generally an indication of higher burstiness.

Table 1. Data sizes and Compression rates

Method	Frame Rate	Numb. Capt. Frames	YUV (KB)	MPEG4 (KB)	Compres. Rate
Original	25 fps	1164	89873	4592	19,57
EDGE	25 fps	1272	21139	4585	4,61
Original	20 fps	1272	98266	5012	19,61
EDGE	20 fps	1056	17774	4288	4,15
Original	15 fps	1176	91210	4661	4,38
EDGE	15 fps	804	13721	3133	4,38

Table 2. Data Frame Statistics

Method	Frame Rate	Mean Size (B)	St. Dev.	Coeff. Var.	Peak	Peak/Mean	Min.
Original	25 fps	3833	2561,6	66,82	14529	3,79	1307
EDGE	25 fps	3574	3042,2	85,1	8009	2,24	67
Original	20 fps	3818	2508,1	65,69	14027	3,67	1420
EDGE	20 fps	3978	3399	85,44	8820	2,22	69
Original	15 fps	3823	2585,6	67,63	16093	4,21	1391
EDGE	15 fps	3775	3240	85,83	8788	2,33	70

Table 3. I-Frame and P-Frame Statistics (25 fps)

Method	Type of Frames	Num. Frames	Mean Size (B)	St. Dev.	Coeff. Var	Peak	Peak/Mean	Min.
Original	I	97	11976	1191	9,94	14529	1,21	9865
Original	P	1067	3093	671	21,70	5949	1,92	1307
EDGE	I	106	6050	457	7,55	7149	1,18	5046
EDGE	P	1166	3349	3077	91,87	8009	2,39	67

Peak frames get to the order of 14000-15000 Bytes while Edge detection have peak frames of no more than 9000 Bytes. Separating, see Table 3, I-frames from P-frames helps to see the different compressions obtained in the traces.

After Object Recognition, VideoOCX saves captured video in AVI format. AER emulator outputs data either in metadata or AVI files. AVI formats use a “one frame in, one frame out” scheme. That means that the Presentation Time Stamp (PTS) and the Decoder Time Stamp (DTS) are the same, not allowing B-frames, as B-frames are constructed by using two frames at once, the previous and following I/P frame. Then, the encoding data will only present I and P frames. Table 4 shows the size of 16 frames taken from two consecutive GoP (the behavior is the same on other GoPs). As can be seen, Edge Detection I frames are reduced more than 50%. The reason is that after obtaining the edge, the Intra-frame coding performs better since the background is homogeneous. The results show that Intra-frame compression is a desired feature in these kind of devices.

The behavior of P-frames is totally different. In general, in MPEG encoding, a P-frame is encoded relative to the past reference frame. A reference frame is a P- or I-frame. The past reference frame is the closest preceding reference frame. Each macroblock in a P-frame can be encoded either as an I-macroblock or as a P-macroblock. An I-macroblock is encoded just like a macroblock in an I-frame. A P-macroblock is encoded as a 16x16 area of the past reference frame, plus an error term. To specify the 16x16 area of the reference frame, a motion vector is included. A motion vector (0,0) means that the 16x16 area is in the same position. This shows that Edge Detection decreases the amount of data to be sent. as the macroblock we are encoding. Other motion vectors are relative to that position. Motion vectors may include half-pixel values, in which case

Table 4. GoP sequence after encoding

Original		Edge Det.	
I	11507	I	5867
P	3996	P	5576
P	3272	P	103
P	3173	P	5801
P	2976	P	130
P	3117	P	5801
P	4755	P	111
P	2567	P	6028
P	3842	P	102
P	2924	P	5885
P	3770	P	113
P	3393	P	5754
I	11787	I	6112
P	3110	P	5594
P	3893	P	134
...

pixels are averaged. The error term is encoded using the DCT, quantization, and run-length encoding. A macroblock may also be skipped which is equivalent to a (0,0) vector and an all-zero error term. The search for good motion vector (the one that gives small error term and good compression) is the heart of any MPEG video encoder and it is the primary reason why encoders are slow.

We observe that the all of the original trace outputs' P-frames have the same order of magnitude. However using Edge Detection, we observe sequences consisting of a high-size P-frame (a few thousands of Bytes) followed by a short-size P-frame (few hundreds of Bytes). We think that the short-size P-frame may be due to small changes in the motion of the edge of the object. The high-size P-frame should encode again the whole edge. In any case, it should be taken into account that motion compensation is possible if simple algorithms can be used.

In AER [15], events are signaled when changes in pixel intensity reach a threshold voltage. AER uses 17 bits to encode each event. If N_{ev} is the number of event per frame, then $N_{ev} * 17$ bits/event is the size of the frame. Edge Detection frames before compression have a size of 152064 bits/frame. AER obtain this sizes with $152064/17 = 8944$ events/frame. After compression, Edge Detection produces I and P frames of different sizes. The mean I and P frames in Edge Detection with 25 fps are 6050 and 3349 bits respectively. AER frames with 350 events/frame will produce frames of 5950 bits/frame. So AER with a number of events in the range of around 350 or 400 events produces the same amount of bits that Edge Detection with a complex coding as MPEG4. However, the advantage of AER imagers is that reduce the high compression overhead produced by encoders as is it is explained in [15].

4 Conclusions

Due to the bandwidth limitations and low power requirements of sensors, today every single aspect of a Wireless Multimedia Sensor Network is practically an open research area (like routing, MAC, network protocols). Unlike traditional sensor networks which transfer only small amount of data, visual data processing could be computationally very expensive simply due to the volume and information content of multimedia data. Hence, identification of traffic patterns could be one of the beneficial tools for working on such open areas. Thus, in order to design suitable networking protocols for wireless multimedia sensor networks, traffic should be characterized. These kinds of networks do not need to support streaming video instead of semantics of the sensed phenomenon which is the main function of the network (e.g., tracking objects in a certain physical environment). Since there are only limited-scope studies in the literature on this subject, it is extremely important to obtain a better understanding of the behavior of such traffic sources. In this study, "Object Recognition Techniques" (ORT) are considered as key components for reducing the amount of information to be sent to the sink since they could decrease the frame sizes. Moreover, coding techniques are further considered to reduce the temporal and spatial redundancies in frames.

Even though, there is no suitable encoder for this purpose today, using an unsuitable encoder such as MPEG4 gives an idea about effects of ORT. The results of our primary experiments suggest that using edge detection for scene description can reduce the bandwidth utilization of the network significantly. However complexity of the algorithms should be handled carefully. Motion estimation makes the MPEG encoding algorithm slow due to the fact that it consumes more time and buffer size. It should be kept in mind that an encoder must satisfy the both mentioned requirements and operate in an optimal operating point by trading off the competing requirements. Nevertheless, data reduction with intelligent object and scene description, without using complex algorithms, is the ultimate goal of our study.

References

1. Akyildiz, I.F., Melodia, T., Chowdhury, K.R.: A survey on Wireless Multimedia Sensor Networks. *Computer Networks* 51, 921–960 (2007)
2. Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed Diffusion for Wireless Sensor Networking. *IEEE/ACM Transactions on Networking* 11(1) (2003)
3. Van Dam, T., Langendoen, K.: An adaptive energy-efficient MAC protocol for wireless sensor networks. In: 1st international conference on Embedded networked sensor systems (SenSys), LA, California, USA, pp. 171–180 (2003)
4. Rose, O.: Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems. In: Proceedings of the 20th Annual Conference on Local Computer Networks (1995)
5. Fitzek, F.H.P., Reisslein, M.: MPEG-4 and H.263 video traces for network performance evaluation. *IEEE Network* 15(6), 40–54 (2001)

6. Chiasserini, C.F., Magli, E.: Energy Consumption and Image Quality in Wireless Video-Surveillance Networks. In: The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2002), Lisboa, Portugal (2002)
7. Feng, W.-C., Code, B., Shea, M., Feng, W.-C., Bavoil, L.: Panoptes: Scalable Low-Power Video Sensor Networking Technologies. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 1(2), 151–167 (2005)
8. Boice, J., Lu, X., Margi, C., Stanek, G., Zhang, G., Manduchi, R., Obraczka, K.: Meerkats: A Power-Aware, Self-Managing Wireless Camera Network for Wide Area Monitoring. In: Workshop on Distributed Smart Cameras (DSC 2006), Boulder, CO (2006)
9. Rahimi, M., Baer, R., Iroezi, O., Garcia, J., Warrior, J., Estrin, D., Srivastava, M.B.: Cyclops: in situ image sensing and interpretation in wireless sensor networks. In: 3rd international conference on Embedded networked sensor systems (SenSys 2002), San Diego, California, USA (2005)
10. Teixeira, S., Culurciello, E., Park, J.H., Lymberopoulos, D.: Address-Event Imagers for Sensor Networks: Evaluation and Modeling. In: Fifth International Conference on Information Processing in Sensor Networks (IPSN), Nashville, Tennessee, USA (2006)
11. Zilan, R., Barcelo-Ordinas, J.M.: Object Recognition Basics and Visual Surveillance. Technical report: UPC-DAC-RR-XCSD-2008-1 (2008)
12. Video OCX, <http://www.videocx.de>
13. Sikora, T.: MPEG Digital Video Coding Standards. *IEEE Signal Processing Magazine* (1997)
14. MPEG4-Parser, Video Trace Research Group, Arizona State University, <http://trace.eas.asu.edu>
15. Culurciello, E., Park, J.H., Savvides, A.: Address-Event Video Streaming over Wireless Sensor Networks. In: IEEE International Symposium on Circuits and Systems (ISCAS 2007), New Orleans, USA (2007)