Exploration of Tasks Partitioning Between Hardware Software and Locality for a Wireless Camera Based Vision Sensor Node

Khursheed Khursheed, Muhammad Imran, Abdul Wahid Malik, Mattias O'Nils, Najeem Lawal,Thörnberg Benny
Department of Information Technology and Media, Division of Electronics Design, Mid Sweden University
Sundsvall, Sweden
{khursheed.khursheed,muhammad.imran,waheed.malik,mattias.onils,najeem.lawal,thörnberg.benny}@miun.se

Abstract. **In this paper we have explored different possibilities for partitioning the tasks between hardware, software and locality for the implementation of the vision sensor node, used in wireless vision sensor network. Wireless vision sensor network is an emerging field which combines image sensor, on board computation and communication links. Compared to the traditional wireless sensor networks which operate on one dimensional data, wireless vision sensor networks operate on two dimensional data which requires higher processing power and communication bandwidth. The research focus within the field of wireless vision sensor networks have been on two different assumptions involving either sending raw data to the central base station without local processing or conducting all processing locally at the sensor node and transmitting only the final results. Our research work focus on determining an optimal point of hardware/software partitioning as well as partitioning between local and central processing, based on minimum energy consumption for vision processing operation. The lifetime of the vision sensor node is predicted by evaluating the energy requirement of the embedded platform with a combination of FPGA and microcontroller for the implementation of the vision sensor node. Our results show that sending compressed images after pixel based tasks will result in a longer battery life time with reasonable hardware cost for the vision sensor node.**

Keywords-Wireless Vision Sensor Networks; Vision Sensor Node; Hardware/Software Partioning; Reconfigurable Architecture;Image Processing

I. INTRODUCTION

Typically Vision Sensor Nodes (VSN) in Wireless Vision Sensor Networks (WVSN) consists of an image sensor for acquiring images of the area of interest, a processor for local image processing and a transceiver for communicating the results to the central base station. Due to the technological development in image sensors, sensor networking, distributed processing, low power processing and embedded systems, smart camera networks can perform complex tasks using limited resources such as batteries, suitable Field Programmable Gate Array (FPGA), a wireless link and limited storage facility. Visual sensor systems are very application specific and is hard to generalize an implementation to take advantage of the low power characteristics of a custom made design. Thus, to achieve some volume for these systems the target architecture needs to be programmable. WVSN have been designed and implemented on microcontroller and microprocessor [1, 4]. Often these solutions have high power consumption and moderate processing capabilities. Due to rapid development in the semiconductor technology, the single chip capacity of FPGA increases greatly while its power consumption decreases tremendously [15]. Presently, FPGA chips consist of many cores which makes it ideal candidate for the designing of VSN. As VSN needs to be capable of performing complex image processing such as morphology, labeling, features extraction and image compression, which put higher processing requirement on the processing unit. High processing requirement is even more increased for an increased resolution of the camera, often required in surveillance applications. Designing of real-time embedded systems can be subject to many different kinds of constraints such as low energy consumption, compact size, light weight, high reliability, and low cost [16]. Traditional methods for designing embedded systems require specifying and designing hardware and software separately. An incomplete specification often written in non-formal languages is developed and sent to the hardware and software engineers. Hardware/software partitioning is decided in advance, any changes in this partitioning may necessitate extensive redesign. With the development of the design complexity of the embedded systems, its design procedure has been revolutionized [17]. The concurrent design of hardware and software has replaced traditional sequential design methods. The co-design approach allows testing the software and hardware concurrently at the early design period. Therefore, the problems on the design could be solved easily at an earlier design stage, and the design periods could be shortened in this way. Attention must be paid to the hardware/software co-design strategy to meet both processing and power requirements of VSN [8]. In [9] the authors designed a novel VSN based on a low cost, low power FPGA plus microcontroller System on Programmable Chip (SOPC). The authors in [10] have implemented a computer vision algorithm in hardware. They have provided a comparison of hardware and software implemented system using the same algorithm. They concluded that hardware implemented system achieved a superior performance, which is an obvious result but they did not discussed hardware design time and efforts, which is normally very high compared to software implementation. A vision based sensor network for health care hygiene was implemented in [11]. The system consisted of a low resolution CMOS image sensor and FPGA processor which were integrated with a microcontroller and a ZigBee standard wireless transceiver. A design methodology for mapping computer vision algorithm onto an FPGA through the use of coarse grain

978-0-7695-4397-0/11 $26.00 © 2011 IEEE
DOI 10.1109/PARELEC.2011.21

reconfigurable data flow graph was discussed in detail in [5] and [13]. The pros and cons of FPGA technology and its suitability for computer vision task were discussed in detail in [3] and its optimization in [12] and [14]. The large amount of data generated by a vision sensor node requires a great deal of energy for processing and transmission bandwidth compared to other types of sensor networks. Both on board processing and communication influence energy consumption of the sensor node and that more on board processing reduces the energy consumption due to communication and vice versa [1]. Different software and hardware approaches have been proposed in literature for minimizing the energy consumption in wireless sensor networks [1, 4, 6]. FireFly Mosaic [4] wireless camera consists of a wireless sensor platform. It uses a real-time distributed image processing infrastructure with a collision free TDMA based communication protocol. FireFly is a low-cost, low power sensor platform that uses a real time operating system and an expansion board. SensEye [6] is a multi-tier of heterogeneous wireless nodes and cameras which aims at low power, low latency detection and low latency wakeup. They used low power elements to wakeup high power elements. Partitioning a task in hardware and software parts has significant effect on the system costs and performance. FPGA can be configured to perform specific task with better performance metric than other programmable embedded platforms. The results in [18] shows that using FPGA for vision processing, microcontroller for communication and central base station for particular visual tasks result in a longer life time for the VSN. In our work, we have implemented all vision tasks i.e. image capturing, subtraction, segmentation, morphology, bubble remover, labeling, features extraction and TIFF Group4 compression on both FPGA and SENTIO32 [2] platform, and there is a possibility to perform some of the vision processing tasks such as morphology, labeling, bubble remover and features extraction on central base station. We have explored different possibilities of performing some vision processing task on FPGA and some on microcontroller (SENTIO32) and the rest in the central base station. The focus in this research work is to find the low energy hardware/software partitioning strategy at the vision sensor node which is discussed in more detail in the discussion section. The experimental system is described in Section II, Section III explains results, Section IV considers discussion and Section V concludes the paper.

II. EXPERIMENTAL SYSTEM

The application for our work is the detection of magnetic particles in a flowing liquid. The particles are classified both by their size and number and this system is used for failure detection in machinery. The flowing liquid in the system might contain air bubbles which can be identified as objects. The removal of the bubbles can be handled in two different ways. In the pixel based method, the individual pixels of each bubble are identified and removed from the image, while in an object based method, the whole bubble is treated as a moving object, which can be identified and removed. The following are the main stages of our algorithm.

A. Pre-Processing:

In this step the image is subtracted from the background. The background is initially stored in the flash memory and this stored background is subtracted from the image in order to detect objects which could be magnetic particles or some flowing bubbles. All pixels having a gray scale value less than a pre-defined threshold are assigned a zero value(representing black) and all other pixels in the image are assigned the value one (representing white). The resulting image after segmentation is a binary image having value 1 or 0 for all pixels. A morphological operation is then performed on the segmented image in order to remove one to two pixel false objects. Morphology includes both erosion and dilation operations.

B. Bubble Remover:

Pixel based bubble remover method is applied for the detection and removal of bubbles. Bubbles can be identified as moving objects, so if an object changes its location in two consecutive frames, this confirms that this moving object is a bubble. In the pixel based method, the corresponding pixels in two consecutive frames are compared and if their binary values are different, which confirms it to be a part of bubble and hence a zero is placed at that pixel location, to remove this part of the bubble. In this way, the bubbles are identified and removed.

C. Labeling and Features Extraction:

Each object is assigned a unique label. Following this, the area and location of each object is determined.

D. Image Compression:

TIFF Group4 compression could be performed after segmentation, morphology or bubble remover as shown in Figure 2. In Figure 2, images are taken from a setup of the system in which I is the image after subtraction, II is the image after segmentation and III is the result after the morphological operation. In images I, II and III bubbles are visible which are removed in image IV by applying pixel based bubbles remover algorithm.

E. Communication

The final data is transmitted to the central base station through an IEEE 802.15.4 transceiver, embedded in the SENTIO32 platform.

F. Target Architecture

In our previous work [7] we proved that, by sending a compressed binary image after segmentation from vision sensor node to the central base station over wireless link, will result in a longer life time for vision sensor node. The reason is that, at this stage the energy consumption due to processing and communication are in such a proportion that it will result in minimum energy consumption. Performing the same vision tasks on FPGA would further improve the results, which was proved in our work in [18]. Communication portion in [18] was handled on SENTIO32 platform [2], while the remaining vision processing tasks such as morphology, labeling, features extraction and

bubbles remover were shifted to the central base station. The target architecture for our current work is presented in Figure 1, which includes FPGA, SENTIO32 and central base station. We have investigated different possibilities of performing different level of vision processing operations in the three available processing units shown in Figure 1. FPGA architecture is very important as the leakage current in FPGA significantly affects the results [18]. For analysis purpose we used Xilinx Spartan 6 Low Power FPGA [20] and Actel IGLOO Low-Power Flash FPGAs [19]. To achieve general result that is not limited by the logic size, memory size and sleep power of today's FPGA technology, we have measured the dynamic power and gate count using Xilinx Spartan 6 technology and the sleep power using Actel IGLOO. This way we can draw general conclusions on technology requirements in the future. SENTIO32 is a platform for wireless sensor networks developed at Mid Sweden University and has a high performance, low power AVR32 32bit RISC MCU running at 60MHz and needs only 23.5mA when operational. It has a CC2520 RF transceiver with 2.4 GHz IEEE 802.15.4, with on-board antenna. It has 256KB flash, 32KB SRAM and has a low sleep current of 60µA. CMOS image sensor is used for capturing the image of the field of view. The central base station will perform labeling, features extraction and will also analyze the results and respond accordingly.

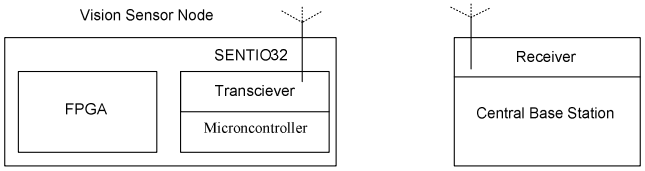

Figure 1. Architecture of the vision sensor node and server.

III. RESULTS

In our current work, we have implemented all vision processing task i.e. background subtraction, segmentation, morphology, labeling, features extraction and TIFF Group4 compression on both hardware and software. Our focus in this research work is to find a combination of hardware and software module at the vision sensor node in order to maximize the life time of the vision sensor node. For effective hardware/software partitioning all possible combinations of the vision processing tasks are analyzed and mentioned in TABLE III, and some of them are discussed in detail in the discussion section (Section IV). The execution time of vision processing tasks i.e. background subtraction, segmentation, morphology, bubble remover, labeling, features extraction and TIFF Group4 compression is calculated for both SENTIO32 and FPGA platforms. For calculating execution time on SENTIO32 a high signal was sent on one of the output pins of the SENTIO32 when vision processing task get started and then made it low when the task finished. During this operation time stamp was recorded using logic analyzer. The execution time for the operation performed on FPGA is determined by the camera speed at which it is capturing images because all of the hardware modules are running at the camera clock speed. The resolution of the CMOS camera used is 400x640 (400 rows and 640 columns) and the operating frequency is 13.5 MHz. It must be noted that there are 32 black (dead) pixels after each row and each vision task has a latency of Lt clock cycle, so the execution time for all vision tasks i.e. Image capturing, background subtraction, segmentation, morphology, labeling, features extraction and TIFF Group4 compression is calculated using equation (1) .

$$T = (400 \times (640 + 32) + Lt) \times (1/(13.5 \times 10^6)) \quad (1)$$

Time spent on communicating the results to the central base station is calculated as

$$T\_IEEE = (X + 19) * 0.000032 + 0.000192 \quad (2)$$

Where X is the number of bytes transmitted.

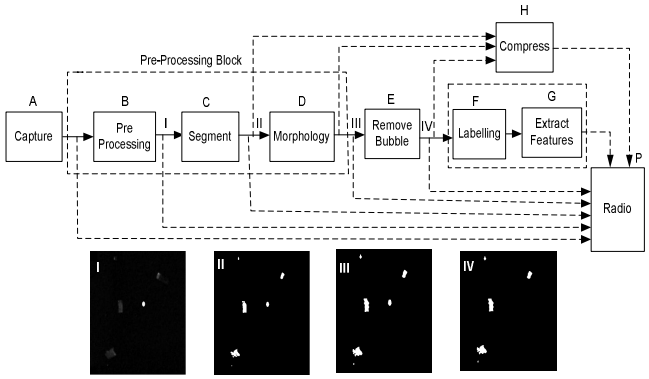

Figure 2. Algorithm flow for all tasks partitioning between hardware and software.

Power consumption of IEEE 802.15.4 is 132mW while that of SENTIO32 is 77.55mW when operating (performing some vision processing operation or communicating the results to the server). The total energy spent on sending data over wireless link is the combination of individual energy consumption of IEEE 802.15.4 and SENTIO32 platforms because both of them are running when data is communicated to the server. The energy consumption of the external flash light used for achieving high enough signal to noise ratio is 0.085mJ, which is included in the energy calculation of the embedded platform for each strategy. Power consumption of the camera is 160 mW and its processing time is 33.33 ms, so its energy consumption for processing one image is 5.3 mJ. Time spent and energy consumed on each individual operation running on SENTIO32 is mentioned in TABLE I. While time spent for performing vision tasks on FPGA is calculated using equation (1), and is used to determine the energy consumption of the modules implemented on the FPGA, mentioned in TABLE II. The power consumption and logic cells required by modules implemented on FPGA are also shown in TABLE II.

Figure 3 shows the life time of the vision sensor node for all possible hardware/software partitioning strategies. The top most curve in this graph represents Strategy36 in Table III, while the second and third top most curves represent Strategy15 and Strategy9 respectively in the same table, which are almost on top of each other. Life time of the vision sensor node is predicted based on the energy requirement of

the embedded platform for the implementation of the vision sensor node.

TABLE I. ENERGY CONSUMPTION OF INDIVIDUAL OPERATION OF THE SOFTWARE IMPLEMENTATION.

Individual Modules	Time(ms)	Energy(mJ)
Subtraction	332.5	25.78
Segmentation	225	17.44
Morphology	2327.1	180.46
Bubble Remover	202.5	15.70
Labeling, Extract Features	1044	202
TIFF compression	345.1	26.76

TABLE II. POWER, ENERGY AND AREA CONSUMED BY MODULE IMPLEMENTED ON FPGA.

Modules ON FPGA	Power(mW)	Energy(mJ)	Logic cells (Spartan 6)	Memory
A	1.44	0.029073	329	0
AB	1.78	0.035842	702	0
ABC	1.91	0.038431	705	0
ABCH	3.35	0.066913	1190	3
ABCD	3.05	0.061197	1388	4
ABCDH	4.49	0.089951	1873	7
ABCDE	3.23	0.064781	1406	5
ABCDEH	4.65	0.093139	1891	8
ABCDEFG	5.93	0.118542	2279	12

IV. DISCUSSION

In current research work, our focus is on finding an optimal point for partitioning vision processing tasks between hardware and software implementation as well as partitioning tasks between sensor node and server. In TABLE III, all hardware/software partitioning strategies are mentioned. TIFF compression of the raw image (after capturing or subtraction) are also possible strategies but as the data that needs to be communicated at these stages is quite high, resulting in high communication cost [7]. Also for strategies 16, 17 and 22 mentioned in Table III, one need to store the whole frame, so all these strategies are not feasible. All strategies in Table III, that produce 32000 bytes (e.g. Strategies 18, 19, 20 etc.) require buffers of 32 Kbytes and are feasible. All the remaining strategies in Table III are feasible because they need very small buffers. In Figure 2, each of the vision processing tasks is symbolized by a capital letter like A, B up to P. These symbolized letters are used in TABLE III, for visualizing all possible hardware/software partitioning at the vision sensor node as well as local and central intelligence partitioning strategies. We explain Strategy3 and Strategy9 as an example here. In Strategy3, modules symbolized by letter A (image capture), B (subtraction) and C (segmentation) are implemented on hardware and then segmented image is compressed (H, compression) and transmitted (P, Radio) using SENTIO32 while the rest of the vision processing tasks are executed on the server. Similarly in Strategy15 tasks symbolized by letter A, B, C, D, E and H are executed on FPGA while compressed image is transmitted to the server using SENTIO32 (module P). The rest of the image processing is performed at the server. It must be noticed here that the power consumption of the embedded platform for strategies 4, 9, 15 and 36 is quite low (almost similar). The reason for this is that SENTIO32 is running for very short time (communication only) and all the vision processing is performed on FPGA. Amount of data that needs to be communicated for each strategy is shown in TABLE III (Data sent) and it is different for different strategies because TIFF Group4 compression produces varying compressed data based on the input image. Also the number of bytes produced for hardware and software implementation is different because of the two different implementation of TIFF Group4 compression.

TABLE III. DIFFERENT MEASURES OF ALL POSSIBLE HARDWARE/SOFTWARE PARTITIONING STRATEGIES.

Strategy	FPGA Tasks	SENTIO32 Tasks	Server Tasks	Energy Embeded platform (mJ)	Data sent (Bytes)	FPGA Logic cells	FPGA BRAM
1	A	BCHP	DEFG	83.76	1218	329	N.A.
2	AB	CHP	DEFG	57.99	1218	702	N.A.
3	ABC	HP	DEFG	40.55	1218	705	N.A.
4	ABCH	P	DEFG	10.21	680	1190	3
5	A	BCDHP	EFG	264.66	1282	329	N.A.
6	AB	CDHP	EFG	238.89	1282	702	N.A.
7	ABC	DHP	EFG	221.45	1282	705	N.A.
8	ABCD	HP	EFG	35.76	1282	1388	4
9	ABCDH	P	EFG	9.03	500	1873	7
10	A	BCDEHP	FG	274.84	458	329	N.A.
11	AB	CDEHP	FG	249.07	458	702	N.A.
12	ABC	DEHP	FG	231.63	458	705	N.A.
13	ABCD	EHP	FG	51.18	458	1388	4
14	ABCDE	HP	FG	35.48	458	1406	5
15	ABCDEH	P	FG	8.07	356	1891	8
16	A	P	BCDEFG	1614.94	256000	329	N.A.
17	A	BP	CDEFG	1640.73	256000	329	N.A.
18	A	BCP	DEFG	250.00	32000	329	N.A.
19	A	BCDP	EFG	430.47	32000	329	N.A.
20	A	BCDEP	FG	446.17	32000	329	N.A.
21	A	BCDEFGP	N.A.	448.23	114	329	N.A.
22	AB	P	CDEFG	1614.96	256000	702	N.A.
23	AB	CP	DEFG	224.24	32000	702	N.A.
24	AB	CDP	EFG	404.70	32000	702	N.A.
25	AB	CDEP	FG	420.41	32000	702	N.A.
26	AB	CDEFGP	N.A.	422.46	114	702	N.A.
27	ABC	P	DEFG	206.79	32000	705	N.A.
28	ABC	DP	EFG	387.26	32000	705	N.A.
29	ABC	DEP	FG	202.56	32000	705	N.A.
30	ABC	DEFGP	N.A.	405.02	114	705	N.A.
31	ABCD	P	EFG	206.81	32000	1388	4
32	ABCD	EP	FG	222.52	32000	1388	4
33	ABCD	EFGP	N.A.	224.57	114	1388	4
34	ABCDE	P	FG	206.82	32000	1406	5
35	ABCDE	FGP	N.A.	208.87	114	1406	5
36	ABCDEFG	P	N.A.	6.47	114	2279	12

If we perform TIFF Group4 compression after segmentation and after morphology on the same image then the size of the resulting compressed image after segmentation will be different from that of morphology because in morphology we remove all one to two pixel false objects, which can change the number of transitions in the image. As TIFF Group4 scheme encodes the changes in the

input image and after morphology the number of changes in the image are different compared to that of segmentation.

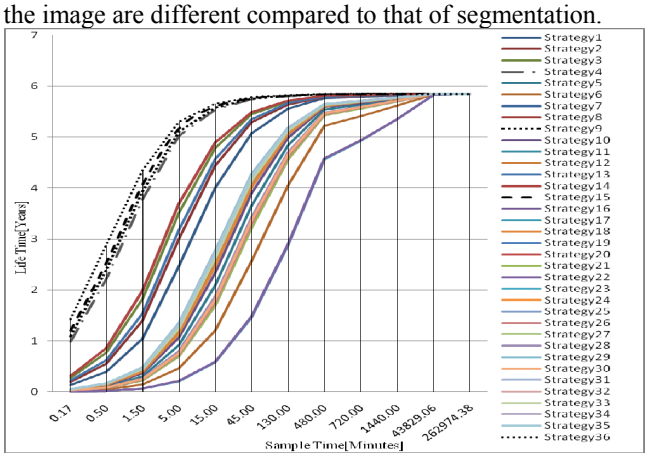

Figure 3. Life time for all possible hardware/software partitioning strategies.

In Strategy9 we perform vision processing operation up to morphology (including morphology) and then transmit the results after TIFF Group4 compression, which can be implemented on Actel IGLOO Low-Power Flash FPGAs (AGL600). In strategies 15 and 36, previous frame needs to be stored for removing bubbles in current frame. Storing a binary frame (640x400) needs 256000 bits memory, not available on our board. For our specific application (based on Strategy9) it must be noticed in Figure 3 that for a sample period of 5 minutes the predicted life time of the vision sensor node is 5.1 years. So Strategy9 is preferable in our specific application. Depending on the requirement and budget of an application, any of the 36 strategies could be implemented except strategies 16, 17 and 22. All 36 strategies have their own pros and cons. E.g. Strategy36 offers lowest power consumption but needs highest design time and hardware cost. The design time and hardware cost for Strategy23 is very low but its energy consumption is quite high. So for any application the specification must carefully be analyzed first and then suitable partitioning strategy should be selected accordingly. In [7] the life time of the vision sensor node was 4.22 years for a sample period of 15 minutes for software implementation. So even for higher sample rate (5 minutes) the life time (5.1 years) of the vision sensor node is more than the lower sample rate (15 minutes) implementation. In TABLE III, one trend must be noted, which is the relationship between design time and the life time of the vision sensor node. All strategies, in which vision processing is performed at FPGA and only communication is performed at SENTIO32, have relative high life time of the vision sensor node. But its disadvantage is quite high design and implementation time. Hence performing more and more tasks on FPGA require sufficient design time and larger FPGA components but result in longer life time of the vision sensor node. On the contrary, if a strategy has less FPGA implementation, while more vision processing in SENTIO32 or server, suffers from limited life time. Its advantage is less design and implementation time.

V. CONCLUSION

While performing more and more tasks on software the energy requirement of the vision sensor node is increased. Hence we will avoid a task partitioning strategy having more modules in software implementation. Similarly, shifting more tasks to hardware results in increased hardware cost, as well as increased design and development time. We have shown that partitioning tasks between hardware and software at the vision sensor node affects the energy requirement of the vision sensor node. Considering this, our results show that the most suitable strategy for our specific application is when we perform vision tasks such as image capturing, background subtraction, segmentation, morphology and TIFF Group4 compression on FPGA and then send the results using transceiver embedded in SENTIO32 platform. The bubble remover, labeling and features extraction is performed at the central base station. In this way the power requirement of the vision sensor node is reduced, which resulted a life time of 5.1 years of the vision sensor node for a sample rate of 5 minutes. A general conclusion of our results is that the highest system life time is achieved for full FPGA solution for the processing steps. However this system requires FPGA technology with large memory resources and low sleep power or low configuration energy.

REFERENCES

[1] L. Ferrigno, S. Marano, V. Paciello, A.Pietrosanto. “Balancing computational and transmission power consumption in wireless image sensor networks”. IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems, Italy, July 2005.

[2] L. Fredrik, C. Peng, B. Oelmann. ” Analysis of the IEEE 802.15.4 standard for a wireless closed loop control system for heavy duty cranes”. IEEE Second International Symposium on Industrial Embedded Systems - SIES'2007.

[3] W. J. MacLean. “An Evaluation of the Suitability of FPGAs for Embedded Vision Systems”. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[4] A. Rowe, D . Goel, R . Rajkumar. “FireFly Mosaic, A vision-enabled wireless sensor networking system”. Proceedings of the 28th IEEE International Real-Time Systems Symposium. Pages: 459-468 2007, ISSN:1052-8725.

[5] M. Sen, I . Corretjer, T . Lv, S.S.Bhattacharyya, F. Haim, W. Wolf, S . Saha, S. Jason. “Dataflow-Based Mapping of Computer Vision Algorithms onto FPGAs”. Hindawi Publishing Corporation EURASIP Journal on Embedded systems Vol. 2007.

[6] P. Kulkarni, D. Ganesan, P. Shenoy, Q. Lu. ”SensEye, A multi-tier camera sensor network”. International Multimedia Conference archive Proceedings of the 13th annual ACM international conference on Multimedia Pages: 229 – 238.

[7] K. Khursheed, M. Imran, M. O’ Nills, N. Lawal. “Exploration of Local and Central Processing for a Wireless Camera Based Sensor Node”. Proceedings of IEEE International Conference on Signal & Electronic System, Gliwice, Poland, Sept. 7-10, 2010.

[8] J. M. Rabaey, M. J. Hammer, J.L. Da Silva, “Picoradio supports ad-hoc ultra low-power wireless networking”. IEEE Computer society Vol 33 (7), 42 (2000).

[9] C.H. Zhiyong, ,L. Y.Pan, Z. Zeng, M.Q.-H Meng. “A Novel FPGA-Based Wireless Vision Sensor Node”. Proceedings of the IEEE International Conference on Automation and Logistics Shenyang, China August 2009.

[10] A. C. Bianchi, A.H. Reali-Costa. “Implementing Computer Vision Algorithms in hardware: An FPGA/VHDL-Based Vision System for

a Mobile Robot. RoboCup 2001, LNAI 2377, pp. 281–286, 2002.Springer-Verlag Berlin Heidelberg 2002.

[11] P. Curran, J. Buckley, B. O'Flynn.,X. Li, J. Zhou, Lacey,G., S. C. O'Mathuna. "VAMP, A Vision Based Sensor Network for Health Care Hygiene". 14th nordic-baltic conference on biomedical engineering and medical physics, IFMBE Proceedings 2008, Vol. 20.

[12] N. Lawal, B. Thörnberg, M. O'Nils. "Power-aware automatic constraint generation for FPGA based real-time video processing systems". Proceedings of IEEE Norchip Conference, 2007, pp. 124-128, 2007.

[13] H. Norell, N. Lawal, M. O'Nils, "Automatic Generation of Spatial and Temporal Memory Architectures for Embedded Video Processing Systems". Eurasip Journal on Embedded Systems, 2007.

[14] B. Thörnberg, M. Palkovic, Q. Hu, L. Olsson, P.G. Kjeldsberg., M. O'Nils, M.F. Catthor. "Bit-Width Constrained Memory Hierarchy Optimization for Real-Time Video Systems". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 26: 4, pp. 781-800, 2007.

[15] L. Shang, A.S. Kaviani, K. Bathala. "Dynamic Power Consumption in Virtex-II FPGA Family". Proceedings of the 2002 ACM/SIGDA 10th International Symposium on Field-Programmable Gate Arrays, pages 157– 164. ACM Press, 2002.

[16] http://embedded.eecs.berkeley.edu/research/hsc/

[17] Z. Yiguo, L. Wenjian, Z. Zeming, L. Bin, W. Xufa. "A hardware/software partitioning algorithm based on artificial immune principles". Science direct, Applied Soft Computing 8 (2008) 383–391.

[18] M. Imran., K. Khursheed, M. O' Nills, N. Lawal. "Exploration of Target Architecture for a Wireless Camera Based Sensor Node". 28th Norchip Conference 15- November 2010, Tampere, Finland.

[19] *http://www.actel.com/*

[20] *http://www.xilinx.com/*