Analysis of Vision systems and Taxonomy Formulation

An abstract model for generalization

Muhammad Imran

Electronics Design Division, in the
Department of Information Technology and Media
Mid Sweden University, SE-851 70 Sundsvall, Sweden

Research report in electronics, No. 1

This report is part of the research work which is conducted in order to develop a benchmark or an abstract model for the vision systems. This work leads to system taxonomy for vision systems which can be used as a reference model for classification and comparison of vision systems. Moreover, it will facilitate in development of generic solutions in vision systems.

Analysis of Vision systems and Taxonomy Formulation

An abstract model for generalization

Research report in electronics, No. 1

ISBN 978-91-86694-79-1

Electronics Design Division, in the
Department of Information Technology and Media
Mid Sweden University, SE-851 70 Sundsvall
Sweden

Telephone: +46 (0)60 148561
Email : muhammad.imran@miun.se

ABSTRACT

Vision systems are increasingly used in many applications including optical character recognition, mechanical inspection, automotive safety, surveillance and traffic monitoring. The current trend in vision systems is to propose solutions for specific problems as each application has different requirements and constraints. There is no generalized model or benchmark, to the best of our knowledge, which can be used for providing generic solutions for different class of vision systems. Providing a generic model in vision systems is a challenging task due to number of influencing factors. However, common characteristic can be identified in order to propose an abstract model. The majority of vision applications focus on the detection, analysis and recognition of objects. These tasks are reduced to vision functions which can be used to characterize the vision systems. In this report, we have analysed different types of vision systems, both wire and wireless, individual vision systems as well as a vision node in a Wireless Vision Sensor Network (WVSN). This analysis leads to the development of a system taxonomy, in which vision functions are considered as characteristics of the systems. The taxonomy is evaluated by using a quantitative parameter which shows that it covers 95 percent of the investigated vision systems and its flow is ordered for 50 percent of the systems. The proposed taxonomy will assist designers to classify their systems and enable researchers to compare their results with a similar class of systems. Moreover, it will help designers/researchers to propose generic architectures for different class of vision systems.

ACKNOWLEDGEMENTS

I am thankful to Professor Mattias O'Nils for his guidance, support and motivation during this work. Dr. Najeem Lawal is acknowledged for useful discussion during this work. Muhammad Hanif, PhD student, Australian National University is acknowledged for his contribution and discussion in relation to the filtering part. I am thankful to Khursheed and Naeem Ahmad for reviewing this article.

Moreover, I am thankful to my colleagues Malik Abdul Waheed and Mohammad Anzar Alam for discussions during this work. Fiona wait is greatly acknowledged for proof reading of this article.

I would like to thank STC@miun and Higher Education Commission (HEC), Pakistan for financial support.

Sundsvall, October 2011

Muhammad Imran

TABLE OF CONTENTS

1 INTRODUCTION

Vision systems can be implemented either on conventional, high performance systems or on embedded platforms. The conventional systems often have no limitations in terms of resources and they, additionally, offer large design and implementation flexibility but, such systems are difficult to deploy for many applications and often requires high installation and maintenance costs. In addition to this, they provide low real time performance. The vision solutions on embedded systems are able to provide both ease of installation and maintenance. Moreover, the embedded solutions are low cost, low power and provide real time performance. Therefore, developing image processing systems on an embedded platform can reduce the production maintenance costs and improve the reliability. Implementing vision systems on an embedded platform requires consideration to be given to different design and implementation challenges. There is an increasing demand in the vision systems to provide a benchmark or at least an abstract model which can be used to classify, analyse and compare the results of different vision systems. Taxonomy building is a starting point and is a step in the right direction and it is based on vision functions being performed by the systems.

Many of the vision functions are common in the smart cameras which have been developed for a number of applications. The majority of smart camera applications target the presence of objects in the field of view, their spatio-temporal movement, or in case of humans or animals, their gesture recognition. Thus, the vision tasks to be performed are reduced to object detection, localization, tracking and classification. The common vision tasks in achieving these tasks consist of background subtraction, filtering, morphological, foreground segmentation, operations and additional foreground processing steps [1]. The major challenge is to perform these tasks with the minimum required complexity, processing resources, bandwidth resources and energy consumption so that processing can be performed on low-cost, battery operated and embedded platforms.

Many different proposals exist to provide solutions in relation to particular vision systems. Generally, the emphasis is narrowed down to a particular solution after considering many alternatives. For example, in some problems, the emphasis is to have systems which are event driven and the functions used involve linear filters and mathematical transforms to achieve the required results while in others, the emphasis is to have event driven systems, and the functions used in these cases are non-linear filters and segmentation. There could be many different classes of vision systems based on these preferences. It is difficult to generalize the flow of vision processing for different applications as each vision problem has different requirements, but, some common characteristics of vision systems can be identified, with vision functions being considered as a main influencing parameter. As an initial step in order to generalize the existing vision approaches, different vision systems can be analysed to find common characteristics. An abstract model is required, which can be used as a reference for different vision systems. When architecture is proposed for a vision system, it can be evaluated and compared with the proposed model. This will enable the designer/researchers to conclude that their architecture is suitable for a particular class of systems and is indeed suitable for a particular problem. This approach may not be final or exhaustive but, it will be a starting point towards the generalization of a solution in vision systems.

To the best of our knowledge, there is no work, which provides the general characteristics of smart cameras. In [37], the authors classify the smart camera research based on platform capabilities, the degree of distributed processing as well as system autonomy. We have analysed many different embedded vision systems for problem formulation but our focus is on Wireless Vision Sensor Network (WVSNs) in which the vision processing is performed on an embedded platform. The image sensor produces 2 dimensional data which places strict requirements on the processing, power and bandwidth because there are limited resources on an embedded platform. In some applications the system would require transmission of data over a limited wireless link and the system is powered by batteries or through alternative energy sources. These challenges can be handled by efficient design and implementation techniques and by investigating the existing vision systems. Wireless Vision Sensor Network (WVSNs) is an area of research in which vision solutions are usually provided on embedded platforms. The applications for WVSNs include, industrial monitoring [10], surveillance [6][24][28][32], recognition[5], environmental monitoring [23], virtual reality [4] and personal care[11]. The technological development has enabled the vision systems to perform more in-node processing in cost effective and energy efficient ways. There has been a tremendous rise in research within the field of wireless sensor networks during the last decade and wireless sensor platforms, capable for commercial sensors have been developed for monitoring and automation. However the field of wireless sensor networks is still immature and there is only a small amount of work which provides generic solutions. Different platforms are developed for WVSN nodes in order to meet the demands of different applications which have caused long and uneven development and adaptation periods [40].

Our aim is to define system taxonomy for vision systems which can be used so as to classify them. We have analysed different vision systems and have abstracted the vision functions which have then be used to develop system taxonomy. We have evaluated the proposed taxonomy with the existing vision systems and shown how a system can be classified according to the proposed taxonomy. This study will assist in classifying and analysing different vision systems based on the functions they perform. Moreover, it will provide the motivation to produce generic solutions for different classes of vision systems as many of these systems do possess common characteristics.

There are many different vision systems in existence and some of those implemented on embedded systems will be described, in particular those developed for WVSNs. The functionalities of these systems are shown in Table 1. By observing these systems, it is possible to conclude that each one is different from an implementation point of view but their functions are, in general, similar. Thus, it is necessary to have a generic model or benchmark which can be used to classify and compare these systems. Wireless visual sensor networks have been designed and implemented on micro-controllers, microprocessors and on hardware platforms. In [11] a lightweight camera sensor network approach is proposed, which operates on symbolic information rather than on images. The work of the authors is in relation to the development of biological-inspired image sensors which are specialized in obtaining useful information with regards to the event. This will reduce the processing, bandwidth and power supply requirements. Additionally the output is in the form of an address event stream which will solve the privacy problems that cause people to feel uncomfortable. Instead of sending raw data, the node sends symbols that summarize the motion activity.

In [41] a multi-tier architecture called SensEye is proposed for wireless nodes. SensEye is a multi-tier of heterogeneous wireless nodes and cameras which aims at low power, low latency detection and low latency wakeup. In this approach low power elements are used to wakeup high power elements. Resource-constraint, low power sensors are used to perform simple tasks while high power sensors conduct more complex tasks. For example, during motion detection, a PID (passive infrared detector) can be used to monitor the area for the majority of the time, which consumes less energy. When an object is detected, it will trigger the high resolution camera to capture a more detailed picture of the scene. This approach will best utilize the energy consumption of the system as simple tasks would consume less energy.

The Stanford's MeshEye Sensor node [42] uses two kilopixel imagers for low resolution images and one high resolution camera module for detailed object snapshots. One of the kilopixel imagers constantly monitors objects entering the field of view. When an object is detected, the second low resolution image sensor is activated and it computes the location and size of the object, based on stereo vision. Subsequently, a high resolution camera is triggered to capture a high resolution grey or coloured region of interest including only the detected object. It is claimed that Stanford's MeshEye

Mote quantifies the reduction in energy consumption through the usage of a hybrid-resolution vision system. Energy consumption can be reduced only when there is no object since the high resolution camera will be in sleep mode. Energy consumption in a visual sensor network varies for different applications and for different scenarios. The hybrid vision is computationally more efficient and consumes less energy for smaller objects within its field of view [43]. The hybrid vision system would suffer from calibration difficulties as variations in the optics and alignment in image sensors would cause problems in object detection in the initial phase. CMUcam3 [44] was presented by Carnegie Mellon University in 2007. It was a low-cost, open source embedded vision platform which consists of a colour CMOS camera, a 1 MB frame buffer and a 32 bit ARM7TDMI (NXP LPC2106 with 60 MHZ processor, 64 KiB of RAM and 128 KiB of flash) micro-controller. The authors developed their own optimized C library. The bottleneck in the CMUcam3 is based on the limited RAM and reduced computation speed, which is not sufficient for complex vision tasks.

In [15], the authors used a bio-inspired approach in which image processing is performed following a hierarchical approach using two levels. The first level of processing tasks with a regular computational flow is on an array of moderately accurate processing elements. The second processing level, where the amount of data is smaller, is implemented on a 32 bit NIOS-II, RISC processor. The system can deliver not only simple signals, but any information extracted from the image and processed. Wireless Autonomous Sensor Prototype, WASP [45] consists of a 600 MHz general purpose processor, 128 MB RAM and 32 MB flash together with a Wi-Fi module and camera module. They have an integrated software framework which includes low level libraries for interfacing with device drivers and high level libraries for video processing. The authors did not select FPGAs or DSPs and argued that a general purpose processor would facilitate both software development and maintenance on the node. Moreover, the Wi-Fi has more power consumption compared to other communication standards i.e. IEEE 802.15.4 and Bluetooth.

DSPcam [47] consists of a Blackfin Processor with 32MB SDRAM and 4MB Flash, CMOS image sensor and an 802.11 b/g communication module which communicates over multiple hops to form a mesh network. An open source image processing library called Camellia is ported to DSPcam. It also integrates with Firefly which uses the IEEE 802.15.4 based wireless communication. Each camera node performs local processing in order to detect the event and will annotate the video stream for the operator as the information is sent to the single point, operator observation station in the network.

An FPGA based vision sensor node has small power consumption in comparison to micro-controller/micro-processor implemented systems. FPGAs provide parallel task execution, which is good for low level image processing tasks having inherent parallelism. FPGA allows for the implementation of several custom components, memory and processing cores, with the same device. Moreover, multiple clock domains are supported which enables separate clocks for different tasks [36][46].

In [46], the design principles for the video node are presented in the context of a long-lifetime. The authors have developed a prototype to validate the approach. In the system, a low power flash FPGA and a CMOS camera are used. The imager used is a 128x64-pixel binary contrast based sensor. However, for the majority of applications a binary image sensor is used and this small resolution would not be feasible.

In [13] the authors presented a low cost, low power and reconfigurable architecture for wireless VSNs. They used a system on a programmable chip which includes an FPGA and Nios II micro-controller unit. An external SRAM is added to provide the necessary memory resources for computation. For wireless communication, a Nordic 2.4 GHz nRF24L01 transceiver is used, which has a sample interface to the Nios II MCU. The FPGA acquires the images from a CMOS camera and performs a JPEG image compression scheme on the data while the Nios II is used to configure the internal registers of the nRF24L01 transceiver. It also manages the system operation performed on the FPGA.

In [48] an image sensor node is presented which includes SAMSUNG's S3C44BOX embedded processor, a memory, a CMOS image sensor, an FPGA, Chipcon's CC1000 as the RF module and a power unit. The FPGA is used for the implementation of vision tasks such as image acquisition, compression and processing circuits, however, no experimental verification was provided with regards to the power consumption reduction in the presence of an embedded processor.

The authors in [12] have proposed a wireless video node, MicrelEye for cooperative video processing applications. The node has a low-cost VGA CMOS image sensor, a reconfigurable processing platform (FPGA) together with a micro-controller and SRAM. For wireless communication a Bluetooth is used but for many applications this is not a suitable choice for communication due to its shorter range and high power consumption.

In [54], the author proposed the use of stereo vision algorithms for robotic applications. Vision sensors are important for robots as they interact physically with the environment and thus are able to carefully plan their actions. Navigation is an important part of mobile robots and it involves image tasks such as capturing, colour enhancement, distortion correction, feature extraction, pose estimation etc. In vision based navigation, a set of images from at least two different viewpoints are required in order to find features for estimating and constructing landmarks.

2 PROBLEM DESCRIPTION

There is an increasing demand to provide a benchmark, or, at least an abstract model, which can be used to classify and compare the results of different vision systems. In order to develop this, our approach is to analyse the existing vision systems and identify the problem space. We have surveyed a number of vision systems as is shown in Table 1. By investigating recently published systems, it is evident that there is an opportunity to provide an abstract model for the classification of vision systems. In order to achieve this, we aim to develop a system taxonomy which is based on the vision functions. The mechanism we have adopted to develop this system taxonomy is shown in Figure 1 in which the problem space is to identify the common characteristics associated with different vision systems.

The conclusion to be drawn based on an investigation of a number of vision systems was that vision functions can be used to characterize the vision systems. Similar vision functions have been grouped together to make an abstraction of the vision functions. These abstracted vision functions are then used to propose system taxonomy. Taxonomy building is an iterative process and thus a repetitive investigation must be made of the systems in order to ensure that the taxonomy covers all the different types of systems. This is represented by an arrow from the system taxonomy to the vision systems. By investigating existing systems and then comparing them with the system taxonomy, improvements can be identified and relevant changes can be made in relation to both the abstraction level and to the system taxonomy. In this work, this iterative approach has been adopted for the development of the system taxonomy which is based on system functionality rather than system implementation and complexity.

This work will be extended to propose implementation taxonomy. In addition to this architecture, one class of vision system will be presented and the relevant design/research challenges associated with this class will be investigated.

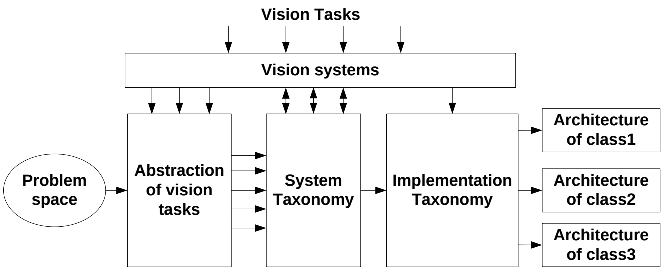


Figure 1. Problem flow

3 VISION SYSTEMS

These are the different vision systems used in the building of the taxonomy.

Table 1. Vision systems, application and description of functionality

number	systems	Application	description
1	Wireless smart camera [10]	Particle detection in fluid	Measuring particles in fluid and sending feature/compressed binary data to server.
2	Lightweight camera[11]	Assisted living and helping the elderly	Instead of giving an image as output, the output involves symbols that summarize motion activity.
3	MicrelEye [12]	Distributed object detection	Binary detection flag after people detection or alternatively the whole image containing the recognized object can be sent.
4	FPGA based WVSN [13]	JPEG compression	After acquiring an image, JPEG compressed data is transmitted.
5	Pervasive smart camera architecture[14]	multi-camera person classification	The system has a multi-camera person classification application. Images are acquired and after object detection, features are extracted and then a soft decision in relation to a person is generated by each camera. The soft decisions are fused at a single node to come to a final classification result.
6	Prototype node for WVSN[15]	surveillance and monitoring	The image after acquisition is segmented and motion activity is detected. An alarm is triggered, The coordinates of the detected events in the scene, or analyse the characteristics of the detected changes and send this information over the network.
7	CITRIC[63]	image compression, target tracking, and camera localization (we used target tracking only)	The system sends post-processed data (for instance, low-dimensional features from an image) in real-time back to the centralized server and clients for further processing. If an event of interest occurs in the network, a query can be sent for the relevant image sequence to be compressed and sent back to the server over a slightly longer period of time.
8	Cyclops [52]	Object detection and Hand Posture Recognition	It can be used to disambiguate ambient conditions or detect motion in security or surveillance applications. Moving Objects are

			detected using a sequence of images. The system also uses hand posture recognition as application.
9	Ultra-low power[46]	People counting	The authors used a binary contrast-based sensor capable of extracting certain visual features and detecting significant motion in the sense. Flash based FPGA is used for system control and data processing.
10	Mesheye [42]	distributed intelligent surveillance	The system captures the raw data, objects are detected and then objects of interest are forwarded. The cameras node can collaborate and transmit the textual description along with snapshot.
11	Foreground detection using feedback [58]	Foreground detection and tracking	The authors presented a light weight and resource efficient foreground object detection and tracking algorithm which is suitable for embedded applications. The proposed algorithm used feedback from the tracking stage of the algorithm in the object detection instead of performing foreground detection and tracking independently.
12	Novel vehicle detect vision system[28]	vehicle detect	The authors developed a vehicle detection system based on a line scan camera. The vehicle's outline region in the captured image is extracted with a wavelet transform algorithm, and the vehicle's speed is estimated by a curve correlation method.
13	Visual Measurement Using a High-speed Line-Scan Camera[30]	visual measurement system for observing pile rebound and penetration movement	This work proposes a visual measurement system for pile rebound and penetration movement including vibration using a high-speed line-scan camera and a specially designed mark to recognize two-dimensional motion parameters of the mark using only a line-scan camera.
14	Vision system Using Line Scan Camera[64]	Street-Parking Vehicle Detection	The authors proposed a detection method for street-parking vehicles by using a line-scan camera.
15	WASP[45]	Human posture classification	Images are acquired, target is detected and then segmentation is performed followed by feature extraction where the features used in this experiment are the mean and variance of the projection of a target blob on the image x and y-axes. After this, posture classification is performed where a KNN classifier is used to differentiate between multiple postures.
16	VAMP[65]	Health care	The vision system captures and processes images locally in real-time, determines if hand-washing procedures have been correctly undertaken and then passes the resulting high-level data over a low-bandwidth wireless link.
17	SensEye[41]	Two applications, namely monitoring of rare	SensEye is a multi-tier camera system. In this surveillance focuses on three tasks, object detection, recognition and tracking.

		species in remote forests and ad–hoc surveillance in disaster management tasks.	
18	Gesture cam[5]	head and hand gesture recognition	GestureCam, is FPGA-based smart camera that aims to classify and recognize simple head and hand gestures.
19	Real time human pose reconstruction WVSN [51]	Real time human pose reconstruction	The vision system targets 3D human pose reconstruction. From the live image data provided by the sensor the smart camera acquires critical joints of the subject in the scene through local processing. The results obtained by multiple smart cameras are then transmitted through the wireless channel to a central PC where the 3D pose is recovered and demonstrated in a virtual reality gaming application.
20	Smart cameras as embedded systems[66]	human gesture recognition	The authors chose to focus initially on human gesture recognition identifying whether a subject is walking, standing or waving his/her arms.
21	DSPcam[47]	surveillance	DSPcam has local processing Capabilities and can detect the event unfolding in its field of view and appropriately annotate the video for the operator. It describes key elements of the visual data being transmitted.
22	Firefly Mosaic[53]	Elderly care monitoring	The system monitors people’s daily activities in the home using 8 cameras. The occupant's activities are discovered and clustered using multiple cameras with overlapping views.
23	Stero vision in robotics application [54]	Robotic navigation	The authors discuss different vision based robotic algorithms. In one work they used Stephen and Harris combined corner and an edge detection algorithm for real time feature extraction.
24	Real time vision feedback system[56]	Automation of nano-assembly manipulator	The work proposes the use of vision feedback control to automate the well-known Zyvex nano-manipulator inside a scanning electron microscope (SEM).
25	CMUCAM3[44]	Various applications	The CMucam3 is an open source, embedded colour computer vision platform. The systems discuss examples of colour tracking and face detection.

4 INVESTIGATION OF DIFFERENT VISION SYSTEMS

At this point, different vision tasks have been listed and an investigation made as to which of the different systems are using these tasks.

Vision systems	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
names	Particle detection	lightweight camera	MicrelEye	Novel FPGA	multip camera object	prototype node for WVSN	CITRIC	Cyclops	ultra low power	mesheye	ataptive object detection tr acking	vehicle detect line scan	pile pentration lin scan	line scan street parking	WASP	VAMP	Sens eye	Gesturcam	human pose WVSN	smart cams	DSPcam	Firefly	stereo vision in robots	Real time vision feedback system	CMUcam3	target tracking feedback
Characteristics of vision																										
Image sources																										
• Camera																										
a. Line scan												yes	yes	yes												
b. Area scan	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes				yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
• Neighbor sensor					yes												yes									
Image capturing																										
• Binary capturing									yes																	yes
• gray capturing								yes		yes	yes	yes	yes	yes												
• Color capturing	yes	yes	yes	yes	yes	yes	yes			yes					yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	
Interpolation																		yes								
Frame/background storage							yes				yes										yes					yes
• No storage																								yes		
• Initial storage	yes																									
• Continuous storage		yes	yes	yes	yes	yes	yes	yes	yes	yes	yes						yes				yes				yes	yes
Frame/Background Subtraction	yes	yes (t	yes	No	yes	yes	yes	yes	yes	yes		yes			yes		yes					yes			yes	
Segmentation																										
• Thresholding	yes	yes			yes	yes	yes	yes	yes	yes	yes		yes	yes	yes		yes(col	yes	color seg(yes)		yes			yes	yes	
• Region based segmentation																										
• Segment using morphological watersheds																										
• The use of motion in segmentation																										
Filtering																										
• Convolution		yes																				yes			yes	
• Averaging filter	yes									low pass filter (yes)					yes		yes(col	yes	yes		yes				yes	
• Gaussian filter															yes								yes			
• Edge detection		yes							yes				yes	yes				yes		yes			yes	yes	yes	
Order statistic filter																										
• Median filter					yes		yes												yes							
• Max/min filter																										
• Midpoint filter																										
• Alpha-trimmed filter																										
Morphology																										
• Binary morphology	yes														yes									yes		yes
• Gray scale morphology					yes																					
Intensity transformation																										
• Linear transformation																										
a. Negative																										
b. Identity transform																										
• Logarithmic transformation																										
a. Log transformation																										
b. Inverse-log transformation																										
• Power (gamma) law																										
a. *n* th power law																										
b. *n* th root transformation																										
• Piecewise-linear transformation																										
a. Contrast stretching																										
b. Intensity level slicing																										
c. Bit-plane slicing																										
• Histogram processing					yes			yes		yes	yes		yes	yes								yes		yes	yes	
Color and grayscale/value Transformation																										
• Color transform																										
a. Color to color (RGB to HSI)				yes												yes					yes	yes	yes		yes	
b. Color to gray	yes	yes					yes								yes			yes						yes		
c. Gray to binary	yes									yes		yes	yes	yes	yes							yes		yes	yes	
• Grayscale/value transform																										
1. Spatial transform																										
i. Matrix transform																	yes									
ii. Image rotate																	yes									
iii. Image scaling																										
iv. Image registration																										
2. Mathematical transform																										
i. DCT				yes											yes						yes	yes			yes	
ii. FFT																										
iii. Hough transform														yes												
iv. Wavelet transform												yes														
v. Hartley transform																										
Post processing																										
• Labeling	yes		yes		yes				yes						yes		yes				yes	yes		yes	yes	yes
• Feature extraction	yes	yes	yes		yes	yes								yes	yes		yes	yes	yes				yes	yes		yes
• Classification	yes	yes	yes		yes				yes	yes					yes	yes		yes		yes					yes	
• Neural networks																										
• Tracking		yes				yes	yes				yes		yes				yes		yes			e			yes	
• Recognition			yes												yes		yes		yes	yes						
• blob search					yes		yes	yes		yes					yes			yes			yes	yes			yes	
• triangulation																							yes			
Output data type																										
• Scalars		yes																								
• Vector	yes		yes	b	yes	a	c		yes	d					f			yes	centroid				yes			yes
• Matrix																					yes					
• Flag																										
feedback of the system											yes													yes		yes
Training of the system		traini	training		yes		1	2	yes						training		g	training	learning		h	training			train	

a	features
b	compressed data
c	coordinates of bounding box
d	textual descip+ snapshot
e	color tracking
f	features,binary blobs and/or raw image
g	neural network learning
h	learning to update background after illumination changes
1	there are three applications, image compression, target tracking and camera localization but we take target tracking
2	there are two appliactions, object detection and hand posture recogntion but we take object detection.

5 CHARACTERISTICS OF VISION SYSTEMS

A vision system can be classified according to the following characteristics, which may not be exhaustive.

1.1 PROCESSING IN VISION SYSTEMS

Vision systems can be divided into three categories based on how often the system is triggered to start processing the vision functions and transmit it to the user [50]. Three categories are

1.1.1 Time driven

1.1.2 Event driven

1.1.3 Query driven

In a time driven system, the vision system periodically sends data to the server or to other nodes, in the case of the WVSN, in an event driven system, the vision system forwards alerts to the user when an event occurs and in a query driven system the vision system collects data only after receipt of a query from the user.

1.2 IMAGE SOURCES

1.2.1 Cameras

Vision systems acquire images through various image sources such as cameras or neighbouring sensors. The image acquisition device plays an important role in the system's performance. The image quality, resolution, frame/pixel rate addressing modes, integration easiness and logic required for acquisition are important factors which must be considered while selecting an image sensor. The cameras are categorized as either an area array or a line scan array.

Image information is transferred out through a readout register. Each type of sensor includes a readout register located just below the active pixel array. Transferring the image information out of the sensor takes time and is dependent upon the camera resolution and the internal timing signals.

- **Area scan camera**

The imaging device in an area scan camera is arranged in a matrix form so that it captures two dimensional images within one frame in a single exposure [29]. These are acceptable for still or slow moving objects but these is arrays are not suitable for high speed moving objects [26]. In an area scan camera array, the image information is moved down by one row in a parallel fashion. The lowest row is moved to the readout register first. From the row, each pixel is transferred one at a time until all the pixels have been moved out. As one row is finished, the remaining rows are moved down by one row so that the next lowest row is moved to the readout register. This processing continues until the pixels in all the rows have been transferred out. The mechanism in relation to the area scan is shown in Figure 2.

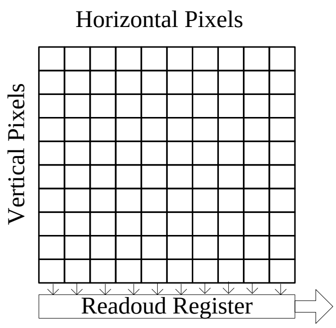


Figure 2. Area scan mechanism

- **Line scan camera**

In a line scan camera, the imaging device consists of contiguous, precisely mounted elements arranged in a single row. For capturing two dimensional images, a line scan camera can either be swept across the scene or it can remain stationary while the scene moves [29]. Line scan cameras send only a single line of pixels per exposure and therefore require less time to transfer the image information into the readout register than is the case for the area array cameras. As the readout is in process, the next exposure period occurs in parallel. An important advantage of the line scan camera is that it can expose new images while the previous image is still reading out its pixels. Line scan sensors are good for high speed objects [26]. In [27], the authors used a line scan camera for a system that can acquire panoramic images quickly for mobile robots. Mobile robots often require panoramic views in order to perform navigational tasks such as identifying landmarks, localizing within the environment and determining free paths in which they move. In [28] , the authors claimed that a vehicle detection algorithm has a lower complexity when using a scan line camera and provides a high performance. The scan line camera detects only moving objects and the background is only a single line and hardly changes. In [30], a visual measurement system for pile penetration and rebound movement including vibration is proposed by using a high-speed line-scan camera. The mechanism for this line scan is shown in Figure 3.

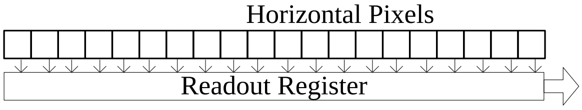


Figure 3. Line scan mechanism

1.2.2 Neighbouring sensors

In some applications, sensor nodes collaborate with other nodes in the network in order to improve both the performance and reliability. The neighbouring sensors can exchange information with each other for some specific applications. The image can be transmitted using a wireless link or a wired link.

1.3 IMAGE CAPTURING

Image sensors producing high quality images impose a high demand on both the processing and communication energy consumption as the processing and transmission times will be high for a large amount of data. Depending on the application, different camera sensors can be employed for image capturing as the conversion from one image format to another would require additional resources. Standard imagers are not necessarily the optimal solution for vision systems. In some applications only particular portions of an image or features are of interest. This has been behind the motivation for using imagers which acquire data of interest. In this way energy will be saved during the acquisition and processing phases [46]. In [3], grey scale data is required for processing but the camera sensor produces data in colour format. The size of the colour image is 37K bytes which cannot be stored due to limited memory. The colour image is converted to a grey scale thus resulting in 15K bytes, which is then transmitted to the central station. Images can be captured using various formats such as binary, grey or in colour.

1.3.1 Binary capturing

Binary images are processed with very fast logical operators. The powerful operation including bar-coding, fingerprint, texture recognition, pattern recognition and morphological filtering can be performed using binary image processing [2]. In [46], the authors used a 128x64-pixel binary contrast based image sensor. The power consumption for this specific application is small as there is no requirement to convert form grey or colour to binary.

1.3.2 Grey scale capturing

It is preferable for some applications to capture the data in a grey scale format. The information content in a remote sensing image for a particular application depends on the grey level resolution of the image. The number of grey levels required to digitize an image for visual quality depends upon the characteristics of the human eye. In general, for remote sensing applications, 256 grey levels (8 bits image) are used. However using 16 grey levels (4 bits image) would require half of the memory as compared to 256 grey level images. It must be noted that the information loss is exponential with respect to the number of grey levels [3]. Upon capturing the grey level images, the system performs other vision tasks such as segmentation and filtering on the data in order to extract information from the image.

1.3.3 Colour capturing

Data is required in colour format, in which each pixel is represented in different colour channels for some applications. Colour image processing is categorized into two areas, Full colour processing or pseudo colour processing. In the first category, the images are acquired with a full colour sensor. Full-colour image processing can be divided into two categories. In the first case, each colour channel is processed individually. In the second category, processing is performed on colour pixels directly in which each pixel has at least three components. Therefore colour pixels are referred to as vectors. In pseudo colour image processing, colours are assigned to grey values in a grey scale image based on specific criteria. The purpose of pseudo colour is for human visualization and the interpretation of grey scale events in an image or sequence of images. Until recently, most digital image processing was performed at the pseudo colour level, however, with the advancement of colour sensors and a processing platform, it is possible to perform colour image processing for a variety of applications [16]. In [4], pedestrians are tracked using colour-based appearance models. In [5], RGB colour based images are used for skin colour classification and gesture recognition. In [6], the authors used the colour based particles filter to track multi objects. In [19], the authors proposed a detection method which combines colour and motion cues and used the RGB colour space for skin classification. There are some image sensors which can be configured to work in RGB, YCrCb and HSI. The majority of image sensors produce a Bayer pattern which is converted into the primary colour components Red, Green and Blue (RGB) using interpolation methods [20].In an RGB wide wavelength light spectrum the intensity is represented on all three components resulting in information redundancy. The YCbCr colour model is another way of encoding the RGB colour model such that the intensity and colour information become less redundant. The Hue, Saturation, and Intensity (HSI) are spectral components that have a more natural and intuitive interpretation [21].

1.4 TRAINING OF THE SYSTEM

There is a new trend in WVSN in which the system is enabled to learn about the phenomenon in the environment rather than simply making measurements and reporting the single effect of interest. There has been an increased interest in enabling WVSN to perform processing according to the situation. For some applications, the system must be trained so that it is adapted to the environment, while, for some applications, the system does not require to be trained. In the former, the systems would learn about the behaviour of the phenomenon in the environment instead of merely processing and reporting the event. This trend will enable the vision network to have more complex forms of awareness about the situational state or the context of the events [39]. The applications in which the training of the system is required, requires storage in relation to the history of events, templates or even the whole frame. In [51] the authors adapt the system to the person's appearance. In order to achieve this they first detect the face, and then look for the torso to learn the colour of the clothes. In [11] a hierarchical inference engine is used which aims to avoid exhaustive training of the entire sensor network for all possible behaviours and all possible instances of behaviours. The authors confined the required training to a low level of hierarchy, which can provide behaviour independent training. The Cyclops [52] uses feature vectors of a training set of gestures for performing pattern recognition. In [53], the authors used a training period immediately after installation to generate our camera topology graphs (called camera network graphs). In [45] K-Nearest Neighbours algorithm (KNN) is used for posture classification. KNN is a lazy classification algorithm that uses the majority label of the nearest training examples in the feature space to predict the class of an example.

1.5 FRAME SUBTRACTION

The common method in image processing to highlight useful information is to perform background subtraction in which the background is subtracted from the foreground. Frame subtraction is a means of highlighting the difference between images and can be used to remove the noise [10]. A major use of image subtraction is in the quality control in circuit boards. Object motion can be measured using frame subtraction [69]. Background subtraction requires background modelling.

Background is the main factor in vision tasks such as tracking, classification and identification. The common method in image processing to highlight useful information is to perform background subtraction. Background modelling includes adaptive and non-adaptive methods for the generation of the background [67]. The main focus of researchers is in relation to adaptive background modelling. Some of the new methods for background modelling are based on the mask, the background modelling on the probability and the background modelling is based on learning. Each method has its own challenges and characteristics but the main challenges are to construct robust background modelling and timely updates of the background [68].

1.6 FRAME/BACKGROUND STORAGE

When implementing vision systems on a general purpose computer, the memory may not be the bottleneck as there is no limitation in relation to the available resources. For an embedded platform, the memory does become the source of the bottleneck when complex vision systems are implemented. Vision systems process large amounts of data in real time and thus there is a requirement for both high processing speeds and memory resources. Memory is the point at which a bottleneck can occur and thus this can restrict the performance of computer systems as the memory speed does not increase with the processor speed and thus the gap becomes larger and larger [7]. A vision sensor produces two dimensional data and is often less frequently reused. The memory developed for general purpose computing may be less effective for vision applications [7]. For certain applications, the algorithms should be optimized to use the existing memories efficiently or, alternatively, the memory system should be redesigned in order to provide essential storage. Depending on the applications, some of the vision sensor does not require frame storage, some would require this at the initial stage on a one time basis, some applications would require frame storage in a continuous fashion and others would require periodic frame storage.

1.6.1 No storage

In some of the WVSN applications, it is not necessary to store the background or, indeed, for any other frame and processing is performed on the incoming pixels [9].

1.6.2 Initial storage

In some of the vision sensors, a frame is stored as a background at the initial stage in the memory and then the current frames are subtracted in order to separate the foreground objects from the background [10].

1.6.3 Continuous storage

For some vision processing tasks it is necessary to store the whole frame in a continuous fashion. Such vision sensors also require a high performance processing unit in addition to a large memory for storage [23], [24]. In some applications the frame/background must be updated and stored after a given duration.

1.7 SEGMENTATION

Segmentation subdivides the image into its constituent regions or objects. The segmentation methods are different depending on the application. The level of detail to which the subdivision is carried out depends on the particular application. Segmentation should cease when the objects or regions of interest in an application have been detected. For example, in the automated inspection of electronic assemblies, interest lies in analysing images of products with the objective of determining the presence or absence of particular anomalies such as a missing component or broken connection paths. A segmentation operation can be performed on monochrome images or on colour images.

Segmentation algorithms for monochrome images are generally based on one of the two basic properties of intensity values, discontinuity and similarity. In the first approach, the image is partitioned based upon the abrupt changes in the intensity such as in relation to edges. The boundaries of regions are sufficiently different from each other and from the background to allow for boundary detection. Edge based segmentation is used in this approach. In the second approach, images are partitioned into regions which are similar according to a set of predefined criteria. Region based segmentation is used in this approach [16]. Colour segmentation is typically used for skin-tone detection such as for face detection [5][36]. There are many general-purpose algorithms and techniques for image segmentation. The selection of particular techniques and the level of segmentation are dependent on the characteristics of the application. Some of the techniques are provided below.

- Edge based
- Clustering
- Thresholding
    a. Manual
    b. Adoptive
- Region based
- Segmentation using morphological watersheds
- The use of motion in segmentation

1.8 FILTERING

Despite using a good camera, lenses, illumination and lighting arrangements it is sometimes the case that the image quality is not satisfactory and requires improvement [25]. ‘Filtering’ is the process of accepting or rejecting and this depends on the area of application. In image processing two-dimensional (2D) filtering is referred to as image restoration and/or image enhancement and this depends on the area of application. Although there are areas of overlap between image enhancement and image restoration, the former is largely a subjective process, while image restoration, for the most part, is an objective process [16].

Noise can be removed from the images or, alternatively, images can be enhanced by either applying filtering techniques in the spatial or frequency domain. Image filtering produces a new improved image as the result of processing the pixels of the input image. Each pixel in the filtered image is computed as a function of one or several pixels in the input image. Image filtering approaches fall into two broad categories: a linear digital filter and a non-linear digital filter, based on the computation performed on the neighbouring pixels. Based on the noise behaviour, filtering can be performed in either the frequency or spatial domain, for example, periodic noise can be handled better in the frequency domain as compared to in the spatial domain [16].

1.8.1 Linear Filtering

Linear filters can be designed and implemented either in the spatial domain (spatial operations) or in the frequency domain (transform operations). The response of a linear spatial filter is given by the sum of products of the filter coefficient and the corresponding image pixels [16]. Linear filtering in the spatial domain can be implemented using a convolution mask. Convolution operators usually refer to a class of 2D kernels that is convolved with an image to produce the desirable effect. Sometimes, performing convolution operations in the frequency domain proves to be efficient as convolution is based on multiplication in the frequency domain. In the frequency domain the convolution can be efficiently implemented by using the Fast Fourier Transform (FFT) approach, which is efficient when the filter size is greater than 13x13 pixels. However, sometimes it is better to implement the corresponding linear filter by using either 3x3 or 5x5 size kernels as direct convolution without using the FFT. Linear spatial filters are simply the average of the pixels contained in the neighbourhood of the filter mask [17]. These filters are sometimes referred to as averaging or low pass filters. These filters have smoothing effects which reduces the noise, but with a slight introduction of blur [18]. In the spatial domain average filters are used and in the frequency domain, high pass, low pass, and band pass filters are used as the linear filters.

1.8.2 Non-linear filters

As well as the optimality and the fact that linear filters are well understood and, they additionally possess some elegant theoretical properties, linear filters offer only an acceptable performance in many situations. For example in the case of non-Gaussian and non-additive noise, linear filters often tend to remove meaningful high frequency components, such as edges and fine details. This unacceptable result favours the use of non-linear filters, especially when there is signal and noise overlap in the frequency domain. Non-linear imaging filters are used in digital image filtering, image enhancement and edge detection. One of the main limitations of non-linear techniques at present is the lack of a unifying theory that can handle all the existing non-linear filters [34].

1.8.2.1 Order statistics filtering

One of the important families of non-linear filters is based on order statistics filters also known as L-filters. These are running estimators and thus are making a compromise between a pure non-linear operation (ordering) and a pure linear operation (weighting) [18]. The median filter is the best known filter of the order statistics filters. Other order statistics filters includes max and min filters, a mid-point filter and an alpha-trimmed filter. The max and min filters are closely related to two morphological operations called, dilation and erosion. Non-linear filters which are based on the order statistics filter possess good robustness properties in the presence of impulsive noise. Impulsive noise is frequently encountered in digital image transmission as a result of man-made noise sources or decoding errors. They retain the edge information, which is important for human perception. Although their theoretical analysis is relatively difficult compared to that for linear filters, however, the computation is relatively easy and fast. One of the main limitations of non-linear techniques at present is the lack of a unifying theory that can handle all the existing non-linear filters [34].

1.8.2.2 Morphological filtering

Morphological operations are used to extract image components that are useful in the representation and description of a region's shape, such as boundaries, skeletons and the convex hull. Mathematical morphology as a theory and methodology for non-linear image processing is more systematic and rigorous than traditional linear image processing methods. It is popular because of its sound mathematical foundation but is mainly based on its inherent ability to exploit the spatial relationships of the pixels. Mathematical morphology regards an image as a set and has then used another smaller set of images as a structuring element or structuring function in the case of grey scale morphology. Mathematical morphology can perform image tasks from the simplest to the most demanding such as noise reduction, edge detection, segmentation, texture and shape analysis. Erosion, dilation, opening and closing are basic operations of morphology. Mathematical morphology was originally proposed for binary images and latter extended to grey scale functions and images [35]. However, because a number of difficulties are associated with the grey scale, these are used infrequently in practice [16].

1.9 INTENSITY TRANSFORMATION

Intensity transformation is a spatial domain category of image processing, working on the single pixels of an image. Spatial domain techniques are efficient computationally and require less processing resources for their implementation [16]. Intensity transformation is mainly used for contrast manipulation and image thresholding. Different types of intensity transforms used for image enhancement now follow.

1.9.1 Linear transformation

There are two types of linear transformation, namely, negative and identity transforms. Identity transform is rather trivial in which the output intensity values are similar to the input intensities whereas for the negative transformation, the output intensities are the opposite to those of the input intensities.

1.9.2 Logarithmic transformation

Logarithmic transformation includes both log and inverse-log transformations. The logarithmic transformation is used to expand the dark pixels values in an image and to compress the high level pixels values. The opposite is true for the inverse-log transform.

1.9.3 Power law (Gamma correction)

The gamma correction is a non-linear mapping of intensity from the input to the output in which the gamma parameter is adjusted in order to improve the visual response. Gamma correction is an effective tool for manipulating the histogram of an image which is either over or under exposed. In analogue video systems, gamma correction is performed with analogue circuitry and the gamma parameter is adjusted manually. In digital video streams, the gamma correction is performed using mathematical operations in digital circuits. The gamma correction can be automatically adjusted in order to compensate for the changes in the scene. Gamma correction is available in many image processing libraries on a software platform but the implementation on a hardware platform is relatively complex. In [38] they implement the gamma correction on a hardware platform for dynamic contrast enhancement.

1.9.4 Piecewise-linear transformation

Compared to the other mentioned three approaches, the piecewise linear function has the advantage of being arbitrarily complex. The disadvantage is that their specification requires more user input. The following are the piecewise linear functions.

a. Contrast stretching

b. Intensity level slicing

c. Bit-plane slicing

1.9.5 Histogram processing

The histogram is the basis for various spatial domain image processing techniques. Histogram manipulation can be used for image enhancement and the information in a histogram can be used for other image processing applications such as compression and segmentation [16].

1.10 COLOR AND GRAYSCALE/VALUE TRANSFORMATION

There are two basic classes of image transforms, colour transforms and grey scale /value transforms. Colour transforms involve the conversion of colour information from one colour space to another, conversion from colour to grey scale and representing colour images with a false colour. In the grey scale value transforms, operations such as pixel level mapping, mathematical and morphological operations are included [31].

1.10.1 Colour transforms

Colour images are stored in various file formats. When the image is stored as a 2D form, it has an associated or implied colour map and the RGB value of an individual pixel is obtained by mapping values in the 2D data into the colour map. In a 3D image, each image plane has individual red, green or blue colour components. If the image data is in unsigned byte format, each plane has pixels within the range of 0 to 255. Depending on the application; images must be converted into various formats. There are many image transform methods. Images of a 2D nature can be converted to 3D RGB and colour images can be converted to grey scale or grey scale can be converted to binary images. In some applications, the requirement is to convert the image from RGB to the HSI or YUV colour space. The segmentation works better in HSI and YUV [36].

1.10.2 Grey scale/value transform

These types of transforms can be applied only on 2D grey images.

1.10.2.1 Spatial transforms

Spatial transforms describe the operations that change the position of the data. These included image transforms, matrix transposes, image rotation, image scaling and image registration.

1.10.2.2 Mathematical transforms

This class of transform includes Discrete, Fourier, wavelet, Hough and Hartley transforms.

In the majority, images processing operations are performed directly in the spatial domain. Spatial domain techniques are efficient computationally and require less processing resources for their implementation. In some cases, it is better to perform operations on the images that are transformed in another domain. After the operations, the inverse transform is applied to bring the images back to the spatial domain. The choice of a particular transform in a given application depends on the amount of reconstruction error that can be tolerated and the computational resources available [16]. The basic steps for performing image processing in the linear transform domain are shown in Figure 4.

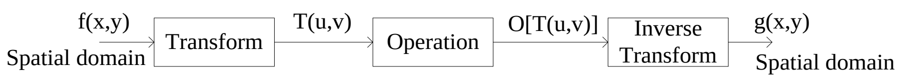


Figure 4. Generalized linear transform domain

a. Discrete cosine transform

The N-point Discrete Cosine Transform (DCT) is related to the 2N-point Discrete Fourier Transform (DFT), from which it can be computed. DCT is similar to DFT but using only real numbers. It has the advantage of good results for the compression of natural images in which it ignores the least important coefficients. It was one of the foundational transforms chosen by the international committee of JPEG (Joint Picture Experts Group) for compression. DCT is often for lossy data compression and signal processing and provides a good compromise between the information packing ability and the computational complexity. The performance difference between the DCT and wavelet transform for both images and video coding is given in [33]. The authors concluded that the main factors in image coding are the quantizer and entropy rather than the difference between the wavelet transform and DCT.

b. Fourier transform

The frequency domain makes large filtering operations much faster. The Fourier transform is good for identifying a periodic component or lattice in an image. Periodic components can be used to remove regular dirty spots or noise from images. Some Fourier components of higher frequency can be used to remove anti-aliasing effect such as removing ugly edges. It is important to note that a computational requirement in this area of image processing is not trivial. Thus, it is necessary to find ways to speed up and simplify the Fourier transform computations. Compared to Discrete Fourier transform, a fast Fourier transform is faster on the same machine [16]. In DFT there are many redundancies.

c. Wavelet transform

A modern transform based image processing technique to inspect, transmit or compress an image is the discrete wavelet transform. Compared to the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small waves called wavelets. These wavelets are of varying frequency and of limited duration. In a wavelet transform, the temporal information is not lost, as is the case in the Fourier transform, so it has both frequency and location (in time) information. The wavelet transform de-correlates the data, allowing for the extraction of interesting features. The partial imaging operations approached from a wavelet point of view included image matching, registration, enhancement, segmentation, de-noising, compression and morphological filtering [16].

d. Hough Transform

The Hough transform is a mapping algorithm used for the detection of features of particular shapes such as lines or circles in digital images. At present, the Hough transform has a number of applications including traffic observation systems, land monitoring and robotics. The advantage of the Hough transform is its tolerance towards holes in the boundary line. The Hough transform requires complex computation, has large memory requirements and is slow. The Fast Hough Transform has reduced memory requirement and is fast [16] [31].

e. Fast Hartley transform

The Hartley transform is similar to a Fourier transform apart from the fact that it uses only real numbers [31].

1.11 OUTPUT DATA TYPE

The amount of data being transmitted from the vision system has a profound impact on the energy consumption in the case of wireless transmission and thus it is necessary to investigate the type of data which is required to be sent. In much of the literature it is demonstrated that the total energy consumption of the system can be reduced by a reduction in the communication burden as it has a greater requirement for power than that required by the computation [49]. The output of each vision system can be different depending on the application. In many applications, it is necessary to send the whole image either to obtain information that cannot be extracted automatically or for safety or for legal problems. In order to reduce the amount of data, the image can firstly be compressed and then sent over the transmission medium [49]. In some applications, the output is reduced to a few bytes such as when features are transmitted which may be in the form of areas or position information with regards to objects. In some applications it is sufficient to send a flag to indicate the presence or absence of an object so that it triggers an alarm or some other action. The output of the wireless VSN can be in different formats such as in matrix form, scalar, vector or flag depending on the application. The large amount of data generated from the sensor node would increase the transmission time and also the transmission energy consumption. In [11] a lightweight camera, the sensor network approach is proposed, which operates on symbolic information rather than on images. The work of the authors is in relation to the development of biological-inspired image sensors which are specialized in obtaining useful information with regards to the event. This will reduce the processing, bandwidth and power supply requirements. Additionally the output is in the form of an address event stream which will solve the privacy problems that cause people to feel uncomfortable. Instead of sending raw data, the node sends symbols that summarize the motion activity.

- Matrix
- Vector
- Scalars

- Flag

1.12 POST PROCESSING

Post processing includes high level image processing tasks which are typically used for object recognition/identification and localization. High level processing such as classification and recognition operates on features or objects which reduce the required data bandwidth; however, this involves highly complex algorithms. Such operations are typically data dependent and have an irregular control flow. Thus, depending on the requirements, these tasks can be performed either on dedicated platforms or multi-core processors or on the server side after data has been transmitted from the sensor to the server [10]. Common features include shapes, contours, colours, texture and sizes. Edge detection and extraction is one of the most used segmented techniques [25]. Some of the posts processing functions are list as follows

- Labelling
- Feature extraction
- Classification
- Neural networks
- Tracking
- Recognition
- Blob search

1.13 FEEDBACK OF THE SYSTEM

Vision systems have been used for recognizing, locating and inspecting stationary parts in robotic applications. The advancement in image processing equipment has enabled the use of vision as a feedback signal, which can be used to control the position and orientation of a manipulator end-effector in real time. In this manner, the system would be able to understand and be adapted to the changing environment. This visual feedback would allow a robot to manipulate and track a moving part without having any previous knowledge of the part's placement or motion [61]. High speed realization of visual feedback requires that image tasks are processed as rapidly as the feedback rate. In some vision systems the difference between the images is small as the frame rate is quite high. By utilizing this feature conventional image processing can be realized by simpler and faster image processing algorithms. In [62], the authors used a 1 ms visual feedback system using a general purpose massively parallel processing vision system. The system can execute a series of operations, from obtaining images and processing them to the feedback of its results within 1 ms.

Feedback for the system can be from different stages of processing and can be control signals in the form of;

- Matrix
- Vector
- Scalar
- Flag

In active perception systems it is necessary to have high speed feedback between the sensing and processing units. The feedback is performed at different levels ranging from an analogue threshold setting and dynamic configuration of the image sensor at low level, up to a region of interest (ROI) selection at high level [55]. In [56] a vision feedback manipulator has been developed and a real-time vision feedback mechanism has been tested as an automation scheme for manipulation. In [57] a feedback system has been developed based on face and eye detection technology for the monitoring of students in a classroom. The authors in [58] presented a feedback method for detection and tracking, which provided significant savings in relation to processing time. Instead of performing foreground detection and tracking independently and sequentially at each frame, the feedback method incorporates the information from the tracking stage into the foreground detection stage. In this manner, foreground detection is performed in smaller regions as opposed to the whole frame. The authors compared a sequential method and a feedback method and claimed that the feedback method significantly reduced the processing of a frame. This saving is then used by switching the processing platform to an idle mode.

6 SYSTEM TAXONOMY

The system taxonomy is developed according to the flow shown in Figure 1. The vision functions are used to propose system taxonomy. Therefore, all possible vision functions used in vision systems are discussed in detail in Table 1. Those similar vision functions are grouped together to construct an abstraction of the functions. The abstracted vision functions are then used to develop a system taxonomy for the vision systems which is shown in Figure 5. Some of the functions are labelled by letters, which are further expanded to lower levels, as shown in Figure 6 to Figure 12.

The system taxonomy is grouped into 9 levels including trigger type, input sources, data depth, training of the system, storage requirement, vision processing, post processing, output type and feedback. The systems are categorized into three types, based on how the system is triggered to start the processing. In time driven systems, the processing is performed after a given time duration. In event driven system, processing the functions starts when an event occurs in the field of view. Other types of systems start processing in response to a query from some user. The system can capture data using three types of sources including area scan, line scan or another sensor. Images can be captured in binary, grey scale or in colour format. Conversion from one format to another would require additional resources. Therefore, a better strategy is to capture the image in the relevant format, as discussed in sub-section 1.3. For some applications, the system must be trained so that it is adapted to the environment while in some applications there is no requirement for training. Similarly, there could be a requirement for storage in some systems in order to store frames for subtraction, for temporal filtering or for template matching while in others, storage is unnecessary.

The vision processing level in Figure 5 shows the abstraction of vision functions, which may not be exhaustive but great efforts have been made to include the maximum functions, required for embedded vision systems. Some classes of systems require training, frame storage, frame subtraction, segmentation, etc. while others may not require these functions. Therefore, each task has an alternative path to show whether it is required or not. After the processing level, the post processing level is shown. It involves more complex vision processing functions such as labelling, feature extraction, classification, compression and neural network. The post processing is discussed in section 1.12.

A vision system is characterized by the output type it produces. The output can be a matrix, vector, scalar or flag. In some systems, it can be sent directly to the user and in others, it can be used for feedback. In some applications, feedback can produce better results as discussed in [58]. The system taxonomy does not show complexity or the vision flow of the systems because it is beyond the scope of this report. It shows the different functions necessary for vision systems. The taxonomy given here provides an abstract reference model. The model can be used by designers to classify their system and by researchers to compare their system's results with other systems. After presenting the system taxonomy, one published system [10] is described in more detail in relation to the taxonomy. The system is mapped on the proposed taxonomy and individual vision functions are discussed.

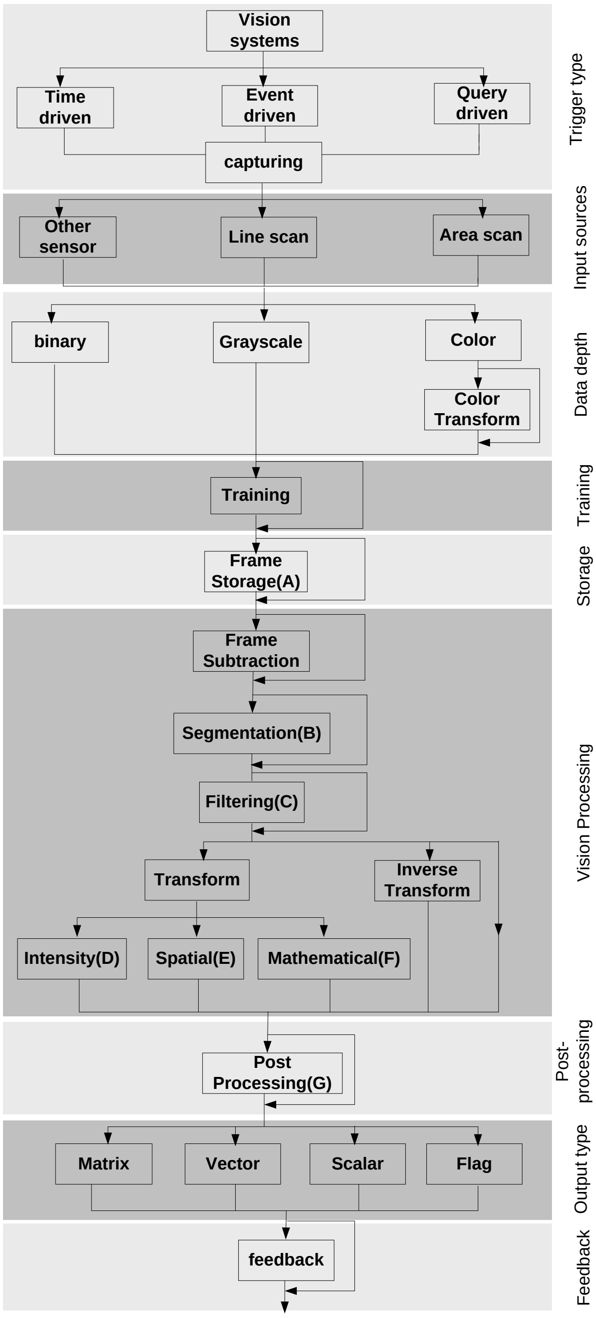


Figure 5.Taxonomy of vision system. The associated functions with labels are shown in Figure 6 to Figure 12.

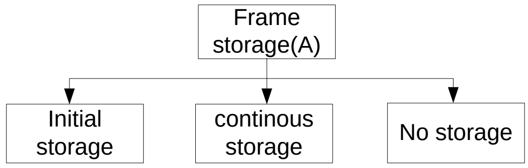


Figure 6. Storage in vision systems

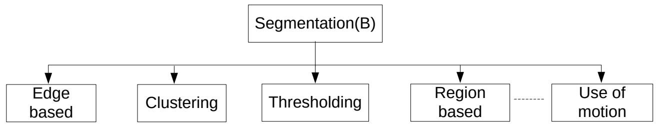


Figure 7. Segmentation in vision systems

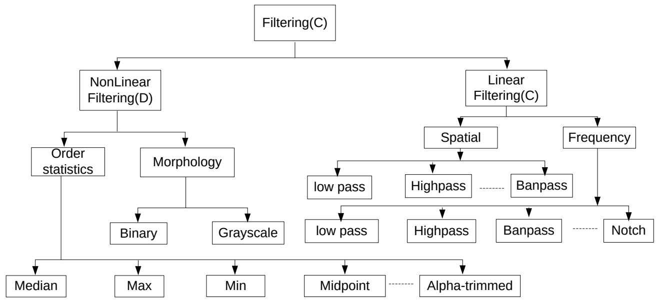


Figure 8. Filtering in vision systems

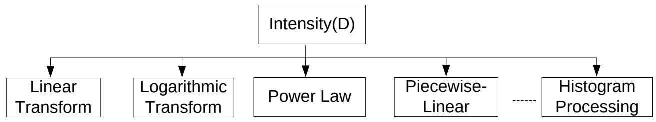


Figure 9. Intensity Transformation in vision systems

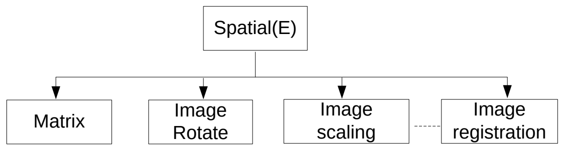


Figure 10.Spatial Transformation in vision systems

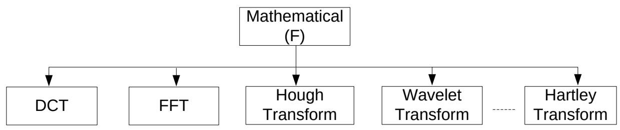


Figure 11. Mathematical Transformation in vision systems

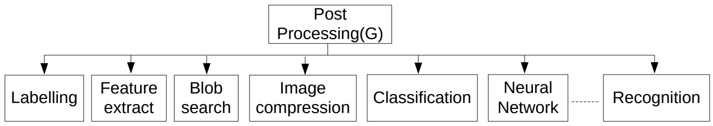


Figure 12. Post-Processing in vision systems

1.13.1 Classification of one system

A WVSN system [10] is used as an example to show the system classification according to the proposed taxonomy. The main focus in this system is to develop image processing/analysis methods in order to detect particles in a flowing liquid, measure their sizes and transmit the features' information to the user. The vision function flow is shown in Figure 19. An experimental optical measurement set-up having a hydraulic system with flowing oil, polluted with particles is shown in Figure 18. The images at the different stages of the algorithm are shown in Figure20. The system is classified by using the proposed taxonomy as shown in Figure 13 where the bold line represents the system flow. The expanded functions are labelled by a capital letter such as storage by A, segmentation by B, etc. These functions are shown from Figure 14 to Figure 17, with similar bold lines. The bold lines show that the system represents a particular class of vision system in which processing is performed after a given time. It means that the system is time driven and an area scan camera is used for capturing. It is not necessary for there to be system training as only targeted objects in the field of view, are required to be detected. The background is stored initially, when there are no objects in the field of view. To detect the objects, the current frame is subtracted from the background. Thus frame storage is required in the system. Other vision functions include, background subtraction, grey scale segmentation, filtering. The post processing functions required for this system are labelling, feature extraction and classification. The output data will be in the form of a vector containing objects/particles areas, number of particles and position information. This vector is transmitted to the user using the IEEE 802.15.4 complaint transceiver. The individual processes involved in this system are discussed in detail.

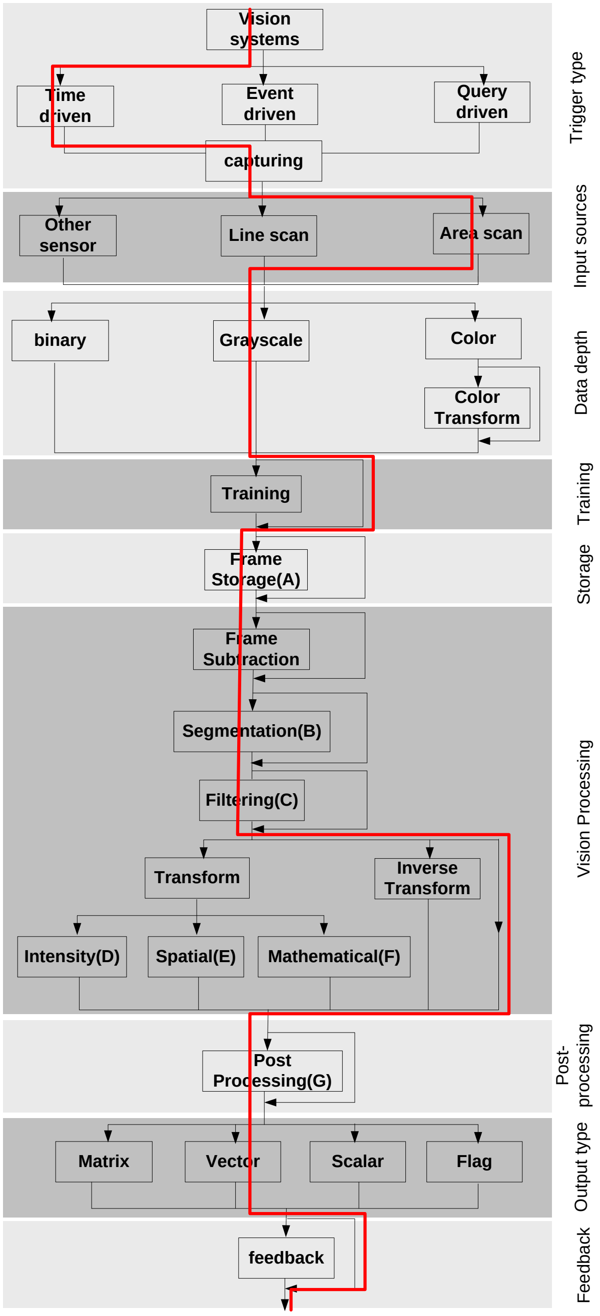


Figure 13.Taxonomy of one class of vision system, represented by bold line. The associated functions with labels are shown in Figure 15 to Figure 17.

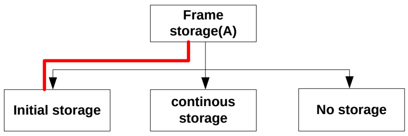


Figure 14. Storage in vision systems

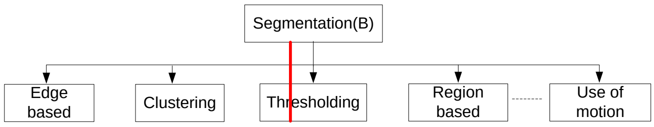


Figure 15. Segmentation in vision systems

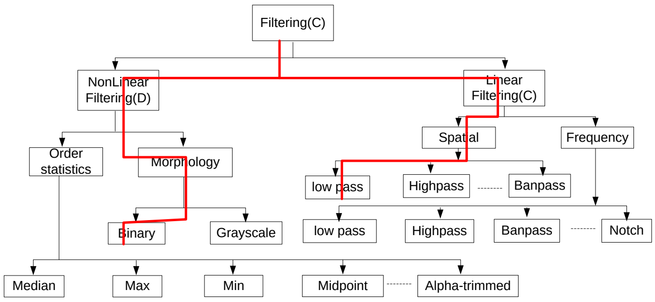


Figure 16. Filtering in vision systems

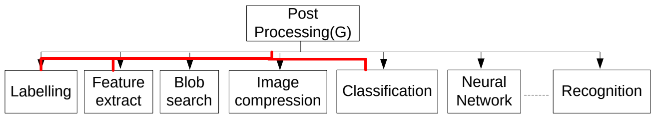


Figure 17.Post processing in vision systems

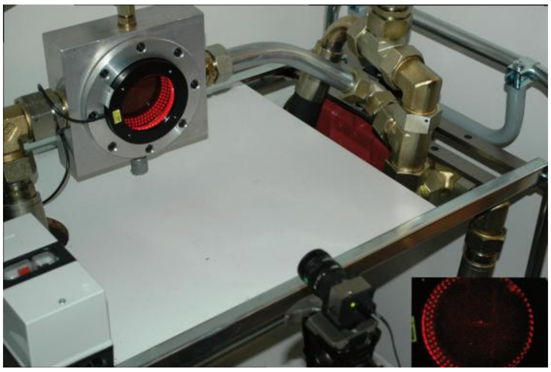


Figure 18. Measurement setup for oil particles

The individual processes are discussed in detail.

1.13.1.1 Image capture

A CMOS active-pixel digital image sensor (MT9V032) from Micron is used in this work for image capturing. Most cameras use CCD or CMOS image sensors but the CMOS normally has a lower power consumption, smaller area and faster operation. The MT9V032 is designed to support the demanding interior and exterior surveillance imaging requirements, which makes it an ideal candidate for a wide variety of imaging applications in real-world environments. The sensor can be operated in its default mode or it can be programmed for frame size, exposure, gain setting, and other parameters. The default mode, outputs a wide-VGA size image at 60 frames per second (fps) with a resolution of 752Hx480V, having a frequency of 26.6 MHz [59]. For this particular application, the camera is programmed through an I2C interface for a resolution of 640x400, running at a frequency of 13.5 MHz.

1.13.1.2 Background subtraction

In an object based method, the background is generated in real time using a spatial filter which is an averaging filter of 21x21 masks. An averaging filter with a mask of 21x21 used for background generation took 5 minutes on the software platform, which is not realistic for the implementation of the VISION SYSTEM. Instead of this approach, the preference was to store the background image in the flash memory. The background image is then subtracted from the current frame in order to detect objects which could be magnetic particles or some flowing bubbles.

1.13.1.3 Segmentation

The process of spatial partitioning of an image into mutually exclusive connected image regions is known as image segmentation [60]. Typical segmentation algorithms are thresholding and clustering, boundary detection and region growing. Thresholding has been used in this application as the objects that require to be detected are white and the background is relatively black. All pixels having a grey scale value less than a pre-defined threshold are assigned a zero value and the remaining pixels in the image are assigned a value of one. The resulting image after segmentation is a binary image having value of 1 or 0 for all pixels.

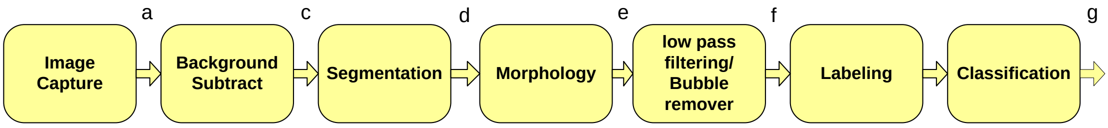
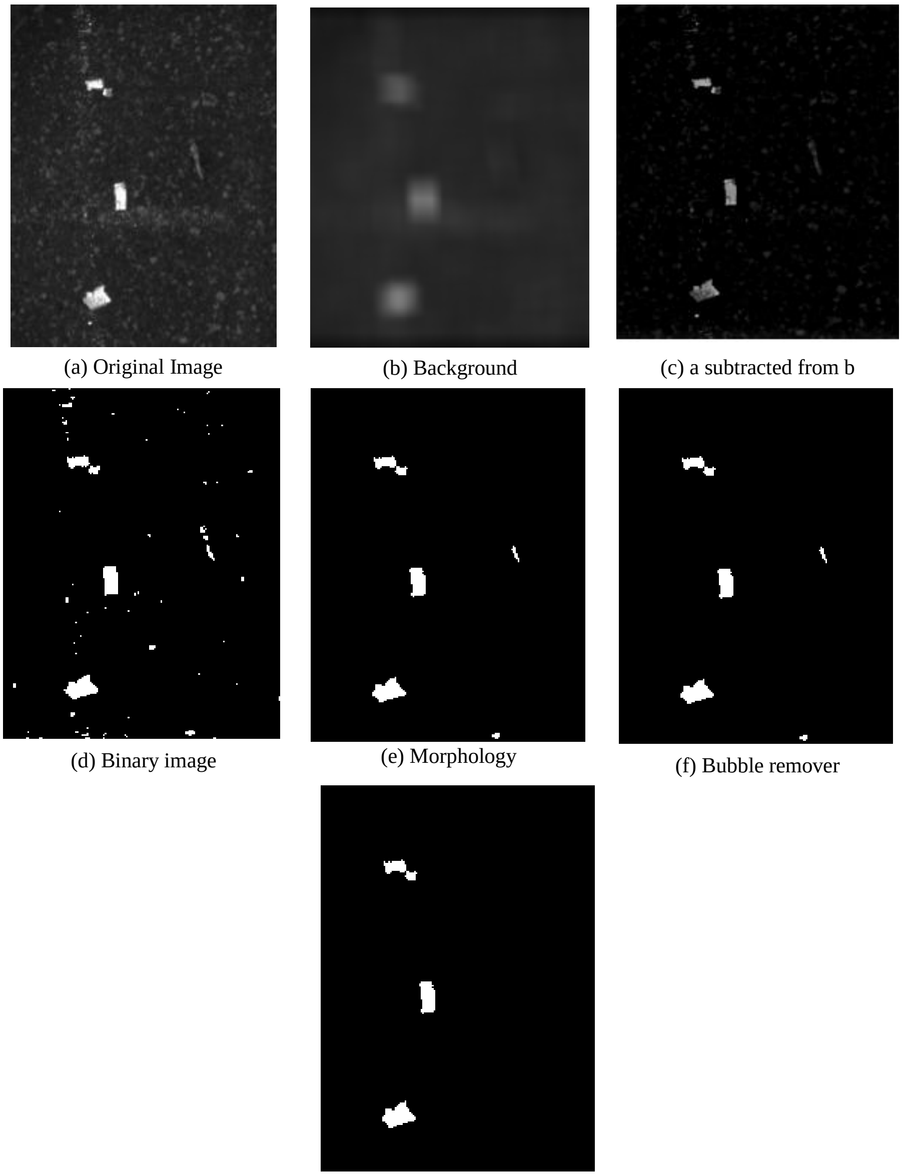


Figure 19. Algorithm flow of different processing strategies



(a) Original Image

(b) Background

(c) a subtracted from b

(d) Binary image

(e) Morphology

(f) Bubble remover

(g) Labeling and classification

Figure20.Image at each Algorithm step

1.13.1.4 Morphology

After segmentation, a morphological operation is performed in order to remove one or two pixel false objects. The morphology operation used in this algorithm is erosion followed by dilation with a mask of 3x3. During erosion and dilation, two complete rows must be stored in line buffers so that the neighbourhood information for the operation is available. Morphology is a spatial domain image processing operation which requires neighbourhood processing where values of a group of pixels in an input image are required to compute only one pixel in the output image. This type of processing requires a processing mask or kernel which defines the operation. Figure 21 (B) shows the memory architecture that provides the necessary storage of the previous pixel values in order to facilitate the sliding window shown in Figure 21 (A). The line buffers must hold the pixels in a row. In the figure, P*ij* represents the pixel data at the *i*-th row and *j*-th column in the neighbourhood. The size of each element in this buffer depends on the dynamic range of the pixel.

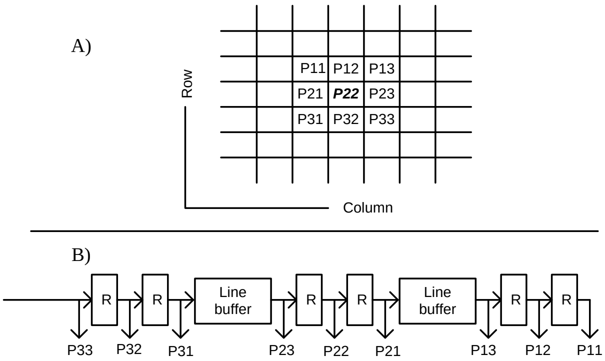


Figure 21. Part A) shows 3x3 neighbourhoods processing, Part B) shows its implementation using line buffers

1.13.1.5 Bubble Remover

The bubble remover is basically a spatial domain low pass filtering operation in which unwanted objects are removed. In the object based method, bubbles as well as particles are treated as objects at the initial stage and the bubbles are removed after the classification stage. Bubbles can be removed by comparing the area and location of objects in the current frame with the area and location of objects from the previous frame. Objects whose area and location are not same will be treated as bubbles and hence removed. The challenge associated with this method is that, sometimes, due to changes in the illumination, it is possible for the area of the object to be decreased or increased in consecutive frames and those magnetic particles might be treated as bubbles. This challenge has been dealt with by introducing a flexibility of one to three pixel variations in the area and location of objects in consecutive frames.

In the pixel based method, bubbles are removed after morphology. Bubbles can be identified as moving objects, so if an object changes its location in two consecutive frames, this confirms that it is a bubble. The corresponding pixels in two consecutive frames are compared to check the binary values and if the values are not same, zeros are placed at these pixel locations to remove the bubbles. For bubble removing in the pixel based method, the entire previous frame must be stored with binary data so that current frame data is compared with the stored frame.

1.13.1.6 Labelling and Features Extraction

During labelling, pixels belonging to the same image component are assigned a unique label. A search is conducted of the entire binary pixel array for white pixels, ignoring the black pixels as these represent the background. The separate connected regions of pixels having a value of 1 are assigned a unique label. Each connected region will have a different label. In the hardware implementation, an architecture has been used which performs real time component labelling [9]. At feature extraction, an image component is described in terms of region features such as area, mean grey value or position. This feature information can then be used for the classification of image components.

The architecture used for the system is shown in Figure 22 in which vision tasks are processed on the embedded platforms such as micro-controller and Field Programmable Gate Arrays (FPGAs). The vision sensor node is powered by 4 AA batteries and the data is transmitted using an IEEE 802.15.4 compliant transceiver to the server.

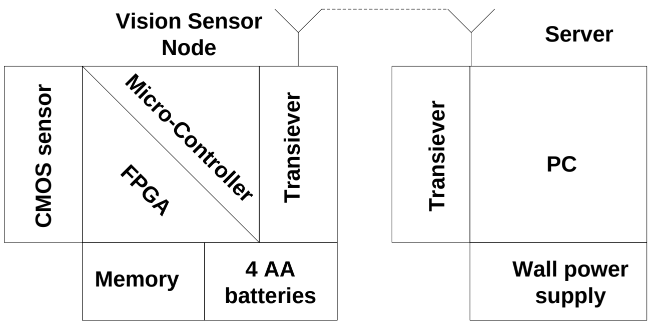
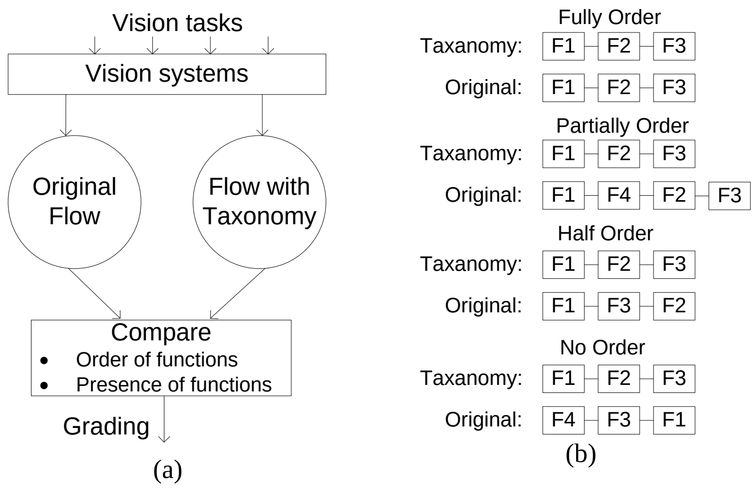


Figure 22.Architecture of the wireless vision sensor node

In the following sections, an evaluation mechanism for the taxonomy will be provided and a discussion in relation to the outcome of the evaluation process. An evaluation criterion for taxonomy is selected based on the assumption that if a certain criterion is fulfilled, the taxonomy is allocated a particular grade.

1.13.2 Evaluation mechanism of Taxonomy

The taxonomy evaluation is a challenging task due to the large number of influencing factors, many of which are difficult to quantify. The outcome of such an evaluation is generally considered to be a subjective judgement. However, it is necessary to conduct an educated debate about this judgement. The objective, in this case,is to obtain quantitative parameters in order to test different vision systems with the taxonomy. These parameters will show how mature the taxonomy is and how well it covers the existing vision systems. The evaluation mechanism is shown in Figure 23 in which different vision systems are analysed and compared with the proposed taxonomy. The actual system flow is compared with the flow of the system, derived from the taxonomy. The two parameters used for this comparison are, the order of the functions and the presence of functions in the system taxonomy. These two parameters are graded on four levels, and where A, B, C and D are used for the ordering of functions and W, X, Y, Z used for the function's presence. This grading will make the evaluation mechanism transparent.



Order of functions
A=Fully order
B=Partially order
C=Half Order
D=No Order

Presence/Missing of Functions Grading
W=(No task missing)
X=(1-2 missing)
Y=(3-4 missing)
Z=(more than 5 functions missing)

Figure 23.Suitability criteria of Taxonomy, ordering and functions coverage of Taxonomy

7 RESULTS

The vision systems used for the taxonomy evaluation are shown in Table 2. The flow of the vision functions based on the taxonomy is shown in column2, whereas the actual system flow is shown in column3. The first evaluation criterion is how well the taxonomy is ordered. Four levels have been allocated to this criterion including fully ordered (A), partially ordered

(B), half ordered (C) and not ordered (D). This is shown in Figure 23 (b). The taxonomy is fully ordered when the flow of the vision functions for any vision system is similar to the flow of the functions, derived from the taxonomy. The taxonomy is partially ordered when the taxonomy flow follows the original flow of the system, except for the missing functions in between the ordered functions. Half ordered is assigned to the taxonomy, when it can be fully/partially ordered by shifting one or two vision functions while no ordering grade is allocated when more than two functions are required to bring half order.

Another parameter used to evaluate the taxonomy is the presence of the functions in the taxonomy in comparison to the functions present in the system. The taxonomy is graded W, when there is no missing functions in the taxonomy in comparison to the original system. The grade X is assigned when one to two functions are missing, Y is assigned when three to four functions missing and Z is assigned when more than five functions are missing in the taxonomy. These grading are shown in column4 and column5 of Table 2. For the first system, it is evident that the taxonomy is fully ordered and has A grade for ordering. Similarly, the presence of functions has a W grade as all functions of this system are present in the taxonomy. For the second system, two functions, down-sampling and centroid to grammar are missing. Although, the task central to grammar is specific for this particular system it has been considered as missing in the evaluation process. The presence of down-sampling in between the ordered function results in the X grade, otherwise, the taxonomy is fully ordered. The absence of two functions results in grade X for functions presence.

It must be noted that all systems have mentioned the type of input sources such as if line scan or area scan camera is used. In addition to this, the input format is also mentioned. Similarly, the function such as how often the system is triggered to start processing and the output type of the systems is intuitive for the systems and hence they are not mentioned separately. Thus the focus is on processing and post processing functions for comparison. The evaluation summary is shown in Table 3. From Table 3, it is evident that the visions functions are well covered. The system taxonomy has a W grade for 10 systems and an X grade for 9 systems, for the functions presence out of the total investigated systems. It means that the taxonomy has covered 95 percent of the investigated systems in which either no task is missing or at most 2 functions are missing. The taxonomy has a Y grade for one system while there is no Z grade. It means that there is no system for which the taxonomy misses more than 5 functions. It must be noted that the system taxonomy does not show the flow of the vision function but rather shows the functionality of the system. Thus the ordering results are not to the level of functions presence results. Out of total investigated systems, the taxonomy has A grade (fully ordered) for 4 systems, B grade (partially ordered) for 6 systems, C grade (half ordered) for 9 systems and D grade (no order) for 1 system.

Table 2. Systems with Taxonomy flow vs Actual flow

systems	Systems using Taxonomy	Systems in Original	Functions: Order	Functions: presence
1	cam → Frame storage → Subtract → Segment → Filtering → Post processing → vector	cam → Frame storage → Subtract → Segment → Morph → Low pass filter → label → Feature extract → vector	A	W
2	cam → Training → Frame storage → Subtract → Filtering → Post processing → scalar	cam → downsample → Training → Frame storage → Frame differece → Centroid calculation → Centroid to grammer → scalar	B	X
3	cam → Training → Frame storage → Subtract → Post processing → vector	cam → Training → Frame storage → Subtract → ROI → Feature exraction → classification → vector	B	X
4	cam → Color transform → Frame storage → Mathematical Transform (DCT) → vector	cam → Color transform → Frame storage → Mathematical Transform (DCT) → vector	A	W
5	cam → Training → Frame storage → Subtract → Segment → Filtering → Intensity Transform (histogram) → Post processing → vector	cam → Training → Frame storage → Subtract → Median Erode Dilate → segment (color segment) → label → Feature extract → Feature fusion → vector	C	X
6	cam → Frame storage → Subtract → Segment → Post processing → vector	cam → Frame storage → Subtract → Segment → Region of interest → Features of ROI → vector	B	X
7	cam → Frame storage → Subtract → Segment → Filter → Post Processing → vector	cam → Frame storage → Subtract → median → Segment → Bounding box → Tracking → vector	C	W
8	cam → Frame storage → Subtract → Segment → Intensity Transform (histogram) → Post processing (blob filter) → vector	cam → Training → Frame storage → Subtract → Histogram → Segment → Blob filter → vector	C	W
9	cam → Training → Frame storage → subtract → Segment → Post processing → vector	cam → Training → Frame storage → subtract → segment → label → classify → vector	A	W
10	cam → Frame storage → subtract → Segment → Filtering → Post processing → scalar	cam → Frame storage → subtract → Segment → Low pass filter → Blob search → Bounding box/feature → ROI → Matrix	B	X
11	cam → Frame storage → Subtract → segment → Features/ bounding box → tracking → feedback → vector	cam → Background modelling → Frame storage → Subtract → segment → Features/ bounding box → tracking → feedback → vector	C	W
12	cam → Frame storage → subtract → Mathematical transform (Wavelet) → Matrix	cam → Background modelling → Frame storage → subtract → Wavelet → Matrix	C	W
13	cam → segment → Filtering → Post processing	cam → segment → Edge detect → Edge tracking	A	W
14	cam → segment → Filtering → Mathematical transform → Post processing	cam → Image division → Edge detect → line extract Houghman → curve (Plot depth) → Segment → Vertical pixel length → Depth curve analysis	D	Y
15	cam → Training → Frame storage → subtract → Segment → Filtering → Filtering → Post processing → vector	cam → Training → Background modeling → Frame storage → subtract → Segment → Feature extraction → classification → Morphology → Vector/images	C	W
16	cam → Color transform → Segment → Post processing	cam → Color transform → Segment → ROI → HOG calculation → Pose classification	B	X
17	cam → Training → Frame storage → subtract → Segment → Post processing → vector	cam → Training → Frame storage → subtract → Segment → Object localization → Boundry detection → tracking → Matrix	B	X
18	cam → Training → Segment → Filtering → Post processing → vector	cam → Training → downsample → Skin color classify → Low pass filter → segment → Feature extract → Gesture classification → vector	C	X
19	cam → Training → segment → Filtering → Post processing → vector	cam → Training → Face detect → [Skin color detect → Hand detect; Shirt color detect → Shoulder detect] (Segment) → Body part position → Median filtering	C	W
20	cam → segment → Filtering → Post processing	cam → ROI → Edge detect → segment → feature → feature matching	C	X

Table 3. Comparison results of 20 systems

Ordering			Functions presence		
Grade	frequency	%age	Grade	frequency	%age
A	4	20	W	10	50
B	6	30	X	9	45
C	9	45	Y	1	5
D	1	5	Z	0	0

8 DISCUSSION

By evaluating the taxonomy for different types of vision systems, the results for the functions presence are promising as compared to those for the ordering of functions. From Table 3, it is evident that the taxonomy has W and X grades for 95 percent of the systems. It means there is no task missing or at most two functions are missing in the taxonomy for these systems. The missing functions for some systems are problem specific as in the case of system5, where feature fusion is required from three vision systems. This may not be required in other systems. With regards to systems 3, 6 and 10, 16 and 20, it must be noted that the Region Of Interest (ROI) is considered as the missing task. This is because it is not mentioned in the taxonomy. One opinion is that ROI is covered in the segmentation which includes region based segmentation. Similarly, there are other missing vision functions which are shown in italics in cloumn3 of Table 2. There are some functions whose functionality can be achieved by using the existing functions in the taxonomy. These functions are considered as present functions such as the background modelling in systems 11, 12 and 15 because the background can be generated using filtering techniques or it can be stored at the initial stage when there is no object in the field of view. The systems whose three to four functions are not covered in the taxonomy form only 5 percent of the total investigated systems. In comparison to the promising results for the functions presence, there is a trade off in another dimension. The ordering of functions is derived from the taxonomy in relation to the ordering of the actual functional flow in the system. From Table 3, it is obvious that the system taxonomy has A and B grades for 50 percent of the systems in relation to the ordering of the functions while C and D grades are in relation to the remaining 50 percent . One of the reasons for this is that the system taxonomy does not show the flow of the functions but rather shows the functionality of the vision systems. For example, filtering is a function which may be implemented before segmentation [14] or after segmentation [10], depending on the requirement of the application. There are many different solutions for vision problems.

The purpose of this work is to provide a benchmark or at least an abstract model for vision systems. This is because; processing the vision functions on an embedded platform is a challenging research area which is still immature. The model can be used by designers and researchers to classify and compare their systems. The designers/researchers can propose generalized architectures for different class of vision systems. Moreover, it will assist the designers/researchers to predict the relevant design and implementation challenges by studying similar classes of implemented vision systems. It will reduce the design time and efforts for similar types of vision systems.

9 CONCLUSION

In this report, the aim has been to provide a benchmark or at least an abstract model which can be used to propose generic architectures for different class of vision systems. Researchers can use it to classify and compare different vision systems. The development of system taxonomy for a vision system is a step in this direction. The system taxonomy, proposed in this work is based on the vision functions performed by vision systems. It can be used to classify various vision systems. It has been demonstrated that it has been possible to classify one system according to the proposed taxonomy and an evaluation has been conducted for it on a number of vision systems. The evaluation criterion for the system taxonomy is based on two parameters including the functions presence and the ordering of functions in the taxonomy, in relation to the actual system. The functions presence in the taxonomy has promising results. The taxonomy has either W or X grades for 95 percent of the investigated systems which means that either no task is missing or at most 2 functions are missing for these systems. However, there is a trade-off between functions presence and functions ordering. This is because; different systems have different function flows depending on the requirements and constraints while the functions for these systems might be similar. The taxonomy is fully/partially ordered for 50 percent of the investigated systems. The reason is that the focus of the system taxonomy is on the functionality of the systems rather than the flow of the functions. This taxonomy can be used to classify and compare different vision systems. The designers can propose architecture for one class of system and it can be beneficial for others wishing to implement a similar class of vision systems. It will assist the researchers to compare their system results with other systems. Moreover, this taxonomy will provide a realistic vision model and a benchmark for future vision systems.

REFERENCES

[1] Stephan Hengstler, "Stereo Vision in Smart Camera Networks", *Department of Electrical Engineering, Stanford University*, USA.

[2] C. S. Hong, R. Hornsey, On-Chip Binary Image Processing with CMOS Image Sensors, University of Waterloo, Waterloo, Ontario, Canada, www.cse.yorku.ca/visor/pdf/Hong_SPIE02_final.pdf

[3] V. Mehta, W. Sheng, T. Chen, Q. shi,"Development and Calibration of a Low Cost Wireless Camera Sensor Network". *The 2009 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Oct. 11-15, 2009 St. Louis, USA.

[4] F. Qureshi and D. Terzopoulos, "Smart Camera Networks in Virtual Reality", *Proceedings of the IEEE*, pp. 1640-1656, vol 96, Oct. 2008.

[5] Y. Shi, R. Taib and S. Lichma, "GestureCam: A Smart Camera for Gesture Recognition and Gesture-Controlled Web Navigation", *9th Intl. Conf. on Control, Automation, Robotics and Vision*, 2006, pp.1-6, Dec 2006.

[6] S. Fleck, F. Busch, P. Biber, W. Straßer, "3D Surveillance–A Distributed Network of Smart Cameras for Real-Time Tracking and its Visualization in 3D", *Conf. on Computer Vision and Pattern Recognition Workshop*, pp.118, June, 2006.

[7] Z. Ge, H.B. Lim, W.F. Wong, Memory Hierarchy Hardware-Software Co-design in Embedded Systems.2005. www.dspace.mit.edu/bitstream/handle/1721.1/7427/CS017.pdf

[8] B. Rinner and W. Wolf. "Multi-Camera Networks, chapter Toward Pervasive Smart Camera Networks", *Elsevier*, pp 483–496, 2009.

[9] A. W. Malik, B. Thörnberg, X. Cheng, N. Lawal, "Real-time Component Labelling with Centre of Gravity Calculation on FPGA", *The Sixth Intl. Conf. on Systems, St. Maarten*, 23-28 January, 2011.

[10] M. Imran, K. Khursheed, M.O'Nils & L. Lawal, "Exploration of Target Architecture for a Wireless Camera Based Sensor Node", *28th Norchip Conf.*, Tampere, Finland, pp. 1-4, 2010.

[11] T. Teixeira, D. Lymberopoulos, E. Culurciello, Y. Aloimonos, and A.Savvides, "A Lightweight Camera Sensor Network Operating on Symbolic Information", *Proc. of 1st Workshop on Distributed Smart Cameras*, 2006.

[12] A. Kerhet, M. Magno, F. Leonardi, A. Boni and L. Benini "A low-power wireless video sensor node for distributed object detection", *journal of real-time image processing*, vol.2, 2007.

[13] C. H. Zhiyong, L. Y. Pan, Z. Zeng, M.Q.-H Meng. "A Novel FPGA-Based Wireless Vision Sensor Node". *Proc. the IEEE Intl. Conf. on Automation and Logistics Shenyang*, China August 2009.

[14] W. Schriebl, T. Winkler, A. Starzacher, B. Rinner," A pervasive smart camera network architecture applied for multi-camera object classification", *Third ACM/IEEE International Conference on Distributed Smart Cameras*, pp.1-8, Sept 2009.

[15] M. Bakkali, R. Carmona-Galá, A. Rodriguez-Vazquez, "A prototype node for wireless vision sensor network applications development", *5th Intl Symposium on I/V Communications and Mobile Network (ISVC),* 2010.

[16] R. C. Gonzalez, R. E. Woods, "Digital Image Processing", book, Ed. 2008.

[17] S. V. Vaseghi "Advanced Digital Signal Processing and Noise Reduction" 2nd Ed. J.Wiley & Sons.

[18] J. Astola and P. Kuosmana, "Fundamentals of Nonlinear Digital Filtering" CRC Press,LLC, 1997.

[19] Y. Meng, "A Mobile Vision System with Reconfigurable Intelligent Agents", *Intl. Joint Conf. on Neural Networks*, 2006.

[20] J. M. Perez, P. Sanchez, M. Martinez, "Low-cost Bayer to RGB bilinear interpolation with hardware-aware median filter" , *16th IEEE Intl. Conf. Electronics, Circuits, and Systems*, 2009. pp. 916 - 919 , Tunisia.

[21] X. Cheng, B. Thörnberg, W. Malik, "Optimized Color Pair Selection for Label Design", *53rd Intl. Symposium*, Sept 2011 .

[22] S. Rein, M. Reisslein, "Low-Memory Wavelet Transforms for Wireless Sensor Networks: A Tutorial", *IEEE* Communications Surveys & Tutorials, pp.291-307, 2011

[23] N. Alt, C. Claus, W. Stechele, "Hardware/software architecture of an algorithm for vision-based real-time vehicle detection in dark environments", *Proceedings of the conf. on Design, automation and test in Europe*, NY, USA, 2008.

[24] Z. Zivkovic, "Wireless smart camera network for real-time human 3D pose reconstruction", *Computer Vision and Image Understanding* , vol. 114, NY, USA, Nov, 2010.

[25] A. N. Belbachir, Chapter 16. Smart Cameras, Springer, pp 295-296.

[26] Linescan Imaging Basics - Teledyne DALSA, Available :www.teledynedalsa.com

[27] M. Barth, C. Barrows, "A fast panoramic imaging system and intelligent imaging technique for mobile robots", Proc. of the 1996 IEEE/RSJ Intl. Conf. on *Intelligent Robots and Systems*, pp 626 – 633, vol.2, Aug 2002.

[28] Z. Xu, X. Zhao, N. Li , "Novel vehicle detection system based on line scan camera", *9th Intl. Conf. on Electronic Measurement & Instruments*, pp 1-748 - 1-751, Aug 2009.

[29] Y. Minghua, C.Sharp, L.S.DeBrunner, V:E. DeBrunner, "Image processing for a line-scan camera system", *Conf. Record of the Thirtieth Asilomar Conf. on Signals, Systems and Computers*, pp: 130 - 134 vol.1, 1996.

[30] L. Se-Na, Y. Bum-Jae, L. Mee-Seub, O. Sang-Rok, H.Song-Soo, L. Sang Heon, "Visual measurement of pile penetration and rebound movement using a high-speed line-scan camera", *IEEE Intl. Conf. on Robotics and Automation*, 2002. pp 4307 – 4312, vol.4, 2002.

[31] IGOR Pro, version 4.0, chapter 22, pp 736-752.

[32] S. Fleck, S. Lanwer, and W. Straßer, "A smart camera approach to real-time tracking," *13^{th} European Signal Processing Conf.* (EUSIPCO 2005), September 4-8 200.

[33] Z. Xiong, K. Ramchandran, M. T. Orchard and Y. Zhang, "A comparative study of DCT- and wavelet-based image coding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol 9, pp. 692-695, 1999.

[34] I. Pitas, A. N. Venetsanopoulos, "Order statistics in digital image processing", Proceedings of the IEEE, vol 80, pp. 1893-1921, Dec. 1992.

[35] Z. Yu, Y. Zhao, X. F. Wang, "Research advances and prospects of mathematical morphology in image processing", *IEEE Conf. on Cybernetics and Intelligent Systems*, Sept 2008.

[36] Hamed Fatemi, "Processor Architecture Design for Smart Cameras", PhD thesis, Technische Universiteit Eindhoven, 2007.

[37] B. Rinner, T. Winkler, W. Schriebl, M. Quaritsch, W. Wolf, "The evolution from single to pervasive smart cameras", *ACM/IEEE Intl. Conf. on Distributed Smart Cameras*, pp.1-10, Sep 2008.

[38] J. Scott, M. Pusateri, "Towards real-time hardware gamma correction for dynamic contrast enhancement", *IEEE Applied Imagery Pattern Recognition Workshop*, pp. 1-5, oct 2009

[39] I. Downes, L. Baghaei, H. Aghajan," Development of a Mote for Wireless Image Sensor Networks", in *Proc. COGnitive systems with Interactive Sensors* (COGIS 2006), Mar. 2006.

[40] R.Virrankoski and S. Keskinen, "GENSEN: A Novel Combination of Product, Application and Technology Platform Development in the Context of Wireless Automation," in *Proc. of the 14th Intl. Conf. on Productivity & Quality Research (ICPQR 2009)*, Alexandria, Egypt, 2009.

[41] P. Kulkarni, D. Ganesan, P. Shenoy, Q. Lu. "SensEye, A multi-tier camera sensor network". *Intl Multimedia Conf.* archive *Proc. the 13th annual ACM intl. conf. on Multimedia*, pp. 229 – 238, 2005.

[42] S. Hengstler, D. Prashanth, F. Sufen, H. Aghajan, "MeshEye: A Hybrid-Resolution Smart Camera Mote for Applications in Distributed Intelligent Surveillance", 6th Intl. Symposium on *Information Processing in Sensor Networks*, 2007.

[43] S. Soro and W. Heinzelman, "A survey of visual sensor networks", *Hindawi Publishing Corporation Advances in Multimedia*, Article I D 640386, vol 2009.

[44] A. Rowe, A. Goode, D. Goel, I. Nourbakhsh, "CMUcam3: An Open Programmable Embedded Vision Sensor", www.cmucam.org.

[45] A. Orfy, A. El-Sayed, M. ElHelw, "WASP: Wireless autonomous sensor prototype for Visual Sensor Networks", *IFIP Wireless Days (WD)*, 2010.

[46] L. Gasparini, R. Manduchi, M. Gottardi, D. Petri, "An Ultralow-Power Wireless Camera Node: Development and Performance Analysis", *IEEE Transactions on Instrumentation and Measurement*, vol PP, pp1-9, 2011.

[47] A. Kandhalu, A. Rowe, R. Rajkumar, "DSPcam: A Camera Sensor System for Surveillance Networks", *Third ACM/IEEE Intl. Conf. on Distributed Smart Cameras*, ICDSC 2009.

[48] Z. Y. Cao, Z. Z. Ji, M. Z. Hu, "An image sensor node for wireless sensor networks", *Intl. Conf. on Information Technology Coding and Computing*, vol. 2, pp. 740 - 745 2005.

[49] L. Ferrigno, S. Marano, V. Paciello, A.Pietrosanto. "Balancing computational and transmission power consumption in wireless image sensor networks*". IEEE Intl. Conf. on Virtual Environments, Human-Computer Interfaces, and Measurement Systems*, Italy, July 2005.

[50] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterl "SensorScope: Out-of-the-Box Environmental Monitoring", *Proc.of the 7th Int. conf. on information processing in sensor networks*, pp.332-343, 2008.

[51] W. Chen, H. Aghajan, R. Kleihorst, "Real-Time Human Posture Reconstruction in Wireless Smart Camera Networks", *Intl. Conf. on Information Processing in Sensor Networks*, ST.Loius, USA. pp.321-331, April 2008.

[52] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Wrior, D. Estrin, M. Srivastava, "Cyclops: in situ image sensing and interpretation in wireless sensor networks", *Proc. of the 3rd intl. conf. on Embedded networked sensor systems*, 2005.

[53] A. Rowe, D. Goel and R. Rajkumar, "FireFly Mosaic: A vision-enabled wireless sensor networking system", *Proc. of the 28th IEEE Int'l Real-Time Systems Sympo*. pp. 459-468, 2007.

[54] Lidholm, Jörgen, Licentiate thesis, School of Innovation, Design and Engineering, Mälardalen University, Sweden, 2011.

[55] F. Pelissier and F. Berry, "Design of a real-time embedded stereo smart cameras", *Advanced concepts for intelligent vision systems*, vol. 6474/2010.

[56] M. Al-Fandi, M. A. Jaradat, A. Abusaif, T. C. Yih, "A real time vision feedback system for automation of a nano-assembly manipulator inside scanning electron microscope", *7th Intl. Multi-Conf. on* Systems Signals and Devices (SSD), pp. 1-5, June 2010.

[57] S. Yang, L. Chen, "A face and eye detection based feedback system for smart classroom", *Intl. Conf. on Electronic and Mechanical Engineering and Information Technology*, pp. 571 – 574, Aug 2011.

[58] M. Casares, S. Velipasalar, "Adaptive Methodologies for Energy-Efficient Object Detection and Tracking With Battery-Powered Embedded Smart Cameras", *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1438 – 1452, Oct 2011.

[59] www.micron.com/imaging

[60] N.R. Pal and S.K. Pal, "A Review on Image Segmentation Techniques", *Pattern Recognition*, vol. 26, no. 9, pp. 1,277–1,294, Sept. 1993.

[61] J. T. Feddema, C. S. G. Lee, O. R. Mitchell, "Weighted selection of image features for resolved rate visual feedback control", *IEEE Transactions on Robotics and Automation*, pp.31-47, Feb 1991.

[62] I. Ishii, Y. Nakabo, M. Ishikawa, " Target tracking algorithm for 1 ms visual feedback system using massively parallel processing", *IEEE International Conference on Robotics and Automation*, pp.2309-2314, April 1996.

[63] P. Chen, P.Ahammad, C.Boyer, S.Huang, L.Lin, E.Lobaton, M.Meingast, S.Oh, S.Wang, P.Yan, A.Y. Yang, C.Yeo, L.Chang, J.D.Tygar, and S.S.Sastry "CITRIC: A low-bandwidth wireless camera network platform", *Second ACM/IEEE International Conference on Distributed Smart Cameras*. pp.1-10, Sept. 2008.

[64] C. H. Zhu, K. Hirahara, K. Ikeuchi, "Street-parking vehicle detection using line scan camera", *IEEE Proceeding Intelligent Vehicles Symposium*, pp.575-580, June 2003.

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

[66] W. Wolf, B. Ozer, T. Lv, "Smart cameras as embedded systems", *computer*, pp.48-53, Sept 2002.

[67] C. Stauffer, W. E. L. Grimson, "Adaptive background mixture models for real-time tracking", *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 1999.

[68] G. Dongfa, J. Zhuolin, Y. Ming, "A New Approach of Dynamic Background Modeling for Surveillance Information", Intl. Conf. on Computer Science and Software Engineering, pp. 850-855, 2008.

[69] J. C. Russ, "The Image Processing Handbook", Fifth Edition, 2006.

[70] C. Micheloni, B. Rinner, G. L. Foresti," Video Analysis in Pan-Tilt-Zoom Camera Networks", *IEEE Signal Processing Magazine*, pp.78-90, 2010.