International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.3, Issue No.3, pp : 216-221 1 march 2014

The Research of Smart Surveillance System Using Hadoop Based On Craniofacial Identification

Venkata Lakshmi.K,Venkateswaran.S
Department of Computer Science Engineering
Arasu Engineering College,Kumbakonam
Corresponding Email: venkatalakshmi.aec@gmail.com

Abstract—Systems operating in a distributed environment need to maintain high standards regarding availability and performance. A decretive concern in distributed computing systems is to efficiently schedule the tasks among all processors so that the overall processing time of the submitted tasks is at a minimum. The increasing need for intelligent visual surveillance in commercial, law enforcement and military applications makes automated visual surveillance systems one of the main current application domains in computer vision. Parallel Hadoop implementation is better suited for large data sizes than for when a computationally intensive application is required. In this paper, we propose a probabilistic approach for face recognition suitable for a multi-camera video surveillance network using Hadoop platform. This proposed design of a hybrid intelligent surveillance system which the face detection and tracking is required for Video Surveillance to capture people. The surveillance deals with the extraction of face datasets from a video and processing of images using a Hadoop image processing interface and craniofacial identification based facial feature extraction.

Keywords— Image processing, MapReduce, Hadoop, distributed file system, CCTV, HDFS

I. Introduction

Video surveillance is a type of surveillance that is found in different kinds of areas, such as public buildings, metro stations and military areas. It has proven to be an effective tool to monitor large areas with limited resources. Because cameras and surveillance systems are continuously developed, these systems become increasingly cost efficient, allowing for larger systems as well.

Traditionally, multiple static security cameras are positioned throughout the area, and attached to monitor screens. These screens are monitored by security personnel in order to detect suspicious behaviour. Depending on the seriousness of the situation, further action may be taken in order to resolve it. The main problem with the current system lies within the nature of the tasks of the security personnel. They must passively watch multiple monitor screens simultaneously. Humans easily get tired and lose concentration, especially in situations when no situations of notice occur for a long time. At regular times the guard will inspect the area in person, leaving the cameras unattended and being exposed to potential attackers. Furthermore, humans are not capable of noticing every small detail, let alone of focusing on multiple things at the same time. A second problem is the fact that camera footage of some areas is being recorded without supervision of a human operator; simply because it would be too expensive to constantly monitor that area. The data is stored as evidence in case some event occurs. This prevents incidents from being detected in a timely manner, preventing swift responses to potentially dangerous situations. Moreover, it is time-consuming to find the correct video images, especially when the event occurred many hours before it is detected, and the system consists of many different cameras. By using computers to support the security personnel in their task to watch the monitor screens, some of these problems may be prevented. Computers, unlike humans, are capable of working continuously, without losing focus. Moreover, computers are well scalable to allow for monitoring a large number of security cameras simultaneously. Additionally, computers can communicate constantly, allowing each computer to have a complete and up-to-date view of the monitored area

Video surveillance systems traditionally consist of cameras attached to monitor screens. These systems are installed to give an overview of a large area to a limited number of operators. The goal is to detect abnormal situations. Depending on the seriousness of the situation, action can be taken. If incidents happen, they warn the security or police. Some monitors show the video stream of a single camera and some show multiple streams on a single monitor simultaneously or sequentially. However, in some areas the monitors are not watched constantly.

Video recorders record the output of each camera. After an incident, the video footage can be used as evidence. One obvious disadvantage of this approach is that operators are not able to prevent incidents or limit their damage, since the videos are only watched afterwards. Another disadvantage is that it takes a significant amount of time to search for the right video images, especially when the suspect arrives at the scene hours before the incident and a large amount of cameras are involved. Thus, in this paper we have a distributed approach to solve abnormal activities by Hadoop and craniofacial identification surveillance applications.

II. CCTV design for smart video surveillance system

Modelling Camera view area are categorized into three different view area such as camera view area, Projections of view area, Camera positions.

a. Camera view area

View area is a three-dimensional geometrical pyramid-shaped figure (a convex tetrahedral angle) with the vertex, starting from a camera lens. All objects (or parts of objects) inside the pyramid will be visible on the screen, when it is not shaded by other objects on the scene. The objects outside the pyramid will not be visible.

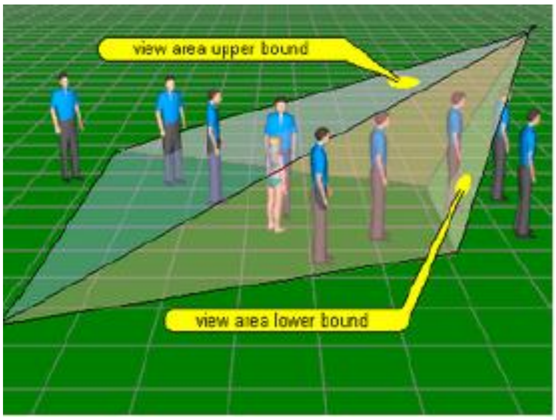

Figure 2.1 Camera view area. Side view

View area can be infinite or limited by ground and other objects. Angles between view area faces are calculated automatically on the base of lens focal length and image sensor format. the top side of this pyramid, which corresponds to the top image border on the screen "view area upper bound", and the bottom side of the pyramid, which corresponds to the bottom border of the screen we will name "view area lower bound".

b. Projections of view area

Horizontal projection of a view area is determined by the following key parameters such as height of view area upper bound, height of view area lower bound and view area upper bound distance. Changing values of these heights, we'll get different sizes of projection, and any object, which is at the height between these bounds, and on a horizontal plane within the limits of horizontal projection of view area, will be visible on the screen, when it is not shaded by other objects on the scene.

c. Camera Positions

Camera position is not determined by the height of installation and angle of slope, but by the height of installation, height of a view area upper bound and view area upper bound distance. Thus, in order to get the sizes and position of a view area projection in relation to a camera, it is necessary to set the following parameters : Image sensor format and a lens focal length, height of the camera installation, heights of the upper and the lower bounds of view area,view area upper bound distance.

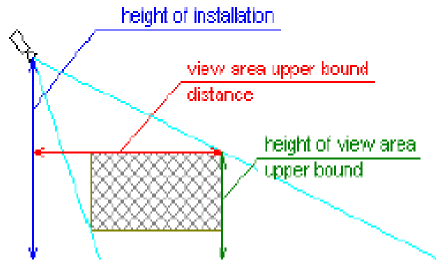

Figure 2.2 Definition of the camera position

III. Hadoop Image Processing Framework

Hadoop provides a distributed file system and a framework for the analysis and transformation of very large data sets using the MapReduce paradigm. An important characteristic of Hadoop is the partitioning of data and computation across many (thousands) of hosts, and executing application computations in parallel close to their data. A Hadoop cluster scales computation capacity, storage capacity and IO bandwidth by simply adding commodity servers. Hadoop clusters at Yahoo! span 25 000 servers, and store 25 petabytes of application data, with the largest cluster being 3500 servers. One hundred other organizations worldwide report using Hadoop.

Hadoop image processing framework was created to empower researchers and present them with capable tool that would enable research involving image processing and vision to be performed extremely easily. With the knowledge that Hadoop framework would be used for researchers and as an educational tool with features to provide an open, extendible library for image processing and computer vision applications in a MapReduce framework, allow for simple filtering of a set of images, present users with an intuitive interface for image-based operations and hide the details of the MapReduce framework and also will set up applications so that they are highly parallelized and balanced so that users do not have to worry about such details.

A. Parallel and Distributed Processing on Hadoop

As the structure of the system, Hadoop consists of two components, the Hadoop Distributed File System (HDFS) and MapReduce, performing distributed processing by single-master and multiple-slave servers. There are two elements of MapReduce, namely JobTracker and TaskTracker, and two elements of HDFS, namely DataNode and NameNode. In Figure 1, the configuration of these elements of MapReduce and HDFS on Hadoop are indicated. There is also a mechanism that checks the metadata for NameNode.

(a) JobTracker

JobTracker manages cluster resources and job scheduling to and monitoring on separate components.

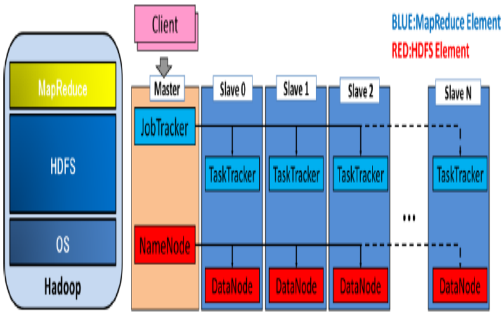

Figure 3.1 Structure with elements of MapReduce and HDFS

(b) TaskTracker

TaskTracker is a slave node daemon in the cluster that accepts tasks and returns the results after executing tasks received by JobTracker.

(c) NameNode

An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. NameNode executes file system name space operations, such as opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes.

(d) DataNode

The cluster also has a number of DataNodes, usually one per node in the cluster. DataNodes manage the storage that is attached to the nodes on which they run. DataNodes also perform block creation, deletion, and replication in response to direction from NameNode.

(e) SecondaryNameNode

SecondaryNameNode is a helper to the primary NameNode. Secondary is responsible for supporting periodic checkpoints of the HDFS metadata.

B. Hadoop Distributed File System (HDFS)

HDFS is designed to reliably store very large files across machines in a large cluster. It is inspired by the Google File System. HDFS is composed of NameNode and DataNode. HDFS stores each file as a sequence of blocks (currently 64 MB by default) with all blocks in a file the same size except for the last block. Blocks belonging to a file are replicated for fault tolerance. The block size and replication factor are configurable per file. Files in HDFS are write-once and can have only one writer at any given time.

C. MapReduce

MapReduce (implemented on Hadoop) is a framework for parallel distributed processing large volumes of data. In programming using MapReduce, it is possible to perform parallel distributed processing by writing programs involving the following three steps: Map, Shuffle, and Reduce. Figure 2 shows an example of the flow when Map and Reduce processes are performed. Because MapReduce automatically performs inter-process communication between Map and Reduce processes, and maintain load balancing of the processes.

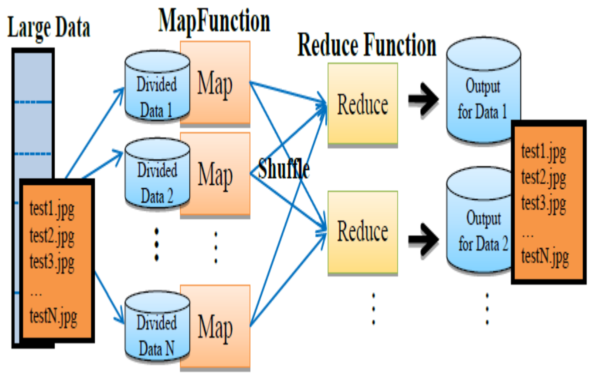

Figure 3.2 Processes performing the map and reduce phases

a. Map concept of data processing

The Map function takes a key-value pair <K, V> as the input and generates one or multiple pairs <K′, V′ > as the intermediate output.

b. Shuffle concept of data processing

After the Map phase produces the intermediate key-value pair or key-value pairs, they are efficiently and automatically grouped by key by the *Hadoop* system in preparation for the Reduce phase.

c. Reduce concept of data processing

The Reduce function takes as the input a <K′, LIST V′ > pair, where "LIST V′ " is a list of all V′ values that are associated with a given key K′. The Reduce function produces an additional key-value pair as the output.

By combining multiple Map and Reduce processes, we can accomplish *complex* tasks which *cannot* be done via a *single Map and Reduce* execution. Figures 3 and 4 respectively show the key-value data model and a Wordcount example of MapReduce.

map: <key, value> ⇒ list <key', value'>
shuffle: list <key', value'> ⇒ {<key'', list(value'')>}
reduce: {<key'', list(value'')>} ⇒ list(value''')

Figure 3.3 Key-value data model of MapReduce

D. MapReduce for Video Database Processing

The Map and Reduce functions of *MapReduce* are both defined with respect to data structured in key-value pairs. In short, we can perform distributed processing by creating key-value pairs in *MapReduce form. However, for* unstructured data such as video data, it can be assumed that it is more difficult to create key-value pairs and perform the processing than processing structured data. In this experiment, in order to use the Ruby programming language, we *utilize* an extension package of Hadoop, namely Hadoop Streaming. With a view to performing parallel distributed processing on MapReduce forms, we need to create programs to be used as Map and Reduce functions in the Ruby programming language. Video database processing is

International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.3, Issue No.3, pp : 216-221 1 march 2014

performed by splitting the data in a video database and creating key-value pairs. For example, the frame number can be used as a key for a video frame. In the case of parallel processing of a video frame, the video frame is divided into multiple parts, and the part numbers can be the keys (identifiers) for these different parts. Sorting is carried out using the key number, and joining separated frames or separated parts are performed by the Reduce function. Figure 6 shows an example of processing flow using MapReduce. In this figure, each video frame is divided into four parts, and each part has a unique key number.

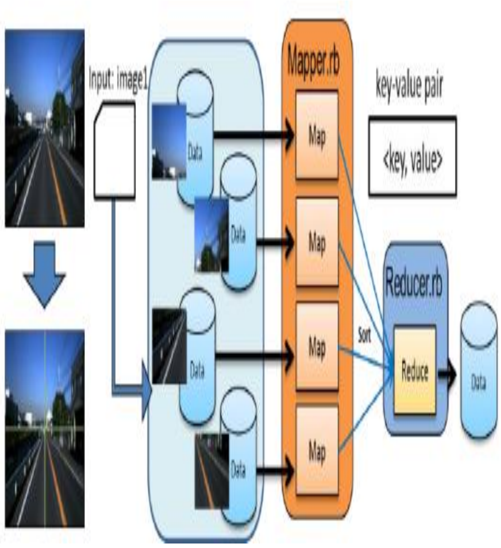

Figure 3.4 Image processing flow using MapReduce

E. MapReduce Processing of Video Database

We implemented the following video database processing steps using Hadoop Streaming. Multiple sequential video frames are input, and image processing is performed for each video frame. We implemented a Map function for image processing. The input of the Map function is a single video frame, and the Map function produces one video frame as output.

We process video frames in parallel with the slave servers, which use HDFS. The video database is stored in HDFS. Each Map process also outputs to HDFS. Figure 7 shows an example of video database processing flow using the Map function and HDFS.

We first create a grayscale image of the original image in parallel by using MapReduce. Then, features of the grayscale image are extracted in parallel by using MapReduce.

Hadoop is composed of a master server which manages slave servers, which perform the actual image processing. Master and slave servers actually run on the same server for this configuration of the MapReduce system (pseudo-distributed mode). The number of copies of video data is set to 1. The parallel processing of the video database using Hadoop Streaming is distributed over all of the cores of a CPU of a single machine.

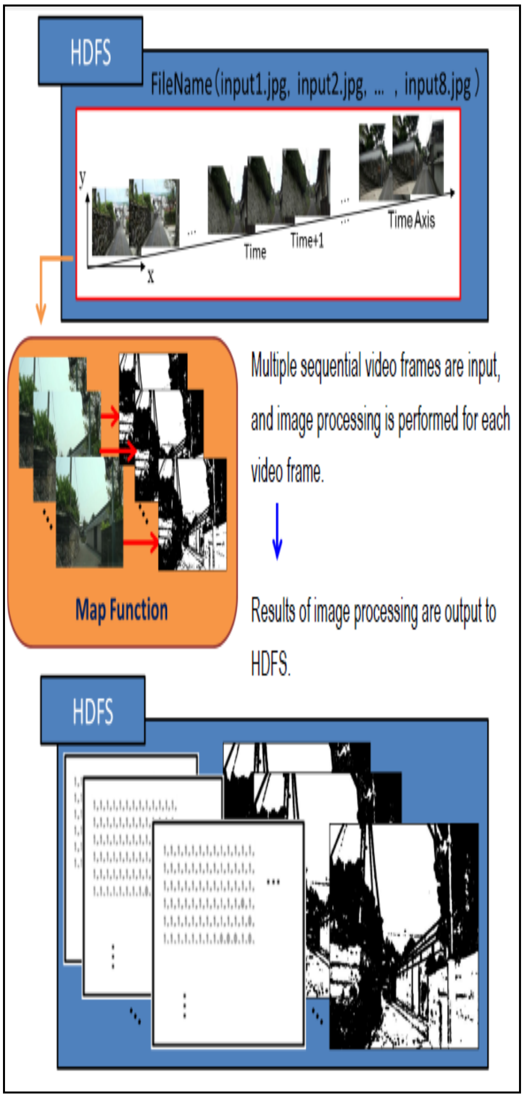

Figure 3.5 Sequential video frame processing flow using Map and HDFS

.

F. Feature Extraction of Video Images by Using MapReduce

In this paper, we describe using a video database of video collected with a video camera. In an experiment, we processed sequences of video frames with MapReduce to create greyscale images and extracted some features of the video images. In the process of creating the greyscale images, each video frame was divided into multiple parts.

In the extraction, frame numbers were used as the key numbers to extract some features of the video images. In future, it is necessary for us to build a distributed environment that combines multiple machines, conduct large-scale experiments involving sequential video images.The advantage of the framework are an isolation of the details of the parallel execution from the application software developer by providing simple API to work with the image, which is loaded into memory.

International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.3, Issue No.3, pp : 216-221 1 march 2014

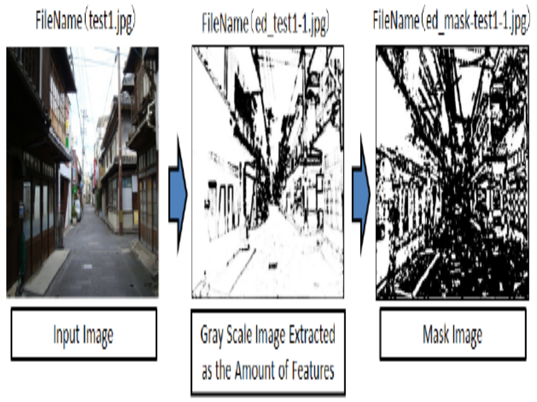

Figure 3.6 Examples of an input image, feature extraction image, and output image. The output image is used as the mask image

IV. Craniofacial Identification based surveillance system

Craniofacial identification methods are used to assist the identification of skeletal remains via the analysis of skulls and faces. This involves the overlay of a face on a skull in an attempt to determine if a match exists between the two (craniofacial superimposition), or the prediction of a face from the skull (facial approximation). Both methods are underpinned by knowledge of human anatomy, despite their use of different technical protocols. Craniofacial superimposition and facial approximation are subsumed within the discipline of physical anthropology. Although they contribute to the identification process; craniofacial superimposition and facial approximation are not generally used to provide identifications in a stand-alone manner. Instead, they are used to help generate information that can be investigated using higher powered methods.

Consequently, the term "craniofacial identification" does not provide a literal description of the methods but rather a convenient short-hand summary of their basis and their realm of application. This is similar to use of "skeletal identification" to describe basic forensic anthropology techniques (e.g., Kerley, 1978), including biological profile assessments, since these methods rarely hold the potency to produce identifications when open and/or large samples of individuals are considered. Superimposition offers the most reliable information (particularly in closed samples where the skeletal remains are a priori known to belong to one of only a few individuals), but in many cases superimposition methods are used as exclusion tools. Facial approximation methods are, at present, much poorer indicators of a person's identity and their results are suggestive at best. Both facial approximation and superimposition methods developed from early attempts to verify the identities of skeletons thought to represent well-known historical figures (examples include Bach & Schiller (see Welcker, 1883; Welcker, 1888)). Such analyses were originally achieved by comparing skulls or constructed faces to artistic portraits rather than images of living people.

A. Craniofacial Superimposition

Superimposition refers to the process of overlaying upon one image, a semi-transparent version of another image. This method can be used with regards to skulls and faces or it can be used in relation to other infracranial body regions and may only involve the comparison of skeletal elements. Craniofacial superimpositions were initially conducted using tracings of skulls and faces made from photographs. Video-superimposition is now the most frequently employed method, and involves mixing a video image of a skull with a photograph of the person in question to determine the degree of anatomical correspondence.

B. Facial Approximation

Facial approximation methods have also been known by many other names previously. The most popular of these is “facial reconstruction”. There are two main problems with this term: i) the face is not reassembled from existing parts as reconstruction implies; and ii) reconstruction overemphasizes the exactness, reliability and scientific validity of the methods. Most practitioners now acknowledge that the term "facial approximation" is the most appropriate name for the prediction of facial appearances from the skull, although some continue to avoid its use due to its less sensational nature. The suitability of "facial approximation" is perhaps best demonstrated that has been summarized the aim of the method as "approximation so close to the appearance of a living person that even an unknown individual could be identified".

V. Conclusion

One of the primary advantages of this system is its ability to make the process of integrating technologies into Hadoop and Craniofacial identification. The use of a database to index events opens up a new area of research in context based exploitation of smart surveillance technologies. This will be one of the key future directions for our research. Additionally, this system will be deployed in a variety of application environments including homeland security, retail, casinos, manufacturing, mobile platform security etc. Each new environment brings with it challenges both in the core video analysis domain and in the indexing and event interpretations domains. The Hadoop image processing framework has implemented create a tool that will make development of large-scale image processing and vision projects extremely accessible in hopes that it will empower researchers and students to create applications. This paper is to propose an enhanced video surveillance system with a powerful feature extraction based on craniofacial identification and extended Hadoop image processing framework implementation to provide a format for storing images for efficient access within the MapReduce pipeline which are most useful for image processing and vision applications.

References

1. Le An, Mehran Kafai, and Bir Bhanu," Dynamic Bayesian Network for Unconstrained Face Recognition in Surveillance Camera Networks", Emerging and Selected Topics in Circuits and Systems, IEEE Journal on (Volume:3 , Issue: 2) , Page(s):155 – 164,ISSN : 2156-3357,June 2013.

2. Navneet Jindal Vikas Kumar," Enhanced Face Recognition Algorithm using PCA with Artificial Neural Networks", International Journal of Advanced Research in Computer Science and Software Engineering(Volume 3, Issue 6),ISSN: 2277-128X,June 2013

3. A Fakhri A Nasir, M Nordin A Rahman and A Rasid Mamat , A Study of Image Processing in Agriculture Application under High Performance Computing Environment, International Journal of Computer Science and Telecommunications [Volume 3, Issue 8, August 2012].

4. Kolhandai Yesu, Kaveri Chetia and Himadri Jyoti Chakravorty.'' Innovative Feature Extraction Method for Artificial Neural Network Based Face Recognition" 978-1-4577-0748-3/12,IEEE- 2012.

5. Florin Dinu,T. S. Eugene Ng," Hadoop's Overload Tolerant Design Exacerbates Failure Detection and Recovery", ACM 978-1-4503-0654-6/11/06,June 2011.

6. Mayank Agarwal, Nikunj Jain, Mr. Manish Kumar and Himanshu Agrawal,"Face Recognition Using Eigen Faces and Artificial Neural Network", International Journal of Computer Theory and Engineering, Vol. 2, No. 4,1793-8201,August 2010.

7. N. Poh, C. H. Chan, J. Kittler, S. Marcel, C. McCool, E. R´ua, J. Castro,M. Villegas, R. Paredes, V. ˇ Struc, N. Paveˇsi´c and, A. Salah, H. Fang, and N. Costen, "An evaluation of video-to-video face verification,"Information Forensics and Security, IEEE Transactions on, vol. 5, no. 4,pp. 781 –801, Dec. 2010.

8. M.Nandini, P.Bhargavi, G.Raja Sekhar,"Face Recoginition using Neural Networks", International Journal of Scientific and Research Publications, Volume 3, Issue 3, March 2013 1 ISSN 2250-3153.

9. Mark W. Powell, Ryan A. Rossi† and Khawaja Shams ,"A Scalable Image Processing Framework for Gigapixel Mars and Other Celestial Body Images", IEEE.2 IEEEAC paper #1533, Version 1,978-1-4244-3888-4, August 2009.

10. Dominik Brunner, Guido Lemoine , Francois-Xavier Thoorens, and Lorenzo Bruzzone, "Distributed Geospatial Data Processing Functionality to Support Collaborative and Rapid Emergency Response", IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing, Vol. 2, No. 1, March 2009.

11. J. Dean, S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In Communications of the ACM - 50th anniversary issue: 1958 – 2008, Volume 51 Issue 1, January 2008, Pages 107-113. http://research.google.com/archive/mapreduce.html, doi: 10.1145/1327452.1327492

12. Jian Zhao and Sen-ching S. Cheung," Multi-Camera Surveillance with Visual Tagging and Generic Camera Placement", First ACM/IEEE International Conference on Distributed Smart Cameras 1-4244-1354-0/07/200, September 2007.

13. Duarte Duque, Henrique Santos and Paulo Cortez, "Prediction of Abnormal Behaviors for Intelligent Video Surveillance Systems",IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007),April 2007.

14. Conner, J. 2009. Customizing input file formats for image processing in hadoop. Arizona State University. Online at: http://hpc. asu. edu/node/97.