Identification of running Vehicle using Vision Based Holistic Property of Registration Technique for Traffic Surveillance in India

Prachi Goyal
Department of Computer Science and Engineering
Noida Institute of Engineering and Technology
Greater noida, Uttar Pradesh, India

Chitvan Gupta
Department of Computer Science and Engineering
Noida Institute of Engineering and Technology
Greater noida, Uttar Pradesh, India

Abstract- The fundamental step of moving object in any image is visualize in different vision-systems including automated visual surveillance and conventional systems. Mostly in India, traffic system related to small geographic areas are more complex to handle. So our objective is to control the congestions and monitor the traffic activities by detecting the vehicle by using vision-based holistic registration technique, which will be helpful to predict the traffic flow. Many researchers apart from our country are doing more and more research and experiments to control and monitor the existing surveillance system. Some existing algorithm for vision-based holistic properties registration are discussed in this paper and we use this hybrid technique to develop this surveillance system based on monocular image sequences. In this paper we proposed a working model of such surveillance framework for detection and identifying the high speed vehicles and traffic flow measurement. The method is natural which is based on the establishment of interfaces between regions and vehicles, while moving through the image sequencing. Due to Background subtraction technique it improves the adaptive background mixture model and makes the system learn faster and more accurately. For the experiments and results Real-life traffic image sequences have been taken measure the effectiveness of the proposed algorithm.

Keywords – Pattern Recognition, Object Recognition and Motion, Neural Network Applications.

I. INTRODUCTION

Intelligent vision systems for urban traffic surveillance have been adopted more frequently. As can be seen in [3], [27] and [14], there are several works related to the traffic video analysis. The traditional approaches are based on detection and counting of individual vehicles. Basically each vehicle is segmented and tracked [3], and its motion trajectory is analyzed to estimate traffic flow [13], vehicle speed[22] and parked vehicles [1]. Some authors also performvehicle classification using blob dimension [7], 3D models[2], shape and texture features [20]. However, most of theexisting work commonly fails on crowded situations (e.g.traffic congestion) due to the large occlusion of moving objects [11]. Basically, in traditional approaches the traffic isclassified by the number of detected vehicles in the scene, but in very crowded scene

the blobs of two or more vehicles may overlap causing error on the vehicle counting.

Therefore, the accuracy of traditional methods tends to decrease as the density of moving objects (e.g. people or vehicles) increases [30, 11]. Alternative methods for dealingwith this problem gave rise to a new field of study called crowd analysis. Crowd behavior analysis of moving objects is an important research field in computer vision. As described in[11], there are two approaches to perform behavioral analysis of crowded scenes. Object-based approaches try to infer crowd behavior by analyzing individual elements of the scene [11]. A typical example is the tracking of some particular individuals to analyze the group behavior. On the other hand, “holistic” approaches evaluate the crowd as an individual entity [11]. Holistic approaches try to obtain global information, such as crowd flows, and skip local information (e.g. single vehicle against the flow) [11, 24]. Some holistic properties can be extracted from crowds behavior analysis like crowd density, speed, localization and direction [25, 30, 24]. It is often hard to track particular objects in crowded scene, so the holistic approaches is more suitable [30, 11]. Some related works that apply holistic methods for traffic and pedestrian behavior analysis can be found in [21, 4, 25, 16, 24, 5]. In [21], the authors have classified traffic videos in five classes (empty, open flow, mild congestion, heavy congestion and stopped) using Gaussian Mixture Hidden Markov Models (GM-HMM) trained from features extracted of MPEG data (DCT coefficients and motion vectors). In [4], Chan and Vasconcelos propose to model traffic flow as dynamic texture, defined as an autoregressive stochastic process, which encode the spacial and temporal components into two probability distri butions. In [25], the authors model crowd events based on crowd movements and it directions to estimate normal andabnormal behaviors. In [16], the authors use the histogram of optical flow vectors to detect anomalies in the urban traffic flow. In [24], the authors model crowd events (e.g. merging, splitting and collision) based on crowd tracking and density based clustering using optical flow motion vectors. Recently, [5] suggests a Spatiotemporal Orientation Analysis to classify traffic flow patterns. The authors have shown that classification is based on matching distributions of space-time orientation structure and demonstrated improved results of Chan and Vasconcelos [4] approach.

This paper proposes a method to classify traffic patterns based on holistic approach (see Figure 1). The method classifies the traffic into three classes (light, medium or heavy congestion) by usage of average crowd density and speed of vehicle crowd. The heavy congestion is represented by a high crowd density and low (or zero) crowd speed. Otherwise, when crowd density is low and crowd speed is high, the system consider that the traffic has light congestion. In intermediate situations, the traffic is classified as medium congestion. In the next sections, pro reduces to estimate the crowd density and speed of vehicles based on crowd segmentation and tracking are introduced. Also, the method of feature extraction and classification used in this work are described. Finally, in the last section, the experimental results are shown as well as conclusions.

In this paper the same data set of [5] and [4] are being used, to allow comparing the results of the proposedapproach with those two previous works.

1 Crowd Segmentation

Crowd density can be estimated from background subtraction [12], optical flow [8] and fourier analysis [10]. In the present work, we estimate the vehicle crowd density by background subtraction process. Firstly, five recently background subtraction methods with Change Detection net video database are evaluated (described later). The BGS method with highest score is used to perform the crowd density estimation. In sections that follows, the background subtraction evaluation step and the process of how to estimate the crowd density are shown. 2.1 Background Subtraction Evaluation To perform background subtraction (BGS), five recent methods have been evaluated with Changedetection.netvideo database 1 . This data set contains 31 real-world videos totalling over 80.000 frames of indoor and outdoor scenes separated into six categories which include diverse motion and change detection challenges (e.g. light varia tions, shadows, camera jitter and others). Here three videos of each category have selected (giving priority to traffic scenes). All BGS methods have been set with default parameters defined in each work. As can be seen in Table 1,the Multi-Layer method proposed by Yao and Odobez [28]had the best score (Avg. Ranking) (highligted in yellow the cells in red represent the best scores for each metric). Therefore, the Multi-Layer method is used to performthe crowd segmentation. Figure 3 shows the crowd segmentation over highway traffic videos using UCSD video database (described later). The data set provides handlabeled traffic patterns (light, medium and heavy congestion) for each video.

1.1Crowd Density Estimation

The crowd density step is performed after the background subtraction process. In very crowded scenes, like congestion, many vehicles can be stationary for a long time. The method has to build an appropriate background model and analyze what can (or cannot) be included in the background model during update. Light variations, shadows, camera jitter and dynamic background are also big challenges.

The crowd density is determined by counting the number of pixels in foreground mask obtained by Multi Layer BGS. This procedure is performed for each video frame. At first, a region of interest (ROI) is set manually to prevent or at least minimize the presence of external unrelated objects to be considered in motion segmentation and density estimation (see Figure 2). In [4] and [5] the authors have used 48x48 path windows centered in the region with higher motion. However, these windows are very small to perform density estimation. In this work, the windows were resized to 190x140 for better results. Figure 4 shows the crowd density variation over three videos of UCSD data set with distinct traffic patterns (light, medium and heavy congestion). The traffic crowd density is estimated by the average of density variation in each video. In case of video streaming or a lengthy video, the density estimation can be determined by blocks of N frames. For each block, the system predicts the crowd density.

KLT is a combination of feature extraction method of [26] and optical flow method originally proposed by [17] and later improved by [29] with pyramidal approach. The advantage of pyramidal implementation is the robustness against big movements and the motion detection with different speeds [24]. Many authors [25, 8] use KLT to compute flow vectors of motion field. The main motivation for choosing the KLT tracker is it fast performance that allows us to get real-time results [25, 24]. Rather than detecting and tracking a specific object, only feature points are tracked. As reported by Shi and Tomasi [26], only the best feature points (that have a strong texture) are selected for tracking.

Basically, us crowd tracking consists of two steps. Firstly, given two consecutive frames (see Figure 5(a)(b)), a certain amount of feature points is extracted from first frame (Figure 5(c)) and stored in memory. In the next step, the stored feature points are tracked on consecutive frame (Figure 5(d)). Some feature points can be lost in tracking (object has entered/left the scene). In this case, if the number of feature points is lower than threshold T_{fp} , the algorithm extract new feature points in the next frame. Here, $T_{fp} = 50$. The displacement of a particular feature point over some frames represent the crowd motion vector. However, many motion vectors may be zero length because some feature points are stationary along frames, or may be too short in the presence of noise, dynamic background or tiny movement of crowd. To reduce this problem on crowd speed estimation, the motion vectors with length smaller than threshold T_{mv} are filtered out (Figure 5(e)). In this work, the threshold is set to $T_{mv} = 3$.

2. Crowd Tracking

In crowded scenes, traditional object-based tracking fails because of the larger degree of occlusion. The crowd tracking focus on the motion of small elements whose structure changes continuously over time. Basically, the crowdtracking is the process by which the speed, direction andlocation of crowd in video sequence can be estimated. To perform the crowd tracking, the traditional KLT (Kanade-Lucas-Tomasi) tracker method was chosen. The To estimate the speed, the average displacement of feature points along all frames is calculated. In the case of video streaming or lengthy videos, the speed estimation can be determined by blocks of N frames. For each block, the system predicts the crowd speed. Figure 6 shows the crowd speed variation over three videos of UCSD data set with distinct traffic patterns (light, medium and heavy congestion).

II. PROPOSED ALGORITHM

The proposed system contains the following modules as shown in “Fig 1”.

Fig 1: Proposed system design see below

Proposed system consists of training phase & testing phase. Our proposed system consists of modules named image acquisition, input image, image enhancement, edge extraction & feature extraction.

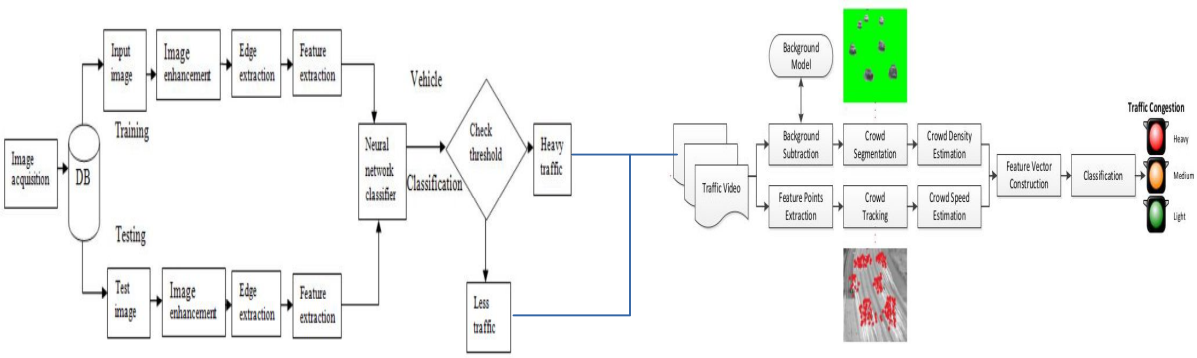


Fig. 1: Proposed system. From left to right: input video based on neural network classified into heavy traffic or less traffic, background subtraction process (top) and feature points extraction (down), crowd segmentation (top) and crowd tracking (down), estimation of crowd density (top) and crowd speed (down), construction of feature vector with crowd density average and crowd speed average, and traffic congestion classification in three classes: light (green), medium (yellow) and heavy (red).

3.1 Image acquisition

Image acquisition in image processing can be broadly defined as the action of retrieving an image from some source, usually a hardware-based source, so it can be passed through whatever processes need to occur afterward. Performing image acquisition in image processing is always the first step in the workflow sequence because, without an image, no processing is possible. One of the ultimate goals of image acquisition in image processing is to have a source of input that operates within such controlled and measured guidelines that the same image can, if necessary, be nearly perfectly reproduced under the same conditions so anomalous factors are easier to locate and eliminate. Image should be collect by reducing the speed, as fast moving vehicles cannot be captured using cameras. In our work image is captured using sony cyber shot digital camera of 8MP, the high mega pixels of camera has good clarity of image. In this work, our database consists of three different class of images like images of only two wheelers, images of only four wheelers & images combining both two & four wheelers.

3.2 Input image

Here in this module we take the image from the database and we pass it to the next module for further processing. Here we can passing an image of any class from our database collection.

3.3 Image enhancement

Image Enhancement involves the modification of digital data for improving the image qualities with the aid of computer. The processing helps in maximizing clarity, sharpness and details of features of interest towards information extraction and further analysis. The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application. Due to wind, road pollution & heavy shutter of camera, the captured image may not be clear & may not be suitable for further processing, hence the captured image must be enhanced before passing it to the next module. Most interpreters are concerned with recognizing linear features in images such as joints and lineaments. Geographers map manmade linear features such as highways and canals. Some linear features occur as narrow lines against a background of contrasting brightness; others are the linear contact between adjacent areas of different brightness. In all cases, linear features are formed by edges. Some edges are marked by pronounced differences that may be difficult to recognize. Contrast enhancement may emphasize brightness differences associated with some linear features. This procedure, however, is not specific for linear features because all elements of the scene are enhanced equally, not just the linear elements. Digital filters have been developed specifically to enhance edges in images and fall into two categories: directional and non-directional.

3.4 Edge extraction

Edges are those places in an image that correspond to object boundaries. Edges are pixels where image brightness changes abruptly. An edge is a property attached to an individual pixel and is calculated from the image function behavior in a neighborhood of the pixel. It is a vector variable (magnitude of the gradient, direction of an edge).

Edge Detection

Edge information in an image is found by looking at the relationship a pixel has with its neighborhoods. If a pixel's gray-level value is similar to those around it, there is probably not an edge at that point. If a pixel's has neighbors with widely varying gray levels, it may present an edge point.

Edge Detection Methods

Many are implemented with convolution mask and based on discrete approximations to differential operators. Differential operations measure the rate of change in the image brightness function. Some operators return orientation information. Other only return information about the existence of an edge at each point. The canny edge extraction method is used in the work.

The Canny Edge Detection Algorithm

The algorithm runs in 5 separate steps

1. Smoothing: Blurring of the image to remove noise.

2. Finding gradients: The edges should be marked where the gradients of the image has large magnitudes.

3. Non-maximum suppression: Only local maxima should be marked as edges.

4. Double thresholding: Potential edges are determined by thresholding.

5. Edge tracking by hysteresis: Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

3.5 Training in neural network

Fig 4 : A 2-layer, Feedforward Network with 4 inputs and 2 outputs

A neural network is defined not only by its architecture and flow, or interconnections, but also by computations used to transmit information from one node or input to another node. These computations are determined by network weights. The process of fitting a network to existing data to determine these weights is referred to as training the network, and the data used in this process are referred to as patterns. Individual network inputs are referred to as attributes and outputs are referred to as classes. In the testing phase of the design, we first take a sample input image then we follow all steps from image enhancement to feature extraction. The extracted features are then compared with the trained features. Based on the result the vehicle in the given input image is classified as two wheeler or a four wheeler.

Algorithm:

1) Read the image from the users.

2) Apply 2D DWT using discrete wavelet over the image.

3) Obtain horizontal, vertical, diagonal & approximate coefficients.

4) Display the resulting image of all four sub-bands.

5) Compute entropy, covariance, standard deviation feature values.

6) Store all features in a library & train the neural network.

7) The neural network parameters like learning rate of 0.04, epochs = 7000, 20 input features, three output class are adopted.

3.6 Vehicle count

The number of vehicles in the input image is counted with the help of region counting method. We set a threshold value for assessing traffic, based on this threshold value the traffic is assessed by comparing the count of vehicle in the image & threshold value. If count value is greater than threshold the output will be heavy traffic (congested) or else output will be no traffic.

IV.CONCLUSION

4.1 .Experimental Evaluation

The data set includes a diversity of traffic patterns likes light, medium and heavy congestion with variety of weather conditions (e.g. clear, raining and overcast). Each video has 42-52 frames with 320x240 resolution recorded at 10 frames per second (fps). The data set also provides a hand-labeled ground truth that describes each video sequence. Table 2 shows a summary of UCSD dataset.

Table 2: UCSD highway traffic data set summary.

Traffic congestion	Description	Number of videos
Light	Low number of vehicles. Vehicles at high speed or maximum speed allowed.	165
Medium	Average number of vehi- cles. Vehicles at reduced speed.	45
Heavy	Large amount of vehicles. Vehicles at low speeds or stop and go. Number of videos	44

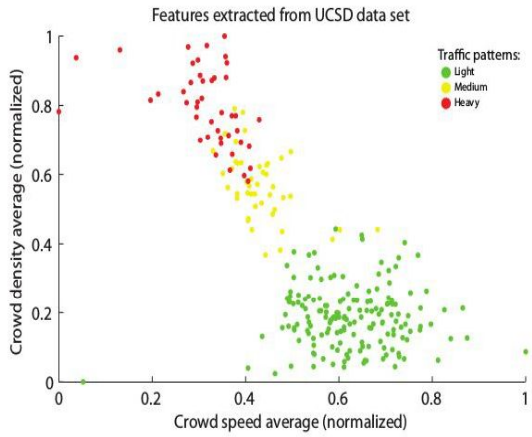
To perform the traffic congestion classification, in the firstplace all extracted features have been normalized between[0,...,1] by:

$$x = (x - \min) / (\max - \min)$$

where x is the original feature value, x is the normalized value, min and max the minimum and maximum value from feature set. Figure 8 shows the normalized features extracted from UCSD data set.

Fig. 2: Normalized features extracted from UCSD traffic videos.

The same training and testing methodology of [4] and [5] is adopted here. The experiment evaluation consists of four trials (T1,...,T4), where in each trial the data set was split with 75% for training and cross-validation and 25% for testing.



Multi-Layer Perceptrons

To perform the classification with neural networks, the feed-forward MLP network was chosen. With hidden layers, the MLP networks are able to solve non-linearly sepa rable data.

The MLP network was configured as follows: a) the input layer has two neurons (one for crowd density and and the other to crowd speed); b) the hidden layer evaluation

Table 6: MLP evaluation in four test trials varying the train ing algorithm (TA), activation function (AF) and number of hidden neurons (HN).

TA	AF	HN	T1	T2	T3	T4	Avg.
RPROP	SIG	2	95.20	95.30	93.80	87.30	92.900
		3	95.20	90.60	93.80	87.30	91.725
		4	95.20	95.30	93.80	93.70	**94.500**
		5	93.70	95.30	92.20	88.90	92.525
	GUI	2	92.10	87.50	92.20	88.90	90.175
		3	96.80	93.80	93.80	88.90	93.325
		4	82.50	95.30	81.30	90.50	87.400
		5	93.70	92.20	93.80	85.70	91.350
B	SI	2	96.80	95.30	93.80	85.70	92.900
		3	96.80	89.10	93.80	87.30	91.750
		4	90.50	93.80	92.20	87.30	90.950

P	G	5	95.20	90.60	92.20	90.50	92.125
R		2	82.50	81.30	81.30	82.50	81.900
O	G	3	92.10	84.40	95.30	85.70	89.375
P	U	4	93.70	89.10	90.60	87.30	90.175
	I	5	88.90	90.60	90.60	93.70	90.950

are made with 2-5 neurons; c) the output layer contains three neurons, one for each traffic patterns (light, medium and heavy). All neurons use the same activation functions. Sigmoid function (SIG) $f(x) = \beta(1 - -\alpha x)/(1 + -\alpha x)$ and gaussian function (GAU) $f(x) = \beta - \alpha x \square x$ are selected with standard parameters $\alpha = 1$ and $\beta = 1$. Two training algorithms are chosen, the tradicional gradient descent backpropagation (BPROP) [15] and resilient backpropaga tion (RPROP) [23].

In Table 6, the MLP network accuracy for each configuration is listed. The best score was achieved with four hidden neurons, sigmoid activation function and RPROP training algorithm.

The experimental results have shown that the MLP network has the best accuracy (94.5%). Table 7 describes the confusion matrix of MLP network. The most critical situation of misclassification occurs between medium and heavy trafficpatterns. This can be seen in Figure 8: several points that represent these two traffic patterns are fairly close. The robustness of background subtraction, feature extraction and tracking algorithms are main factors to get the best results, but given the nature of the traffic conditions the boundary values are varies some times.

REFERENCES

[1] A. Albiol, L. Sanchis, A. Albiol, and J. Mossi. Detection of parked vehicles using spatiotemporal maps. IEEE Transactions on Intelligent Transportation Systems (ITS), 12(4):1277–1291, dec. 2011.

[2] N. Buch, J. Orwell, and S. Velastin. Detection andclassification of vehicles for urban traffic scenes. In 5th International Conference on Visual InformationEngineering (VIE), pages 182–187, aug 2008.

[3] N. Buch, S. Velastin, and J. Orwell. A review of computer vision techniques for the analysis of urban traffic. IEEE Transactions on Intelligent TransportationSystems (ITS), 12(3):920–939, sept. 2011.

[4] A. Chan and N. Vasconcelos. Classification and retrieval of traffic video using auto-regressive stochastic processes. In IEEE Intelligent Vehicles Symposium(IVS), pages 771–776, june 2005.

[5] K. G. Derpanis and R. P. Wildes. Classification of traffic video based on a spatiotemporal orientation analysis. In Proceedings of the 2011 IEEE Workshop on Applications of Computer Vision, WACV'11, pages 606–613, 2011.

[6] F. El Baf, T. Bouwmans, and B. Vachon. Fuzzy integral for moving object detection. In IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pages 1729–1736, june 2008.

[7] S. Gupte, O. Masoud, R. Martin, and N. Papanikolopoulos. Detection and classification of vehicles. IEEE Transactions on Intelligent TransportationSystems (ITS), 3(1):37–47, mar 2002.

[8] W. He and Z. Liu. Motion pattern analysis in crowdedscenes by using density based clustering. In 9th International Conference on Fuzzy Systems and Knowl-edge Discovery, pages 1855–1858, may 2012.

[9] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Bacground segmentation with feedback: The pixel-based adaptive segmenter. june 2012.

[10] W.-L. Hsu, K.-F. Lin, and C.-L. Tsai. Crowd density estimation based on frequency analysis. In Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), pages 348–351, oct. 2011.

[11] J. Jacques Junior, S. Musse, and C. Jung. Crowd analysis using computer vision techniques. IEEE SignalProcessing Magazine, 27(5):66–77, sept. 2010.

[12] P.-M. Jodoin, V. Saligrama, and J. Konrad. Behaviorsubtraction. IEEE Transactions on Image Processing,21(9):4244–4255, sept. 2012.

[13] S. JunFang, B. ANing, and X. Ru. A reliable counting vehicles method in traffic flow monitoring. In 4th In-ternational Congress on Image and Signal Processing(CISP), volume 1, pages 522–524, oct. 2011.

[14] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis. A survey of video processing techniques for traffic applications. Image and Vision Computing, 21(4):359–381,2003.

[15] Y. Lecun, L. Bottou, G. B. Orr, and K. R. M¨uller. Efficient backprop. In G. Orr and K. M¨uller, editors,Neural Networks—Tricks of the Trade, volume 1524 of Lecture Notes in Computer Science, pages 5–50. Springer Verlag, 1998.

[16] J. Lee and A. Bovik. Estimation and analysis of urbantraffic flow. In 16th IEEE International Conference onImage Processing, pages 1157 – 1160, nov 2009.

[17] B. D. Lucas and T. Kanade. An iterative image regis-tration technique with an application to stereo vision. In Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2, pages 674–679, 1981.

[18] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. IEEE Transactions on Image Processing, 17(7):1168–1177, july 2008.

[19] L. Maddalena and A. Petrosino. A fuzzy spatial coherence-based approach to background/foregroundseparation for moving object detection. Neural Comput. Appl., 19(2):179–186, March 2010.

[20] N. C. Mithun, N. U. Rashid, and S. M. M. Rahman. Detection and classification of vehicles from video using multiple time-spatial images. IEEE Transactions on Intelligent Transportation Systems (ITS), PP(99):1–11, 2012.

[21] F. Porikli and X. Li. Traffic congestion estimation using hmm models without vehicle tracking. In IEEE Intelligent Vehicles Symposium (IVS), pages 188–193,june 2004.

[22] C. Pornpanomchai and K. Kongkittisan. Vehicle speed detection system. In IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pages 135–139, nov. 2009.

[23] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In IEEE International Conference on Neural Networks, pages 586–591, 1993.

[24] F. Santoro, S. Pedro, Z.-H. Tan, and T. B. Moeslund. Crowd analysis by using optical flow and density based clustering. Proceedings of the European Signal Processing Conference, 18:269–273, 2010.

[25] S. Saxena, F. Br´emond, M. Thonnat, and R. Ma.Crowd behavior recognition for video surveillance. In Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems, ACIVS ’08, pages 970–981, Berlin, Heidelberg, 2008. Springer-Verlag.

[26] J. Shi and C. Tomasi. Good features to track. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pages 593–600, jun 1994.

[27] M. Valera and S. Velastin. Intelligent distributed surveillance systems: a review. IEEE Proceedings on Vision, Image and Signal Processing, 152(2):192–204, april 2005.

[28] J. Yao and J. Odobez. Multi-layer background subtraction based on color and texture. In IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pages 1–8, june 2007.

[29] J. yves Bouguet. Pyramidal implementation of the lucas kanade feature tracker. Intel Corporation, Microprocessor Research Labs, 2000.

[30] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.-Q. Xu. Crowd analysis: a survey. Machine Vision Applications, 19(5-6):345–357, 2008.