Crowd Density Analysis Using Co-occurrence Texture Features

Wenhua Ma
Institute of Automation
Chinese Academy of Sciences
Beijing, China
wenhua.ma@ia.ac.cn

Lei Huang
Institute of Automation
Chinese Academy of Sciences
Beijing, China
lei.huang@mail.ia.ac.cn

Changping Liu
Institute of Automation
Chinese Academy of Sciences
Beijing, China
changping.liu@mail.ia.ac.cn

***Abstract*— Crowd density analysis is crucial for crowd monitoring and management. This paper proposes a novel method for crowd density analysis. According to the framework, input images are firstly divided into patches, and each patch is associated with a density label based on its texture features. Finally, local information is synthesized for global density estimation. Local image content is described by features based on co-occurrence textures and visual words processing chain. Experiments show that the system is highly robust to scene changes and background noise yet remain discriminative for crowd detection.**

I. INTRODUCTION

Crowd density analysis is crucial for crowd monitoring and control. Its target areas include pavements, squares, shopping malls, train stations and bus stops, where are usually crowded with people. The statistic data of crowd density can be used to provide assistance for early detection of unusual events, investigation data of volume of commuters, or guidelines for the design of public spaces. However, crowd density estimation is among one of the most difficult problems in surveillance. Crowds can be greatly various in their distribution and color patterns. And more importantly, a crowd does not have a well-defined shape as a single object does. All of these pose great challenges to a computer vision system for crowd density estimation.

A. Introduction to crowd density

Although crowd density is defined as the number of pedestrians per unit area, people counting is not always necessary for density analysis. Polus et.al[1] provide a clear idea of the problem of level of services for a pedestrian flow, and according to that, crowd density is quantized to 5 levels: very low, low, medium, high and very high. The qualitative information is valuable, for crowd of different density levels should receive different extent of attentions in surveillance. Besides, it is infeasible to distinguish individuals in some extreme cases. Human-beings are capable to determine density level of an area at a glance, without numerating the total number of pedestrians in it. And this inspires us to search for a way that can directly mapping from image features to crowd density.

B. Related works

Actually, crowd density analysis can be divided into two categories based on result: one is people counting through pedestrian detection; the other is density level estimation based on feature regression.

Most of the leading approaches of the former borrow ideas from pedestrian detection. And one of the most commonly used features is motion. Motion based methods usually resort to background models[2] or reference images[3] for motion detection, and classify objects by analyzing their shapes and/or trajectories. However, people that are stagnant for a long period of time would very likely be ignored by motion based method, which makes the estimation unreliable in the space where people often pause, such as squares and parks. In contrast to motion based methods, appearance based methods detect people using appearance cues, such as face[4] and contour[5]. However, it is difficult to segment individuals in a crowd, especially when the density is high and occlusion is serious. Some research addresses the problem by using part-body models of humans[6], [7], and report excellent performance in relatively small monitoring areas and sparse-crowded scenes. However, sophisticated methods are still needed to cope with extreme clutter and large-field monitoring.

Feature regression based methods takes crowd as a whole and try to establish a relationship between image feature and crowd density. It is argued that crowd is like a kind of texture, which exhibits some”human-like” properties in local structure and repeats spatially[8]. Moreover, images of low-density crowds tend to present coarse texture, while images of dense crowds tend to present fine textures[9]. Methods of this kind exploit these properties by extracting sparse or dense texture features from the image and associate them with crowd size. Commonly used texture features in density estimation include grey level co-occurrence matrix (GLCM) [9], [10], fractal dimension[11], LBP[12], histogram of gradients(HOG)[13] etc. Recently, the bag-of-words approach has achieved great success in computer vision[14], [15], which provides robustness to local descriptor variations as similar patch descriptors are mapped to the nearest key-point. However, it has not been applied to density estimation so far, as to our knowledge.

C. Our approach

The achievements listed above inspire the motivation of this work. Feature regression based crowd density analysis studies the crowd from macroscopic view and works robustly in most situations. However, it suffers from lack of explicit

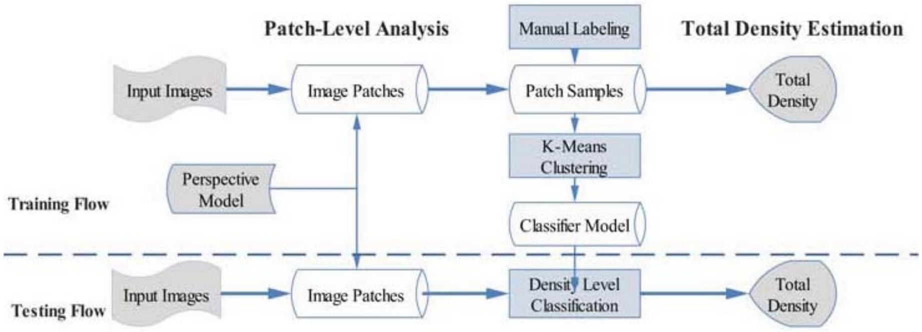

Fig. 1. Flow chart of the patch-based crowd density analysis,which contains two parallel flows: the training flow and the testing flow, and each can be further divided into two parts:the patch-level analysis, and the total density estimation.

measurements of density levels and powerful features for crowd detection. In this work, we propose a patch-based density analysis framework, and specifies explicit measurements of density levels based on image pathes. For each patch, a novel texture descriptor based on co-occurrence of gradient patterns is extracted and used for classification. Experiments in several practical monitoring areas show the merits of our work.

The rest of this paper is organized as follows: the overall architecture of the framework is introduced in Section 2, and the novel texture feature for crowd representation is introduced in section 3. Experiments and discussions can be found in section 4. Section 5 concludes the paper and gives future marks.

II. System Architecture

The entire system architecture is illustrated in Fig.1. There are two flows in this chart: the training flow and the testing flow. And each flow can be further divided into two parts: the patch-level analysis, and the total density estimation. The input sequences or images are firstly divided into patches, and then texture features are extracted from these patches. Each patch is assigned with a density level, and finally all the local information is synthesized for total density estimation. Given the extremely large within-category variations in crowds as well as the huge content contained in background, this patch-based architecture is feasible for breaking the problem down.

A. Patch-Level processing

The role of patch-level processing is to generate image patches within the region of interest(ROI) and assign local image patches to different density levels.

First of all, a uniform and explicit standard for crowdedness definition under various surroundings as well as different locations is necessary. This is achieved by ensuring that different image patches have the same ”capacity”(the maximum number of people a patch can cover). Due to perspective distortion, objects farther away from the camera appear smaller[10]. Therefore, sizes of the patches are selected carefully to compensate the distortion. This can be achieved by camera calibration[10]. For simplicity, approximation method is used here. Assuming that pedestrians have similar size in nature, under a certain scene, we first pick a ROI and specify size of the nearest and farthest image patches (assign the size as the height of a reference pedestrian). After that, we can approximate the perspective model by linearly interpolating between the two extremes of the scene and generating other patches. Examples of the generated samples in a square scene is shown in Fig.2.

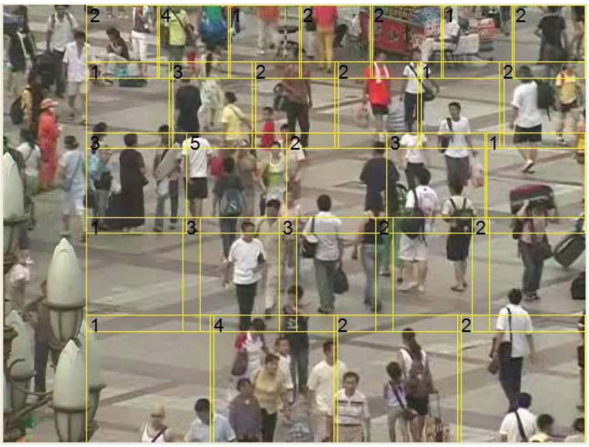

Fig. 2. Patch Samples Generated and Labeled in a square scene

As regards density measurement, we adopt Polus’s definition[1] of crowd density in substance, while made some modifications to apply it to patches. Specifically, five density levels are defined on patch level, ranging from 1 to 5, meaning the crowd density is very low, low, medium, high and very high respectively. Three measures are used to assess crowd density, with arrangements in descending priorities. Their threshold values are shown in Table I.

Total Number (**NP**): This is the total number of people in one patch.

Area Ratio (**AR**): This is proportion of areas occupied by people in the patch. High value of AR often indicates more people.

Number of Layers (**NL**): People tend to be arranged in layers in the image. More layers indicate more people.

During the training phase, all the training samples(image

TABLE I
DENSITY MEASURES FOR DENSITY LEVEL DEFINITION. FOR DETAILS SEE SECTION II-A

Class	1	2	3	4	5
NP	0∼2	2∼4	4∼6	6∼8	⩾8
AR	0∼0.3	0.2∼0.5	0.4∼0.7	0.6∼0.8	⩾0.8
NL	0∼1	1∼2	2∼3	3∼4	⩾4

patches of the training images) are involved. Texture features are extracted from the samples and compared using specified similarity measures. In our work, L_1 norm is adopted for its low computational complexity and high accuracy. For samples of the same density class, nonparametric clustering is used to simulate prototype distributions. As result, K clusters (K may be different for specific ranges) are generated for each density class and act as represents of the class. For each cluster, we use cluster center to denote its location and radius vector to denote its divergence:

$$m_i = \frac{\sum_{k=1}^{n} x_k}{n} \tag{1}$$

$$R_i(b) = \frac{\sum_{k=1}^{n} |x_k(b) - m_i(b)|}{n} \tag{2}$$

where m_i is the center of the ith cluster, n is number of members in that cluster and $R_i(b)$ is the bth dimension of the radius vector of the ith cluster. During testing phase, image patch is simply assigned to the same class with that of its nearest cluster.

B. Total density estimation

After patch-wise classification, we estimate the total crowd density as Eq.(3), where TD is short for total density, N is the number of patches in the image or region of interest in it, and $d(x_i)$ is the density level of sample x_i.

$$TD = \sum_{i=1}^{N} d(x_i)/N \tag{3}$$

III. CROWD REPRESENTATION

In order to explore the typical local structure of crowd, co-occurrence matrix is used. One of the commonly used co-occurrence matrix is GLCM, which is a tabulation of how often different combinations of pixel grey levels occur in an image. We refer to Mryka H.B.[16] for detailed introduction of GLCM. Even the same person can be various greatly in colors and luminance in different situations, therefore, it is more advisable to use contours or shapes for crowd estimation. This can be illustrated by Fig.3: original images are shown in column a) and corresponding gradient maps are shown in column b). In despite of the huge differences in colors and textures of original crowd samples (Row 3 and Row 4), their gradient maps are similar, exhibiting some typical silhouette of human. Based on the above analysis, we choose the gradient maps for crowd representation and propose the gradient orientation co-occurrence matrix(GOCM).

A. The gradient orientation co-occurrence matrix

Given an intensity image $\mathbf{Q}$, in which the intensity value of pixel p is defined as $I(p)$. Gradient at pixel p is a two-dimensional vector $G(p)$, composed of $G_m(p)$ (gradient magnitude) and $G_\theta(p)$ (gradient orientation). In our work, the gradient magnitude and orientation are calculated by Eq.(4)and(5). Then the orientation values are quantized into nine discrete values, The range $[0^\circ, 180^\circ]$ is divided into nine bins evenly, which correspond to the integers from 0 to 8 respectively. An angle $G_\theta(p)$ within range $[180^\circ, 360^\circ]$ has the same quantized value as its symmetry $360^\circ - G_\theta(p)$.

$$G_m(p) = \sqrt{(\nabla I_x(p))^2 + (\nabla I_y(p))^2} \tag{4}$$

$$G_\theta(p) = \arctan \frac{\nabla I_y(p)}{\nabla I_x(p)}, \quad G_\theta(p) \in [0, \pi] \tag{5}$$

In our work, GOCM is a 9×9 matrix extracted from the gradient map $\mathbf{G}$. GOCM considers two pixels at a time, calls the reference pixel and the neighbor pixel. Each pixel in the image becomes the reference pixel in turn, starting in the upper left corner and proceeding to the lower right.

```
Algorithm1 : Calculation of GOCM
Input    Gradient map $\mathbf{G} = \{G(p), p \in \mathbf{Q}\}$
         Number of candidate neighbors $n$
         Distance $d$ between the reference pixel
         and the neighbor pixel
Output   A $9 \times 9$ matrix, denoted as $\mathbf{P} = \{P_{i,j},$
         $i, j \in \{0, 1, \ldots, 8\}\}$

1.   Specify a zero-matrix with the same size as $\mathbf{P}$,
     denoted as $\widehat{\mathbf{P}} = ZERO(9,9)$.
2.   Specify a reference pixel $p_r = (x_r, y_r)$ in $\mathbf{G}$.
3.   Search among the $n$ neighbor pixels around $p_r$,
     denoted as
     $E_r = \{p_i, |p_i - p_r| = d \wedge i = 0, \ldots, n-1\}$,
     for its neighbor pixel $p_b$ so that:
     $\forall p_i \in E_r, |G_m(p_r) - G_m(p_b)| \le |G_m(p_r) - G_m(p_i)|$.
4.   Denote $G_\theta(p_r) = o_r, G_\theta(p_b) = o_b$,
     $\widehat{P}(o_r, o_b) = \widehat{P}(o_r, o_b) + G_m(p_r)$.
5.   If this is the last reference pixel, go to 6,
     else go to 2(turn to the next reference).
6.   Normalize the matrix and output values:
     $P_{i,j} = \frac{\widehat{P}_{i,j}}{\sum_{i,j} \widehat{P}_{i,j}}$
```

B. GOCM features for crowd estimation

In Fig.3, column b), some visible, continuous, curvy and vertical edges can be found in the crowd samples (Row 3 and Row 4). In contrast, either some very noisy, blurry edges or straight and regularly connected ones exist in the background samples (Row 1 and Row 2). Aiming at describing these differences explicitly, following features are deduced from GOCM. In all of the following formulas, the GOCM is denoted as $P_{i,j}$, while the matrix before normalization is denoted as $\widehat{P_{i,j}}$, and $i, j = 0, 1, \ldots, 8$.

a) Summation(SUM): Sum of all the entries in GOCM *before normalization*. The value is related to local accumulation of gradient magnitude.

$$SUM = \sum_{i,j} \widehat{P}_{i,j} \tag{6}$$

b) Homogeneity(HOM): High value of HOM indicates low contrast of the gradient map[16].

$$HOM = \frac{P_{i,j}}{\sqrt{1+(i-j)^2}} \tag{7}$$

c) Max Element(MAX): High MAX value occurs if one combination of pixels dominates[16].

$$MAX = \max_{i,j} P_{i,j} \tag{8}$$

*d) Diagonal(**DIA**):* This is a nine-dimensional vector. Entries in it represent occurrence-probabilities of pixel pairs. Two elements in the pair are approach in spatial, gradient magnitude and gradient orientation. The probabilities are weighted by gradient magnitude, indicating some strong and continuous edges.

$$DIA_i = \frac{P_{i,i}}{\sum_{i=0}^{8} P_{i,i}} \tag{9}$$

*e) Conditional Mean(**COM**):* This is a nine-dimensional vector indicating the expected gradient orientation of the neighbor pixel given the gradient orientation of the reference pixel.

$$COM_i = \sum_j j \cdot P_{i,j} \tag{10}$$

*f) Marginal Distribution(**MD**):* This is a nine-dimensional vector which indicates the marginal distribution of the reference pixel. It is a first-order texture.

$$MD_i = \sum_j P_{i,j} \tag{11}$$

The GOCM feature vector in our work is formulated by contacting all the features above. As shown in Fig.3, column c), the GOCM feature is robust to variations of crowd(features of Row 3 and Row 4 are similar) yet remain discriminative(features are different between Row 1,2 and Row 3,4). Therefore, it is expected to bring a good performance to crowd detection and segmentation.

C. Bag of words model based description

The probabilistic Latent Semantic Analysis (pLSA) model[17] is a popular ”topic model” build on top of the bag-of-words representation that is efficiently fitted to data using a simple EM algorithm. Topic models consider the bag-of-words as a mixture of several ”topics”, i.e. the visual words obtained from a visual scene can be modeled as a mixture of words belonging to background, and several objects.

The visual words model is used in our work to discriminate between classes at patch-level. It is expected to improve the performance. Crowd samples of different levels mainly differ in spatial structure, while the local textons are relatively stable. Bag of words model is able to capture the local typical structure of crowd while at the same time reveal the composition upon these typical structures.

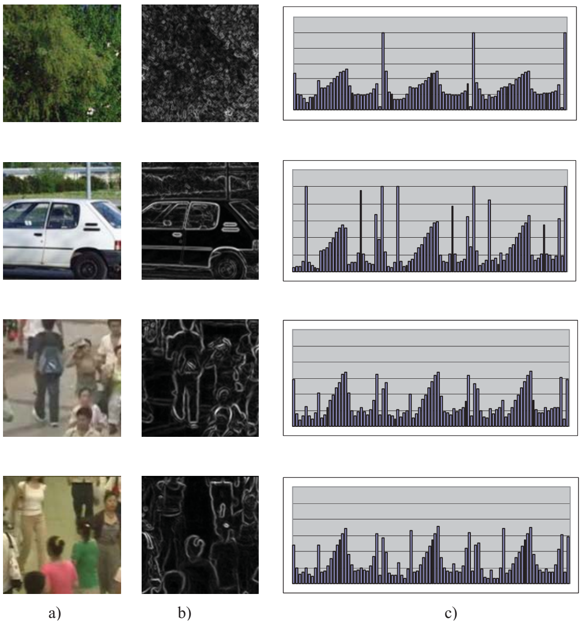

Fig. 3. Gradient information in samples. Row 1 and Row 2 are background samples, and Row 3 and Row 4 are crowd samples. Original image patches are shown in column a), gradient maps in column b), and GOCM feature vectors(shown as histograms) in column c).

Visual words are obtained by quantization of low level image descriptors. The quantization is performed by k-means algorithm, such as those in Ref.[14], [18]. During training phase, image fragments are extracted randomly from each patch sample firstly, and local features are computed on those fragments. The number of fragments in one patch is determined to be 100 through experiment. Visual vocabulary is constructed by K-means quantization done on the local features. Based on the vocabulary, image patches are represented by their cluster histograms, which denote how often each visual word occurs in the image patch.

To classify a new image patch, image fragments of the patch are matched to visual words in the vocabulary, resulting its cluster histogram. And the distance between two cluster histograms is measured by L_1 norm.

IV. Experiment

In this section, we present results on crowd detection and estimation using real video data. To our knowledge, no video database is available for benchmarking crowd estimation algorithms, so we collect the videos by ourselves.

The database contains four videos: subway, square, hall, platform. The ”subway” and ”square” sequences are for training, while the other two are for testing. All the sequences are taken in places where the ground is roughly horizontal. A commercial digital camera is used for monitoring the crowd, which looks down with a tilt angle about $30^\circ \sim 45^\circ$. Sample frames of the five videos are shown in Fig.4. It can be seen that the testing scenes are different from the training scenes

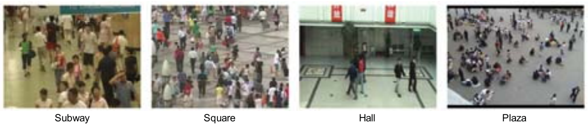

Fig. 4. surveillance videos in the dataset

TABLE II
DATABASE CONSTITUTION

Class	1	2	3	4	5	Total
Train	473	703	1110	1342	1712	5340
Test	425	302	752	180	341	2000
Total	898	1005	1862	1522	2053	7340

in many aspects such as pedestrian size and appearance, background constitution, depth of the field and so on. So that the robustness of the algorithm can be fully tested.

For training and testing the patch-level classifier, images are extracted from the videos at time interval around 2 seconds and divided into patches according to the method described in section II-A. Totally 7340 patches are generated in this way, of which 5340 are from training set, and 2000 are from testing set. All the image patches are labeled manually with one of the density levels. Table II shows the density distribution of the patch-level samples.

As regards multi-category classification, two metrics are defined based on the confusion matrix. Rank-1 classification accuracy ($CA1$) denotes the proportion occupied by correctly-classified samples. Because there is no sharp boundary between neighboring density levels, samples that are classified into adjacent classes of the ground truth are also acceptable. Therefore, Rank-2 classification accuracy ($CA2$) is defined to cover these samples.

$$CA1 = \frac{\sum correct}{\sum total} \tag{12}$$

$$CA2 = \frac{\sum correct + \sum adjacent\_correct}{\sum total} \tag{13}$$

A. Affection of feature parameters

There are two parameters in the proposed feature: number of neighbors N, and the distance between reference and neighbor d.

We have done five-class categorizing on the training set and ten-fold cross validation to optimize the parameters of the new texture feature. Fig.5 shows $CA1$ of different parameter combinations. For simplicity but without loss of generality, N is chose between 4 and 8, d is chose among 1, 2 and 3. Summing up the above, six combinations are tested: $1d8N$, $1d4N$, $2d8N$, $2d4N$, $3d8N$ and $3d4N$. Bars in the chart are arranged just as the same order. As we can see, the proposed feature is rather robust to different parameters, since the accuracy difference between different combinations is less than 2%. Overall, $2d$ outperforms $1d$ and $3d$, and $8N$ is almost better than $4N$, except in $3d$ case. Therefore, $2d8N$ gives the best performance among the six, about 91.95%, and it is used throughout our work.

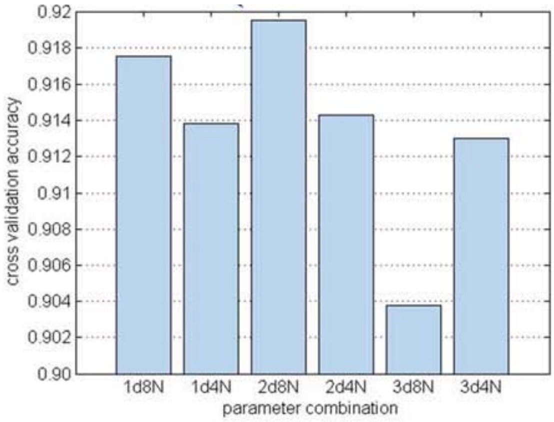

Fig. 5. Affection of feature parameters.

TABLE III
FEATURE COMPARISON ON PATCH-LEVEL DENSITY ESTIMATION

Feature	Dimension	Training		Testing	
		$CA1(\%)$	$CA2(\%)$	$CA1(\%)$	$CA2(\%)$
GLDM	48	77.13	90.03	70.26	87.54
HOG	2268	88.95	97.26	78.45	91.02
ULBP	531	83.23	96.01	76.03	90.78
ALBP	144	88.65	95.98	83.28	95.10
GOCM	90	90.03	98.27	87.62	96.01

B. Experiment in patch-level classification

Experiments in this stage contain two parts: firstly, different features are compared; secondly, the feature with best performance in the first experiment is selected for visual words based feature calculation.

1) Feature Comparison: For features are involved in this experiment: HOG[13], ULBP[19], ALBP[12], and the proposed feature GOCM. HOG is a dense texture descriptor. Here, it is calculated with blocks of 32 × 32-pixel. ULBP is a 1-D histogram of uniform local binary patterns. ALBP is proposed by Wenhua Ma et.al[12], which is a modification of ULBP.

The results are shown in Table III. It can be observed that, in all features used, hardly any confusion between distant classes, while neighboring classes, such as 1 and 2, are confused more frequently. Among the features compared, GLCM has the lowest dimension, however, the performance is not satisfying. HOG achieves good result on the training set, performance drops sharply on the testing set, implying an over-learning problem caused by the high feature dimension. ULBP and ALBP both achieves quite good result, and the difference between the two is relatively small. GOCM feature outperforms the others and yielding 94% $CA2$ and 79% $CA1$ on testing set. The results prove that GOCM feature is excellent in capturing mutual information among local gradient maps, and the information is very important in crowd detection and estimation. Based on the above analysis, GOCM feature is used for visual vocabulary construction.

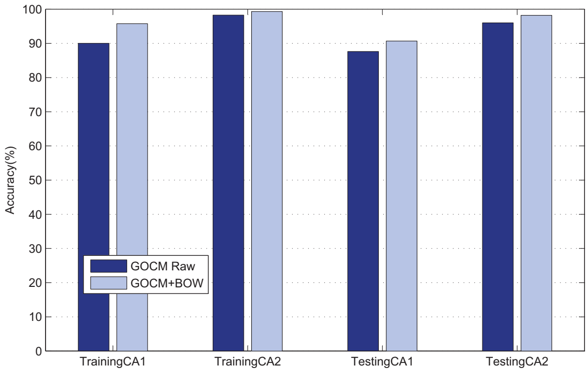

Fig. 6. Improvement made by visual words based feature

2) Improvement made by visual words: In this experiment, GOCM is selected as raw feature and compared with its visual-word-based feature. Visual vocabulary is constructed by K-means quantization during training phase. And visual words based feature is extracted by voting to the visual words in the vocabulary. It can be seen from Fig.6 that, by applying visual words based feature, both of the $CA1$ and $CA2$ on testing set increased about 4%. A significant improvement is also witnessed on the training set. This promising result indicates that visual words based feature is effective in describing repetitive textures with typical local structures, such as the crowd, thus be greatly helpful in categorization.

C. Total density analysis

Total density estimation accuracy is about 96.04% in the training scenes, and 92.35% in the testing scenes. The ”subway” sequence is taken within one arriving period of the train. During the arriving, crowds poured out of it and move outside. The crowd density varies sharply, from very low(Class 1) to very high(Class 5), and finally to low(Class 2) again. The density level of ”square” sequence hovers between medium (Class 3) to high (Class 4). The range of vision is large and individual object looks small and oblique. People in the ”hall” sequence talks and walks through the hall, and the density level is below medium(Class 3). And in the ”plaza” sequence, the total density level is high(Class 4). The estimation goes with the ground truth quite well in all the testing scenes, except that in the ”Hall”, the estimation is a little higher than ground truth, mainly due to disturbance made by the potted plants, which has quite similar boundaries as human.

V. Discussion and conclusions

In this paper, we have proposed a novel framework for crowd estimation under complex environments, in which the local texture feature is used to distinguish crowd of different density. The system is designed to input images of crowd scenes, and output the estimated crowd density map, as well as the total crowd density. This narrow, yet challenging scope was selected so as to provide focus to our efforts toward crowd surveillance as a whole. Experiment results on real image sequences demonstrate the value of our work. Further work includes making the system more accurate in performance by adding rough people counting function.

References

[1] Polus.A., Schofer.J., and Ushpiz.A. Pedestrian flow and level of service. *Journal of Transportation Engineering*, 109(1):46–56, 1983.

[2] D.Kong, D.Gray, and H.Tao. A viewpoint invariant approach for crowd counting. *IEEE International Conference on Pattern Recognition (ICPR)*, 3:1187–1190, 2006.

[3] Yin J., Velastin S., and Davies A. Image processing techniques for crowd density estimation using a reference image. *IEEE Conf. Asia-Pacific Confonerence on Computer Vision*, 3:6–10, 1995.

[4] S. Harasse, L. Bonnaud, and M. Desvignes. People counting in transport vehicles. *Proc. of World Academy of Science, Engineering and Technology*, pages 221–224, 2005.

[5] L.S.Davis I.Haritaoglu, D.Harwood. w^4: Real-time surveillance of people and their activities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 809–830, 2000.

[6] Lin S.F., Chen J.Y., and Chao H.X. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Trans. System, man and cybernetics*, 31(6):645–654, 2001.

[7] Leibe B., Seemann E., and Schiele B. Pedestrian detection in crowded scenes. *IEEE Conference on Computer Vision and Pattern Recoginition*, 1:878–885, January 2005.

[8] Arandjelovic O. Crowd detection from still images. *British Machine Vision Conference(BMVC)*, pages 1–4, Augst 2008.

[9] A. Marana, L. da Costa, R. Lotufo, and S. Velastin. On the efficacy of texture analysis for crowd monitoring. *SIBGRAPHI ’98: Proceedings of the International Symposium on Computer Graphics, Image Processing, and Vision*, pages 354–361, 1998.

[10] Rahmalan H., Nixon M.S., and Carter J.N. On crowd density estimation for surveillance. *IEEE Internation Conference on Crime Detection and Prevention*, pages 540–545, June 2006.

[11] A. Marana, L. da Costa, R. Lotufo, and S. Velastin. Estimating crowd density with minkowski fractal dimension. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 6:3521–3524, 1999.

[12] W.H.Ma, L.Huang, and C.P.Liu. Advanced local binary pattern descriptors for crowd estimation. *Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, pages 958–962, December 2008.

[13] N.Dalal. *Finding people in images and videos*. Phd dissertation, Inst.Nat’l Polytechnique de Grenoble, July 2006.

[14] C.Dance, J.Willamowski, L.Fan, C.Bray, and G.Csurka. Visual categorization with bags of keypoints. *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[15] P.Sabzmeydani and G.Mori. Detecting pedestrians by learning shaplet features. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[16] Mryka H.B. The glcm tutorial. *http://www.fp.ucalgary.ca/mhallbey/tutorial.htm/*.

[17] Josef S., Bryan C.R., Alexei A.E., Andrew Z., and William T.F. Discovering objects and their location in images. *Tenth IEEE International Conference on In IEEE International Conference on Computer Vision.*, 1:370–377, 2005.

[18] L.Fei-Fei and P.Perona. A bayesian hierarchical model for learning natural scene categories. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:524–531, 2005.

[19] T.Ojala, M.Pietikainen, and T.Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.