Real-Time Crowd Density Estimation Using Images

A.N. Marana[1], M.A. Cavenaghi[1], R.S. Ulson[1], and F.L. Drumond

UNESP (Sao Paulo State University) - FC (School of Sciences),
DCo (Department of Computing) - LCAD (Laboratory of High Performance Computing),
Av. Eng. Luis Edmundo Carrijo Coube, sn, 17033-360, Bauru, SP, Brazil
[1]{nilceu, marcos, roberta}@fc.unesp.br

Abstract. This paper presents a technique for real-time crowd density estimation based on textures of crowd images. In this technique, the current image from a sequence of input images is classified into a crowd density class. Then, the classification is corrected by a low-pass filter based on the crowd density classification of the last *n* images of the input sequence. The technique obtained 73.89% of correct classification in a real-time application on a sequence of 9892 crowd images. Distributed processing was used in order to obtain real-time performance.

1 Introduction

For the problem of real-time crowd monitoring there is an established practice of using closed circuit television systems (CCTV), which are monitored by human observers. This practice has some drawbacks, like the possibility of human observers lose concentration during this monotonous task. Therefore, the importance of the development of robust and efficient automatic systems for real-time crowd monitoring is evident.

Efforts for crowd estimation in train stations, airports, stadiums, subways and other places, have been addressed in the research field of automatic surveillance systems. Davies et al. [1] and Regazzoni and Tesei [2, 3] have proposed systems for crowd monitoring and estimation based on existing installed CCTV. The image processing techniques adopted by theirs systems remove the image background and then measure the area occupied by the foreground pixels. The number of foreground pixels is used to estimate the crowd density. Lin et al. [4] proposed a technique based on the recognition of head-like contour, using Haar wavelet transform, followed by an estimation of the crowd size, carried out by a support vector machine. For crowd classification, Cho et al. [5] proposed a hybrid global learning algorithm, which combines the least-square method with different global optimization methods, like genetic algorithms, simulated annealing and random search. The techniques proposed by Marana et al. [6,7,8] estimate crowd densities using texture analysis with gray level dependence matrices, Minkowski fractal dimension and wavelets.

This paper presents a technique for real-time automatic crowd density estimation based on texture descriptors of a sequence of crowd images. The motivation for the use of texture descriptors to estimate crowd densities was inspired by the fact that

G. Bebis et al. (Eds.): ISVC 2005, LNCS 3804, pp. 355 – 362, 2005.

images of different crowd densities tend to present different texture patterns. Images of high-density crowd areas are often made up of fine (high frequency) patterns, while images of low-density crowd areas are mostly made up of coarse (low frequency) patterns. In order to improve the estimation accuracy and to provide real-time estimation, a distributed algorithm was developed. The technique obtained 73.89% of correct classification in a real-time application on a sequence of 9892 crowd images.

2 Material

The technique described in this paper for automatic crowd density estimation was assessed on a sequence of 9892 images extracted (one per second) from a videotape recorded in an airport area. From the total set of images, a subset of 990 images was homogeneously obtained (one image from each ten). Then, human observers manually estimated the crowd densities of these images. The manual estimations were used to assess the accuracy of the automatic technique.

After the manual estimation, the 990 images were classified into one of the following classes: very low (VL) density (0-20 people), low (L) density (21-40 people), moderate (M) density (41-60 people), high (H) density (61-80 people), and very high (VH) density (more than 80 people). Figure 1 shows samples of crowd density classes.

(a) (b) (c) (d) (e)

Fig. 1. Samples of crowd density classes. (a) Very low density (15 people); (b) Low density (29 people); (c) Moderate density (51 people); (d) High density (63 people); (e) Very high density (89 people).

Finally, the images of each class were grouped into train and test subsets. The train subset was used to train the neural network classifier and the test subset was used to assess the accuracy of the technique. Table 1 shows the distribution of the train and test subsets of images into the five classes of crowd densities.

Table 1. Distribution of the train and test subset of images into the five classes of crowd densities

	VL	L	M	H	VH
Train	14	179	169	101	33
Test	14	179	169	100	32
Total	28	358	338	201	65

3 Methods

Figure 2 presents a diagram of the technique proposed for crowd density estimation using texture descriptors. The first step of the technique consists in the classification of each pixel of the input image into one of the previously identified texture classes. The classification is carried out by a self-organizing map (SOM) neural network [9] using feature vectors composed of texture descriptors extracted from co-occurrence matrices [10], computed using a *w*x*w* window centered in the pixel being classified.

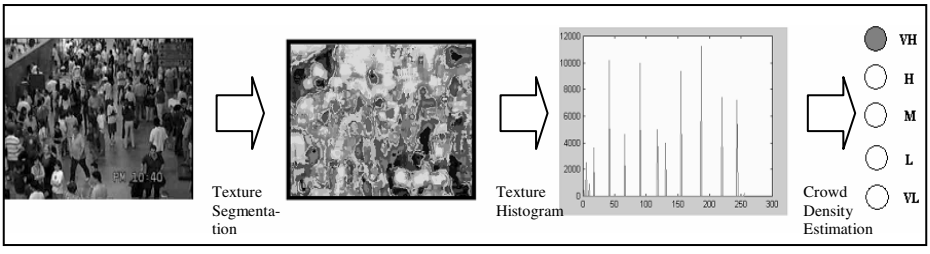

Fig. 2. Diagram of the technique for crowd density estimation using texture and neural network classifier

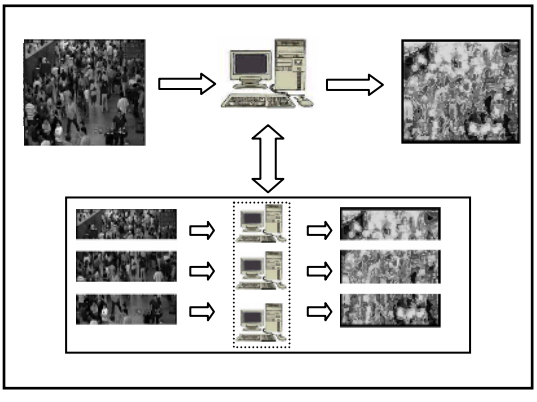

Fig. 3. Diagram of the proposed master-slave strategy for texture segmentation in PVM distributed environment, using *n* slave processors (in this example, *n=3*)

As the classification of all pixels of the image is a time-consuming process (more than 100 seconds per image), in order to obtain real-time estimation it was implemented a distributed algorithm for the Beowulf environment, using Parallel Virtual Machine (PVM) [11]. This algorithm has the following steps:

- The master processor divides the input image in n fragments (n is the number of slave nodes in the cluster);
- Each image fragment is sent to a slave processor;

- Each slave processor performs the texture classification of its image fragment pixels using a sequential algorithm;
- The slave processors send their classified fragments to the master;
- The master processor assembles all fragments into a final texture-segmented image.

Figure 3 shows a diagram of the master-slave strategy adopted in this work to obtain a texture-segmented image.

In the next step, the texture histogram, computed from the texture-segmented image, is used as feature vector by a second SOM neural network to classify the input crowd image into one of the crowd density classes.

The second neural network learns the relationship among the texture histogram profiles and the crowd density levels during the training stage, in a supervised way.

4 Experimental Results

This section presents the results obtained with the application of the proposed technique for real-time crowd density estimation on a sequence of 9892 crowd images.

During the experiments, it was used a cluster with eight Pentium IV processors, connected by a Fast-Ethernet switch.

Fig. 4. Texture patterns from where texture-training samples were extracted.

Pixel texture classification was carried out on a 15x15 window centered on the pixel, from where four co-occurrence matrices were calculated (distance d=1 and directions θ = 0º, 45º, 90º and 135º). From these four matrices, four texture features were extracted: energy, entropy, homogeneity and contrast [10], making up 16 features.

The SOM neural network used in the first step for texture classification was trained to classify crowd image pixels into 12 patterns of texture. Figure 4 shows the 12 texture patterns from where 100 training samples of each texture class were randomly extracted.

Figure 5(b) presents the result of the texture segmentation of the crowd image presented in Figure 5(a), obtained by the SOM neural classifier, using a 15x15 window and the texture patterns showed in Figure 4. Figure 5(c) presents the texture histogram obtained from the texture-segmented image.

It is possible to observe in Figure 5 that higher crowd density areas of the input image are associated with lighter gray level areas in the texture-segmented image, and that lower crowd density areas of the input image are associated with darker gray level areas in the texture-segmented image.

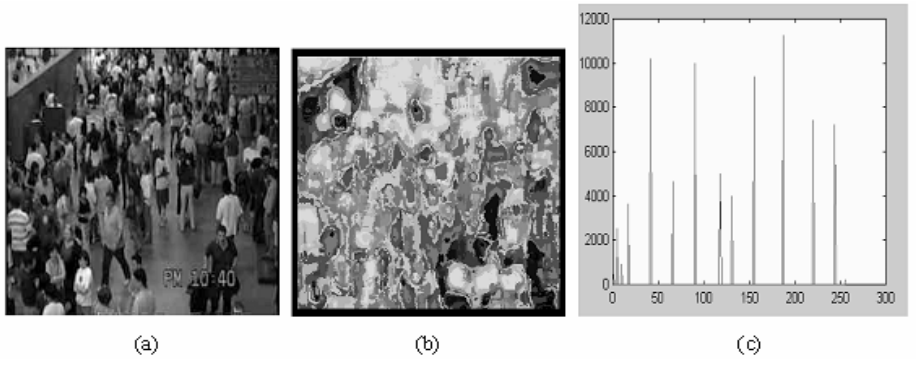

Fig. 5. Example of texture segmentation of a crowd image. (a) Input image; (b) Texture-segmented image; (c) Texture histogram obtained from the texture-segmented image.

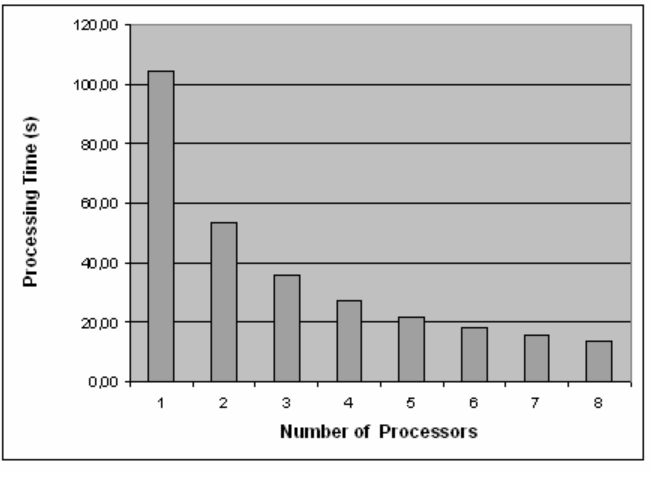

Fig. 6. Processing time (in seconds) necessary to classify all pixels of the input image (using a 15x15 window) and to estimate its crowd density, varying the number of processors of the cluster

Figure 6 shows the processing times for texture segmentation and crowd density classification of a single crowd image, using the 8-processors cluster. The efficiency obtained by the insertion of new processors in all cases was always almost maximum, since the decreasing of processing time was always near to 87%. The processing time for each crowd image was around 105 seconds when only one processor was used and around 14 seconds when all 8 processors were used.

Since the best processing time performance obtained (14 seconds) was not enough for real-time estimation, it was assessed the possibility of only part of pixels be classified.

In the first experiment carried out, the crowd images were divided into 4x4 sub-images and only one pixel from each sub-image (the top-left pixel) was classified. Table 2 shows the confusion matrix obtained in this experiment, where the last 7 images of the input sequence were used by a median low-pass filter to correct the current crowd density estimation. In this experiment, 73.89% of the 494 test crowd images were correctly classified. The best result (90.63% of correct classification) was obtained by the VH class, and the worst result (59% of correct classification) was obtained by the H class. Real-time requirement was reached, since the crowd density estimation for each image took 1.025 seconds. It is possible to observe in Table 2 that all miss-classified images were assigned to a neighbor class of the correct one. Some miss-classification was expected since the borders between the crowd density classes are very tenuous (for instance, an image with 20 people belongs to VL class, but it can be easily classified as belonging to L class).

Table 2. Results obtained when part of the input image pixels were classified and the crowd density classification were corrected applying the median low-pass filter in the last 7 estimations of the input sequence

	VL	**L**	**M**	**H**	**VH**
VL	64.29	35.71			
L	6.7	81.01	12.29		
M		10.65	72.78	16.57	
H			10.00	59.00	31.00
VH				9.38	90.63

Table 3. Results obtained by the technique applying a 3x3 mean filter on the texture-segmented image before calculating the texture histogram and correcting the estimation applying a low-pass (median) filter in the estimation of the last 10 images of the input sequence

	VL	**L**	**M**	**H**	**VH**
VL	71.43	28.57			
L	8.38	82.68	8.94		
M		13.02	77.51	9.47	
H			15.00	67.00	18.00
VH				18.75	81.25

In the second experiment, where all pixels of the input image were classified, it was obtained 77.33% of correct estimation. In this experiment, a 3x3 mean filter was applied to enhance (remove noise) the texture-segmented image and the last 10 estimations were used by a median low-lass filter to correct the crowd density classifications. But, in this case, the requirement for real-time estimation was not reached, since the estimations took 14 seconds. Table 3 shows the confusion matrix obtained in this experiment.

In the Table 3 it is also possible to observe that all miss-classified images were assigned to a neighbor class of the correct one (this is a very favorable result).

5 Conclusions

In this paper, the problem of crowd density estimation was addressed and a technique for real-time automatic crowd density estimation was proposed, based on texture features extracted from a sequence of images and processed in a distributed environment. The proposed approach takes into account the geometric distortions caused by the camera's position, since the farther areas (from the camera) under surveillance are mapped on finer textures and the closer areas are mapped on coarser textures. Crowd density estimations of a group of 494 test crowd images resulted in 77.33% of correct estimation. When real-time constraint was demanded, it was obtained 73.89% of correct estimation. These results can be considered quite good since the variance of crowd density estimations for each class were very small and their means were the expected values.

Acknowledgements

The authors thank FAPESP (process number: 01/09649-2) for the financial support.

References

1. Davies, A.C., Yin, J.H., and Velastin, S. A., "Crowd Monitoring Using Image Processing", *Electron. Commun. Eng. J*., vol. 7, pp.37-47, 1995.
2. Regazzoni, C.S., and Tesei, A., "Distributed Data Fusion for Real-Time Crowding Estimation", *Signal Proc*., vol. 53, pp. 47-63, 1996.
3. Tesei, A., and Regazzoni, C.S., "Local Density Evaluation and Tracking of Multiple Objects from Complex Image Sequences", *Proc. 20th Intern. Conf. IECON*, vol.2, Bologna, Italy, pp. 744-748, 1994.
4. Lin, S.F., Chen, J.Y., and Chao, H.X., "Estimation of Number of People in Crowd Scenes Using Perspective Transformation", *IEEE Trans. Sys., Man, Cyber. A*, vol.31, pp. 645-654, 2001.
5. Cho, S.Y., Chow, T.W.S., and Leung, C.T., "A Neural-Based Crowd Estimation by Hybrid Global Learning Algorithms", *IEEE Trans. Sys., Man, Cyber. B,* vol.29, pp. 535-541, 1999.
6. Marana, A.N., Velastin, S.A., Costa, L.F, and Lotufo, R.A., "Automatic Estimation of Crowd Density Using Texture", *Safety Science,* vol. 28, 165-175, 1998.

7. Marana, A.N., Costa, L.F., Lotufo, R.A., and Velastin, S.A., “Estimating Crowd Density with Minkowski Fractal Dimension”, *IEEE Proceedings of the International Conference on Acoustics, Speech and Signal Processing,* vol. VI**,** 3521-3524, 1999.
8. Marana, A. N., Verona, V.V., “Wavelet Packet Analysis for Crowd Density Estimation”, *Proc. IASTED Inter. Symposia on Applied Informatics,* Acta Press, pp. 535-540, Innsbruck, Austria, 2001.
9. Kohonen, T., "The Self-Organizing Map", *Proceedings of the IEEE*, vol.78, pp. 1464-1480, 1990.
10. Haralick, R. M., "Statistical and Structural Approaches to Texture", *Proceedings of the IEEE*, vol. 67(5), pp. 786-804, 1979.
11. 11.Geist, A.; Beguelin, A.; Dongarra, J.; Jiang, W.; Manchek, R.; Sunderan, V., “*PVM: Parallel Virtual Machine – A User’s Guide and Tutorial for Networked Parallel Computing”*, The MIT Press, 1994.