

Panoramic Background Modeling for PTZ Cameras with Competitive Learning Neural Networks

Karl Thurnhofer-Hemsi*, Ezequiel López-Rubio*, Enrique Domínguez*,
Rafael Marcos Luque-Baena* and Miguel A. Molina-Cabello*

**Department of Computer Languages and Computer Science
University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain
Emails: {karlkhader,ezeqlr,enriquetd,rmluque,miguelangel}@lcc.uma.es*

Abstract—The construction of a model of the background of a scene still remains as a challenging task in video surveillance systems, in particular for moving cameras. This work presents a novel approach for constructing a panoramic background model based on competitive learning neural networks and a subsequent piecewise linear interpolation by Delaunay triangulation. The approach can handle arbitrary camera directions and zooms for a Pan-Tilt-Zoom (PTZ) camera-based surveillance system. After testing the proposed approach on several indoor sequences, the results demonstrate that the proposed method is effective and suitable to use for real-time video surveillance applications.

1. Introduction

With their high mobility and zoom capability, pan-tilt-zoom (PTZ) cameras have become increasingly popular in monitoring public areas [1], [2], [3]. Omnidirectional cameras are promising candidates for monitoring multiple latent activities in the area of interest [4]. However, these kind of cameras have nonuniform resolution and are unable to provide close observations of particular targets. In these cases, where PTZ cameras are needed, the combination of these two types of cameras (omnidirectional and PTZ) is proposed in order to facilitate a continuous monitoring of the whole surveillance area and detailed observations of specific targets simultaneously [5]. Nevertheless, this dual-camera system may be still an expensive and complex solution in some scenarios. For this reason, in this paper we are focusing on an active sensing approach to multiple object detection and tracking using a single PTZ camera.

Most cameras used in surveillance are static, and the scenes taken from this type of cameras only show one specific view of the surveillance area. For these images/videos, the most common and efficient approach to moving object detection is background subtraction, that consists in maintaining an up-to-date model of the fixed background and detecting moving objects as those that deviate from such model. Compared to other approaches, such as optical flow, this approach is computationally affordable for real-time applications, is independent of moving object velocity, and is not subject to the foreground aperture problem.

However, traditional background subtraction algorithms assume the cameras are static and this leads to false detection when the camera moves [6], [7]. Due to this camera movement, even pixels belonging to static objects appear to move in the camera frame (called ego-motion effect).

Extensive research has been carried out regarding object detection for moving cameras. Some proposals are based on the optical flow clustering, that consists in calculating dense or sparse optical flows and clustering them to identify moving object regions [8]. Another methods are based on the estimation of the transformation parameters between consecutive frames [9]. Our approach is based on mosaicing the background [10], [11], that consist in creating a mosaiced or panoramic background image and then using a background subtraction technique to extract moving object regions.

The problem of moving objects detection for PTZ cameras is addressed in this paper, and we propose a method based on building a panoramic background model using a competitive neural network. Apart of the traditional and frequently cited seminal papers related to competitive learning [12], [13], [14], recent successful applications in the computer vision field can be found in the literature [15], [16], [17], [18]. In our approach, a competitive neural network is used to build a panoramic background model for object detection. Due to the huge input information, a large number of neurons has been used and the neuron prototypes have been organized in a quad-tree in order to be quickly evaluated.

The rest of the paper is organized as follows. In section 2, a more detailed description of the proposed neural model is presented. In section 3, we present the results achieved with the implementation of the proposed approach. Finally, section 4 includes some concluding remarks.

2. The model

In this section a competitive learning based system to learn the background of a panoramic scene from the input of a PTZ camera is proposed. First the data acquisition procedure to transform the input video frames into input samples for the competitive learning network is considered (Subsection 2.1). Then the competitive learning model is described (Subsection 2.2). Finally, an interpolation procedure

is designed to estimate the background from the final state of the competitive learning network, which is based on a Delaunay triangulation (Subsection 2.3).

2.1. Data acquisition

The data from which we start are the acquired video frames of the PTZ camera. These are video frames with fixed width and height, and every pixel has two frame coordinates (x, y) . To transform these frame coordinates to another coordinate system in the panoramic image, the first step is to obtain the spherical coordinates associated to the frame coordinates and then carry out a scalar transformation to the dimensions of the panoramic image.

In order to find out required polar coordinates (θ, ϕ) of an arbitrary point A belonging to projection plane (x, y) a pinhole camera model is used, consisting of a sphere centered on the coordinate origin and a projection plane (see Fig. 1a), whose bounds are the width and height of the frame window in pixels: w, h . In this work the virtual PTZ library [19] is employed, so we must follow the coordinate system criteria used by that library. We start with the case of camera orientation $(pan, tilt) = (0, 0)$, which corresponds to see the positive Z axis from the origin. The general case, when $pan \neq 0, tilt \neq 0$ is obtained from the particular case by means of a coordinate system transformation.

The coordinates of vector OA in coordinate system O can be found as projections on the coordinate axes:

$$(vx, vy, vz) = (x - w/2, h/2 - y, r) \quad (1)$$

where $vz = r$, since points A and C are located on the same plane. vx and vy is calculated taking in account that A on the projection plane is based relatively of the left top corner of the frame. Radius r is calculated using the vertical field of view (FOV), which is known. We only need to notice that $\widehat{FOC} = \widehat{FOV}/2$, and FOC is a right triangle, so

$$\tan(\widehat{FOV}/2) = \frac{FC}{r} \Leftrightarrow r = \frac{h/2}{\tan(\widehat{FOV}/2)} \quad (2)$$

This way the camera coordinates are computed. In order to extend the particular case to the generic case and obtain the world coordinates we need to recalculate vector OC from the O coordinate system into the rotated coordinate system O' . The coordinates are found by means of multiplication by the inverse of a double rotation matrix. This matrix can be found by multiplication of two rotation matrices: around X and Z axes by $(pan, tilt)$:

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(tilt) & -\sin(tilt) \\ 0 & \sin(tilt) & \cos(tilt) \end{pmatrix} \cdot \begin{pmatrix} \cos(pan) & -\sin(pan) & 0 \\ \sin(pan) & \cos(pan) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

$$A' = (vx', vy', vz') = R^{-1} \cdot (vx, vy, vz) \quad (4)$$

Now, when the new coordinates A' are computed, we only need the spherical coordinates transformation to obtain θ and ϕ :

$$\begin{cases} vx' = R \cos(\phi) \sin(\theta) \\ vy' = R \sin(\phi) \sin(\theta) \\ vz' = R \cos(\theta) \end{cases} \quad (5)$$

By rearranging the formulas:

$$\begin{cases} \phi = \arctan\left(\frac{vy'}{vx'}\right) \\ \theta = \arccos\left(\frac{vz'}{R}\right) = \arccos\left(\frac{vz'}{\sqrt{vx'^2 + vy'^2 + vz'^2}}\right) \end{cases} \quad (6)$$

The obtained samples are real valued vectors, whose two first components are the spherical coordinates ϕ and θ of a pixel, while the third last ones are the observed RGB color values at that pixel:

$$\mathbf{x} = (\theta, \phi, r, g, b) = (x_1, x_2, x_3, x_4, x_5) \quad (7)$$

so that $\mathbf{x} \in \mathbb{R}^5$. At each time instant t , a new episode $\mathcal{S}_t \subset \mathbb{R}^5$ is acquired which contains one sample for each pixel of the incoming video frame at time t . Therefore the cardinal of \mathcal{S}_t is the number of pixels in the video frame.

2.2. Competitive learning network

Next a competitive learning model for panoramic scenes captured by PTZ cameras is developed. In order to learn the details of the scene, a large number of neurons N is employed, which is lower but in the same order of magnitude as the number of pixels of the full panorama. Following the strategy in [20], the input vectors $\mathbf{x} \in \mathbb{R}^5$ are divided into two sections. The first section contains the positional information in the video frame, while the second section contains the color features. In our case, the first section comprises the two spherical coordinates ϕ and θ , while the second section contains the RGB color values. All the components of the input vectors are used to update the neuron prototypes $\mathbf{w}_i \in \mathbb{R}^5$, but only the two first ones participate in the competition. This way, only the positional information is employed to determine which neuron is the winner. Therefore, each neuron represents the average color in a receptive field which is one region of a Voronoi tessellation of the full panorama. That is, the neurons specialize on small pixel neighborhoods of the panorama. Since the positional components can have fractional values, $(x_1, x_2) \in \mathbb{R}^2$, the network can learn fractional positional information without having to round to integer pixel positions in the panorama, which avoids losing valuable information.

Each neuron contains a prototype $\mathbf{w}_i \in \mathbb{R}^5$ and also a Boolean flag $b_i \in \{true, false\}$. The flag is required to control the initialization of the neuron. Since the PTZ camera does not cover the entire panorama at a time, it is not possible to initialize all the neurons at the same time. They can only be initialized while their positional components fall into the current field of view. To this end, the Boolean flags are initialized to false. The first time that a neuron wins, its prototype is set to the input sample, and its flag is set to true.

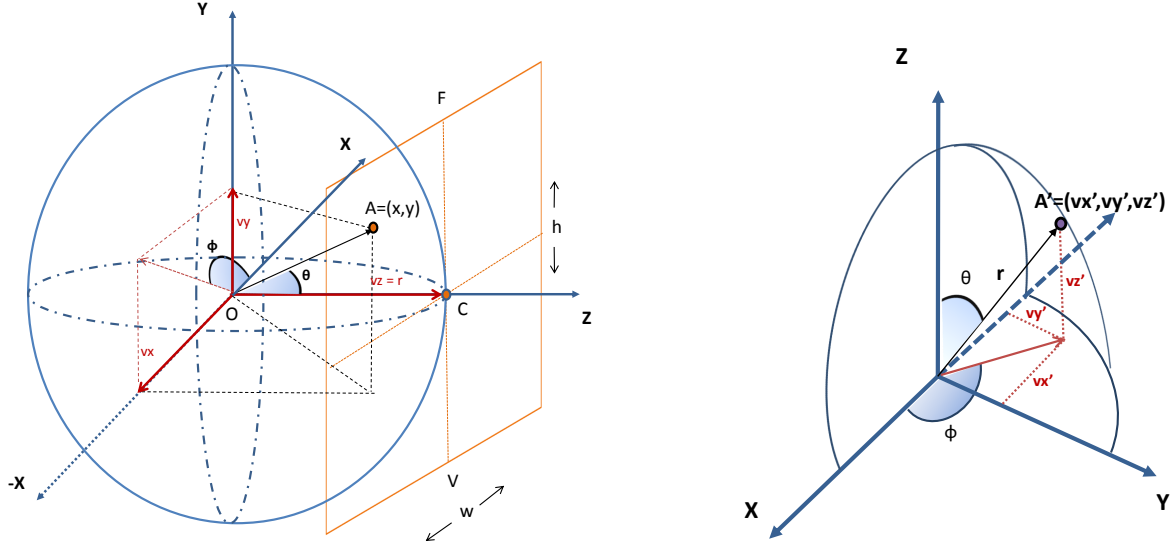


Figure 1. Schemes of the PTZ camera model. a) Left: pinhole camera model with the original orientation. b) Right: reoriented scheme to obtain spherical coordinates.

From that point on, the neuron will be updated according to the competitive learning rule.

The proposed learning algorithm is as follows:

- 1) Draw a training sample \mathbf{x} at random from the current episode \mathcal{S}_t , which has not been considered before, i.e. a random sampling without replacement is done.
- 2) Find the nearest neuron q in terms of Euclidean distance according to the first two vector components:

$$q = \arg \min_{i \in \{1, \dots, N\}} \|(w_{i,1}, w_{i,2}) - (x_1, x_2)\| \quad (8)$$

- 3) If the flag b_q is true, then go to step 4. Otherwise, set the prototype of the winning neuron q to the training sample and set its flag to true:

$$\mathbf{w}_q = \mathbf{x} \quad (9)$$

$$b_q = true \quad (10)$$

Then go to step 5.

- 4) Update the winning neuron prototype according to the standard competitive learning rule:

$$\Delta \mathbf{w}_q = \eta(t) (\mathbf{x} - \mathbf{w}_q) \quad (11)$$

where $\eta(t)$ is a decaying learning rate that varies depending on the time step t .

- 5) If all the samples of the current episode \mathcal{S}_t have already been processed, then go to step 6. Otherwise, go to step 1.

- 6) If the last time instant t has been reached, then stop. Otherwise, increment the time instant counter t , load the next episode and go to step 1.

There are two phases in the learning process: first the ordering phase where η experiences a linear decay (initial learning rate η_I); and then the convergence phase where η remains constant at a small value (η_C). This is because the ordering phase is required for the warm-up of the algorithm only, and after that the system runs for an indefinitely long time. The change of phase is monitored by a step parameter n .

$$\eta(t) = \begin{cases} \eta_I(1 - t/n), & \text{if } t < n \\ \eta_C, & \text{otherwise} \end{cases} \quad (12)$$

A theoretical analysis of the above algorithm can be carried out. Since the competition is done on the first two components, the algorithm seeks a local minimum of an energy function \mathcal{E} which only takes into account these components:

$$\mathcal{E} = \sum_{i \in \{1, \dots, N\}} \sum_{\mathbf{x} \in F_i} \|(w_{i,1}, w_{i,2}) - (x_1, x_2)\|^2 \quad (13)$$

where F_i is the receptive field of the i -th neuron:

$$F_q = \left\{ \mathbf{x} \mid q = \arg \min_{i \in \{1, \dots, N\}} \|(w_{i,1}, w_{i,2}) - (x_1, x_2)\| \right\} \quad (14)$$

On the other hand, the prototype update is carried out on the three last components too, so those components approximate the average color of the receptive field:

$$(w_{i,3}, w_{i,4}, w_{i,5}) \approx E[(x_3, x_4, x_5) \mid F_i] \quad (15)$$

As a last remark, it must be pointed out that the large number of neurons N to be used for this application requires a considerable optimization of the competition equation (8). This is accomplished by inserting all the first sections $(w_{i,1}, w_{i,2})$ of the neuron prototypes into a quad-tree [21]. This way (8) is evaluated very quickly, even for values of N in the millions.

2.3. Delaunay triangulation based interpolation

Since the positions of the neuron prototypes are given by real numbers, there is no direct way to obtain the estimated color for the integer valued pixel coordinates of the panorama. In order to overcome this difficulty, we propose to build the Delaunay triangulation [22] of the set formed by the first sections $(w_{i,1}, w_{i,2})$ of each neuron prototype. This way, a triangulation of the panorama is obtained. Then, for each integer valued pair of pixel coordinates $(y_1, y_2) \in \mathbb{N}^2$, the triangle which it belongs to is computed, along with its barycentric coordinates with respect to that triangle:

$$\mathbf{y} = \lambda_i (w_{i,1}, w_{i,2}) + \lambda_j (w_{j,1}, w_{j,2}) + \lambda_k (w_{k,1}, w_{k,2}) \quad (16)$$

$$i, j, k \in \{1, \dots, N\} \quad (17)$$

$$\lambda_i, \lambda_j, \lambda_k \geq 0 \quad (18)$$

$$\lambda_i + \lambda_j + \lambda_k = 1 \quad (19)$$

Then the estimated color $(y_3, y_4, y_5) \in \mathbb{R}^3$ is obtained by linear interpolation with weights equal to the barycentric coordinates:

$$(y_3, y_4, y_5) = \lambda_i (w_{i,3}, w_{i,4}, w_{i,5}) + \lambda_j (w_{j,3}, w_{j,4}, w_{j,5}) + \lambda_k (w_{k,3}, w_{k,4}, w_{k,5}) \quad (20)$$

Therefore a continuous, piecewise linear function is employed to estimate the color over the panorama.

3. Experimental results

In this section we report the computational experiments we have carried out and their results. The software and hardware that have been used are specified in Subsection 3.1. Then, the tested video sequences are described in 3.2. The descriptions of the used parameters are in Subsection 3.3 and the descriptions of the competitors in 3.4. Finally, the obtained results from the experiments are reported in Subsection 3.5.

3.1. Methods

The camera control module is based on the *virtualptz library* [19]. It simulates a PTZ camera from a panoramic video sequence, and it is accessible from its website ¹. The implementation is written in C++ and it uses the OpenCV and OpenGL libraries. This virtual camera has limitations in its vertical movement, going from 0 (up) to 180 (down) degrees.

To generate the input data for the competitive neural net and for the competitors, an exhaustive scanning of the scene has been carried out. To simulate the real behavior of a PTZ camera in a practical setting, some limitations in the scan have been imposed. Starting at the top (vertical 0 degrees), we turn 360 degrees to the right with a step of 10 degrees, and when we come back to the initial point, we go down 10 degrees. This process is repeated until we arrive to the bottom and we start to go up again. Furthermore, a random zoom (vertical field of view) is applied to each frame, but again, we try to simulate a real situation. To this end, we generate a random number in an interval $[FOV, FOV + 5]$ and move this interval by steps of 5 degrees between a minimum and maximum value for the field of view, $[70, 140]$ degrees, in order to avoid strange images with irregularities.

A total of 3000 camera frames have been saved in binary files to read them synchronously by all the methods, which have been implemented in Matlab R2015b. The reported experiments have been carried out on a 64-bit Personal Computer with an eight-core Intel i7 3.60GHz CPU, 32 GB RAM and standard hardware. The implementation of our approach does not use any GPU resources.

Panoramic ground truth (GT) has been calculated doing the median of the raw panoramic video frames. For each method, we compared the panoramic image obtained with the ground truth. Three quality measures were used to evaluate the proposed approach: the first was the Mean Squared Error (*MSE*) metric (lower is better), which is commonly used in image processing; the second was the Structural Similarity index (higher is better), which focuses on structural similarities between images:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (21)$$

where μ_x and μ_y are the mean value of images x and y , σ_x and σ_y are the standard deviation of images x and y , σ_{xy} is the covariance of x and y , $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ (L is the dynamic range, $k_1 = 0.01$ and $k_2 = 0.03$). The values obtained from (21) are averaged over the three RGB channels to obtain the performance for color images. The third quantitative performance measure is the Bhattacharyya coefficient *BC* [23], which measures the closeness of the two discrete pixel probability distributions P and \hat{P} corresponding to the GT and modeled images:

$$BC = \sum_{j=0}^{255} P(j)\hat{P}(j) \quad (22)$$

1. https://bitbucket.org/pierre_luc_st_charles/virtualptz_standalone

TABLE 1. CONSIDERED PARAMETER VALUES FOR OUR COMPETITIVE LEARNING METHOD

Parameter	Value
Number of neurons N	$(\text{Number of pixels})/2$
Initial learning rate η_I	0.4
Convergence learning rate η_C	0.01
Number of steps n	1000

where $BC \in [0, 1]$ and higher is better.

In addition, CPU times were measured offline for each input binary file to obtain the real running time of each algorithm per frame, without including the extra waiting time to acquire the next frame that would be required if we did it online.

3.2. Sequences

For the experiments we used two video databases. One of them has three videos which are available on the virtualptz website. They are three indoor sequences, two of them with the same scene, and they are named *scenario3*, *scenario4* and *scenario5*. Because the two first ones are very similar, we have just shown the image results of the *scenario3* (3500x1750 pixels and 566 video frames). The video *scenario5* (3500x1750 pixels and 1957 frames) shows a room with people moving on in and doing different actions. On the other hand, we used one video sequence of the Littlstar web page². We named this video as *scenario6* (2880x1440 pixels and 1169 frames) and shows a beach with people moving and playing beach volleyball. Statistics were calculated using the four sequences.

3.3. Parameter selection

A set of tuned parameters is needed to define the competitive learning model. These fixed parameters are reported in Table 1.

In addition, a study of the number of neurons has been done and the results are displayed in Fig. 2. As seen, the equilibrium state is reached quickly and excellent results are obtained from only 612500 neurons (10% of total pixels).

3.4. Competitors

We have compared our proposed method with other three methods. The competing methods read each frame, then position it in the panoramic image matrix and finally compute the mean over all the frames.

Since data input coordinates are almost always fractional numbers, the panoramic integer coordinates where the incoming frame must be placed were calculated by rounding the original fractional coordinates. Besides, the coordinate pairs do not define a regular strictly monotonic grid, i.e., the points have no structure or order between their relative

locations, so that the usual interpolation methods defined to obtain the interpolated RGB values in a panorama cannot be used. Therefore, the scattered interpolant method from MATLAB was employed to manage this situation. It provides the following interpolation variants: 'nearest' (nearest-neighbor interpolation), 'linear' (linear interpolation), and 'natural' (natural-neighbor interpolation). The same input data was provided to all the competitors as well as our method, i.e. the 3000 camera frames saved in binary files.

3.5. Results

A comparison of all the evaluated methods for one of the selected videos is reported in Fig. 3. This figure shows the comparative evolution of the methods with respect to the number of captured PTZ camera frames, i.e. the frame index in the video sequence. It can be observed that when a certain number of frames is reached that covers almost all the panoramic image, which is around 500 frames, our method (in red) attains the best performance values, as compared to the competing methods. In particular, MSE enhances substantially, where lower values indicates that the obtained RGB pixel values are more precise. When the number of acquired frames increases to more than 2500, all methods tend to perform similarly, but our method still remains as the best one.

Something similar happens for the other three videos. To summarize it, we have calculated the mean and standard deviation of the four sequences for each method and for each performance measure. These qualitative results are shown in Table 2. As we can see, our method outperforms the competing methods, particularly in the mean squared error. Higher values of SSIM and BC confirm that our competitive neural model produces the best approximation of the background of the scene.

The CPU time required to process one frame is a very important feature to be assessed. Table 2 shows the mean required time to process a binary file for a frame. Our method is around 85% faster than the interpolation methods. It computes the winner neurons and updates the quad-tree where they are stored in just over a second. If we consider that a movement of the virtual PTZ camera and the generation of the binary file for the current frame takes between one and three seconds, it turns out that our method is the only one that can be executed concurrently with the PTZ camera frame acquisition process. The competitors triplicate the required time and incur in a big time delay to obtain the panoramic background. The utilization of a GPU would improve between 25 and 50 times the processing rate.

In order to get a qualitative point of view about the suitability of our approach, the panoramic backgrounds generated by each method have been compared. In Fig. 5 a window of 50x50 pixels of two scenarios are displayed, with their respective raw panoramic video frame. As it can be observed, the qualitative results offered by our approach are the most similar to the ground truth. This can be better appreciated on the top and the bottom of the panoramic image, where the competitors produce black pixels that

2. <https://littlstar.com/>

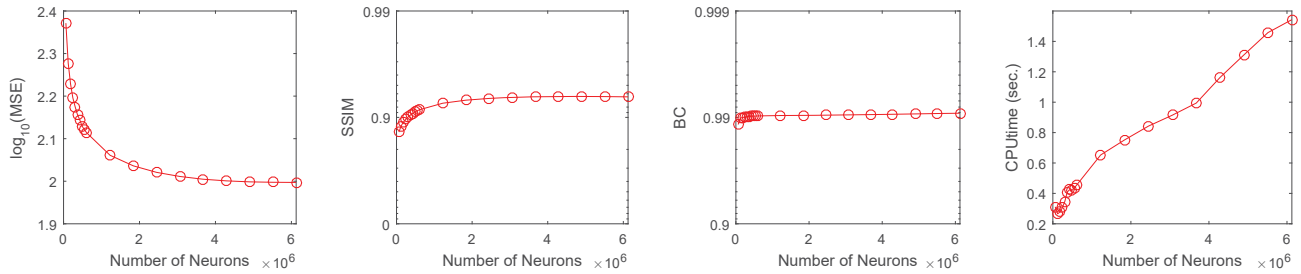


Figure 2. Evolution of the MSE, SSIM, BC and CPU time varying the number of neurons. *scenario3* was used for the analysis. MSE has been displayed in a logarithmic scale (lower is better), SSIM and BC in a reverse logarithmic scale (higher is better).

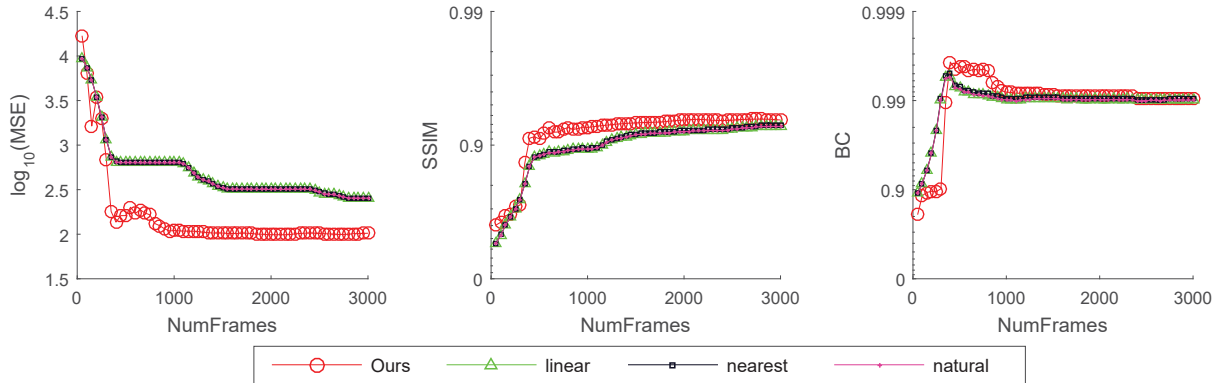


Figure 3. Comparative of the MSE, SSIM and BC for the four methods with *scenario3*. MSE has been displayed in a logarithmic scale (lower is better), SSIM and BC in a reverse logarithmic scale (higher is better).

should not be there. The window of *scenario6* is a clear example of this. Also, in the left and the right of the images produced by the competitors there are more black pixels than in our approach. With respect to the moving people of the video frame, all methods remove them efficiently and no big differences are noticed. An example of the result of our method is shown in Fig. 4.

4. Conclusion

In this work, a methodology to model the panoramic background of PTZ cameras is presented. It consists of an online learning method based on a competitive neural network to read each camera frame and process it to generate a panoramic image of the scene. Four scenes have been tested to check the feasibility of the system, obtaining suitable and successful results. Also, it has been demonstrated that our approach outperforms several competing methods. It is remarkable that the proposed model uses very little CPU time to process each input video frame, which permits a easy integration in a real time PTZ camera video surveillance system.

Acknowledgments

This work is partially supported by the Ministry of Economy and Competitiveness of Spain under grant TIN2014-

53465-R, project name Video surveillance by active search of anomalous events. It is also partially supported by the Autonomous Government of Andalusia (Spain) under projects TIC-6213, project name Development of Self-Organizing Neural Networks for Information Technologies; and TIC-657, project name Self-organizing systems and robust estimators for video surveillance. Finally, it is partially supported by the Autonomous Government of Extremadura (Spain) under the project IB13113. All of them include funds from the European Regional Development Fund (ERDF). The authors thankfully the grant of the University of Málaga and acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga. They also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU. Karl Thurnhofer-Hemsi is funded by a PhD scholarship from the Spanish Ministry of Education, Culture and Sport under the FPU program.

References

- [1] Chung-Chen Chen, Yi Yao, A. Drira, A. Koschan, and M. Abidi, "Co-operative mapping of multiple PTZ cameras in automated surveillance systems," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1078–1084.
- [2] C. Ding, B. Song, A. Morye, J. Farrell, and A. Roy-Chowdhury, "Collaborative sensing in a distributed ptz camera network," *IEEE*

TABLE 2. MEAN (STANDARD DEVIATION) MEASURES FOR EACH METHOD ALONG THE FOUR SCENARIOS

	<i>MSE</i>	<i>SSIM</i>	<i>BC</i>	<i>Time(sec/frame)</i>
Ours	106.87 (23.09)	0.9184 (0.0371)	0.9703 (0.0466)	1.1427 (0.1284)
Linear	233.92 (35.60)	0.9167 (0.0288)	0.9701 (0.0460)	7.6549 (0.8106)
Nearest neighbor	235.00 (35.12)	0.9176 (0.0282)	0.9701 (0.0463)	7.7251 (0.8171)
Natural neighbor	234.00 (35.63)	0.9162 (0.0292)	0.9700 (0.0459)	9.1097 (1.0477)



Figure 4. Panoramic images of *scenario5* for frame 153 and the proposed method, resp.

- Transactions on Image Processing*, vol. 21, no. 7, pp. 3282–3295, 2012.
- [3] K. Konda, N. Conci, and F. De Natale, “Global coverage maximization in ptz-camera networks based on visual quality assessment,” *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6317–6332, 2016.
- [4] T. Boulton, X. Gao, R. Micheals, and M. Eckmann, “Omni-directional visual surveillance,” *Image and Vision Computing*, vol. 22, no. 7, pp. 515–534, 2004.
- [5] C.-H. Chen, Y. Yao, D. Page, B. Abidi, A. Koschan, and M. Abidi, “Heterogeneous fusion of omnidirectional and ptz cameras for multiple object tracking,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1052–1063, 2008.
- [6] S. Kim, K. Yun, K. Yi, S. Kim, and J. Choi, “Detection of moving objects with a moving camera using non-panoramic background model,” *Machine Vision and Applications*, vol. 24, no. 5, pp. 1015–1028, 2013.
- [7] K. Xue, Y. Liu, G. Ogunmakin, J. Chen, and J. Zhang, “Panoramic gaussian mixture model and large-scale range background subtraction method for ptz camera-based surveillance systems,” *Machine Vision and Applications*, vol. 24, no. 3, pp. 477–492, 2013.
- [8] P. D. Z. Varcheie and G. A. Bilodeau, “Adaptive fuzzy particle filter tracker for a PTZ camera in an IP surveillance system,” *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 2, pp. 354–371, 2011.
- [9] F. J. López-Rubio and E. López-Rubio, “Foreground detection for moving cameras with stochastic approximation,” *Pattern Recognition Letters*, vol. 68, pp. 161–168, 2015.
- [10] P. Azzari and A. Bevilacqua, “Joint spatial and tonal mosaic alignment for motion detection with PTZ camera,” *Lecture Notes in Computer Science*, vol. 4142, pp. 764–775, 2006.
- [11] A. Bevilacqua and P. Azzari, “High-Quality Real Time Motion Detection Using PTZ Cameras,” *Proceedings of the IEEE International Conference on Video and Signal Based Surveillance (AVSS 2006)*, pp. 1–6, 2006.
- [12] S. Ahalt, A. Krishnamurthy, P. Chen, and D. Melton, “Competitive learning algorithms for vector quantization,” *Neural Networks*, vol. 3, no. 3, pp. 277–290, 1990.
- [13] T. Uchiyama and M. Arbib, “Color image segmentation using competitive learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 12, pp. 1197–1206, 1994.
- [14] E. Yair, “Competitive learning and soft competition for vector quantizer design,” *IEEE Transactions on Signal Processing*, vol. 40, no. 2, pp. 294–309, 1992.
- [15] H.-P. Chen, X.-J. Shen, and J.-W. Long, “Histogram-based colour image fuzzy clustering algorithm,” *Multimedia Tools and Applications*, vol. 75, no. 18, pp. 11 417–11 432, 2016.
- [16] E. Ozan, S. Kiranyaz, and M. Gabbouj, “Competitive quantization for approximate nearest neighbor search,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 11, pp. 2884–2894, 2016.
- [17] R. Valente and T. Abrao, “MIMO transmit scheme based on morphological perceptron with competitive learning,” *Neural Networks*, vol. 80, pp. 9–18, 2016.
- [18] H. Xie, X. Luo, C. Wang, S. Liu, X. Xu, and X. Tong, “Multispectral remote sensing image segmentation using rival penalized controlled competitive learning and fuzzy entropy,” *Soft Computing*, vol. 20, no. 12, pp. 4709–4722, 2016.
- [19] G. Chen, P. St-Charles, W. Bouachir, T. Joesseint, G. Bilodeau, and R. Bergevin, “Reproducible evaluation of pan-tilt-zoom tracking,” *CoRR*, vol. abs/1505.04502, 2015. [Online]. Available: <http://arxiv.org/abs/1505.04502>
- [20] R. M. Luque-Baena, E. López-Rubio, E. Domínguez, E. J. Palomo, and J. M. Jerez, “A self-organizing map to improve vehicle detection in flow monitoring systems,” *Soft Computing*, vol. 19, no. 9, pp. 2499–2509, 2015.
- [21] S. F. Frisken and R. N. Perry, “Simple and efficient traversal methods for quadtrees and octrees,” *Journal of Graphics Tools*, vol. 7, pp. 1–11, 2002.
- [22] D. Lee and B. Schachter, “Two algorithms for constructing a delaunay triangulation,” *International Journal of Computer & Information Sciences*, vol. 9, no. 3, pp. 219–242, 1980.
- [23] A. Bhattacharyya, “On a measure of divergence between two multinomial populations,” *Sankhy: The Indian Journal of Statistics (1933-1960)*, vol. 7, no. 4, pp. 401–406, 1946. [Online]. Available: <http://www.jstor.org/stable/25047882>





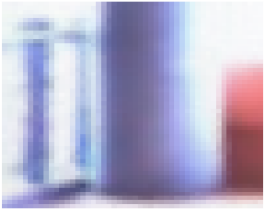


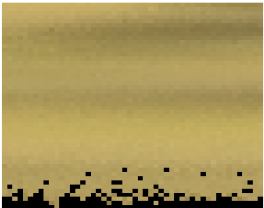
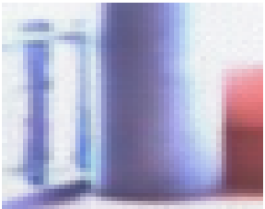
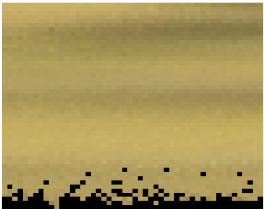

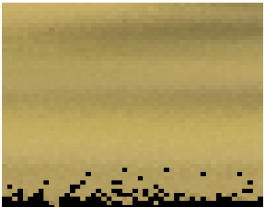
	<i>scenario3</i>	<i>scenario6</i>
Frame		
GT		
Proposed		
linear		
nearest		
natural		

Figure 5. Graphical depiction of the operation of the proposed method. Each row show, from left to right, up to down: a frame of a raw panoramic video sequence, and sections (red square) of its ground truth, our algorithm, and the three competitors algorithms. Frame 153 of scenario3 and frame 859 of scenario6 are shown.