PTZ Camera Network Reconfiguration

C. Piciarelli
University of Udine
Via Margreth 3
33100 Udine, Italy
claudio.piciarelli@dimi.uniud.it

C. Micheloni
University of Udine
Via Margreth 3
33100 Udine, Italy
christian.micheloni@dimi.uniud.it

G.L. Foresti
University of Udine
Via Margreth 3
33100 Udine, Italy
gianluca.foresti@dimi.uniud.it

***Abstract*—Vision Network based surveillance systems are more and more common in public places. Typically, a mixture of static and Pan-Tilt-Zoom (PTZ) cameras is used. Modern systems task PTZ cameras as a consequence of particular events needing further investigation; anyhow, the configuration of the network can be considered fixed and determined at the moment of deployment. In this work, we deal with a problem that has not yet been widely addressed: how a network can automatically change its configuration to enhance the monitoring capabilities. In particular, we propose a novel network reconfiguration algorithm that, given a map of activities, configures the Pan, Tilt and Zoom parameters of all the cameras in order to improve the detection. A spherical model to project all the activities in the monitored area with respect to the optical centre of each camera is introduced. Such a model leads to an optimization problem that can be solved by means of the Expectation-Maximization algorithm and whose solutions are the new Pan, Tilt and Zoom values for each PTZ camera. Experimental results will be proposed with both synthetic and real data to show how the proposed algorithm can be applied to different cases.**

I. Introduction

Sensor networks and computer vision have been the core of many researches in the last decade. A natural application has been video surveillance by means of video networks. In this field, a lot of effort has been conducted and many achievements have been obtained. Complex and wide systems have been proposed to solve problems from object detection to tracking and finally behaviour analysis [5]. From single footages acquired by static cameras, more recent developments exploit large video networks equipped with different kind of visual sensors like common cameras, forward looking infrared (FLIR) and Pan Tilt Zoom (PTZ) cameras. Problems that seemed to be almost solved for single cameras, determined new situations that opened new questions once projected in a multi-sensor environment. Among these, tracking is one of the most studied problems. In this context, problems like the handover between cameras with or without overlapping fields of view have been successfully addressed for static cameras. Instead, when PTZ cameras are considered, many problems arise. Indeed, while static networks require only post-acquisition analysis, dynamic networks need also a management strategy for the sensors. In particular, two stages can be determined. First, a static configuration of the sensors has to be decided to maximize coverage and minimize resources. Then, a strategy for planning, programming and controlling the network of sensors is worthwhile to acquire data from the dynamic environment. Moreover, it is interesting to define such a strategy in order to improve the performance of the network while keeping the required resources. These two phases have been marginally studied if compared to the current research emphasis on sensors networks [1] that principally considers distribution levels for the communication [6], power consumption [4], embedding [2], scalability [9] and security [3]. Considering the distribution levels connected to data acquisition represents a novel research field. For example, sensors crashes, communication breakdowns, data degradation or black out due to wrong coverage require new methodologies allowing the network to react in order to reduce the effects of such problems.

Concerning the static configuration of the sensors, Mittal and Davis [10], [11] proposed an interesting approach for the optimal deployment of static sensors. With respect to previous works the proposed methodology considers not only occlusions due to static objects in the scene but also dynamic objects. A probabilistic framework for the visual coverage of the dynamic scene is therefore introduced to take into account the average environmental situation. Sensor network deployment works like those proposed by Karuppiah *et al.* [8] or Park *et al.* [13] can be adopted to determine the optimal subset of cameras necessary for optimally acquiring the targets. In [8], two metrics based on the dynamics of the scene are introduced to determine the pair of cameras that maximize the probability of tracking people moving within the monitored scene. In [13], a distributed look-up table based approach is proposed to determine the cameras' viewing frustums that allows to select the cameras for tracking purposes. Such an approach is interesting since allows to reduce the network traffic for the camera selection task.

The aforementioned approaches propose solutions for static sensors and do not consider the possibility for reconfiguring a dynamic sensor network. In this context, to solve the reconfiguration problem, a first approach has been proposed by Kansal *et al.* [7]. In particular, a laser system is proposed to define map of all the static obstacles in the scene. When an event of interest is detected, the map is exploited to select the best high resolution sensor. Such a camera is redirected to the location of the event of interest while low resolution cameras are reconfigured to cover the remaining zones. Starting from such a work we want to introduce a novel approach to answer the question "*how can a network automatically change its* *configuration to enhance the monitoring capabilities?*". In particular, we propose a network reconfiguration algorithm that, given a map of activities, configures the Pan, Tilt and Zoom parameters of all the cameras in order to improve the detection. A spherical model to project all the activities in the monitored area with respect to the optical centre of each camera is introduced. Such a model leads to an optimization problem that can be solved by means of the Expectation-Maximization algorithm and whose solutions are the new Pan, Tilt and Zoom values for each PTZ camera. It is worth noticing that the proposed solution can foresee a development that allows each camera to compute its reconfiguration parameters thus allowing a distributed reconfiguration strategy. As a matter of fact each camera can autonomously determine its reconfigured parameters by knowing the activity map and the position of the other cameras.

II. System Architecture

The proposed solution (see Figure 1) can find application in any visual surveillance system that employees PTZ cameras even in static mode (i.e., the configuration of Pan-Tilt-Zoom parameters is kept fixed). Static cameras configuration with wide angle of view can monitor a large area of the environment, but cannot be reliably used for many surveillance tasks because of their low resolution. On the other hand, PTZ cameras can be pointed and zoomed on specific areas of interest in order to gather useful information (e.g. faces, license plates, etc.) but cannot have a global view of the scene. The proposed work is based on the idea that static PTZ cameras can acquire global information on the activities happening within the scene, and this information can be used to automatically schedule the PTZ cameras to more specific tasks. We will focus on monitoring the zones with highest activity. Depending on the activity of interest (i.e. moving people or vehicles, event of interest, task of interest, etc.) the video surveillance systems extract useful information from all video sensors. In the current development, the activity of interest is the localization of the moving objects in the scene. Thus, the video streams acquired by static PTZ cameras are processed in order to detect and track moving objects; the object trajectories can in turn be processed in order to obtain an activity density map, as shown in Figure 1. The density map is computed by subsampling the scene map in a low-resolution grid, and by counting how many objects pass through each cell. The final aim of this work is to propose an algorithm to automatically reconfigure the PTZ cameras in order to optimally cover the zones with highest density in the activity density map.

III. Network Reconfiguration

The task we are going to tackle with has many similarities with 2D data fitting problems, in which the data distribution is approximated by a mixture of density functions (i.e., Gaussians): in our case there is an activity density map that should be fit by the coverage areas of the PTZ cameras. One of the most popular Mixture-of-Gaussians data fitting algorithms is Expectation-Maximization (EM) [12]; we will first give a short description of the EM algorithm, then we will show how EM can be applied to our map coverage problem.

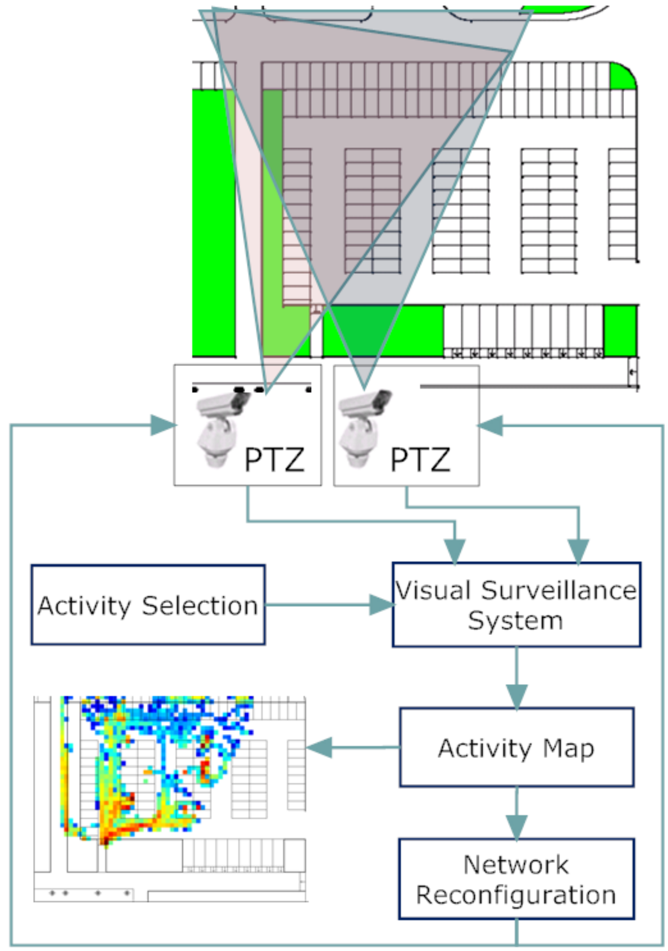


Fig. 1. A real-life scenario with the corresponding activity density map.

A. EM data fitting

Expectation-Maximization is a popular tool for data fitting; although it can be applied to several data models, we here describe the special case of 1D Gaussian fitting. Let us suppose to have a set of data $\{x_1, \ldots, x_n\}$, which can be seen as the realizations of the random variables $X = \{X_1 \ldots X_n\}$. The data are drawn from k Gaussian distributions, thus each element has a label denoting which Gaussian it has been drawn from. The set of labels is denoted $\{z_1, \ldots, z_n\}$ (realizations of the random variables $Z = \{Z_1, \ldots, Z_n\}$) and it is initially unknown. Moreover, the data generation process is governed by a set of hidden parameters Φ, representing mean, variance and weight of each Gaussian distribution. Aim of the data fitting process is to find the set of parameters $\hat{\Phi}$ maximizing the probability of the data set $p(X|\Phi)$. By using the total probability theorem, we get

$$p(X_i = x_i|\Phi) = \sum_{z=1}^{k} p(X_i|Z_i = z, \Phi)p(Z_i = z|\Phi) \quad (1)$$

The term $p(X_i|Z_i = z, \Phi)$ is the likelihood, and it is a Gaussian distribution; $p(Z_i = z|\Phi)$ is the prior probability, constant for each class and independent from i. Eq. 1 can thus be written as

$$P(X_i = x_i|\Phi) = \sum_{z=1}^{k} G(x_i, \mu_z, \sigma_z)c_z \tag{2}$$

where

$$G(x_i, \mu_z, \sigma_z) = \frac{1}{\sqrt{2\pi}\sigma_z} \exp \frac{(x_i - \mu_z)^2}{2\sigma_z^2}$$

and c_z are the weights of each Gaussian model. The optimization problem can thus be expressed as the search for the unknown parameters $\hat{\Phi} = (\mu_1, \sigma_1, c_1, \ldots, \mu_k, \sigma_k, c_k)$ such that

$$\begin{aligned} \hat{\Phi} &= \underset{\Phi}{\operatorname{argmax}}\, p(X|\Phi) \\ &= \underset{\Phi}{\operatorname{argmax}} \ln p(X|\Phi) \\ &= \underset{\Phi}{\operatorname{argmax}} \ln \prod_{i=1}^{n} p(X_i|\Phi) \\ &= \underset{\Phi}{\operatorname{argmax}} \sum_{i=1}^{n} \ln \sum_{z=1}^{k} G(x_i, \mu_z, \sigma_z)c_z \end{aligned} \tag{3}$$

A solution for the optimization problem (3) can be found by setting to zero the partial derivatives w.r.t. μ_z, σ_z and c_z respectively. This leads to the equations

$$\mu_j = \frac{\sum_{i=1}^{n} x_i p_{ij}}{\sum_{i=1}^{n} p_{ij}} \tag{4}$$

$$\sigma_j^2 = \frac{\sum_{i=1}^{n} (x_i - \mu_j)^2 p_{ij}}{\sum_{i=1}^{n} p_{ij}} \tag{5}$$

$$c_j = \frac{1}{n} \sum_{i=1}^{n} p_{ij} \tag{6}$$

for each Gaussian j, where $p_{ij} = p(Z_i = j|X_i = x_i, \Phi)$ is the posterior probability, and can be defined as

$$p_{ij} = \frac{G(x_i, \mu_j, \sigma_j)c_j}{\sum_{z=1}^{k} G(x_i, \mu_z, \sigma_z)c_z} \tag{7}$$

As it can be seen, the unknown parameters depend on the posterior probability and vice-versa; they can thus be computed with an iterative process, starting from a random choice for μ, σ and c. The two iterative steps are respectively called the Expectation step (in which the posterior probability is computed based on the previous values for μ, σ and c) and the Maximization step (in which the Gaussian parameters and weights are found using the previously computed posterior probabilities), hence the name of the algorithm. The iterative process is proven to converge to a local maximum.

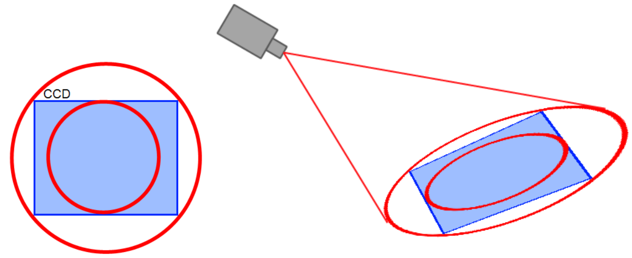


Fig. 2. Approximation of the real camera coverage with two bounding ellipses.

B. Reconfiguration Model

How can EM data fitting be applied to network reconfiguration? The basic idea is that each camera has its own cone of view, representing the portion of space observed by the camera, and the intersection of a cone of view with the ground plane is an ellipse (assuming the camera is not looking above the horizon, otherwise the intersection would be a parabola or an hyperbola). The ellipse represents the ground area observed by a camera and its position, orientation and eccentricity are uniquely defined by the pan, tilt and zoom parameters of the sensor. The problem of network reconfiguration for optimal coverage can thus be reduced to a problem of fitting ellipses to a data set, something strictly related to EM data fitting. A main approximations is used here: even though the optic of a camera has a circular shape, the imaging sensor (the CCD) has not, and thus the observed area has not an elliptical, but rather a trapezoidal shape. In the present work we will not consider this problem and will assume that the observed area can be approximated by an ellipse; a better approximation could be obtained by considering the inscribing and encircling circles of the CCD, and thus approximating the trapezoid by bounding it between two ellipses, as shown in Figure 2.

Another problem to be faced is that EM works with Gaussian distributions, not simply ellipses. However, the isovalue contours of a bivariate Gaussian distribution are ellipses, and thus the camera coverage area can be approximated as the support region of a fixed quantile of a Gaussian distribution. In particular, we can consider ellipses as the set of all the points x with a given Mahalanobis distance R, defined as

$$R = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \tag{8}$$

where μ and Σ respectively are the mean and the covariance matrix of the Gaussian distribution. By mathematical properties of Gaussian distributions, we know that the area enclosed by the support region with $R = 2$ covers the 95% of the total probability distribution, and thus can be a safe approximating choice for our map coverage problem. In other words, if the data set is fitted by a set of n Gaussians, and for each Gaussian we use the ellipse with $R = 2$ as the coverage area for a given camera, the cameras will be granted to cover the 95% of the entire data set.

Finally, which kind of data the data set should be composed of? As stated in the previous section, EM works with a

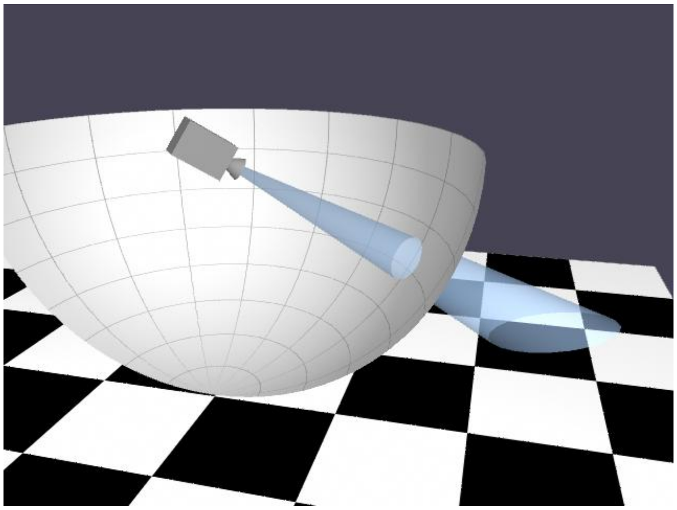


Fig. 3. A camera with the corresponding feature space (the *camera space*). Ellipses on the ground plane become circles in the camera space. Any circle in the camera space is associated to a valid PTZ configuration; the same is not true for ellipses on the ground plane.

discrete set of data elements (in our map-coverage problem, the elements will be two-dimensional vectors representing map coordinates); however what we have is an activity density map as described in section II. The activity map is a matrix A in the form $A(i,j) = n$, meaning that n moving objects passed in the map sector (i,j) during a given observation period. The activity map can be easily converted in a usable form for EM maximization by creating a set X of $m = \sum_i \sum_j A(i,j)$ elements in the form of

$$X = \{\overbrace{(i,j), \ldots, (i,j)}^{A(i,j)\text{times}}\} \quad \forall i,j \tag{9}$$

C. Constrained EM

In the above sections we have shown how the activity density map can be converted in a data set that can be processed by the Expectation-Maximization algorithm, and we discussed how the resulting mixture-of-Gaussians solution can be converted in a set of ellipses covering almost all the data set. However, standard EM cannot be directly applied to network reconfiguration problems, since the resulting ellipses do not necessarily represent the map coverage of any of the available cameras (even though this information could be useful if the network topology has not been fixed yet and the optimal position of the cameras must be chosen). In order to perform network reconfiguration a constrained EM problem must be solved, where only valid ellipses can be selected. The problem can be solved by defining the ellipses no more in terms of mean and covariance values, but expliciting their dependencies in terms of Pan, Tilt and Zoom camera parameters, and solving the optimization problem (3) in terms of these variables. However, the resulting equations quickly become very complex, and an analytic solution can hardly be found, thus requiring the use of more sophisticated, non-analytical optimization strategies.

Instead of following the complex approach of solving an explicitly constrained Expectation-Maximization problem, we propose a much simpler method, consisting in projecting the data in a set of non-constrained feature spaces where standard EM can be applied. An intuitive view of what the new feature spaces should represent is clearly given in Figure 3. For each camera with position (X_c, Y_c) on the map and height Z_c, consider a sphere surrounding the camera, with centre in (X_c, Y_c, Z_c) and radius Z_c. The entire spherical surface is spanned by two coordinates, ϕ and θ, respectively representing the pan and tilt angles of the camera. If the ground plane is projected on the surface of the sphere (actually, only on the lower semisphere), we can represent any map point in terms of (ϕ, θ) coordinates; in other words we shift to a spherical coordinates system:

$$\begin{cases} \phi = \arctan\left(\frac{y-y_c}{x-x_c}\right) \\ \theta = \arctan\left(\frac{(x-x_c)^2+(y-y_c)^2}{z_c}\right) \end{cases} \tag{10}$$

thus transforming the original ground plane (Figure 4(a)) in a spherical space (Figure 4(b)). We further transform this space in polar coordinates in order to make explicit the fact that the system is centred in the camera origin:

$$\begin{cases} u = \theta \cos \phi \\ v = \theta \sin \phi \end{cases} \tag{11}$$

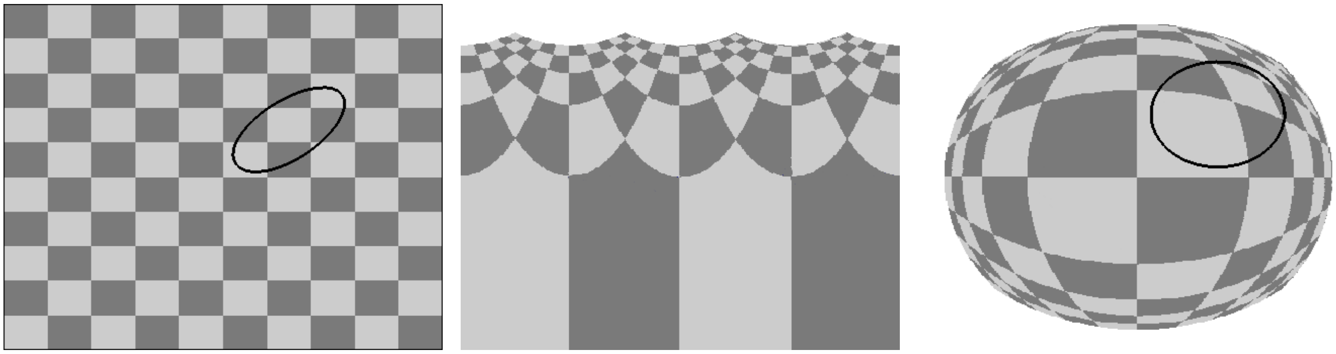
thus moving from the space of Figure 4(b) to the one shown in Figure 4(c). Coverage ellipses in the (x, y) ground plane space become circles in the new (u, v) space (from now on, the *camera space*), as it is intuitively evident by looking at Figure 3.

The advantage of using the camera space is that *any* circle in the camera space corresponds to a valid pan/tilt/zoom camera configuration, as opposed to the original ground plane space, where only a subset of the possible ellipses could be really obtained by the intersection of the camera cone of view with the ground. In other words, the constraints of the original problem disappear if the processing is done in the camera space, and the problem becomes unconstrained.

The standard EM algorithm can thus be applied for each camera in its own camera space, with the only difference that circles must be found, rather than ellipses, and thus the covariance matrix must be in the form $\begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$ where σ is the circle radius. We force this by defining $\sigma^2 = \max(\lambda_1, \lambda_2)$, where λ_1, λ_2 are the eigenvalues of the data covariance matrix; this way the circle radius is set to the length of the major axis of the ellipse.

The whole EM process is now defined as

- for each camera j with position (X_j, Y_j, Z_j), project all



(a) (b) (c)

Fig. 4. Definition of the new feature space. (a): the original (x, y) ground plane; (b) the spherical pan/tilt space; (c) polar version of the pan/tilt space.

the data (x_i, y_i) in the camera space:

$$\phi_{ij} = \arctan\left(\frac{y_i - Y_j}{x_i - X_j}\right)$$
$$\theta_{ij} = \arctan\left(\frac{(x_i - X_j)^2 + (y_i - Y_j)^2}{Z_j}\right)$$
$$u_{ij} = \theta_{ij}\cos\phi_{ij}$$
$$v_{ij} = \theta_{ij}\sin\phi_{ij}$$

- iterate until convergence
  - for each camera $j\ldots$
    - E step:
$$p_{ij} = \frac{G_j(i)c_j}{\sum_{z=1}^{k} G_z(i)c_z}$$
    - M step:
$$\boldsymbol{\mu}_j = \left[\frac{\sum_{i=1}^{n} u_{ij}p_{ij}}{\sum_{i=1}^{n} p_{ij}}, \frac{\sum_{i=1}^{n} v_{ij}p_{ij}}{\sum_{i=1}^{n} p_{ij}}\right]$$
$$\sigma_x^2 = \sum_{i=1}^{n}(u_{ij} - \mu_{j,1})^2 / \sum_{i=1}^{n} p_{ij}$$
$$\sigma_y^2 = \sum_{i=1}^{n}(v_{ij} - \mu_{j,2})^2 / \sum_{i=1}^{n} p_{ij}$$
$$\sigma_{xy} = \sum_{i=1}^{n}(u_{ij} - \mu_{j,1})(v_{ij} - \mu_{j,2})) / \sum_{i=1}^{n} p_{ij}$$
$$\sigma_j^2 = \max\left(\text{eigenvalues}\begin{pmatrix}\sigma_x^2 & \sigma_{xy}\\ \sigma_{xy} & \sigma_y^2\end{pmatrix}\right)$$
$$c_j = \frac{1}{n}\sum_{i=1}^{n} p_{ij}$$

where $G_j(i)$ is the j-th bivariate Gaussian applied to the i-th data element *projected in the j-th camera space*. As it can be seen, there is no need of a unified feature space: each camera can run the optimization process in its own camera space; the only "interaction" between cameras is in the E step, when the probability of a given point in all the camera spaces is needed. Because of this, a distributed implementation is also feasible, in which each camera performs its own optimization process; the only data to be transferred over the network connecting the sensors are the mean and variance of the cameras at each iteration.

Once the iterative process has converged to a solution and $\boldsymbol{\mu}_j$ and σ_j^2 are computed for each camera j, the corresponding pan and tilt angles can be easily computed by applying the inverse of eq. 11:

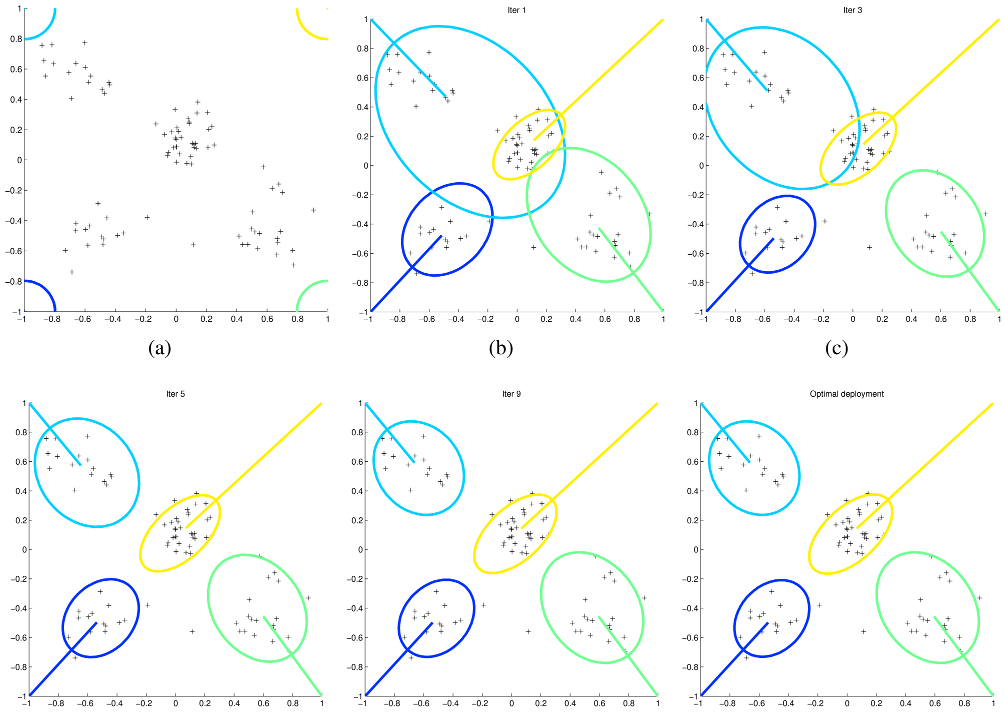
$$\begin{cases}\phi_j = \arctan\left(\frac{\mu_{j,2}}{\mu_{j,1}}\right)\\ \theta_j = \sqrt{\mu_{j,1}^2 + \mu_{j,2}^2}\end{cases} \qquad (12)$$

and $2\sigma_j$ is the angular width of the field of view of camera j. Note that, in practical applications, additional constraints on the values of σ should be applied, in order to model the camera minimum and maximum zoom limits.

IV. Experimental Results

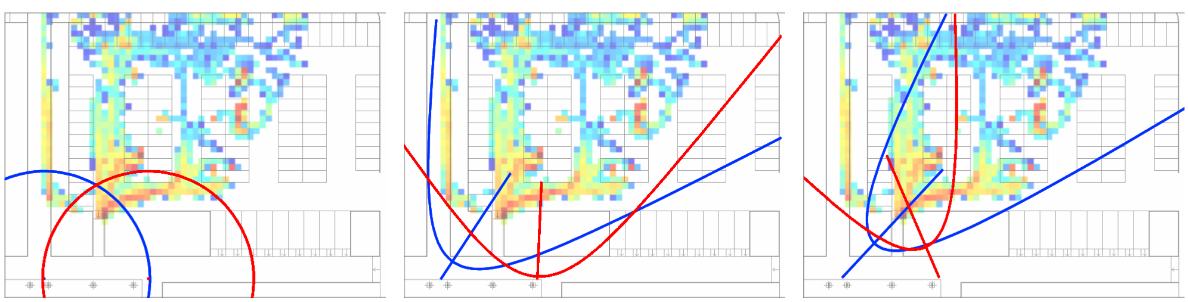
The proposed network reconfiguration technique has been tested both on synthetic and real-world data. In the first case, the dataset has been manually generated in the range $[-1, 1] \times [-1, 1]$ as shown in Figure 5, and it is composed of 85 points. Four cameras with height 1 are placed in the map corners, their initial configuration is $\phi = 0, \theta = 0, \sigma = 0.1$. The algorithm converges in 13 iterations, with a good approximation achieved already after 5 iterations (Figure 5(d)). The iterative process is stopped when no camera has moved by more than 10^{-4} radians in pan or tilt direction among two consecutive iterations. The ellipses shown in Figure 5 are the coverage areas of each camera, associated to the isovalues of each Gaussian where $\sigma = 2$; the straight lines connect the cameras with the centres of the observed areas in the image plane projected into the map (note that this is different from the centre of the ellipses).

Figure 6 instead shows an example taken from a real-world scenario. The map represents a portion of a parking lot, covering an area of 82×58 metres. Two PTZ cameras are mounted on the roof of an adjacent building at an height of 12.1 metres. Initially, the cameras have been configured by a human operator who has been asked to setup the network for monitoring the left section of the parking lot and the passage-way bringing to the building. Two hours long footages from the two cameras have been processed to localize humans and



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

Fig. 5. Network reconfiguration with synthetic data; four cameras are placed in the map corners. (a) Initial configuration; (b) iteration 1; (c) iteration 3; (d) iteration 5; (e) iteration 9; (f) optimal solution found at iteration 13.



(a) (b) (c)

Fig. 6. Network reconfiguration in a real scenario. (a) initial configuration; (b) iteration 1; (c) optimal configuration found at iteration 37.

(a) (b)

Fig. 7. Views of the left ((a)) and right ((b)) PTZ cameras after the proposed reconfiguration.

vehicles in the monitored scene. Localization positions on a 2D map of 640×480 pixels with resolution of $0.128m \times 0.120m$ have been recorded. From such data an activity map of 64×48 cells, each covering a ground area of $1.28m \times 1.2m$, has been created in such a way that each cell contains the number of objects that entered such an area. The reduction from the original map size to the activity map size has been done to smooth data and speed up the EM-analysis. The activity map's data converted in real measurements (i.e., meters) together with the positions of the two cameras have been passed to the proposed algorithm. Using the same initial configuration and termination criterion described above for the synthetic case, the algorithm converges in 37 iterations. The final configurations for the two cameras are $\phi_1 = 45.35, \theta_1 = 68.44, FoV_1 = 29.41$ and $\phi_2 = 114.49, \theta_2 = 65.93, FoV_2 = 36.42$, corresponding to the views shown in Figure 7. As it can be seen, the cameras have been configured in order to cover all the regions of interest.

V. Conclusions

In the current work a novel method for reconfiguring a network of video sensors has been proposed. An EM-based data fitting algorithm has been exploited to determine an optimal configuration of the PTZ parameters to cover the entire area based on an activity map. A new space centred in the optical centre of the camera has been introduced to project the activity map into a space in which EM computation is easier. Results demonstrate the applicability of the proposed algorithm for determining an optimal configuration of the video sensors. In fact, in a real scenario the computed reconfiguration minimizes overlapped zones with respect to the original configuration and acquire the same area with higher definition (higher zoom level for both cameras) thus improving the recognition performance of the system. In our future works we plan to adopt the proposed algorithm to reconfigure the network as consequence of particular events, for example a malfunctioning sensor not acquiring data from its area of interest. The system can be applied by removing the sensor from the EM analysis (i.e., reducing the number of Gaussians) to find out the optimal coverage with the operating sensors. Future developments will also investigate different definitions of the activity map for solving different problems (best configuration for people detection and/or recognition, different configuration for different hours, etc.).

VI. Acknowledgements

This work was partially supported by the Italian Ministry of University and Scientific Research within the framework of the project "Ambient Intelligence: event analysis, sensor reconfiguration and multimodal interfaces" (PRIN 2007-2008).

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[2] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach, "Distributed embedded smart cameras for surveillance applications," *Computer*, vol. 39, no. 2, pp. 68–75, Feb. 2006.

[3] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Conference on Computer and Communications Security*, 2002.

[4] W.-C. Feng, E. Kaiser, W. C. Feng, and M. L. Baillif, "Panoptes: scalable low-power video sensor networking technologies," *ACM ransactions on Multimedia Computing, Communications, and Applications*, vol. 1, no. 2, pp. 151–167, 2005.

[5] G. Foresti, C. Regazzoni, and R. Visvanathan, "Scanning the issue / technology - special issue on video communications, processing and understanding for third generation surveillance systems," *Proceedings of IEEE*, vol. 89, no. 10, pp. 1355–1367, 2001.

[6] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *Mobile Computing and Communications Review*, vol. 5, pp. 10–24, 2002.

[7] A. Kansal, W. Kaiser, G. Pottie, M. Srivastava, and G. Sukhatme, "Reconfiguration methods for mobile sensor networks," *ACM Transaction on Sensor Networks*, vol. 3, no. 4, p. 22, 2007.

[8] D. Karuppiah, R. Grupen, A. Hanson, and E. Riseman, "Smart resource reconfiguration by exploiting dynamics in perceptual tasks," in *International Conference on Intelligent Robots and Systems*, Edmonton, Canada, Aug 2-6 2005, pp. 1513–1519.

[9] K. Martinez, J. K. Hart, and R. Ong, "Environmental sensor networks," *Computer*, vol. 37, pp. 50–56, 2004.

[10] A. Mittal and L. S. Davis, "Visibility analysis and sensor planning in dynamic environments," in *European Conference on Computer Vision*, Prague, CZ, May 2004.

[11] ——, "A general method for sensor planning in multi-sensor systems: extension to random occlusion," *International Journal of Computing Vision*, vol. 76, pp. 31–52, 2008.

[12] T. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[13] J. Park, P. C. Bhat, and A. C. Kak, "A look-up table based approach for solving the camera selection problem in large camera networks," in *Workshop on Distributed Smart Cameras*, Boulder, CO, USA, Oct. 31 2006.