Occlusion-aware multiple camera reconfiguration

C. Piciarelli
Department of Mathematics and Computer Science
University of Udine
Via delle Scienze 206, 33100 Udine, Italy
claudio.piciarelli@uniud.it

C. Micheloni
Department of Mathematics and Computer Science
University of Udine
Via delle Scienze 206, 33100 Udine, Italy
christian.micheloni@uniud.it

G.L. Foresti
Department of Mathematics and Computer Science
University of Udine
Via delle Scienze 206, 33100 Udine, Italy
gianluca.foresti@uniud.it

ABSTRACT

This paper deals with the problem of camera networks reconfiguration. In particular, the case of Pan-Tilt-Zoom (PTZ) cameras is considered, and a method is proposed in order to automatically change the pan, tilt and zoom parameters in order to maximize the coverage of relevant portions of the observed environment. Here, the "relevant portions" are defined in terms of motion activity maps, measuring the passage of moving objects over a map of the monitored scene, however the method can be applied to arbitrary maps. Moreover, occlusions are explicitly handled, so that the map is different for each camera, depending on which portions of the scene are visible from a given point of view. The proposed technique works by approximating the observed zones with ellipses and finds a locally optimal solution by using the Expectation Maximization algorithm. In order to avoid unfeasible solutions (ellipses that cannot be obtained by any PTZ configuration) the computation is performed in a proper space where the geometric constraints due to the camera positions become null.

1. INTRODUCTION

Networks of video sensors are nowadays becoming more and more common in order to meet the requirements of many real-world scenarios, generally focused on security-oriented applications or more generally on ambient monitoring. While networks of static cameras were the only sensible choice in the past years, today the recent developments in the hardware industry have lead to the production of cheap Pan-Tilt-Zoom (PTZ) cameras, which can be successfully used in modern systems. However, the use of such sensors is often limited to human operators manually changing the camera orientation and/or zoom. An important aspect of PTZ networks has not been deeply investigated yet: how can the pan, tilt and zoom parameters be configured in order to maximize the network performance according to a specific task? For example, consider a surveillance system whose aim is to detect activity patterns (e.g. passing people): what is the best camera configuration in order to guarantee maximum coverage in the areas where the most relevant activities have been detected? The question is not trivial, since it must take in consideration several aspects such as camera constraints (the position of the cameras are generally fixed, and only PTZ parameters can be modified), presence of occlusions in the monitored environment, and dynamic changes (e.g. in the activity data, or in the network topology because of the failure of a sensor) that could make the previous configuration obsolete.

Since this is a relatively recent research area, the state of the art in this sector is limited. Karuppiah *et al.* [3] proposed two new metrics that, based on the dynamics of the scene, allow to choose the pair of cameras that maximize the detection probability of a moving object. In [7], a distributed look-up table based approach is proposed to determine the cameras' viewing frustums that allows to select the best cameras for tracking purposes. In [2], Kansal *et al.* proposed an optimization process for the determination of the network configuration that maximizes a proper metric. It is interesting to note how the adopted metrics are concretely bound to real sensors thus propose a feasible instrument for real applications. More recently, Mittal and Davis [5,6] introduced a method for determining good sensor configurations that would maximize performance measures for a better system performance. In particular, the authors based the configuration on the presence of random occluding objects and proposed two techniques to analyze the visibility of the objects. Qureshi and Terzopoulos [8] proposed a proactive control of multiple PTZ cameras through a solution that plans assignment and handoff. In particular, the authors cast the problem of controlling multiple cameras as a *multibody* planning problem in which a central planner controls the actions of multiple physical agents. In the context of person tracking, their approach computes the relevance of a PTZ camera to an observation task by considering five factors: a) camera-pedestrian distance, b) frontal viewing direction, c) PTZ limits , d) observational range and e) handoff success probability. The planing is then achieved by employing a greedy best-first search to find the optimal sequence of states.

A different approach to network reconfiguration for person tracking by means of PTZ camera can be developed by employing the game theory. Arslan *et al.* [1], demonstrate the Nash equilibrium for the strategies lies in probability distribution. From this formulation, different approaches [4, 9] proposing a set of utility functions solve the camera assignment problem by maximizing a global utility function. Dif-

ICDSC 2010 August 31 – September 4, 2010, Atlanta, GA, USA

ferent mechanisms to compute the utilities can be provided as in [4, 9, 10], then a bargaining process is executed on the predictions of person utilities at each step. The cameras with the highest probabilities are used to track the target thus providing a solution to the handoff problem in a video network. On the other hand, when a PTZ camera is reconfigured to track an object or switched on/off to save power the topology of the network is modified. As consequence, a new configuration is required to provide optimal coverage of the monitored environment. Song *et al.* [9] adopt a uniform distribution of the targets and the coverage resolution utility to negotiate the new network reconfiguration.

The aim of this paper is to propose a PTZ camera network optimal reconfiguration technique. The pan, tilt and zoom parameters of each camera are automatically modified in order to maximize the environment coverage according to a predefined activity map describing the presence of moving objects over the observed area: the idea is that the cameras should focus on the zones of highest activity, possibly leaving out the least used zones if no full coverage is possible. The camera coverages could be limited by the presence of environmental occlusions, which are explicitly handled by the proposed method. The method is based on a slightly modified version of the popular Expectation-Maximization (EM) algorithm, applied in a proper space where the constraints due to the camera positions become null.

The paper is structured as follows: in section 2 the basics of the Expectation-Maximization algorithm are briefly given. The main idea of performing the reconfiguration of a network of sensors is described in section 3, in which we discuss both which are the optimality criteria to be satisfied and how EM could be used to perform this task. The details of the proposed algorithm are given in section 4, where a variation of the EM algorithm is presented in order to compute only feasible solutions. Finally, section 5 presents some experimental results performed on both synthetic and real data.

2. EXPECTATION MAXIMIZATION

In order to understand the rest of the proposed technique, it is useful to briefly recall here the basic principles of Expectation Maximization. Expectation Maximization (EM) is a popular tool for data fitting: given a family of probability density functions (PDF), it searches for the function f that best describes the distribution of a given data set $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$. We will here consider the special case of the mixture of isotropic Gaussian functions, where the PDF are defined as

$$f(\boldsymbol{x}; \boldsymbol{\Phi}) = \sum_{k=1}^{K} G(\boldsymbol{x}, \boldsymbol{\mu}_k, \sigma_k) c_k \tag{1}$$

and G is an isotropic, multivariate (D-dimensional) Gaussian function[1]:

$$G(\boldsymbol{x}, \boldsymbol{\mu}_k, \sigma_k) = \frac{1}{(\sqrt{2\pi}\sigma_k)^D} e^{-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}_k\|^2}{2\sigma_k^2}} \tag{2}$$

The vector $\boldsymbol{\Phi}$ contains all the parameters needed to uniquely define a PDF, this is the weights $c_1, \ldots, c_K$, the means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ and the variances $\sigma_1, \ldots, \sigma_K$.

[1]The focus on isotropic functions will be justified in the next section.

The "best" PDF is defined by the set of parameters $\hat{\boldsymbol{\Phi}}$ maximizing the likelihood function Λ, defined as:

$$\Lambda(X, \boldsymbol{\Phi}) = \prod_{n=1}^{N} f(\boldsymbol{x}_n; \boldsymbol{\Phi}) \tag{3}$$

The optimal parameters $\hat{\boldsymbol{\Phi}}$ are thus defined as:

$$\begin{aligned} \hat{\boldsymbol{\Phi}} &= \operatorname*{argmax}_{\boldsymbol{\Phi}} \Lambda(X, \boldsymbol{\Phi}) \\ &= \operatorname*{argmax}_{\boldsymbol{\Phi}} \ln \Lambda(X, \boldsymbol{\Phi}) \\ &= \operatorname*{argmax}_{\boldsymbol{\Phi}} \sum_{n=1}^{N} \ln f(\boldsymbol{x}_n; \boldsymbol{\Phi}) \\ &= \operatorname*{argmax}_{\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{c}} \sum_{n=1}^{N} \ln \sum_{k=1}^{K} G(\boldsymbol{x}, \boldsymbol{\mu}_k, \sigma_k) c_k \end{aligned} \tag{4}$$

A solution for the optimization problem (4) can be computed by setting to zero the partial derivatives of $\ln \Lambda(X, \boldsymbol{\Phi})$ with respect to $\boldsymbol{\mu}, \sigma$ and c respectively. This leads to the equations

$$\begin{aligned} \boldsymbol{\mu}_k &= \frac{\sum_{n=1}^{N} \boldsymbol{x}_n p(k|n)}{\sum_{n=1}^{N} p(k|n)} \\ \sigma_k^2 &= \frac{\sum_{n=1}^{N} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 p(k|n)}{D \sum_{n=1}^{N} p(k|n)} \\ c_k &= \frac{1}{N} \sum_{n=1}^{N} p(k|n) \end{aligned} \tag{5}$$

for each Gaussian distribution k, where the term $p(k|n)$ is the probability that, given a data point $\boldsymbol{x}_n$, it has been drawn from the distribution k, and it is defined as

$$p(k|n) = \frac{G(\boldsymbol{x}_n, \boldsymbol{\mu}_k, \sigma_k) c_k}{\sum_{z=1}^{K} G(\boldsymbol{x}_n, \boldsymbol{\mu}_z, \sigma_z) c_z} \tag{6}$$

The equations (5) and (6) cannot be easily solved since they are mutually dependent. The Expectation-Maximization algorithm solves the problem by starting from an arbitrary set of values for $\boldsymbol{\mu}, \sigma$ and c for each Gaussian, and iteratively alternating the computation of (6) (Expectation step) and (5) (Maximization step).

3. DEFINING THE OPTIMAL CONFIGURATION

In order to search for the optimal configuration for the pan, tilt and zoom parameters of the cameras, an optimality criterion must be defined. In this work, we have chosen to work with *relevance maps*, 2-dimensional discrete functions in the form of $m : \mathbb{N}^2 \mapsto \mathbb{R}$ representing a relevance value for each point of a discrete map of the scene. For example, figure 1(a) shows a relevance map based on moving object activity. In this case, several cameras have been observing the area of a parking lot, and moving objects have been detected with standard change detection algorithms. The position of each moving object has been projected on a discretized version of the scene map using homographic transformations, and each "cell" of the map is incremented whenever a moving object crosses that region. The map thus associates to each observed region its usage in terms of passing objects. Even though the experimental section of this

paper will be focused on tests performed on these activity maps, the proposed optimization technique is independent from the type of relevance maps used: the maps could even be computed from different types of sensors such as acoustic ones.

If there are no occlusions in the camera views, a single relevance map would be sufficient. However, the presence of static occlusions lead to different relevance maps for each camera, since the areas behind the occlusion cannot be observed and thus should have a relevance of 0 (see figure 1). Handling occlusions is a fundamental step in a sensor reconfiguration system, since the occluded zone not visible from a given camera could be covered by other sensors in the network.

Under the assumption of a planar scene, the intersection of a camera cone of view with the ground plane is an ellipse (assuming that the camera is not looking above the horizon). The problem of finding an optimal camera configuration can thus be reduced to a data fitting problem, in which a set of ellipses (one for each camera) matches the relevance maps in order to maximize their coverage. This can be obtained by applying the Expectation-Maximization algorithm, prior a minor modification to the basic algorithm in order to support different relevance weights for each point (equations in the next section). Switching from the Gaussians computed by EM to the required ellipses is straightforward, since the isoprobability curves of a generic bivariate Gaussian distribution are ellipses, and each ellipse corresponds to a specific quantile of the distribution. For example, given a Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix Σ, the isoprobability ellipse with equation $\sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} = 2$ corresponds to the 95% quantile: in other words, by interpreting that ellipse as a camera coverage, we ensure that the region observed by the camera will contain the data associated to that specific Gaussian function with a probability of 0.95 (these will be the values used in the experimental section).

However, only a small subset of all the possible ellipses can really be interpreted as an intersection of a cone of view with the ground plane, since their main axis must be oriented toward the camera, as clearly shown by figure 2. The vast majority of the solutions found by a direct application of EM would actually be unfeasible. A simple yet effective solution to this problem is proposed in the next section.

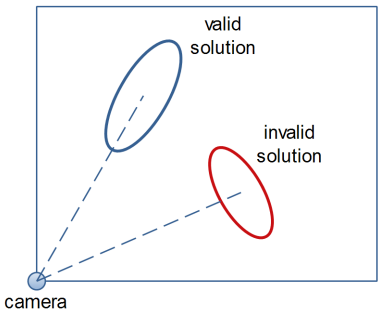


Figure 2: Top view of a camera and its observed elliptical region. No configurations of the pan, tilt and zoom parameters could lead to an observed region shaped as the red ellipse.

4. FINDING ONLY FEASIBLE SOLUTIONS

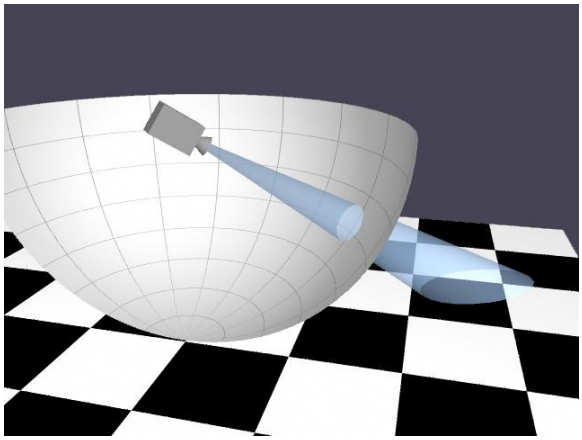


Figure 3: The pan-tilt space of a camera can be represented as a sphere surrounding the camera. Observe that ellipses in the ground plane become circles in the pan-tilt space. Moreover, any circle in the pan-tilt space can be associated to a valid PTZ configuration, a property that does not hold for generic ellipses in on the ground plane.

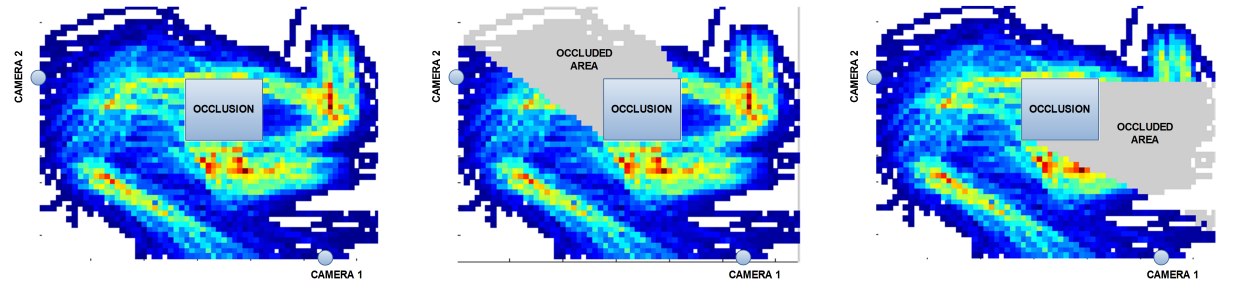
As discussed in the previous section, EM cannot be directly applied on relevance maps, since it could possibly compute solutions that cannot be interpreted as the observed regions for a set of cameras. We propose to solve this problem by performing the computations in a new space where any possible solution is acceptable. Figure 3 gives an intuitive interpretation of the proposed method: any valid ellipse on the ground plane is projected into a circle on the surface of a sphere centered on the camera. If the data fitting process is performed in the spherical space, the problem becomes unconstrained, since any circle corresponds to a valid pan, tilt and zoom configuration of the camera: the pan and tilt parameters will define the position of the circle, while the zoom parameter will define its radius. The proposed method thus relies on applying EM in this new space, which essentially implies a switch to spherical coordinates. In particular, for any point (x, y) on the ground plane and a given camera placed at coordinates (X_c, Y_c, Z_c), we compute the pan (ϕ) and tilt (θ) coordinates of the point as:

$$\begin{cases} \phi = \arctan\left(\frac{y-Y_c}{x-X_c}\right) \\ \theta = \arctan\left(\frac{\sqrt{(x-X_c)^2+(y-Y_c)^2}}{Z_c}\right) \end{cases} \tag{7}$$

A further coordinate system change is applied by moving to polar coordinates:

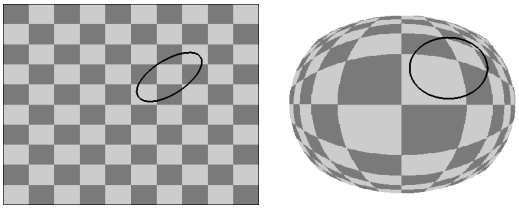
$$\begin{cases} u = \theta\cos\phi \\ v = \theta\sin\phi \end{cases} \tag{8}$$

Ellipses in the ground plane correspond to circles in the newly defined uv-space, as shown in figure 4. Performing the computation of EM in the uv-space thus will always lead to feasible solutions if the algorithm is forced to find only circular-shaped solutions. This can be obtained by using the equations for isotropic Gaussians, as done in equations (5) and (6). Note that each camera will have its own uv-space,



(a) (b) (c)

Figure 1: Relevance maps based on motion activity. (a) the global relevance map; (b) as seen from camera 1; (c) as seen from camera 2. Map colors represent the degree of detected activity (blue=low activity, red=high activity).



(a) (b)

Figure 4: Ellipses in the ground plane are mapped to circles in the uv-space. (a) the ground plane; (b) the uv-space.

but this does not influence the solving equations, since each Gaussian "interacts" with the other ones only in the Expectation step (equation (6)), and in that equation the only requirement is the possibility of computing the probability of a given point for each Gaussian, a computation that can be performed in a separate uv-space for each camera.

Finally, the original EM equations do not consider the possibility of having different weights for each point (the values of the relevance maps), but the extension is straightforward. Moreover, if different sets of weights are used for each camera, it is also possible to handle occlusions, since it will be sufficient to set the weight to zero for all those point not directly visible from the camera.

The proposed algorithm can thus be defined in this way: for each point $\boldsymbol{x}_n$ with coordinates (x_n, y_n) on the ground plane, compute its projection $\hat{\boldsymbol{x}}_{nk}$ on the uv-space of each camera k centered in (X_k, Y_k, Z_k):

$$\phi_{nk} = \arctan\left(\frac{y_n - Y_k}{x_n - X_k}\right)$$

$$\theta_{nk} = \arctan\left(\frac{\sqrt{(x_n - X_k)^2 + (y_n - Y_k)^2}}{Z_k}\right)$$

$$\hat{\boldsymbol{x}}_{nk} = [\theta_{nk}\cos\phi_{nk}, \theta_{nk}\sin\phi_{nk}]$$

Then, find an EM solution by iteratively applying the following steps:

- Expectation step:

$$p(k|n) = \frac{w_{nk} G(\hat{\boldsymbol{x}}_{nk}, \boldsymbol{\mu}_k, \sigma_k) c_k}{\sum_{z=1}^{K} w_{nz} G(\hat{\boldsymbol{x}}_{nz}, \boldsymbol{\mu}_z, \sigma_z) c_z} \tag{9}$$

- Maximization step:

$$\begin{aligned} \boldsymbol{\mu}_k &= \frac{\sum_{n=1}^{N} w_{nk}\ p(k|n)\ \hat{\boldsymbol{x}}_{nk}}{\sum_{n=1}^{N} w_{nk}\ p(k|n)} \\ \sigma_k^2 &= \frac{\sum_{n=1}^{N} w_{nk}\ p(k|n)\ \|\hat{\boldsymbol{x}}_{nk} - \boldsymbol{\mu}_k\|^2}{2\sum_{n=1}^{N} w_{nk}\ p(k|n)} \\ c_k &= \frac{\sum_{n=1}^{N} w_{nk}\ p(k|n)}{\sum_{n=1}^{N} w_{nk}} \end{aligned} \tag{10}$$

where w_{nk} is the weight (relevance) of point $\boldsymbol{x}_n$ taken from the relevance map associated to camera k.

Once the algorithm has converged to a solution, the pan and tilt values for each camera k can be found by computing the inverse polar transformation for the point $\boldsymbol{\mu}_k$:

$$\begin{cases} \phi_k = \arctan\left(\frac{\boldsymbol{\mu}_k(2)}{\boldsymbol{\mu}_k(1)}\right) \\ \theta_k = \sqrt{\boldsymbol{\mu}_k(1)^2 + \boldsymbol{\mu}_k(2)^2} \end{cases} \tag{11}$$

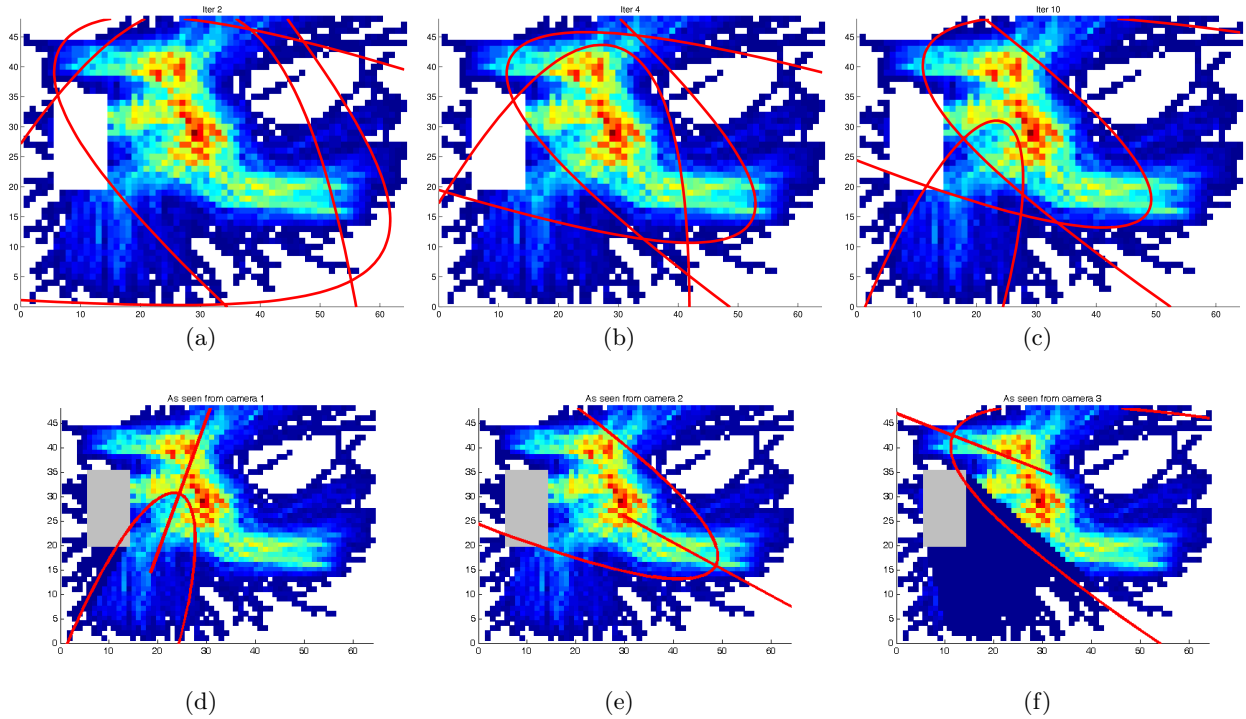
while the zoom level will be proportional to σ_k.

5. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed algorithm, we first tested it on synthetic data and then on real measurements coming from a surveillance system. We used a Matlab trajectory generator we previously developed for a trajectory analysis work and which is publicly available[2]. The trajectory generator creates a set of random trajectories grouped in a variable number of clusters, plus some outliers. The trajectory map is then discretized into a 48×64 grid and for each grid cell we count how many trajectories cross it; this way a relevance map is created. Once the map is created, a rectangular-shaped occlusion of limited size is randomly put on the map (figure 1(a)). The number and position of the cameras is again randomly chosen (always on the map borders) and the new relevance maps for each cameras are computed by identifying the occluded areas (figures 1(b) and 1(c)).

Figures 5(a)–(c) show the computed results after 2, 4 and 10 iterations respectively in the case of three cameras and an occlusion; after the 10th iteration the algorithms has converged to a stable solution. Figures 5(d)–(f) show the final result as seen from each one of the three cameras. Observe

[2] `http://avires.dimi.uniud.it/papers/trclust/`



(a) (b) (c)

(d) (e) (f)

Figure 5: Network reconfiguration for optimal relevance map coverage. (a)–(c) results after 2, 4 and 10 iteration respectively; (d)–(f) the relevance maps and the observed regions for camera 1, 2 and 3 respectively.

how camera 3 focuses only on its visible region, while discarding all the zones behind the occlusion; however, the hidden region is covered by camera 1, from which the zone is fully visible.

In order to measure the quality of the data coverage, we define a score s_n for each data point $\boldsymbol{x}_n$ defined as

$$s_n = \sum_{k=1}^{K} w_{nk}\ G(\hat{\boldsymbol{x}}_{nk}, \boldsymbol{\mu}_k, \sigma_k) \tag{12}$$

This way, the score reflects that the Gaussians should be centered on the most relevant points (with high weights w_{nk}); moreover the score is higher if high-relevance points are covered by more than one Gaussian. A global configuration score can then be defined as

$$\frac{1}{N}\sum_{n=1}^{N} s_n \tag{13}$$

This score allows to monitor the iterative EM process, thus making possible to check if the algorithm really converges to a better solution than the initial one. Figure 6 shows the score values at each iteration for the experiment shown in figure 5. As it can be seen, the score has increased up to convergence, meaning that the algorithm has really found a better solution at each iteration step. This behavior has been observed in all the performed tests.

Even though the score can measure the performance improvement as the algorithm iterates, it does not have an immediate and intuitive meaning, thus making it not suitable for batch results. we thus defined another performance measure which has a more intuitive interpretation, the coverage c defined as:

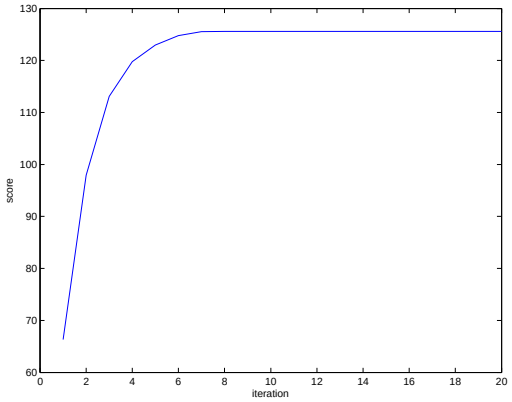


Figure 6: The overall score at different iteration steps for the experiment shown in figure 5.

$$c = \frac{\sum_{n\in E} \max_k(w_{nk})}{\sum_{n=1}^{N} \max_k(w_{nk})} \tag{14}$$

where E is the set of all the points falling within at least one ellipse. The value c is thus a measure of the coverage of the data points, ranging from 0 to 1 (total coverage) and giving more importance to the points with high relevance. This performance metric gives an intuitive meaning to the results shown in table 1. Here, 50 tests have been performed, each one with a random number and position of cameras and with random relevance maps with occlusions.

The test data are publicly available in order to encourage future comparative results[3]. As it can be seen, the proposed method seems to give good results in terms of coverage, with a minimum coverage of 0.9333 and a maximum coverage of 1, with a mean of 0.9757 and a standard deviation of 0.0157. Observe that the score values are not easily comparable between different tests, since they depend also on the number of Gaussians (cameras), which is randomly chosen for each test. The scores are here reported only for comparison with other techniques applied on the same data sets.

test	score	coverage	test	score	coverage
1	66.7272	0.9851	26	149.7472	0.9773
2	19.0007	1.0000	27	34.8725	0.9920
3	192.4728	0.9799	28	24.9652	0.9695
4	19.9578	0.9958	29	122.9014	0.9670
5	49.7638	0.9840	30	126.9778	0.9632
6	67.3638	0.9800	31	75.1907	0.9833
7	143.3025	0.9983	32	184.6799	0.9584
8	72.9799	0.9973	33	151.1612	0.9905
9	126.6761	0.9868	34	34.8747	0.9512
10	233.0987	0.9654	35	33.6040	0.9870
11	137.9270	0.9830	36	58.9277	0.9925
12	78.4417	0.9798	37	256.8763	0.9630
13	41.7085	0.9746	38	58.1713	0.9778
14	71.5236	0.9757	39	144.4666	0.9726
15	41.4097	0.9758	40	39.0397	0.9333
16	120.6578	0.9678	41	72.6748	0.9588
17	26.5087	0.9986	42	99.7195	0.9596
18	35.2319	0.9792	43	50.1147	0.9839
19	26.6857	0.9909	44	109.4096	0.9883
20	128.2982	0.9436	45	33.1720	0.9639
21	189.8183	0.9760	46	116.3147	0.9597
22	17.9085	0.9399	47	163.1278	0.9887
23	188.4652	0.9624	48	69.1296	0.9778
24	155.6098	0.9652	49	67.8829	0.9992
25	191.8158	0.9610	50	194.3093	0.9814

Table 1: Test results over 50 random data sets. For each data set (available online) the score and coverage metrics are given.

Finally, an example taken from real world is shown in figure 7. In this case, the parking lot in front of our university building has been monitored using wide field-of-view cameras for 4 hours, during which the moving objects have been detected and tracked (figure 7(a)). This has led to the motion activity map shown in figure 7(b). The proposed algorithm has been applied in order to reconfigure two PTZ cameras mounted on the roof of the building. The two cameras can span full 360ř in pan and 0ř-120ř in tilt direction, and thus can achieve any feasible solution computed by the system. The minimum and maximum zoom levels instead imposed a lower and an upper bound to the computed values of σ^2 (see eq. 10). The final result is shown again in figure 7(b); the final coverage is 0.9681.

6. CONCLUSIONS

In this paper we have presented a method for achieving an automatic and optimal reconfiguration of a network of PTZ cameras. The reconfiguration consists in finding the pan, tilt and zoom parameter for each camera that lead to an optimal coverage of relevance maps. Relevance maps represent the points visible from each camera and their importance (also considering occlusions); the maps used in this work are based on the number of detected moving objects in each zone across the observed scene, however other relevance maps could be used. Reconfiguration is achieved by using a variation of the Expectation-Maximization algorithm, where all the computations are performed in a proper data space where any solution found is automatically feasible, in the sense that it can be expressed as a pan, tilt and zoom configuration for a given camera. Experimental results give a performance measurement of the proposed technique.

[3]available in Matlab format at `http://avires.dimi.uniud.it/papers/icdsc10/icdsc2010_dataset.zip`

Acknowledgment

This work is partially supported by the Interreg IV Italy-Austria project n. 4697 “SRSNet - Intelligent Audio/Video Sensor Networks”.

7. REFERENCES

[1] G. Arslan, J. Marden, and J. Shamma. Autonomous vehicle-target assignment: A game-theoretical formulation. *ASME Journal of Dynamic Systems, Measurement and Control*, 129(5):584–596, 2007.

[2] A. Kansal, W. Kaiser, G. Pottie, M. Srivastava, and G. Sukhatme. Reconfiguration methods for mobile sensor networks. *ACM Transaction on Sensor Networks*, 3(4):22, 2007.

[3] D. Karuppiah, R. Grupen, A. Hanson, and E. Riseman. Smart resource reconfiguration by exploiting dynamics in perceptual tasks. In *IEEE International Conference on Intelligent Robots and Systems*, 2005.

[4] Y. Li and B. Bhanu. Utility-based dynamic camera assignment and hand-off in a video network. In *IEEE/ACM International Conference on Distributed Smart Cameras*, pages 1–9, Stanford, USA, 7-11 Sep. 2008.

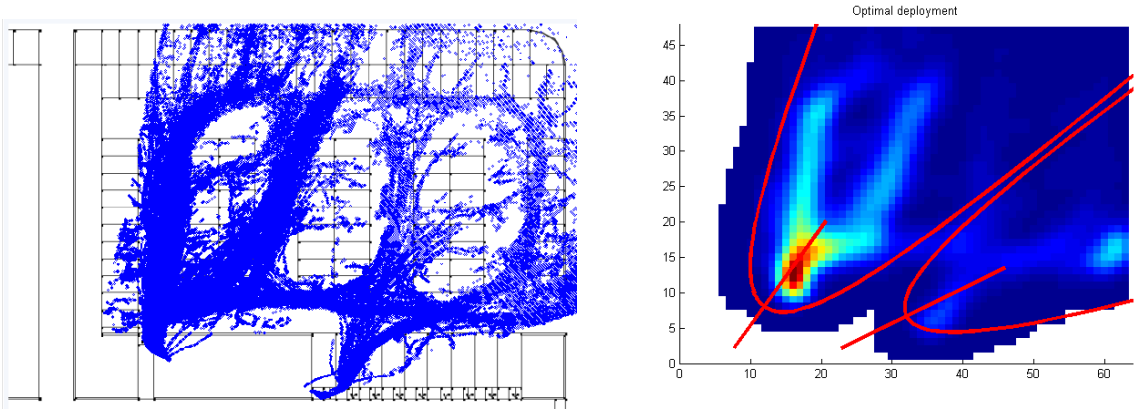
[5] A. Mittal and L. S. Davis. Visibility analysis and sensor planning in dynamic environments. In *European Conference on Computer Vision*, Prague, CZ, May 2004.

[6] A. Mittal and L. S. Davis. A general method for sensor planning in multi-sensor systems: extension to random occlusion. *International Journal of Computing Vision*, 76:31–52, 2008.

[7] 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.

[8] F.Z. Qureshi and D. Terzopoulos. Planning ahead for ptz camera ssignment and handoff. In *International conference on Distributed Smart Cameras*, pages 1–8, Como, Italy, Aug-Sep 2009.

[9] Bi Song, C. Soto, A. K. Roy-Chowdhury, and J.A. Farrell. Decentralized camera network control using game theory. In *IEEE/ACM International Conference on Distributed Smart Cameras*, pages 1–8, Stanford, USA, 7-11 Sep. 2008.



(a) (b)

Figure 7: A real-world camera reconfiguration example. (a) the detected presence of moving objects. For each detected moving object, a marker has been drawn on the map at regular time intervals (no information on the activity density is given here); (b) the corresponding relevance map (in red, the zones with higher activity density) and the final camera configuration.

[10] C. Soto, B. Song, and A. Roy-Chowdhury. Distributed multi-target tracking in a self-configuring camera network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1486–1493, Miami, USA, 20-25 Jun 2009.