Auto-Organized Visual Perception Using Distributed Camera Network

Richard Chang, Sio-Hoi Ieng, Ryad Benosman, Loïc Lachèze, Thibaud Debaecker

Auto-Organized Visual Perception Using Distributed Camera Network

Richard Chang, Sio-Hoi Ieng, Ryad Benosman, Loic Lacheze, and Thibaud Debaecker

Institut des Systemes Intelligents et de Robotique
Universite Pierre et Marie Curie, Paris6
4 Place Jussieu 75252 Paris,France
richard.chang@isir.fr

Abstract. Camera networks are complex vision systems difficult to control if the number of sensors is getting higher. With classic approaches, each camera has to be calibrated and synchronized individually. These tasks are often troublesome because of spatial constraints, and mostly due to the amount of information that need to be processed. Cameras generally observe overlapping areas, leading to redundant information that are then acquired, transmitted, stored and then processed. We propose in this paper a method to segment, cluster and codify images acquired by cameras of a network. The images are decomposed sequentially into layers where redundant information are discarded. Without need of any calibration operation, each sensor contributes to build a global representation of the entire network environment. The information sent by the network is then represented by a reduced and compact amount of data using a codification process. This framework allows structures to be retrieved and also the topology of the network. It can also provide the localization and trajectories of mobile objects. Experiments will present practical results in the case of a network containing 20 cameras observing a common scene.

1 Introduction

As cameras are becoming common in public areas they are a powerful information source. Camera networks have been intensively used in tracking or surveillance tasks [1, 2]. Most multi-camera systems assume that the calibration and the pose of the cameras are known, standard networks applications also imply other highly constraining tasks such as : 3D reconstruction, frames synchronization, etc... Baker and Aloimonos [3], Han and Kanade [4] introduced pioneering approaches of calibration and 3D reconstruction from multiple views. The reader may refer to [5–7] for interesting works on camera networks.Most of applications implying the use of a set of cameras are processing information by incrementing acquired data. Every single camera acts as an individual entity that does not necessarily interact with the other ones. Usually the camera transfers its information regardless to the behavior of the other ones. Thus, if the network

is dense enough, obvious redundancies are unavoidable and resources like bandwidth, mass storage system are simply wasted. One can expect to overcome these problems by coordinating smartly the efforts of each camera relying on the main idea that they are forming a unique vision sensor. Data compression methods preserving relevant information should then be used. Scenes can be described using their contents relying on lines and edges to build geometric models from images [8]. In other cases, visual features can be merged with other modalities such as ultrasound sensors [9] to introduce robustness. Several aspects of the environment can also be extracted from images like walls, doors and vacant spaces [10]. Recent works on bag-of-features [11] representations have become popular, as they introduce geometry free features to characterize local subimage using statistical tools.
The aim of this paper is to introduce a geometry-free method that allows camera networks systems to estimate their topology and auto-organize their own activities according to the content of the scene and the task to be achieved. The estimation of the topology is retrieved using statistical approach as in [12] but without any correspondence between the images.

The paper introduces a common description visual language used by all cameras to exchange information about scenes. A sampling method of acquired images into subimages combined with bag-of-feature allowing their codification is presented. In a second stage, a multilayer data reduction architecture is introduced, it is inspired by the statistical organization of the human retina [13]. This convergent structure as will be seen allows to remove redundancies. Finally a functional layer gathers cameras as single visual entities performing identified tasks.
This paper is organized as follows : in the next section, the multi-layer coding is presented. Each transition from the lowest stage to the higher one is detailed. In the third section we show that geometric structures can be recovered from such coded camera network : scene object localization can be estimated up to some metric properties. In the last section, experiments are tested on real images and results are provided.

2 Multi-layer image coding

Camera networks are usually represented by a concatenation of single cameras. The cameras act individually and does not interact with the others which leads to resources' wastage as computational load. We propose in this section a hierarchical representation where each layer encodes information about the preceding one. The network is then seen as a combination of items which represent an information provided by the cameras. Figure. 1 summarizes the whole codification process.

To allow an easier handling of the camera network and the location of cameras, a planar topology of the network is introduced. As shown in Figure. 2 the 3D locations of cameras are orthographically projected onto a plane ν^0 set as the first layer.

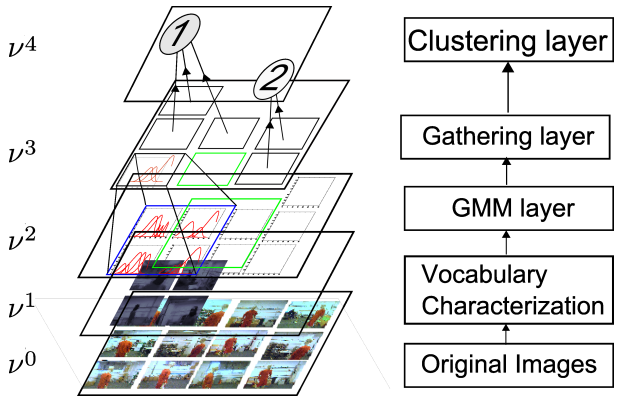


Fig. 1. General overview of the method: (1) layer ν^0 Original images, (2) layer ν^1 Images coded in patches, (3) GMM Decomposition of the codified images, (4) Extraction of the main histograms (5) The network is divided into clusters according to information data.

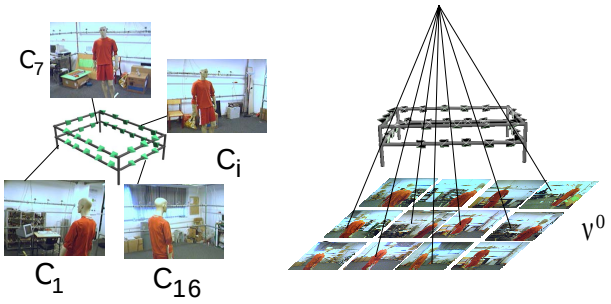


Fig. 2. Orthographic projection of the cameras location in 3D onto a plane representing the first layer ν^0.

In what follows ν^j is a plane at level j and ν_i^j will represent its i^{th} element.

2.1 From acquired images to codified images (ν^0 to ν^1)

The goal of this section is to sample acquired images into representative patches. Each patch as will be seen will be compared to a codebook, and a codified image is produced. It is important to notice that the codebook is the same for all cameras, allowing further comparisons.

Decomposition of images An efficient decomposition must produce a possibly unique partitioning of images. In addition it would be interesting to produce less patches, but of variable size so that they can cover homogeneous texture zones.

In order to achieve the generation of patches, a quadtree-like algorithm is set up. The quadtree algorithm cuts recursively images into subimages. Starting

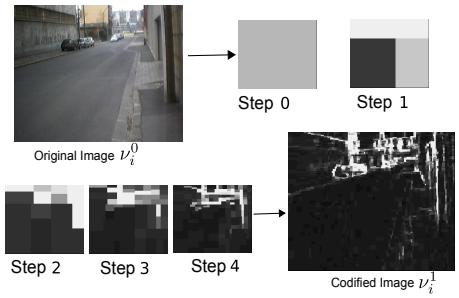


Fig. 3. Example of codification of a scene using the optimal entropic method. The codebook has a maximal size of 32 patches.

from the initial image, each subimage is cut into four equal subimages. The idea is to use the same principle, but at the contrary of the regular quadtree approach, the division of subimage will be driven by an entropy measure. The idea is to cut a subimage at the location were the difference of the quantity of information between possible subimages is minimal. This quantity of information is given for an subimage m by :

$$H(m) = -\sum_{c=0}^{c=255} P(m=c)\log P(m=c) \tag{1}$$

with $P(m=c)$ the number of times the pixel value c appears in m, $P(c)$ is the probability of appearance of the grey value c within m.

An illustration of the algorithm is given in Figure. 3. It appears clearly that the image is decomposed more coherently, a complete overview of the method can be found in [11].

Characterizing texture In order to make the comparison with the codebook, each patch of the image has to be characterize according its texture. Texture can be measured using different approaches. In what follows we choose to use a measure similar to [14]. It relies on the computation of a histogram of the difference between the value of pixels of images. Given a subimage m, each value of its histogram of differences h_m is given by :

$$h_m(i) = \sum_{x,y,x',y' \in m}^{x \neq x' \vee y \neq y'} \text{diff}(m,x,y,x',y',i),\ i \in [0,255] \tag{2}$$

with

$$\text{diff}(m,x,y,x',y',i) = \begin{cases} 1 & \text{if } |m(x,y) - m(x',y')| = i \\ 0 & \text{else} \end{cases}$$

In a second stage, the histogram h_m is normalized, to ensure an invariance according to the size of I.

Let $T = \{h_{z_0}, h_{z_1}, ...h_{z_n}\}$ be the set containing all texture descriptors of patches z_i of I. The idea is to sample T to reduce the number of descriptors to $m \leq n$. We then add to T a metric function expressed by $dist(h_{z_i}, h_{z_j})$ and a reference texture patch h_{ref}. The reference patch is set to a patch containing a single color, corresponding to a uniform area. In a second stage all the representation of patches contained in T are compared to h_{ref} and sorted, from the less to the more textured. The set T_s corresponding to the ordered set T becomes :

$$T_s = \{h_{ref}, h'_{z_0}, h'_{z_1}, ...h'_{z_n}\} \quad \text{with } dist(h_{ref}, h'_{z_i}) \leq dist(h_{ref}, h'_{z_j}) \text{ if } i < j \quad (3)$$

The mahalanobis distance is used as a metric function and is set so for the rest of the paper. At this point, T_s is then sampled into m areas. For each area, only the median patch is selected. The resulting selection gives the codebook V :

$$V = \{h_{ref}, h'_{z_0}, h'_{z_1}, ...h'_{z_m}\},\ V \subset T_s \quad (4)$$

that corresponds to the most representative patches. The whole codebook is computed offline from a subset of images of the sequence.

Let I_{acq} be an acquired image, I_{acq} is decomposed into z_{acq_i} patches. Each computed patch must be compared to the content of V.

In case a new patch is detected, it is added to the codebook as a new entry. The acquired image I_{acq} is then codified using the patches of the codebook, the resulting image I_{cod_i} given by a set of vocabulary patches.

2.2 From patches to GMM-histograms (ν^1 to ν^2)

Each element ν_i^1 represents an image ν_i^0 coded into patches using the common vocabulary. It is then possible to express the statistical content of ν_i^1 using an histogram giving the distribution of patches within ν_i^1. The size of the codebook is set to 32 elementary words, and can be adjusted according to the complexity of scenes. To lower the data load, histograms are then decomposed as a combination of gaussians using Gaussian Mixture Models (GMM) [15, 16]. This decomposition models a signal as a sum of normal distribution (ND). The content of an element ν_i^2 of the next layer ν^2 is the GMM decomposition of the histogram of the content of an image ν_i^1 of ν^1, it is defined as an histogram $H_{\nu_i^1}(x)$:

$$H_{\nu_i^1}(x) \simeq \sum_{n=1}^{Nbg} m_n N_{(\mu_n, \sigma_n)}(x) \quad (5)$$

where $N_{(\mu_n, \sigma_n)}(x)$ is the normal distribution whose standard deviation is σ and whose mean is μ.

In Eq. 5, m_n is the corresponding weight of the normal distribution n, and $H_{\nu_i^1}(x)$ is composed by Nbg distributions. It is obvious that $0 \leq \mu \leq 31$ due

to the size of the codebook that contains 32 words. One ND fits with one class of pixels existing in the neighborhood V. Therefore it is important to put together the similar classes (i.e., distribution with close means) and to keep aside distributions which are not representative (i.e., distributions whose weight are insignificant). Finally, the most representative NDs are sorted according to their weights.

2.3 From the GMM-histograms layer to the gathering layer (ν^2 to ν^3)

In order to lower the data load, redundant information stored in ν^2 must be merged. Elements of ν^2 are gathered according to spatially neighborhood areas defined by the orthographic projection (see Figure. 2). Each area contains a collection of ν_i^2, the common normal distributions are transmitted to ν^3 while the others are eliminated. An element ν_i^3 contains the common information of a set of ν_i^2, in what follows the elements are gathered according to windows of size 4×4. It is important that spatial gathering windows overlap, as eliminated minor information within a gathering window might be of major interest to a close one. Thus, an element ν_i^3 contains main information data computed from four cells ν_i^2, and allows a wide area coverage with reduced amount of information.

Let X and Y be two distributions of same size, Bhattacharyya proximity is introduced as

$$P_B(X,Y) = \sum_i \sqrt{X(i) \cdot Y(i)} \tag{6}$$

To illustrate the principle, four cameras are considered (C_1, C_2, C_3, C_4) observing a common scene (Fig. 4). The corresponding histogram is then given to the next layer as a main information. Layer ν^0 represents the original images. The codified images are shown in ν^1, while the GMM decomposition is computed in ν^2. Layer ν^3 contains the most common information data collected from all cameras.

2.4 From the gathering layer to the clustering layer (ν^3 to ν^4)

It is now important to gather the elements of ν^3 according to their content. Similar ν_i^3 must be merged into a single ν_i^4 corresponding to a set of cameras observing a scene or an object from different view points not necessarily close to each other. In order to represent efficiently information provided by the cameras, a clustering layer is set up. This layer deals with an agglomeration of different elements ν_i^3 according to the correlation of their values, spatial neighborhood has no effect on this process. Correlated cells of ν^3 will be clustered into a new cell ν_i^4 representing their content data. Thus, each element ν_i^4 represents an unique information about the scene. The correlation between two elements ν_i^3 and ν_j^3 is defined as:

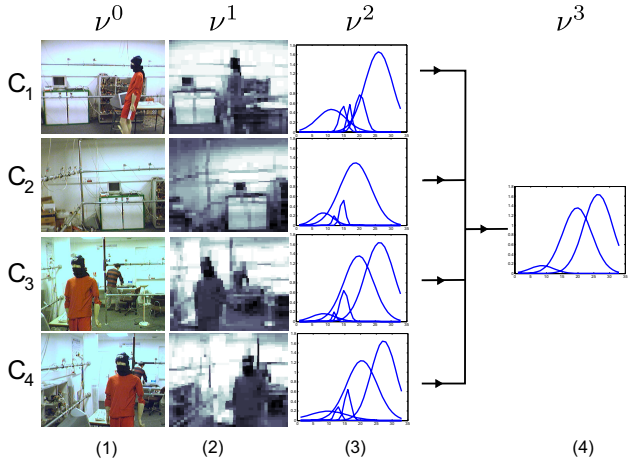


Fig. 4. Extraction of main visual features of four cameras of the network (1) Original image (2) Decomposition into patches, (3) GMM Decomposition (4) Gathering step, the main information is extracted.

$$Corr(\nu_i^3, \nu_j^3) = \frac{\sum(\nu_i^3 - \bar{\nu_i^3}) \cdot (\nu_j^3 - \bar{\nu_j^3})}{\sqrt{\sum(\nu_i^3 - \bar{\nu_i^3})^2} \cdot \sqrt{\sum(\nu_j^3 - \bar{\nu_j^3})^2}} \tag{7}$$

Layer ν^4 is then a clustering of the elements of ν^3 according to their content, the definition of an element ν_i^4 is then given by $\nu_i^4 = \left\{ \nu_j^3 \ / \ Corr(\nu_i^3, \nu_j^3) < \epsilon \right\}$ At this stage, the network is organized as a set of sorted information and not as a concatenation of single cameras. Redundant information have been gathered providing the main information extracted from cameras.

3 Structures retrieval

In the following section we will set the relative positions of the cameras as unknown. The hypothesis of calibrated camera is also a highly constraining condition, in what follows it is released. In case of dynamic networks, each camera can move and be active through time. If each camera is taken individually and assumed not calibrated, one cannot easily expect to be able to estimate its position, hence the global structure is not recoverable. On the other hand, if the contribution of each camera is combined with others as shown by the previous model, it becomes then possible to provide an estimation of the global topology with no need of a precise calibration of the network and the knowledge of the exact positions of cameras. We will show that it is possible to retrieve the global topology of the whole network using the lowest stages of the codification process.

3.1 Estimating network topology

Let $C = \{C_i\}\ i \in N$ be the set of N uncalibrated cameras. A camera C_i produces an image ν_i^0 that is coded by a common vocabulary to ν_i^1 (as shown in section 2). In this subsection, the network topology is estimated by analyzing the objects of the scene. The whole codification chain is not necessary, the process is carried out from segmented images ν_i^0 up to layer ν^1. Each image ν_i^1 is characterized by its histogram of patches $H_{\nu_i^1}$. A cross-correlation score $Corr(H_{\nu_i^1}, H_{\nu_j^1})$ is computed between two images coming from two cameras C_i and C_j (eq. 7). The correlation score depends on the viewpoint of the two cameras. The score will be high for close viewpoints, and low for two farther cameras.

The amount of details of objects in the scene increases as the distance between the camera and the objects decreases. In this case the entropy of the segmented images of layer ν^0 is a relevant measure to provide an estimation of this distance. By analyzing the entropy for a given position of the object, the distances to all the cameras can be estimated. The relative positions of the cameras are then determined from the distances. Given the acquired images, the object is segmented from the background. Then, the entropy Q_i of each segmented image (of ν^0) is computed.

The correlation and the entropy are computed for a video sequence acquired by the cameras. By combining these values between the cameras, spatial coherence can be determined. Cameras are then aggregated in order to satisfy the coherence and the correlation values. It is not necessary that all the cameras observe the same area. The method only requires an overlap between pairs of adjacent cameras to determine their correlation.

3.2 Localizing a new camera in the network

Once the global topology of the network known, the whole codification chain is processed. The network is then represented by different sets of cameras grouped according to their information content. The configuration of these sets are not necessarily the same as the spatial configuration.

Let C_p be a new camera viewing the same scene, its image ν_p^0 is then coded by the common vocabulary to ν_p^1. In order to compare the information given by C_p and the one of the network, the element ν_p^2 is computed. To localize C_p in the network, a top-bottom search is performed on the codification structure. At each layer ν^i of the structure, a correlation value (eq. 7) is computed with ν_p^2. The highest correlation score gives the closest camera or group of cameras closest to C_p.

3.3 Localizing scene objects

It is possible to provide an estimation of the position of objects in the scene according to the cameras. We assume in this section that the positions of the cameras are known. The goal is to determine the localization of the objects without any calibration method. The position of the object can then be set as

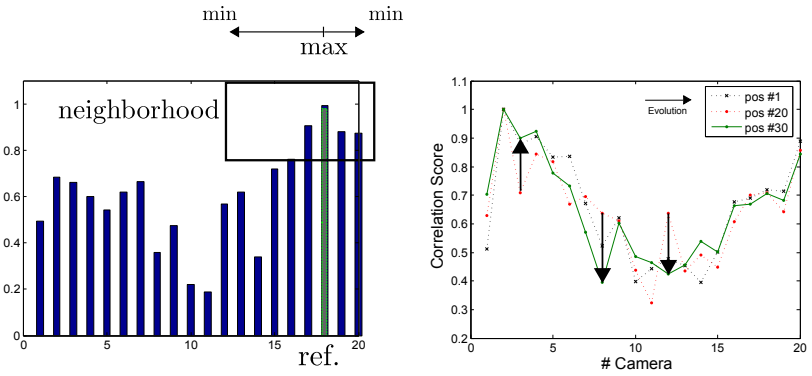
the linear combination of the positions of these cameras according to the values of the entropy computed :

$$P_i = \sum_{i}^{N} \alpha_i Pos(C_i) \tag{8}$$

where α_i is a decreasing function of the distance object to camera and the $Pos(C_i)$ is the position of camera C_i. As the cameras are uncalibrated, the function α_i cannot be determined precisely. The object is then localized up to this scale.

4 Experimentation

Experiments are carried out on a camera network containing 20 uncalibrated cameras regularly placed around the scene. They all acquire images at the frequency of 30Hz. The whole calibration process relies on image sequences taken by all cameras of a person moving freely and randomly inside the observed area. No assumptions are made on the metric or appearance of the walking person.



(a) (b)

Fig. 5. (a) Correlation score between the camera C_{18} and the other cameras. The score is similar for the cameras of the same side (closest cameras) and is very different for the others cameras. (b) Different correlation scores for three consecutive positions of the walking person. The score is related to the position of the walking person in the scene. The correlation score between the cameras depends on the position of the object in the scene. By analyzing different positions in the scene, the neighborhood can be retrieved from a more robust correlation score.

4.1 Topology estimation

The whole codification process is performed on all images provided by the network. The estimation of the topology of the camera network relies on the compu-

tation of two quantities from each camera : the correlation of its ν^2 codifications values with all other cameras, and the computation of its entropy and comparison with all others. The cross-correlation score can be computed for every pair of images taken by the camera network. The highest scores give a high probability for two cameras to be close. The entropy measure is computed to confirm the results given by the correlation score. Figure. 5(a) presents the mean correlation results between camera C_{18} and the rest of the cameras during the whole image sequence. The result is normalized with respect to the highest value corresponding to C_{18} correlated with itself. As expected, nearer cameras to C_{18} give the highest scores. Two cameras are set as 'neighbours' if their correlation is at least equal to 80% (set up using experimental measurements). The correlation value is computed by each camera for all the positions of the walking person inside the scene. The results are then averaged providing a mean value of all the scores for each camera. Figure. 5(b) shows the evolution of the correlation results for three different consecutive positions of the walking person.

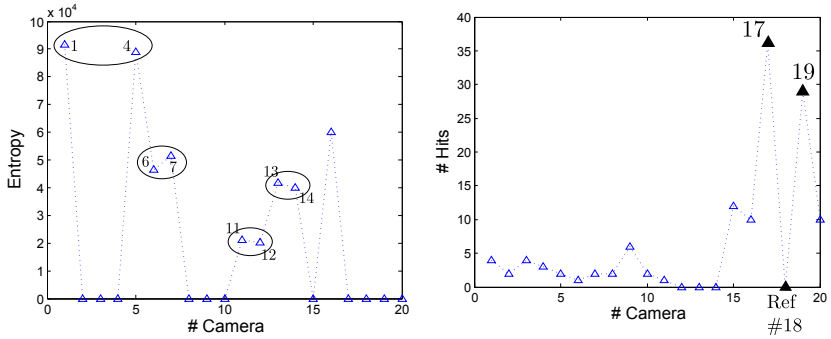


Fig. 6. (a) Entropy value for a given position of the walking person inside the scene. The similar values indicates the close cameras. (b) The hit graph of entropy values of camera 18, cameras 17 and 19 are again the closest cameras.

The computation of the entropy value of each segmented image at ν^0 of the sequence is also of great importance as it provides complementary information for establishing the topology. As explained in section 3.1, the distance from the camera to the object in the scene is also related to the value of the entropy. Figure. 6(a) shows the entropy computed for all the cameras at a given position of the walking person The entropy is set to zero for the cameras which do not see the walking person. Different groups of cameras can then be set: $1-4$, $6-7$, $11-12$, and $13-14$. One can notice that the similar values indicate that the distance camera-object is similar but the cameras are not necessarily neighbours as for $1-4$. By combining the results of the image sequence in which the man

is moving in the scene, close cameras will statistically in time produce the same entropies.

Entropy is computed for each camera at each frame, each camera stores the amount of times another camera reaches its level of information beyond a threshold, it is then considered as a hit. Figure. 6(b) presents the combined results for camera 18. The number of hits is the highest for the cameras 19 and 17 which are its actual neighbours. From the correlation scores and the entropy values, a graph representation of the neighborhoods of each camera can be built. This score is computed as a weighted sum between the two normalized values. Figure. 7 shows the groups of cameras marked as neighbours via this summation score. In case the value is low between a camera and the others, this camera is rejected out the structure.

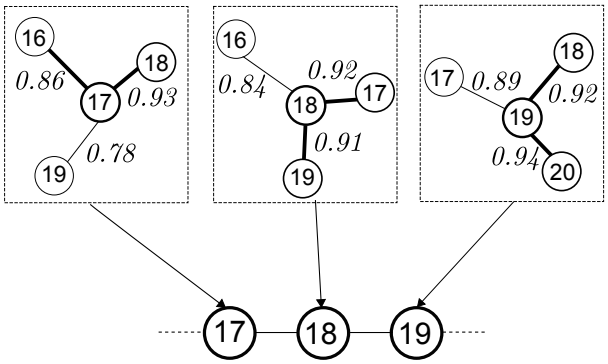


Fig. 7. The connection between the cameras can be retrieved using the correlation score and the entropy values. The neighbors cameras have been connected each other. The correlation values is shown above the connections. The bold connections show the results of the entropy analysis with the corresponding correlation values between the cameras.

Finally, using an iterative process the whole topology of the network can then be estimated. Figure. 8 shows the trajectory of the walking person inside the scene and the cameras layout. The method is not limited to grid topologies. The only constraint on the cameras is to have an overlap between two adjacent cameras.

4.2 Localizing a new camera in the network

Let C_p be a new camera added to the network at a location to be determined. The images provided by C_p are coded up to layer ν^2. The whole codification process is performed on all images provided by the network up to layer ν^4. The most representative information within the network at a certain time are expressed.

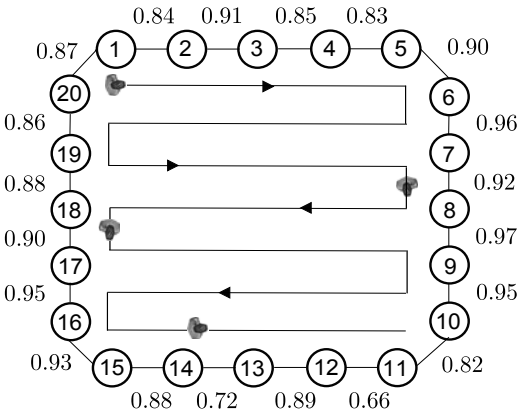


Fig. 8. Retrieved global topology of the network from cross-correlation and entropy. These values have been computed from the image sequences of the man following the shown trajectory.

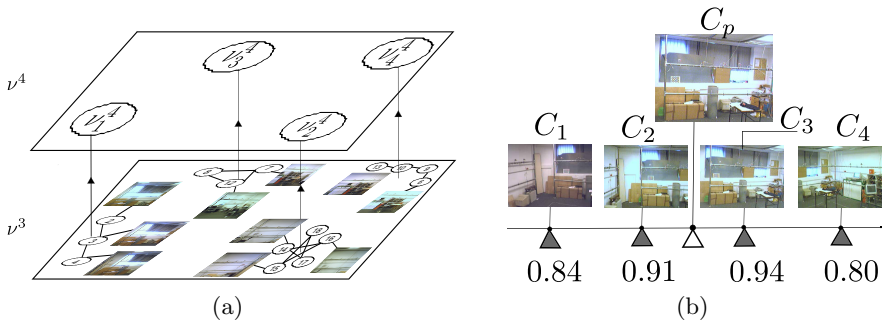


Fig. 9. (a) Layer ν^4 for the camera network, cameras expressing the same content are merged into a single node. (b) Cross-correlation value between the camera C_p and the other cameras C_1 to C_4 determined as the closest ones. C_p is then located between C_2 and C_3 according to the correlation.

As shown in figure.9(a) cameras expressing the same content are merged into a single node. The presented representation shows that the whole scene is expressed by four representative nodes. Given the new camera C_p, a cross-correlation value is computed between ν_p^2 and the different sets of nodes of ν^4. The highest correlated node in time includes the set of the closest camera to C_p. The same process is the recursively applied at the previous layers ν^3 and ν^2 following a top-bottom search model. The camera C_p is finally localized as neighbour of four elements of layer ν^2. Figure. 9(b) shows the correlation score between C_p and the four cameras ν_k^2 given by the closest element in ν^3 and their corresponding images. C_p is finally inserted at its corresponding location.

Instead of comparing C_p with all the cameras of the network, this top-bottom process acts as a graph analysis to find the closest elements to C_p. The computational load is reduced significantly. Time remains an important factor of the process. This architecture introduces a simplified and efficient camera management and eases the control of dynamic camera network.

4.3 Trajectory estimation

The topology and the position of the cameras are now assumed being known. The data reduction and the clustering from raw images to top levels are performed as explained in section 2.4.

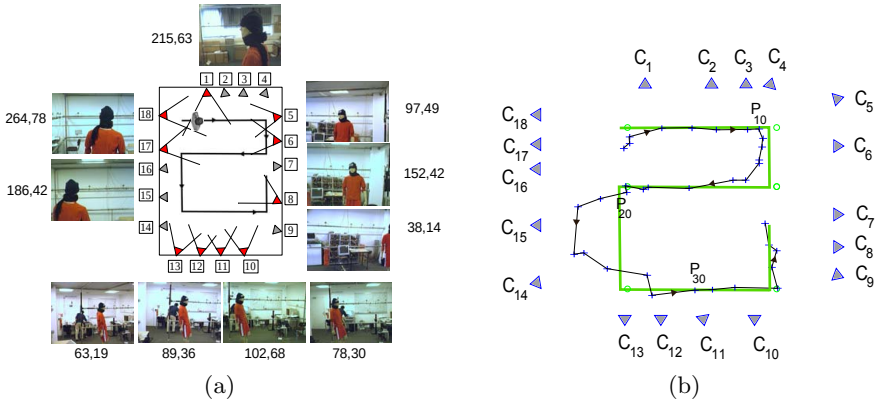


Fig. 10. (a) Quantity of information computed from the cameras for a given position of the man. These values are high when the man is close to camera and decrease according to the distance to the cameras. (b) The trajectory of the man is retrieved using the quantity of information given by all the cameras. The ground truth is drawn in green.

With the clustering technique, each camera contributes to provide a representation of the global perception of the entire network. We are also able to estimate trajectories. The positions of the walking person are estimated as a linear combination of the active cameras position as previously presented, with the α_i set proportionally to the entropy Q_i computed at each camera location. Figure. 10(a) shows the entropy of the cameras for a given position of the walking person. The entropy is maximal when the man is close to the camera (C_{18}, C_1) and decreases according to the distance (C_6, C_7). The trajectory can be globally retrieved by concatenating the positions of the image sequence. Figure. 10(b) shows the ground truth trajectory superimposed with the estimated one up to a scale. Because of the choice of the α_i and the avoidance of camera calibration, the metric is not available but can be added if assumptions on the height of the object are added.

5 Conclusion

Most of multi-camera systems deal with single cameras acting as individual entities. Each camera provides information to the system without interaction with the other ones and the network is only viewed as a concatenation of sensors. Many constraints on the cameras or on the scene make it difficult to achieve standard tasks due to the huge amount of collected information that are unavoidably redundant leading to a resources' wastage. An approach considering each camera as a part of a unique entity is presented to overcome these problems. This paper presented a model which allows a system to retrieve and adapt its own structure and sort acquired signals according to a given task. Time is an important factor as iterative processes are the fundaments of the whole procedure.

References

1. Black, J., Ellis, T., Makris, D.: Wide area surveillance with a multi-camera network. In: Intelligent Distributed Surveillance Systems. (2003)
2. Gilbert, A., Bowden, R.: Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. In: Proc European Conference Computer Vision. (2006)
3. Baker, P., Aloimonos, Y.: Complete calibration of a multi-camera network. In: Omnivis. (2000)
4. Han, M., Kanade, T.: Multiple motion scene reconstruction from uncalibrated views. In: ICCV. (2001)
5. Sinha, S., Pollefeys, M.: Synchronization and calibration of camera networks from silhouettes. In: ICPR. (2004)
6. Svoboda, T., Matinec, D., Pajdla, T.: A convenient multi-camera self-calibration for virtual environments. In: PRESENCE. (2005)
7. Domke, J., Aloimonos, Y.: Multiple view image reconstruction: A harmonic approach. In: CVPR. (2007)
8. Basri, R., Rivlin, E.: Localization and homing using combinations of model views. In: Artificial Intelligent. (1995)
9. Kortenkamp, D., Weymouth, T.: Topological mapping for mobile robots using a combination of sonar and vision sensing. In: Proc. of National Conference on Artificial Intelligence. (1994)
10. I Horswill, P.: A vision-based artificial agent. In: Proc of the National Conference on Artificial Intelligence. (1993)
11. Lacheze, L., Benosman, R.: Visual localization using an optimal sampling of bags-of-features with entropy. In: IROS. (2007)
12. Tieu, K., Dalley, G., Grimson, W.E.L.: Inference of non-overlapping camera network topology by measuring statistical dependency. In: ICCV. (2005)
13. Debaecker, T., Benosman, R.: Bio-inspired model of visual information codification for localization: from retina to the lateral geniculate nucleus. In: Journal of Integrative Neuroscience. (2007)
14. Osada, Funkhouser, T., chazelle, B., Dobkin, D.: Shape distributions. In: ACM Trans. Graph. (2002)
15. Hastie, T., Tibshirani, R.: Discriminant analysis by gaussian mixtures. In: J R Stat Soc B. (1996)
16. Everitt, B., Hand, D.: Finite mixture distributions. In: Chapman and Hall. (1981)