Distributed EM Learning for Appearance Based Multi-Camera Tracking

Thomas Mensink, Wojciech Zajdel, Ben Kröse

DISTRIBUTED EM LEARNING FOR APPEARANCE BASED MULTI-CAMERA TRACKING

Thomas Mensink, Wojciech Zajdel and Ben Kröse

Informatics Institute, University of Amsterdam
Kruislaan 403 1098SJ Amsterdam
{tmensink,wzajdel,krose}@science.uva.nl

ABSTRACT

Visual surveillance in wide areas (e.g. airports) relies on cameras that observe non-overlapping scenes. Multi-person tracking requires re-identification of a person when he/she leaves one field of view, and later appears at another. For this, we use appearance cues. Under the assumption that all observations of a single person are Gaussian distributed, the observation model in our approach consists of a Mixture of Gaussians. In this paper we propose a distributed approach for learning this MoG, where every camera learns from both its own observations and communication with other cameras. We present the Multi-Observations Newscast EM algorithm for this, which is an adjusted version of the recently developed Newscast EM. The presented algorithm is tested on artificial generated data and on a collection of real-world observations gathered by a system of cameras in an office building.

Index Terms— Wide-area video surveillance, Data association, Mixture of Gaussian, EM algorithm, Distributed Computing

1. INTRODUCTION

With the increasing use of camera surveillance in public areas, the need for automated surveillance solutions is rising. A particular problem is camera surveillance in wide areas, such as airports, shopping centres etc. Such areas typically cannot be fully covered by a single camera, and surveillance of such places relies on a network of sparsely distributed cameras. Every camera observes a scene which is (partly) disjoint from the scenes observed by other cameras, as indicated in figure (1).

In this particular setting the problem of tracking persons across all cameras is difficult. Someone is first observed by one camera, then he is out of sight of any camera, and later on he reappears at another camera. We would like to know whether the two observed persons are in fact the same individual.

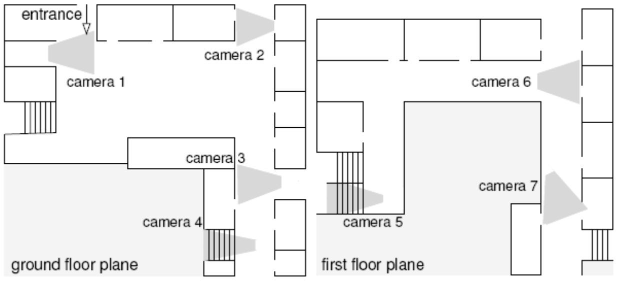

Fig. 1. A map of sparsely distributed cameras in a wide area video tracking setting

In current systems, a single computer collects all observations from all cameras and learns a model which describes the correspondence between observations and persons. Problems with such a central system include: privacy issues, because all observations are sent over a network and are centrally stored; network bottleneck, because all observations are sent to the same node; and the sheer risk of having one central system.

In this paper we present an alternative approach: a distributed system. In this approach there is no central processing unit nor a central repository and there is no all-to-all broadcasting communication. Each camera is a standalone tracking unit, which stores its own observations and exchanges only limited data with other cameras.

The local observations in combination with the exchanged data allow each camera to learn its own local model. In this paper we show that these local models converge quickly to the global model. Motivations for such a distributed system include, (i) it could use information sources that are spatially distributed more efficiently, (ii) it is more secure, because observations are never sent over the network and (iii) it could enhance the performance of computational efficiency, bandwidth usage and/or reliability [1, 2].

Similar to other approaches [3] we use appearance cues such as average colour, or length to find the correspondence between observations and persons. Since the same person will appear differently each time he is observed (because of illumination and pose) we model the observations as a stochastic

The work described in this paper was conducted within the EU Integrated Project COGNIRON (”The Cognitive Robot Companion”) and was funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

variable. We assume that the observations are samples drawn from a Gaussian distribution with person specific parameters, which are constant over time. In a system where p-persons are monitored, observations of all persons are generated by a Mixture of Gaussians (MoG) with p-kernels.

This distribution can be considered as the *observation* model which, together with the *transition* model, is needed to find the most likely tracks given a sequence of observations. In this paper we only consider the learning of the parameters of the MoG with the Expectation-Maximization (EM) algorithm [4]. Given the learned MoG we assign the most likely person to each observation. A track is the collection of all observations assigned to a person.

The use of the MoG model, allows us to exploit recently developed algorithms for distributed learning of the parameters of an MoG [5, 6]. The EM algorithm for MoG relies on a model's sufficient statistics (mean and covariance) over the observations. Interestingly, these sufficient statistics can be computed efficiently by a distributed system without the need to gather all observations in a central repository.

The research described in this paper is limited to appearance cues only. Spatial and temporal constraints (e.g. minimum travel time between two cameras) are neglected for clarity of presentation. Nevertheless the key ideas presented in this paper are verified with real-world data and can be extended to handle spatial/temporal constraints.

In the following section we first provide some background about learning parameters of an MoG with EM and introduce the approaches for learning an MoG in distributed systems. In our system we use Multi-Observations Newscast EM, presented in section 3, which is an extension of the gossip-based Newscast EM algorithm from [5]. In section 4 we present experimental results on both artificial and real data.

2. EM FOR MIXTURE MODELLING

When a person is passing through the field of view of a camera, the camera typically records several images of the person. In our approach we assume that passing through a field of view of a camera results in a *single* observation that encodes the appearance description of the person and the camera identifier. The exact appearance features will be defined in Section 4, for the moment we just will assume existence of some high-dimensional features describing the appearance of a person. When multiple persons are observed by multiple cameras the data consists of a sequence of observations.

In this paper we assume that the number of persons (p) and number of cameras (c) are known in advance. In contrast to [3], our system uses only appearance based features and we do not use a motion model. Each person is modelled as a single Gaussian function, with parameters $\theta_s = \{\pi_s, \mu_s, \Sigma_s\}$, where π_s is the mixing weight, μ_s the mean and Σ_s the covariance matrix for person s. The mixture of Gaussians is given by $X = \{\theta_s\}_{s=1}^{p}$.

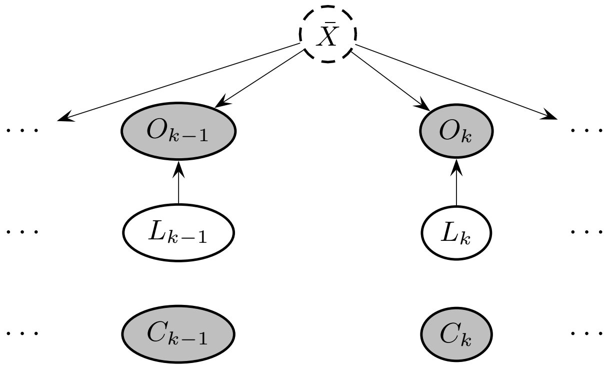

Fig. 2. Graphical Model of Appearance Based Tracking.

The model above assumes that most of variations in appearance features due to changing poses can be captured by a Gaussian pdf. In order to remove artefacts due to varying illumination at different cameras the appearance features have to be preprocessed [3].

In this section we first describe the generative model for these data. Secondly we describe how parameters of this model are learnt in the central case, where all data are available. Thirdly we present two different approaches for distributed EM.

2.1. Generative Model

A Bayesian Network is a convenient way to express the probabilistic relations between different variables in a system. The generative model, which is used by the central and distributed implementation of our Appearance Based Tracking system is shown in figure 2. In the model, X represents the Mixture of Gaussians, O the appearance part of an observation, L the label which uniquely identifies a person and C a camera identifier which is also part of an observation.

In the model L_k denotes the hidden variable which identifies the person appearing at the k's observation. This observation y_k of a person s consist of two parts, $y_k = \{o_k, c_k\}$. The first part o_k represents the appearance features of the person. The probability of these features given the model (X) and the label (L_k) is $p(o_k|X, L_k) = \mathcal{N}(o_k|\mu_{L_k}; \Sigma_{L_k})$. The second part ($c_k$) is the camera location where the person is seen, which is assumed to be noise free. Because we ignore camera specific variations, there are no links between the camera identifier C and any other nodes.

2.2. Learning the Parameters

A traditional (centralized) approach for learning the parameters of a MoG from the observations is the EM algorithm. The EM algorithm is an iterative procedure where for a number of iterations, first the responsibility $q_{k,s}$ for each observation and each kernel is calculated. This is the E-step (1), and it uses the current parameters $\{\pi_s, \mu_s, \Sigma_s\}$. Secondly for each kernel the parameters are optimized based on the current responsibilities, the M-step (2-4).

$$q_{k,s} = \frac{\pi_s\, \mathcal{N}(o_k|\mu_s; \Sigma_s)}{\sum_{r=1}^{p} \pi_r\, \mathcal{N}(o_k|\mu_r; \Sigma_r)} \tag{1}$$

$$\pi_s = \frac{\sum_{k=1}^{n} q_{k,s}}{n} \tag{2}$$

$$\mu_s = \frac{\sum_{k=1}^{n} q_{k,s}\ o_k}{n\pi_s} \tag{3}$$

$$\Sigma_s = \frac{\sum_{k=1}^{n} q_{k,s}\ o_k o_k^T}{n\pi_s} - \mu_s \mu_s^T \tag{4}$$

For the first iteration of the EM algorithm, the parameters have to be initialised. This can be done randomly by setting $q_{k,s}$ to some random positive value and then normalize it so that $\sum_{s=1}^{p} q_{k,s} = 1$. The parameters could also be initialised with the K-Means algorithm [7]. There are many EM implementation for systems with all observations available [7, 8].

2.3. Distributed EM

The previous section described a method for learning the parameters of an MoG if all data are available. In the distributed setting, the data are distributed over a number of *nodes*. In our situation, a node corresponds to a camera in the system.

There are two kinds of solutions for distributed EM, the first relies on a fixed routing scheme along all nodes, the other on gossip-based randomized communication.

In both approaches the E-step of the EM algorithm is computed locally. However for the M-step statistics from all nodes are necessary to create a global model. During the M-step, the parameters have to be updated using all observations, according to functions (2-4). It is important to note that all these functions are averages.

Using a fixed routing scheme along all nodes [6, 9] the complete statistics of all observations are gathered and sent over the network. Once all nodes have received the complete statistics they are able to update the parameters correctly. There are some variations which are less restrictive in their communication routing. However these may be converge to suboptimal maxima more often than standard EM [6].

Gossip based methods are fundamentally different. They do not rely on a fixed routing scheme, but on randomization. A gossip-based protocol could be used to calculate the mean value of a distributed dataset [5, 10].

Each node estimates its own mean, based on the data available at its node. Using a randomized communication protocol each node polls the estimates of several other nodes. With those gathered estimates the node can re-estimate its own mean. Each node's estimate of the mean will converge very fast to the correct mean. This class is robust and simple to implement.

One of the advantages of the gossip-based approach is that all nodes are running the same protocol in parallel, whereas in the other approach only one node at a time is carrying out computations.

3. MULTI-OBSERVATIONS NEWSCAST EM

In this section we present Multi-Observations Newscast EM, which is a generalisation of the Newscast EM [5]. Multi-Observations Newscast EM is a gossip-based distributed algorithm for learning the parameters of a mixture of Gaussians.

The distributed tracking system is seen as a network of nodes, each with a number of observations. In this network arbitrary point-to-point communication is possible between all nodes. The Newscast EM algorithm assumes that each node holds exactly one observation. Our Multi-Observations Newscast EM algorithm allows each node to have any number of observations.

Next, we introduce Multi-Observations Newscast Averaging which is the underlying principle for Multi-Observations Newscast EM. Thereafter we will describe the algorithm itself.

3.1. Multi-Observations Newscast Averaging

The Newscast Averaging algorithm [11] can be used for computing the mean of a set of observations that are distributed over a network. We present Multi-Observations Newscast Averaging, which can handle any number of observations at a node.

Suppose that observations $o_1, \ldots, o_n$ are arbitrarily distributed over a network of cameras $c_1, \ldots, c_c$. Each camera i in the network stores a number of observations n_i. The observations at node i are $o_{i,1}, \ldots, o_{i,n_i}$. The original Newscast EM assumes $n_i = 1$, for every node i.

The mean of all observations is given by:

$$\mu = \frac{1}{n} \sum_{k=1}^{n} o_k = \frac{1}{\sum_{i=1}^{c} n_i} \sum_{i=1}^{c} \sum_{k=1}^{n_i} o_{i,k}$$

To compute this mean distributively each node i sets $\hat{\mu}_i = \frac{1}{n_i} \sum_{k=1}^{n_i} o_{i,k}$ as its local estimate of μ, and it sets $w_i = n_i$ as its local estimate of n/c. Then it runs the following steps for a number of cycles:

1. Contact node j, which is chosen uniformly at random from $1, \ldots, c$.

2. Update nodes i and j mean estimates as follows

$$w_i' = w_j' = \frac{w_i' + w_j'}{2}$$

$$\hat{\mu}_i' = \hat{\mu}_j' = \frac{\hat{\mu}_i w_i + \hat{\mu}_j w_j}{w_i + w_i}$$

With this protocol each node's estimate rapidly converges to the correct mean. Important is the fact that the weighed mean of the local estimates is always the correct mean μ:

$$\mu = \frac{1}{\sum_{i=1}^{c} w_i} \sum_{i=1}^{c} w_i \hat{\mu}_i$$

It has been shown that the variance of the local estimates in Newscast Averaging, decreases at an exponential rate. After t cycles of Newscast the initial variance ϕ_0 of the local estimates is reduced on average to $\phi_t \leq \frac{\phi_0}{(2\sqrt{e})^t}$. The same bound of variance reduction can be proven for the proposed Multi-Observations Newscast Averaging algorithm. With the variance reduction we can derive the maximum number of cycles needed in order to guarantee, with high probability, that all nodes know the correct answer with a specific accuracy.

3.2. The Multi-Observations Newscast EM algorithm

Multi-Observations Newscast EM (MON-EM) is a distributed implementation of the EM algorithm for Gaussian Mixture learning. It uses the previously described gossip-based averaging algorithm for estimating the parameters of a Mixture. Multi-Observations Newscast EM is almost identical to a standard EM algorithm. The main difference is the M-step which is implemented as a sequence of gossip-based cycles.

Assume there is a set of observations $\{o_1, \ldots, o_n\}$, distributed over some nodes $\{c_1, \ldots, c_c\}$. The observations are assumed to be a set of independent samples from a common p-component mixture of Gaussian with the unknown parameters $\theta = \{\pi_s, \mu_s, \Sigma_s\}_{s=1}^{p}$. The task is to learn the parameters in a decentralized manner. All learning steps should be performed locally at the nodes, and these steps should involve as little communication as possible.

The E-step of our algorithm is identical to the E-step of standard EM, it can be performed locally and in parallel at all nodes. Each node i computes the new responsibilities $q_{i,k,s}(o_{i,k})$ (1) for every local observation $o_{i,k}$.

The M-step is implemented as a sequence of gossip-based cycles. Each node i starts with a local estimate $\hat{\theta}_i$ of the *correct* parameter vector θ. Then in every cycle, each node contacts at random another node. Both nodes replace their estimates with the weighed average of both. After some cycles each local estimate $\hat{\theta}_i$ has converged to the correct parameter θ.

The EM algorithm for node i, which runs identically and in parallel for each node is as follows:

1. **Initialisation** set the responsibilities $q_{i,k,s}(o_{i,k})$ randomly or with K-Means

2. **M-Step** initialise the local parameter estimates for each component s as follows:

$$w_i = n_i,$$
$$\hat{\pi}_{i,s} = \frac{1}{n_i} \sum_{k=1}^{n_i} q_{i,k}(s),$$

$$\hat{\mu}_{i,s} = \frac{1}{\hat{\pi}_{i,s}} \sum_{k=1}^{n_i} q_{i,k}(s)\ o_{i,k},$$
$$\hat{C}_{i,s} = \frac{1}{\hat{\pi}_{i,s}} \sum_{k=1}^{n_i} q_{i,k}(s)\ o_{i,k}\ o_{i,k}^T.$$

Then repeat for t cycles

   (a) Contact a node j, randomly chosen from $1, \ldots, c$.

   (b) Update the estimates of node i and j for each component s as follows:

$$w_i' = w_j' = \frac{w_i' + w_j'}{2}$$
$$\hat{\pi}_{i,s}' = \hat{\pi}_{j,s}' = \frac{\hat{\pi}_{i,s} w_i + \hat{\pi}_{j,s} w_j}{w_i + w_i}$$
$$\hat{\mu}_{i,s}' = \hat{\mu}_{j,s}' = \frac{\hat{\pi}_{i,s} \hat{\mu}_{i,s} w_i + \hat{\pi}_{j,s} \hat{\mu}_{j,s} w_j}{\hat{\pi}_{i,s} w_i + \hat{\pi}_{j,s} w_j}$$
$$\hat{C}_{i,s}' = \hat{C}_{j,s}' = \frac{\hat{\pi}_{i,s} \hat{C}_{i,s} w_i + \hat{\pi}_{j,s} \hat{C}_{j,s} w_j}{\hat{\pi}_{i,s} w_i + \hat{\pi}_{j,s} w_j}$$

3. **E-Step** Compute for each component s the new responsibilities $q_{i,k,s}(o_{i,s})$, using the M-step estimates $\pi_{i,s}$, $\mu_{i,s}$, $\Sigma_{i,s} = C_{i,s} - \mu_{i,s}\ \mu_{i,s}^T$.

4. **Loop** repeat the M-step and E-step until a stopping criterion is satisfied.

Important is the fact that, the weighed averages of the local estimates are always the EM-correct estimates. For the mean this is shown in equation (5), but it holds for all parameters in θ. So in each communication cycle the parameters converge at an exponential rate to the correct values (as is proven in [5]).

$$\mu_s = \frac{\sum_{i=1}^{c} w_i \pi_{i,s} \mu_{i,s}}{\sum_{i=1}^{c} w_i \pi_{i,s}} \tag{5}$$

In Multi-Observations Newscast EM, the initialisation of the M-step and the E-step are completely local to each node. Also a stopping criterion, based on the parameters, could be implemented locally. This will require that each node knows its parameter estimate of the previous EM-iteration. Only the M-step update requires communication between the nodes.

3.3. Initialisation

The original Newscast EM algorithm uses parameters which are initialised at random [5]. However, randomly initialised EM's more often yield suboptimal solutions than K-Means initialised EM's. Therefore we propose to initialise Newscast EM with Newscast K-Means, a distributed gossip based implementation of the K-Means algorithm.

K-Means is an iterative clustering procedure, which successively assigns each observation to the closest cluster mean and updates the cluster means according to the assigned observations. These steps are analogous to the E-step and M-step of EM, only they use different parameters and update functions.

In the E-step, for each observation k, assign $r_{k,s} = 1$ for the closest cluster s. In the M-step, for each cluster s, the cluster mean is set to:

$$\mu_s = \frac{1}{\sum_{k=0}^{n} r_{k,s}} \sum_{k=0}^{n} o_k r_{k,s}$$

Because the cluster mean is calculated as an average over all observations, given the responsibilities, this can be computed with the Newscast Averaging algorithm. Several experimental results have shown us that Newscast K-Means performs identically to a central K-Means implementation.

4. EXPERIMENTS

We have performed a series of tests with the presented Multi-Observations Newscast EM algorithm. The performance of the algorithm will be compared with a standard EM implementation. We evaluate the algorithms on both artificially generated and real data.

Setup The artificial data is generated randomly according to the model shown in figure 2. A set of 100 observations from 5 persons, which are distributed over 25 cameras is used as a standard set. Every observation $y_k = \{o_k, c_k\}$, consists of a 9-dimensional appearance vector o_k and a camera identifier c_k. The observations are randomly distributed over the cameras according to a uniform pdf. The difficulty of the generated data is measured by the c-separation and eccentricity values[12]. An increasingly difficult recognition problem is indicated by increasing eccentricity or decreasing c-separation values. The dataset has a c-separation of 1 and a eccentricity of 10.

Several datasets are generated to investigate the performance of the algorithm by variations in the number of persons, the number of cameras, and the distribution of the observations over the cameras.

The real data is collected from seven disjoint locations at the university building as in figure 1. In total we gathered 70 observations of 5 persons, with an equal number of observations per person. For this set the data association is manually resolved to have a ground truth. This data set is also used in [3].

The assumptions of Gaussian distributed noise in appearance features (due to illumination and pose) most likely will not hold without suitable preprocessing of the images. To minimize effects of variable illumination at different cameras (intensity and colour) we use a, so called, channel-normalized colour space [13]. To minimize non-Gaussian noise due pose we use a geometric colour mean [3], which is the mean colour of three separate regions of the person. This results in a 9-dimensional appearance vector. Instead of these colour features also other appearance characteristics, like texture features, could be used.

Evaluation The evaluation criteria should reflect two aspects of proper clustering. It is desirable that (i) all observations within a single reconstructed cluster belong to a single person, and (ii) all observations of a single person are grouped together in a single reconstructed cluster. These criteria are analogous to the precision and recall criteria often used in Information Retrieval settings. In order to evaluate both systems on one parameter we use the F1-measure.

Because the considered clustering problem is unsupervised, the true and proposed clusters are arbitrarily ordered. Therefore we define the precision (6) and recall (7) for a proposed cluster over the best match with a real cluster. Importantly precision and recall have to be considered jointly, because it is trivial to gain a recall of 1. This is done by clustering all observations into one cluster. However, this will result in a very low precision. The F1-measure (8) is the harmonic mean of precision and recall and will penalize for such cases.

$$Pr = \frac{1}{p} \sum_{s=1}^{p} \frac{\max_i |\hat{C}_s \cap C_i|}{|\hat{C}_s|} \tag{6}$$

$$Rc = \frac{1}{p} \sum_{i=1}^{p} \frac{\max_s |\hat{C}_s \cap C_i|}{|C_i|} \tag{7}$$

$$F1 = \frac{2 * Pr * Rc}{Pr + Rc} \tag{8}$$

4.1. Results on Artificial Data

In this article we have proposed to initialise Multi-Observations Newscast EM (MON-EM) with a distributed implementation of the K-Means algorithm. In table 1 we show the performance of randomly initialised and K-Means initialised MON-EM and standard EM using the standard set. The results show that the performance of MON-EM and standard EM are almost equal, both on the F1 measure as on the number of EM iterations. They also show an enormous increase of the F1-measure when the K-Means initialisation is used. The number of EM-iterations is also lower, but the iterations of the K-Means algorithm itself are not taken into account.

In Figure 3 we show the results of MON-EM when the number of cameras increases. The observations are randomly

Table 1. **Initialisation.** The performance of Newscast EM and standard EM on the standard set are shown. Both algorithms are initialised at random and with K-Means.

	standard EM	MON-EM
F1-Measure		
Random	.70 ± .07	.73 ± .07
K-Means	.89 ± .05	.87 ± .05
EM-iterations		
Random	33 ± 11	34 ± 12
K-Means	11 ± 6	13 ± 9

distributed over the cameras according to a uniform distribution. The performance is compared with standard EM, but for this algorithm nothing changes. The number of observations was fixed at 100. The figure shows that the performance of MON-EM is independent of the number of cameras.

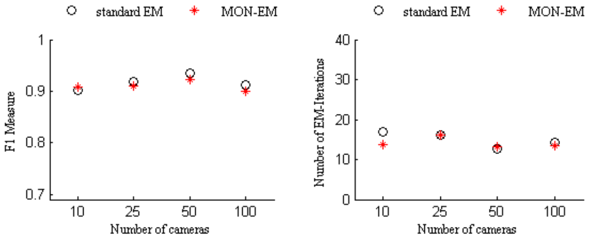

Fig. 3. **Number of cameras.** *Left* the performance and *right* the number of EM-iterations of the systems with a different number of cameras are shown

In figure 4 we show the results of MON-EM and standard EM when more persons are monitored by the system. The total number of observations is 10 times the number of persons tracked by the system. Even though the number of observations per person is not fixed, but random according to a uniform distribution. The performance of both algorithms on F1-measure and EM-iterations are comparable.

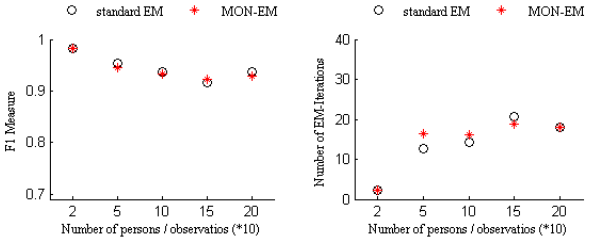

Fig. 4. **Number of persons.** *Left* the performance and *right* the number of EM-iterations of the systems with a different number of persons monitored are shown. The total number of observations is ten times the number of persons.

We would like to know if MON-EM is vulnerable when the distribution of the observations over the cameras is either rare or bad. Therefore we set up an experiment with the standard dataset, but with different distributions of the observations over the cameras. The distribution of the observations over the cameras is influenced by a distribution value. The higher the distribution value, the more the distribution is like a random uniform distribution. With a very low distribution value, the distribution is peaky. By the lowest distribution value used in the experiment, all cameras, except one, have only one single observation. All the remaining observations are at that one camera.

Figure 5 shows the performance of MON-EM and standard EM for different distribution values. The data distribution does not have an influence on the standard EM algorithm, because all data is gathered at a central site. The MON-EM algorithm performs comparable to the standard EM algorithm for any distribution value. Also the number of EM-iterations is not influenced significantly. Therefore we can conclude that Multi-Observations Newscast EM is not vulnerable for a rare or bad distribution of the observations over the cameras.

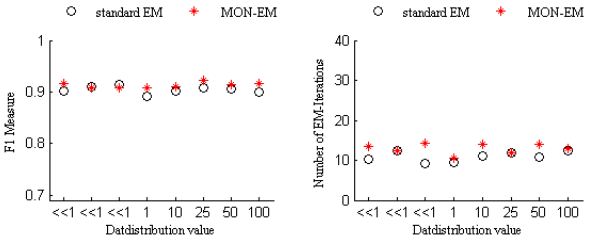

Fig. 5. **Data distribution.** The figure shows (*left*) the performance and (*right*) the EM-iterations of the system with different data distributions over the cameras.

4.2. Results on Real data

The result of the experiment with real data collected from 7 locations at the university building are shown in table 2. The results indicate that the MON-EM and standard EM perform equally on the real data.

Table 2. Performance on real data

	standard EM	MON-EM
F1-Measure	.63 ± .04	.64 ± .04
EM-iterations	9 ± 3	9 ± 3

In figure 6 the results of a typical clustering of the real data is shown. Each row represents a proposed cluster of the MON-EM algorithm. Especially the clusters from row 2, 4 and 5 score reasonably well. In these clusters many images of the same person are gathered together, even when the appearances do not correspond that much. For example in row 5, where the second and sixth observation are from the same person.

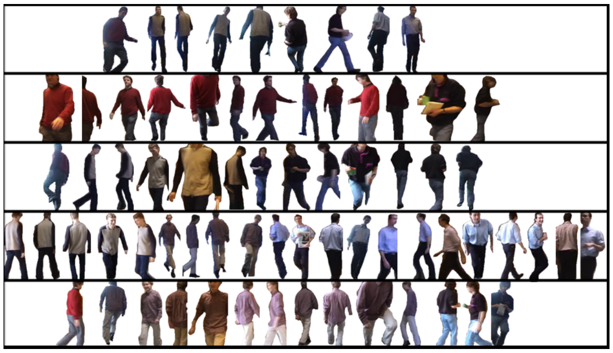

Fig. 6. **Typical Clustering result.** This figure shows the results of clustering the Real Dataset. Each row is a proposed cluster.

4.3. Discussion

The results of MON-EM and standard EM on artificial data shows that both algorithms perform equally well. Even in different settings, with a different number of cameras or number of persons; or a rare distribution of the observations over the cameras, Multi-Observations Newscast EM performs equally well to Central EM. The performance is comparable for both the F1-measurement as well as the number of EM iterations needed to achieve convergence. Also in many other test settings we have run, both algorithms perform quite the same.

The results on real data also show that Multi-Observations Newscast EM and standard EM perform almost equally. The c-separation and eccentricity values of the real data differs from the artificial data, which explains why the F1-measure is lower than on the standard set of the artificially generated data.

5. CONCLUSION

In this paper we have described an approach for multi-camera appearance based tracking, in a distributed system. We have presented the Multi-Observations Newscast EM (MON-EM) algorithm, which is a generalisation of the gossip-based Newscast EM algorithm, to learn the parameters of a Mixture of Gaussians. The experiments reveal that on both artificially generated data and on real data MON-EM performs equal to a standard EM implementation.

Although in real-world applications the simple Gaussian noise model may have to be replaced with a more complex model, the general idea of solving distributed tracking by distributed probabilistic learning remains valid. The Multi-Observation Newscast Averaging algorithm is able to compute almost any kind of statistics over a set of observations distributed over some nodes.

The presented system does not take into account temporal and spatial constraints on tracks (e.g. minimum travel time between cameras). The probabilistic model could be enhanced with a transition model, using discrete features, like camera index and wall clock time. We have planned to incorporate such a transition model into the algorithm along the ideas presented in [3, 14].

The presented system assumes that the number of persons are known in advance. In [3] a more elaborate model was used, in which the most likely number of persons could be inferred from the data. In our approach we did not use this elaborate model, but restricted ourself to a simple model to show the distributed learning algorithm. In a future study we plan to use our Multi-Observation Newscast EM in the setting of this elaborative model. Another extension could be, to change Multi-Observation Newscast EM into a *greedy* EM algorithm, which gradually yields more kernels. This could be done according to ideas presented in [8, 15].

6. REFERENCES

[1] P. Remagnino, A.I. Shihab, and G.A. Jones, "Distributed intelligence for multi-camera visual surveillance," *Pattern Recognition*, vol. 37, no. 4, pp. 675–689, april 2004, Special Issue on Agent-based Computer Vision.

[2] Katia P. Sycara, "Multiagent systems," *AI Magazine*, vol. 19(2), pp. 79–92, 1998.

[3] Wojciech Zajdel, *Bayesian Visual Surveillance - From Object Detection to Distributed Cameras*, Ph.D. thesis, Universiteit van Amsterdam, 2006.

[4] Arthur P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[5] W. Kowalczyk and N. Vlassis, "Newscast EM," *Advances in Neural Information Processing Systems*, vol. 17, 2005.

[6] R. Nowak, "Distributed em algorithms for density estimation and clustering in sensor networks," *Signal Processing, IEEE Transactions on*, vol. 51, pp. 2245–2253, 2003.

[7] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[8] J. Verbeek, N. Vlassis, and B. Kröse, "Efficient greedy learning of gaussian mixture models," *Neural Computation*, vol. 15, pp. 469–485, 2003.

[9] George Forman and Bin Zhang, "Distributed data clustering can be efficient and exact," *SIGKDD Explor. Newsl.*, vol. 2, no. 2, pp. 34–38, 2000.

[10] David Kempe, Alin Dobra, and Johannes Gehrke, "Gossip-based computation of aggregate information,"

44th annual IEEE symposium on Foundations of Computer Science, p. 482, 2003.

[11] Márk Jelasiy, Wojtek Kowalczyk, and Maarten van Steen, “Newscast computing,” Tech. Rep., Vrije Universiteit Amsterdam, 2003.

[12] Sanjoy Dasgupta, “Learning mixtures of gaussians,” in *FOCS ’99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, 1999, p. 634, IEEE Computer Society.

[13] Mark S. Drew, Jie Wei, and Ze-Nian Li, “Illumination-invariant color object recognition via compressed chromaticity histograms of color-channel-normalized images,” in *Int. Conf. on Computer Vision*, 1998, pp. 533–540.

[14] W. Zajdel and B.J.A. Kröse, “A sequential bayesian algorithm for surveillance with non-overlapping cameras.,” *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 8, pp. 977–996, 2005.

[15] N. Vlassis, Y. Sfakianakis, and W. Kowalczyk, “Gossip-based greedy gaussian mixture learning,” in *10th Panhellenic Conf. on Informatics.*, 2005.