Multi-Observations Newscast EM for Distributed Appearance Based Tracking

Thomas Mensink   Wojciech Zajdel   Ben Kröse

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 people, when they leave one field of view and later enter 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: one component for each person. In this paper we propose a distributed approach for learning this MoG, where every camera learns from its own observations and the communication with other cameras. We propose a modified version of the recently developed Newscast EM algorithm for this. We test our alogirhm on artificial generated data and on a collection of real-world observations gathered by a system of cameras in an office building.

1 Introduction

Automated surveillance systems are required to track multiple persons. In wide areas, such as airports and shopping centres, the surveillance relies on many cameras, where the field-of-view of one camera covers only a relatively small scene that does not overlap with the scenes observed by other cameras. Tracking in this scenario requires camera-to-camera association of observations when a person leaves one scene and later appears at some other.

In current systems, a single computer collects all observations from all cameras and learns a model which describes the correspondence between observations and persons, see Figure 1(*left*). Problems with such a central system include: (i) privacy issues, because all observations are send over a network; (ii) network bottleneck, because all observations are send to one node; and (iii) the risks of a single point of failure when having one central system.

In this paper we present an alternative approach: a distributed system. The cameras collectively track the persons using the observations provided by all cameras. There is no central repository and there is no all-to-all broadcasting communication. In this approach each camera is a standalone tracking unit, which stores its own observations and exchanges only a limited amount of data with other cameras. The distributed system is shown in Figure 1(*right*).

Motivations for such a distributed system include, (i) it could use information sources that are spatial distributed more efficiently, (ii) it is more secure, because observations are never send over the network and (iii) it could enhance the performance of computational efficiency, bandwidth usage and/or reliability [10, 11].

Similar to other approaches [13] we use appearance cues such as average colour, and length to find the correspondence between observations and persons. The appearance cues are modelled as a stochastic variable, because a person will appear differently each time he is observed (due to illumination and pose). 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 components. The ideas presented in this paper could also be used with other distributions from the exponential family.

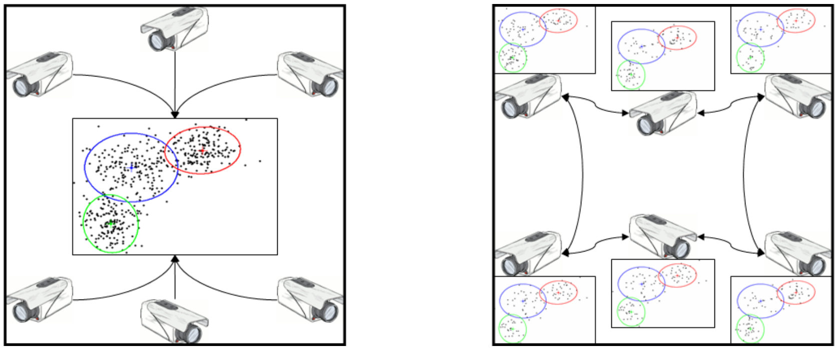


Figure 1: Central and Distributed System, (*left* a central system, where all cameras send their observations to a single computer which learns a certain model, and (*right*) a distributed system, where all cameras learn the same model by inference over their own observations and by communication with the other cameras.

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 [3]. The transition model, that defines spatial and temporal constraints, is neglected for clarity of presentation.

The use of the MoG model, allows us to exploit recently developed algorithms for distributed learning of the parameters of an MoG [7, 9]. The EM algorithm for MoG relies on the model's sufficient statistics (averages) of the observations. Interestingly, such summary statistics can be computed very efficiently by a distributed system without the need to gather all observations in a central repository.

In the following section we describe the probabilistic generative model, and provide some background about learning the parameters of an MoG with the EM algorithm. In Section 3 we present Multi-Observations Newscast EM to learn the parameters of an MoG from distributed observations. We present experimental results on both artificial and real data in Section 4. Finally in Section 5 we present our conclusions and some possibilities for further research.

2 EM for Mixture Modelling

We consider the problem of camera-to-camera tracking of p persons observed with c sparsely distributed cameras in an office-like environment. Since our method focuses on camera-to-camera trajectories we assume that a pass through a field of view of a camera results in a *single* observation that encodes the appearance description of the person and a 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 contrast to [13], we use only appearance based features and we do not use a motion model. Each person is modelled as a single Gaussian function, with parameters $\theta_s = \{\pi_s, \mu_s, \Sigma_s\}$, which are assumed to be constant over time. For person s his parameters are π_s the mixing weight, μ_s the mean and Σ_s the covariance matrix. The MoG is given by $X = \{\theta_s\}_{s=1}^{p}$. This model assumes that most of the variations in the appearance features are due to changes in pose can be captured by a Gaussian probability density function (pdf). In order to remove artefacts due to varying illumination at different cameras the appearance features will be preprocessed [13].

In this section we first describe the generative model for these data. Then we describe how parameters of this model are learnt in the central case, where all data are available.

2.1 Generative Model

Cameras cannot directly observe the identity of a person, but instead they provide appearance measurements (like colour or height of a person) and spatial-temporal cues (like position or time) about an observation of a person. Since these measurements do not uniquely identify an individual person, a probabilistic model is used to encode the ambiguous relation between the measurement and the identities.

A Bayesian network is a convenient way to express the probabilistic relations among different variables in a system. The described generative model, will be used by the central and by the distributed implementation of our appearance based tracking system.

In the model L_k denotes the hidden variable which identifies the person appearing at the kth observation. This observation of a person consist of two parts, $\{o_k, c_k\}$. The first part o_k represents the appearance features of the person. The second part (c_k) is the camera location where the person is seen, which is assumed to be noise free.

In the model X represents the parameters of the MoG. It is assumed that the observation o_k is a noisy reading, generated from this MoG. The probability of the observation given the model (X) and the label (L_k) is $p(o_k|X, L_k) = \mathcal{N}(o_k|\mu_{L_k}; \Sigma_{L_k})$. Because we ignore camera specific variations, there are no probabilist relations between the camera identifier c_k and any other variables.

2.2 Tracking People

To track people with the gathered observations, the parameters of the MoG have to be learned and the observations have to be clustered. A traditional approach for learning the parameters of an MoG from observations is the Expectation-Maximization (EM) algorithm [3, 1].

The EM algorithm is general method for finding the maximum likelihood parameter estimate of an underlying distribution from data. It is an iterative procedure which assumes that all data is available. For a number of iterations two steps are performed. First the responsibility $q_k(s)$ for each data point o_k ($k \in \{1, \ldots, n\}$) and for each kernel s is calculated based on the current parameters $\{\pi_s, \mu_s, \Sigma_s\}$: the E step (1). Second for each kernel the parameters are optimized based on the current responsibilities: the M step (2).

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

$$\pi_s = \frac{\sum_{k=1}^{n} q_k(s)}{n} \qquad \mu_s = \frac{\sum_{k=1}^{n} q_k(s) o_k}{n\pi_s} \qquad \Sigma_s = \frac{\sum_{k=1}^{n} q_k(s) o_k o_k^T}{n\pi_s} - \mu_s \mu_s^T \tag{2}$$

The performance of the EM algorithm depends highly on the initialisation of the parameters. It is common to use the k-means algorithm to find a suitable initialisation for starting the EM algorithm [1]. For the central system there are many EM algorithm, such as [1, 12].

3 Multi-Observations Newscast EM

The previous section described a method for learning the parameters of an MoG if all data are available. In this section we present Multi-Observations Newscast EM (MON-EM) for the distributed setting, where the data are distributed over a number of *nodes*. In our situation, a node corresponds to a camera in the system.

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 between all nodes is possible. MON-EM is a generalisation of the gossip-based Newscast EM [7] algorithm. Newscast EM assumes that each node holds exactly one observation, our MON-EM algorithm allows each node to have any number of observations.

The E step of the EM algorithm is computed at each node, for all data locally available. However, during the M step the parameters have to be updated using all observations from all nodes according to the functions (2). It is important to note that all these functions are averages.

Gossip-based methods could be used to calculate the mean value of a set of distributed data [7, 6]. Using a randomized communication protocol each node repeatedly contacts another node. The nodes exchange their local parameter estimates and combine them by weighed averaging. Each node's local parameter estimates will converge exponentially fats to the correct mean.

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

3.1 Multi-Observations Newscast Averaging

The Newscast Averaging algorithm [5] can be used for computing the mean of a set of observations that are distributed over a network. We present Multi-Observations Newscast Averaging (MON-Averaging) as an extension of Newscast Averaging. MON-Averaging uses additional variables to take into account the number of observations at a node, and does not have a constraint on the number of observations at a node.

Suppose that observations $o_1, \ldots, o_n$ are arbitrarily distributed over a network of c cameras. Each camera i in the network stores a number of observations n_i, the observations at camera i are $o_{i,1}, \ldots, o_{i,n_i}$.

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 camera 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. Nodes i and j update their local parameter estimates as follows:

$$w_i' = w_j' = \frac{w_i + w_j}{2} \qquad\qquad \hat{\mu}_i' = \hat{\mu}_j' = \frac{\hat{\mu}_i w_i + \hat{\mu}_j w_j}{w_i + w_i}$$

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.

It has been proven that the variance of the local estimates in Newscast Averaging, decreases at an exponential rate [5, 7]. 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 MON-Averaging algorithm.

With this bound of variance reduction we can derive the maximum number of cycles that are needed in order to guarantee with high probability that all nodes know the correct answer with a specific accuracy. We have performed an experiment to investigate wether the maximum number of cycles are dependent on the distribution of the obervations over the nodes. The system is considered to be converged when the absolute difference between the current estimate of the mean and the previous one is smaller than 10^{-5}.

For this experiment we have generated several sets of 1000 observations distributed over 100 nodes. The distribution of the observations over the cameras is influenced by a distribution value. The higher this distribution value is, the more the distribution is like a uniform distribution. While by very low distribution values, the distribution is very peaky.

In Figure 2 we compare our presented MON-Averaging algorithm with weight updating, to an implementation without these updates. Without these updates, the sum of the weights keeps equal to the number of observations in the system, and therefore the weighed average of all local estimates also equals μ. However, as can be seen in the figure the presented MON-Averaging algorithm will always coverge quickly, while the implementation without weight updating needs much more communication cycles for low distribution values.

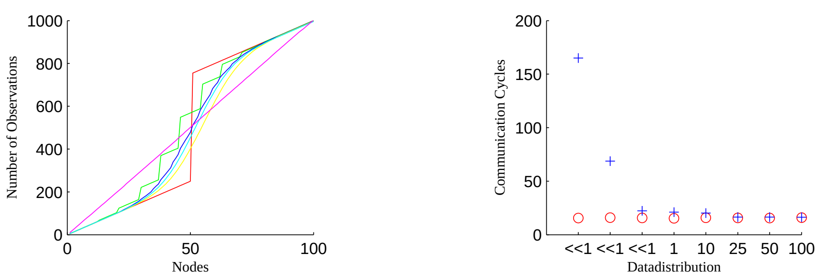


Figure 2: Different distributions of the data over the nodes. (*Left*) The distribution of observations over the nodes, each line presents a different distribution value. (*Right*) The number of communication cycles in order to converge for the different data distributions. The $\circ$ mark indicates the MON-Averaging with updating weights, and the $+$ mark indicates the implementation without updating weights

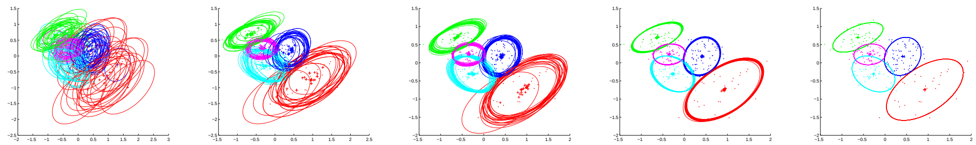


Figure 3: Several communication cycles during a M step of MON-EM. Each sub-figure shows the local parameter estimates from 25 nodes for the 5 clusters during the M-update step. The figures show the result before communication and after 1, 2, 5, 10 communication rounds respectively.

3.2 Multi-Observations Newscast EM

Multi-Observations Newscast EM (MON-EM) is a distributed implementation of the EM algorithm for Gaussian Mixture learning. The main difference with standard EM is that the M step uses the previously described gossip-based averaging algorithm for updating the parameters of the MoG.

Assume there is a set of observations $\{o_1, \ldots, o_n\}$, distributed over c cameras. 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, where 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. Each node i computes the new responsibilities $q_{i,k}(s)$ (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 for a number of cycles, each node contacts at random another node, and both nodes replace their estimates with the weighed average.

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

1. **Initialisation** set the responsibilities $q_{i,k}(s)$ randomly or with a distributed k-means algorithm.

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

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

   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:

$$w'_i = w'_j = \frac{w_i + w_j}{2} \qquad \hat{\pi}'_{i,s} = \hat{\pi}'_{j,s} = \frac{\hat{\pi}_{i,s} w_i + \hat{\pi}_{j,s} w_j}{w_i + w_i}$$

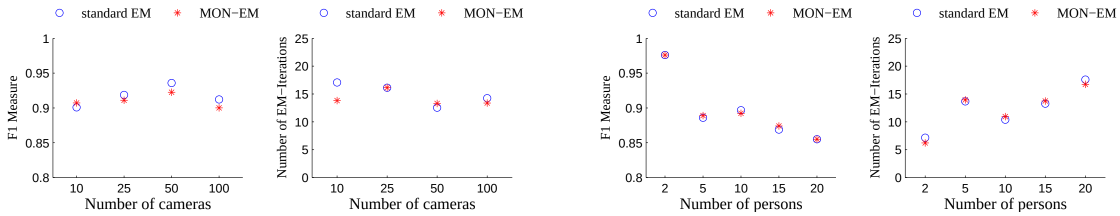
$$\hat{\mu}'_{i,s} = \hat{\mu}'_{j,s} = \frac{\hat{\pi}_{i,s}\hat{\mu}_{i,s} w_i + \hat{\pi}_{j,s}\hat{\mu}_{j,s} w_j}{\hat{\pi}_{i,s} w_i + \hat{\pi}_{j,s} w_j} \qquad \hat{C}'_{i,s} = \hat{C}'_{j,s} = \frac{\hat{\pi}_{i,s}\hat{C}_{i,s} w_i + \hat{\pi}_{j,s}\hat{C}_{j,s} w_j}{\hat{\pi}_{i,s} w_i + \hat{\pi}_{j,s}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

3. **E step** Compute for each component s and for every local observation $o_{i,k}$ the new responsibilities $q_{i,k}(s)$. 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.

A few observations about the algorithm are in order. First, it is important to see that, the weighed averages of the local estimates are always the EM-correct estimates. Therefore in each communication cycle the parameters converge at an exponential rate to the correct values (as is proven in [7]). Within one communication cycle each node initiates a single contact, thus a communication cycles involves c updates of the parameters. Second, in MON-EM, only the M step update involves communication between nodes. The other steps, the initialisation of the M step and the E step, are computed locally at each node. Also a stopping criterion, based on the parameters, could be implemented locally. Third, the amount of communication is independent of the number of observations.

In Figure 3 MON-EM is shown to learn the parameters of an MoG with 5 components. For each node the mean and the ellipse of the covariance is drawn, before communication and after some communication cycles.



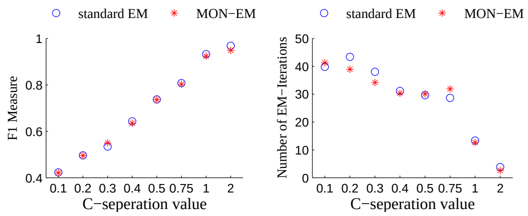


Figure 4: Performance of MON-EM and standard EM. The average F1 performance and average number of EM-iterations of both algorithms in different settings: increasing number of cameras (*top left*), increasing number of persons (*top right*), and different c-separation values (*bottom left*).

4 Experiments

In this section we present results of experiments on artificial generated data and on real data. The performance of the presented Multi-Observations Newscast EM algorithm will be compared with a standard EM algorithm. Both EM algorithms are initialised with the k-means algorithm. For MON-EM we have used a distributed implementation of k-means, which is implemented analogous to the MON-EM algorithm. This distributed k-means algorithm uses the MON-Averaging algorithm to update the parameters of the means. For the standard EM algorithm we used a standard k-means implementation.

4.1 Artificial Data

For these experiments we randomly generated data according to the model described in Section 2.1. Each dataset consists of 100 observations from $p \in \{2, 5, 10, 15, 20\}$ persons, which are randomly distributed over $c \in \{10, 25, 50, 100\}$ cameras according to a uniform pdf. Each observation $y_k = \{o_k, c_k\}$, consists of a 9-dimensional appearance vector o_k and a camera identifier c_k. The difficulty of the generated data is measured by the c-separation and eccentricity values [2]. An increasingly difficult recognition problem is indicated by increasing eccentricity or decreasing c-separation values. The datasets have an eccentricity of 10, while the c-separation value range between $c \in \{.1, \ldots, 2\}$.

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. Because the considered clustering problem is unsupervised, the true and proposed clusters are arbitrarily ordered. Therefore we define the precision and recall (3) for a proposed cluster i over the best match with a real cluster s.

$$Pr = \frac{1}{p} \sum_{s=1}^{p} \frac{\max_i |\hat{C}_s \cap C_i|}{|\hat{C}_s|} \qquad Rc = \frac{1}{p} \sum_{i=1}^{p} \frac{\max_s |\hat{C}_s \cap C_i|}{|C_i|} \qquad F1 = \frac{2 * Pr * Rc}{Pr + Rc} \tag{3}$$

In order to evaluate MON-EM and standard EM on one parameter we use the F1-measure (3), which is the harmonic mean of precision and recall. Besides, we also compare both algorithms on the number of EM iterations needed to converge. MON-EM and standard EM are implemented with a stopping criterion based on the parameters. The algorithm will stop when the absolute difference between the current parameters and the previous ones is less than 10^{-3}.

In Figure 4 we show the results of MON-EM and standard EM for various test settings. In the experiment where the number of cameras changes, the dataset contains observations from 5 persons, and it has a c-separation value of 1. The performance of MON-EM is compared with standard EM, but for the latter algorithm nothing changes. The results show that the performance of MON-EM is independent of the number of cameras.

When more persons are monitored by the system, the number of kernels grow equally. In this experiment the dataset has observations from $p \in \{2, 5, 10, 15, 20\}$ persons. The c-separation value is 1, and the observations are randomly distributed over 25 cameras. The performance of both algorithms on F1-measure and EM-iterations is almost identically.

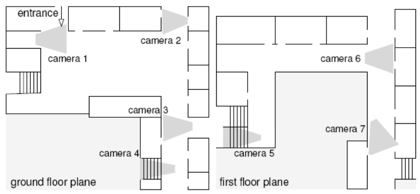


Figure 5: A map of sparsely distributed cameras in a wide area video tracking setting

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

In this experiment we investigate the influence of the data difficulty on MON-EM. The c-separation value of the generated dataset ranges from $c \in \{.1, \dots, 2\}$. The dataset contains observations from 5 persons. The results show that, when the dataset becomes easier (i.e the c-separation value increases) the F1 measures increases, which implies that the clustering is better. They also show that there is almost no performance difference between MON-EM and standard EM.

4.2 Results on Real data

The real data is collected from seven disjoint locations at the university building, as in Figure 5. 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 [13].

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 [4]. To minimize non-Gaussian noise due pose we use a geometric colour mean [13], 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.

To compare the performance of MON-EM and standard EM the same criteria (F1-measure and number of EM iterations) are used as by the experiments with artificial data. For MON-EM the F1 measure of the resulted clustering is $.64 \pm .04$ and for standard EM it is $.63 \pm .04$. The number of EM iterations needed to converge is for both algorithms equal to 9 ± 3. These results indicate that the MON-EM and standard EM perform equally on the real data.

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. None of the clusters gain a precision or recall of one. But still some proposed clusters score reasonably well, for example the first and second row. In these cluster most of the observations belong to one single person, even when the appearances do not correspond that much. For example in row 1, where the first and second observation are from the same person.

4.3 Discussion

The results of MON-EM and standard EM prove that the algorithms perform equally well, on both artificial generated and real data. Even in different settings, with a different number of cameras or number of persons; or different data difficulty, MON-EM performs the same as standard EM. The performance is comparable for both the F1-measurement as well as the number of EM iterations needed to achieve convergence.

The c-separation value of the real data is around $.44$, which explains why the F1-measure is lower than on some of the artificially generated data sets used.

5 Conclusion

In this paper we have considered an approach for appearance based tracking of persons with multiple cameras. For this tracking the parameters of a Mixture of Gaussians have to be learned. We have presented the Multi-Observations Newscast EM (MON-EM) algorithm for learning these parameters from a set of observations distributed over the cameras. MON-EM is a generalisation of the gossip-based Newscast EM algorithm. The experiments reveal that MON-EM performs equally well to a standard EM implementation, on both artificially generated data and on real data.

We have proven that learning from distributed data does not have to be an approximation of learning from central data. Learning from distributed data with MON-EM is equal to learning from central data with standard EM, this will hold for any kind of data. The performance of MON-EM is independent of the number of cameras, and the distribution of the observations over the cameras.

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 [13, 14].

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 a number of cameras.

The ideas presented in this paper could be extremely useful in other domains as well. Especially in domains where the E step is computational heavy, for example clustering of images based on SIFT descriptors. With the presented distributed Multi-Observations Newscast EM, the computational load could be balanced over several nodes.

References

[1] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] S. Dasgupta. Learning Mixtures of Gaussians. In *Proceedings of the Symposium on Foundations of Computer Science*, 1999.

[3] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1977.

[4] M. Drew, J. Wei, and Z. Li. Illumination-Invariant Color Object Recognition via Compressed Chromaticity Histograms of Color-Channel-Normalized Images. In *Proceedings of the International Conference on Computer Vision*, 1998.

[5] M. Jelasiy, W. Kowalczyk, and M. van Steen. Newscast Computing. Technical report, Vrije Universiteit Amsterdam, 2003.

[6] D. Kempe, A. Dobra, and J. Gehrke. Gossip-Based Computation of Aggregate Information. *Proceedings of the IEEE symposium on Foundations of Computer Science*, 2003.

[7] W. Kowalczyk and N. Vlassis. Newscast EM. *Advances in Neural Information Processing Systems*, 2005.

[8] T. Mensink. *Multi-Observations Newscast EM for Distributed Multi-Camera Tracking*. Master thesis, Universiteit van Amsterdam, 2007.

[9] R. Nowak. Distributed EM Algorithms for Density Estimation and Clustering in Sensor Networks. *IEEE Transactions on Signal Processing*, 2003.

[10] P. Remagnino, A. Shihab, and G. Jones. Distributed Intelligence for Multi-Camera Visual Surveillance. *Pattern Recognition*, 2004.

[11] K. Sycara. Multiagent Systems. *AI Magazine*, 1998.

[12] J. Verbeek, N. Vlassis, and B. Kröse. Efficient Greedy Learning of Gaussian Mixture Models. *Neural Computation*, 2003.

[13] W. Zajdel. *Bayesian Visual Surveillance - From Object Detection to Distributed Cameras*. PhD thesis, Universiteit van Amsterdam, 2006.

[14] W. Zajdel and B. Kröse. A Sequential Bayesian Algorithm for Surveillance with Non-Overlapping Cameras. *International Journal of Pattern Recognition and Artificial Intelligence*, 2005.