Spatiotemporal Multiple Persons Tracking Using Dynamic Vision Sensor

Ewa Piątkowska, Ahmed Nabil Belbachir, *Member* IEEE, Stephan Schraml
Safety and Security Department
AIT Austrian Institute of Technology
Donau-City Strasse 1/5, A-1220 Vienna, Austria
{ewa.piatkowska; nabil.belbachir; stephan.schraml}@ait.ac.at

Margrit Gelautz, *Member* IEEE
Institute for Software Technologies and Interactive Systems
Vienna University of Technology
Favoritenstrasse 9-11, A-1040 Vienna, Austria
gelautz@ims.tuwien.ac.at

Abstract

Although motion analysis has been extensively investigated in the literature and a wide variety of tracking algorithms have been proposed, the problem of tracking objects using the Dynamic Vision Sensor requires a slightly different approach. Dynamic Vision Sensors are biologically inspired vision systems that asynchronously generate events upon relative light intensity changes. Unlike conventional vision systems, the output of such sensor is not an image (frame) but an address events stream. Therefore, most of the conventional tracking algorithms are not appropriate for the DVS data processing. In this paper, we introduce algorithm for spatiotemporal tracking that is suitable for Dynamic Vision Sensor. In particular, we address the problem of multiple persons tracking in the occurrence of high occlusions. We investigate the possibility to apply Gaussian Mixture Models for detection, description and tracking objects. Preliminary results prove that our approach can successfully track people even when their trajectories are intersecting.

1. Introduction

Motion analysis plays a crucial role many computer vision applications and that is why much effort has been made to develop robust and reliable tracking methods. The goal of tracking is to estimate the trajectory of an object as it moves around the scene, using the information about the change of the object's location over time. Due to the motion complexity as well as changing appearance of the tracked object (non-rigidity of the objects, occlusions), tracking can be a very challenging task. Over the past two decades, a large number of different tracking algorithms have been proposed including point-based [10], feature-based [11] [4] [5] trackers as well as contour [3] or silhouette trackers. Furthermore, stochastic methods have been successfully used to model an object's dynamics; i.e. Bayesian filtering and its specific variants such as Kalman [2] or Particle filters [4] [16].

All abovementioned methods were proposed for conventional, frame-based video processing. In this work, however, we investigate the task of tracking in the context of data captured by the Dynamic Vision Sensor (DVS) [1].

The significant difference in the type of input data enforces the development of new processing techniques to obtain a better performance and stability of the tracking algorithm. Only few attempts have been made to track objects using the DVS, including work by Litzenberger et al. [8] and Schraml et al. [13]. A more detailed description of those algorithms is provided in section 2. In this work we address the problem of multiple objects tracking in the occurrence of occlusions. The proposed method is based on clustering techniques extended by a stochastic prediction of the objects' states in particular time steps.

This paper is organized as follows. First, we introduce the idea of the Dynamic Vision Sensor and specify the key issues connected with processing data captured by this kind of sensor. Then, methods proposed for tracking with the DVS are discussed. In Section 3 we provide the theoretical overview of the method used in our approach. Finally, the proposed algorithm is described in Section 4 along with a presentation of results. We conclude with a short summary.

2. Problem Domain

In this section, the Dynamic Vision Sensor is briefly described to explain the main differences between the data captured by conventional and dynamic vision systems. Knowing the characteristics of the input data is important for the choice of the tracking method. Then, we present and discuss two existing algorithms for tracking in the address events domain.

2.1. Dynamic Vision Sensor

The Dynamic Vision Sensor (DVS) [7] introduced in 2005 is a biologically inspired vision device. In this work, we use the stereo DVS which consists of two sensing elements of self-spiking pixels. Pixels operate independently and asynchronously generate events upon relative light intensity changes in the scene.

978-1-4673-1612-5/12/$31.00 ©2012 IEEE

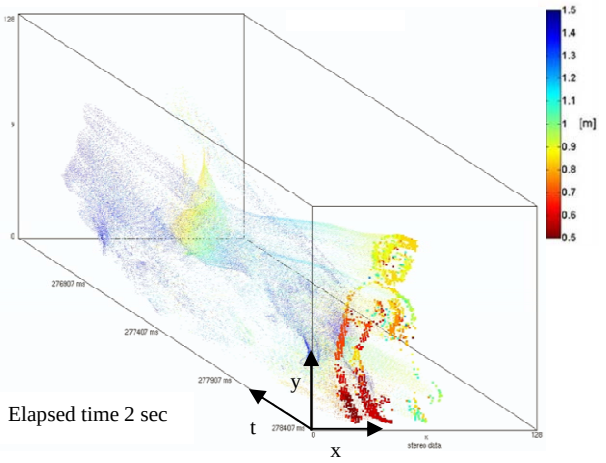

Figure 1: The representation of the spatiotemporal data from the Dynamic Vision Sensor in 4D(x,y,t,z). The depth information is color coded.

The increase or decrease of light intensity is encoded by the polarity of the events, which can be ON or OFF, respectively. Using the stereo configuration allows to reconstruct the depth information of the captured data by the method proposed in [14]. Wide dynamic range and high temporal resolution of the Dynamic Vision Sensor are a motivation to develop reliable tracking algorithms that could be beneficial in many applications, especially dealing with difficult environment conditions (outdoor applications) or high-speed motion analysis.

2.2. Tracking using DVS

There is a significant difference between tracking using conventional vision systems and the Dynamic Vision Sensor. In the former, tracking usually consists of several steps. First, the representation of the target model needs to be specified to detect an object in each frame. Then, the motion can be estimated from the change in the objects' location over time. Main problems connected with frame-by-frame analysis are false detections dealing with object to tracker association in a multiple objects tracking task. In contrary, the Dynamic Vision Sensor operates in frame free mode, which means that events are generated as the change in light intensity occurs. In result, the obtained address event stream represents continuous motion and the frame free approach eliminates the problem of object to tracker association. Furthermore, tracking is supported by the sensor as only the scene dynamics is captured and segmentation of the moving object from the static background is done on-chip.

Having only one object moving in the scene, the task of tracking is trivial while all generated events represent motion of this object (not counting additional noise). However, multiple objects tracking is rather challenging as events need to be associated with particular objects' motion paths.

As the sensor encodes only the relative light intensity change we cannot use appearance features such as color or texture to differentiate objects among each other. The only information that could be used is depth information, while occluding objects are usually in different distance from the sensor. While estimating the object's motion path there are two constraints to be met: the spatial consistency and temporal smoothness. Spatial consistency means that events assigned to the object occupy approximately the same 3D space region. Temporal smoothness, on the other hand, ensures that the position as well as the size of the object does not change drastically over time.

2.3. Related Work

The Dynamic Vision Sensor technology is relatively new and is not yet widely used in the field of computer vision. Thus, only few attempts were made to provide tracking algorithms designed for processing address events data. As it was described previously, the DVS is sensitive to the change in light intensity and that is why each moving object generates a bulk of corresponding address events. The most intuitive way to process an address events stream is to use a clustering method where each cluster represents a moving object. Such approach was used in the work by Litzenbeger et al. [8] and Schraml et al. [13]. Litzenberger et al. [8] proposed an embedded vision system for tracking vehicles using a monocular Dynamic Vision Sensor. Incoming events are assigned to the circular clusters by a Euclidean distance criterion where the position of the event is evaluated by the value of the cluster's seek radius. Events are not buffered to keep the low-memory constraint, which is important in embedded algorithms. The algorithm is inspired by a mean-shift approach, as the center of the cluster moves toward the occurrence of the majority of the most recently added events. The second method was proposed by Schraml et al. [13] for people tracking in crowded areas using a Stereo Dynamic Vision Sensor mounted in an overhead position. Incoming events are assigned to the clusters according to the Manhattan distance in space and time. Additionally, depth information as well as local density of address events is used for noise suppression. Cluster form and size are determined by the radial dilation factor which depends on the density of the assigned address events.

The mentioned algorithms are well suited for embedded vision systems due to low memory usage and real-time address events assignment. However, the proposed clustering procedures are dependent on experimentally adjusted parameters (i.e. object size limits) that are specific for a particular application. What is more, the problem of occlusions has not been addressed yet in the domain of address events processing and that is the main motivation for this work.

3. Methodology

In our method we investigate the possibility to overcome the problem of tracking multiple objects in the occurrence of severe occlusions. As proved in previous attempts [8] [13], clustering performs well in processing address events. Therefore, we focus on extending the clustering task by using the Gaussian Mixture Models (GMM) [6].

In this section we introduce the fundamentals of the Gaussian Mixture Models and their applicability to the problem of clustering. Then, we provide a discussion on the choice of the method and the benefits we can get from modeling objects with Gaussians.

3.1. Gaussian Mixture Models

A mixture model is a probabilistic model that assumes that underlying data belong to a mixture distribution. A mixture distribution is a linear combination of the probability density functions of its components:

$$p(x) = w_1 p_1(x) + w_2 p_2(x) + \ldots + w_n p_n(x), \qquad (1)$$

where n is the number of mixture components $p_{1\ldots n}(x)$ and $w_{1\ldots n}$ are the mixture weights, also called coefficients. As the combination is convex, the mixture coefficients should sum to unity. A Gaussian Mixture distribution assumes that the mixture components are Gaussian density functions, as follows:

$$p(x) = w_1 N(x|\mu_1, \Sigma_1) + \ldots + w_n N(x|\mu_n, \Sigma_n). \qquad (3)$$

The complete GMM is defined by parameters of each Gaussian component: the weight, mean vector and covariance matrix:

$$\lambda = \{w_i, \mu_i, \Sigma_i\}_{i=1..n} \qquad (4)$$

Suppose we have set of unlabeled data X, so the goal is to find such a Gaussian Mixture Model that best matches the distribution of X. The problem of estimating the GMM parameters is solved by the Expectation Maximization (EM) algorithm. The idea behind the EM method is to introduce a latent variable λ which could simplify the maximization of GMM likelihood given training set X:

$$p(X\,|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda) \qquad (5)$$

Thus, the algorithm estimates which mixture component generated a particular data point and iteratively changes the model parameters to maximize the likelihood between the sample and the corresponding Gaussian distribution.

The EM algorithm consists of two steps:

(1) Expectation, when given GMM parameters are evaluated for each sample from the d-dimensional dataset $x \in \{X\}_{t=1\ldots T}$ by calculating the a posteriori probability for i^{th} component by formula:

$$P(i|x_t, \lambda) = \frac{w_i g(x_t|\mu_i, \Sigma_i)}{\sum_{k=1}^{n} w_k g(x_t|\mu_k, \Sigma_k)}\ , \qquad (6)$$

where $g(x_t|\mu_i, \Sigma_i)$ is defined as:

$$g(x_t|\,\mu_i, \Sigma_i) = \\ = \frac{1}{(2\pi)^{\frac{d}{2}}\,|\Sigma_i|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x_t - \mu_i)'\Sigma_{\mathrm{i}}^{-1}(x_t - \mu_i)\} \qquad (7)$$

(2) Maximization, when the parameters are re-estimated according to the a posteriori probability calculated in previous step. Following formulas are used to calculate new GMM parameters:

$$\overline{\mathrm{w}}_{\mathrm{i}} = \frac{1}{\mathrm{T}} \sum_{\mathrm{t}=1}^{\mathrm{T}} \mathrm{P}(\mathrm{i}|\mathrm{x}_{\mathrm{t}}, \lambda) \qquad (8)$$

$$\overline{\mu}_{\mathrm{i}} = \frac{\sum_{\mathrm{t}=1}^{\mathrm{T}} \mathrm{P}(\mathrm{i}|\mathrm{x}_{\mathrm{t}}, \lambda)\, \mathrm{x}_{\mathrm{t}}}{\sum_{\mathrm{t}=1}^{\mathrm{T}} \mathrm{P}(\mathrm{i}|\mathrm{x}_{\mathrm{t}}, \lambda)} \qquad (9)$$

$$\overline{\sigma}_{\mathrm{i}}^2 = \frac{\sum_{\mathrm{t}=1}^{\mathrm{T}} \mathrm{P}(\mathrm{i}|\mathrm{x}_{\mathrm{t}}, \lambda)\, \mathrm{x}_{\mathrm{t}}^2}{\sum_{\mathrm{t}=1}^{\mathrm{T}} \mathrm{P}(\mathrm{i}|\mathrm{x}_{\mathrm{t}}, \lambda)} - \overline{\mu}_{\mathrm{i}}^2 \qquad (10)$$

Steps are repeated until reaching the convergence, which could be controlled by likelihood improvement threshold or maximum number of iterations.

3.2. Discussion

The choice of clustering method was crucial in the proposed approach, as tracking mainly relies on the clustering results. There are several reasons why we decided to choose clustering with Gaussian Mixture Models.

First of all, this method is well-suited to the type of data we are dealing with. Since the address events are generated asynchronously, we obtain valuable information about the captured motion such as intensity of the events. This information is used while estimating the parameters of a GMM that is based on density of samples from the dataset. Figure 2 illustrates how the intensity of the events is modeled by Gaussian Mixtures. Secondly, the distance-based clustering methods usually assume hard boundaries of the clusters. In the occurrence of high occlusions, the resulting motion paths are overlapping. That is why we benefit from GMM soft cluster boundaries.

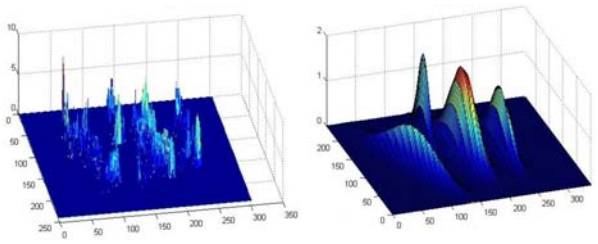

Figure 2: The intensity of the events generated in 20ms by moving persons (top). On the bottom, the same data modeled with Gaussian Mixtures.

Furthermore, using a distance similarity measure to assign events to clusters may presume circularity of a cluster's shape. However, this representation is not practical in most of the real world applications. Preferably, we suggest Gaussian distributions as a better representation of tracked objects. In that approach modeled clusters have a center of gravity specified by the mean vector while the size or shape is represented by the covariance matrix.

4. Proposed Algorithm

The objective of the tracking algorithm is to retrieve trajectories of the objects moving in the scene. In address event data representation the motion of the object can be observed as a cloud of events in space-time. Therefore events generated by object forms clusters that can be modeled by Gaussian Mixtures (Figure 3). Tracking can be seen as clustering incoming events by finding the best model parameters for underlying data. Events are represented by 3 dimensional vectors of spatial coordinates:

$$ev = (x, y, z), \tag{11}$$

Once the model is initialized, the events are assigned to the clusters by maximum a posteriori probability calculated according to the Formula (7).

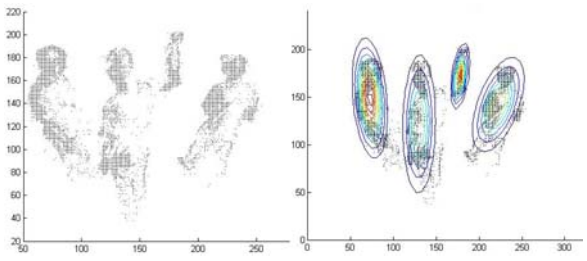

Figure 3: The left picture depicts the scene in address events representation. In addition, the right image shows the Gaussian distributions of the objects in the scene. Each Gaussian mixture component represents cluster.

Clusters evolve in time due to the objects' motion, and that is why the model should be updated. The update procedure is performed in particular time steps to collect a representative amount of data for better model estimation. As mentioned before, an efficient approximation of the model parameters can be obtained by the Expectation Maximization algorithm. Although EM accurately estimates the GMM parameters, it is very sensitive to new data, which can lead to tracking errors. For instance in situations when the objects are not well represented by events the model is fitted to data but not to the objects. That is why, the Maximum a Posteriori (MAP) method [9] is used for the clusters update, so the temporal smoothness constraint is satisfied. The EM algorithm is used only for significant configuration changes (entrance, exit) or when the likelihood of the model drops drastically.

The first part of the update with MAP is analogical to the Expectation step in EM, in which the prior model is tested for the data collected in the last 10ms. The expected parameter values for the i[th] component are calculated as follows:

$$\mathrm{n_i} = \sum_{\mathrm{t=1}}^{\mathrm{T}} \mathrm{P}\left(\mathrm{i}\middle|\mathrm{x_t}, \lambda_{\mathrm{prior}}\right) \tag{12}$$

$$\mathrm{E_i(x)} = \frac{1}{\mathrm{n_i}} \sum_{\mathrm{t=1}}^{\mathrm{T}} \mathrm{P(i|x_t}, \lambda_{\mathrm{prior}})\, \mathrm{x_t} \tag{13}$$

$$\mathrm{E_i(x^2)} = \frac{1}{\mathrm{n_i}} \sum_{\mathrm{t=1}}^{\mathrm{T}} \mathrm{P(i|x_t}, \lambda_{\mathrm{prior}})\, \mathrm{x_t^2} \tag{14}$$

Then, the prior model is adapted to new data using the values from the expectation step:

$$\widehat{\mathrm{w}}_\mathrm{i} = [\alpha_i^w n_i + (1 - \alpha_i^w) w_i]\, \gamma \tag{15}$$

$$\hat{\mu}_i = \alpha_i^m E_i(x) + (1 - \alpha_i^m)\mu_i \tag{16}$$

$$\widehat{\sigma}_i = \alpha_i^s E_i(x^2) + (1 - \alpha_i^s)(\sigma_i^2 + \mu_i^2) - \hat{\mu}_i^2 \tag{17}$$

The coefficients $\{\alpha_i^w, \alpha_i^m, \alpha_i^s\}$ control the balance between the old and new model parameters and their values depend on the expected weights of the mixtures. The γ coefficient in Formula (15) is used to ensure that weights sum to unity. The higher amount of new events is assigned to the cluster, the more it will be changed in the adaptation step. Additionally, in the task of multi-object tracking we need to handle special situations, namely the entrance of a new object, exit, stop and occlusion. A new cluster for the object is created when a significant amount of events do not fit to any cluster.

There are three possible reasons why an object disappears, either it can exit the scene, stop or be occluded. If an object is absent for more than one second, we consider it as an exit. In order to detect occlusions, we first measure the distance between the clusters and then calculate the likelihood of pairs of overlapping clusters for a given data. Occlusion is indicated by a high amount of events which can be assigned to both clusters with similar probabilities. The clusters marked as occluded have low weights, so their update mostly relies on the prior model. Therefore the possible confusion with event assignment is solved as the higher weight increases the probability of non-occluded component.

4.1. Experimental Results

The accuracy of the algorithm cannot be extensively measured because there is no ground-truth available for DVS data sequences. That is why our results are based on the limited test data which were labeled manually. Although the test sequences are not large, they include different case scenarios to check how robust the algorithm is dealing with occlusions, entrances and stops.

According to [15] there are three key properties of the multi-object tracking that need to be evaluated, namely: configuration, identification and speed. The configuration refers to the correct estimation of location and number of objects in the scene. In order to check how well the objects are located in comparison to ground truth, we performed the coverage test for 180 correctly detected objects E and corresponding ground truth GT. The results in Table 1 show that our method precisely describes objects, as both mean precision (v) and recall (ρ) are high.

Furthermore, the algorithm should automatically start tracking an object on the move, and in our case this is supported by the sensor as only the dynamics is being captured.

Thus, all incoming data correspond to the detected movement. What is more, the number of clusters is determined dynamically by outlier detection and component weights. In most cases the proposed algorithm can correctly detect the entrance of a new object.

TABLE 1: RESULTS OF THE COVERAGE TEST.

Measure	Formula	Mean value
Recall	$\rho = \frac{\lvert E \cap GT \rvert}{\lvert GT \rvert}$	0.9862
Precision	$v = \frac{\lvert E \cap GT \rvert}{\lvert E \rvert}$	0.7872

However, we observed that our algorithm fails to distinguish objects which enter the scene at approximately the same time and which are close to each other. The entrance test scenario is shown in the Figure 4.

The second property measures the accuracy of object identification, which is tracker to object assignment. The most important advantage of DVS is that it captures almost “continuous” motion in frame-free mode due to high temporal resolution. Therefore, there is no need to connect tracker to object as corresponding clusters are just propagated in time. However, it is not trivial to recover from the identification error because we cannot use any appearance features to differentiate objects. Figure 5 shows the results of tracking dealing with occlusions.

Embedded vision systems require algorithms to be low in memory usage and computational cost as well. Although, the Gaussian Mixture clustering is more complex in comparison to [8] and [13], it still remains fast and can operate in real-time.

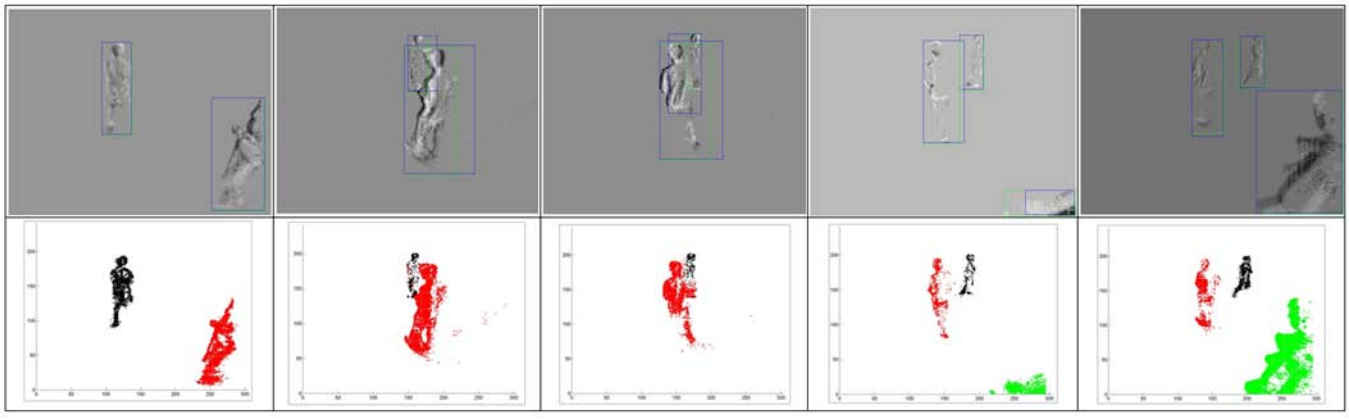

Figure 4: Part of the test sequence where three people enter the scene. Images on the top show the result of tracking by bounding boxes in green (ground truth) and blue (result of the algorithm). At the bottom, the identified objects are differentiated by colors.

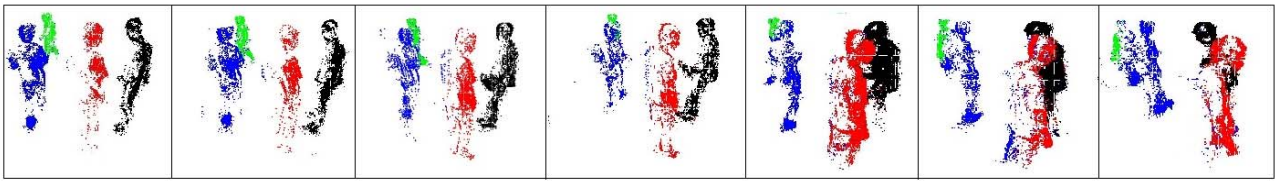

Figure 5: Results of the tracking algorithm in the occurrence of high occlusions. It can be observed, that our algorithm successfully detects objects even when the clusters overlap.

5. Conclusions

This paper presents an algorithm for multiple persons tracking in the occurrence of high occlusions. Although a large number of different tracking algorithms have been proposed in literature, only few attempts have been made to track objects using DVS.

There is a significant difference between tracking using conventional vision systems and the Dynamic Vision Sensor. One of the advantages of using DVS is capturing of the scene dynamics and segmentation of the moving object from the static background that is done on-chip. Nonetheless, the task of multiple objects tracking still remains challenging, especially when dealing with occlusions. Since the sensor supports the task of tracking, there is a strong motivation to keep the tracking algorithm as simple as possible to maintain efficiency of the processing but still achieve accurate results. Hence, we investigated the possibility to use Gaussian Mixture models, not only to detect objects but also to track their evolution in time. Preliminary results indicate that applying Gaussian Mixture Models to spatiotemporal DVS data has a large potential for efficient and robust object tracking.

Acknowledgements

This work was supported by the project “CHANGES” of the program FEMtech funded by Austrian Federal Ministry for Transport, Innovation and Technology.

References

[1] A.N. Belbachir, “Smart Cameras,” Springer, New York, 2009

[2] T. Broida and R. Chellapa, “Estimation of Object Motion Parameters from Noisy Images,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, issue 1, pp 90–99, 1986

[3] Y. Chen, Y. Rui and T. Huang, “JPDAF Based HMM for Real-time Contour Tracking,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp 543–550, 2000

[4] D. Comaniciu, V. Ramesh and P. Meer, “Kernel-based Object Tracking,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, pp 564–575, 2003

[5] P. Fieguth and D. Terzopoulos, “Color-based Tracking of Heads and Other Mobile Objects at Video Frame Rates,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp 21–27, 1997

[6] M.A.T. Figueiredo, A.K. Jain, "Unsupervised Learning of Finite Mixture Models," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 3, pp 381-396, 2002

[7] P. Lichtsteiner, C. Posch and T. Delbrück, “A 128×128 120dB 15us Latency Asynchronous Temporal Contrast Vision Sensor,” IEEE Journal of Solid State Circuits, vol. 43, pp. 566 - 576, 2008

[8] M. Litzenberger, C. Posch, D. Bauer, A.N. Belbachir, P. Schön, B. Kohn, and H. Garn, “Embedded Vision System for Real-Time Object Tracking Using an Asynchronous Transient Vision Sensors,” 12th IEEE Workshop on Digital Signal Processing and Signal Processing Education DSP/SPE 2006, in John Pierre and Cam Wright (Eds), pp. 173-178, Wyoming, USA September 2006

[9] D. A. Reynolds, “Gaussian Mixture Models,” Encyclopedia of Biometric Recognition, Springer, February 2008.

[10] V. Salari and I. K. Sethi, "Feature Point Correspondence in the Presence of Occlusion," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 1, pp.87–91, 1990

[11] J. Shi and C. Tomasi, “Good Features to Track,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp 593–600, 1994

[12] S. Schraml, A.N. Belbachir, N. Milosevic and P. Schön, “Dynamic Stereo Vision for Real-time Tracking,” in Proc. of IEEE International Symposium on Circuits and Systems, 2010

[13] S. Schraml, A.N. Belbachir, “A Spatio-temporal Clustering Method Using Real-time Motion Analysis on Event-based 3D Vision,” in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Three Dimensional Information Extraction for Video Analysis and Mining, San Francisco, 2010

[14] S. Schraml, N. Milosevic and P. Schön, “Smartcam for Real-Time Stereo Vision - Address-Event Based Stereo Vision,” in Proc. of Computer Vision Theory and Applications, INSTICC Press, pp. 466 – 471, 2007

[15] K. Smith, D. Gatica-Perez, J.M. Odobez, S. Ba, “Evaluating Multi-Object Tracking,” in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Workshop on Empirical Evaluation Methods in Computer Vision, San Diego, June 2005.

[16] S. Zhou, R. Chellapa and B. Moghaddam, “Visual Tracking and Recognition Using Appearance Adaptive Models in Particle Filters,” in IEEE Transactions on Image Processing vol. 11, pp 1434–1456, 2004