Gaussian Mixture Models and Split-Merge Algorithm for Parameter Analysis of Tracked Video Objects

GuoQing Yin and Dietmar Bruckner

Institute of Computer Technology
Vienna University of Technology, Austria, Europe
{yin, bruckner}@ict.tuwien.ac.at

***Abstract*- Parameters of tracked video objects (for example: the angles of moving objects) are discrete random variables and the amount of data increases over time. In this paper we use a new method to analyze the parameter angle: the video frame is segmented into small sections and in each section the angle values during some time period are gathered. Through analysis the angle data in each section these angles can be modeled, therefore also in whole frame. The build model will be used to find abnormal behavior of moving objects.**

To build a statistical model of the angle of moving objects from the video data is a question of cluster analysis in real time. For this application, Gaussian Mixture Models and Split-Merge Algorithm provide a powerful solution.

Keywords: Gaussian Mixture Models, Split-Merge Algorithm, Object Tracking, Parameter Analysis, Clustering, Real Time Analysis.

I. INTRODUCTION

There are many papers about cluster analysis: an overview of cluster Analysis is given by [1], there many types of clustering and criteria are discussed, [2] deals with the problem of determining the structure of clustered data, without prior knowledge of the number of clusters or any other information about their composition, [3] contributed methods for normal mixture modeling and model-based clustering. The authors of [4] stated and proved an estimate on measure concentration for the problem of cluster analysis posed as a suitable integer linear program, [5] propose a self-splitting Gaussian mixture learning (SGML) algorithm for Gaussian mixture modeling with application to speaker identification, [6] present a split and merge EM algorithm to overcome the local maximum problem in Gaussian mixture density estimation. In [7], the authors developed a new methodology for fully Bayesian mixture analysis, using reversible jump Markov chain Monte Carlo methods to jumping the parameter subspaces and the different numbers of components in the mixture, while [8] propose a new kind of dynamic merge-or-split learning (DMOSL) algorithm to deal with the selection of number of Gaussians in the mixture, [9] introduced a split-and-merge operation in order to alleviate the problem of local convergence of the usual EM algorithm. In [10] and [11] authors use cluster analysis methods for video image analysis, in particular, in [10] they describe HMM-based clustering for learning motion patterns over time and to detect abnormal activity in a video surveillance scene; [11] present a novel dynamical Gaussian mixture model (DGMM) for tracking elliptical living objects in video frames, the parameters about the objects position and shape will be analyzed using GMM; especially [12] propose novel methods to evaluate the performance of object detection algorithms in video sequences, it deals with region splitting or merging. However, there is no paper using this method to analyze the parameter *angle* of tracked video objects and to build statistical models of the data.

1.1 Project SENSE background

This work is part of the SENSE[1] [13] project. The SENSE (Smart Embedded Network of Sensing Entities) project is a European FP6 project about security for public spaces (see project website). The project will develop methods, tools and a test platform for the design, implementation and operation of smart adaptive wireless networks of embedded sensing components. A test platform for a civil security monitoring system will be developed as a test application, composed of video cameras and microphones. The test platform will be installed in an airport, to yield real data and performance goals from a realistic test environment.

The SENSE network adapts to its environment, learns (unsupervised learning) the "normality" in the environment in order to detect unusual behavior (e.g. people "lurking" in an area, people go in a wrong direction), situations (e.g. baggage left unattended) and to inform the user [14], [15]. The relevant parts of the project here are concerned with video surveillance systems [16] and parameter inference. This paper focuses on video data parameter inference. It uses Gaussian Mixture Models and Split-Merge Algorithm to analysis the parameter of tracked video objects and builds statistical

[1] This work is partially funded by the European Commission under contract No. 033279.

models. The statistical models are used to find abnormal parameters, which points to unusual behavior of persons.

1.2 Video objects, parameters, section, cluster, angle

In the video modality of the SENSE system, objects (O) are detected. This can be e.g. persons, group of persons, luggage, or unclassified objects. Each object is surrounded by a bounding box in pixel coordinates. The input data for post processing consists of: frame number (F, from 0 to 9 fps; for each second) and time (T, second), ID of objects (L, from 1 to n; video processing has a tracking mechanism implemented), the object's position (X, Y) in pixel, and its width (W) and height (H). These will be later called basic input data set. From this basic input data set we can get the different parameters, for example: using ID of objects and object's position (X, Y) we can get each moving object's angle (A_{object}).

If we segment the video frame (480x640 pixels) in smaller pixels sections (32x32 pixels), we get 15x20 smaller sections for the whole frame. When objects move in some of the sections, each section gets different angle of the respective moving objects (A_{object}). With increased time each section ($S_{i,j}$, here i and j are index for the sections position at frame in Cartesian coordinate system) will gather a data set of angles $\{A_{(O1,T1)}, A_{(O2,T1)} \ldots A_{(On,T1)}, A_{(O1,T2)}, A_{(O2,T2)} \ldots A_{(On,T2)} \ldots A_{(On,Tm)}, n \geq 0$ and $m > 0\}$. For each section a model is constructed which represents the characteristics of the parameter for this section. Using the model and comparing it with new values after the learning phase we can find abnormal angles, therefore abnormal moving objects' directions and unusual object behavior. Build an angle model in this situation is a mathematic question of cluster analysis. From Gaussian Mixture Models and Split-Merge Algorithm we can get useful and affordable results.

II. Gaussian Mixture Models And Merge-split Algorithm

This section deals with the mathematical background of the algorithms.

2.1 Gaussian mixture model

A standard Gaussian function is defined as

$$\varphi_{\mu,\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1)$$

Here μ is the expected value (mean value of the clustering angle μ_A), σ is the standard deviation of each angle cluster (σ_A). x is the angle value (A).

Before we use split-merge algorithm for clustering the angle set, we have to define the range of the parameters: $0 \leq \mu_A \leq 360$, $50 \leq \sigma_A \leq 100$, $0 \leq A \leq 360$. The number of initial components S = 3, that means there are sets $\{\mu_{A1}, \mu_{A2}, \mu_{A3}; \sigma_{A1}, \sigma_{A2}, \sigma_{A3}; P_{A1} P_{A2} P_{A3}\}$. Here P is the percent value of each angle component and $\sum P_{(A1,A2,A3)} = 1$ and these prior parameters are random variables. Threshold value for split, merge and delete components: $\mu_{threshold}$, $\sigma_{threshod}$, $\sigma_{threshod2}$, $P_{threshold}$. The maximum number of angles for adjusting the learning rate is T and current angle count is T'. Each new observation angle that gets into the angle set is A_r ($r \geq 1$). The index *s* is the component index within the mixture model.

With these parameters and definition we can begin to cluster the angle set with split-merge algorithm.

2.2 Split-Merge Algorithm

- Compute and then normalize posteriors

$$P_s(A_r) = P_s \cdot \varphi_{\mu,\sigma^2}(A_r);\ P_s(A_r) = P_s(A_r) / \sum P_s(A_r) \quad (2)$$

- Compute new means

$$\mu_{A,s} = (1 - P_s(A_r)) \cdot \mu_{A,s} + P_s(A_r) \cdot (T' \cdot \mu_{A,s} + A_r)/(T'+1) \quad (3)$$

- Compute new variances

$$\sigma_{A,s} = ((1 - P_s(A_r)) \cdot \sigma_{A,s} + P_s(A_r) \cdot (T' \cdot \sigma_{A,s} + |\mu_{A,s} - A_r|)/(T'+1) \quad (4)$$

- Compute new priors

$$P_s = (T' \cdot P_s + P_s(A_r)) / (T' + 1) \quad (5)$$

- Keep the learning rate and adaptability

$$\text{If } T' \geq T,\ T' = T \quad (6)$$

- After some initial iterations, start checking if it is necessary to split components: If $\sigma_{A,s} > \sigma_{threshod}$, then create new component (index S) from old component (index s)

$$\mu_{A,S} = \mu_{A,s} + \sigma_{A,s} / 2;\ \mu_{A,s} = \mu_{A,s} - \sigma_{A,s} / 2 \quad (7)$$

$$\sigma_{A,S} = \sigma_{A,s} / 2;\ \sigma_{A,s} = \sigma_{A,s} / 2 \quad (8)$$

$$P_S = P_s = P_s / 2 \quad (9)$$

- If necessary, merge components (s' and s'')

If ($|\mu_{A,s'} - \mu_{A,s''}| < \mu_{threshold}$ and $|\sigma_{A,s'} - \sigma_{A,s''}| < \sigma_{threshold2}$) then merge component s'' into s' and delete component s''. Here $\sigma_{threshold2} <= \sigma_{threshod}$.

$$\mu_{A,s'} = (\mu_{A,s'} \cdot P_{A,s'} + \mu_{A,s''} \cdot P_{A,s''}) / (P_{A,s'} + P_{A,s''}) \quad (10)$$

$$\sigma_{A,s'} = \max(\sigma_{A,s'}, \sigma_{A,s''}) \quad (11)$$

$$P_{A,s'} = P_{A,s'} + P_{A,s''} \quad (12)$$

- If any component's prior decreased too much so that $P(s') < P_{threshold}$ then delete the component and adjust the other priors P(s):

$$P(s) = P(s) / \sum P(s) \tag{13}$$

- Repeat with all new values.

III. Result And Conclusion

3.1 Result

The first example is just random angle value shown in Fig. 1.

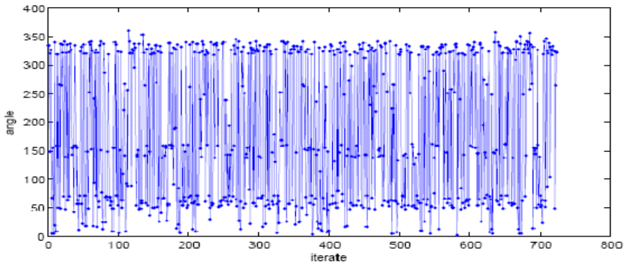

Fig. 1. Test angle data for the algorithms, random value.

Fig. 2 shows the learned result (the green line) with the histogram together. In Fig. 2 we can see that all the important angle group are found through learning.

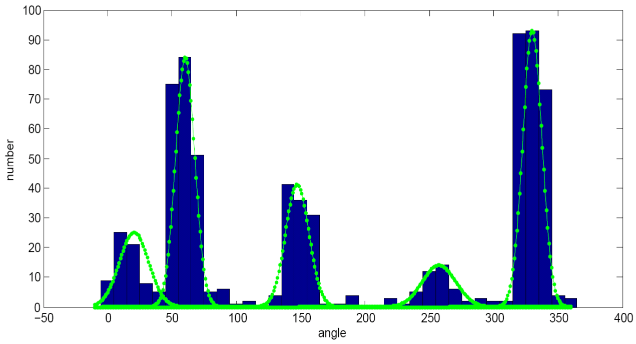

Fig. 2. To compare the learn result with the histogram

Above is the learning result with random angle data (it happens only in one data sets), in practical we use the algorithm for SENSE project, for object moving direction analysis. As we said before we gather the objects moving direction as angle and clustering (using split and merge algorithm) these angles, from the learning result we know in where (15x20 sections, that means there will be 15x20 data sets) how the objects with what kind of direction moved.

Fig. 3 shows the learning result from a real video data, the video duration is 207.3 seconds and there are 2074 frames (pictures, 10 frames each second) and totally 924 different objects moved in the frames everywhere with different direction and duration, from 0.1 seconds till minutes long.

In Fig. 3 shows the 3 main directions for each smaller section (there are 15x20 sections in the whole frame). Most of the angles are split and merged in three groups: between 0 to 100 degree, nearly 200 degrees and between 250 to 350 degree. The std value for each section is smaller than 20 degree and only at the left top and right bottom location there are bigger std value, that because in these location are seldom objects there and the few objects moving angles are not much enough to make a reliable learning result and the std value stay at bigger before the split and merge begin (T'<T).

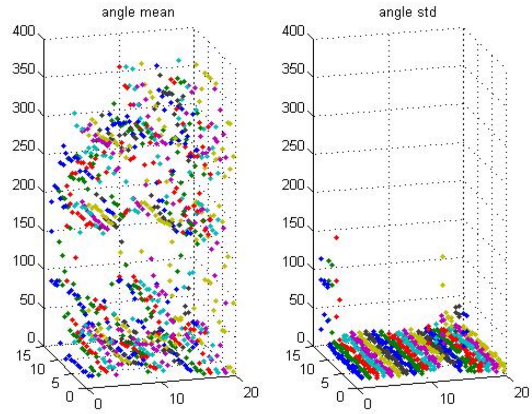

Fig. 3. The learned result of each section.

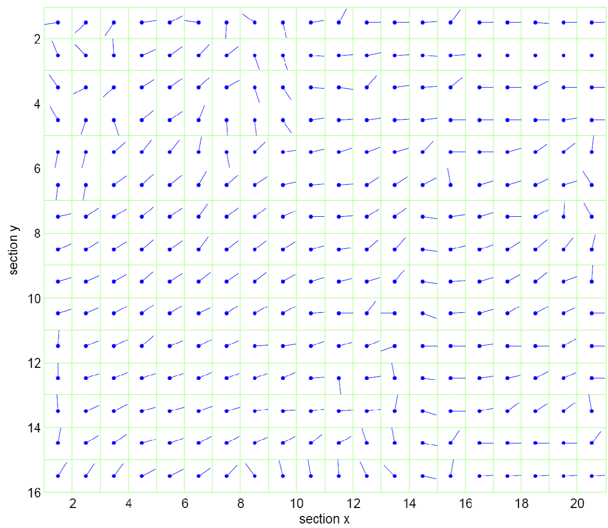

Fig. 4. The most likely learned result of each section as vector.

Fig. 4 shows the most important objects moving direction (with the maximum percent value; so to say the main direction for each cluster) for each section, from it we can observe the general moving situation of objects in all the video data.

For a comparison we run the same video data again but with different beginning mean, std, and percent value ($\{\mu_{A1}, \mu_{A2}, \mu_{A3}; \sigma_{A1}, \sigma_{A2}, \sigma_{A3}; P_{A1}\, P_{A2}\, P_{A3}\}$, all these value are random variables), the result is shown in Fig. 5 and Fig. 6.

At first we compare Fig. 3 with Fig. 5: the learned angle mean value is nearly the same at the same section and the angle std value only at the left top and right bottom location has some different value. From the result we can say that the learning algorithm split and merge the same angle data and get the same result in spite of with different random initial value.

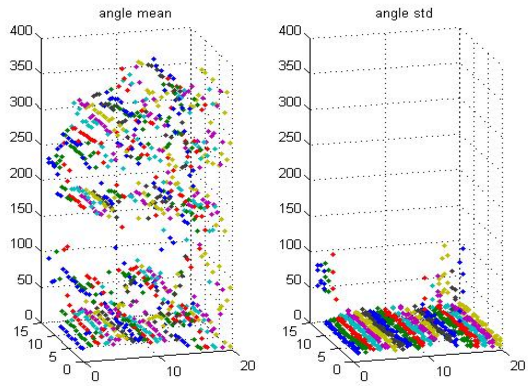

Fig. 5. The learn result of each section.

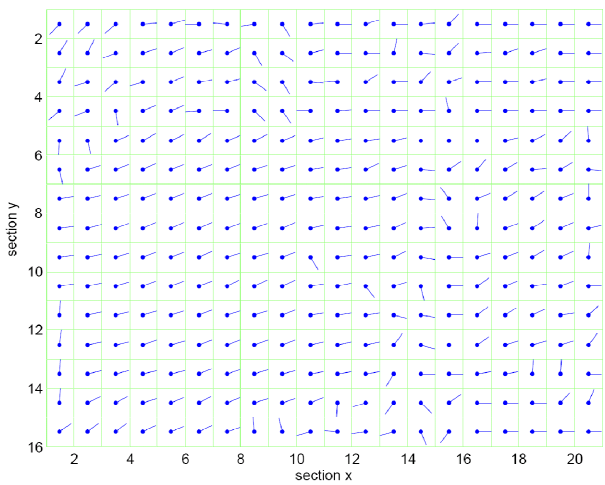

Fig. 6. The learn result of each section.

Second we compare Fig.4 and Fig. 6: at most of the section there are nearly the same directions, only at the big std value location (at the left top and right bottom) there are different angle directions.

From the comparison we can be sure that the built model from the real video angle data is reliable. With this model we can find the unusual object's moving directions. That means in some sections if object's moving directions mismatch the build model, these objects should be marked with "unusual behavior".

3.2 Conclusion

From above result we can say that Gaussian Mixture Model and the split and merge algorithm is a powerful tool for unsupervised clustering and very useful in practical. It is important that enough learning data is provided, ensuring a reliable learning result, avoiding the result to be affected by the initial value.

As a real-time learning algorithm generally it needs less computing time than the real video duration.

IV. OUTLOOK

In the future we have to investigate how the parameters influence the learning result. For example with different $\mu_{threshold}$ $\sigma_{threshold}$, $P_{threshold}$, iterate T Parameter: how they influence the learning process.

REFERENCES

[1] Pierre Hansen, Brigitte Jaumard (1997). "Cluster analysis and mathematical programming", in Mathematical Programming 79 191.215.

[2] C. Fraley and A. E. Raftery (1998). "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis", in Technical Report No. 329, Department of Statistics, University of Washington.

[3] Chris Fraley and Adrian E. Raftery (2006). "MCLUST Version 3 for R: Normal Mixture Modeling and Model-Based Clustering", in Technical Report No. 504, Department of Statistics, University of Washington.

[4] Daniel Z. Zanger (2003). "Concentration of measure and cluster analysis" in Statistical & Probability Letters 65 65-70.

[5] Shih-Sian Cheng, Hsin-Min Wang, Hsin-Chia Fu (2004). "A Model-Selection-Based Self-Splitting Gaussian Mixture Learing with Application to Speaker Identification", in EURASIP Journal on Applied Signal Processing 17, 2626-2639.

[6] Naonori Ueda, Ryohei Nakano, Zoubin Ghahramani, Geoffery E. Hinton (1998). "SPLIT AND MERGE EM ALGORITHM FOR IMPROVING GAUSSIAN MIXTURE DENSITY ESTIMATES", 0-7803-5060-X/98, IEEE 274.

[7] SYLVIA RICHARDSON, PETER J. GREEN (1997), "On Bayesian Analysis of Mixtures with an Unknown Number of Components", in J. R. Statist. Soc. B 59, No. 4, *pp*. 731-792.

[8] Jinwen Ma and Qicai He (2005). "A Dynamic Merge-or-Split Learning Algorithm on Gaussian Mixture for Automated Model Selection", Department of Information Science, School of Mathematical Science and LMAM, Peking University, Beijing, 100871, China.

[9] Zhihua Zhang, Chibiao Chen, Jian Sun, Kap Luk Chan (2003). "EM algorithms for Gaussian mixture with split-and-merge operation" in Pattern Recognition 36 1973-1983.

[10] Eran Swears, Anthony Hoogs and A.G. Amitha Perera, Kitware Inc (2008). "HMM-Based Clustering for Learning Motion Patterns in Surveillance Video".

[11] Guanglei Xiong, Chao Feng, Liang Ji (2006). "Dynamical Gaussian mixture model for tracking elliptical living objects", in Pattern Recognition Letters 27 838-842.

[12] Jacinto C. Nascimento and Jorgs S. Marques (2006). "Performance Evaluation of Object Detection Algorithms for Video Surveillance", in IEEE TRANSACTIONS ON MULTIMEDIA, VOL 8, NO, 4.

[13] http://www.sense-ist.org (SENSE project website).

[14] D. Bruckner, J. Kasbi, R. Velik, and W. Herzner (2008). "High-level Hierarchical Semantic Processing Framework for Smart Networks", In: Proceedings of the 2008 IEEE Conference on Human System Interaction (HSI), S. 6, Krakow.

[15] B. Sallans, D. Bruckner, and G. Russ (2006). "Statistical Detection of Alarm Conditions in Building Automation Systems", In: Proceedings of 2006 IEEE International Conference of Industrial Informatics INDIN´06, S. 6, Singapore.

[16] D. Bruckner, B. Sallans, and G. Russ (2007). "Hidden Markov Models for Traffic Observation", In: Proceedings of 2007 IEEE International Conference on Industrial Informatics INDIN07, S. 1015 - 1020.