UNSUPERVISED LOCATION-BASED SEGMENTATION OF MULTI-PARTY SPEECH

G. Lathoud, I.A. McCowan and J.M. Odobez

{lathoud,mccowan,odobez}@idiap.ch
IDIAP, Martigny, Switzerland

ABSTRACT

Accurate detection and segmentation of spontaneous multi-party speech is crucial for a variety of applications, including speech acquisition and recognition, as well as higher-level event recognition. However, the highly sporadic nature of spontaneous speech makes this task difficult. Moreover, multi-party speech contains many overlaps. We propose to attack this problem as a tracking task, using location cues only. In order to best deal with high sporadicity, we propose a novel, generic, short-term clustering algorithm that can track multiple objects for a low computational cost. The proposed approach is online, fully deterministic and can run in real-time. In an application to real meeting data, the algorithm produces high precision speech segmentation.

1. INTRODUCTION

Segmentation of spontaneous multi-party speech, as found in meetings, is a hard task for two main reasons. The first reason is that speech from each person is sporadic: each speaker is silent most of the time, including many silences between utterances. Each utterance can be very short, however we still want to capture all of them. For an extreme example: if over an entire meeting a person says only one word "yes", as an answer to "do you approve next year's budget?", we still want to capture that answer. The second reason is the "multi-party" effect: overlaps occur as a non-negligible proportion of speech. In [1], it was estimated that around 10-15% of words and 50% of contiguous speech segments contain some degree of overlapping speech. In spite of these two difficulties, the rewards of successfully segmenting spontaneous multi-party speech can be high for applications based on it. For example, in a meeting browser, it would be very useful for a person to quickly access the part of the meeting they are interested in. Generally, automatic, higher-level analysis of large amounts of data can benefit greatly from precise speech segmentation.

In previous work [2, 3] we showed the interest of using microphone arrays to segment multi-party speech, producing one speech/silence segmentation for each speaker location. High segmentation performance was obtained, including on overlapped speech. However, the set of locations was assumed fixed and known *a priori*. We propose here to remove this assumption, by building an application based on a novel short-term clustering algorithm.

Assuming we have audio source location information, we can define the segmentation problem as a tracking task, i.e. to use location cues alone to detect and segment the various speech events. Tracking can be viewed as a filtering task: a lot of extremely valuable work has been done along this line: Kalman Filtering [4] and its variants [5, 6, 7], and Particle Filters [8, 9] are two examples of such approaches. However, the higly sporadic and concurrent nature of the speech events we need to track leads to data association issues. Although Particle Filters can model multiple objects via multi-modal distributions, deciding which modes are significant and which objects they belong to is an open issue. Moreover, when the number of active objects varies very often along time, complex birth/death rules are needed. This by no means reduces the interest of filtering approaches for modalities where each event is observable over relatively long durations of time, such as radar and video. In the "sporadic" context involved here, we propose a novel, threshold-free short-term clustering algorithm that is online and fully deterministic. It does not require any random sampling. We also report investigations on synthetic data involving multiple objects and trajectory crossings.

In the meeting segmentation task presented here (on a publicly available database), the new scheme achieves results that compare well with a lapel baseline system, which instead uses energy from lapel microphones. Particularly good performance is obtained on overlapped speech.

The aim of this paper is twofold: to present the novel, generic, short-term clustering algorithm, as well as an application to multi-party speech segmentation. The application is the focus of this paper, therefore the algorithm is presented in a summarized manner. For full details the reader is invited to read [10].

The rest of this paper is organized as follows: Section 2 summarizes the short-term clustering algorithm, and presents multi-object tracking examples on synthetic data. Section 3 presents the meeting segmentation application, along with results on the meeting corpus. Section 4 discusses the results and openings for future work, and Section 5 concludes.

2. SHORT-TERM SPATIO-TEMPORAL CLUSTERING

In this section we present the proposed short-term spatio-temporal clustering approach. As the focus of this paper is the meeting segmentation application (Section 3), the theory presented here is summarized, and the interested reader is refered to [10] for complete details.

Throughout this paper the context will be one microphone array on a table, recording multi-party speech in a meeting room. It is used to provide audio source location information. For each time frame t, there can be zero, one or multiple location estimates. Although the algorithm presented here is applicable to variable number of location estimates, in the rest of this paper we restrict the context to exactly one location estimate per time frame t. Within each time frame t, the location estimate θ is the azimuth of the dominant sound source. These location estimates may be obtained using standard acoustic source localization techniques (see [11] for a recent review).

2.1. Assumption on Local Dynamics

Let $X_i = (\theta_i, t_i)$ for $i = 1 \dots N$ be all instantaneous location estimates of events emitted by the various objects. This includes the desired events (speech sounds) as well as noise. $\theta_i \in \mathbb{R}^D$ is a location in space, while $t_i \in \mathbb{N} \setminus \{0\}$ is a time frame index: $t_i \in (1, 2, 3, \dots)$.

The notation p designates a probability density function (pdf) or likelihood. The notation P designates a probability or a posterior probability. The notation $1 : n$ designates the vector of integers $(1, 2, \dots, n)$.

For any pair of location estimates (X_i, X_j) such that $i < j$, we define the hypotheses:

- $H_0(i,j) \triangleq$
  "X_i and X_j correspond to **different** objects"

- $H_1(i,j) \triangleq$
  "X_i and X_j correspond to the **same** object"

The two hypotheses are complementary: $H_1(i,j) = \overline{H_0(i,j)}$. We observed the values of the difference $\theta_i - \theta_j$ for short delays $|t_i - t_j|$ up to T_{short}, where T_{short} is a small number of time frames (e.g. 10). See Fig. 1 for an example. Our interpretation is as follows: the two location estimates X_i and X_j either correspond to the same object or not. In the first case the difference $\theta_i - \theta_j$ is small: an object does not move a lot during a short time period. Hence the zero-mean central peak in the histogram. In the second case the difference $\theta_i - \theta_j$ is random: trajectories of two objects are independent, at least in the short-term. We therefore propose the following

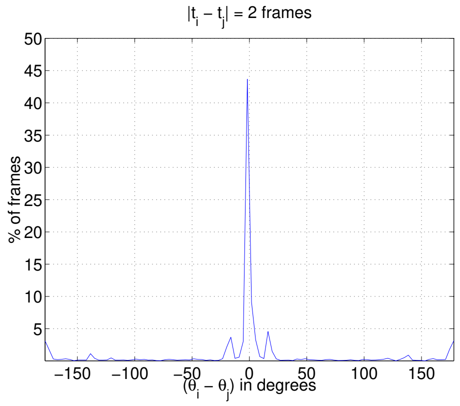

Fig. 1. Histograms of angle variation for a 2-frame delay, on real data (one meeting).

model for local dynamics, i.e. for $|t_i - t_j| \leq T_{short}$:

$$\begin{cases} p(\theta_i - \theta_j | H_0(i,j)) \sim \mathcal{N}\left(0, \sigma^{diff}_{|t_i - t_j|}\right) \\ \\ p(\theta_i - \theta_j | H_1(i,j)) \sim \mathcal{N}\left(0, \sigma^{same}_{|t_i - t_j|}\right) \end{cases} \quad (1)$$

where $\forall T \; \sigma^{same}_T < \sigma^{diff}_T$. Although an intuitive choice in the case of H_0 would be a uniform distribution, we opted for a Gaussian in order to capture the dependency of σ^{diff}_T on the delay T. This dependency was observed on real data, examples can be found in [10].

We note that the standard deviation σ^{same}_T accounts for short-term variations of location estimates due to both local motion and measurement imprecision. We argue that there is no need to distinguish between the two, as long as the analysis is restricted to short delays $T \leq T_{short}$. For each delay T, σ^{same}_T and σ^{diff}_T can be estimated by simply training a bi-Gaussian model on the entire data $\{\theta_i - \theta_j\}$ for $|t_i - t_j| = T$. The mean of each Gaussian is fixed to zero.

Finally, we note that the hypothesis $\sigma^{same}_T < \sigma^{diff}_T$ does not account for trajectory crossings, see Sections 2.5 and 2.6 for further discussion on this topic.

2.2. Threshold-Free Maximum Likelihood Clustering

Given the local dynamics, our task is to detect and track events. We propose to view the problem as follows: find a partition

$$\Omega = \{\omega_1, \cdots, \omega_{N_\Omega}\} \quad (2)$$

of $(X_1, X_2, \dots, X_N)$ that maximizes the likelihood of the observed data $p(X|\Omega)$. Each cluster ω_k contains locations for one event, e.g. a speech utterance. We are *not* trying to produce a single trajectory per object, but rather an

1. Train standard deviations σ_T^{same} and σ_T^{diff} over the entire data $X_{1:N}$ for $1 \leq T \leq T_{short}$.
   Initialize $t_0 \leftarrow 0$.
2. $F \leftarrow [t_0, t_0 + T_{future}]$.
   Define all possible partitions of location estimates in F. Choose the most likely partition $\hat{\Omega}_F^{ML}$.
3. $P \leftarrow [t_0 - T_{past}, t_0 - 1]$.
   Define all possible merges between $\hat{\Omega}_P^{ML}$ and $\hat{\Omega}_F^{ML}$. Choose the most likely merged partition and update $\hat{\Omega}_{[1,t_0+T_{future}]}^{ML}$.
4. $t_0 \leftarrow t_0 + T_{future}$ and loop to Step 2.

Table 1. The sliding window Maximum Likelihood (ML) algorithm. $T_{short} = T_{past} + T_{future}$. The likelihood of a partition is defined by Eq. (3).

oversplitted solution where N_Ω is the number of individual events, for example speech utterances. The value of N_Ω is not important for this algorithm: we only want to be *sure* that all location estimates within each cluster ω_k corresponds to the *same* object.

Using location cues alone, we can relate location estimates in the short-term only. We therefore propose to maximize the following "short-term criterion":

$$\boxed{p_{ST}(X|\Omega) \propto \prod_{\substack{i<j \\ |t_i - t_j| \leq T_{short}}} p\left(\theta_i - \theta_j | H^\Omega(i,j)\right)} \tag{3}$$

where $H^\Omega(i,j)$ is either $H_0(i,j)$ or $H_1(i,j)$, depending on whether or not X_i and X_j belong to the same cluster ω_k in candidate partition Ω. Each term of the product is expressed using Eq. (1). Since even short recordings contain thousands of time frames, thousands of locations estimates are extracted. It is thus untractable to try all possible partitions Ω. We propose to find a suboptimal solution by using a sliding analysis window spanning $T_{short} = T_{past} + T_{future}$ frames. Two consecutive windows have an overlap of T_{future} frames. The algorithm is described in Table 1. It is important to note that likelihood estimations are *local* only, irrespective of frames outside F (Step 2) and $P \cup F$ (Step 3).

The result of this algorithm is an estimate $\hat{\Omega}^{ML}$ of the ML partition of all data $X_{1:N}$. We note that the entire process is deterministic and threshold-free.

2.3. Computational Load

One interest of this approach is **bounded computational load**. We take the case of exactly one location estimate per time frame and $T_{past} = T_{future} = T_{short}/2$. We calculated the number of partitions to evaluate at Step 2. For $T_{future} \leq 6$, there are at most 203 such partitions. We also calculated the *worst case* number of merges to evaluate at Step 3, that is when each half of the analysis window has $T_{short}/2$ 1-element clusters. We found that for $T_{short}/2 \leq 6$, the number of merges is at most 13,327. With unoptimized code and full search, we obtained real-time computations for $T_{short}/2 \leq 6$. Evaluating a candidate partition (Step 2) or merge (Step 3) following Eq. (3) is easily implemented through a sum in the log domain over location estimates within F (Step 2) or $P \cup F$ (Step 3).

For larger windows $T_{short}/2 \geq 7$ and/or multiple location estimates per time frame, it is possible to design simple pruning heuristics that exclude most of the unlikely partitions (resp. merges) at Step 2 (resp. Step 3). Indeed, this is necessary only in periods where most clusters contain only one element. On those periods, most locations estimates are unrelated to each other, so oversplitting is not a big loss.

For more details on computational load, the interested reader is refered to [10].

2.4. Online Implementation

We note that the proposed algorithm is intrinsically online: the loop defined by Steps 2, 3 and 4 relies on a sliding window of T_{short} frames. Only Step 1 needs batch training on the entire data. However, Step 1 can also be implemented online in a straightforward manner with a forgetting factor (see [10]).

2.5. Confidence Measure

This Section defines a confidence measure for each possible individual decision $H_d(i,j)$ (d is 0 or 1), and explains how it allows to detect and solve low confidence situations such as trajectory crossings.

With an equal priors assumption on all possible partitions $\{\Omega\}$:

$$P(H_d(i,j)|X) \propto \sum_{\substack{\Omega \\ H^\Omega(i,j) = H_d(i,j)}} p(X|\Omega) \tag{4}$$

$P(H_d(i,j)|X)$ is a posterior probability. It can be interpreted as the confidence in the local decision $H_d(i,j)$.

2.6. Application to Clean Data: Confident Clustering

We define "clean data" as data where each location estimate corresponds to an object in the physical world (no noisy location estimate). In such a case each location estimate belongs to the trajectory of an object. We would like to determine when trajectories cross.

To do so, within each analysis window of the sliding window ML algorithm, we propose to determine whether the ML algorithm is *confident* which cluster each location estimate belongs to. In Step 2 and Step 3, we add the following post-processing:

- For all (X_i, X_j) in the analysis window, estimate $P\left(H^{\hat{\Omega}^{ML}}(i,j)\,|X\right)$ using Eq. (4). We use the set of candidate partitions (Step 2) or candidate merged partitions (Step 3) as $\{\Omega\}$.

- Step 2: whenever a decision $H_0(i,j)$ given by the ML algorithm has "low confidence", split in two parts the cluster containing X_i, at time t_i. Idem for X_j. Additional one-element clusters $\{X_i\}$ and $\{X_j\}$ are created.

- Step 3: whenever a decision $H_0(i,j)$ given by the ML algorithm has "low confidence", cancel the merge between the cluster containing X_i and the cluster containing X_j.

Fig. 3 gives an example: X_i and X_j are very close, yet the ML algorithm leads to the decision $H^{\hat{\Omega}^{ML}} = H_0(i,j)$. Confidence in the latter is therefore expected to be low. In order to detect "low confidence" in a decision $H_0(i,j)$, we compare it to all decisions $H_1(r,s)$ given by the ML algorithm, where X_i, X_j, X_r and X_s are all in the same analysis window. Formally, a "low confidence" decision is defined as:

$$\begin{aligned} H^{\hat{\Omega}^{ML}}(i,j) = H_0(i,j) \quad & and \\ P\left(H_0\left(i,j\right)|X\right) \quad < \quad & M_1\left(\hat{\Omega}^{ML}\right) \end{aligned} \tag{5}$$

where:

$$M_1\left(\hat{\Omega}^{ML}\right) \triangleq \max_{\substack{r<s \\ H^{\hat{\Omega}^{ML}}(r,s)=H_1(r,s)}} P\left(H_1\left(r,s\right)|X\right) \tag{6}$$

2.7. Multi-Object Tracking Examples

We generated clean data that simulates "sporadic" and "concurrent" events by restricting to one location estimate per time frame, yet with trajectories that look continuous enough so that it is still a tracking problem. The task is twofold:

1. From instantaneous location estimates, build the various trajectories accurately.

2. Extract pieces of trajectory, where each piece must belong to a single object. This implies that no cluster extends beyond any trajectory crossing.

Fig. 2 compares the result of the ML clustering with the result of the confident clustering described in Section 2.6. In all test sequences, the number of active objects varies over time, and trajectories cross several times. We can see that, although the ML clustering correctly builds the various trajectories (task 1), it produces arbitrary decisions around the points of crossing. On the contrary, the confident clustering correctly splits the trajectories at all crossing points (task 2).

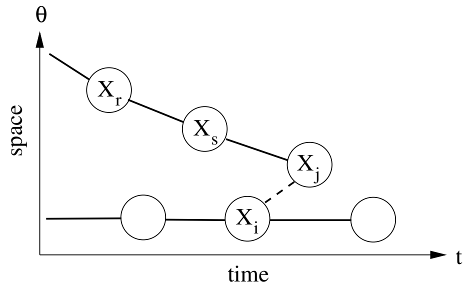

Fig. 3. An example of low confidence situation: a trajectory crossing. Each circle is a location estimate. Each ML cluster is depicted by a continuous line. The low confidence $H_0(i,j)$ decision is depicted by a dashed line.

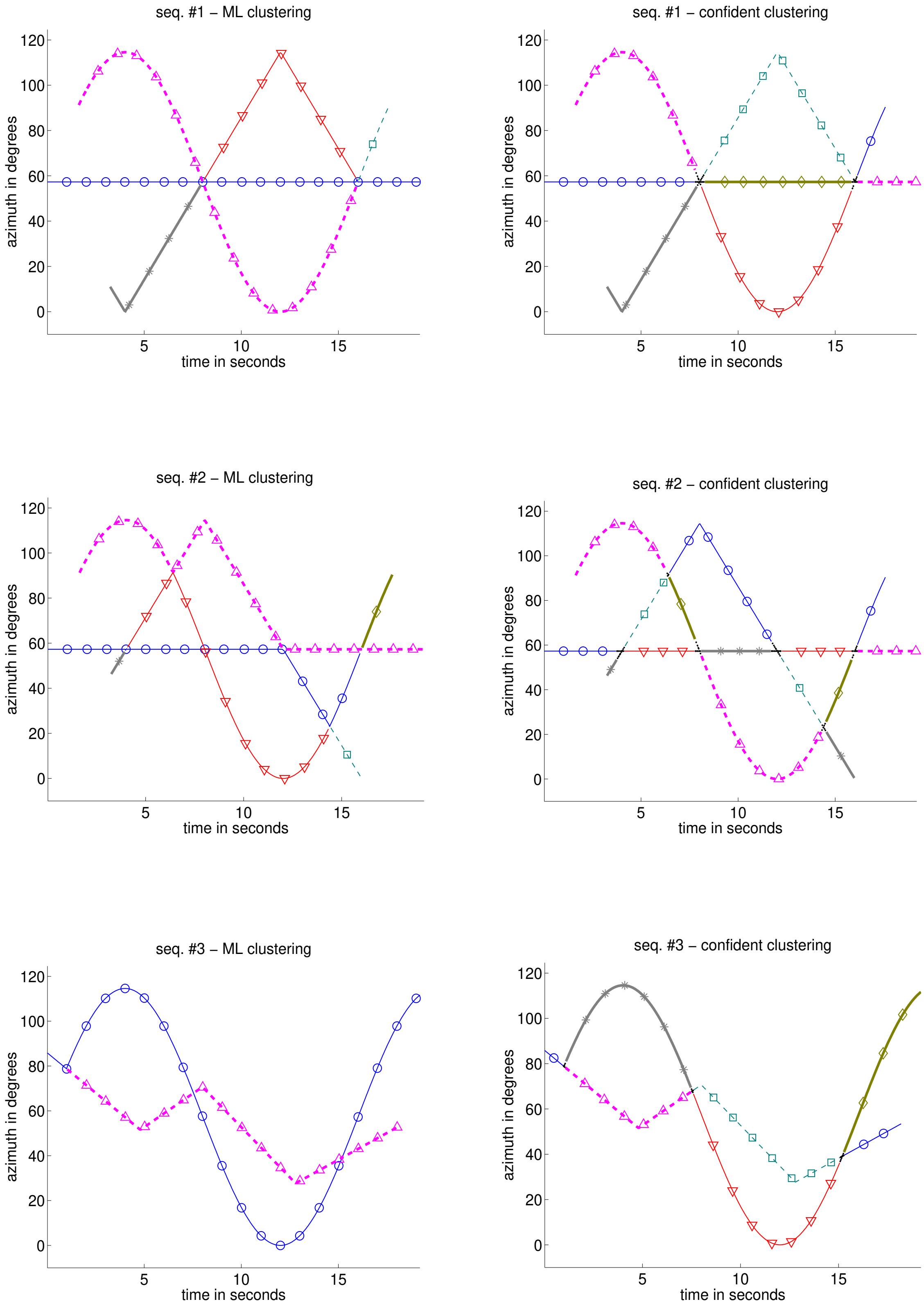

Fig. 2. Comparison ML clustering / confident clustering on multiple object cases. The number of active objects varies over time. We can see that the ML clustering algorithm makes arbitrary decisions at trajectory crossings. On the contrary, the confident clustering correctly splits the clusters at each trajectory crossing. (Changes of colors, markers and linestyles indicate beginning and end of clusters.)

3. MEETING SEGMENTATION APPLICATION

In this Section we report experiments conducted on real meeting data recorded with one circular microphone array. We first describe the proposed approach, which incorporates the ML short-term clustering algorithm presented in Section 2. We then describe the test data and define performance measures. Finally, results are given that validate the proposed approach and compare it with alternative approaches.

3.1. Proposed Approach

The proposed implementation uses distant microphones only, and produces a discrete set of regions, along with a speech/silence segmentation for each region.

The differences between a previous work [12] and the approach presented here are that:

- We are focusing on the speech segmentation task only, not on the speaker identification/clustering task.
- We use distant microphones only (no lapel).
- We segment each meeting independently.
- The proposed approach does not rely on a Hidden Markov Model (HMM).

We segment each meeting independently, with a 3-step algorithm:

1. **Frame-Level Analysis:** within each time frame, estimate the location of the dominant sound source θ_i. We used a direct grid-based search for the global maximum of the SRP-PHAT measure [11].

2. **Short-Term Analysis:** run the ML short-term clustering algorithm described in Table 1 to cluster location estimates $X_{1:N}$ into $\Omega = (\omega_1, \dots, \omega_{N_\Omega})$ (We used $T_{past} = T_{future} = 7$ frames). Keep only clusters ω_i spanning more than 100 ms of duration. This value was set as a strict minimum for a speech utterance to be significant.

3. **Long-Term Analysis:** apply K-means on the centroids of the remaining short-term clusters. The product is a list of regions defined by their centers $(\theta^{(1)}, \dots, \theta^{(L)})$. L is selected automatically, as in [12].

The speech/silence segmentation for each region l is directly defined by the short-term clusters $\{\omega_k\}$ for which $\theta^{(l)}$ is the closest region center. Frames in those short-term clusters are classified as speech for region l, other frames as silence. This can be opposed to methods relying on other cues than location, e.g. a prior Voice Activity Detection. The interest of using location cues alone has already been noticed in [13].

Fig. 4 shows that Step 2 has a strong denoising effect. However, we can see that it still keeps short segments. This will be confirmed by results in Section 3.5. This is very important in order to detect a speaker that would say only a few words over the whole meeting: these words may be for example part of key decisions taken in the course of the meeting.

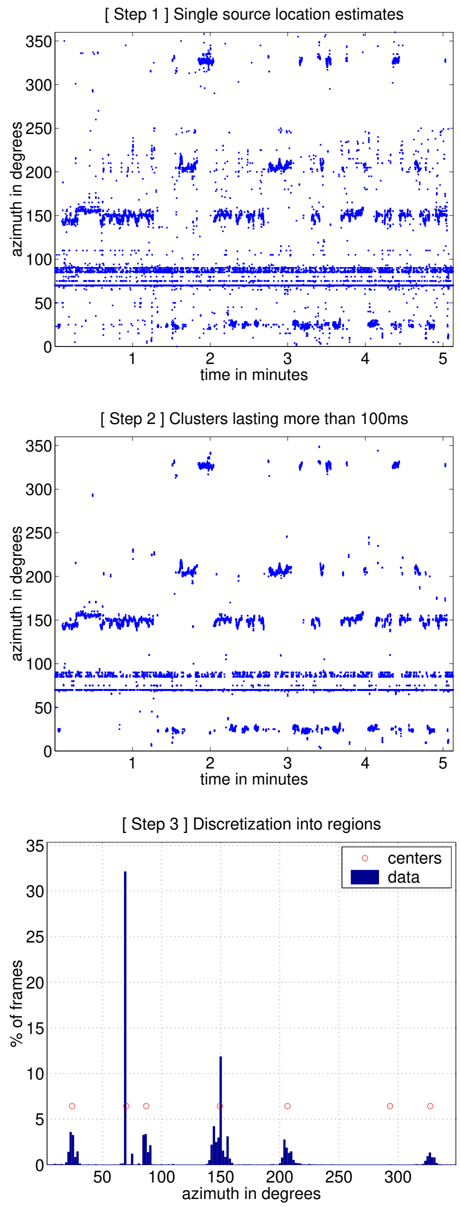

Fig. 4. 3-step segmentation algorithm: output of each step

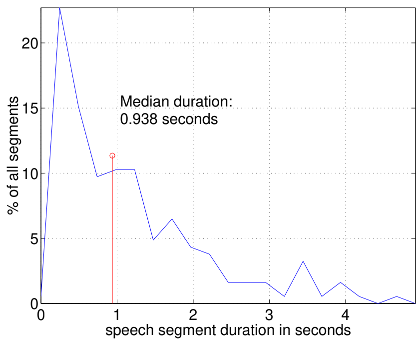

Fig. 5. Histogram of speech segment durations in the ground-truth

3.2. Test Data

The test corpus includes 17 short meetings from a publicly available database (`http://mmm.idiap.ch`). The total amounts to 1h45 of multichannel speech data. In all meetings, an independent observer provided a very precise speech/silence segmentation. Because of this high precision, the ground-truth includes many very short segments. Indeed, 50% of the speech segments are shorter than 0.938 seconds, see Fig. 5.

3.3. Performance Measures

We evaluated speech/silence segmentation for regions containing speakers only, ignoring the additional regions corresponding to machines such as the projector (which generate clusters of acoustic locations).

We first counted false alarms and false rejections in terms of frames, on each speech/silence segmentation. For each meeting we summed the counts and deduced False Alarm Rate (FAR), False Rejection Rate (FRR) and Half Total Error Rate HTER = (FAR + FRR) / 2.

Second, we evaluated precision (PRC) and recall (RCL). Since most of the speech segments are very short, we defined precision and recall at the frame level, rather than at the segment level. F-measure is defined as:

$$F = \frac{2 \times PRC \times RCL}{PRC + RCL} \qquad (7)$$

3.4. Lapel Baseline

Our scheme uses distant microphones only. We decided to compare with a lapel-only baseline. The latter is a simple energy-based technique that selects the lapel with the most energy at each frame, and applies energy thresholding to classify the frame as speech or silence. The lapel baseline output is smoothed with a low-pass filter. We could not use Zero-Crossing Rate (ZCR), because it was degrading the results. We found that ZCR is very sensitive to noises such as writing on a sheet of paper. Finally, we must mention that a dilation of a few frames was applied to the resulting segments in both approaches, in order to capture beginning and ends of speech segments. To tune this value we used the same extra set of 3 meetings (not included in the test set) in both cases, maximizing the F-measure.

	Proposed	Lapel baseline
PRC	79.7 (**55.4**)	84.3 (46.6)
RCL	**94.6** (**84.8**)	93.3 (66.4)
F	86.5 (**67.0**)	88.6 (54.7)

Table 2. Segmentation results on 17 meetings. The proposed approach uses distant microphones only. Values are percentages, results on overlaps only are indicated in brackets.

	Proposed	HMM-based
HTER	5.3	17.3

Table 3. Comparison with previous work: segmentation results on 6 meetings, with silence minimum duration of 2 seconds. Values are percentages.

3.5. Results

Table 2 gives the results for the proposed approach and the lapel baseline on the 17 meetings. We can see that the proposed approach gives good results, and compares well with the lapel baseline. The proposed approach yields major improvement on overlapped speech. These results are particularly significant, given the high precision of the ground-truth and the fact that we use distant microphones only. The slight decrease in F-measure is due to the higher number of low-energy segments detected by the proposed approach, such as breathing.

From the applicative point of view mentioned in Section 3.1, we can see that the proposed approach fulfills the goal of capturing as many utterances as possible, especially on overlaps (RCL figures in Table 2).

We also compared our approach to a HMM-based previous work [12], on a slightly different task: only 6 meetings are segmented, and the task excludes silences smaller than 2 seconds. Results are reported in Table 3. There is a clear improvement. However, the previous work was attacking a wider task: speech segmentation and speaker clustering. This comparison shows that we can obtain a very good segmentation with event location cues alone.

4. DISCUSSION

In Section 2 we introduced a novel short-term clustering algorithm, motivated from observations on real data. It is based on a very simple hypothesis on local dynamics. It is threshold-free, intrinsically online and fully deterministic. It can run in real-time for reasonable context durations. Moreover, in the case of clean data, we described an efficient way of detecting and solving low-confidence situation trajectory crossings. Tracking experiments on synthetic data show the effectiveness of the proposed approach. Future work will investigate application of the confidence measure to real, noisy data.

In Section 3 we showed that the performance of the proposed approach on the meeting segmentation task is very good, especially on overlaps. This is particularly significant because we used distant microphones only, and output of a single source localization algorithm. Our algorithm compares very well with a lapel-only baseline, while giving a major improvement on overlapped speech. Our interpretation is that the proposed algorithm is particularly efficient to track concurrent events, as shown in Section 2.7. We can expect even better results when using a multiple sources localization algorithm to produce the instantaneous location estimates.

We can compare with previous work in the domain of location-based speaker segmentation. Offline methods [2] and online methods [3] already achieved very good results, especially on overlaps. However, both works were based on the prior knowledge of the locations of all speakers. On the contrary, the approach presented in this paper is unsupervised: local dynamics are extracted from the data itself, and short-term clustering is threshold-free. The segmentation application based on it is also unsupervised: it does not rely on any prior knowledge of speakers' locations.

5. CONCLUSION

Accurate segmentation and tracking of speech in a meeting room is crucial for a number of tasks, including speech acquisition and recognition, speaker tracking, and recognition of higher-level events.

In this paper, we first described a generic, threshold-free scheme for short-term clustering of sporadic and concurrent events. The motivation behind this approach is that with highly sporadic modalities such as speech, it may not be relevant to try to output a single trajectory for each object over the entire data, since it leads to complex data association issues. We propose here to track in the short-term only, thus avoiding such issues. We described an algorithm based on a sliding-window analysis, spanning a context of several time frames at once. It is online, fully deterministic and can function in real-time for reasonable context durations. It is unsupervised: local dynamics are extracted from the data itself, and the short-term clustering is threshold-free. We also presented initial investigations on the problem of trajectory crossings, in the case of clean data.

Second, we described a speech specific application of this algorithm: segmentation of speech in meetings recorded with a microphone array. This application is unsupervised: it does not rely on prior knowledge of speakers' locations. We showed it compares well to a lapel-only technique, and yields major improvement on overlapped speech. We also compared the proposed approach with a HMM-based technique using both distant microphones and lapels. Clear improvement is obtained. These results validate the short-term clustering algorithm, as well as the idea of using location cues alone for obtaining high precision segmentation of multi-party speech.

Future work will test the short-term clustering algorithm on recordings with more complex human motions. We will also investigate applications of this approach to various scenarios such as higher dimensionality (e.g. azimuth/elevation location estimates) and multiple location estimates per time frame. In a complementary direction, we will investigate the use of the short-term tracking algorithm for speech acquisition and subsequent speaker identification. Finally, we also plan to extend the use of confidence measures to real, noisy data.

6. ACKNOWLEDGMENTS

The authors acknowledge the support of the European Union through the M4 and HOARSE projects. This work was also carried out in the framework of the Swiss National Center of Competence in Research (NCCR) on Interactive Multi-modal Information Management (IM)2.

7. REFERENCES

[1] E. Shriberg, A. Stolcke, and D. Baron, "Observations on overlap: findings and implications for automatic processing of multi-party conversation," in *Proceedings of Eurospeech 2001*, 2001, vol. 2, pp. 1359–1362.

[2] G. Lathoud and I. McCowan, "Location based speaker segmentation," in *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-03)*, Hong Kong, April 2003.

[3] Guillaume Lathoud, Iain A. McCowan, and Darren C. Moore, "Segmenting multiple concurrent speakers using microphone arrays," in *Proceedings of Eurospeech 2003*, Geneva, Switzerland, September 2003.

[4] H. Sorenson, *Kalman Filtering: Theory and Application*, IEEE Press, 1985.

[5] S.J. Julier, J.K. Uhlmann, and H.F. Durrant-Whyte, “A new approach for filtering nonlinear systems,” in *Proceedings of the 1995 American Control Confernce*, 1995, pp. 1628–1632.

[6] S.J. Julier and J.K. Uhlmann, “A new extension of the Kalman filter to nonlinear systems,” in *Proceedings of AeroSense: the 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*. 1997, Multi Sensor Fusion, Tracking and Resource Management II, SPIE.

[7] J. LaViola, “A comparison of Unscented and Extended Kalman Filtering for estimating quaternion motion,” in *Proceedings of the 2003 American Control Conference*. June 2003, pp. 2435–2440, IEEE Press.

[8] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian bayesian state estimation,” in *IEEE Proceedings*, 1993, vol. 140, pp. 107–113.

[9] J.R. Larocque, J.P. Reilly, and W. Ng, “Particle filters for tracking an unknown number of sources,” *IEEE Transactions on Signal Processing*, vol. 50, no. 12, December 2002.

[10] G. Lathoud, J.M. Odobez, and I.A.McCowan, “Short-term spatio-temporal clustering of sporadic and concurrent events,” IDIAP-RR 04-14, IDIAP, 2004.

[11] J. DiBiase, H. Silverman, and M. Brandstein, “Robust localization in reverberant rooms,” in *Microphone Arrays*, M. Brandstein and D. Ward, Eds., chapter 8, pp. 157–180. Springer, 2001.

[12] J. Ajmera, G. Lathoud, and I.A. McCowan, “Segmenting and clustering speakers and their locations in meetings,” in *Proceedings the 2004 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2004.

[13] Daniel Gatica-Perez, Guillaume Lathoud, Iain McCowan, and Jean-Marc Odobez, “A Mixed-State I-Particle Filter for Multi-Camera Speaker Tracking,” in *2003 IEEE Int. Conf. on Computer Vision Workshop on Multimedia Technologies for E-Learning and Collaboration (ICCV-WOMTEC)*, 2003.