Short-Term Spatio-Temporal Clustering of Sporadic and Concurrent Events

Guillaume Lathoud [a,b] Jean-Marc Odobez [a]
Iain A. McCowan [a]

IDIAP–RR 04-14

April 2004

to appear in
Proceedings of the 2004 ICASSP-NIST Meeting Recognition Workshop,
Montreal, Canada, May 2004

Dalle Molle Institute
for Perceptual Artificial
Intelligence • P.O.Box 592 •
Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[a] IDIAP Research Institute, CH-1920 Martigny, Switzerland
[b] Swiss Federal Institute of Technology (EPFL), CH-1015 Lausanne, Switzerland

IDIAP Research Report 04-14

Short-Term Spatio-Temporal Clustering of Sporadic and Concurrent Events

Guillaume Lathoud Jean-Marc Odobez Iain A. McCowan

April 2004

to appear in

Proceedings of the 2004 ICASSP-NIST Meeting Recognition Workshop, Montreal, Canada, May 2004

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 multitarget 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. We also define a confidence measure for short-term clustering, and show on synthetic data that it can be used to detect and solve trajectory crossings.

1 Introduction

Usually, the task termed as "object tracking" aims at producing a single spatio-temporal trajectory for each tracked object, over the entire data. From data we can extract location estimates of the object, as well as cues on the identity of the object - e.g. acoustic features extracted from recorded speech, visual features extracted from images containing the face of a person. We can term these data-derived features as "location cues" and "identity cues" respectively. Each event is the observation of a signal emitted by an object. Ideally, both types of cues are integrated in a probabilistic framework that tries to find the hypothesis for objects' trajectories that optimize best the likelihood of the observed events, given the probabilistic model and chosen parameters. A lot of extremely valuable work has been done along this line: Kalman Filtering [1] and its variants [2, 3, 4], and Particle Filters [5] are two examples of such approaches.

However, for some modalities such as spontaneous speech, we have to deal with events that are sporadic and concurrent. "Sporadic" means that an event happens only a minor part of the time. For example in spontaneous speech, each person is silent most of the time: silences between utterances constitute the major part of that person's recording. "Concurrent" means that multiple events can happen concurrently. For example in multi-party speech, speakers overlap on each other in a non-negligible part of the recordings [6]. For these two reasons, using the frameworks mentionned above on real multi-party speech can lead to the definition of heuristical object birth and death processes and/or require strong hypotheses on motion dynamics. Determining the number of objects is a hard task in itself.

In the literature we found an excellent example of approach that uses event location cues only, with Particle Filtering [7]. It stresses the interest of *jointly* estimating the trajectories with the number of objects. It achieves correct multitarget tracking of varying number of sources on several simulation examples, including one case of trajectories crossing, without need for parameter tuning. However, it relies on complex rules called birth, death, split and merge. Moreover, in all simulation examples objects are emitting at all time frames, which may well be the case for radar-based localization but not for speech. Indeed, in the middle of speech segments, there are often a few frames for which the emitted signal is either null or too weak to be localized. While that approach is certainly useful e.g. for multitarget radar tracking, one can question its use on modalities that are highly sporadic. This by no means reduces the interest of using Particle Filters in the case of modalities where each individual event is observable continuously over a long duration of time, such as radar or video.

To avoid these data association issues, we propose to first solve the multitarget tracking and segmentation problem in the short-term. The idea is to produce short segments and trajectories in respectively time and space, where each such segment corresponds to only one object *for sure.* We argue that such segments are a solid basis for further long-term processing, e.g. speaker clustering using speech signals acquired within those segments. On the application side, successfully implementing this first step is already interesting in itself: this paper reports a speech segmentation evaluation in the case of real multi-party meetings recorded with arrays of far-distance microphones. Results show that the produced segmentation compares well with lapels, while providing an additional speaker location information. This is particularly important, considering the fact that location information allows for a less constrained environment, namely no lapel to wear and variable number of persons or acoustic objects. This opens the way towards a wide range of applications that could hardly be implemented with lapels, such as audio-visual speaker tracking, surveillance, camera steering and high-level meeting analysis [8].

We propose to view this problem as short-term multitarget tracking of individual events, using location cues only. The core idea is that locally, an object's location remains constant. Therefore locally, the location of an observed event (e.g. a speech utterance) also remains constant. We will demonstrate in this paper that a very simple model for local dynamics is sufficient to extract short-term trajectories *with high confidence.* In order to achieve this, we propose to attack the multitarget tracking problem as a clustering approach: given two instantaneous location estimates obtained at two different times, we determine whether or not we are *highly confident* that they correspond to the same object. The result is a partition of all instantaneous location estimates into clusters. We put two location estimates in the same cluster only when we are *highly confident* that they correspond to the same object. Our approach aims at finding the partition that optimizes the likelihood of instantaneous location estimates, given the local dynamics model and the chosen partition. The local dynamics model is extracted from simple statistics over the data.

To summarize, we are targetting tracking and segmentation of events in the short-term only, without trying to identify objects causing the events. The closest existing work we found is [9], which is close in essence to our approach. However, it deals with visual observations, where events are observable almost all the time. It therefore seemed difficult to apply it to the present context of sporadic events. Moreover, a fundamental difference is that our work does *not* rely on object identity cues, but on event location cues only.

The aim of this paper is twofold. The first goal is to introduce a generic approach for short-term clustering of sporadic and concurrent events. The proposed Maximum Likelihood (ML) approach is online, threshold-free, fully deterministic and does not require any random sampling. We then propose to turn the ML approach into a confident clustering approach, that detects and solves trajectory crossings. In all synthetic data test cases, the confident approach was found to be fully effective.

The second aim of this paper is to prove the validity of this approach on real data. We describe a multi-party speech segmentation application that uses the short-term clustering algorithm. It uses location information extracted from a microphone array, without prior voice activity detection. Results on a publicly available corpus of meetings validate the approach, and show that it compares well with a lapel baseline, which instead uses energy from lapel microphones.

The rest of this paper is organized as follows: in Section 2 we present the model for local dynamics, justifying it with observations on real data. In Section 3 we describe the clustering approach, along with a sliding-window algorithm. Section 4 presents a confidence-based approach to detect and solve trajectory crossings. Section 4.3 presents tests on synthetic data that validate the proposed algorithm. Section 5 presents a speech segmentation application on 17 real meetings. We conclude on further possible applications of the proposed approach.

2 Local Dynamics

Throughout the paper the notation p designates a probability density function (pdf) or likelihood. The notation P designates a probability or a posterior probability.

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

For each time frame t, there can be zero, one or multiple location estimates. For example:

- There may not exist any location estimate X_i such that $t_i = t$.
- There may exist two location estimates X_i and X_j such that $t_i = t_j$ and $i \neq j$.

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

$$\begin{cases} H_0(i,j) \triangleq \textit{"}X_i \textit{ and } X_j \textit{ correspond to } \textbf{different} \textit{ objects."} \\ H_1(i,j) \triangleq \textit{"}X_i \textit{ and } X_j \textit{ correspond to the } \textbf{same} \textit{ object."} \end{cases} \quad (1)$$

The two hypotheses are complementary: $H_1(i,j) = \overline{H_0(i,j)}$. In the rest of this Section we will define a likelihood model for each of the two propositions. We first give some preliminary observations on real data that justify the following model.

2.1 Observations on Real Data

In the rest of this paper, we will use the following 1-dimensional context to both explain our approach and illustrate its benefits: one horizontal planar circular microphone array, placed on a table in a

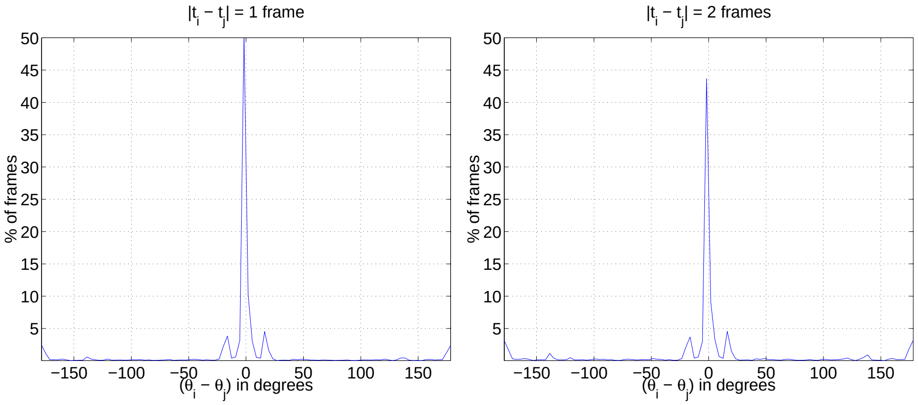

Figure 1: Histograms of azimuth angle variation for 1-frame and 2-frame delays.

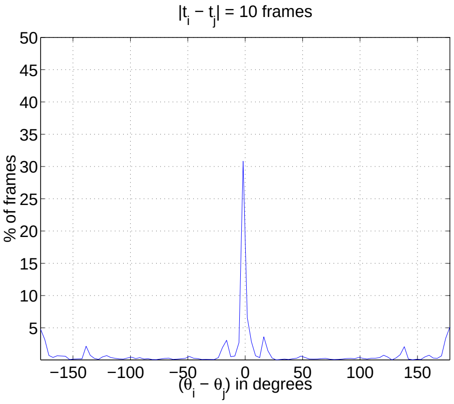

Figure 2: Histogram of azimuth angle variation for a 10-frame delay.

meeting room. We chose "event location cues" θ_i to be azimuth estimates (1-dimensional) of the dominant sound source. One location estimate is provided per frame: all frames are considered regardless of whether they contain speech or not.

We used 5 minutes of real meeting data from a corpus publicly available at `http://mmm.idiap.ch`. 32 ms-long time frames were defined every 16 ms (62.5 Hz, 50% overlap). We first ran a direct grid-based search for the dominant sound source location θ_i at each time frame t_i, based on the SRP-PHAT measure [10]. We plotted the histogram of differences $\theta_i - \theta_j$ for each possible delay $|t_i - t_j|$ up to $T_{short} = 10$ frames (160 ms). See Figs. 1 and 2 for some of these histograms.

A first glance at those histograms shows that they share a similar structure: a central peak around zero deviation, in the middle of a background noise. We note that the central peak gets smaller as the delay $|t_i - t_j|$ gets longer. We can therefore expect this structure to be valid in the short term only.

How to choose a proper statistical model? We first thought of using a single Gaussian pdf for the H_1 hypothesis (central peak of the histogram) and a uniform pdf for H_0 hypothesis. To check whether the uniform pdf over $[-pi, +pi]$ is a correct model for H_0, we computed the mean and standard deviation of all values with $|\theta_i - \theta_j| \geq 10$ degrees (this excludes the central peak). Results are reported in Table 1. We see that the distribution is zero-mean, but also that the standard deviation depends on $|t_i - t_j|$. In the following, we therefore chose to use a single Gaussian pdf for $p(\theta_i - \theta_j | H_0)$, rather than a uniform pdf, in order to capture this dependency.

To precise observations in a numerical manner, for each possible delay $|t_i - t_j|$ we divided the

T	$\mu_T^{(u)}$	$\sigma_T^{(u)}$	$\frac{\mu_T^{(u)}}{\sigma_T^{(u)}}$
1	-0.970	107.318	-0.009
2	-0.633	115.021	-0.005
3	-0.314	120.361	-0.003
4	0.942	123.371	0.008
5	0.488	125.615	0.004
6	0.131	126.818	0.001
7	0.279	127.444	0.002
8	-0.605	127.711	-0.005
9	0.008	128.370	0.000
10	-0.105	128.448	-0.001
11	-0.222	128.738	-0.002
12	0.242	128.967	0.002
13	0.044	129.066	0.000
14	-0.478	129.137	-0.004

Table 1: Mean $\mu_T^{(u)}$ and standard deviation $\sigma_T^{(u)}$, in degrees, of all values $(\theta_i - \theta_j) \geq 10$ degrees, and increasing delay $T = |t_i - t_j|$. We can see the distribution is approximately centered around zero, and that the standard deviation depends on T.

T	μ_T	σ_T	$\frac{\mu_T}{\sigma_T}$	μ_T^{noise}	σ_T^{noise}	$\frac{\mu_T^{noise}}{\sigma_T^{noise}}$
1	0.006	0.480	0.012	-0.710	88.656	-0.008
2	0.010	0.749	0.013	-0.499	102.587	-0.005
3	0.100	1.145	0.088	-0.363	113.069	-0.003
4	0.005	1.858	0.003	0.859	121.719	0.007
5	0.002	2.074	0.001	0.447	124.828	0.004
6	0.024	2.107	0.012	0.077	125.938	0.001
7	-0.016	2.027	-0.008	0.238	126.360	0.002
8	0.009	2.027	0.004	-0.629	126.673	-0.005
9	-0.010	2.041	-0.005	-0.040	127.358	-0.000
10	0.004	2.119	0.002	-0.131	127.731	-0.001
11	0.012	2.151	0.006	-0.250	127.975	-0.002
12	-0.017	2.151	-0.008	0.195	128.178	0.002
13	-0.019	2.134	-0.009	0.017	128.115	0.000
14	-0.035	2.149	-0.016	-0.512	128.180	-0.004

Table 2: Result of the bi-Gaussian training for each possible time difference $T = |t_i - t_j|$, on meeting data. T is a value in frames, μ and σ are values in degrees.

T	μ_T	σ_T	$\frac{\mu_T}{\sigma_T}$	μ_T^{noise}	σ_T^{noise}	$\frac{\mu_T^{noise}}{\sigma_T^{noise}}$
1	-0.022	2.726	-0.008	-2.012	78.027	-0.026
2	-0.028	3.256	-0.009	1.856	79.546	0.023
3	-0.292	4.187	-0.070	0.965	84.572	0.011
4	-0.435	5.453	-0.080	0.247	86.172	0.003
5	-0.516	5.581	-0.092	0.703	87.713	0.008
6	-0.593	6.878	-0.086	-1.082	89.274	-0.012
7	-0.646	6.544	-0.099	-0.174	89.413	-0.002
8	-1.066	6.688	-0.159	0.435	88.029	0.005
9	-0.917	7.205	-0.127	-0.621	88.600	-0.007
10	-1.399	6.879	-0.203	0.107	87.755	0.001
11	-1.337	7.605	-0.176	0.759	89.388	0.008
12	-1.496	7.759	-0.193	0.521	88.885	0.006
13	-1.316	8.624	-0.153	0.911	89.777	0.010
14	-1.893	9.558	-0.198	3.859	91.838	0.042

Table 3: Result of the bi-Gaussian training for each time difference $T = |t_i - t_j|$, on data with several speakers, always moving when speaking. T is a value in frames, μ and σ are values in degrees.

histogram in two parts, each part modelled by a Gaussian and recursively estimated the Gaussians/relabelled the data. This converged to the values indicated in Table 2 for a 5-minute meeting and Table 3 for a recording with mostly moving speakers. In both cases we can see that all Gaussians are essentially centered around zero, and that the standard deviation mostly increases with increasing delay $|t_i - t_j|$.

2.2 Proposed Model

We now make the following assumption:

$$\left[\begin{array}{c} \begin{array}{l} \forall (i,j) \; such \; that \\ |t_i - t_j| \leq T_{short} \end{array} \left\{ \begin{array}{l} p(\theta_i - \theta_j | H_0(i,j)) \sim \mathcal{N}\left(0, \sigma^{diff}_{|t_i - t_j|}\right) \\ p(\theta_i - \theta_j | H_1(i,j)) \sim \mathcal{N}\left(0, \sigma^{same}_{|t_i - t_j|}\right) \end{array} \right. \\ \\ \forall T \quad \sigma^{same}_T < \sigma^{diff}_T \\ \\ \forall T < T_{short} \quad \sigma^{same}_T \leq \sigma^{same}_{T+1} \end{array} \right] \tag{2}$$

where the standard deviations σ_T^{same} and σ_T^{diff} only depend on the delay (time difference) $|t_i - t_j|$, and T_{short} is the maximum delay for which we assume this model to be true. This model can be interpreted as follows:

- The zero-centered Gaussian assumption for $p(\theta_i - \theta_j | H_1(i,j))$ can be seen as a zero-motion assumption for any given object. This means we do not make any assumption on motion direction. This is a weak constraint, which allows the model to take into account a high variety of situations, as will be seen in Section 4.3. We note that the variance captures not only measurement noise but also actual motion of the object. We can see in Tables 2 and 3 that the variance increases with the time difference $|t_i - t_j|$, then reaches a plateau. The value of the plateau is more or less high, depending whether people are mostly moving or seated. We also note that the simple fact of having this model for each possible time difference of several time frames, allows for short interruptions in the emission of an object. This can be compared

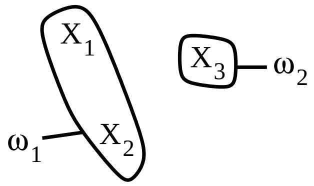

Figure 3: Example of partition of 3 elements: $\Omega = \{\{X_1, X_2\}, \{X_3\}\}$. We have $\omega_1 = \{X_1, X_2\}$ and $\omega_2 = \{X_3\}$. The equivalent graph notation is: $(H^{\Omega}_{1,2} = 1,\ H^{\Omega}_{1,3} = 0,\ H^{\Omega}_{2,3} = 0)$.

to other solutions such as auto-regressive model fitting, which would need explicit definition of birth/death processes, and may not easily recover from short interruptions (silences) in the emission and sharp turns in the trajectories. The explicit model would potentially be very complex, and rely on many tunable parameters.

- The zero-centered Gaussian assumption for $p(\theta_i - \theta_j | H_0(i,j))$ corresponds to the observed fact that the variance depends on the time difference $|t_i - t_j|$, as mentioned in Section 2.1.
- The $\sigma_T^{same} < \sigma_T^{diff}$ hypothesis corresponds to the observed, very large value for σ^{noise} in Section 2.1. Intuitively, it simply means that if two locations are very close, there is a high chance that they correspond to the same object.
- The last hypothesis $\sigma_T^{same} \leq \sigma_{T+1}^{same}$ is not really necessary on real data. It is only included here to allow the model to cope with very specific cases of synthetic data, having peculiar, not peaky distribution for specific values of the time difference $|t_i - t_j|$. We note that it also corresponds to the observed increasing behaviour of the variance mentioned above.

In this Section, we have introduced and justified an assumption on local dynamics. Experiments reported in Sections 4.3 and 5 will demonstrate the validity of this assumption.

3 Threshold-Free Maximum Likelihood Clustering

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

$$\Omega = \{\omega_1, \cdots, \omega_{N_\Omega}\} \tag{3}$$

of $X = (X_1, X_2, \ldots, X_N)$ that maximizes the likelihood of the observed data $p(X|\Omega)$. Each cluster ω_k is a subset of X, it ideally contains locations for one event, e.g. a speech utterance. We are *not* trying to produce a single trajectory per object, but rather an oversplitted solution where N_Ω is the number of individual events, for example speech utterances. The exact value of N_Ω is not important for this algorithm: the main goal is to be *sure* that all location estimates within each cluster ω_k correspond to the *same* object.

The event "partition Ω" can be written with the equivalent graph notation:

$$\bigcap_{i<j} H^{\Omega}(i,j) \tag{4}$$

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. See Fig. 3 for an example of partition.

Ideally the goal is to maximize the posterior probability $p(\Omega|X)$. Here we make an equal prior assumption on all possible partitions $\{\Omega\}$, thus viewing the problem as a Maximum Likelihood approach. Although this assumption is not fully justifiable, we argue that it corresponds well to the nature of the data. Intuitively, since there can be many discontinuities, and many noisy location estimates whenever no object is active, we cannot use priors that would favor e.g. partitions with many continuities. In other terms, no regularization constraint is used in this approach.

The joint distribution of the observed location estimates and the candidate partition can be expressed as:

$$P(X,\Omega) = P\left(X, \bigcap_{i<j} H^{\Omega}(i,j)\right) = P\left(\bigcap_{i<j} \left(X_i, X_j, H^{\Omega}(i,j)\right)\right) \tag{5}$$

With a simplifying independence assumption:

$$P(X,\Omega) = \prod_{i<j} P\left(X_i, X_j, H^{\Omega}(i,j)\right) \tag{6}$$

The equal priors assumption on $\{\Omega\}$ implies equal priors on all possible pair-wise decisions $\{H^{\Omega}(i,j)\}$. Therefore:

$$P(X,\Omega) \propto \prod_{i<j} p\left(X_i, X_j | H^{\Omega}(i,j)\right) \tag{7}$$

Finally the likelihood of the observed data X given the candidate partition Ω:

$$p(X|\Omega) \propto \prod_{i<j} p\left(X_i, X_j | H^{\Omega}(i,j)\right) \tag{8}$$

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

$$\boxed{p_{ST}\left(X|\Omega\right) \propto \prod_{\substack{i<j \\ |t_i - t_j| \leq T_{short}}} p\left(X_i, X_j | H^{\Omega}(i,j)\right)} \tag{9}$$

Each term can be expressed using Eq. (2). This criterion is parameter-free: no threshold is needed, and the number of active objects does not need to be explicitely defined, as would be the case with e.g. the K-means algorithm. Here the number of objects is implicitely modelled and selected through optimization of this criterion.

A practical implication of the choice of a value for T_{short} is that whenever a source stops emitting during about T_{short} frames, the emissions before and after this pause will be clustered separately. This directly impacts on the granularity of the clustering. For example, in speech, T_{short} determines the order of magnitude of the minimum silence duration.

Optimization Algorithm

We naively counted all possible partitions for an increasing number of elements. Results are reported in Table 4. We can see that trying all possible partitions in order to find

$$\hat{\Omega}^{ML} = \arg\max_{\Omega} p\left(X|\Omega\right) \tag{10}$$

quickly becomes intractable: even with only one location estimate per time frame, real recordings involve thousands of time frames and therefore thousands of location estimates.

Hence the proposed suboptimal implementation of Eq. (9) and Eq. (10): we will construct $\hat{\Omega}^{ML}$ progressively, using a sliding analysis window spanning $T_{short} = T_{past} + T_{future}$ time frames. We note

number of elements	number of possible partitions
1	1
2	2
3	5
4	15
5	52
6	203
7	877
8	4140
9	21147
10	115975
11	678570
>11	prohibitive

Table 4: Number of possible partitions, for each possible number of elements (Step 2 of the algorithm).

1. Train standard deviations σ_T^{same} and σ_T^{diff} over the entire data X 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 5: The sliding window Maximum Likelihood (ML) algorithm. $T_{short} = T_{past} + T_{future}$. The likelihood of a partition is defined by Eq. (9).

that T_{past} and T_{future} are not necessarily equal. Two consecutive analysis windows have an overlap of T_{future} frames. The implementation is described in Table 5.

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

Of course there are other possibilities for optimizing Eq. (9). We made this choice for the following reasons:

- Online analysis leads to easy real-time implementation, which allows other applications such as camera steering towards the active speaker.
- Locally, we have a true optimum - within “future” frames of each window.
- We tried to implement Steps 2 and 3 in a single step, but it did not yield particular improvement, and had a much higher computational cost. Pruning strategies were necessary.

Computational Complexity

One interest of this approach is **bounded computational load**. For both Step 2 and Step 3, evaluating a candidate partition (Step 2) or merge (Step 3) following Eq. (9) is easily implemented

T_{half}	N_{worst}
1	2
2	7
3	34
4	209
5	1,546
6	13,327
7	130,922
8	1,441,729
9	17,572,114
10	234,662,231

Table 6: Worst case number of merges to evaluate (Step 3 of the algorithm)

through a sum in the log domain over location estimates within F (Step 2) or $P \cup F$ (Step 3). The question is: how many partitions must be evaluated?

We calculated the number of partitions to evaluate at Step 2, shown in Table 4. We can see that 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, when each half of the analysis window has T_{half} 1-element clusters ($T_{past} = T_{future} = T_{half} = T_{short}/2$). It is possible to show that this is:

$$N_{worst} = \sum_{k=0}^{T_{half}} \frac{1}{k!} \left(\frac{T_{half}!}{(T_{half} - k)!} \right)^2 \tag{11}$$

Values of N_{worst} for increasing values of T_{half} are shown in Table 6. We can see that for $T_{half} \leq 6$, the number of merges is at most 13,327. With unoptimized code and full search, we obtained real-time computations for $T_{half} \leq 6$.

We also tried to use larger windows : $T_{half} = 7$ and 8. In practice, we found that, although the computations are slower than real-time, the number of merges in Step 3 very rarely was above 10,000. Even when it is the case, it is possible to design a simple heuristic to prune most of the merges: forbid window partitions Ω including “new” decisions $H^{\Omega}(i,j) = H_1(i,j)$ whenever

$$\frac{p(\theta_i - \theta_j | H_1(i,j))}{p(\theta_i - \theta_j | H_0(i,j))} \leq \epsilon \tag{12}$$

where ϵ is a small value, e.g. 10^{-10}. This pruning is required only in periods where most clusters contain only one element in both past frames and future frames, which happens only when most locations estimates are unrelated to each other. On those periods, oversplitting is therefore not a big loss. On tests with synthetic data (Section 4.3) we obtained the same results with pruning or without pruning.

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 training on the entire data. However, Step 1 can also be implemented online because of the two following reasons:

- Only two single Gaussians are trained, therefore the minimum amount of data needed is small, as compared e.g. with training of a Gaussian Mixture Model. This is verified in practice by experiments in Section 4.3, that are successful on less than 20 seconds of data.

- Once the decisions are made (Steps 2 and 3 finished), a simple online strategy can be designed to update σ_T^{same} and σ_T^{diff}. For each pair (i,j), only one of the two standard deviations is updated, depending on the nature of the decision $H^\Omega(i,j)$. The update is done with a forgetting factor $\alpha \in]0,1[$ (e.g. $\alpha = 0.9$):
$\sigma'_{|t_i-t_j|} \leftarrow \alpha \cdot \sigma_{|t_i-t_j|} + (1-\alpha)\cdot|\theta_i - \theta_j|$

In this Section, we proposed a fully deterministic, threshold-free, online algorithm that analyses in the short-term only. The algorithm is based on a simple hypothesis on local dynamics, and has low computational complexity for reasonable context durations. Section 4 proposes a way to detect and solve low confidence situations, such as trajectory crossings.

4 Confident Clustering

In this Section we derive from the model a confidence measure for each possible individual decision $H_d(i,j)$ (d is 0 or 1), and explain how it allows to detect and solve low confidence situations such as trajectory crossings.

4.1 Confidence Measure

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

$$P(H_d(i,j)|X) \;=\; \sum_{\Omega} \underbrace{P(H_d(i,j)|X,\Omega)}_{\texttt{0 or 1}} \; P(\Omega|X) \tag{13}$$

$$P(H_d(i,j)|X) \;=\; \sum_{\substack{\Omega \\ H^\Omega(i,j)=H_d(i,j)}} P(\Omega|X) \tag{14}$$

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

$$P(H_d(i,j)|X) \;\propto\; \sum_{\substack{\Omega \\ H^\Omega(i,j)=H_d(i,j)}} p(X|\Omega) \tag{15}$$

How to evaluate this value? Practically, it is possible to estimate it by approximating the set of all possible partitions $\{\Omega\}$ with a subset obtained as a by-product of the search algorithm. For example, the sliding window algorithm described in Section 3 constructs such a subset within each analysis window.

4.2 Application to Synthetic Data

We consider "synthetic data", where each location estimate corresponds to an object in the physical world (no meaningless location estimate). In such a case each location estimate belongs to the trajectory of an object. We would like to determine when trajectories cross. In more details, the task is to extract segments of trajectories that are as long as possible, while systematically splitting at each point of trajectory crossing. Indeed, considering objects that can move with very sharp turns, it is dangerous to make any hypothesis on dynamics around the points of trajectory crossings (see examples in Fig. 7).

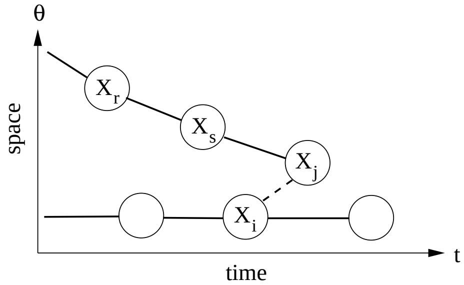

Figure 4: An example of low confidence situation: a trajectory crossing. Each circle is a location estimate. Each ML cluster is depicted by a continuous line. The low confidence $H_0(i,j)$ decision is depicted by a dashed line.

The idea is to compare to each other all decisions made within each analysis window of the sliding window ML algorithm, using the confidence measure defined in 4.1. Whenever a location estimate X_i is involved in a "not confident" decision, we split around X_i.

To do so, in Step 2 and Step 3, we add the following post-processing:

- For all pairs (X_i, X_j) in the analysis window, estimate $P\left(H^{\hat{\Omega}^{ML}}(i,j)\,|X\right)$ using Eq. (15). 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. 4 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:

$$\begin{aligned} H^{\hat{\Omega}^{ML}}(i,j) = H_0(i,j) \quad & and \\ P\left(H_0\left(i,j\right)|X\right) \quad < \quad & M_1\left(\hat{\Omega}^{ML}\right) \end{aligned} \tag{16}$$

where:

$$M_1\left(\hat{\Omega}^{ML}\right) \triangleq \max_{\substack{r<s \\ H^{\hat{\Omega}^{ML}}(r,s) = H_1(r,s)}} P\left(H_1\left(r,s\right)|X\right) \tag{17}$$

In this Section we have proposed to estimate the confidence in a local decision $H_d(i,j)$ as a posterior probability. In the case of synthetic data, we proposed to use this confidence measure in order to detect and solve low-confidence situations such as trajectory crossings. Tests in Section 4.3 will validate this approach, called "confident clustering" hereinafter.

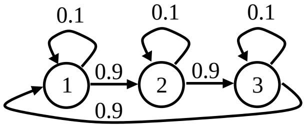

Figure 5: Markov chain used to generate synthetic data. Each state represents an object. "0.1" and "0.9" are transition probabilities. At each time frame, the generated location estimate corresponds to the object indicated by the current state of the chain.

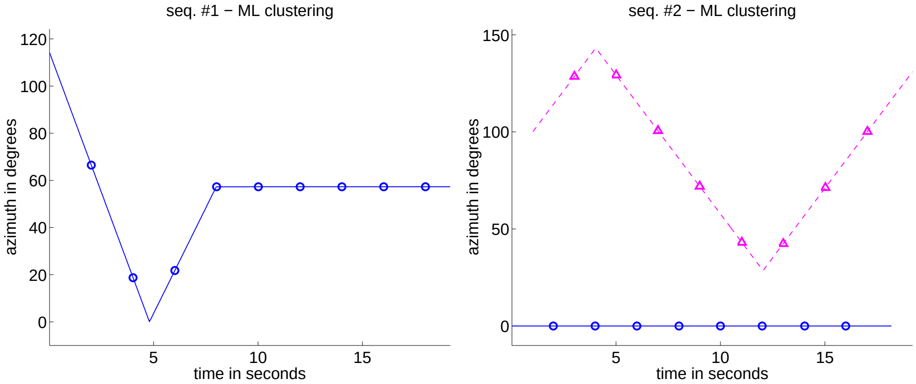

Figure 6: ML clustering for 1-object and 2-object cases. Changes of colors, markers and linestyles indicate beginning and end of clusters. Note that markers do not indicate data points, there are many more data points than markers.

4.3 Multi-Object Tracking Examples

We generated synthetic 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 trajectories, where each piece must belong to a single object. Given that objects can have very sharp turns, this implies that no cluster extends beyond any trajectory crossing.

We generated short sequences of azimuth location estimates (one estimate per frame, 62.5 frames per second) in various cases : one object, two objects or three objects emitting concurrently. In multiple objects sequences, the number of active objects vary with time: different objects appear and disappear at different times. In order to have only one location estimate per time frame, we randomly selected the object to be located at each time frame, generated by a left-to-right Markov chain of 2 or 3 states, all with self-transition probability set to 0.1. The 3-state chain is depicted in Fig. 5. In all experiments reported in this Section, we used full search with $T_{past} = T_{future} = 6$ frames. The unsupervised bi-Gaussian training was done on each sequence separately, in an offline manner.

Fig. 6 shows results on two cases: a single object moving with sharp turns, and two concurrent objects. Each cluster is represented with a set of linked points. We used different colors, markers and linestyles to help distinguish between clusters. Note that markers do not indicate data points, there are many more data points than markers. We can see that the result is correct on both test sequences. On seq. #2 we can see that the ML clustering method copes well with the appearance and disappearance of an object.

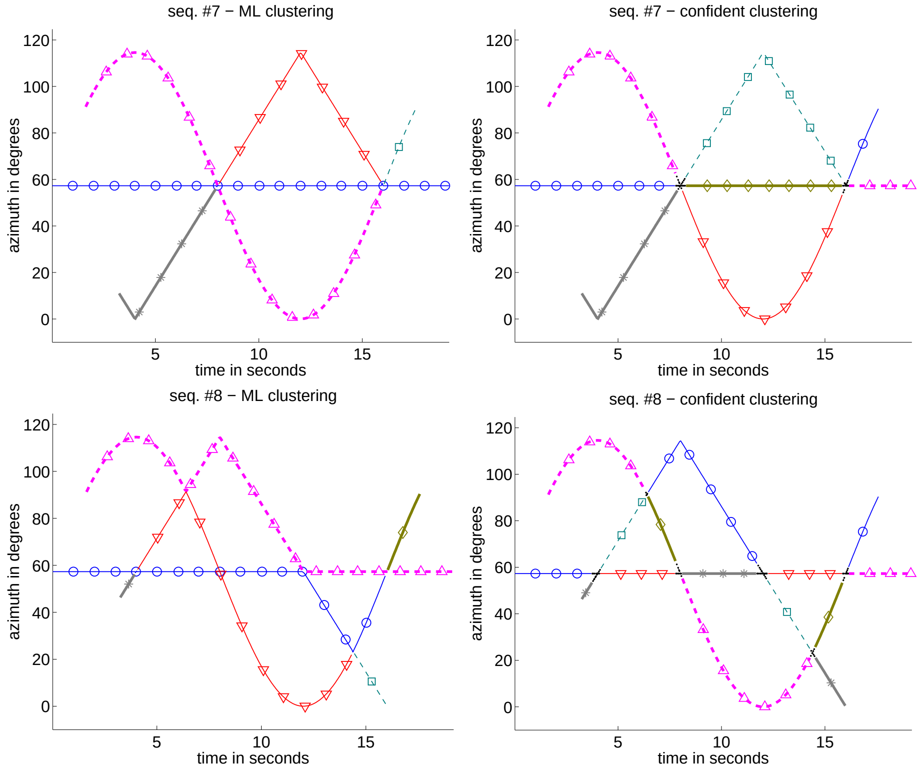

Figure 7: Comparison ML clustering / confident clustering on 3-object cases. We can see that the confident clustering accurately splits the ML clusters at the trajectory crossings. Changes of colors, markers and linestyles indicate beginning and end of clusters. Note that markers do not indicate data points, there are many more data points than markers.

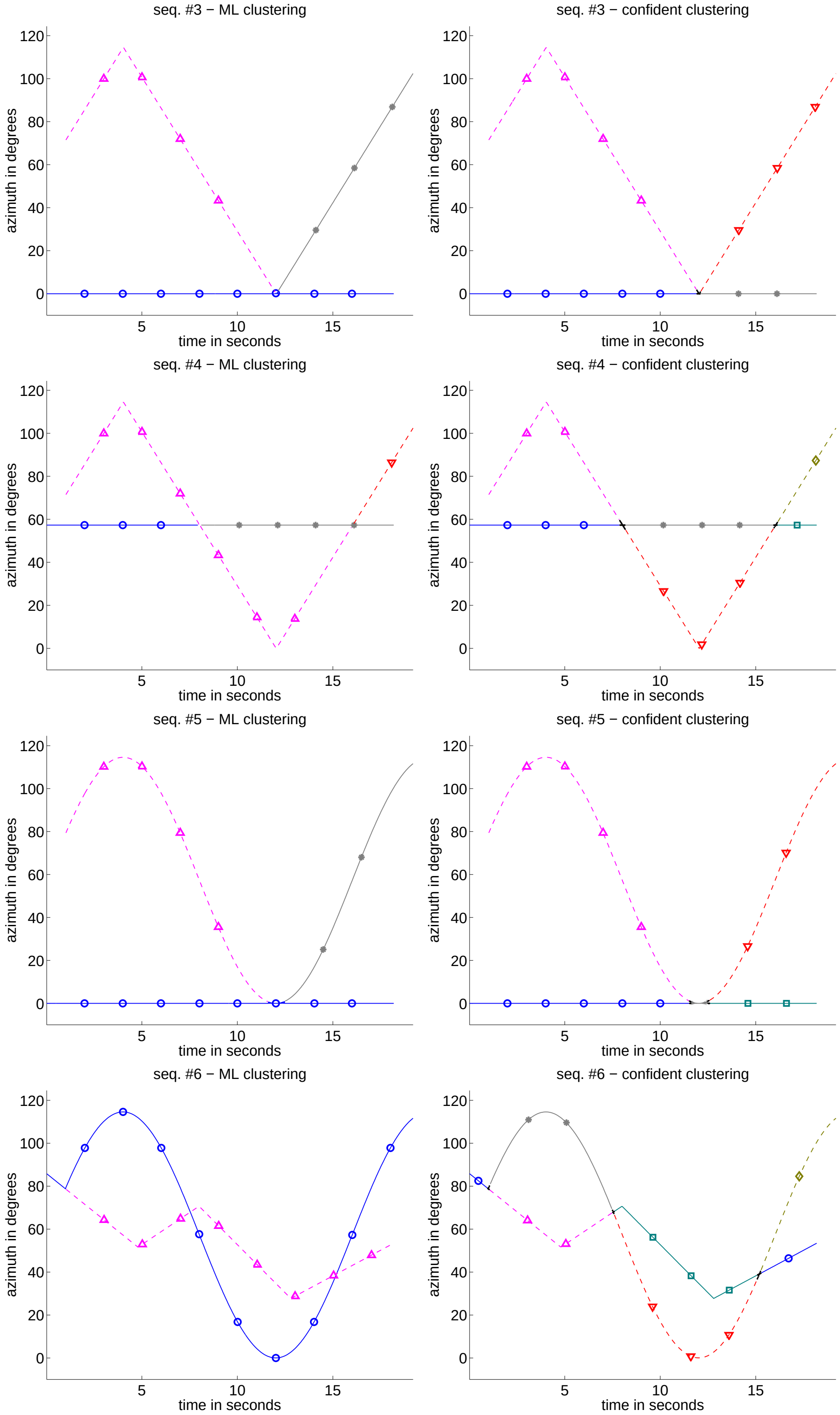

Figure 8: Comparison ML clustering / confident clustering on 2-object cases. We can see that the confident clustering accurately splits the ML clusters at the trajectory crossings. Changes of colors, markers and linestyles indicate beginning and end of clusters. Note that markers do not indicate data points, there are many more data points than markers.

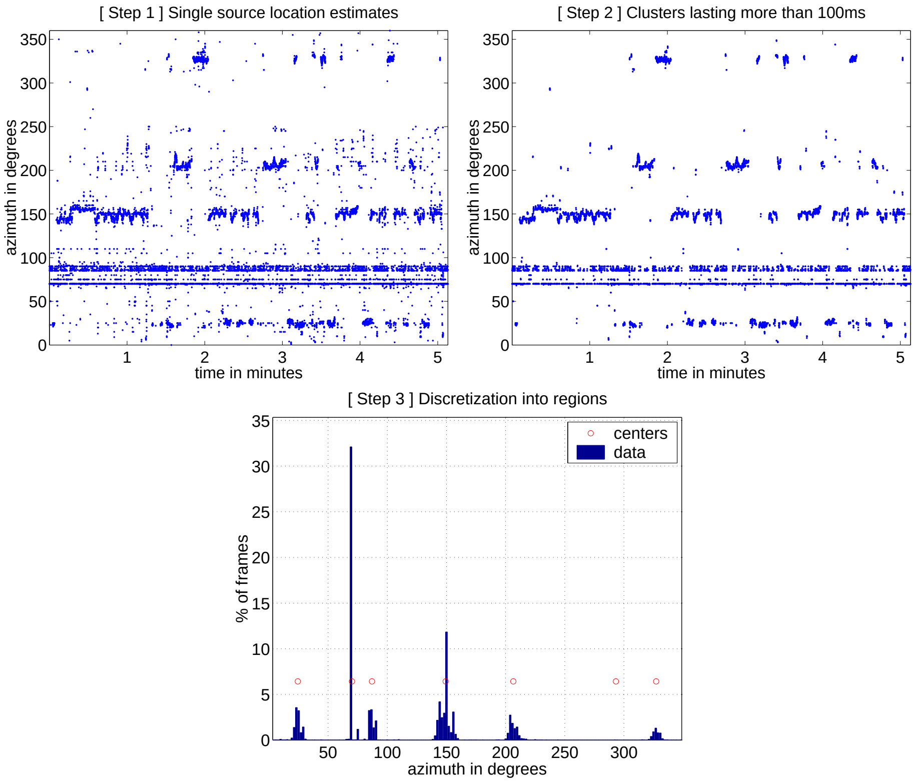

Figure 9: Segmentation algorithm: output of each step

Figs. 7 and 8 compare the result of the ML clustering with the result of the confident clustering described in Section 4.2, in various cases of variable number of objects having several trajectory crossings. The number of active objects varies over time. 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). We noted particularly that the confidence-based approach manages to detect a zero-speed crossing (seq. #5).

5 Real Data: a Meeting Segmentation Application

In this Section we report experiments conducted on real meeting data recorded with one circular microphone array. The task is to provide speech/silence segmentation for each of the few regions occupied by the speakers. We first describe the test data, and the proposed approach, which incorporates the ML short-term clustering algorithm presented in Section 3. We then define performance measures. Finally, results are given that validate the proposed approach and compare it with alternative approaches.

5.1 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

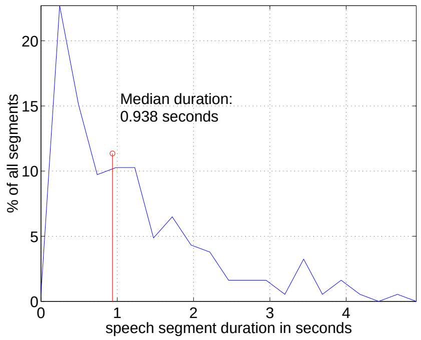

Figure 10: Histogram of speech segment durations in the ground-truth

includes many very short segments. Indeed, 50% of the speech segments are shorter than 0.938 seconds, see Fig. 10.

5.2 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 [11] 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, while the previous approach was segmenting all meetings together.
- The proposed approach does not rely on on a minimum duration to train the models. The previous approach needed a minimum duration of 2 seconds in order to have sufficient data for segmenting the data with GMM models.
- Defining a discrete set of active regions within the room was necessary in the previous work. On the contrary, it is not necessary for the short-term clustering algorithm presented here; regions are used for evaluation purposes only.

We segment each meeting independently, with a 2-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 [10]. Time frames were defined every 16 ms, each time frame containing 32 ms of data.
2. **Short-Term Analysis:** run the ML short-term clustering algorithm described in Table 5 to cluster location estimates X 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.

For evaluation purposes, K-means is applied on the centroids of the remaining short-term clusters ("Step 3"). The product is a list of regions defined by their centers $(\theta^{(1)}, \dots, \theta^{(L)})$, and their respective speech/silence segmentations. L is selected automatically, as in [11]. 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 [12].

Fig. 9 shows the location estimates produced by Step 1, retained by Step 2, and the regions defined by the evaluation Step 3. We note 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 5.5. 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.

5.3 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 did not use the usual window-based definitions. Instead, we defined precision and recall at the frame level. F-measure is defined as:

$$F = \frac{2 \times PRC \times RCL}{PRC + RCL} \tag{18}$$

5.4 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 tried to use Zero-Crossing Rate (ZCR), but 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 microphone array and lapel approaches, maximizing the F-measure.

5.5 Results

Table 7 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 5.2, we can see that the proposed approach fulfills the goal of capturing as many utterances as possible, especially on overlaps (RCL figures in Table 7).

We also compared our approach to a HMM-based previous work [11], on a slightly different task: only 6 meetings are segmented, and the task excludes silences smaller than 2 seconds. To implement this task with the proposed approach, we simply removed silences shorter than 2 seconds from both ground-truth segmentation and result segmentation. Results are reported in Table 8. 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.

	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 7: 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 8: Comparison with previous work: segmentation results on 6 meetings, with silence minimum duration of 2 seconds. Values are percentages.

6 Discussion

In Section 3 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, we described an efficient way of detecting and solving low-confidence situations such as 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 data.

In Section 5 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 4.3. 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 [13] and online methods [14] 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.

7 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, successfully tested on synthetic 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.

8 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. This paper benefited from the valuable comments of Mathew Magimai.-Doss.

References

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

[2] 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*, pages 1628–1632, 1995.

[3] 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.* Multi Sensor Fusion, Tracking and Resource Management II, SPIE, 1997.

[4] J. LaViola. A comparison of Unscented and Extended Kalman Filtering for estimating quaternion motion. In *Proceedings of the 2003 American Control Conference*, pages 2435–2440. IEEE Press, June 2003.

[5] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian bayesian state estimation. In *IEEE Proceedings*, volume 140, pages 107–113, 1993.

[6] 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*, volume 2, pages 1359–1362, 2001.

[7] J.R. Larocque, J.P. Reilly, and W. Ng. Particle filters for tracking an unknown number of sources. *IEEE Transactions on Signal Processing*, 50(12), December 2002.

[8] I.A. McCowan, S. Bengio, D. Gatica-Perez, G. Lathoud, F. Monay, D. Moore, P. Wellner, and H. Bourlard. Modeling human interactions in meetings. In *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-03)*, 2003.

[9] D. Ramanan and D. A. Forsyth. Finding and tracking people from the bottom up. In *Proceedings of the 2003 Computer Vision and Pattern Recognition (CVPR) conference*, June 2003.

[10] J. DiBiase, H. Silverman, and M. Brandstein. Robust localization in reverberant rooms. In M. Brandstein and D. Ward, editors, *Microphone Arrays*, chapter 8, pages 157–180. Springer, 2001.

[11] 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.

[12] 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.

[13] 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.

[14] 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.