A Bayesian stochastic machine for sound source localization

Raphael Frisch*§, Raphaël Laurent ¶, Marvin Faix*, Laurent Girin†‡, Laurent Fesquet§,
Augustin Lux‡, Jacques Droulez‖, Pierre Bessière‖, Emmanuel Mazer*,
Univ. Grenoble Alpes, CNRS, Grenoble INP, Inria, LIG, F-38000 Grenoble France
†Univ. Grenoble Alpes, Grenoble INP*, GIPSA-Lab, F-38000 Grenoble, France
‡INRIA Grenoble Rhône-Alpes, France
§Univ. Grenoble Alpes, CNRS, Grenoble INP*, TIMA, F-38000 Grenoble, France
¶ProbaYes S.A.S, 82 Allée Galillée, 38330 Montbonnot, France
‖CNRS: ISIR/UPMC, 4 place Jussieu, 75005 Paris, France

***Abstract*—Compared to conventional processors, stochastic computing architectures have strong potential to speed up computation time and to reduce power consumption. We present such an architecture, called Bayesian Machine (BM), dedicated to solving Bayesian inference problems. Given a set of noisy signals provided by low-level sensors, a BM estimates the posterior probability distribution of an unknown target information. In the present study, a BM is used to solve a sound source localization (SSL) problem: the BM computes the probability distribution of the position of a sound source given acoustic signals captured by a set of microphones. Assuming free field wave propagation (no reverberations), we express the SSL problem as the maximization of a likelihood function fed with audio features provided by the time-frequency (TF) analysis of the captured audio waves. The proposed BM uses bitwise parallel sampling to fuse the resulting multi-channel information. As the number of channels to fuse is large, the standard BM architecture encounters the so-called "time dilution problem" (long delays are necessary to obtain valid samples). We tackle this problem by using max-normalization of the distributions combined with a periodic re-sampling of the bit streams after processing a reasonably small subset of evidences. Finally, we compare the localization performance of the proposed machine with the results obtained using a standard version of the machine. The re-sampling leads to an impressive acceleration factor of 10^3 in the computation.**

I. Introduction

A. Stochastic machines for probability computation and Bayesian inference

Artificial intelligence and robotics face more and more difficulties to solve problems with incomplete and uncertain knowledge. Logic, the essence of traditional computer science, is not the most appropriate paradigm to do so. On the contrary, probability theory, an extension to logic, may be used as an alternative for rational reasoning under uncertainty [16]. Our general research objective is to imagine and build new programmable machines based on probability rather than on logic. As Moore's law [19] reaches its limit, no more exponential progresses can be expected for computers with conventional architecture. One of our more specific goals is thus to conceive new architectures taking advantages of new nano-devices to solve probabilistic inference problems. Within the former Bambi project,[1] we designed several prototypes of those machines (see e.g. [12] and [7]) leading to a first generation of stochastic machines dedicated to Bayesian Inference. In the on-going MicroBayes project,[2] more complex and realistic problems are tackled. In the present paper, we present a stochastic machine dedicated to localizing a sound source from signals received by several microphones. We show how this problem may be expressed as a Bayesian inference with many evidences. We devise a new stochastic architecture, called the Sliced-BM, which overcomes the slow convergence induced by this large number of evidences.

B. Related work

The goal of an inference machine is to compute a (posterior) probability distribution based on a number of evidences and priors. There are several ways to represent and process probability distributions. On Von Neumann machines, a probability distribution is represented by a set of parameters, such as the mean and the variance for a normal distribution, or by an array of probability values for a histogram, all stacked in memory. In his Bayesian inference machine, Vigoda [28] used analog signals to code probability values and a message passing algorithm to compute exact inferences. Blanche et al. [2] used the intensity of light at different wave lengths to simultaneously represent all the values defining a probability distribution, allowing to multiplex the processing on the same optical hardware. In [18], "Strain switched Magneto Tunneling Junction (SMTJ)" devices are used to code probability values. Two magneto-electric circuits perform additions and multiplications on this representation. Since inference only uses these two types of operation on probability values, one can map any inference into a circuit by spatially organizing these devices on a silicon substrate. Friedman et al. [13] used Muller C-Elements to combine stochastic signals and achieved naive Bayes fusion for binary random variables.

In the above-mentioned architectures, a specialized hardware is dedicated to computing probability values. Another approach is to represent a probability distribution with sequences

*Institute of Engineering Univ. Grenoble Alpes

[1] https://www.bambi-fet.eu

[2] https://persyval-lab.org/en/sites/content/microbayes

of samples. Even with a good entropy source, obtaining "true" samples of a distribution is a complex issue [20]. Designers of Stochastic Programming systems [15] address this issue by defining stochastic programming languages leading to exchangeable stochastic samples. Algorithms can then be run on Von Neumann architectures or transposed at the hardware level. Jonas [17] designed such a specialized machine using standard random bit generators and fixed-point arithmetic to sample integer variables for applications ranging from signal processing to three-dimensional image analysis. One direction of research is to bridge the gap between the circuit and the dedicated entropy generator technology such as the already existing STRNG [5] or the more experimental MTJ [24]. Indeed, the system integration requires today the ability to integrate all the bayesian machine elements on one die. For example, Thakur et al. [27] used stochastic electronics to process hidden Markov models and Bayesian networks. Finally, Faix [11] designed a general purpose sampling machine based on a binary version of the Gibbs sampler [10], [3]. Moreover, contrary to the other above-mentioned architectures, the machine of Faix [11] is programmable: it is not necessary to map a particular program into a particular layout. A compiler translates any Bayesian program [1] into a binary code which becomes the input of a general purpose Gibbs sampler.

In the present paper, we present another type of sampling machine dedicated to sensor fusion problems with many evidences. Coninx et al. [7] proposed an initial architecture for a small number of evidences which was successfully implemented on a Field-Programmable Gate Array (FPGA). Results of fault injection campaigns at the RTL level provide the first evidences of the intrinsic robustness of such architectures [6].

C. Paper outline

This paper is organized as follows. Section II gives a presentation of the Sound Source Localization (SSL) problem and how it can be addressed with a stochastic machine. It also reminds the initial Standard-BM architecture and the issues about convergence when too many evidences are at hand. Section III presents the improved BM architecture applied to SSL. In particular, it presents the solution to address the temporal dilution problem, i.e. the re-sampling mechanism used to regenerate the stochastic signal associated to the max-normalization process. Experimental results are presented and discussed in Section IV. A comparison with the previous Standard-BM architecture is provided. Finally, some conclusion is drawn from this work and the ongoing work is stated.

II. Background

A. Sound source localization

Sound Source Localization (SSL) consists in estimating the location of a sound source in a given environment from recorded multichannel signals emitted by this sound source. This problem has been extensively studied in the acoustic signal processing community. Recent works deal with probabilistic models designed to link inter-channel acoustic features extracted from the sensor signals to the source position, e.g. [22], [26], [23], [29], [8], [9], [21]. These models generally combine a physical model of wave propagation with machine learning techniques (mostly Gaussian mixture models).

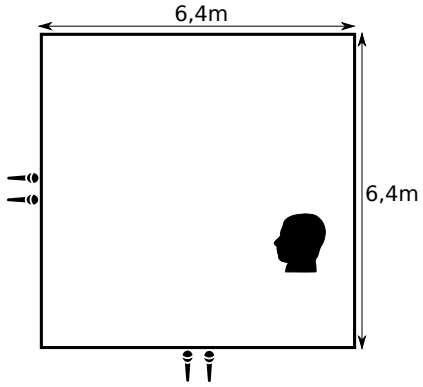

Fig. 1. Schema of the proposed sound source localization setup.

The configuration and the recording set-up considered in the present study are illustrated in Fig. 1. The sound source is a person speaking in a large room (6.4 m $\times$ 6.4 m $\times$ 3.1 m). The person is assumed to be still, but a set of experiments were conducted with different static source positions. The recording set-up, inspired from [9], is composed of two pairs of microphones placed in the center of the walls. In the present study, we target SSL in the first two dimensions of the room.

Straightly stated, SSL is based on the analysis of delays between the signals received on each pair of microphones [4]. The combination of delay information from all microphone pairs is closely related to the 2D source position [9]. We assume that the acoustic propagation follows a free-field model, i.e. the microphones are omni-directional and fixed on stands that have no effect on sound propagation (in other words, all microphones are "floating" in the room). Also, the reverberations on the walls are assumed to be negligible, and the source to microphone distance L is large enough to consider the acoustic waves reaching the microphones as plane waves. Let us first consider one pair of microphones for simplicity of presentation. Let $y_1(t)$ and $y_2(t)$ denote the recorded signals on channel 1 and 2 respectively. With the above assumptions, both sensor signals are attenuated and delayed versions of the speech signal $s(t)$ emitted by the speaker, and $y_2(t)$ is a delayed version of $y_1(t)$:

$$\begin{aligned} y_1(t) &= a \cdot s(t - t_s), \qquad \text{with} \quad 0 < a < 1, \\ y_2(t) &= y_1(t - t_0). \end{aligned} \tag{1}$$

The delay t_0 corresponds to the wave path difference between the two microphones (we assume that the attenuation on this part of the wave path is negligible). It depends on the azimuth θ of the source, which is defined as the angle between the axis perpendicular to the inter-microphone axis and the source direction (see Fig. 2). Assuming the source-to-microphone distance L much larger than the the inter-microphone distance d, we have:

$$t_0 = \frac{d}{C} \sin(\theta), \tag{2}$$

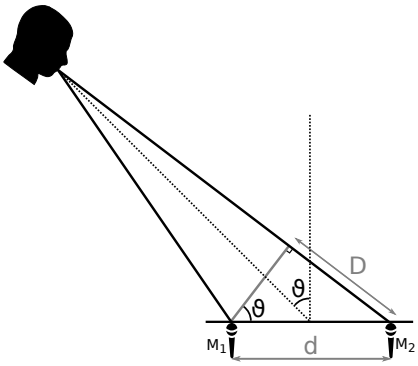

Fig. 2. Schema of the source-to-microphones wave propagation.

where C is the speed of sound ($\approx$ 340 m.s^{-1} in the air). Therefore, a measure of t_0 (or an equivalent information) can lead to an estimation of the source azimuth. Merging the azimuth information provided by several microphone pairs (at least two) can then lead to an estimate of the absolute source position. This principle is formalized below within a probabilistic model, after we further characterize the link between t_0 and the sensor signals.

To this aim, the microphone signals are sampled at sampling frequency $f_s = 1/T_s$ and we calculate their Short-Time Fourier Transform (STFT), i.e. a sequence of Discrete Fourier Transforms (DFT) calculated on a sliding analysis window [25]:

$$\begin{aligned} Y_1(k,l) &= \text{STFT}(y_1(nT_s)) = \sum_{i=0}^{N-1} y_1((i+lH)T_s)e^{-j2\pi\frac{ik}{N}} \\ Y_2(k,l) &= \text{STFT}(y_2(nT_s)) = \sum_{i=0}^{N-1} y_2((i+lH)T_s)e^{-j2\pi\frac{ik}{N}}, \end{aligned} \tag{3}$$

where k, l are the frequency and time-frame indexes, N is the size of the analysis window, and H is the size of the window shift. Inserting model (1) into (3), we obtain:

$$Y_2(k,l) \approx Y_1(k,l)e^{-j2\pi\frac{kt_0}{NT_s}}. \tag{4}$$

Eq. (4) is an approximation mainly because of the finite size of the STFT window, but if model (1) holds, (4) is a very good approximation in practice for STFT bins that contain a significant amount of energy.

Furthermore, we define a 64×64 regular grid of 2D source positions within the room (one position every 10 cm in both dimensions). For every candidate source position (x, y), we calculate the corresponding source azimuth $\theta_m(x, y)$ with respect to each microphone pair indexed by m (in the present study we use two pairs of microphones, and we set $d_m = d$ for $m \in \{1, 2\}$). The corresponding (theoretical) candidate delay is given by (2) with $\theta = \theta_m(x, y)$. Using (4) the corresponding (theoretical) inter-channel STFT coefficient ratio $R_m(k, l)$ for microphone pair m is given by:

$$R_m(k,l) = \frac{Y_{2,m}(k,l)}{Y_{1,m}(k,l)} \approx e^{-j2\pi\frac{kd}{NCT_s}\sin(\theta_m(x,y))}. \tag{5}$$

From (5), we have:

$$\phi_m(k,l) = \arg R_m(k,l) \approx -2\pi\frac{kd}{NCT_s}\sin(\theta_m(x,y)). \tag{6}$$

This motivates the following probabilistic model, in the spirit of the previous works on SSL cited above, in which $\phi_m(k, l)$ is often referred to as the Inter-channel Phase Difference (IPD). Let $\phi_m^{\text{meas}}(k, l)$ denote the argument of the inter-channel ratio calculated on the measured sensor signals (in short $\phi_m^{\text{meas}}(k, l)$ is the measured IPD). Given an acoustic source emitting from position (x, y), $\phi_m^{\text{meas}}(k, l)$ is assumed to follow a Gaussian distribution centered on $\phi_m(k, l)$ (given by (6)) and of arbitrary variance σ_ϕ^2:

$$\phi_m^{\text{meas}}(k,l) \sim \mathcal{N}(\phi_m(k,l), \sigma_\phi^2). \tag{7}$$

Note that the choice of having a variance that is independent of k and l is just an arbitrary simplifying assumption, which does not prevent this model to work well. In practice, (6) and thus (7) hold for all STFT bins that contain significant signal energy. We thus have a large set of STFT ratios that inform about the source location. However, since $\phi_m^{\text{meas}}(k, l)$ is a phase measure calculated from sensor signals, it actually consists of a principal angle value within $[0, 2\pi[$. To ensure that (6) and (7) are not spoiled by phase ambiguity, a simple solution consists in i) setting d to a small value to minimize t_0, and ii) given d and other parameters, selecting the low-frequency bins for which t_0 is assumed to be less than a period of the spectral component,[3] i.e:

$$\frac{kd}{NCT_s} \ll 1 \Leftrightarrow k \ll \frac{NCT_s}{d}. \tag{8}$$

In practice, we set $d = 5$ cm and (8) is verified for a large range of frequency bins k.

In summary, our probabilistic model for SSL consists of a series of distribution values

$$p(\phi_m^{\text{meas}}(k,l)|x,y) = \frac{1}{\sqrt{2\pi}\sigma_\phi}\exp\left(-\frac{(\phi_m^{\text{meas}}(k,l)-\phi_m(k,l))^2}{2\sigma_\phi^2}\right) \tag{9}$$

conditioned on source position (x, y) through (6), and evaluated i) for each point of the 64×64 source position grid, ii) for a series of low-frequency TF bins where the sensor signals are assumed to have significant energy and (8) holds.

In the next section, values of (9) represent the evidences of the Bayesian fusion model. Because the BM machine uses probability values corresponding to discrete variables, in our practical implementation the measured IPDs $\phi_m^{\text{meas}}(k, l)$ are actually quantized (with a resolution that is appropriate for the SSL problem), and values of the continuous distribution (9) are turned into probability values. This is further detailed in the next section.

[3] Another solution would be to use a circular distribution such as the von Mises distribution instead of (7), but in the present study it is very easy to ensure (8) and thus using (7) is fine.

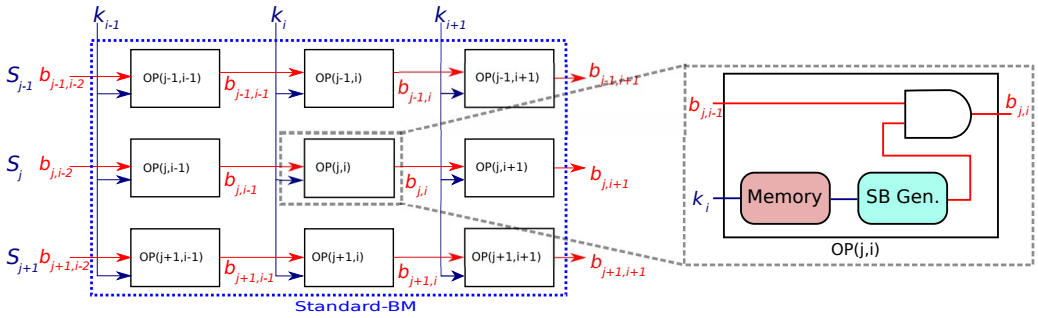

Fig. 3. Architecture of the Standard-BM including a zoom on a single OP block of the machine. Bit-streams are represented by red arrows. Blue arrows illustrate fix point values.

B. Bayesian machines

1) Bayesian fusion: The goal of our work is to propose new architectures to compute posterior probability distributions from a Bayesian model. Let us consider a discrete searched variable S, a discrete known variable K, and their joint distribution $P(S \wedge K)$. S and K can be themselves conjunction of variables. The inference over n known variables is done using the Bayes rule:

$$P(S|K_1,\ldots,K_n) = \frac{1}{Z}P(S)\prod_{i=1}^{n}P(K_i|K_1,\ldots,K_{i-1},S) \tag{10}$$

where $P(S)$ is the prior, $P(K_i|K_1,\ldots,K_{i-1},S)$ are the conditional distributions and Z is the normalization constant. Notice that the inference is made by multiplying the terms. Hence no need of addition operation. In the case of naive Bayesian fusion, each conditional distribution is seen as a likelihood of independent sensor variables (a so-called evidence) and (10) simplifies to:

$$P(S|K_1,\ldots,K_n) = \frac{1}{Z}P(S)\prod_{i=1}^{n}P(K_i|S). \tag{11}$$

In the SSL problem, the searched variable S is equal to (X,Y), i.e. the discrete Cartesian coordinates of the source in the room (or equivalently, S is the index of the source position on the localization grid). Each known variable K_i is a quantized version of the measured IPD $\phi_m^{\text{meas}}(k,l)$ for a given frequency bin k and for a given microphone pair m (SSL will be performed independently for each time frame l). The corresponding conditional discrete distribution $P(K_i|S)$ is evaluated using (9). More specifically, during the design of the BM, a large codebook of quantized (prototype) values of $\phi_m^{\text{meas}}(k,l)$ is first used to compute a large set of probability values $P(K_i|S)$ corresponding to (9). These latter are stored in memory. Then, during actual SSL, quantized values of the measured IPDs obtained from sensor signals are used as inputs to the BM: they index the corresponding evidence values $P(K_i|S)$ in memory. After this preliminary mapping between sensor information and evidence values, the problem of the BM is the calculation of (11).

2) Stochastic bit stream representation: The Bayesian machine is based on computing (11) by sampling each term. Probability values are encoded by streams of stochastic bits [14], drawn from a Bernoulli distribution. Each sample 0 or 1 represents $p = P(X = x_i)$. Discrete temporal integration over n_T steps gives an approximation of p: this is done by counting the number n_1 of 1 and dividing by n_T, so we have:

$$\frac{n_1}{n_T} \xrightarrow[n_T\to\infty]{} p. \tag{12}$$

The main operation to carry out in (11) is the product between different evidences and between prior and evidences. The bit stream representation can perform a probability product computation with a simple AND gate. Indeed, let p_1 and p_2 be two probability values respectively encoded by their bit stream chain B_1 and B_2. The chain B_3 resulting from applying an AND gate over B_1 and B_2 encodes the probability $p_3 = p_1 \times p_2$.

3) Standard Bayesian Machine: In this subsection, we rapidly present the architecture of the standard BM, on which we build the improved BM. More details can be found in [7].

Let us focus on a specific value of the discrete search variable $S = s_j$. From (11), the machine computes :

$$P(S = s_j|k_1,\ldots,k_n) = \frac{1}{Z}P(S = s_j)\prod_{i=1}^{n}P(k_i|S = s_j). \tag{13}$$

This computation (for index j) is represented on the second line of the left part of Fig. 3. Indeed, as shown in this figure, the architecture of the machine is shaped as a matrix of elementary blocks, each block representing an individual probability product operator. Each input obtained from the sensor signals (i.e. a quantized value of measured IPD for a given frequency bin and a given microphone pair, represented by the value k_i) is sent to every block of a given column. Hence, the different columns correspond to different inputs (different frequency bins and microphone pairs). The different lines of the matrix correspond to the different values of the search variable S. A block at position (j,i) in the matrix takes as inputs: i) the quantized IPD value k_i (or equivalently the index representing this value in the codebook of quantized IPDs) as stated above, and ii) the bit stream $b_{j,i-1}$ representing the product (13) up to index $i-1$, which is the output of the previous column for the same line. The right part of Fig. 3 details the basic operator (OP). It is composed of

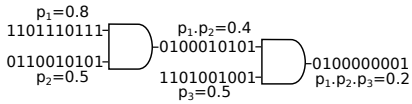

Fig. 4. An example of temporal dilution multiplying 3 probabilities.

the AND gate to perform the product between $b_{j,i-1}$ and the bit stream representing the evidence value corresponding to k_i. As mentioned before, this evidence value is read in the memory where all evidence values are stored. Then, a stochastic bit stream generator (SBG) generates samples with Bernoulli distribution corresponding to the evidence value. The output of the AND gate is the new bit stream $b_{j,i}$. In summary, each column-wise process updates the current state of knowledge with a new evidence.

The machine generates samples at each calculation step, for each value of the searched variable. For each line, i.e for each searched value, the result of the n cascaded AND gates after m calculation steps is stored in counters. This discrete temporal integration allows to recover the target distribution. Indeed, the output of the machine is the normalized values of the counters $\frac{counter_j}{\sum_l counter_l}$ which are the approximate values of the searched distribution $P(S = s_j|k_1, \ldots, k_n)$.

III. Tackling temporal dilution with the Sliced-BM

A. The temporal dilution problem

Because of the data representation, when bit streams go through AND gates, this inevitably leads to a decrease of the number of "1" in the output. We call this effect the temporal dilution. In particular, since we mainly deal with low probability values, the bit stream representation requires long streams to represent such low values. Fig. 4 illustrates this problem. The product between $p_1 = 0.8$, $p_2 = 0.5$ and $p_3 = 0.5$ is performed with two cascaded AND gates. The resulting bit stream, encoding $p_1 \times p_2 \times p_3 = 0.2$, is composed of only two "1" in a chain of 10 bits.

B. Max-normalization

To address this problem, we first propose to maximize each set of probability values (priors and evidences) while keeping unchanged the ratios between the different values. To this aim, each prior value is normalized by the maximum of prior values and the same is done for the evidences:

$$\begin{aligned} \text{Priors} &: \frac{P(S = s_j)}{\underset{s \in S}{Max}\{P(S = s)\}}, \\ \text{Evidences} &: \frac{P(K_i = k_i|S = s_j)}{\underset{s \in S}{Max}\{P(K_i = k_i|S = s)\}}. \end{aligned} \tag{14}$$

This process is referred to as max-normalization. At the end of each line of the BM "matrix", one can evaluate:

$$\begin{aligned} &P(S = s_j|k_1, \ldots, k_n) \\ &= \frac{1}{Z} \frac{P(S = s_j)}{\underset{s \in S}{Max}\{P(S = s)\}} \prod_{i=1}^{n} \frac{P(k_i|S = s_j)}{\underset{s \in S}{Max}\{P(k_i|S = s)\}}. \end{aligned} \tag{15}$$

Therefore, the "true" posterior probabilities can be obtained by conventional normalization of (15).[4]

To demonstrate the efficiency of the max-normalization, let us take a simple example with a uniform distribution on priors and evidences. For simplicity, let m be here the dimension of the search space and let n be the number of evidences corresponding to the number of columns of the matrix. The probability of generating a "1" for each prior or each evidence is $p = \frac{1}{m}$. Through the n AND gates, the probability for each line to finally emit a "1", and fill the corresponding counter, is $p_{out} = \frac{1}{m^{n+1}}$. This probability quickly tends towards "0" even with small values for n and m. In this case, the machine needs a very long time to obtain useful information incrementing the counters. Yet, in this example, the important information is that all counters encode the same value. Using the max-normalization subtlety over both priors and evidences, each normalized probability value becomes equal to 1. Then, all tiles of the matrix output a bit stream only composed of "1" encoding the value 1. At each step computation, all counters are incremented, hence the values of the different counters remain equal. To get the approximate value of $P(S = s_j|k_1, \ldots, k_n)$, again, we compute the ratio $\frac{counter_j}{\sum_l counter_l} = \frac{counter_j}{m \times counter_j} = \frac{1}{m}$ which is the expected result of the uniform law.

C. Sliced BM architecture with dynamic re-sampling

Based on the previous considerations, we propose an improved BM, so-called the Sliced-BM, which architecture is shown in Fig. 5. It is composed of multiple Standard-BM (see Section II-B3) of limited number of columns, called slices, with a re-sampling unit between each pair of consecutive slices. Fig. 5 presents a Sliced-BM with two slices. In the experiment section, we will run a Sliced-BM with 10 slices to solve our SSL problem.

To limit the temporal dilution, we apply the max-normalization described in Section III-B over all probability distributions. The max-normalization allows to have at least one tile per column with only "1s" as inputs which maximizes the probability of having a "1" at the corresponding output. However, when the number of evidences, and hence the number of columns of the matrix, becomes large, the temporal dilution problem is still present. The concept of dynamic re-sampling is used to tackle this problem. It consists in regenerating the stochastic signal after a subset of groups of evidences (i.e. here after a slice). In the same way that the Standard-BM uses an output counter bank to store the samples of the target distribution, the Sliced-BM uses a counter bank in each re-sampling unit to regenerate signals "with more 1s". To implement the max-normalization into each re-sampling unit, as shown in Fig. 6, we set a re-sampling threshold (RT) value for all counters. When a counter reaches this value, the machine activates the process for the next slice, with prior probability $p_j = \frac{counter_j}{\text{RT}}$ for line j. The re-sampling unit

[4]For an application such as SSL where we look for the maximum over j of $P(S = s_j|k_1, \ldots, k_n)$, this final normalization is not even necessary.

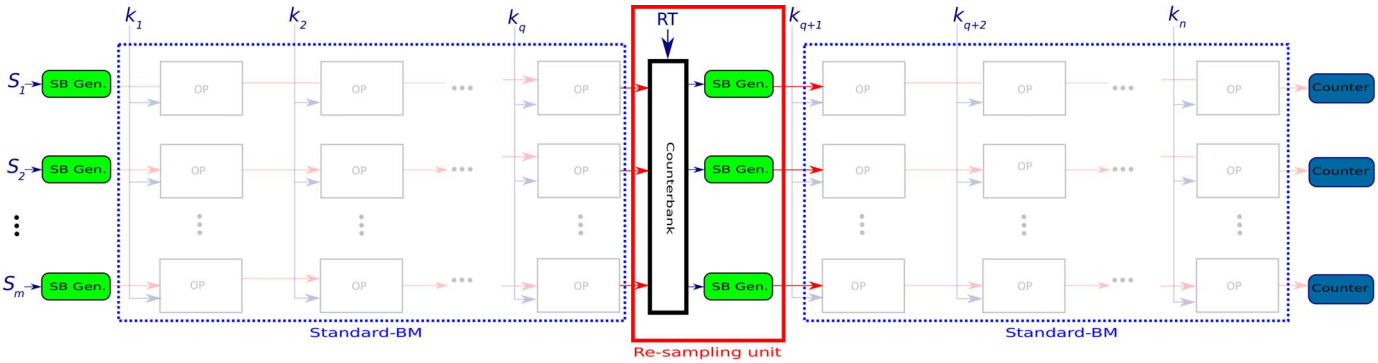

Fig. 5. Architecture of a Sliced-BM with 2 slices of $q = n/2$ columns each and 1 re-sampling unit between them.

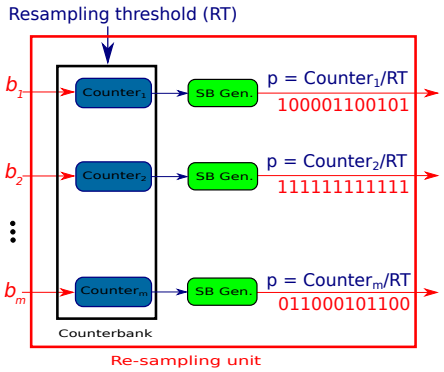

Fig. 6. Re-sampling unit.

maximizes the output probability of a slice, and allows to send a maximum number of "1s" (useful information) as input of the next slice. We illustrate the particular case of a line reaching the threshold value in Fig. 6: The stochastic bit stream generated on line 2 is composed of only "1s".

IV. Experimentations

This section presents the experimentations conducted to evaluate the proposed Sliced-BM architecture. To this aim, a high-level simulator of the Sliced-BM architecture has been implemented in C++ and Python, simulating the different components and connections of the BM.

A. Experimental setup

1) Signals and SSL setup: The source (speech) signals were mono-channel 16-kHz signals. The sensor signals $y_{i,m}(t)$ were generated using the room impulse response (RIR) simulator of AudioLabs Erlangen and a reverberation time of 150ms.[5] The IPDs were calculated using the Short-Time Fourier Transform (STFT) as stated in Section II-A, with $N = 1{,}024$. As already mentioned in Section II-A, the 6.4 m $\times$ 6.4 m room has been discretized in 10 cm $\times$ 10 cm tiles, leading to a grid of $64 \times 64 = 4{,}096$ candidate source positions (x, y). The set of quantized measured IPDs and theoretical IPDs were used to calculate the evidences $P(K_i|S)$ of the Sliced-BM for all candidate source positions. In order to deal with uncertain sensors we have chosen a fairly high value $\sigma_\phi^2 = 0.75$ rad^2 for the variance in (9).

A first round of experimentations with the sound source placed in the center of the room provided very good localization results with both the Standard-BM and the Sliced-BM. Therefore, we choose to present in detail the results obtained for a more difficult case, in which the temporal dilution problem becomes a real problem. In this case, the speaker is placed at $(x, y) = (4.8$ m$, 1.6$ m$)$, as sketched in Fig. 1.

2) Sliced-BM configuration: In the Sliced-BM, each line j corresponds to a candidate position in the room. Thus, the Sliced-BM has $4{,}096$ lines. The column inputs k_i of the Sliced-BM are the measured IPDs for the different frequency bins k and microphone pairs m. To limit computations, we selected a range of frequency bins where i) the voice spectrum generally has high energy, and ii) Eq. (8) is verified. We have chosen the frequency band [200 Hz, 1 kHz], corresponding to bins 12 to 64 of the STFT. As we have 52 frequency bins and 2 pairs of microphones, there are $52 \times 2 = 104$ columns in the Sliced-BM. The size of each slice has been set to 10 columns. The re-sampling threshold has been set to 128. Thus each re-sampling unit waits for a counter to reach 128 before re-sampling the stochastic signal and activating the next slice of the machine.

B. Results

In this section, we present the experimental results. First, we characterize the localization performance of the Sliced-BM with the obtained probability map. Second, the Sliced-BM and the Standard-BM are compared using the Kullback-Leibler divergence.

1) Localization performance: Fig. 7 shows the probability map obtained after running the Sliced-BM for $5{,}000{,}000$ steps. Typically, running the machine for such a large number of steps provides a probability map with high precision. The green point gives the position of the maximal counter, i.e. the line of the Sliced-BM with the highest counter, representing the maximum of the probability distribution. The real position of the source is given by the red point. Clearly the localization is not perfect, though the two points are globally in the same region. It is important to note that the localization obtained

[5] www.audiolabs-erlangen.de/fau/professor/habets/software/rir-generator

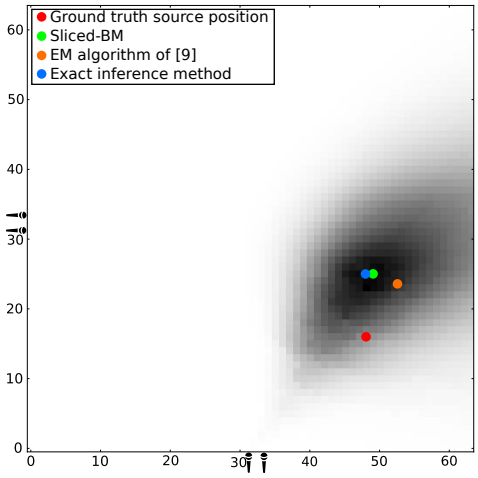

Fig. 7. Probability map for Sliced-BM at 5.000.000 steps. Maximum points of Ground truth source position (red), Source position estimated by the Sliced-BM (green) which stays stable after 10.000 steps, Source position estimated by the EM algorithm of [9] (orange), Source position estimated by exact inference method (blue).

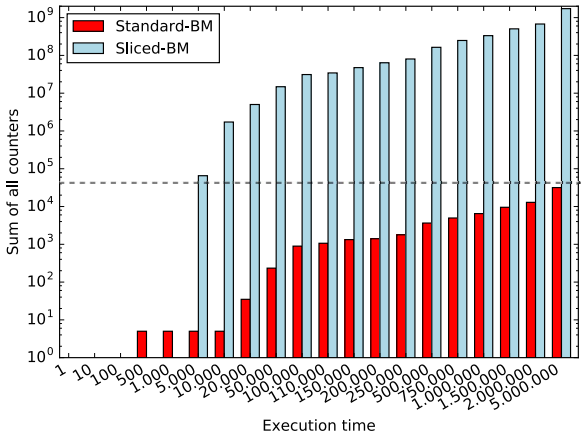

Fig. 8. Sum of all output counters (on a log-scale) as a function of the number of computation steps.

with the Sliced-BM is similar to the one obtained with the SSL state-of-the-art algorithm of [9] using the same setup (shown as orange point in Fig. 7). It also is very close to the localization provided by exact inference (i.e. floating-point calculation of (11); blue point; of course all methods were fed with the same sensor information). Therefore, the difference between estimated and true localization is not due to the accuracy of the inference of the Sliced-BM but it is rather due to i) the low amount of used STFT frames, ii) the limited number of microphone pairs, and iii) the approximations made by the SSL model are not sufficient for this situation. In particular, room reverberations which are neglected in the model probably perturb the localization, since the speaker is located relatively close to the walls.

2) Computation speed: Since the distributions provided by Bayesian machines are approximate and the accuracy increases with computation time, it is interesting to inspect the results at different computation steps. In this section, the computation speed and output accuracy of the two BM versions is compared. First, Fig. 8 displays the sum of all output counters of the machine as a function of the number of computation steps. This plot illustrates the temporal dilution since it shows how many "1s" are produced at the output of the machine. One can observe that both architectures have a certain "warm-up" time before providing a first significant approximate result. The Sliced-BM bars (lightblue) raise considerably between $1{,}000$ and $5{,}000$ steps. This is due to the re-sampling threshold (RT) which is set to 128. Since we have 10 slices in the machine, we need to wait at least $10 \times 128 = 1{,}280$ steps before getting a first output. The Standard-BM architecture (red bars) exhibits a much longer warm-up time which is clearly due to the time dilution problem. Moreover, the sum of all output counters reaches a much higher value after $5{,}000{,}000$ steps with the Sliced-BM (about $1{,}728{,}000{,}000$) than with the Standard-BM (only $31{,}741$).

Now, we inspect the accuracy of the posterior probability distribution computed by the BMs, denoted P_{exp}. To this aim, we first calculate the theoretical probability distribution P_{th} corresponding to an exact inference method, directly combining (9) and (11). Then we calculate the Kullback-Leibler divergence (KLD) between P_{exp} and P_{th}:

$$D_{KL}(P_{th}, P_{exp}) = \sum_i P_{th}(i) \log \frac{P_{th}(i)}{P_{exp}(i)}. \qquad (16)$$

The KLD is a classical measure of the "distance" between two probability distributions.[6] Fig. 9 displays the KLD values for both Standard-BM and Sliced-BM, as a function of the number of computing steps. Note that, since each BM needs some "warm-up" time as shown in Fig. 8, the plotted bars start at the number of steps where the machine computed enough useful information for comparison with the exact inference. The KLD value difference between Standard-BM (in red) and Sliced-BM (in lightblue) is due to the re-sampling method which strongly reduces the temporal dilution. In the Standard-BM, the number of "1s" present at the end of all columns (and incrementing the final counters) is very low. However, in the Sliced-BM, the number of "1s" is much higher so is the number of incremented counters at the output level. This allows to obtain a much faster and a much better approximation of the target distribution. For example, after only $5{,}000$ steps the Sliced-BM nearly gets the same KLD value as the Standard-BM after $5{,}000{,}000$ steps, hence an impressive acceleration factor of 10^3.

V. Conclusion & Future Work

In this work, a new architecture for a Bayesian machine has been presented. This architecture tackles the temporal dilution problem by combining max-normalization and re-sampling

[6]Although not symmetric, the KLD behaves as a distance. In particular it is positive and a KLD equal to 0 indicates that the two distributions are identical.

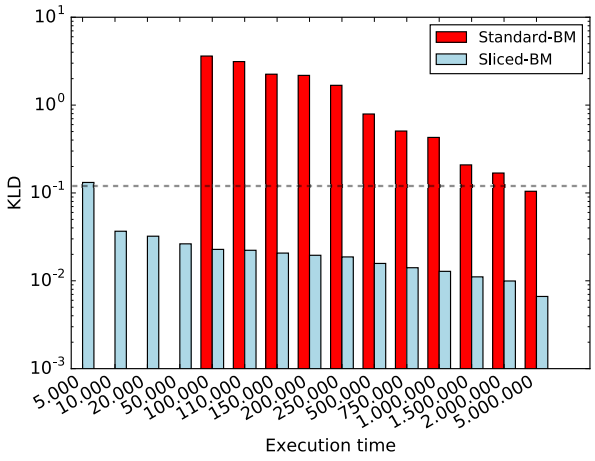

Fig. 9. Comparison of the Kullback-Leibler divergence between the distribution computed by the BM (standard and sliced version) and the exact distribution as a function of the number of computation steps.

of the probability distribution calculated after integrating a subset of evidences. This technique provides very good results compared to the existing Standard-BM architecture, as shown by SSL experiments. An acceleration by a factor of 10^3 in the computation has been obtained.

As future work, we consider improving the architecture using a filter-like approach to perform the localization using multiple STFT frames and also support localization of moving sound sources. Moreover, we will study the effect of varying the number of columns in the slices and the value of the re-sampling threshold. Furthermore, a measurement technique may be developed to provide the user with an insight of the localization quality depending on the content of the sensor signals in the different time frames. Finally, the proposed Sliced-BM architecture will be integrated on an FPGA to get an analysis of the resource consumption and perform faster and more realistic simulations.

Acknowledgment

This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01).

References

[1] P. Bessière, E. Mazer, J. M. Ahuactzin, and K. Mekhnacha, *Bayesian programming*. CRC Press, 2013.

[2] P. Blanche, M. Babaeian, M. Glick, J. Wissinger, R. Norwood, N. Peyghambarian, M. Neifeld, and R. Thamvichai, "Optical implementation of probabilistic graphical models," 2016.

[3] R. Canillas, R. Laurent, M. Faix, D. Vaufreydaz, and E. Mazer, "Autonomous robot controller using bitwise gibbs sampling," in *IEEE Trans. Cogn. Inf. & Cogn. Comp.*, 2016, pp. 72–76.

[4] J. Chen, J. Benesty, and Y. Huang, "Time delay estimation in room acoustic environments: an overview," *EURASIP J. on Appl. Signal Process.*, vol. 2006, pp. 170–170, 2006.

[5] K. Cherkaoui, V. Fischer, A. Aubert, and L. Fesquet, "A very high speed true random number generator with entropy assessment," *Workshop on Cryptographic Hardware and Embedded Sys.*, pp. 179–196, 2013.

[6] A. Coelho, R. Laurent, M. Solinas, Jr., J. Fraire, E. Mazer, N. E. Zergainoh, S. Karaoui, and R. Velazco, "On the robustness of stochastic bayesian machines," *IEEE Trans. Nucl. Sci.*, vol. PP, no. 99, pp. 1–1, 2017.

[7] A. Coninx, P. Bessière, E. Mazer, J. Droulez, R. Laurent, M. A. Aslam, and J. Lobo, "Bayesian sensor fusion with fast and low power stochastic circuits," in *Proc. of IEEE Int. Conf. on Rebooting Computing*, 2016.

[8] A. Deleforge, R. Horaud, Y. Y. Schechner, and L. Girin, "Co-localization of audio sources in images using binaural features and locally-linear regression," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 4, pp. 718–731, 2015.

[9] Y. Dorfan and S. Gannot, "Tree-based recursive expectation-maximization algorithm for localization of acoustic sources," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 10, pp. 1692–1703, 2015.

[10] M. Faix, R. Laurent, P. Bessière, E. Mazer, and J. Droulez, "Design of stochastic machines dedicated to approximate bayesian inferences," *IEEE Trans. Emerg. Topics Comput.*, vol. PP, no. 99, 2016.

[11] M. Faix, "Design of stochastic machines dedicated to Bayesian inferences," Thèses, Université Grenoble Alpes, 2016. [Online]. Available: https://hal.archives-ouvertes.fr/tel-01451857

[12] M. Faix, E. Mazer, R. Laurent, M. O. Abdallah, R. Le Hy, and J. Lobo, "Cognitive computation: a bayesian machine case study," in *IEEE Trans. Cogn. Inf. & Cogn. Comp.* IEEE, 2015, pp. 67–75.

[13] J. S. Friedman, L. E. Calvet, P. Bessière, J. Droulez, and D. Querlioz, "Bayesian inference with muller c-elements," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 6, pp. 895–904, 2016.

[14] B. Gaines, "Stochastic computing systems," in *Advances in information systems science*. Springer, 1969, vol. 2, pp. 37–172.

[15] N. Goodman, V. Mansinghka, D. Roy, K. Konawitz, and D. Tarlow, "Church: a language for generative models," *arXiv preprint arXiv:1206.3255*, 2012.

[16] E. T. Jaynes, *Probability Theory: the Logic of Science*. Cambridge University Press, 2003.

[17] E. Jonas, "Stochastic architectures for probabilistic computation," Ph.D. dissertation, Massachusetts Institute of Technology, 2014.

[18] S. Khasanvis, M. Li, M. Rahman, A. K. Biswas, M. Salehi-Fashami, J. Atulasimha, S. Bandyopadhyay, and C. A. Moritz, "Architecting for causal intelligence at nanoscale," *Computer*, vol. 48, no. 12, pp. 54–64, 2015.

[19] L. B. Kish, "End of moore's law: thermal (noise) death of integration in micro and nano electronics," *Physics Letters A*, vol. 305, no. 34, pp. 144 – 149, 2002.

[20] W. Krauth, *Statistical Mechanics : Algorithms and Computations*. Oxford University Press, 2008.

[21] X. Li, L. Girin, R. Horaud, and S. Gannot, "Estimation of the direct-path relative transfer function for supervised sound-source localization," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 11, pp. 2171–2186, 2016.

[22] M. I. Mandel, R. J. Weiss, and D. P. Ellis, "Model-based expectation-maximization source separation and localization," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 18, no. 2, pp. 382–394, 2010.

[23] T. May, S. van de Par, and A. Kohlrausch, "A probabilistic model for robust localization based on a binaural auditory front-end," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 19, no. 1, pp. 1–13, 2011.

[24] A. Mizrahi, N. Locatelli, R. Matsumoto, A. Fukushima, H. Kubota, S. Yuasa, V. Cros, J. V. Kim, J. Grollier, and D. Querlioz, "Magnetic stochastic oscillators: Noise-induced synchronization to underthreshold excitation and comprehensive compact model," *IEEE Trans. Magn.*, vol. 51, no. 11, pp. 1–4, 2015.

[25] A. V. Oppenheim and R. W. Schafer, "Digital signal processing," 1975.

[26] M. Raspaud, H. Viste, and G. Evangelista, "Binaural source localization by joint estimation of ILD and ITD," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 18, no. 1, pp. 68–77, 2010.

[27] C. Thakur, S. Afshar, R. Wang, T. Hamilton, J. Tapson, and A. V. Schaik, "Bayesian estimation and inference using stochastic electronics." *Frontiers Neuroscience*, vol. 10, no. 104, 2016.

[28] B. Vigoda, "Analog logic: Continuous-time analog circuits for statistical signal processing," Ph.D. dissertation, Massachusetts Institute of Technology, 2003.

[29] J. Woodruff and D. Wang, "Binaural localization of multiple sources in reverberant and noisy environments," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 20, no. 5, pp. 1503–1512, 2012.