Online Multimodal Speaker Detection for Humanoid Robots

Jordi Sanchez-Riera[†], Xavier Alameda-Pineda[†], Johannes Wienke[‡],
Antoine Deleforge[†], Soraya Arias[†], Jan Čech[†], Sebastian Wrede[‡] and Radu Horaud[†]

[†]INRIA Grenoble Rhône-Alpes
655, Avenue de l'Europe
38334 Monbonnot, France
name.lastname@inria.fr

[‡]CoR-Lab, Bielefeld University
Universitätsstraße 25
33615 Bielefeld, Germany
{jwienke,swrede}@cor-lab.uni-bielefeld.de

***Abstract*— In this paper we address the problem of audio-visual speaker detection. We introduce an online system working on the humanoid robot NAO. The scene is perceived with two cameras and two microphones. A multimodal Gaussian mixture model (mGMM) fuses the information extracted from the auditory and visual sensors and detects the most probable audio-visual object, *e.g.,* a person emitting a sound, in the 3D space. The system is implemented on top of a platform-independent middleware and it is able to process the information online (17Hz). A detailed description of the system and its implementation are provided, with special emphasis on the online processing issues and the proposed solutions. Experimental validation, performed with five different scenarios, show that that the proposed method opens the door to robust human-robot interaction scenarios.**

I. Introduction

Humanoid robots acting in populated spaces require a large variety of communication skills. Perceptive, proprioceptive as well as motor abilities are mandatory to make the information flow natural between people and robots participating in interaction tasks. On the perceptive side of the communication process, the tasks are mainly detection, localization and recognition. Depending on the available sensory modalities, the robot should be able to perform tasks such as: sound/speech detection/recognition, action/gesture recognition, identity/voice recognition, face detection/recognition, etc. Moreover, if several modalities are combined, multimodal tasks such as audio-visual event recognition or audio-visual speech processing are known to be more robust than unimodal processing and hence multi-sensory perception can drastically improve the performance of a large variety of human-robot interaction activities.

The problem of data fusion and multi sensory integration has been recognized for a long time as being a key ingredient of an intelligent system, *e.g.,* [12]. More recently, multimodal integration has been used in action recognition applications [11], [17]. In these papers, the authors exploit the fact that for some actions such as "talk phone" the auditory information is relevant for describing the action. Another multimodal approach was followed in [10], in which the auditory information was used to recognize objects that can be partially occluded or difficult to detect. Notice that, the visual information is also very helpful when the auditory data is strongly corrupted by noise or by multiple sound sources. Another example can be found in [7] where the authors combine information coming from the two modalities to perform beat tracking of a person playing the guitar. Auditory and visual information is combined together to better address the problems of beat-tracking, tempo changes, and varying note lengths. Also this different-modality combination is used for improvement of simultaneous speach signals in [13]. Using a pair of cameras faces are first detected and then in 3D. Using two microphones sound-source separation by ADPF (Active Direction Pass Filter) is applied. Finally these data is integrated at two levels, at the signal level and at the word level.

Work supported by the EC project HUMAVIPS FP7-ICT-2009-247525

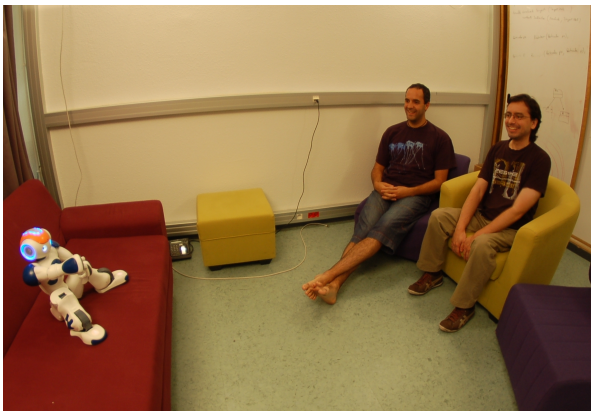

Fig. 1: A typical scenario in which a companion humanoid robot (NAO) performs audio-visual fusion in an attempt to detect the auditory status of each one of the speakers in the room. The system described in this paper processes the raw data gathered with the robot's camera and microphone pairs. The system output is a speaking probability of each one of the actors together with the 3D location of the actors' faces.

Among all possible applications using audio-visual data, we are interested in detecting multiple speakers in informal scenarios. A typical example of such a scenario is shown in Fig. 1, in which two people are sitting and chatting in front of the robot. The robot's primary task (prior to speech

recognition, language understanding, and dialog) consists in retrieving the auditory status of several speakers along time. This allows the robot to concentrate its attention onto one of the speakers, *i.e.,* turn its head towards in the speaker's direction to optimize the emitter-to-receiver pathway, and attempt to extract the relevant auditory and visual data from the incoming signals. We note that this problem cannot be solved within the traditional human-computer interaction paradigm which is based on *tethered* interaction (the user must wear a close-range microphone) and which primarily works in the single-person-to-robot communication case. This considerably limits the range of potential interactions between robots and people engaged in a cooperative task or simply in a multi-party dialog. In this paper we investigate *untethered* interaction thus allowing a robot with its *onboard sensors* to perceive the status of several people at once and to communicate with them in the most natural way.

The original contribution of this paper is a complete real-time audio-visual speaker detection and localization system that is based binocular and binaural robot perception as well as on a generative probabilistic model able to fuse data gathered with camera and microphone pairs. The remainder of the paper is organized as follows. Section II describes previous work in human-robot interaction and audio-visual processing that is closely related to our approach. Next, the audio-visual fusion model is depicted in section III; this model is specifically designed to handle visual and auditory observations gathered at different sampling rates and which live in different observation spaces. Being based on a mixture model, the proposed fusion framework is able to estimate the number of audio-visual objects (speakers) in the scene and to deal with intermittent auditory data. Sections III-A and III-B briefly describe the visual processing and auditory processing modules while the audio-visual data alignment method, *i.e.,* calibration is described in section IV. The overall system architecture is described in section V together with implementation details. Results and experiments are described in section VI. Finally, section VII draws some conclusions and discusses topics for future research.

II. Related Work and Contributions

Audio-visual processing has been studied by many researchers. In [3] the authors describe a speaker detection probabilistic graphical model fusing the information coming from one camera and two microphones. An EM algorithm estimates the model's parameters, i.e., the audio-visual appearance and the position of the speaker. In [5] the author proposes to use maximally informative projections to retrieve the main speaker. One camera per potential speaker and one microphone are used to gather the raw data, which is projected in order to subsequently select the speaker, based on information-theoretic criteria. This approach is well suited for applications such as video-conferencing.

A second group of methods deal with interaction in smart-room environments. These methods assume the existence of several sensors distributed in the scene. For instance, [14] uses data acquired in a room equipped with a multi-camera system and an array of microphones. A tracking system is developed to complement information for a room with a smart interaction environment. The authors of [19] present an application aiming at making meetings more dynamic for people who are remotely connected. Based on one camera and one microphone array, this methodology is able to detect the speaking persons in real-time.

Because of on-line and on-board processing constraints associated with humanoid platforms, the computational load and complexity are constraints that need to be taken into account. Furthermore, the robot does not have a distributed sensor network, but merely a few sensors, which are all located in its head – *an agent-centered sensor architecture*. Hence, one should achieve a trade-off between performance and complexity.

In this paper we present both a novel method and an original system approach to tackle the problem of on-line audio-visual detection of multiple speakers using the companion humanoid robot NAO[1]. The proposed method uses data coming from a stereo pair of cameras and two microphones. Implemented on a hardware- and sensor-independent middleware, the software runs on-line with good performance. The 3D positions of the speakers' heads are obtained from the stereo image pair, and interaural time difference (ITD) values are extracted from the binaural signals. These features are then fused in a probabilistic manner in order to compute, over time, the probability of each person's speaking activity.

The approach exhibits a number of novelties with respect to previous work addressing audio-visual fusion for speaker detection: (i) visual features are obtained from a stereoscopic setup and thus represented in 3D, (ii) auditory features are obtained from only two microphones, while most of previous work uses an array of microphones, (iii) the software is reusable with other robot sensor architectures, due to the flexibility of the underlying middleware layer, and (iv) good on-line performance in a complex environment, *e.g.,* echoic rooms, simultaneous auditory sources, background noise, uncontrolled lighting, cluttered scenes, etc.

III. An Audio-Visual Fusion Model

The overall goal is to retrieve the audio-visual (AV) state of the speakers in front of the robot. That is, the number of speakers as well as their positions and their speaking state. In order to reach this goal, we adopted the framework proposed in [2]. Based on a multimodal Gaussian mixture model (mGMM), this method is able to detect and localize audio-visual events from auditory and visual observations. We chose this framework because it is able to account for several issues: (i) the observation-to-speaker assignment problem, (ii) observation noise and outliers, (iii) the possibility to weight the relevance of the two modalities, (iv) a generative

[1] `http://www.aldebaran-robotics.com`

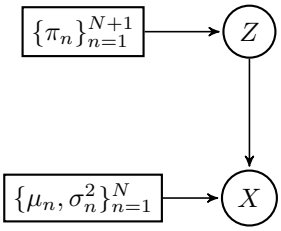

Fig. 2: Graphical model generating the audio-visual observations. The hidden variable Z follows a multinomial distribution with parameters $\pi_1, \ldots, \pi_{N+1}$. The audio-visual observations X follow the law described by the probability density function in Equation 2.

formulation linking the audio and visual observation spaces, and (v) the possibility to deal with a varying number of speakers through a principled model selection method.

In a first stage, the low-level auditory and visual features are extracted. While the former correspond to the interaural time differences (ITDs), the latter correspond to interest points in image regions related to motion which are further reconstructed in the 3D space using a stereo algorithm. These 3D points will be referred to as the visual features.

The following direct sound propagation model:

$$\text{ITD}(\boldsymbol{S}) = \frac{\|\boldsymbol{S} - \boldsymbol{M}_1\| - \|\boldsymbol{S} - \boldsymbol{M}_2\|}{\nu}, \qquad (1)$$

is assumed. In this equation $\boldsymbol{S}$ corresponds to the sound source positions in the 3D space, *e.g.,* a speaker, $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are the 3D coordinates of the microphones in some robot-centered frame, and ν denotes the sound speed. Equation (1) maps 3D points onto the 1D space of ITD observations. The key aspect of our generative audio-visual model [8], [2] is that (1) can be used to map 3D points (visual features) onto the ITD space associated with two microphones, on the premise that the cameras are aligned with the microphones [9]. Hence the fusion between binaural observations and binocular observations is achieved in 1-D.

The underlying multimodal GMM (mGMM) is a one-dimensional mixture of Gaussians. Each mixture component is associated with an *audio-visual object* centered at μ_n and with variance σ_n^2. This mixture has the following probability density function:

$$\text{prob}(x\,;\,\Theta) = \sum_{n=1}^{N} \pi_n \mathcal{N}(x\,;\,\mu_n, \sigma_n^2) + \pi_{N+1}\,\mathcal{U}(x), \qquad (2)$$

where N is the number of components, *i.e.,* audio-visual objects, π_n is the weight of the n^{th} component, $\mathcal{N}(x\,;\,\mu_n, \sigma_n^2)$ is the value of the Gaussian distribution at x, $\mathcal{U}$ is the value of the uniform distribution accounting for outliers, and $\Theta = \{\pi_n, \mu_n, \sigma_n^2\}$. In this equation, x stands for a realization of the random variable X, shown in the corresponding graphical model on Fig. 2, that could be either an auditory observation, *i.e.,* an ITD value or an observed 3D point, *i.e.,* a visual feature, mapped with (1). Notice that both Θ and the hidden variable Z (modeling the observation-to-object assignments) need to be estimated. This is done using an Expectation-Maximization (EM) algorithm, derived from the probabilistic graphical model. Notice that with this formulation the number of AV objects N can be estimated from the observed data by maximizing a Bayesian information criterion (BIC) score [2]. However, this implies to run the EM algorithm several times with different values of N, which is prohibitive in the case of an on-line implementation. From a practical point of view the problem of estimating N can be overcome by replacing the 3D visual points with *3D faces* as described below.

A. Visual Processing

The initial implementations of the nGMM EM algorithm was using 3D points [8], [2] as just described. Alternatively, one can replace 3D points with 3D faces, more precisely with 3D face centers which are fair approximations of 3D mouth positions, *i.e.,* the 3D acoustic emitters. In practice we start by detecting faces in images using [15]. Face centers are then detected in the left image of the stereo camera pair. Each left-image face center is then correlated with the right image along an epipolar line in order to obtain a stereo correspondence between the face center and a associated point along the epipolar line in the right image. This allows to reconstruct a 3D point, $\boldsymbol{S}_n$, that can be viewed as 3D face center. See [6] for more details. The use of faces drastically simplifies the complexity of the approach because a single semantically-meaningful face center replaces a cloud of points associated with a, possibly moving, 3D object. Initial means can be easily obtained from (1), *i.e.,* $\mu_n = \text{ITD}(\boldsymbol{S}_n)$ while N, the number of AV objects can be easily estimated using the face detector [15].

B. Auditory Processing

As already mentioned we use ITDs, *i.e.,* the time delay between the signals received at the left and right microphones. Notice that, due the symmetric nature of the ITD function, there is a front/back ambiguity, which is however slightly attenuated by the transfer function of the robot head. There are several methods to estimate ITDs (see [4] for a review); We chose the cross-correlation method, since it optimizes a trade-off between performance and complexity. ITD values are obtained in real-time by computing the cross-correlation function between the left and right perceived signals during an integration time window of length W, expressed in number of time samples, or frames. The time delay τ corresponding to the maximum of the cross-correlation function in the current integration window is computed as follows:

$$\tau = \frac{1}{F_s} \underset{d \in [-d_M, d_M]}{\text{argmax}} \sum_{t=1}^{W} l(t) r(t+d) \qquad (3)$$

where l and r are the left and right audio signals, F_s is the sampling frequency and d_M denotes the maximum possible delay between microphones, *i.e.,* $d_M = \|\boldsymbol{M}_1 - \boldsymbol{M}_2\|/\nu$. The time window W is a trade-off between reliability and significance. On one hand, a high W value implies more reliable ITD values, since the effect on the local maxima of the cross-correlation function is reduced. On the other hand, a small W value speeds up the computation. The parameter f denotes the shift of the sliding window used to compute the ITD. In order to extract one ITD value, two conditions need to be satisfied. First, there should be enough samples available within the integration window W. Second, the mean energy of the signals in the integration window should be higher than a given threshold E_A. In this way, we avoid to compute ITD values when the audio stream contains nothing but noise. Notice the method does not assume that the perceived sound signals are associated with some semantic *i.e.,* speech, pulse-resonance sounds, etc.

IV. System Calibration

The audio-visual fusion model outlined above, and 1 in particular, implies that the visual and auditory observations are computed in a common reference frame. This allows visual data to be *aligned* with auditory data. In practice it means that the cameras' extrinsic calibration parameters (position and orientation) and the microphones' positions are expressed in a common reference frame. Extrinsic camera calibration is performed using the state-of-the-art algorithm of [18].

Audio-visual calibration can be achieved using (1). A sound-source is placed in a known position $\boldsymbol{S}$ while $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are unknown and hence must be estimated. The method of [9] (i) uses an audio-visual target (a loud-speaker emitting white noise coupled with a small red-light bulb) to precisely position the sound source in the camera-pair reference frame, and (ii) estimates the unknown parameters $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ by considering several target positions and by solving a non-linear system of equations of the form of (1).

This calibration procedure does not take into account the fact that the microphones are plugged into the robot head, as already mentioned above. To account for head effects we introduce two corrective parameters, α and β, to form of an affine transformation.

$$\text{ITD}_{\text{AD}}(\boldsymbol{S}) = \alpha \frac{\|\boldsymbol{S} - \boldsymbol{M}_1\| - \|\boldsymbol{S} - \boldsymbol{M}_2\|}{\nu} + \beta, \qquad (4)$$

These parameters are estimated using the same audio-visual target mentioned above. The audio-visual target is freely moved in front of the robot thus following a zigzag-like trajectory. The use of white noise greatly facilitate the task of cross-correlation, *i.e.,* there is single shapr peak, and hence, makes the ITD computation extremely reliable. The reverberant components are suppressed by the direct component of the long lasting white noise signal. However, it is possible to set up the experimental conditions such as to reduce the effects of reverberation, *e.g.,* the room size is much larger than the target-to-robot distance. If the microphone positions are estimated in advance, the estimation of α and β can be carried out via a linear least-square estimator derived from 4. Fig. 3 shows the extracted ITDs (red-circle), mapped 3D face centers before the adjustment (blue), *i.e.,* using (1) and after the adjustment (green), *i.e.,* using (4). Clearly, the affine correction model allows a better alignment between the visual and auditory data.

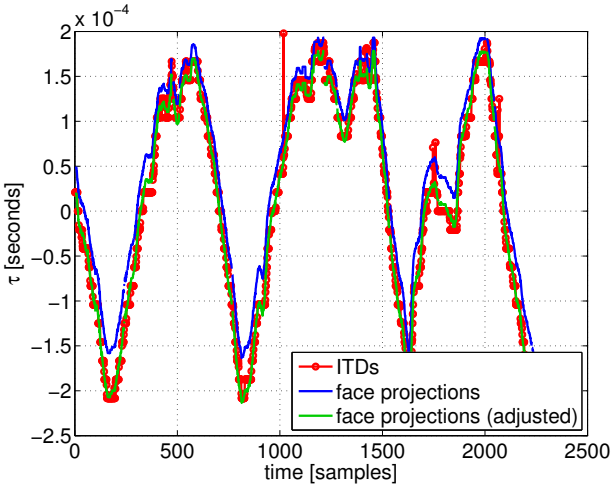

Fig. 3: The effect of the corrective parameters α and β onto audio-visual calibration. ITD values estimated as peaks of the auto-correlation function are shown with red circles. 3D face centers are mapped onto the ITD space without the corrective parameters (shown in blue) and once these parameters have been estimated (shown in green).

V. System Architecture

We implemented the method described above as a number of components connected within the framework of robotics services bus (RSB) [16]. RSB is a platform-independent event-driven middleware specifically designed for the needs of distributed robotic applications. It is based on a logically unified bus which can span over several transport mechanisms like network or in-process communication. The bus is hierarchically structured using scopes on which events can be published with a common root scope. Through the unified bus, full introspection of the event flow between all components is easily possible. Consequently, several tools exist which can record the event flow and replay it later, so that application development can largely be done without a running robot. RSB events are automatically equipped with several timestamps, which provide for introspection and synchronization abilities. Because of these reasons RSB was chosen instead of NAO's native framework NAOqi and we could implement and test our algorithms remotely without performance and deployment restrictions imposed by the robot platform. Moreover, the resulting implementation can be reused for other robots.

One tool available in the RSB ecosystem is an event synchronizer, which synchronizes events based on the attached timestamps with the aim to free application developers from such a generic task. However, several possibilities of how to synchronize events exist and need to be chosen based on the intended application scenario. For this reason, the synchronizer implements several strategies, each of them synchronizing events from several scopes into a resulting compound event containing a set of events from the original scopes. We used two strategies for the implementation. The *ApproximateTime* strategy is based on the algorithm available in [1] and outputs sets of events containing exactly one event from each scope. The algorithm tries to minimize the time between the earliest and the latest event in each set and hence well-suited to synchronize events which originate from the same source (in the world) but suffered from perception or processing delays in a way that they have non-equal timestamps. The second algorithm, *TimeFrame*, declares one scope as the primary event source and for each event received here, all events received on other scopes are attached that lie in a specific time frame around the time stamp of the source event.

ApproximateTime is used in our case to synchronize the results from the left and right camera as frames in general form matching entities but due to independent grabbing of both cameras have slightly different time stamps. Results from the stereo matching process are synchronized with ITD values using the *TimeFrame* strategy because the integration time for generating ITD values is much smaller than for a vision frame and hence multiple ITD values belong to a single vision result.

A. Modular Structure

The algorithm is divided into four components which are described in the pipeline shown in Fig. 5. Each component is represented with a different color and white is reserved for modules provided by the RSB middleware as synchronization functionalities. Red relates to the auditory processing (see section III-B). Green to the visual features (see section III-A). Purple is for the audio-visual fusion method (described in section III) and finally blue is the visualization tool (described at the end of this section).

In detail, the visual part has five different modules. *Left video* and *Right video* stream the images received from left and right cameras. The *Left face detection* module extracts the faces from the left image. These are then synchronized with the right image in *Face-image Synchronization*, using the *ApproximateTime* strategy. The *3D Head Position* module computes the 3D head (or face) centers.

The auditory component consists of three modules. Interleaved audio samples coming from the two microphones are streamed by the *Audio* module. These are de-interleaved by *Sound formatting* and stored into two circular buffers; for the left and right microphone's signals. Finally, the module called *ITD extraction* is in charge of compute the ITD values.

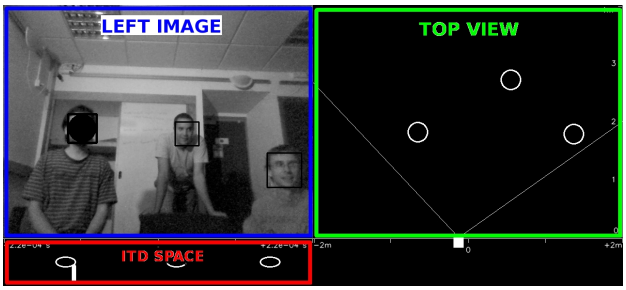

Fig. 4: Snapshot of the visualization tool. Top-left (blue-framed): The original left image overlaid with one bounding box per detected face. In addition, an intensity-coded circle appears when the face emits a sound. The darker the circle, the higher the speaking probability. Top-right (green-framed): A bird-view of the 3D scene, in which each circle corresponds to a detected head. Bottom-left (red-framed): The ITD space. The projected faces are displayed with an ellipse while extracted ITD values are shown as bars in a histogram.

Both visual and auditory features flow until the *Audio-visual synchronization* module; the *TimeFrame* strategy is used here to find the ITD values coming from the audio pipeline associated to the 3D head positions coming from the visual processing. These synchronized events feed the *Audio-visual fusion* module, which is in charge of estimating the emitting sound probabilities p_n.

Finally, we developed the module *Visualization*, in order to get a better insight of the proposed algorithm. A snapshot of this visualization tool can be seen in Fig. 4. The visualization plot consists of three parts. The top-left part displays a bounding box around each detected face overlaid onto the original left image. In addition to the bounding boxes, a solid circle is plot on a face whenever it emits a sound. The intensity encodes the emitting sound probability, the higher it is, the darker the circle. The top-right part, framed in green, is a bird-view of the 3D reconstructed scene, in which the detected 3D faces are shown with circles. The bottom-left part, with a red frame, represents the ITD space in which both the mapped face centers (ellipses) and the histogram of ITD values are plot.

B. Implementation Details

Several considerations need to be done regarding the details of the algorithm implementation in order to guarantee the repeatability of the experiments. When computing the ITD values, a few parameters need to be set. As explained previously (see section III-B), there is a trade off when setting the integration window W and the frame shift f. A good compromise between low computational load, high rate, and reliability of ITD values was found for $W = 150$ ms and $f = 20$ ms. Finally, we set the activity threshold to $E_A = 0.001$. Notice that this parameter could be controlled by a higher level module which would learn the characteristics

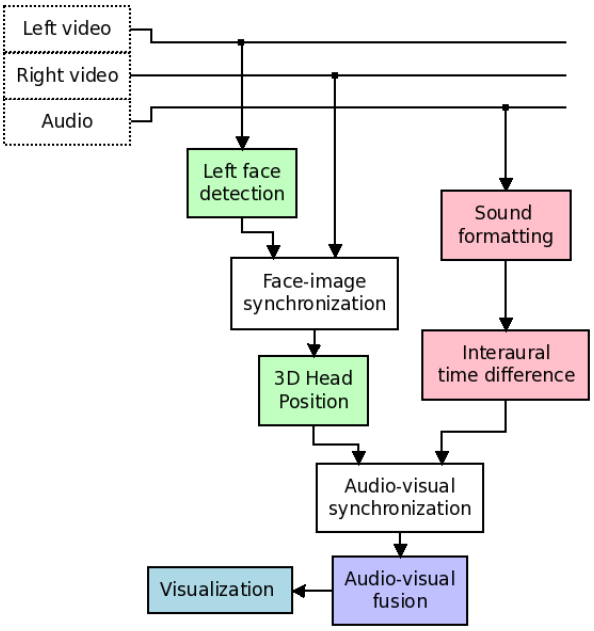

Fig. 5: Modular structure of the processing pipeline. There are five types of modules: streaming & synchronization (white), visual processing (green), auditory processing (red), audio-visual fusion (purple) and the visualization module (blue).

of the scene and infer the level of background noise. When computing the probabilities p_n, the variances of the mGMM are set to $\sigma_n^2 = 10^{-9}$, we found this value big enough to take into account the noise in the ITD values and small enough to discriminate speakers that are close to each other. The entire pipeline was running on a laptop with an i7 processor at 2.5 GHz. Last but not least, we used a new audio-visual head for NAO (see Fig 6). The novelty of this head is the synchronized camera pair delivering a VGA stereoscopic video flow.

VI. Experimental Validation

To validate our algorithm we performed a set of experiments with five different scenarios. The scenarios were recorded in a room around 5×5 meters, and designed to test the algorithm in different conditions in order to identify the limitations of the proposed approach. Each scenario is composed by several sequences in which people count from one up to sixteen. Except for the first scenario, composed by one sequence due to its simplicity, the rest of scenarios were recorded several times. Moreover, a video is recorded to show the different scenarios and have a visual validation of the results.

In scenario **S1**, only one person is in the room sitting in front of the robot and counting. In the rest of the scenarios (**S2**-**S5**) three persons are in the room. People are not always in the field of view (FoV) of the cameras and sometimes they move. In scenario **S2** three persons are sitting and counting alternatively one after the other. The configuration

Fig. 6: Within this work we used a new audiovisual head that is composed of a synchronized camera pair and two microphones. This "orange" head replaces the former "blue" head and is fully interfaced by the RSB middleware described in this paper.

of scenario **S3** is similar to the one of **S2**, but one person is standing instead of sitting. These two scenarios are useful to determine the precision of the ITDs and experimentally see if the difference of height (elevation) affects the quality of the extracted ITDs. The scenario **S4** is different from **S2** and **S3** because one of the actors is outside the FoV. This scenario is used to test if people speaking outside the FoV affect the performance of the algorithm. In the last scenario (**S5**) the three people are in the FoV, but they count and speak independently of the other actors. Furthermore, one of them is moving while speaking. With **S5**, we aim to test the robustness of the method to dynamic scenes.

In Fig. 7 we show several snapshots of our visualization tool. These frames are selected from the different scenarios aiming to show both the successes and the failures of the proposed system. Fig. 7a shows an example of perfect alignment between the ITDs and the mapped face, leading to a high speaking probability. A similar situation is presented in Fig. 7b, in which among the three people, only one speaks. A failure of the ITD extractor is shown in Fig. 7c, where the actor in the left is speaking, but no ITDs are extracted. In Fig. 7d we can see how the face detector does not work correctly: one face is missing because the actor is too far away and the other's face is partially occluded. Fig. 7e shows a snapshot of an AV-fusion failure, in which the extracted ITDs are not significant enough to set a high speaking probability. The Fig. 7f, Fig. 7g and Fig. 7h show the effect of reverberations. While in Fig. 7h we see that the reverberations lead to the wrong conclusion that the actor on the right is speaking, we also see that the statistical framework is able to handle reverberations (Fig. 7f and Fig. 7g), hence demonstrating the robustness of the proposed approach.

(a) S1 **(b) S2**

(c) S4 **(d) S5**

(e) S5 **(f) S2**

(g) S3 **(h) S3**

Fig. 7: Snapshots of the visualization tool. Frames are selected among the five scenarios such as to show both the method's strengths and weaknesses. (a) Good results on **S1**. (b) Good results on **S2**, three people. (c) The ITD extractor does not work correctly, thus missing the speaker. (d) Misses of the face detection module. (e) The audio-visual fusion fails to set a high probability to the current speaker. (f,g) The audio-visual fusion model is able to handle reverberations. (h) The reverberations are too close to the mapped head, leading to a wrong decision.

	CD	FP	FN	Total
S1	14	0	0	14
S2	76	12	3	79
S3	75	19	0	75
S4	60	13	2	62
S5	26	20	0	26

TABLE I: Quantitative evaluation of the proposed approach for the five scenarios. The columns represent, in order: the amount of correct detections (CD), the amount of false positives (FP), the amount of false negatives (FN) and the total number of counts (Total).

The scenarios are manually annotated such that we get the ground truth. In order to systematically evaluate the proposed system we adopted an overlap-based strategy. The ground truth of actor n is split in speaking intervals I_n^k indexed by k and silent intervals J_n^l indexed by l. For clarity purposes let us denote by $p_n(t)$ the detected speaking state of actor n at time t. For each of the speaking intervals I_n^k we compute $c_n^k = \sum_{t \in I_n^k} p_n(t)/|I_n^k|$. If $c_n^k \geq 0.5$ we count one correct detection, otherwise we count one false negative. We also compute $\tilde{c}_n^k = \sum_{t \in J_n^l} p_n(t)/|J_n^l|$. In case $\tilde{c}_n^l \geq 0.5$ we count one false positive. In summary, if the speaker is detected during more than half of the speaking time, we count on correct detection (CD), otherwise a false negative (FN). And if it is detected more than half of the speaking time, we count a false positive (FP).

Table I shows the results obtained with this evaluation strategy on the presented scenarios. First of all we notice the small amount of false negatives: the system misses very few speakers. A part from the first scenario (easy conditions), we observe some false positives. These false positives are due to reverberations. Indeed, we notice how the percentage of FP is severe in **S5**. This is due to the fact that high reverberant sounds (like hand claps) are also present in the audio stream of this scenario. We believe that an ITD extraction method more robust to reverberations will lead to more reliable ITD values, which in turn will lead to a better speaker detector. It is also worth to notice that actors in different elevations and non-visible actors do not affect the performance of the proposed system, since the results obtained in scenarios **S2** to **S3** are comparable.

VII. Conclusions and Future Work

We presented a system targeting speaker detection working on the humanoid robot NAO in regular indoor environments. Implemented on top of a platform-independent middleware, the system processes the audio-visual data flow from two microphones and two cameras at a rate of 17 Hz. We proposed a statistical model which captures outliers from the perception processes. The method runs in normal echoic rooms with just two microphones mounted inside the head of a companion robot with noisy fans. We demonstrated good performance on different indoor scenarios involving several actors, moving actors and non-visible actors. This works contributes to a better understanding of the audio-visual scene using a robocentric set of sensors mounted in an autonomous platform, such as NAO, under the constraints of an on-line application.

It is worth noticing that the module limiting the performance of the system is the ITD extraction, due to the room reverberations. We will work on making this module more robust to this kind of interferences. Moreover, audio-visual tracking capabilities are also a desirable property for any robot, since they provide for temporal coherence of the scene. In a more developed stage, it would be desirable that NAO is able to choose regions of interest, so that it could perform active learning, and enhance its audio-visual skills.

References

[1] "message_filters/approximatetime," http://www.ros.org/wiki/message_filters/ApproximateTime, accessed: 06/21/02012.

[2] X. Alameda-Pineda, V. Khalidov, R. Horaud, and F. Forbes, "Finding audio-visual events in informal social gatherings," in *ICMI*, 2011.

[3] M. J. Beal, H. Attias, and N. Jojic, "Audio-visual sensor fusion with probabilistic graphical models," in *ECCV*, 2002.

[4] Y. Chan, W. Tsui, H. So, and P. Ching, "Time-of-arrival based localization under NLOS conditions," *Vehicular Technology, IEEE Transactions on*, vol. 55, no. 1, pp. 17–24, 2006. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1583910

[5] J. W. Fisher and T. Darrel, "Speaker association with signal-level audiovisual fusion," in *IEEE Transactions on Multimedia*, vol. 6, no. 3, 2004.

[6] M. Hansard and R. Horaud, "Cyclopean geometry of binocular vision," *Journal of the Optical Society of America A*, vol. 25, no. 9, p. 23572369, September 2008.

[7] T. Itohara, T. Otsuka, T. Mizumoto, T. Ogata, and H. G. Okuno, "Particle-filter based audio-visual beat-tracking for music robot ensemble with human guitarist," in *IROS*, 2011.

[8] V. Khalidov, F. Forbes, and R. Horaud, "Conjugate mixture models for clustering multimodal data," in *Neural Computation*, vol. 23, no. 2, 2011, pp. 517–557.

[9] V. Khalidov, F. Forbes, and R. P. Horaud, "Calibration of a binocular-binaural sensor using a moving audio-visual target," INRIA Grenoble Rhone-Alpes, Tech. Rep. 7865, January 2012.

[10] L. Lacheze, Y. Guo, R. Benosman, B. Gas, and C. Couverture, "Audio/video fusion for objects recognition," in *IROS*, 2009.

[11] N. A. Lili, "A framework for human action detection via extraction of multimodal features," in *International Journal of Image Processing*, vol. 3, no. 2, 2009.

[12] R. C. Luo and M. G. Kay, "Multisensor integration and fusion in intelligent systems," in *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, no. 5, 1989, pp. 901–931.

[13] K. Nakadai, D. Matsuura, H. G. Okuno, and H. Tsujino, "Improvement of recognition of simultaneous speech signals using av integration and scattering theory for humanoid robots," *Speech Communication*, pp. 97–112, 2004.

[14] S. T. Shivappa, B. D. Rao, and M. M. Trivedi, "Auvio-visual fusion and tracking with multilevel iterative decoding: Framework and experimental evaluation," in *Journal of Selected Topics in Signal Processing*, 2010.

[15] J. Šochman and J. Matas, "Waldboost – learning for time constrained sequential detection," in *CVPR*, 2005.

[16] J. Wienke and S. Wrede, "A middleware for collaborative research in experimental robotics," in *2011 IEEE/SICE Int. Symposium on System Integration, SII2011*, IEEE. Kyoto, Japan: IEEE, 2011.

[17] Q. Wu, Z. Wang, F. Deng, and D. Feng, "Realistic human action recognition with audio context," in *DICTA*, 2010.

[18] J. Yves Bouguet, "Camera calibration toolbox for matlab," http://www.vision.caltech.edu/bouguetj/calib_doc/.

[19] C. Zhang, P. Yin, Y. Rui, R. Cutler, P. Viola, X. Sun, N. Pinto, and Z. Zhang, "Boosting-based multimodal speaker detection for distributed meeting videos," in *IEEE Transactions on Multimedia*, vol. 10, no. 8, 2008.