AcousticFusion: Fusing Sound Source Localization to Visual SLAM in Dynamic Environments

Tianwei Zhang[1], Huayan Zhang[1], Xiaofei Li[*,2], Junfeng Chen[1], Tin Lun Lam[*,1,3], Sethu Vijayakumar[4,5]

***Abstract*— Dynamic objects in the environment, such as people and other agents, lead to challenges for existing simultaneous localization and mapping (SLAM) approaches. To deal with dynamic environments, computer vision researchers usually apply some learning-based object detectors to remove these dynamic objects. However, these object detectors are computationally too expensive for mobile robot on-board processing. In practical applications, these objects output noisy sounds that can be effectively detected by on-board sound source localization. The directional information of the sound source object can be efficiently obtained by direction of sound arrival (DoA) estimation, but the depth estimation is difficult. Therefore, in this paper, we propose a novel audio-visual fusion approach that fuses sound source direction into the RGB-D image and thus removes the effect of dynamic obstacles on the multi-robot SLAM system. Experimental results of multi-robot SLAM in different dynamic environments show that the proposed method uses very small computational resources to obtain very stable self-localization results.**

I. Introduction

Visual-SLAM is a core technique for a robot to understand the external environment and perform self-orientation. However, in real workspaces, there are many dynamic objects such as moving human talkers and other robots (in multi-robot systems). These dynamic objects disrupt most existing vision-SLAM systems: In the case of localization, visual odometry fails because the moving camera cannot acquire enough static visual features from the background that is obscured by dynamic objects. For environmental mapping, these moving obstacles with distorted shapes are not supposed to appear in the final map. To deal with this dynamic environment problem, an intuitive idea is to introduce a detector for moving objects, find and remove these objects by pre-processing them in the SLAM front end, and then enable the static SLAM algorithm. Many dynamic SLAM methods [1], [2], [3], on the one hand, introduce learning-based object detectors to segment bounding boxes or templates of specific movable objects, such as people and vehicles. Other methods, which fall under the motion segmentation category [4], [5], [6], decouple dynamic pixels from static background pixels by comparing camera motion consistency — clustering dynamic pixel points and removing them. Whether the approach is for motion segmentation or object recognition, both frameworks expend huge computational resources for discovering and removing the dynamic components of the visual perceptual input. These approaches typically rely on a dedicated GPU to cope with dynamic obstacles for real-time performance, which limits their application in online, mobile robotic systems with limited computational resources.

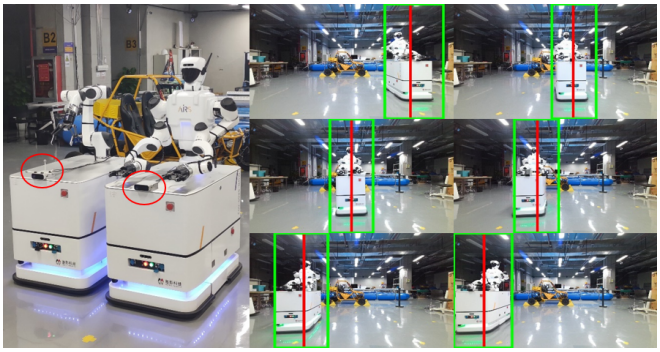


Fig. 1: AcousticFusion SSL testing on the AIRS Mobile Manipulation system robots. An Azure Kinect was installed on the mobile base to keep the camera stable while the robots were moving. We show six images taken by this sensor with the SSL confidence interval marked in green.

Audio Human-robot interaction (HRI) requires the detection of different sound sources in the environment. This function usually requires the use of a microphone array to localize, track, and decompose the different sound sources in real-time. In the field of SSL research, to localize and track the speakers in real-world environments, the classical methods are mostly based on estimating the time differences of arrival (TDOA) between microphones. Knapp *et al.* proposed a classic TDOA estimation method in [7] with generalized cross-correlation. Chen *et al.* proposed a TDOA estimation framework for single-speaker localization in [8]. In the case of multiple speakers, DiBiase *et al.* provided a beamforming-based method named steered-response power in [9], and Ishi *et al.* proposed a famous multiple signals classification approach in [10]. Recently, In [11], Li *et al.* proposed a direct-path relative transfer function method combined with exponential gradient for the simultaneous localization and tracking of multiple moving speakers. This method is robust against the reverberation effect, which is especially important for indoor SLAM applications. Note that, limited by the compact structure of the microphone array, Sound Source Localization (SSL) mentioned here usually only estimates the direction of sound sources, and the range/depth estimation is

[1]The Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen, China

[2]Westlake University & Westlake Institute for Advanced Study, Hangzhou, China.

[3]The Chinese University of Hong Kong, Shenzhen.

[4]School of Informatics, The University of Edinburgh and The Alan Turing Institute, United Kingdom.

[5]The author is a visiting researcher with the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS).

*Corresponding authors: lixiaofei@westlake.edu.cn; tllam@cuhk.edu.cn

not conducted.

In this paper, we use the SSL method to detect sounding obstacles and mark areas of these obstacles in the image and remove them to enable visual-SLAM in the dynamic environment. This work uses the Microsoft Azure Kinect sensor. It is compact in design ($10\ cm \times 12.5\ cm$) and includes a microphones array and an RGB-D camera. The Azure Kinect captures asynchronous sound signals and images, and our method processes and fuses the sound signals into the images. Hence we named the proposed method AcousticFusion. There are several advantages of fusing sound and visual signals of Azure Kinect for the mobile robot: 1) lower on-board power and computation costs. The SSL method *e.g.*, [11] performed efficient online multiple sound source azimuth detection using an on-board CPU. 2) As shown in Fig. 1, the microphone array is small in size and low in power consumption but brings 360-degree azimuth detection and tracking capability (RGB-D cameras have a 90-degree azimuth field-of-view). 3) The Azure Kinect SDK can provide human speaker recognition and voice-to-text functions promising in HRI applications. In summary, this work contributes to:

1) A novel **dynamic SLAM approach** based on sound source detection and sparse feature visual odometry.
2) An efficient and robust method for **fusing SSL and RGB-D image**.
3) A visual odometry database with **synchronized sound and RGB-D images**.

The database and code will be open-sourced depending on acceptance.

II. Related Works

A. Dynamic visual SLAM

Most of the existing dynamic SLAM solutions try to deal with the dynamic environment problem by finding and removing dynamic objects. Based on their object recognition approaches, we divide the current state-of-the-art into motion segmentation-based and object detection-based methods.

Object detection-based dynamic SLAM methods usually utilize advanced deep learning-based object detectors to remove dynamic objects and then enable the classical static SLAM frameworks in the dynamic environments. Bescos *et al.* [1] proposed DynaSLAM which applied Mask-RCNN [12] to detect human objects in RGB images and adopted ORB-SLAM2 (ORB2) [13] for camera tracking. DynaSLAM performed accurate human silhouette segmentation, but it spent abound $300ms$ per frame.

Motion segmentation based approaches attempted to find dynamic pixels or point clouds rather than recognizing moving objects. Scona *et al.* proposed StaticFusion (SF) [4] that combined scene flow computation with Visual Odometry to achieve real-time static background reconstruction in a small-sized room. Zhang *et al.* proposed FlowFusion [5] that utilized optical flow residuals for dynamic object segmentation and remove the dynamic point clouds for dense background reconstruction. Judd *et al.* provided a multi-object motion segmentation method in [14], which applied sparse feature points alignment to separate and track multiple rigid objects. Dai *et al.* proposed to distinguish dynamic or static feature points using motion consistence in [15]. All of the above dynamic SLAM solutions can not work on real-time mobile robot platforms without GPUs.

B. Audio-Visual Fusion Methods

Hospedales *et al.* [16] proposed a Bayesian model-based audio-visual fusion framework to segment, associate, and track multiple objects in audiovisual sequences. Li *et al.* presented an SSL-based HRI system in [17]. They calibrated the sound sources' corresponding pixel coordinates. Hence an NAO robot head with four microphones performed robust azimuth localization under difficult acoustic conditions. Ban *et al.* proposed an audio-visual fusion method for multi-speaker tracking in [18], which fused direct-path related transfer function features into the Bayesian face observation model. Then, they updated this multi-speaker tracking module in [19]. It can track multiple speakers and locate the sound source into the bounding box of the speaker's head. In that work, CNN-based offline person detection is required, which is two frames per second (*fps*) on a GTX 1070 GPU.

The audio-visual fusion works [17], [18], [19], all use static robots to track the observer. For the moving robot, in [20], Evers *et al.* proposed an acoustic SLAM framework that is different from the general concept of SLAM. Acoustic SLAM applied the SSL technique passively localize a moving observer and simultaneously mapped the positions of surrounding sound sources. It does not work on robot self-localization and environment mapping. A recent work [21] proposed to use two moving microphone arrays to do SSL separately and to estimate the sound source location using the intersection of sound source direction extension lines.

III. Approach

A. Overview

Our proposed AcousticFusion framework combines SSL technology into a mobile robot vision-SLAM system. The flowchart is shown in Fig. 2: The SSL module takes the sound signals collected by seven microphones as input, and after feature extraction and clustering, outputs time-varying sound source azimuth angles, which are then fused into the image space. For the RGB-D images acquired by the same device, we first invalidate the depth values within the SSL region and then extract the visual features for the following ego-motion estimation and mapping.

B. Online Multiple Sound Source Localization

1) Extraction of Localization Feature

In the time domain, we represent the microphone signal as: $y^m(t) = h^m(t) * x(t)$, where $m = 1,\ldots,M$ and t denote the microphone and time indices, respectively. The m-th microphone signal $y^m(t)$ is the convolution (denoted as $*$) of the source signal $x(t)$ and room impulse response (RIR) $h^m(t)$. In the short-time Fourier transform (STFT) domain, this convolution is represented with convolutive transfer function (CTF): $y^m_{p,k} = h^m_{p,k} * x_{p,k}$, where $p = 1,\ldots,P$ and

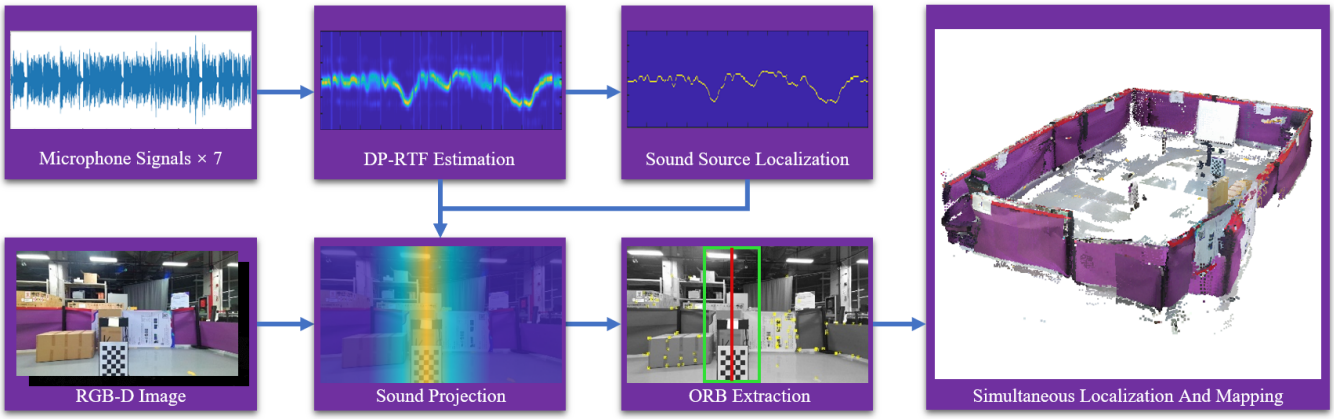


Fig. 2: AcousticFusion Flowchart. Azure Kinect captures sound signals and RGB-D images. These sound signals from seven microphones are processed by feature extraction and clustering to output time-varying azimuths of the sound sources. Then, these sound source azimuth angles are fused into the image space. For the RGB-D images, we first invalidate the depth values within the sound source localization region and then extract the visual features (the yellow dots) for camera motion estimation and mapping.

$k = 0, \ldots, K-1$ denote the time-frame and frequency indices, respectively. Analogous to the time-domain representation, the STFT coefficients of microphone signal $y^m_{p,k}$ is the convolution (along the time-frame axis) of the STFT coefficients of source signal $x_{p,k}$ and CTF $h^m_{p,k}$ (the STFT representation of RIR). The CTF coefficients encode the RIR taps for one subband, and preserve the reverberation structure of RIR. The direct-path propagation presents at the first RIR taps, and thence at the first CTF coefficient. Localization of sound source relies on estimating the direct-path propagation of the source signal to multiple microphones. In the following, we first estimate the entire CTFs based on the cross-relation method [22] from the microphone signals, and then extract the first CTF coefficients for sound source localization. Since the CTF estimation is independently conducted for each frequency, for notational simplicity, the frequency index k will be omitted until the next section.

For one microphone pair (m,n), we have the cross-relation $y^m_p * h^n_p = y^n_p * h^m_p$. Let Q denote the number of CTF coefficients, $\mathbf{h}^m = [h^m_0, \ldots, h^m_{Q-1}]^T$ and $\mathbf{y}^m_p = [y^m_p, \ldots, y^m_{t-Q+1}]^T$ denote the vector form of CTF and microphone signal, where ^{T} denotes matrix/vector transpose. The cross-relation can be written in vector form as

$$\mathbf{y}^{m\ T}_p \mathbf{h}^n = \mathbf{y}^{n\ T}_p \mathbf{h}^m. \tag{1}$$

The CTF vector to be estimated of all channels are concatenated as $\mathbf{h} = [\mathbf{h}^{1\ T}, \ldots, \mathbf{h}^{M\ T}]^T$. To represent the pairwise cross-relation with respect to $\mathbf{h}$, the microphone signal vectors are concatenated as:

$$\mathbf{y}^{mn}_p = [0, \ldots, 0, \mathbf{y}^{n\ T}_p, 0, \ldots, 0, -\mathbf{y}^{i\ T}_p, 0, \ldots, 0]^T, \tag{2}$$

where the zero-elements are set to respond to the CTF vectors other than the m-th and n-th microphones, so that the cross-relation (1) can be written as $\mathbf{y}^{mn\ T}_p \mathbf{h} = 0$. There exist one trivial solution for this equation, namely $\mathbf{h}$ equals 0. To avoid this solution, the first CTF coefficient of the reference channel, say $m = r$, is constrained to be equal to 1, namely

$$\mathbf{y}^{mn\ T}_p \mathbf{h} = 0, \quad s.t. \quad h^r_0 = 1 \tag{3}$$

This can be realized by dividing $\mathbf{h}$ by h^r_0, which yields a new equation $\mathbf{y}^{mn\ T}_p \mathbf{h}/h^r_0 = 0$. Moving the constant term from the left side to the right side, we have

$$\tilde{\mathbf{y}}^{mn\ T}_p \tilde{\mathbf{h}} = z^{mn}_p, \tag{4}$$

where $\tilde{\mathbf{y}}^{mn}_p$ is $\mathbf{y}^{mn}_p$ with the entry corresponding to h^r_0 removed, and $-z^{mn}_p$ is such entry. The new variable $\tilde{\mathbf{h}}$ is $\mathbf{h}$ with h^r_0 removed, and then divided by h^r_0. In $\tilde{\mathbf{h}}$, the elements corresponding to the first CTF coefficient of multiple microphones (other than the r-th microphone), i.e. $h^m_0/h^r_0, m \neq r$, represent the ratio between the direct-path transfer function of two microphones, and are referred to as direct-path relative transfer functions (DP-RTFs). DP-RTFs encode the localization cues, namely the inter-channel phase/magnitude difference of the direct-path signal propagation.

Sound source localization amounts to estimate the DP-RTFs by solving the linear problem Eq. (4). We note that Eq. (4) is defined for one microphone pair at one time frame. For online processing, we receive the microphone signals $\tilde{\mathbf{y}}^{mn}_p$ and z^{mn}_p frame by frame, and accordingly the estimate of $\tilde{\mathbf{h}}$ will also be updated frame by frame. For one frame, all the $I = M(M-1)/2$ distinct microphone pairs are utilized. For notational convenience, we use $i = 1, \ldots, I$ denote the index of microphone pair to replace mn. Define the fitting error of (4) as $e^i_p = \tilde{\mathbf{y}}^{i\ T}_p \tilde{\mathbf{h}} - z^i_p$. At one current frame p, exploiting the microphone pairs up to i, online processing aims to minimize

$$J^i_p = \sum_{p'=1}^{p-1} \lambda^{p-p'} \sum_{i'=1}^{I} |e^{i'}_{p'}|^2 + \sum_{i'=1}^{i} |e^{i'}_p|^2, \tag{5}$$

which sums up the fitting error of all the currently available frames and microphone pairs. Along with the increase of p or i, this error is recursively updated with one new error term, i.e. $|e^i_p|^2$, for which $\tilde{\mathbf{h}}$ can be efficiently estimated with the recursive least squares algorithm (please find more details from [11]). At each frame p, one estimate of $\tilde{\mathbf{h}}$ is obtained, denoted as $\hat{\mathbf{h}}_p$. For the dynamic case (either speaker or microphone array is moving), $\tilde{\mathbf{h}}$ is time-varying, and the estimate $\hat{\mathbf{h}}_p$ reflects the current value of $\tilde{\mathbf{h}}$. To catch up the variation of $\tilde{\mathbf{h}}$, the older frames are exponentially forgotten

by the the forgetting factor $\lambda \in (0,1]$. This factor can be set to 1 for the static case in which $\tilde{\mathbf{h}}$ is constant.

Up to now, we consider the noise-free single-speaker case. To suppress noise, the inter-frame spectral subtraction algorithm proposed in [23] can be easily integrated into the current framework. As for the multiple-speaker case, the W-disjoint orthogonality assumption [24] is used, which assumes that the speech signal is dominated by only one speaker in each small region of the STFT domain, because of the natural sparsity of speech signals in this domain. Based on this assumption, the CTF estimates (with frequency index k added), i.e. $\hat{\mathbf{h}}_{p,k}$, belongs to only one of the multiple speakers. Finally, at frame p, from $\hat{\mathbf{h}}_{p,k}$, we extract the DP-RTFs as localization features, denoted as $a^m_{p,k}$, $m \in [1,M], m \neq r; k \in [0,K-1]$, and each feature is associated with a single speaker. Note that those time-frequency bins dominated by noise or multi-speaker are not used for the following localization step.

2) Feature Clustering for Localization

The complex Gaussian mixture model is used to cluster the features to each active speaker. Each component of the mixture model is set to represent one candidate source location. Let $d = 1,\ldots,D$ and w^d ($w^d \geq 0$ and $\sum_{d=1}^{D} w^d = 1$) denote the d-th candidate location and the prior probability of the d-th mixture component, respectively. The probability, that one feature $a^m_{p,k}$ is emitted by candidate locations, is the mixture of complex Gaussian probabilities:

$$P(a^m_{p,k}) = \sum_{d=1}^{D} w^d \mathcal{N}_c(a^m_{p,k}; \bar{a}^{m,d}_k, \sigma^2), \tag{6}$$

where the mean $\bar{a}^{m,d}_k$ is the constant theoretical DP-RTF, which can be precomputed using the theoretical model of the signal's direct-path propagation. The variance σ^2 is empirically set as a constant value. The prior probability (weight) w^d is the only free model parameter, and can be estimated by maximizing the log-likelihood of all the available features, namely

$$\max_{w^d,\ d=1,\ldots,D} \sum_{a^m_{p,k}} \log(P(a^m_{p,k})). \tag{7}$$

This likelihood maximization problem can be easily solved by the well-known expectation-maximization algorithm. For the dynamic case, w^d is time-varying (thence denoted as w^d_p), and can be estimated with recursive expectation-maximization algorithm. The optimized weight w^d_p represents the probability that an active speaker is present at the d-th candidate location. Sound source localization can be conducted by detecting the peak of w^d_p along the d axis.

In this work, we use the microphone array embedded on an Azure Kinect to conduct SSL. The topology of the microphone array is shown in 3, which is composed of seven microphones arranged in a 2D plane. The microphone array is placed to be parallel to the horizontal plane, which is thus suitable to perform horizontal (azimuth) localization. A total of $D = 72$ candidate azimuth angles are set with 5 degrees gap between two adjacent angles to cover the whole 360 degrees azimuth space.

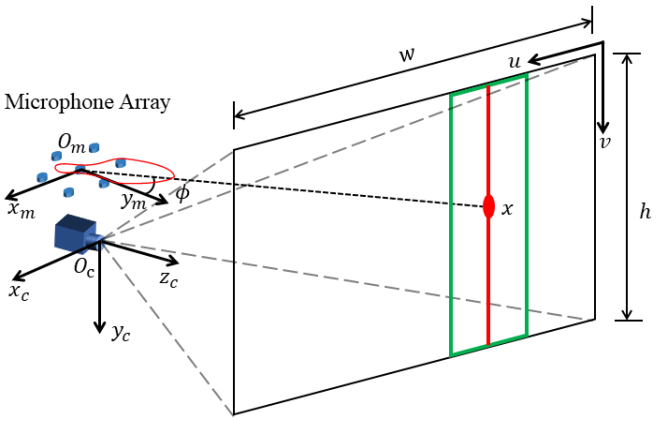


Fig. 3: Projecting the source direction onto the image plane: We first obtain the source azimuth ϕ from the SSL peak, then warp the camera image to the microphone frame and extend the azimuth ray to intersect this image plane. The location of the sound source is located on the vertical line through this intersection point x.

C. Audio-Visual Data Association

In [17], Li *et al.* provided an audio-visual dataset that contains sound source directions that correspond to image pixels. They obtained these correspondences by manually labeling the loudspeaker's positions in the image. This audio-visual information association method is not robust to reverberation condition changing. The sound source to pixel correspondence changes When the robot moves. For the mobile robot SLAM, we should update these audio-visual information correspondence time by time.

Following the RGB-D SLAM method[4], given two image frames: camera image frames C and microphone image frame M, at the sound sampling frame p a pixel x^p_C in frame C can be warped to frame M by:

$$\mathbf{x}^p_M = W(\mathbf{x}^p_C, T(\xi), D_C) \tag{8}$$

where the image warping function W is given by:

$$W(\mathbf{x}^p, T, D) = \pi(T\pi^{-1}(\mathbf{x}^p, D(\mathbf{x}^p))) \ . \tag{9}$$

$\mathbf{x}$ represents a pixel in the 2D image, $D(\mathbf{x})$ is the depth of pixel x. The projection function $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ projects 3D points onto the image plane using the camera intrinsic matrix. The extrinsic matrix $T(\xi) \in SE(3)$ between the camera frame and microphone frame is computed using device hardware parameters.

As the SSL module output *fps* is much higher than the RGB-D camera frame rate, to update the audio-visual correspondence, for each camera image, we warp its pixels $x_p \in C$ to the microphone frame M using Eq. 8, label the newest estimated sound source azimuth on M, and warp the labeled pixel back to the camera frame. For instance, in Fig. 3, assume there is only one sound source d_p, fetch its sound source azimuth $\phi \in (-180, 180]$ (the direction of the protrusion of the red circle) from $\mathscr{D}_p$, extend the azimuth ray to intersect the warped image plane and take the intersection point $x^p_M = (\mu', \nu')$. Then the sound source location should

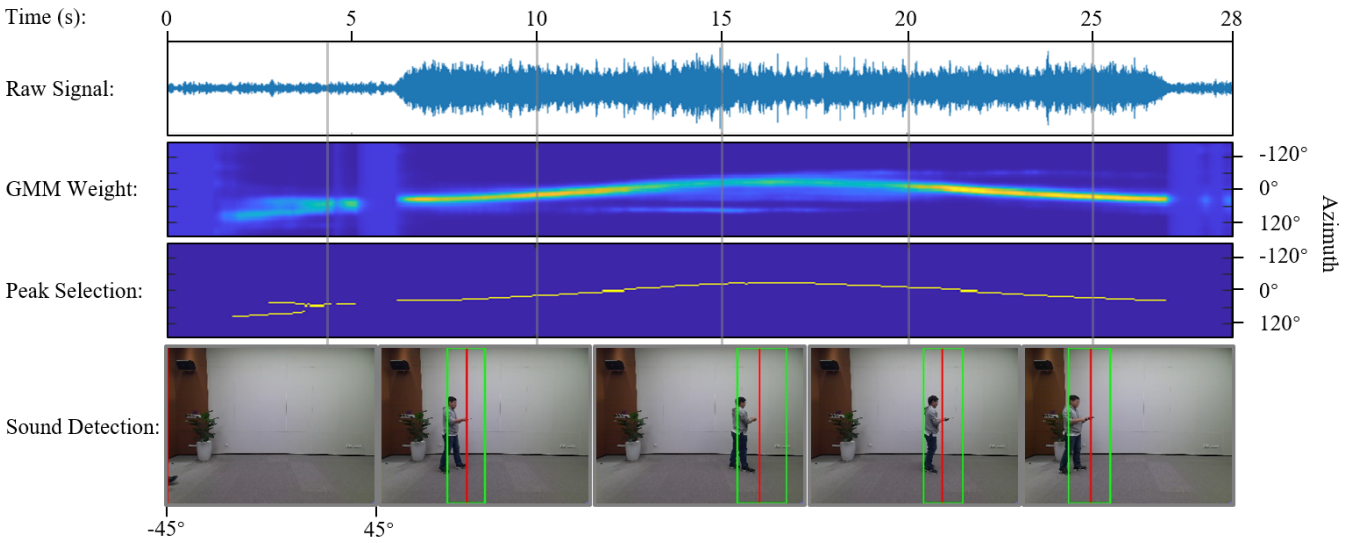
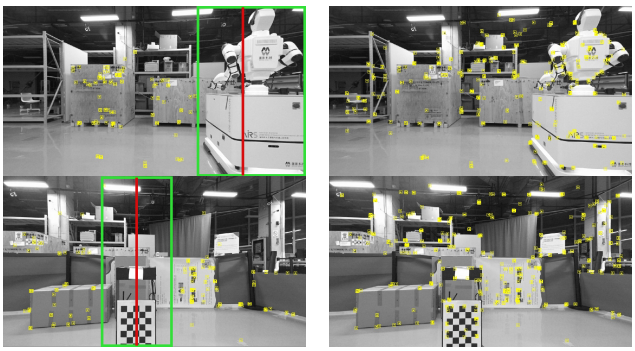


Fig. 4: An example of audio-visual integration. In this scene, a pedestrian is playing a song with his cell phone. The first row is the original sound signal of a microphone channel. After sound feature extraction and clustering, the second and third rows are the results of SSL output, i.e., Gaussian probability weights and directional peaks. The song starts to play from 6.4 and ends at 26.7 seconds, and the two peak curves before 5 seconds are footsteps. The red line is the projection of the sound source peak on the image, and the width of the dynamic object bounding box in green is determined by the GMM weights.

belong to the vertical line through x_M^p, so as x_C^p. However, according to the warping function Eq. 9, the unknown D_w is necessary to warp x_M^p back to x_C^p. In real cases, the distance of the sound source is always further than $1m$, much larger than the distance from the center of the microphone array to the optical center of the camera ($7.4cm$). Thus in this work, we use D_C instead D_M.

Note that the image FOV of the camera is only 90 degrees, which is smaller than the 360 degrees of the SSL module. Therefore, when the SSL result exceeds the camera FOV, the fused sound source red line may be on the left or right border of the image, as shown in the bottom left figure of Fig. 4. This red line labels the detected sound source objects, which will be removed as exceptions in the following SLAM visual feature extracting phase.

D. Visual Feature Extraction and Visual Odometry



(a) Ours (b) Openvslam

Fig. 5: Visual feature extraction in multi-robot scene. We set the width of the feature-free area according to the SSL result. In (a), our method only extracts features from the static backgrounds. In (b), Openvslam extracts wrong environmental features from the other moving robot surfaces.

The SSL module outputs the weights (probabilities) of candidate azimuth angles associated with active speakers, and the localized sound source directions obtained by detecting the peaks of weights. With the sound source direction as the center, we grab an image region to cover the dynamic obstacle as much as possible according to the weights of candidate azimuth angles.

Sound source directions are estimated by detecting the peak of GMM weights w_p^d along the d axis. The candidate locations corresponding to the peaks of w_p^d are denoted as $d_{p,j} \in [1, D],\ j = 1, \ldots, J$, where J denotes the number of detected sound sources. To cover the whole visual obstacles, we need to estimate the obstacle regions in the image. The center of obstacle regions are set to be the sound source directions $d_{p,j} \in [1, D],\ j = 1, \ldots, J$. The region boundaries are separately determined for each visual obstacle. To determine each of the left and right region boundaries for sound direction $d_{p,j}$, for example the right boundary $b_{p,j}^{\text{right}}$, the GMM wights w_p^d are checked one by one from $d_{p,j}+1$ to its right candidates until the following condition is satisfied:

$$w_p^{d+1} \geq w_p^d \quad \text{or} \quad w_p^{d+1} < \delta w_p^{d_{p,j}}, \tag{10}$$

then $b_{p,j}^{\text{right}}$ is set to d. This condition means i) the weight cannot increase, as the increasing weight indicates the emerging of a new sound source; ii) the weight should not be smaller than $\delta w_p^{d_{p,j}}$, where $0 \leq \delta \leq 1$ is empirically set to reflect our prior knowledge about the size of the visual obstacle. Finally, at frame p and for sound source j, the obstacle region is represented by the region center $d_{p,j}$ (red line as shown in Fig. 1, 2, 3, 4 and 5), and the region boundaries $\{b_{p,j}^{\text{left}}, b_{p,j}^{\text{right}}\}$ (green bounding box as shown in Fig. 1, 2, 3, 4 and 5). These time-varying image regions track the multiple moving visual obstacles and will be adopted for the following SLAM step.

For example, in the flowchart Fig. 2, after the azimuth

TABLE I: Dynamic Environment SLAM ATE RMSE (m)

Sequence	ORB2	SF	ORB2+SSL	Ours
Spark-T Robot				
Sp1	0.24	3.4	0.079	0.1
Sp2	0.23	12.8	0.088	0.078
Sp3	0.35	26.47	0.22	0.14
Sp4	0.2	3.69	0.12	0.13
Sp5	0.25	2.35	0.2	0.19
AIRS Mobile Manipulation Robot				
Mo1	1.32	14.2	0.13	0.12
Mo2	1.59	18.28	0.089	0.088
Mo3	0.94	0.71	0.038	0.04
Mo4	1.45	1.53	0.18	0.18

projection of the sound source, we compute this region (green bounding box) in the RGB-D frame and invalidate the depth values in this region in the Depth image. Then, we extract ORB visual features on this pair of RGB and Depth images so that the obtained features avoid moving object regions to ensure visual odometry robustness in these dynamic scenes. Thereafter, the loop detection and mapping tasks are done using Openvslam [25].

IV. Experiments and Evaluations

In this paper, the proposed method was tested on a laptop with Intel Core^{TM} i7-10875H CPU @ 2.30 GHz $\times$ 8, 64 GB System memory. In StaticFuison [4]'s comparison experiments, a GeForce RTX 2080 Ti GPU was used.

A. Sound Source Localization Results

As already partially mentioned above, $D = 72$ azimuth directions at every 5 degrees in $(-180, 180]$ degrees are used as candidate directions to perform 360 degrees azimuth localization. The sampling rate of sound signals is 16,000 Hz. The STFT has a window length of 256 samples and a hop size of 128 samples, correspondingly SSL has an output rate of 125 Hz. The CTF length Q is set to 8. Fig. 1 and 4 indicate the SSL module performance when the robot is static. The localization effect is stable when the target is within 3 *m*. Beyond 3 *m* the localization error increases with distance.

In addition to the sound source direction, the estimation results of the sound source area have a significant impact on the subsequent visual odometry. In multi-robot SLAM experiments, the sound source area width should be adjusted appropriately according to the size of the robot because large-sized moving targets obscure more pixels at the same distance. *e.g.*, in Fig. 5, To avoid extracting undesired visual features from dynamic object surfaces, the width of the sound source area was set to 10 and 20 degrees for the Spark-T and AIRS Dual Arm Mobile Manipulation robots, respectively. Also, the man in Fig. 4 appeared outside the bounding box. Because in that scene, the sound source was the phone that was farther from the center of his body.

B. Dynamic SLAM Experiments

We evaluate the proposed method by comparing the Absolute Trajectory Error (ATE) of the camera trajectory with the original ORB2, ORB2 with sound source object removal (ORB2+SSL) and state-of-the-art dense reconstruction dynamic SLAM methods SF in multi-robot dynamic environments. Sequences starting with "Sq" and "Mo" using Spark-T robots and AIRS Dual Arm Mobile Manipulation system robots respectively. The ground truth camera trajectories were obtained from a motion capture system.

TABLE II: Time Cost Evaluation

Method	*fps*	GPU
ORB2 [13]	26	$\times$
DynaSLAM [1]	0.3	✓
SF [4]	17	✓
Ours	14	$\times$

Table I lists the ATE Root-Mean-Square-Error (RMSE) of these methods. The original ORB2 method achieved around 25 *cm* ATE in the Sp1 and Sp2 sequences, it can be noticed from the camera trajectories that initially it tracked ground truth well in the Sq1 sequence, and after the moving obstacles appeared, the trajectory went wrong. Our method achieved about one-third of the errors of ORB2 in this scene. The ground truth curves are well tracked by our camera trajectories in Fig. 6. SF VO was not robust in these sequences. Its ATE exceeds several meters, which is due to the tendency of its motion segmentation algorithm to judge moving obstacles as static backgrounds and blocks of pixels in the background as dynamic obstacles when large areas are occluded. This also causes the large and sharp changes in the SF camera trajectories (blue color) and maps in the third image of Fig. 7. ORB2+SSL method also obtained small ATE in several sequences, but it's not robust in loop detection. Therefore, we chose Openvslam over ORB2 because it has better global loop closure capability, as evaluated in the Sq3 sequence, the proposed approach achieved 8 *cm* less ATE than ORB2+SSL, and accurate mapping result (see Fig. 7).

For sparse visual feature-based methods like ORB2, the environment map can be synthesized after acquiring the camera trajectory. Fig. 7 shows the final maps produced by the three methods in the Sq3 and Mo3 sequences, where ORB2 distorts the map severely after the presence of obstacles; SF incorrectly segment another robot as a static background, and our method reconstructs the accurate environment map.

Tab. II compares the online efficiency of several methods. Among them, ORB2 is robust and efficient in static environments and is therefore often used as a base SLAM framework. DynaSLAM and SF are based on GPU support. DynaSLAM is based on ORB2 and pre-processes dynamic objects using Mask R-CNN with very low frame rates. SF achieves efficient online motion segmentation performance but loses robustness when occlusion is high and is therefore not suitable for multi-robot SLAM. Our approach does not rely on GPU and achieves a 14 *fps* while processing seven microphones with 16,000 Hz sound signal sampling and 125 Hz SSL output.

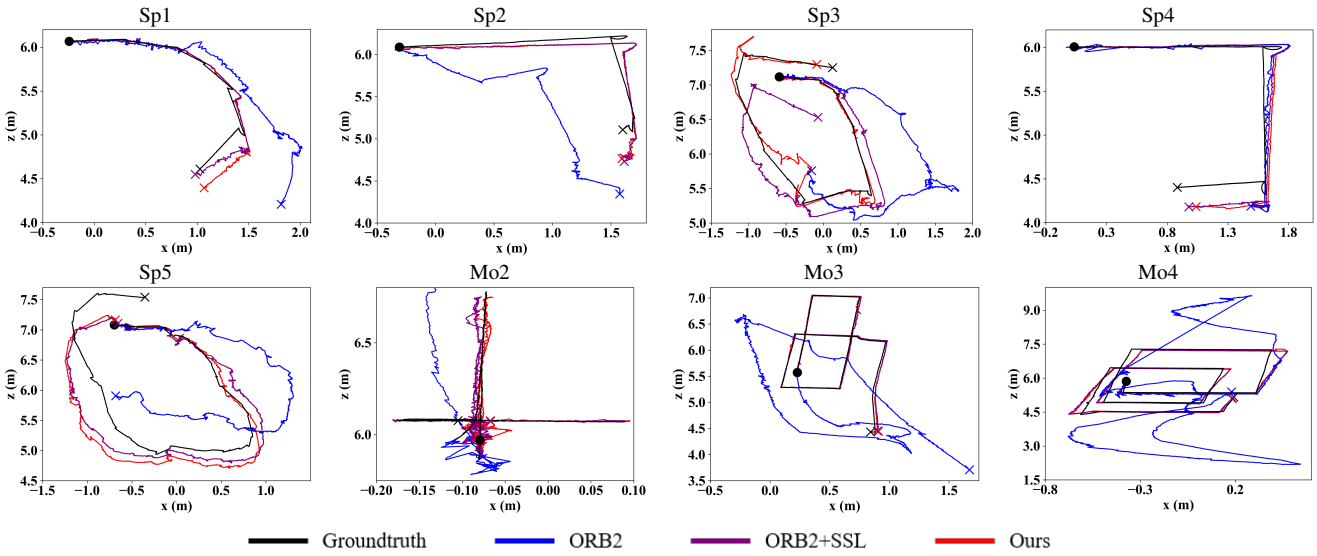


Fig. 6: The plotted trajectories of Spark-T and AIRS Mobile robots sequences. Our method obtains the smallest ATE, and accordingly the red curves are closest to the black ground truth curves.

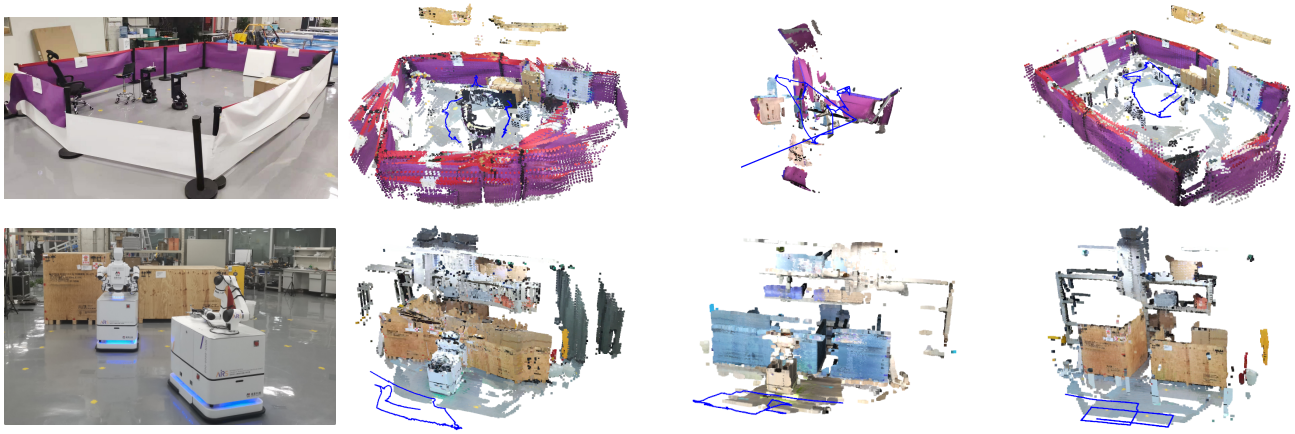


Fig. 7: Dynamic scene mapping results. Top, Sq3 scene. Bottom, Mo3 scene. In both, a robot performs SLAM, another as a dynamic obstacle. The ORB2 and SF methods fail, while our approach build accurate environment maps. The estimated camera trajectories are displayed in blue.

V. Discussions

Large-area occlusion is the biggest problem we encountered in the practical experiments of multi-robot SLAM. When multiple robots approach each other, the images from their cameras are obscured over a large area for the robot behind them. We found that ORB2 vision odometry loses robustness if more than 50% of the pixels are removed as dynamic objects. This is the reason why the proposed method works better on Spark-T than on the much larger AIRS Dual Arm Mobile robot. A similar effect occurs when **multiple dynamic targets** appear in the visual field at the same time, resulting in large areas of invalid visual features. The introducing an ego-motion prior is a promising approach to cope with such rigid object occlusion problem [26].

VI. Conclusions

In this paper, we have presented a new audio-visual fusion approach that fused SSL into visual SLAM. We apply the SSL results as a dynamic object detector to enable dynamic environment SLAM for mobile robots. Experimental results for two different sizes of robots indicate that the proposed method significantly improves the robustness of the visual odometry for the case of severe occlusion in a multi-robot SLAM system. In a multi-robot occlusion scenario, the proposed SSL-based SLAM framework achieves real-time performance using a single CPU and outperforms state-of-the-art GPU-based dynamic SLAM solutions. The future direction of our work is to integrate sound identification into current audio-visual systems for human-robot cooperation.

Acknowledgement

This work is supported by the Shenzhen Institute of Artificial Intelligence and Robotics for Society (2019-ICP002), The Alan Turing Institute and EU H2020 project Enhancing Healthcare with Assistive Robotic Mobile Manipulation (HARMONY, 9911237).

REFERENCES

[1] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[2] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1168–1174.

[3] T. Zhang and Y. Nakamura, "PoseFusion: Dense RGB-D SLAM in dynamic human environments," in *Proceedings of the 2018 International Symposium on Experimental Robotics*. Springer International Publishing, 2020.

[4] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "StaticFusion: Background reconstruction for dense RGB-D SLAM in dynamic environments," in *IEEE International Conference on Robotics and Automation*, 2018.

[5] T. Zhang, H. Zhang, Y. Li, Y. Nakamura, and L. Zhang, "Flowfusion: Dynamic dense rgb-d slam based on optical flow," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7322–7328.

[6] S. Li and D. Lee, "Rgb-d slam in dynamic environments using static point weighting," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2263–2270, 2017.

[7] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE transactions on acoustics, speech, and signal processing*, vol. 24, no. 4, pp. 320–327, 1976.

[8] J. Chen, J. Benesty, and Y. Huang, "Time delay estimation in room acoustic environments: An overview," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, pp. 1–19, 2006.

[9] J. H. DiBiase, H. F. Silverman, and M. S. Brandstein, "Robust localization in reverberant rooms," in *Microphone arrays*. Springer, 2001, pp. 157–180.

[10] C. T. Ishi, O. Chatot, H. Ishiguro, and N. Hagita, "Evaluation of a music-based real-time sound localization of multiple sound sources in real noisy environments," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 2027–2032.

[11] X. Li, Y. Ban, L. Girin, X. Alameda-Pineda, and R. Horaud, "Online localization and tracking of multiple moving speakers in reverberant environments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 88–103, 2019.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[13] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[14] K. M. Judd, J. D. Gammell, and P. Newman, "Multimotion visual odometry (mvo): Simultaneous estimation of camera and third-party motions," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3949–3956.

[15] W. Dai, Y. Zhang, P. Li, Z. Fang, and S. Scherer, "Rgb-d slam in dynamic environments using point correlations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[16] T. M. Hospedales and S. Vijayakumar, "Structure inference for bayesian multisensory scene understanding," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 12, pp. 2140–2157, 2008.

[17] X. Li, L. Girin, F. Badeig, and R. Horaud, "Reverberant sound localization with a robot head based on direct-path relative transfer function," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2819–2826.

[18] Y. Ban, X. Li, X. Alameda-Pineda, L. Girin, and R. Horaud, "Accounting for room acoustics in audio-visual multi-speaker tracking," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6553–6557.

[19] Y. Ban, X. Alameda-Pineda, L. Girin, and R. Horaud, "Variational bayesian inference for audio-visual tracking of multiple speakers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[20] C. Evers and P. A. Naylor, "Acoustic slam," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1484–1498, 2018.

[21] S. Michaud, S. Faucher, F. Grondin, J.-S. Lauzon, M. Labbé, D. Létourneau, F. Ferland, and F. Michaud, "3d localization of a sound source using mobile microphone arrays referenced by slam," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 402–10 407.

[22] G. Xu, H. Liu, L. Tong, and T. Kailath, "A least-squares approach to blind channel identification," *IEEE Transactions on signal processing*, vol. 43, no. 12, pp. 2982–2993, 1995.

[23] X. Li, L. Girin, R. Horaud, and S. Gannot, "Estimation of relative transfer function in the presence of stationary noise based on segmental power spectral density matrix subtraction," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 320–324.

[24] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on Signal Processing,*, vol. 52, no. 7, pp. 1830–1847, 2004.

[25] S. Sumikura, M. Shibuya, and K. Sakurada, "Openvslam: A versatile visual slam framework," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2292–2295.

[26] R. Long, C. Rauch, T. Zhang, V. Ivan, and S. Vijayakumar, "Rigidfusion: Robot localisation and mapping in environments with large dynamic rigid objects," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3703–3710, 2021.