Combining SRP-PHAT and two Kinects for 3D Sound Source Localization

Lucas Adams Seewald, Luiz Gonzaga Jr., Mauricio Roberto Veronez, Vicente Peruffo Minotto, Cláudio Rosito Jung

A R T I C L E   I N F O

Keywords:
Sound Source Localization
SRP-PHAT
Kinect devices

A B S T R A C T

The Kinect™ has been developed to recognize gestures and voice commands, through a set of cameras and microphones, respectively. This paper proposes and evaluates low-cost Sound Source Localization (SSL) solution based on this off-the-shelf equipment. It consists of employing a pair of Kinect devices as an alternative for microphone array, and executing the Steered Response Power using the PHase Transform (SRP-PHAT) localization algorithm over acquired sound data. A fully functional prototype has been implemented and put to test under a realistic scenario. Experimental results indicate that although our approach is capable of achieving limited position estimation, and it can accurately point towards the source’s direction. Two different high performance versions of the algorithm have been implemented to improve overall system performance under 3D Sound Source Localization setup.

1. Introduction

Our starting point is the satellite navigation systems area, where a couple of solutions have been provided for outdoor global geo-spatial position, whereas limitations for indoor geopositioning have been identified. On the other hand, many real world applications such as video conferencing (Wang & Chu, 1997), speech recognition (Nakadai et al., 2011), robotics (Nakadai et al., 2011; Hwang & Choi, 2011) and surveillance (Galatas et al., 2013) have adopted high cost devices to achieve Sound Source Localization (SSL). Because SSL field has experienced a significant growth over the last decades and a variety of algorithmic approaches based on microphone arrays have been proposed and developed. Then, we argue that SSL plus off-the-shelf equipments can be additional technologies for supporting the development of general and unlimited positioning system for costumers, if the high cost of equipments can be amortized.

This research intends to analyse off-the-shelf equipment Microsoft’s kinect, 2013 as an economic alternative for microphone arrays for Sound Source Localization. This paper presents the strengths and weakness of this device, and creates the basis for tests using different devices in future and the integration with outdoor geo-spatial positioning. Our approach, only requires access to raw sound data, being easily adaptable for alternative capture devices, as alternative to some resources regarding echo cancelation and beamforming are already provided by the device’s architecture (Tashev, 2012).

The SSL algorithm of choice is the SRP-PHAT, proposed by DiBiase (2000), due to its accurate results on real case scenarios (DiBiase, 2000; Zhang, Florencio, & Zhang, 2008) and wide popularity. While the accuracy of our solution is still under evaluation, the goal of the current document is to analyse and compare its performance while running over different architectures.

There is few available work searching to exploit Kinect’s microphones as a SSL solution. Many problems can be solved by Kinect’s basic beamforming functionality, which points at the sound source direction in a discrete way instead of giving a full position estimate. An example can be seen in Anzalone, Ivaldi, Sigaud, and Chetouani (2013), where the authors employed the device to make robots capable of looking at human talkers. Others like Galatas et al. (2013) require localization, but achieve it by adding a second Kinect to their system and using both for 2D localization. By crossing two direction lines they are able to locate someone calling for help in an intelligent room built to recognize emergency situations. 3D localization is achieved by the authors through the use of Kinect’s depth camera. Both works resort to a multimodal approach where sound data acts as a complement to visual data acquired by cameras.

Still on the Kinect and SSL subject, the solution that related most to our goal was presented by Tómasson (2012). With the same purpose as Anzalone et al. (2013) (improving human robots...
interaction) the author implemented the GCC-PHAT SSL algorithm to work in pair with the device. However, once again there is no need for actual localization and the method is used to determine the source direction.

Extensive material concerning the SRP-PHAT and its performance is available in literature. A number of real-time implementations have been proposed (Lee & Kalker, 2010; Cobos, Martí, & Lopez, 2011; Pourmohammad & Ahati, 2012). Among prominent optimizations are the Stochastic Region Contraction (Do, Silverman, & Yu, 2007), Coarse-to-Fine Region Contraction (Do & Silverman, 2007) and Stochastic Particle Filtering (Do & Silverman, 2009). A comparative analysis between multithreaded and GPU implementations of the algorithm can be found on da Silveira, Minotto, Jung, and Lee (2010) and Peruffo Minotto, Rosito Jung, da Silveira, and Lee (2013) Test scenarios include a 4 cores CPU executing respectively 1 and 8 threads, and CUDA code running on two different video cards.

Conducted experiments consisted in a series of recording sessions where the sound emitter occupied three different positions. Signals were recorded by a prototype developed for this purpose using two Kinect sensors. Since the emitter positions are known, SRP-PHAT estimates can be easily compared to measured positions. Estimates are evaluated in terms of distance from real position and angle difference from Kinect to both. With the exception of Zhang’s reformulation (Zhang, Zhang, & Florencio, 2007), no optimizations were done to the SRP-PHAT algorithm. All tests perform an extensive search of a given space, which makes them computationally intensive.

Experiments show that our solution is capable of pointing at the sound source with error margins inferior to 4°. It is quite accurate when compared to Kinect’s SDK standard solution, which deals with discrete directions 10° apart. Nonetheless, most observed position estimates were inaccurate, restricting its usefulness in some scenarios.

Following this introduction, Section 2 offers some insight on Kinect’s functionalities. Some concepts and strategies concerning Sound Source Localization are presented in Section 3. The SRP-PHAT itself is detailed in Section 3.1, where its fundamentals are shown. Aspects concerning the developed prototype and its functionalities are explained in Section 4. Section 5 describes our experimental setup further detailing chosen test scenarios. Finally, Sections 6–8 show experimental data, high performance implementations and discuss its implications respectively.

2. Kinect

Microsoft Kinect is an interface device originally created for the Xbox 360 videogame console in 2010. A subsequent version called “Kinect for Windows” aimed at PC platforms was released early 2012. The peripheral was made popular by its differential approach on user interaction, employing gesture and speech recognition instead of a conventional game controller to receive user input.

The device is equipped with an RGB camera, a depth camera and a microphone array. It is probably best known for the way it handles images and depth data in real time, as demonstrated by its dancing games. This paper, however, is focused on the audio capabilities.

Kinect audio arrays are formed by four supercardioid microphones, all facing down. These are asymmetrically distributed along the device, sharing a common horizontal axis. From left to right their distances from each other are 149 mm, 40 mm and 37 mm (Jana, 2012). Each microphone supports a 16 kHz sample rate while offering 32 bit accuracy.

Kinect itself provides resources regarding echo cancellation, beamforming, noise suppression and automatic gain control (Jana, 2012). They can be accessed through Kinect’s SDK, which requires Windows 7 or 8 along with Visual Studio 2010 or higher. Although its beamforming capabilities are able to point at the sound source direction, they do not offer a position estimation. Besides, the pointed direction itself is discrete, covering a 100° angle in 11 directions separated by intervals of 10° each. Also, when judging the devices audio functionalities it is important to remember that they were designed with the human voice in mind. Being that the case, frequencies below 80 Hz and above 1100 Hz are cut off during its processing pipeline.

Professional microphone arrays are often sold as parts of larger solution kits, like the ClearOne (2013) ceiling microphone array, each microphone being sold separately at the cost of around $ 480.00. Also, Microsoft’s Polycom CX5000 (Polycom, 2013), which includes 6 directional microphones and a 360° camera, costs from $ 3000.00 to $ 3600.00. The original Kinect for Xbox 360 costs $ 99.99, while its Windows counterpart costs $ 249.99 ($ 149.99 for students) (Microsoft Corporation, 2012).

3. Sound Source Localization

In the last decades Sound Source Localization (SSL) has been tackled by different approaches, leading to different algorithmic solutions. Many of these methods are based on Time Difference Of Arrival (TDOA) estimation, which addresses the time difference taken by the sound to propagate from a single source towards two spatially separated microphones. Having no previous knowledge about where the sound source is, the TDOA value can lead to a handful of plausible candidate positions. By considering sound propagation speed and acquiring information from more than a single microphone pair, SSL algorithms are able to estimate the position from whence sound originates.

A number of TDOA estimators have been proposed in the literature. The best know is the Generalized Cross Correlation (GCC), proposed by Knapp and Carter (1976). Their method consists of prefiltering sound signals from microphone pairs then calculating their Cross Correlation. The authors originally employed a Maximum Likelihood (ML) (Hamon & Hannan, 1974) weighting function in conjunction with their solution. They also considered prefiltering signals with the following functions: Roth Impulse Response (Roth, 1971), SCOT (Carter, Nuttall, & Cable, 1973), PHAT (Do, 2010) and Eckart (1952).

Weighting functions are usually applied to estimators to address noise and reverberation issues. ML and PHAT filters are the most popular choices, distinguishing themselves for having good performance in the presence of noise and reverberation respectively (Do, 2010). ML filters assume that existing noise in a captured signal bears no correlation to the original signal itself or to any other noise signal captured by another microphone. Although this assumption might be true in most scenarios, it does not apply to reverberation, which clearly relates to the original signal. PHase Transform or PHAT, on the other hand, removes amplitude information from the signal allowing it to focus exclusively on the signal phase. By doing so, it makes the signal less susceptible to general noise, in particular reverberation.

Some SSL algorithms are built upon particular geometric interpretations of the problem and try to solve it directly from a TDOA value set. One example would be the Linear Intersection (Li) (Brandstein, Adcock, & Silverman, 1997) algorithm. The authors point that a single TDOA value from a microphone pair describes an hyperboloid of candidate source positions in three dimensional space. The midpoint of the pair occupies the center of the hyperboloid. According to them, given sufficient distance between the sound source and the microphones, the hyperboloid can be abstracted into a pair of cones. These cones can be further abstracted into lines following their orientation, allowing the source position to be estimated from these lines intersection.
The Spherical Interpolation (SI) (Smith & Abel, 1987) sees the problem from a different angle. It chooses a reference microphone and interpolates a sphere whose surface overlaps this microphone and has its distance from the other ones determined by their TDOA in relation to this reference. Once this sphere is known, the sound source is estimated to be positioned in its center. An important characteristic of this algorithm is that it does not try to compute an exact location, but an approximation which minimizes a chosen error criterion. This approach sacrifices accuracy in order to attain a higher performance gain.

Another group of algorithms uses discrete space representation and beamforming (DiBiase, 2000; Do, 2010) techniques to estimate the source position. Beamforming takes advantage of the fact that noise signals are usually uncorrelated. Using a given TDOA set, a delay-and-sum beamformer performs a temporal alignment of all signals and obtains their averages. Since noise is expected to be uncorrelated, this average would act as a noise suppressor and enhance any sound coming from the chosen position. The Steered Response Power (SRP), a beamformer based SSL algorithm, defines a discrete search grid and fabricates sets of propagation times for each of its points. By executing a delay-and-sum beamformer over each position and measuring its response power, the SRP can associate energy values to every grid point. A higher response power indicates higher proximity to the sound source position. As pointed by DiBiase (2000), this strategy is equivalent to the summation of all possible GCC combinations among microphone pairs. It is interesting to notice that this strategy is able to localize multiple sound sources simultaneously.

Closed-form estimators such as the Li and SI do not require much computational power and may offer accurate results in the absence of noise. This makes them appealing solutions for real-time applications. Unfortunately, at least some amount of ambient noise and reverberation is to be expected for most real case scenarios, which may decrease accuracy considerably (Do, 2010). The SRP-PHAT, detailed next, is built upon the SRP approach is known to perform very well in low SNR scenarios (Zhang et al., 2008). Its computational cost, however, is still considerably higher in comparison to closed-form solutions.

3.1. SRP-PHAT

Proposed by DiBiase (2000), the Steered Response Power using the PHAse Transform (or SRP-PHAT) algorithm, as denoted by its name, combines the SRP localization strategy with the PHAT weighting function. In order to present mathematical aspects of this method some particular SSL solutions and definitions must be detailed first.

Considering an array composed by M microphones, the signal captured by the mth microphone \(x_m(t)\) could be modeled as follows:

\[
x_m(t) = s(t - \tau_m) * h_m(t) + v_m(t), \quad \text{for} \quad m = 1, 2, \ldots, M.
\]

Here the signal is represented by three separated components: the original sound signal \(s(t)\) emitted by the source, the impulse response \(h_m(t)\) caused by the environment on the mth microphone and the uncorrelated noise captured by the same microphone, denoted by \(v_m(t)\). Reverberation is encompassed in the convolution term \(s(t - \tau_m) * h_m(t)\), meaning it is the original signal affected by the structural characteristics of the room (differently from noise, which is considered independent from the source signal). The term \(\tau_m\) is due to the source signal \(s(t)\) reaching each microphone at different moments, that is, \(\tau_m\) represents the travel time of \(s(t)\) to microphone \(M\) commonly denoted as time of arrival (TOA) in most works. Therefore, the term \(s(t - \tau_m)\) allows for a time-aligned representation of the capture signals \(x_m(t)\), for \(m = 1 \ldots M\), which simplifies the formulation of the algorithms that are described next.

A coherence measure is needed in order to tell which estimated delay times yield a stronger relation between signals from different microphones. Applying cross correlation over two frequency domain converted signals has become standard practice for this purpose (Hamon & Hannan, 1974). Assuming microphones \(m\) and \(n\), their cross correlation can be calculated as (Zhang et al., 2007):

\[
C_{mn}(\tau_{mn}) = \int_{-\infty}^{\infty} X_m(\omega)X_n^*(\omega)e^{j2\pi\tau_{mn}}d\omega,
\]

where signals \(X_m(\omega)\) and \(X_n(\omega)\) are the Fourier Transform of the captured signals \(s_m(t)\) and \(s_n(t)\), respectively. The term \(\omega\) stands for the frequency range, while \(e\) and \(j\) are Euler’s number and an imaginary unit respectively. The \(\ast\) symbol represents the complex conjugate. Here \(\tau_{mn}\) represents the TDOA both signals, in other words \(\tau_{mn} = \tau_m - \tau_n\). The idea of this technique is that values of \(\tau_{mn}\) belonging to positions close to the dominant sound source should generate high correlation values, while positions distant from it should generate low ones, and having the actual sound source position earning the highest one.

The SRP is an extension of the cross-correlation algorithm, by taking into account Eq. (2) for every combination microphone pairs. It consists in the discretization a given search space into point grid, from which the sound source of interest will be searched for (assuming it is located within this region). This search is performed by computing, for every point in the grid, the accumulation of Eq. (2) for all possible pairs of microphones. Each accumulation represents the SRP of that given point, and may be mathematically defined as,

\[
P(\tau_1, \ldots, \tau_M) = \sum_{m=1}^{M} \sum_{n=1}^{M} \hat{\Psi}_{mn}(\omega) X_m(\omega) X_n^*(\omega) e^{j2\pi\tau_{mn}} d\omega
\]

where the set of TOAs \(\tau_1, \ldots, \tau_M\) are related to the search point, and \(\hat{\Psi}_{mn}\) represents a frequency domain weighting function relative to the microphone pair \(mn\). Through this method it is possible to attribute a response power (the SRP) to any given point defined inside the search space. Analyzing the SRP of every point constituting the search grid should reveal which one is closer to the sources position, that is, the ones yielding a stronger response power.

Although SRP alone is able to estimate the source position, noise and reverberation can still degrade its accuracy significantly, as previously mentioned. This matter may be mitigated by a good choice for weighting function \(\hat{\Psi}_{mn}\) Eq. (3). The PHAT function, proposed by Brandstein and Silverman (1997), is known for performing well in realistic environments (DiBiase, 2000; Zhang et al., 2007; Rui & Florencio, 2004; Benesty, 2000). Its equation can be seen below:

\[
\hat{\Psi}_{mn}(\omega) = \frac{1}{|X_m(\omega)X_n^*(\omega)|}
\]

Here the weight of frequencies equals the inverse of their components amplitude, that is the removal of signals’ amplitudes. This imposes that loud noises do not have advantage over the original signal. By joining Eqs. (3) and (4), thus combining PHAT and SRP, the SRP-PHAT equation takes form as shown by Zhang et al. (2007):

\[
P(\tau_1, \ldots, \tau_M) = \sum_{m=1}^{M} \sum_{n=1}^{M} \frac{X_m(\omega)X_n^*(\omega)e^{j2\pi\tau_{mn}} d\omega}{|X_m(\omega)X_n^*(\omega)|}.
\]

An optimized form was presented in Zhang et al. (2007), which managed to reduce the number of iterations through microphones and, consequently, reducing computational costs:

\[
P(\tau_1, \ldots, \tau_M) = \int_{-\infty}^{\infty} \sum_{m=1}^{M} \frac{X_m(\omega)e^{j2\pi\tau_{mn}}}{|X_m(\omega)|^2} d\omega.
\]
Once again, the higher the response power, the closest the calculated TDOA set is to the source position.

4. The prototype

A prototype was developed in order to test and evaluate the proposed solution. Its main functionalities are communicating with up to two Kinects and performing the SRP-PHAT. Additional features include showing signal frequency spectrum, storing captured signal into WAVE files and executing over input files instead of directly through microphones.

Sound data is acquired raw from Kinect’s buffers, presented in 32 bit IEEE float PCM format. Sound sample values range from \(-1.0 \text{ to } 1.0\). Each microphone has a channel of its own, so there is a total of four channels in a buffer. Channels are written in interleaved fashion, one sample per sound channel. Fig. 1 shows an example of a buffer filled with sound samples. The set formed by four 32 bit samples captured in the same time interval is called a frame.

Search grid size can be passed as argument to the prototype when started. By telling the program how many points each spatial dimension should have the user controls how SRP-PHAT discretizes the search space. For simplicity’s sake Fig. 2 illustrates a 2D search area and its grid, instead of a 3D search volume as used throughout this work. The Kinect is placed right outside the search space, facing it, as seen in the bottom center of the image. Grid points are marked by circles, indicating positions to be considered by the algorithm. This example has a 4 m wide and 3 m large area divided by a 5 \times 4 point grid, which defines 20 candidate positions to be searched with a 1 m\(^2\) resolution.

The prototype’s main window can be seen in Fig. 3. Here a pair of Kinect sensors is functioning while computing the SRP-HPAT in a 3D space. The first four panels to the left show sound signals as captured by the first Kinect microphones, while right panels show the second Kinect microphones. The left bottom panel shows the energy map generated by SPR-PHAT, in gray-scale, where brighter colors mean higher values. The circled spot in the figure marks the brightest position (largest value), where the program considers as belonging to the sound source. It’s height is hinted at the right bottom panel where a solid color is drawn. Dark tones indicate lower positions and light tones higher ones, ranging from black to white.

The program was fully written in C++ language. User interface was created with the help of Qt library (Digia, 2013). Some parts of the microphone access code resourced to Microsoft’s Kinect SDK (Microsoft’s kinect, 2013). Finally, the prototype used a Fast Fourier Transform implementation made available by the FFTW (Matteo Frigo, 2013) library, which efficiently converted captured time domain signals into frequency domain ones.

5. Experimental setup

To evaluate the proposed solution’s accuracy the prototype was tested in an indoor environment where microphone and emitter positions were known. The place in question is computer laboratory measuring \(4.77 \times 5.90 \times 3.44\) m and is pictured on Fig. 4. By comparing estimated and known emitter positions one can measure the solution’s accuracy. Aside from the emitter, other sound (noise) sources include computers, an air conditioner, crickets, cicadas and human steps. The only noise coming from within the laboratory are from computers functioning, so other sources are not prevalent.

All tests were recorded for latter processing, which ensures test reproducibility and allow for different program configurations over the same sound input. Two Kinects were used for the sound capture, being handled as a single 8 microphone array. To avoid having all microphones horizontally aligned and provide some height difference among their positions, one Kinect was placed vertically, as shown by Fig. 5(a). They were kept relatively close at an arbitrary distance of 24 cm from each other. The role of sound emitter was played by a standard PC speaker seen on Fig. 5(b). The signal of choice is white Gaussian noise at 44.1 kHz sampling rate, which was verified to perform well under additional tests not included here for brevity’s sake.

A top view representation of the laboratory can be seen in Fig. 6. The chosen coordinate system has its axis aligned to the room’s walls. The \(X, Y\) and \(Z\) axis denote depth, width and height respectively. The origin is indicated at the bottom left corner of the image, where 0 height means floor level. A decimal unit in this system measures 1 m.

Although the prototype does compute the Kinetics’ microphone position individually, for clarity we use the devices center as a reference position. The center of the horizontal Kinect, labeled \(k1\), is at coordinate \((0.58, 2.68, 0.78)\), whereas the center of the vertical one, \(k2\), is at \((0.58, 2.92, 0.92)\). Three emitter positions were chosen for the tests, also indicated in Fig. 6 by \(e1\), \(e2\) and \(e3\). Their coordinates are \((4.19, 1.05, 0.14)\), \((3.31, 4.43, 1.68)\) and \((1.60, 3.62, 0.88)\) respectively. All experiments were performed under a single sound source, so there would never be more than one emitter at a given time during experiments.

For each of the three emitter position 10 sound recording sessions were performed. Captured signals were stored in 33 s long WAVE files at 16 kHz sample rate. Since each Kinect’s data is stored individually, a single recording produces two WAVE files. These are four channel files, following the same format described in the previous chapter when dealing with Kinect’s buffers.

Apart from the emitter recordings, 10 additional samples were captured. These were background noise samples where the emitter

![Fig. 1. Example of sound samples in Kinects buffer.](source)
is silent. To make them more representative, recordings were made during different moments along the whole experimental process. Background noise signals were stored in WAVE files following the same specifications from the previous files. This makes a total of 80 sound files created, from which experimental data are acquired.

Background noise samples were used for the signal-to-noise ratio estimation. The 10 samples average power represents noise power. Signal power for each emitter position is acquired by averaging their respective samples, and attained ratios concerning every microphone are presented in the following chapter, along with experimental results.

Moving on to the software’s configuration, a 0.256 s long time window was used for the SRP-PHAT execution. This means the algorithm is performed a total of 130 times for a 33 s recording. Search space was defined right in front of both Kinects, measuring 4.19 \times 5.90 \times 3.44 \text{m}. Since they are placed 58 cm away from the left wall (as seen on Fig. 6), the search volume is 58 cm shorter in length than the environment itself. This volume is made discrete by a 84 \times 118 \times 69 point grid, which makes for a 5 \text{cm}^3 resolution. As a result a total of 6,83,928 SRP-PHATs are verified during a single search.

6. Experimental results

Acquired signal-to-noise ratios for the 8 microphones in each emitter position are illustrated on Fig. 7. Their respective values, the same shown in the figure, are written in Table 1. The ratios range from 14 dB to 22 dB, which ensures that background noises
really were not prevalent over the Gaussian noise. As to be expected, e3, which is closest to the microphones, has yielded the highest signal-to-noise ratio. Differences observed between e1 and e2 were very small though. These ratios also reveal that k2’s microphones were more sensitive to our sound source than k1’s.

Error averages and standard deviations of every SRP-PHAT estimation were computed for each emitter position. Table 2 contains experimental results from precision tests. Position errors correspond to the average and standard deviation from Euclidean distances between estimated and measured positions. For the directional error, consider two lines, one crossing k1 and the emitter’s position and the other crosses k1 and the estimated position. The horizontal error is obtained by measuring the angle between their horizontal plane projections. Analogously the vertical error corresponds to the internal angle formed by their vertical plane projections. As for position, direction error data is presented as average and standard deviation values.

As mentioned before, the SRP-PHAT performs 130 times for every pair of sound files. Since each emitter position is tested 10 times, the algorithm is executed a total of 1300 times per position. So the average and standard deviation values concern 1300 position and direction estimates.

Estimates of emitter positions e1 and e2 have shown large error amounts. While the first average shows about 1.90 m error distance, the second had the position missed by 2.36 m. Their height standard deviation values also points to drastic changes to the algorithm’s answer along its many iterations. Results for the e3 position on the other hand performed quite reasonably. An average error of 12 cm is far from the 5 cm ideal limit suggested by the 5 cm² grid resolution, but is still an improvement over previous positions. Standard deviation values also indicate that e3’s estimates were much more consistent in general.

Although position estimates offered weak results, direction estimates were very accurate. Even emitter coordinates e1 and e2, which had unstable position estimates, had shown less than 4° average direction errors. The worst case occurred while the emitter was placed over e2, resulting in horizontal and vertical average errors of 1.05° and 3.81° respectively. While the SRP-PHAT may choose different positions through its iterations, direction error’s standard deviation do not cross the 2° line. In most cases this error does not reach 1°, which means the program pointed at the emitter consistently through the test process.

In short, our solution is not capable of telling the source’s position with precision, but it can accurately point at its direction. Since two Kinects were employed it is capable of pointing at it in three dimensional space. On top of that experimental results suggest higher accuracy than the standard discrete method provided by the Kinect SDK, which is restricted to a 10° error margin.

Besides room characteristics and prototype settings described here, a number of factors may have influenced the obtained results. Most notably Kinects distance from each other and their facing directions. The solution might benefit from decreasing their proximity or aligning them to different axis. Furthermore, questions arise concerning SRP-PHAT working premises. The algorithm assumes the existence of a direct path between sound source and microphones. Kinect microphones, however, are directed to its base, not the same direction the device is facing. Not only that, but Kinect’s casing itself could be defying this premise.

![Fig. 6. Precision test setup.](image)

![Fig. 7. Signal-to-noise ratio.](image)
more devices and bigger search space can be configured and tested.

throughput maximization. Then, as modern GPU are emerging,
better performance, justified by GPU architecture centered in
mer scenario indicates that amount of data increases GPUs have
214 times as fast as the non-parallelized implementation. The for-
than sequential, while for the scenario 2, the speedup is more than
scenario 01, OpenCL GPU version have obtained 72 times faster
(0.0339 s, 0.1453 s) and OpenCL TESLA (0.03011 s,0.087 s). Fir the
CPU with vectorization (0.0306 s, 0.2778 s), OpenCL QUADRO
(0.0339 s, 0.1453 s) and OpenCL TESLA (0.03011 s,0.087 s). Fir the
scenario 01, OpenCL GPU version have obtained 72 times faster
than sequential, while for the scenario 2, the speedup is more than
214 times as fast as the non-parallelized implementation. The for-
mer scenario indicates that amount of data increases GPUs have better
performance, justified by GPU architecture centered in
throughput maximization. Then, as modern GPU are emerging,
more devices and bigger search space can be configured and tested.

7. High performance directions

Adopted SRP-PHAT have a computational complexity propor-
tional to the number of microphones in the equipment, and mainly
the number of discretized search space. In 3D case, the latter
parameters increase drastically and imposes impracticable levels
of processing in realtime applications.

In order to achieve better performance, we have implemented
two preliminary versions of the algorithms: multi-threaded and
OpenCL implementations, additionally to sequential one. So, we
have planned a set of seven experiments, where these three ver-
sions of the algorithm are evaluated in order to make comparisons
of their execution times, while running on different devices: Intel
Core i7-3930 K (3.20 GHz, 6-cores), NVIDIA QUADRO 5000 GPU
(352-cores), and NVIDIA TESLA GPU (480-cores).

Two search space scenario have been created: Scenario 1
(50 × 50) and Scenario 2 (150 × 150) candidate positions, with fol-
lowing results in Sequential implementation in CPU: 2.18 s and
18 s, respectively. For 6-threads in CPU, the execution times were:
0.382 s (Scenario 1) and 3.494 s (Scenario 2), while for 12-threads
we measured 0.241 s (Scenario 1) and 2.162 s (Scenario 2), respec-
tively. The OpenCL implementation has been tested in three config-
urations OpenCL CPU (without auto-vectorization) OpenCL CPU
(with auto-vectorization), OpenCL GPU (QUADRO) and OpenCL
GPU (TESLA). The results are summarized in (Scenario 1, Scenario
2). OpenCL CPU without vectorization (0.144 s, 1.256 s), OpenCL
CPU with vectorization (0.0306 s, 0.2778 s), OpenCL QUADRO
(0.0339 s, 0.1453 s) and OpenCL TESLA (0.03011 s,0.087 s). Fir the
scenario 01, OpenCL GPU version have obtained 72 times faster
than sequential, while for the scenario 2, the speedup is more than
214 times as fast as the non-parallelized implementation. The for-
mer scenario indicates that amount of data increases GPUs have better
performance, justified by GPU architecture centered in
throughput maximization. Then, as modern GPU are emerging,
more devices and bigger search space can be configured and tested.

8. Conclusion

This paper proposed an economic alternative for 3D Sound
Source Localization based on Kinects microphone array and the
SRP-PHAT algorithm. A fully functional prototype was successfully
developed. This approach allows easily integration with other
applications and supports a higher number of Kinects and addi-
tional SSL algorithms.

After the experiments and results, we have identified that
experiments concerning white Gaussian noise show that, even
though our solution can poorly estimate the source's position, it
can accurately point at the source's direction. This system will be
available for free in VIZLab site (VIZLab.io), for other researchers
and users can test and adapt other devices.

This makes the technique limited fit for fields requiring high
accuracy position data, but useful for navigational systems in gen-
eral whereas geo-spatial position can introduce inaccuracy up cen-
timeters. So, improvements and new evaluation scenarios have
been considered to make the technique and the prototype feasible
to navigational system applications. Currently, it can may be well
suited for room vigilance, video conferencing, gaming scenarios,
and other contexts where cameras or microphones must be redi-
ected according to the situation.

The strength of our approach is to established metrics for joint
usage of off-the-shelf device and SRP-PHAT algorithm for indoor
position estimation and accurate sound source direction. Other-
wise, this approach weakness is on the position estimation, limit-
ing device usage and requiring more effort to fix it. The paper
has provide technological contributions for practical applications,
and information for future adoption or not off-the-shelf low cost
devices for SSL without requiring additional effort.

The 3D positioning has introduced an additional feature to SSL
system, mainly for application that requires spatial indoor posi-
tioning and direction estimation. However, a computation cost
has been added too. So, we have implemented two high perfor-
ance versions of algorithms based on CPU multithreaded, and
CPU and GPU manycore OpenCL, which has achieve to improve
system speedup hugely.

Future steps in this research line include expanding the proto-
type to support additional Kinects, optimizing the SRP-PHAT
through implementation strategies available in literature and fur-
ther evaluating the solution. Main aspects requiring further experi-
mentation are Kinect’s placement alternatives, focusing on position
and orientation, and using human voices as sound source. Also,
extracting their casings for future experiments might prove to be
interesting as well, since paths between source and microphones
are hardly unobstructed considering the plastic structure surround-
ing them. But our main direction is fix position estimation error, and
thus make feasible adoption of Kinect and similar device broadly.
Other scenarios for high performance implementation has been
planned and amplified to improve overall system performance for
different computational systems, mainly mobile manycore devices.

References
engagement with icub. In Biologically inspired cognitive architectures 2012 (pp.
59–64).


