Underwater Sound Filtering

Ines Simic
Rotterdam University of Applied Science,
Research Centre RDM,
Heijplaatsstraat 23, 3089 JB Rotterdam
the Netherlands
inasleeps@gmail.com

Rutger van Aalst
Rotterdam University of Applied Science,
Research Centre RDM,
Heijplaatsstraat 23, 3089 JB Rotterdam
the Netherlands
rutgervanaalst@gmail.com

ABSTRACT

The cocktail party algorithm is one of the most widely used algorithms for source separation of sound. The algorithm aims to find an automated solution for a problem that everyone experiences regularly, namely how to make oneself heard in a noisy environment. The cocktail party algorithm picks up the sound from different microphones, and then applies smart filters once the system has determined which sounds originate from the same source.

This problem also becomes topical when developing sensors based on passive sonar, for instance for autonomous aquatic drones who have to develop awareness of ships and other possible obstacles on a busy shipping lane. It is possible to deploy multiple hydrophones to localize sound sources under water, but the system will be hindered considerably by the sound that the drone itself makes, such as the sound produced by the propellers.

This paper describes a possible solution to the underwater sound filtering problem, using Blind Source Separation. The problem regards splitting sound from a boat engine and the water waves to prove the possibility to extract one sound fragment from the other on the open sea. The illustrations shown further in the report are tests performed in MATLAB to prove the theory.

1. Introduction

Passive sonar concerns techniques to detect and process underwater sound without generating a sound burst first. The latter – active sonar, or sonar for short- is the better known counterpart, and an (in-)famous part of any movie on submarines. As sound tends to travel at long distances under water, hydrophones (underwater microphones) mounted on a ship can experience a mixture of all kinds of sounds, coming from the boat itself, such as the engine, propellers, people talking or working on board, not forgetting the sounds produced around the boat, such as the splashing of water and sea waves against the hull of the ship. In order to focus on a specific source one needs to filter away the sound one does not need. For instance, if hydrophones are deployed to detect ships passing an autonomous aquatic drone, the system will need to block all the sounds that the drone itself produces. Conversely, one may also want to focus on the sounds the drone produces, for instance for maintenance purposes.

Blind signal separation (BSS) – sometimes also referred to as blind source separation- concerns techniques to separate a set of source signals from a set of mixed signals[1]. The cocktail party algorithm provides an interesting solution to this problem, as it aims to filter specific sound sources in very loud and unpredictable environments (people, cities, music). The research carried out at the Rotterdam University of Applied Science aimed see if the cocktail party algorithm is an efficient solution for blind signal/source separation for use under water, which could in principle be used in fairly cheap microcontrollers, such as used in Arduino prototyping boards.

2. Specific problem

As part of a larger research in smart aquatic drones ('wet' robots) carried out at the Research Centre RDM of the Rotterdam University of Applied Science, supervised by professor dr. ir. Ing. C.P. Pieters in cooperation with students from the Informatics department , a research programme is currently carried out to map sound sources under water, using passive sonar. One of the challenges this research aims to address is that an autonomous robot has to learn to recognize a wide variety of sounds which are meaningful for its operation, from the heavy throbbing of a large container vessel, to the calm slapping of waves against a mooring pole, both of which are possible sources of obstructions. The problem of recognition of these wave patterns is exacerbated by the equally wide range of interferences. Also, a single sound source may, directly or indirectly, produce multiple patterns. For instance, the possibility that water waves and engine waves will be correlated is more likely than the correlation of sound of a storm and the sounds produced by loading and unloading containers on board of a ship. The research that was conducted specifically addressed the occurrence of related sound patterns produced by a boat engine and the churning of propellers under water. Intuitively there is a correlation between the sound produced by the motors which are carried through the propeller shafts, and the sounds produced by the propeller blades cutting through the water. We aim to prove that the these sounds can be filtered adequately with BSS, by using the cocktail party algorithm for filtering. Our goal is to extract one sound input from another underwater. We used the unit approach for decreasing complexity, the extraction from one source from a mix of two. The cocktail party algorithm is used to extract sounds which are above water so that the input becomes clear instead of hearing a mixture of interference.

3. Theory

Blind source separation is the separation of a set of source signals from a set of mixed signals. These signals can be easily seen as sound fragments or multidimensional data such as images and tensors. Blind source separation is an interesting approach for maritime application, as one source van usually produce a variety of seemingly unrelated sounds. For instance, when a boat picks up speed, the frequency of the sound of the engines increase, the amplitudes of waves hitting the hull increases, as do the sounds of air and water being mixed around the propeller blades. Without additional help, a system will consider these sounds to be coming from separate sources, while they are in fact related. By mixing these patterns, the system can be trained to see the similarities in the various sound sources, after which techniques can be deployed to either enhance these similarities or filter them.

The theory behind blind source separation involves maximum likelihood estimations, combined with neural networks with applied learning algorithms to obtain a maximum accuracy for a source separated signal. There is, however, one particular class of BSS which concerns *simplified problems*, where it should not be necessary to use such massive calculations. Simplified problems of BSS involves the separation of two sound fragments from a mixture. Massive calculations will not necessarily yield better results for this class of problems, because a maximum likelihood estimation will not necessarily improve the output. The similarities can often be quite straightforward, through for instance amplitude comparison, or Fast Fourier Transformations.

The extraction of two inputs is also known as a two source problem. We chose for a two source problem since we used a sound fragment which exists from an engine fragment and a water fragment which we obtained from the Royal Dutch Navy test recordings published online [6].

$$\begin{bmatrix} \bar{S}_1 \\ \bar{S}_2 \end{bmatrix} \rightarrow \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \rightarrow \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} \rightarrow \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \rightarrow \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \end{bmatrix}$$

$$\bar{S}_1 = [S_{11} \quad S_{12} \quad \Lambda \quad S_{1(n-1)} \quad S_{1n}],\ \bar{S}_2 = [S_{21} \quad S_{22} \quad \Lambda \quad S_{2(n-1)} \quad S_{2n}]$$

Illustration 1: Two source problem

The input comes from the signals which are shown in Illustration 1 here the two signals (S1 and S2) represent two vectors which contain the sampled points of the original fragments in time or frequency domain. The vectors are mixed through a mixing matrix which can be either a time dependent, or independent model. The matrix is displayed in the Illustration as (A). The model we used is space-fixed and time independent. Time dependent models are more likely used when the sources and sensors move in space as function of time. In our case we deal with a simplified sound fragment which does not have this extra requirement. Our fragments both have the same duration. This does not have to be a problem when applying the matrix in real-time situations, as the produced samples (bursts) can be intermittently time-dependent, Matrix (A) uses real input signals so the mixing matrix remains unknown. The (X)'s are the represented mixed signals which are the output of the mixing matrix. Both contain a mix which is specified by the mixing matrix (A), and contains the information of the original sources (S). The (X)'s are the actual two signals which are used to make the separation, since we are extracting a mixture of (S). The weight matrix displayed as (W) is the crucial step in the whole model since this is the place where the estimation takes place, this weight is applied on the mixed signals. The most important objective of blind source separation is to find a linear filter -the weight matrix- so that the output gives a reconstruction of the original source input. To make a linear filter of the known mixing matrix (A) we need to use an inverse of (A). This inverse should be represented in the weight matrix after it is trained.

Although there are different ways to make a linear filter, this solution was chosen since the inverse calculation is the most common form of a two source approach problem. Other research also tend to apply gradient descend methods. For the purposes of our research, where ideally the filtering should be performed on cheap, readily available microcontrollers such as used in the Arduino, a simpler approach is preferable if it yields 'good enough' results. After applying the weight matrix to the signals, (Y1 and Y2) are the resulting estimations of the input source signals (S1 and S2).

$$\begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} * \begin{bmatrix} \bar{S}_1 \\ \bar{S}_2 \end{bmatrix} \text{ and } \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} * \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}$$

Illustration 2: Rewritten input model

In the specific situation of the sounds picked up by hydrophones, we can rewrite the default BSS to the representation of illustration 2, since we actually begin with the mixed signals (X). A microphone always records a blend of all the sounds in its environment. The system therefore receives two mixed inputs (Y), and the weight vector can either separate the mixed signals, or filter them. The result vector is are a mixture of the weight matrix and the mixed signals.

We cannot consider the result vector (Y) to be identical to the source input (S) since there is always a small rustle percentage. What we measure is the sound after it is extracted from the filtered output. As we are mainly interested in filtering the sounds produced by the boat on which the hydrophones are mounted, the effective noise reduction is the most important outcome of the system. The research demonstrated clearly that the sound played after the filtering gives a clear result without interference of the second factor which proves that the extraction is possible. This result is shown further in the research, Chapter five.

4. Approach

To test our theory of applying the cocktail party algorithm to a complex sound surface (the sounds of a mechanical object versus those of water waves), two sound fragments were used that are faithful representations of the kind of sound ones may hear under water. A real testing environment was not practical since we did not have availability of high quality hydrophones and good recording software. We obtained a sound fragment published by the Royal Dutch Navy. One sound fragment is the sound of a boat engine, while the other fragment concerns the water waves of the same boat as it travels through the water. These two fragments were mixed subsequently to isolate the engine from the water and vice versa. The configuration is shown on illustration 3.

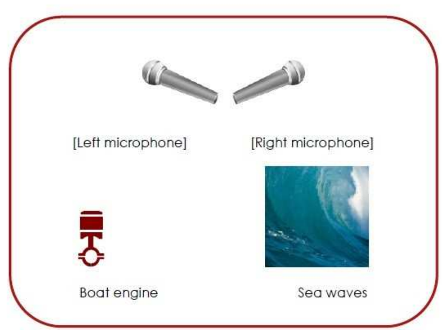

Illustration 3: Configuration sound fragments

5. Matlab

We entered the combined sound fragments of the water and the engine and attempt to separate the two sources/signals. For this we used an implementation of the Cocktail Party Algorithm developed by J.Blyund, for Matlab, which makes the data uncorrelated.

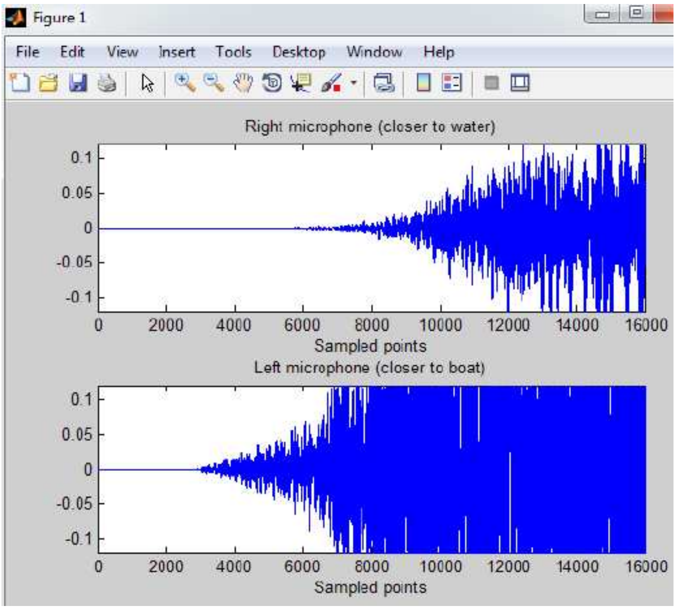

Illustration 4: Sampled points of the two sound fragments

Illustration 4 shows the sampled points of the two fragments. It is obvious that the dominant source is coming from the boat engine. The differences between the two sound fragments remain after a Fourier transform calculation. We implemented the audio files into the matrix, the implementation contains amplitude in the time domain of each sampled point.

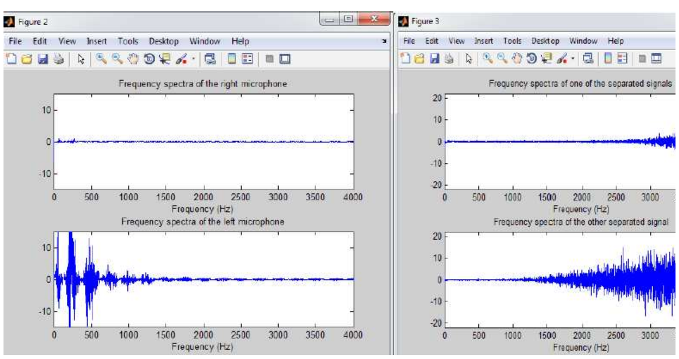

Illustration 5: Spectra of the sounds fragments [left] spectra of the sounds separated [right]

The frequency spectra of the two signals are shown in illustration 5.The dominant factor (hydrophone closer to the boat) shows a huge difference in comparison with the more silent fragment of the microphone closer to the water side. The sampled points are made statistically independent. This is accomplished by random permutation of the order within the sample points. The data is normalized by subtracting the mean from the amplitude in each point, by then multiplying the statistically independent vectors with the inverse of the square root of the covariance matrix. The variance of the two signals should then be almost equal, and the covariance between them close to zero. Subsequently the two signals were divided into small blocks which contain a small number of points. When this was accomplished the actual source separation could finally start.

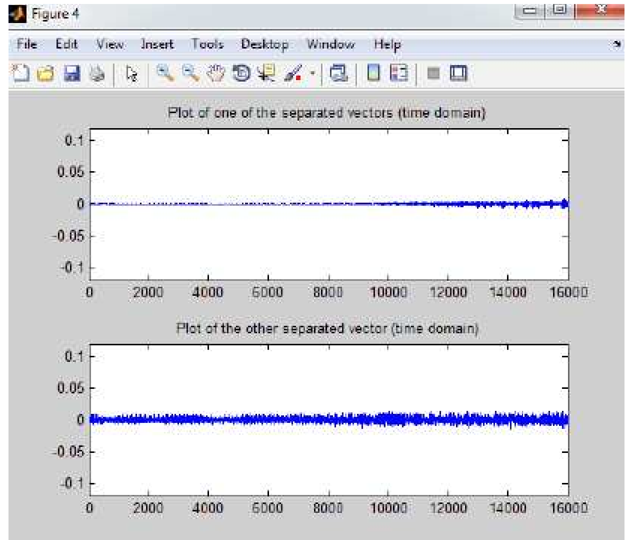

Illustration 6: Time domain (separation)

The separation goes through the blocks and adjusts the weights after each iteration. Each point is multiplied with the value of the corresponding weight. If the weight-factor is large, the amplitude of that corresponding point in the resulting vector will be high. This procedure is then looped several times, resulting in a further amplification of the dominant points. The weight is calculated by taking the previous weight and adding it to the product of the learning rate and the function f. Illustration 6 shows a result of the separated sound fragments after filtering. In Illustration 4 we start with the fragment which contains the engine sound and the water sound, these two combined show a huge mutual interference. After filtering out one sound fragment from the combination of two Illustration 6 shows the signal which was not filtered away, in our case this is the sound of water.

6. Conclusion

Our research demonstrated that it is possible to filter sound underwater but it has to be a clear sound fragment, otherwise it is not possible to filter the sound away with a two-source approach. The most remarkable issue during this research was that it is more successful to filter away two object sounds than two human voices interacting with each other. The background research showed that human voices have a large Gaussian extract which causes similarity while combined sounds work on a other frequency domain and are less complicated to divide than a bunch of people talking in the same room, which obviously is the target problem for the Cocktail Party Problem. This gives ample possibilities for further application in maritime environments, and for filtering sounds which can only be found under water.

The method we used is dynamic and can be applied to different sound filtering issues which are not bound to a time dependent model. We used a time independent source for high leveled filtering. When a time dependent model is needed which moves in space as a function of time it gets more complex, projects like these rely on great factors which include calculating likelihoods, higher evolved algorithms and neural networking. We tested our theory with a time independent model which gave us a positive outcome and more knowledge regarding underwater filtering. We also encountered the difference between the starting position and the ending position of the visualizations. They show a different kind of data visualization since we were not able to eliminate the small percentage of rustle. This rustle came into existence when the recording took place of this particular sound. It could be a rustle from the microphone or something else in the environment. We are not sure what caused it but it plays a huge roll why the data can't be identical. For such eliminations it can take a real sound signal expert years of research to reduce the rustle.This surely was a great challenge for two computer science students.

7. Acknowledgments

We want to thank J. Bylund from the University of Sydney who wrote the Matlab implementation code, his work helped us a lot to solve the underwater filtering problem. We also want to thank Kees Pieters, Professor of Applied Science from the department Mainport Information Research" of our research centre RDM for helping us in the research that was conducted

8. References

[1] Amari S. Cichocki A. and Yang H.H. (1996) " A new learning algorithm for blind signal separation." Advances in Neural Information Processing Systems 7, MIT press [online] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.1433 .

[2] Amari S. (1997) "Natural Gradient Works Efficiently in Learning" [online] http://www.maths.tcd.ie/~mnl/store/Amari1998a.pdf .

[3] Paraga N. Nadal J-P(1996) "Blind Source Separation with Time Dependent Mixtures" [online] www.ft.uam.es/neurociencia/ARTICLES/timdepsigproc.ps.gz .

[4] Blyund, J (2001) "Scientific Computing Project" [online] http://www.vislab.uq.edu.au/

[5] Ng, Andrew (2013) Machine Learning, Cocktail Party Algorithm (Coursera) [online] www.coursera.org .

[6] Dutch Royal Navy (2011) Sound Fragment [online] Visited on 10.07.2014 15:30 http://www.dutchfleet.net/ , http://wwww.sounddogs.com

[7] Dima Batenkov (2005) "Fast Fourier Transform" [online] http://www.wisdom.weizmann.ac.il/~naor/COURSE/fft-lecture.pdf

[8] E.Oran Brigham (1988) "The Fast Fourier Transform and its applications" [book] ISBN-13: 978-0133075052

[9] Pierre Comon, Christian Jutten (2010) "Handbook of Blind Source Separation" [book] ISBN: 978-0-12-374726-6

[10] Xianchuan Yu, Dan Hu, Jindong Xu (2014) "Blind Source Separation" [book] ISBN: 978-1-118-67984-5

SRC 2016, November 30, 2016, TheNetherlands.