Neural Network Enhanced Sound Source Localization Model for Mobile Terminal

Anthony Y. Fu [1], Danxia Lin [2], and Gang Liu [3]

[1]Department of Computer Science, City University of Hong Kong, Hong Kong SAR
anthony@cs.cityu.edu.hk
http://www.cs.cityu.edu.hk/~anthony
[2]National Engineering Research Center for Enterprise Information Software, Beijing, China
lindanxia@tsinghua.org.cn
[3]MOFCOM China International Electronic Commerce Center Cofortune Information Technology Co. Ltd., Beijing, China
liugang2001@tsinghua.org.cn

Abstract. Video mobile terminal technology has been mature and came into people's daily life. Face to face conversation and group meeting through video mobile phone become standard usage of video mobile service. People will put the mobile phone in front of his/her face when they use this kind of service. But users in noisy places will be interfered by surrounded noise. Former researchers have tried to use microphone array to solve this problem. And sound source localization model (SSLM) is the first approach. In this paper, we used feed forward neural network (FFNN) to simulate the hyperplane to map the detected time intervals to space coordinates. Thus the SSLM computation complexity is reduced and automatic adaptability is achieved. Experiments have been done to show the feasibility and high accuracy of this method. Furthermore, Levenberg-Marquardt (LM) algorithm has been proposed to be our training algorithm.

1. Introduction

Mobile systems have been developed dramatically fast in the past decades. And the 3G mobile phone became new adorable communication tool for recently. The 3G mobile services provide us up to 384k bps data rate, and one of the most attractive services is video phone. People could talk to each other face to fact using this service. But there still is a noise problem for it. Users in noisy places are suffering more in this kind of occasions. They are interfered by surrounded people and machine with high decibel. Therefore, whether expected voice or unexpected noise are both transmitted through microphone. Former researchers have design or used microphone array [3, 5, 10, 12] to reduce the noise. And also, researches on sound source tracing [6, 11] and SSLM [1, 4, 7, 9] have been done. To guarantee accuracy, former SSLMs have used mathematical and power consuming techniques to do computation. E.g., IBM has built up a microphone array to trace the speaker direction with 4 sensors in 2001, the 4 sensors are linearly installed and they only filter in the voice from a vertical plane in 3D space. So any voice from the plane could be filtered in to the video mobile

terminal. And they used complicated hardware to solve the high computation complexity problem. But they are not adaptive and not fit for power-limited mobile devices. In this paper, a new SSLM has been introduced to eliminate these disadvantages. There are 4 microphones have been used to localize the sound source position in 3D space (It can be proved that 4 is the theoretical minimum microphone number). Moreover, we have used FFNN with back propagation (BP) to simulate a hyperplane to simulate the mapping from the detected time intervals to 3D space coordinates. So the sound source could be localized in a 3D space rather than in a planar space; the computation complexity is be reduced; and automatic adaptively is achieved. Experiments have been done to testify the method. To compare the performance of the NN training algorithms with different hidden neuron numbers, we have used ten training algorithms combinating with 10 different neuron numbers ranging from 5 to 50. Thus we could find the least needed hidden neuron number and training algorithm with the best performance.

2. Model and Analysis

A. Sound Source Localization Model

The space coordinate for the SSLM is established for the first step, as shown in Fig.1. The discussion is based on 3D coordinate *XYZ.* The origin point is *O* (*0, 0, 0*). The formula of plane *ABC* is $x + y + z = a$. *A* (*a, 0, 0*), *B* (*0, a, 0*) and *C* (*0, 0, a*) are intercepts of plane *ABC*. And *a* is the supposed length of *OA, OB* and *OC*. There are four microphones at *A, B, C* and *O* respectively. The four microphones should be installed on a video mobile terminal, and the plane *ABC* is supposed to be parallel with the mobile phone front face. Suppose there is a sound source *S* (s_x, s_y, s_z) in the first quadrant of *XYZ* and out side of pyramid *OABC*. Thus the constrains of *S* could be presented as

$$s_x + s_y + s_z > a \text{ , where } s_x, s_y, s_z > 0$$

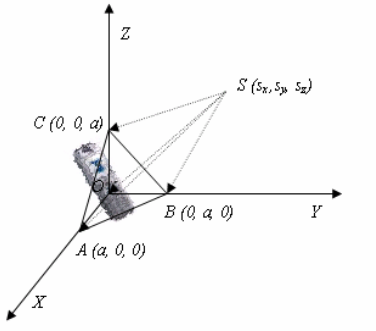


Fig. 1. VFM Geometric Model

The length of *SO, SA, SB, SC* will be represented as Formula (1). Suppose sound is generated from S, it will be received at *O, A, B and C* at time T_0, T_A, T_B and T_C respectively. Although the microphones cannot detect the voice generating time, the

3D coordinate of *S* still could be calculated by solving the following series of equations.

$$\begin{cases} SO^2 = s_x^2 + s_y^2 + s_z^2 \\ SA^2 = (s_x - a)^2 + s_y^2 + s_z^2 \\ SB^2 = s_x^2 + (s_y - a)^2 + s_z^2 \\ SC^2 = s_x^2 + s_y^2 + (s_z - a)^2 \\ SO, SA, SB, SC > 0 \end{cases} \quad (1)$$

Suppose $T_a=T_O-T_A$, $T_b=T_O-T_B$ and $T_c=T_O-T_C$. The speed of sound in the air could be represented as $v \approx 331.4 + 0.6T \ \ m/s$, where T is the Celsius temperature. Then we will have Formula (2). The fully simplified root of s_x, s_y, s_z could be presented as it is shown in Formula (3).

$$\begin{cases} T_a v = SO - SA = \sqrt{s_x^2 + s_y^2 + s_z^2} - \sqrt{(s_x - a)^2 + s_y^2 + s_z^2} \\ T_b v = SO - SB = \sqrt{s_x^2 + s_y^2 + s_z^2} - \sqrt{s_x^2 + (s_y - a)^2 + s_z^2} \\ T_c v = SO - SC = \sqrt{s_x^2 + s_y^2 + s_z^2} - \sqrt{s_x^2 + s_y^2 + (s_z - a)^2} \\ v \approx 331.4 + 0.6T \end{cases} \quad (2)$$

$$\begin{aligned}
&Sx == \\
&\quad \left(a^5 Tc - a^3 Tc \left(Ta^2 + Tb^2 + Tc^2 - Ta\,(Tb + Tc)\right) v^2 + a\,Ta\,Tc \left(-Tb^3 - Tc^3 + Ta\left(Tb^2 + Tc^2\right)\right) v^4 + \right. \\
&\qquad Ta \\
&\qquad \sqrt{\left(a^2 Tc^2 v^2 \left(3\,a^6 + 2\,a^4 \left(-2\,Ta^2 - 2\,Tb^2 + Tb\,Tc - 2\,Tc^2 + Ta\,(Tb + Tc)\right) v^2 + \right.\right. \\
&\qquad\quad a^2 \left(Ta^4 + Tb^4 - 2\,Tb^3\,Tc + 4\,Tb^2\,Tc^2 - 2\,Tb\,Tc^3 + Tc^4 - 2\,Ta^3\,(Tb + Tc) + 4\,Ta^2\left(Tb^2 + Tc^2\right) - 2\,Ta\left(Tb^3 + Tc^3\right)\right) v^4 - \\
&\qquad\quad \left.\left.\left(Tb^2\,(Tb - Tc)^2\,Tc^2 + Ta^4\left(Tb^2 + Tc^2\right) - 2\,Ta^3\left(Tb^3 + Tc^3\right) + Ta^2\left(Tb^4 + Tc^4\right)\right) v^6\right)\right)\Big/\left(2\,a^2\,Tc\left(a^2 - \left(Ta^2 + Tb^2 + Tc^2\right) v^2\right)\right), \\
&Sy == \\
&\quad \left(a^5 Tc - a^3 Tc \left(Ta^2 - Ta\,Tb + Tb^2 - Tb\,Tc + Tc^2\right) v^2 + a\,Tb\,Tc \left(-Ta^3 + Ta^2\,Tb + (Tb - Tc)\,Tc^2\right) v^4 + \right. \\
&\qquad Tb \\
&\qquad \sqrt{\left(a^2 Tc^2 v^2 \left(3\,a^6 + 2\,a^4 \left(-2\,Ta^2 - 2\,Tb^2 + Tb\,Tc - 2\,Tc^2 + Ta\,(Tb + Tc)\right) v^2 + \right.\right. \\
&\qquad\quad a^2 \left(Ta^4 + Tb^4 - 2\,Tb^3\,Tc + 4\,Tb^2\,Tc^2 - 2\,Tb\,Tc^3 + Tc^4 - 2\,Ta^3\,(Tb + Tc) + 4\,Ta^2\left(Tb^2 + Tc^2\right) - 2\,Ta\left(Tb^3 + Tc^3\right)\right) v^4 - \\
&\qquad\quad \left.\left.\left(Tb^2\,(Tb - Tc)^2\,Tc^2 + Ta^4\left(Tb^2 + Tc^2\right) - 2\,Ta^3\left(Tb^3 + Tc^3\right) + Ta^2\left(Tb^4 + Tc^4\right)\right) v^6\right)\right)\Big/\left(2\,a^2\,Tc\left(a^2 - \left(Ta^2 + Tb^2 + Tc^2\right) v^2\right)\right), \\
&Sz == \\
&\quad \left(a^5 - a^3 \left(Ta^2 + Tb^2 - (Ta + Tb)\,Tc + Tc^2\right) v^2 + a\,Tc \left(-Ta^3 + Ta^2\,Tc + Tb^2\,(-Tb + Tc)\right) v^4 + \right. \\
&\qquad \sqrt{\left(a^2 Tc^2 v^2 \left(3\,a^6 + 2\,a^4 \left(-2\,Ta^2 - 2\,Tb^2 + Tb\,Tc - 2\,Tc^2 + Ta\,(Tb + Tc)\right) v^2 + \right.\right. \\
&\qquad\quad a^2 \left(Ta^4 + Tb^4 - 2\,Tb^3\,Tc + 4\,Tb^2\,Tc^2 - 2\,Tb\,Tc^3 + Tc^4 - 2\,Ta^3\,(Tb + Tc) + 4\,Ta^2\left(Tb^2 + Tc^2\right) - 2\,Ta\left(Tb^3 + Tc^3\right)\right) v^4 - \\
&\qquad\quad \left.\left.\left(Tb^2\,(Tb - Tc)^2\,Tc^2 + Ta^4\left(Tb^2 + Tc^2\right) - 2\,Ta^3\left(Tb^3 + Tc^3\right) + Ta^2\left(Tb^4 + Tc^4\right)\right) v^6\right)\right)\Big/\left(2\,a^4 - 2\,a^2\left(Ta^2 + Tb^2 + Tc^2\right) v^2\right)
\end{aligned} \quad (3)$$

It is obvious that these formulas are complicated. It will be a too heavy SSLM consumption for mobile terminal CPU, saying a 3G mobile phone. Moreover, there are errors coursed by calculation, microphone response time, sound reflection and electrical legacy, etc. exist. We have used FFNN to reduce the errors and it can also make the SSLM to be adaptive, which means error coursed by microphone response time, sound reflection and electrical legacy could be eliminated. The mapping relationship between sound source location and detected time intervals can be simulated by FFNN. The weight matrix and bias vector can be trained in advance before putting them into use.

B. Neural Network Model

As NN have been researched and built for decades of years. The most fantastic usages of them are pattern recognition and function approximation. And here, the time intervals to sound source coordinates mapping approach is based on FFNN with BP. We used 2 neuron layers to accomplish the hyperplane simulation tasks, one hidden layer and one output layer. The inputs parameters are a series time intervals T_a, T_b and T_c. The output is the calculated sound source coordinates, as shown in Fig.2.

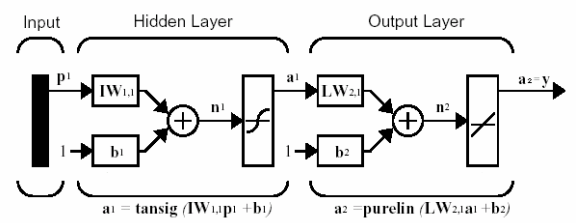


Fig. 2. Feed forward neural network with back propagation

The input parameters are represented as an R by 1 vector $p1$.

$$a1=tansig\ (n1)$$
$$=\frac{e^{IW_{1,1}p1+b1}-e^{-IW_{1,1}p1-b1}}{e^{IW_{1,1}p1+b1}+e^{-IW_{1,1}p1-b1}}, \quad (4)$$

where $n1=IW_{1,1}\ p1+b1$.

In the hidden layer, tangent sigmoid function has been used as transfer function. Suppose we use S neurons in this layer, and the input weight matrix from input p1 to layer 1 is denoted as $IW_{1,1}$, which is an S by R matrix. And the bias vector $b1$ is S by 1. The output of hidden layer $a1$ is an S by 1 vector too, which can be represented as Formula (4)

$$a2=LW_{2,1}\times a1+b2$$
$$=LW_{2,1}\ tansig\ (n1)+b2$$
$$=LW_{2,1}\times\frac{e^{IW_{1,1}\ p1+b1}-e^{-IW_{1,1}\ p1-b1}}{e^{IW_{1,1}\ p1+b1}+e^{-IW_{1,1}\ p1-b1}}+b2, \quad (5)$$

where $n1=IW_{1,1}\ p1+b1$.

In the output layer, pure-linear function is used and only one neuron is used. The layer weight matrix from layer 1 to layer 2 can be denoted as $LW_{2,1}$, and it is a 1 by S vector. The bias vector b2 is a 1 by 1 scalar. The output of output layer $a2$ is a 1 by 1 scalar too. It physically means the waiting time prediction result. And it is represented as $a2=purelin\ (LW_{2,1}\times a1+b2)$, where $purelin\ (n2)=n2$. So, we get $a2=n2$. And finally, we have Formula (5).

After training process on this NN, $IW_{1,1}$ and $LW_{2,1}$ are S by R and S by 1 matrices respectively with determined factor values. The physical meaning of R is the parameter number in one input vector and S is the neuron number in the hidden layer.

There are many kinds of back propagation algorithms. Based on the experiments, we propose LM algorithm as our training algorithm, because LM has the best training speed on the same precision basis. The LM algorithm is a variation of Newton methods, and this algorithm was designed to approach second-order training speed

without having to compute the Hessian matrix. As we used mean squared error (MSE) to be the performance function and the output vector a2 is scalar, Hessian matrix can be approximated as $H=J^TJ$, where J is Jacobian matrix that contains first derivatives of the neural network error with respect to the weights and biases. The Jacobian matrix can be computed through a standard back propagation technique [20] that is much less complex than computing the Hessian matrix [18].

The LM algorithm update could be represented to be Formula (6).

$$x_{k+1} = x_k - \alpha_k g_k, \quad \alpha_k = [J_k^T J_k + \mu I]^{-1},$$
$$g_k = J_k^T e_k$$
$$\text{so, } x_{k+1} = x_k - [J_k^T J_k + \mu I]^{-1} J_k^T e_k, \quad (6)$$

$$J_k = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_{1,1}^1} & \frac{\partial e_{1,1}}{\partial w_{1,2}^1} & \cdots & \frac{\partial e_{1,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,1}}{\partial b_1^1} & \cdots \\ \frac{\partial e_{2,1}}{\partial w_{1,1}^1} & \frac{\partial e_{2,1}}{\partial w_{1,2}^1} & \cdots & \frac{\partial e_{2,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{2,1}}{\partial b_1^1} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_{S^M,1}}{\partial w_{1,1}^1} & \frac{\partial e_{S^M,1}}{\partial w_{1,2}^1} & \cdots & \frac{\partial e_{S^M,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{S^M,1}}{\partial b_1^1} & \cdots \\ \frac{\partial e_{1,2}}{\partial w_{1,1}^1} & \frac{\partial e_{1,2}}{\partial w_{1,2}^1} & \cdots & \frac{\partial e_{1,2}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,2}}{\partial b_1^1} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

S^M denotes for max row number in the output layer

$e_{p,q}$ denotes for the error of output value indexed by (p, q) in the output matrix

$w_{q,r}^p$ denotes for the value indexed by (q, r) in the weight matrix of layer p

b_q^p denotes for p^{th} value in the biases vector of layer p

x_k is a vector of current weights and biases
g_k is the current gradient
α_k is the learning rate
J_k is the Jacobian matrix

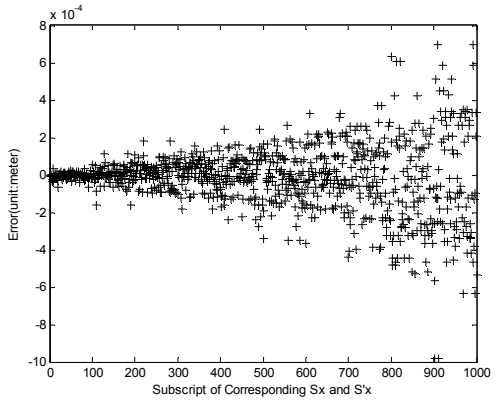
3. Simulation and Analysis

Based on the SSLM and FFNN model, we did experiments to show the feasibility and high performance of this method.

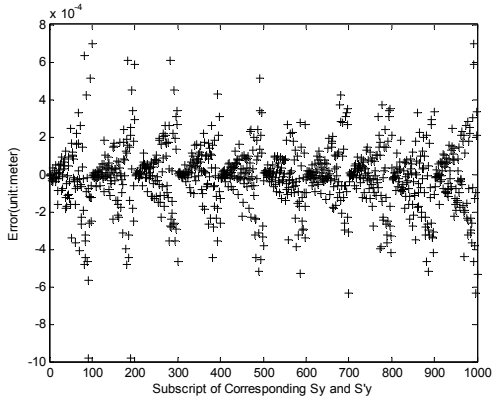
Step1, Generate the mapping relationship from T_a, T_b and T_c to sound source coordinate.

Suppose the sensor distance $a=5$ cm, and speed of sound in air $v=340$ m/s (when air temperature $T=14.3\ ^oC$) for explicit discussion. S_x is assigned to be a series of equal-distance numbers 5, 10, 15 … 50 cm, and the same to S_y and S_z. So we got 1000 samples of sound source point $S(S_x, S_y, S_z)$. $S_1=(5, 5, 5)$, $S_2=(5, 5\ 10)$ … $S_{1000}=(50, 50, 50)$. Time difference vector $T_v(T_a, T_b, T_c)$ can be calculated according to Formula (2).

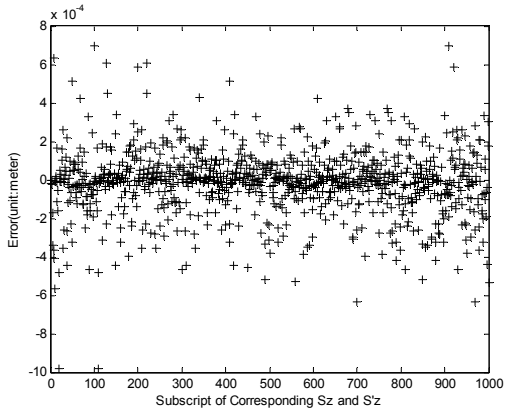
Thus the accuracy is guaranteed, and we got 1000 time difference vectors T_{v1}, T_{v2}… T_{v1000}. *S'* recalculated using the Formula (3), as shown in Fig.3.(a), (b) and (c)



(a)



(b)



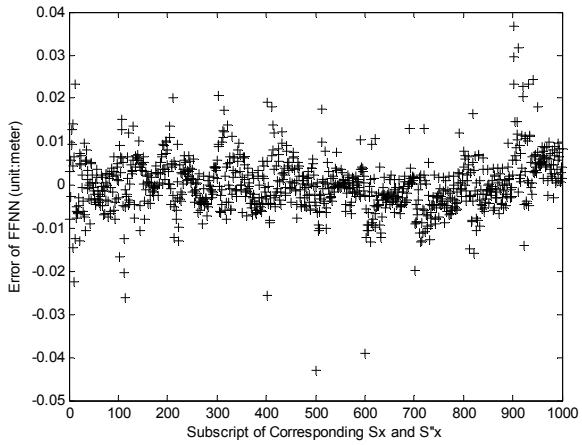
(c)

Fig. 3. The error of recalculated S'

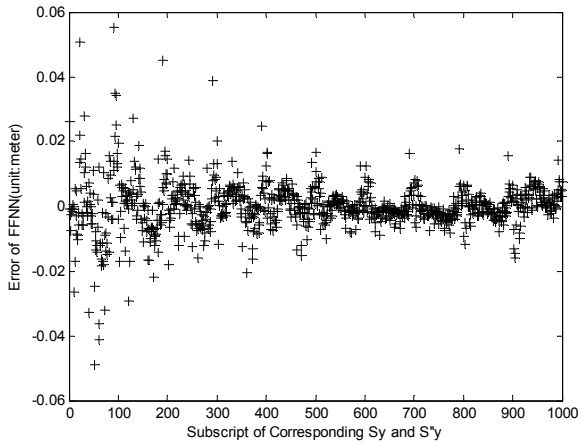
It is obvious that largest error is no more than 0.001 cm, which means that the series of T_v are reliable.

Step 2, Do FFNN training to simulate the mapping from T_v to S.

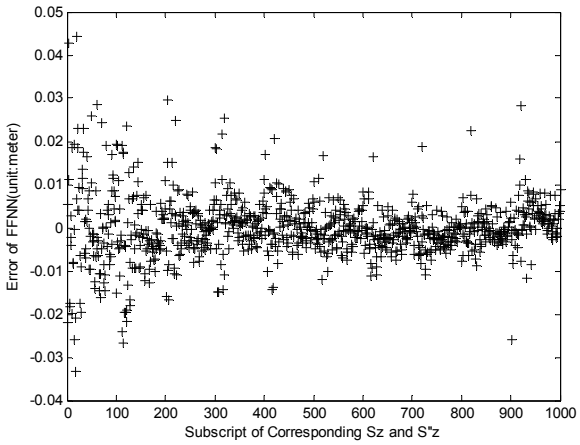
There are 10 different FFNN have been used in the experiments. The only difference among them is the hidden neuron number. These numbers ranges from 5 to 50. And each of type of neural network has been tried using 9 training algorithms, BFGS-Quasi- Newton (BFG), Resilient Back propagation (RP), Scaled Conjugate Gradient (SCG), Conjugate Gradient with Powell/Beale Restarts (CGB), Fletcher-Powell Conjugate Gradient (CGF), Polak-Ribiére Conjugate Gradient (CGP), One-Step Secant (OSS) and Variable Learning Rate Back propagation (GDX), which are the most famous training algorithms.



(a)



(b)



(c)

Fig. 4. Error of The trained FFNN with 25 Hidden Neurons

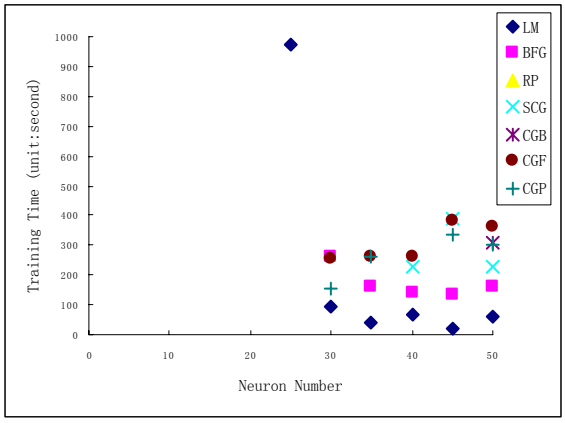


Fig. 5. Training Time with Different Hidden Neuron Numbers and Training Algorithms

There are totally 10×9=90 types of neural network training have been done. We have set mean squared error MSE≤0.001m^2 to be the training goal. Moreover, the ceiling of training epoch number to 5000. That means whether MSE is measured to be less than 0.001m^2 or the epoch number is up to 5000, the training process will stop. And other cases will treated as neural network cannot be well trained or training algorithm is too slow to converge. It can be found from Appendix that RP, OSS and GDX are not good training algorithms for this SSLM. They can not let MSE converge to be less than 0.001 in 5000 epochs using any hidden neuron numbers been provided. Other training algorithms can let MSE converge to be less than 0.001 in 5000 epochs using some neural networks that have been provided. And it is also found that neural networks with hidden neuron number less than 25 behave relatively bad in the experiments, even if MSE can be converged to 0.001 in 5000 epochs in some of them. The trained neural network with 25 or more hidden neurons could simulate the time

latency to sound source coordinate with similar precision. The largest error in one dimension is less than 6 cm. As shown in Fig.4. So we propose to use more than 25 hidden neurons for precision. And we could find in that the training time of LM algorithm is decreasing when the hidden neuron number is increasing, as shown in Fig.5. It should be mentioned that all data represented in Fig.5. meet 2 conditions. First, the training processes should be accomplished in 5000 epochs to reduce MSE to be less than 0.001, and second, error should be less than 6 cm. Obviously, LM use less time than other training algorithms. So we propose use LM for this SSLM.

As the largest error in one dimension is less than 6 cm. The accuracy can be guaranteed if sound source is not near to the theoretical boarder we expected. As shown is Fig.6. Suppose we define the boarder to be a cone surface. The vertex of the cone is at O, and the black circle is one of section of cone surface. And suppose there is a red cone in side the black one. The red circle is one section of the red cone surface. When any points in the red cone is tuned to be more than 6 cm away from the black cone. The voice from sound source in the red cone could be received with no problem. SSLM could be used to calculate any sound source coordinate in a defined volume space and the guaranteed space volume could be calculated, vice versa.

4. Conclusion

As mobile terminal with video phone service is becoming a common service in modern mobile communication world. The face to face talking model needs a utility to calculate the sound source coordinate, which will be helpful to reduce the surrounding noise in open environments. We provided a new method to calculate the coordinate of sound source in a 3D space. And FFNN has been used to simulate the mapping relationship between microphone detected time legacy and sound source position. The usage of FFNN in this SSLM provided an efficient and adaptive method to accomplish sound source localization task. Experiments have been done to show the feasibility of this method. Moreover, a training algorithm LM has been proposed for best FFNN training performance.

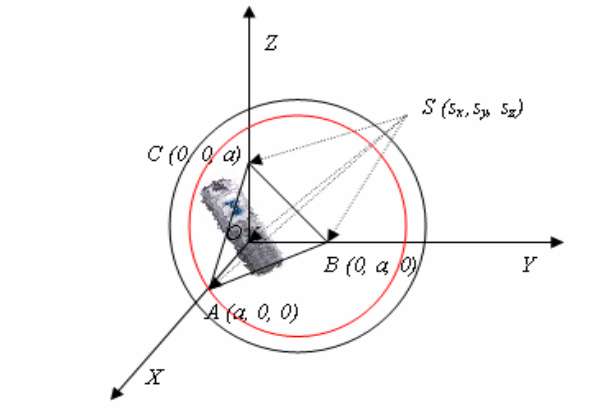


Fig. 6. Guaranteed Space Volume for VFM

5. Future Work

FFNN can be used to simulate SSLM the mapping relationship between time legacy and sound source position. And the mapping accuracy has been analyzed in this paper through experiments. This work has been done in the virtual environment, and this SSLM method is looking forward to be turned into reality. It is obvious that FFNN has very good property to mapping processes. So even in case of electronic legacy, interferences and other unexpected reasons exist, the mapping from time legacy to sound source coordinates will still behave as normal as discussed in this paper. Therefore, the most direct and efficient future work is to use real hardware to simulate neural network and realize this SSLM.

Reference

1. Kozick, R.J.; Sadler, B.M.; Source localization with distributed sensor arrays and partial spatial coherence, IEEE Transactions on Signal Processing, Volume: 52 ,Issue: 3 , March 2004, Pages:601 – 616
2. Zheng, Y.R.; Goubran, R.A.; El-Tanany, M.; Robust Near-Field Adaptive Beamforming With Distance Discrimination, IEEE Transactions on Speech and Audio Processing, Volume: 12, Issue: 5, Sept. 2004, Pages:478 – 488
3. Ma, W.-K.; Ching, P.-C.; Vo, B.-N.; Crosstalk Resilient Interference Cancellation in Microphone Arrays Using Capon Beamforming, IEEE Transactions on Speech and Audio Processing, Volume: 12, Issue: 5, Sept. 2004, Pages:468 – 477
4. Mungamuru, B.; Aarabi, P.; Enhanced Sound Localization, IEEE Transactions on Systems, Man and Cybernetics, Part B,, Volume: 34 , Issue: 3 , June 2004, Pages:1526 – 1540
5. Zheng, Y.R.; Goubran, R.A.; El-Tanany, M.; Experimental Evaluation of a Nested Microphone Array with Adaptive Noise Cancellers, IEEE Transactions on Instrumentation and Measurement,, Volume: 53 , Issue: 3 , June 2004 ,Pages:777 – 786
6. Potamitis, I.; Chen, H.; Tremoulis, G.; Tracking of Multiple Moving Speakers with Multiple Microphone Arrays, IEEE Transactions on Speech and Audio Processing, Volume: 12, Issue: 5, Sept. 2004, Pages: 520 – 529
7. de Haan, J.M.; Grbic, N.; Claesson, I.; Nordholm, S.E.; Filter bank design for subband adaptive microphone arrays, IEEE Transactions on Speech and Audio Processing,, Volume: 11 , Issue: 1 , Jan. 2003 , Pages:14 – 23
8. Yiu, K.F.C.; Xiaoqi Yang; Nordholm, S.; Kok Lay Teo; Near-field broadband beamformer design via multidimensional semi-infinite-linear programming techniques, IEEE Transactions on Speech and Audio Processing, Volume: 11 , Issue: 6 , Nov. 2003 , Pages:725 – 732
9. Ryan, J.G.; Goubran, R.A.; Application of near-field optimum microphone arrays to hands-free mobile telephony, IEEE Transactions on Vehicular Technology, Volume: 52 , Issue: 2 , March 2003 , Pages:390 – 400
10. Grbic, N.; Nordholm, S.; Cantoni, A.; Optimal FIR subband beamforming for speech enhancement in multipath environments, IEEE Signal Processing Letters, Volume: 10 , Issue: 11 , Nov. 2003 , Pages:335 – 338
11. Potamitis, I.; Fishler, E.; Speech activity detection of moving speaker using microphone arrays, Electronics Letters , Volume: 39 , Issue: 16 , 7 Aug. 2003 , Pages:1223 – 1225
12. Doclo, S.; Moonen, M.; GSVD-based optimal filtering for single and multimicrophone speech enhancement, IEEE Transactions on Signal Processing, Volume: 50 , Issue: 9 ,

Sept. 2002 , Pages:2230 – 2244
13. Yamada, T.; Nakamura, S.; Shikano, K.; Distant-talking speech recognition based on a 3-D Viterbi search using a microphone array, IEEE Transactions on Speech and Audio Processing, Volume: 10 , Issue: 2, Feb. 2002 , Pages:48 – 56
14. Neo, W.H.; Farhang-Boroujeny, B.; Robust microphone arrays using subband adaptive filters, IEE Proceedings- Vision, Image and Signal Processing, Volume: 149 , Issue: 1, Feb. 2002, Pages:17 – 25
15. Lo, K.W.; Ferguson, B.G.; Flight path estimation using frequency measurements from a wide aperture acoustic array,IEEE Transactions on Aerospace and Electronic Systems, Volume: 37, Issue: 2, April 2001, Pages: 685 – 694
16. Shiping Chen; Ziping Sun; Bridge, B.; Traffic monitoring using digital sound field mapping, IEEE Transactions on Vehicular Technology , Volume: 50 , Issue: 6 , Nov. 2001 , Pages:1582 – 1589
17. Lai, K.; Baker, M; Net timer: A tool for measuring bottleneck link bandwidth, Proceedings of the USENIX Symposium on Internet Technologies and Systems, March 2001.
18. Demuth, H.; Beale, M; Neural Network Toolbox, MathWorks Cooperation, 2000.
19. Carter, R. L.; Crovella, M. E; Dynamic server selection using bandwidth probing in wide-area networks, Technical Report BU-CS-96-007, Computer Science Department, Boston University, March 1996
20. Hagan, M. T., and M. Menhaj; Training feedforward networks with the Marquardt algorithm, IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989–993, 1994.
21. Murata, M.; Oie, Y.; Suda, T.; Miyahara, H; Analysis of a discrete-time single-server queue with bursty inputs for traffic control in ATM networks, IEEE Journal on Selected Areas in Communications, Volume: 8, Issue: 3, April 1990, Pages: 447 – 458
22. Cooper, C.A.; Park, K.I; Toward a broadband congestion control strategy, IEEE Network, Volume: 4, Issue: 3, May 1990 Pages: 18 – 23
23. DARPA Neural Network Study, Lexington, MA: M.I.T. Lincoln Laboratory, 1988.
24. Parker, D. B; Learning-logic: Casting the cortex of the human brain in silicon, Technical Report TR-47, Center for Computational Research in Economics and Management Science, MIT, Cambridge, MA, 1985.

Appendix: Training Convergence Time Table

Note: Data listed here meet the following 2 conditions. First, the training processes should be accomplished in 5000 epochs to reduce MSE to be less than 0.001, and second, error should be less than 6 cm. Otherwise, it will be marked 'N'. (unit: second)

Algorithm	5	10	15	20	25	30	35	40	45	50
LM	N	N	N	N	970	94	40	69	22	59
BFG	N	N	N	N	N	261	159	142	132	158
RP	N	N	N	N	N	N	N	N	N	N
SCG	N	N	N	N	N	N	N	227	387	224
CGB	N	N	N	N	N	N	N	N	N	306
CGF	N	N	N	N	N	258	258	263	380	361
CGP	N	N	N	N	N	153	262	N	337	304
OSS	N	N	N	N	N	N	N	N	N	N
GDX	N	N	N	N	N	N	N	N	N	N