Convolution Models with Shift-invariant kernel based on Matlab-GPU platform for Fast Acoustic Imaging

Ning Chu, Nicolas Gac, José Picheral, Ali Mohammad-Djafari

Iranian Society of Acoustics and Vibration
4th ISAV2014 International Conference on Acoustics and Vibration
10-11 Dec 2014 Tehran - Iran
Iran University of Science & Technology

Convolution Models with Shift-invariant kernel based on Matlab-GPU platform for Fast Acoustic Imaging☆

Ning CHU[a,c,d], Nicolas GAC[a], José PICHERAL[b], Ali MOHAMMAD-DJAFARI[a]

[a]*Laboratoire des signaux et systèmes (L2S), CNRS-SUPELEC-UNIV PARIS SUD, 91192 GIF-SUR-YVETTE, FRANCE*
[b]*Département Signaux et Systèmes Electroniques, École Supérieure d'Électricité (SUPELEC), 91192 GIF-SUR-YVETTE, FRANCE*
[c]*College of Electrical Science and Engineering, National University of Defense Technology, Changsha, Hunan, 410073, China*
[d]*Principal corresponding author:* ***chuning1983@gmail.com***

Abstract

Acoustic imaging is an advanced technique for acoustic source localization and power reconstruction from limited noisy measurements at microphone sensors. This technique not only involves in a forward model of acoustic propagation from sources to sensors, but also its numerical solution of an ill-posed inverse problem. Nowadays, the Bayesian inference methods in inverse methods have been widely investigated for robust acoustic imaging, but most of Bayesian methods are time-consuming, and one of the reasons is that the forward model causes heavy matrix multiplication. In this paper, we focus on the acceleration of the forward model by using a 2D-invariant convolution and a separable convolution respectively; For hardware acceleration, the Matlab-Graphics Processing Unit application are discussed. For method validation, we use the simulated and real data from the wind tunnel experiment in automobile industry.

Keywords: Source localization; acoustic imaging; Deconvolution; GPU;

1. INTRODUCTION

Nowadays, the acoustic imaging plays an essential role in acoustic source detections on the stationary, moving and rotating objects etc[1]. In general, it should be considered the forward model of acoustic power propagation[1], as well as the numeric solution of the inverse problem[2]. In this paper, we mainly focus on the signal processing and the inverse problems applied in acoustic imaging. The conventional Beamforming [3] method can give a direct and fast acoustic power imaging, but its spatial resolution is often very coarse at low working frequencies. In fact, the beamforming result can be interpreted as the source power image deteriorated by the 2D convolution caused by different microphone array responses which are called convolution kernels, namely Point Spread Functions (PSF). To achieve high spatial resolutions, the Deconvolution Approach for Mapping of Acoustic Source (DAMAS) method [4] can iteratively deconvolve the Beamforming result. However, conventional DAMAS suffers from slow convergence due to the spatially-variant array responses. For the fast deconvolution, extended DAMAS [5] assumes one spatially-invariant

☆Paper partly based on authors' paper published in Berlin Beamforming Conference (BeBeC) 2014, Fev.19-20, 2014, Berlin, Germany.

convolution kernel, but this assumption inevitably affects spatial resolutions, since the DAMAS is the deconvolution without regularizations. To overcome these drawbacks, Bayesian inference methods [2, 6] have been a powerful methodology for solving ill-posed inverse problem. It can adaptively estimate both unknown ran'dom variables and unknown model parameters by updating the probability law, in which, a posterior probability can be obtained from the likelihood and all prior models. And the likelihood depends on the noise prior, and it can be derived from the forward model using measured data. Moreover, the prior models can be assigned according to useful information on the unknowns, and these priors promote useful regularizations on ill-posed inverse problems. For parameter estimation, our previous work [6] proposed to use the Bayesian approach with a sparsity enforcing prior based on the Joint Maximum A Posterior (JMAP) optimization. However, the Bayesian JMAP often causes tremendous computational burden due to non-quadratic optimization and huge matrix multiplication in the forward model.

In this paper, our motivation is to propose a fast Bayesian JMAP approach of acoustic imaging on the vehicle surface, which can be widely used in wind tunnel tests for automobile industry. Our main contributions are to accelerate the forward model using invariant and separable convolutions, as well as Graphics Processing Unit (GPU) implementation.

This paper is organized as: Section 2 states the problem and presents the classical forward model of acoustic propagation. Section 3 introduces the proposed 2D invariant and separable convolution models. Section 4 and 5 validate the proposed approaches on simulations and real data respectively. Finally Section 6 concludes this paper.

2. Conventional forward model based on matrix multiplication

We assume that acoustic sources are uncorrelated monopoles [4, 5]; microphones are omni-directional with unitary gain; background noises at the microphones are Additive Gaussian White Noise (AGWN), independent and identically distributed (i.i.d); the complex reverberations in the open wind tunnel could be neglected.

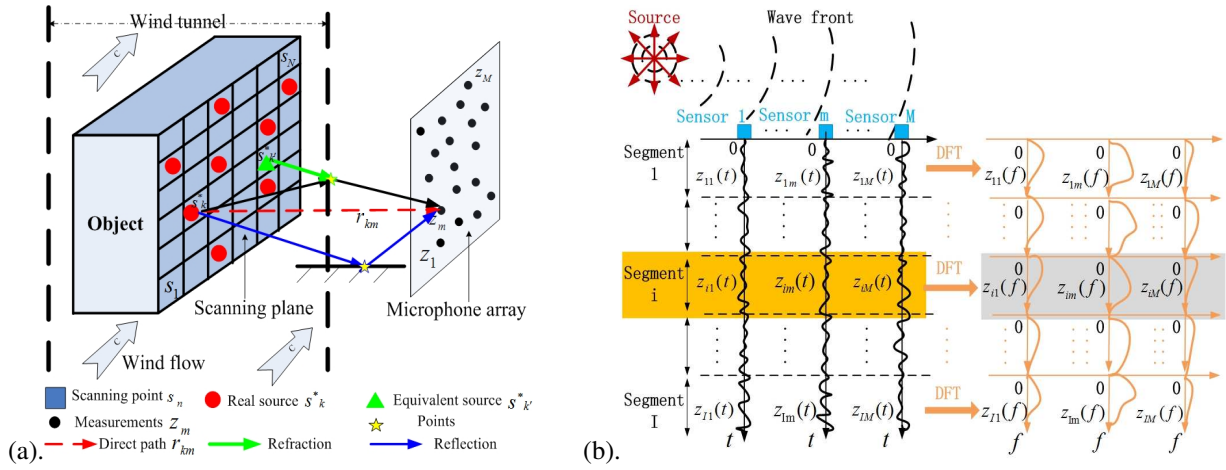

Figure 1: (a). Illustration of the acoustic signal propagation in wind tunnel[6]. (b). Illustration of the signal processing procedure in Eq.(1).

Figure 1(a) illustrates the acoustic signal propagation from the source plane to the microphone array in the wind tunnel, where microphones are installed outside the wind flow. On the source plane, we suppose K unknown original source signals $\mathbf{s}^* = [s_1^*, \cdots, s_K^*]^T$ at unknown positions $\mathbf{P}^* = [\mathbf{p}_1^*, \cdots, \mathbf{p}_K^*]^T$, where $\mathbf{p}_k^*$ denotes the 3D coordinates of kth original source signal s_k^*, notation $(\cdot)^*$ represents the original sources, and operator $(\cdot)^T$ denotes the transpose. On the microphone plane, we consider M microphones at known positions $\bar{\mathbf{P}} = [\bar{\mathbf{p}}_1, \cdots, \bar{\mathbf{p}}_M]^T$. The source plane is then equally discretized into N grids at known positions $\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_N]^T$. We assume that K original sources $\mathbf{s}^*$ sparsely distribute on these grids, satisfying $N > M > K$ and $\mathbf{P}$ including $\mathbf{P}^*$. We thus get N discrete source signals $\mathbf{s} = [s_1, \cdots, s_N]^T$ at known positions $\mathbf{P}$, satisfying $s_n = s_k^*$, for $\mathbf{p}_n = \mathbf{p}_k^*$; $s_n = 0$ others. Since K<<N, $\mathbf{s}$ is full of zero, and it becomes a sparse signal with K-sparsity in the space domain. Therefore, to reconstruct $\mathbf{s}^*$ is to reconstruct K-sparsity signal $\mathbf{s}$. And $\mathbf{p}_k^*$ can be deprived from the discrete position $\mathbf{p}_n$, where s_n is non-zero.

2.1. *Forward model of acoustic signal propagation*

Signal processing procedure is illustrated in Fig.1(b). For the mth microphone with $m \in [1, \cdots, M]$, there are T samplings of acoustic signals in time domain. Then these T temporal samplings are divided into I blocks with L samplings in each block. We note $\mathbf{z}_{i,m}(t)$ as the received signal of the ith sampling block ($i \in [1, \cdots, I]$) at the mth microphone in the sampling time $t \in [(i-1)L+1, \cdots, iL-1]$, and total sampling number is noted by $T = I \times L$. Since original source signals are usually of wide-band, we apply the Discrete Fourier Transform (DFT) in time domain to treat measured signals $\mathbf{z}_{i,m}(t)$ at each block so as to obtain L narrow frequency bins f_l (l$\in [1, \cdots, L]$). Let $\mathbf{z}_i(f_l) = [z_{i,1}(f_l), \cdots, z_{i,M}(f_l)]^T$ denote all measured signals in frequency domain. The signal processing is made independently for each frequency bin, thus in the following, we omit f_l for simplicity. Thus $\mathbf{z}_i$ can be modeled [4, 6] as

$$\mathbf{z}_i = \mathbf{A}(\mathbf{P})\,\mathbf{s}_i + \mathbf{e}_i\,, \tag{1}$$

where $\mathbf{A}(\mathbf{P}) = [\mathbf{a}(\mathbf{p}_1) \cdots \mathbf{a}(\mathbf{p}_N)]$, $\mathbf{A}(\mathbf{P}) \in \mathbb{C}^{M \times N}$ consists of N steering vectors

$$\mathbf{a}(\mathbf{p}_n) = \left\{ \frac{1}{r_{n,1}} \exp\left\{-j(2\pi f_l \tau_{n,1})\right\}, \cdots, \frac{1}{r_{n,M}} \exp\left\{-j(2\pi f_l \tau_{n,M})\right\} \right\}^T\,, \tag{2}$$

where $r_{n,m}$ denotes the distance from source n to sensor m, $\tau_{n,m}$ propagation time during $r_{n,m}$. For $r_{n,m}$, we also consider the ground reflection and wind refraction in authors' paper [6]. For simplicity, $\mathbf{a}(\mathbf{p}_n)$ is short as $\mathbf{a}_n$ afterwards.

In summary, the forward model of signal propagation in Eq.(1) is a linear but under-determined (M<N) system of equations for solving K-sparsity signal $\mathbf{s}$.

2.2. *Forward model of acoustic power propagation*

Using Beamforming [3], the signal model in Eq.(1) can be transfered into the power propagation model as:

$$\mathbf{y} = \mathbf{C}\,\mathbf{x} + \sigma_e^2\, 1_a\,, \tag{3}$$

where $\mathbf{y} = \{y_n\}_N^T$ denotes the Beamforming power vector; y_n can be interpreted as the estimated source power at grid n. And $\mathbf{y} = \tilde{\mathbf{A}}^\dagger\, \mathbb{E}[\mathbf{z}\mathbf{z}^\dagger]\, \tilde{\mathbf{A}}$ can be directly obtained from Eq.(1), where $\tilde{\mathbf{A}} = [\tilde{\mathbf{a}}(\mathbf{p}_1) \cdots \tilde{\mathbf{a}}(\mathbf{p}_N)]$, $\tilde{\mathbf{A}}(\mathbf{P}) \in \mathbb{C}^{M \times N}$ denotes the Beamforming steering matrix, and $\tilde{\mathbf{a}}(\mathbf{p}_n) = \frac{\mathbf{a}_n}{\|\mathbf{a}_n\|_2^2}$, operator $(\cdot)^\dagger$ denotes conjugate transpose, $\mathbb{E}[\cdot]$ denotes mathematical expectation. In practice, $\mathbb{E}[\mathbf{z}\mathbf{z}^\dagger] \approx \frac{1}{I} \sum_i^I \mathbf{z}_i \mathbf{z}_i^\dagger$ is approximated. $\boldsymbol{x} = diag\,\{\mathbb{E}[\boldsymbol{s}\boldsymbol{s}^H]\}$ denotes the unknown source power vector, and $diag\{\cdot\}$ denotes diagonal items; thus $\mathbf{x}$ is a signal as K-sparsity as $\mathbf{s}$. And σ_e^2 denotes the variance of i.i.d AGWN noises $\boldsymbol{e}$. Notation $1_a = [\frac{1}{\|\mathbf{a}_1\|^2}, \cdots, \frac{1}{\|\mathbf{a}_N\|^2}]^T$ represents the noise attenuation for different grids. $\mathbf{C} = \{c_{i,j}\}_{N \times N}$ denotes the power propagation matrix, defined as:

$$c_{i,j} = \frac{\|\mathbf{a}_i^H \mathbf{a}_j\|_2^2}{\|\mathbf{a}_i\|_2^2} = \left| \frac{1}{\sum_{m=1}^M \frac{1}{r_{im}^2}} \sum_{m=1}^M \frac{1}{r_{im}\, r_{jm}}\, e^{-j\frac{2\pi f_l}{c_0}(r_{jm} - r_{im})} \right|^2\,, \tag{4}$$

where $\mathbf{a}_i$ is defined in Eq.(1); $r_{i\,m}$ denotes the propagation distance from ith discrete source (at the position $\mathbf{p}_i$ on the discrete source plane) to the mth microphone; f_l denotes the lth frequency bin; M is the total number of microphones. According to Eq.(4), it yields $0 \le c_{i,j} \le 1$ and $c_{i,i} = 1$. In fact, $c_{i,j}$ can represent the power contribution (%) of the microphone array from the jth source to the ith position on the source plane. So that $c_{i,j}$ can also be seen as the Point Spread Function (PSF) of the microphone array. This PSF is determined by two factors: the microphone array topology and the distance from the source plane. In ideal case, $c_{i,j} = \delta_{i,j}$ becomes the Dirac function, and it derives $\mathbf{y} = \mathbf{x} + \sigma_e^2\, 1_a$ from Eq.(3), which is easy to solve.

In short, compared with signal propagation model of Eq.(1), the power propagation model of Eq.(3) is a linear and determined system of equations for solving K-sparsity source powers $\mathbf{x}$. Furthermore, as recently stated in our paper [7], the power propagation matrix $\mathbf{C}$ obtained from Eq.(4) can be approximated into a Symmetric Toeplitz Block Toeplitz (STBT) matrix in the far-field condition. Therefore, we can derive an invariant convolution model as:

$$[\mathbf{C}\,\mathbf{x}]_i = [\mathbf{h} * \mathbf{x}_0]_{p,q}\,, \quad i = p + (q-1)\,N_r\,, \tag{5}$$

where we define the source power image $\mathbf{x}_0 = [x_{p,q}]_{N_r \times N_c}$ with $p \in [1, \cdots, N_r]$, $q \in [1, \cdots, N_c]$, where N_r and N_c denote row and column number respectively, provided $N_r \leq N_c$ for a rectangular image. And $\mathbf{x}_0$ can be vectorized to $\mathbf{x} = [x_j]_N$ in column-first order as: $x_j = x_{p,q}\,, \quad j = p + (q-1)\,N_r$. The index $[\cdot]_i$ represents the ith item of a vector; index $[\cdot]_{p,q}$ represent the pth row, qth column item of a matrix. Finally, the invariant convolution kernel $\mathbf{h} = [h_{k,l}]$ with $k, l \in [1, \cdots, N_r]$ can be derived as:

$$\begin{cases} h_{k,l} = \frac{1}{M^2} \left| \sum_{m=1}^{M} e^{j\frac{2\pi f_l}{c_0}(r_{i,m} - r_{j,m})} \right|^2 , \\ i = \lfloor \frac{N+1}{2} \rfloor, \quad j = i + (\lfloor \frac{N_r+1}{2} \rfloor - k)\, N_r + \lfloor \frac{N_r+1}{2} \rfloor - l \end{cases} , \tag{6}$$

where $\mathbf{h}$ is can be a $N_r \times N_r$ square matrix, this size is not absolutely accurate but very reasonable, because the STBT matrix consists of $N_r \times N_r$ subblocks, and we will discuss different kernel size on simulation part; operator $\lfloor \cdot \rfloor$ denotes integer part. The 'invariant' kernel in Eq.(6) does not change along with convolution output $[\cdot]_i$, but remains the same $i = \lfloor \frac{N+1}{2} \rfloor$. The kernel derivation can be seen in our previous work [7].

3. Proposed separable convolution model

In order to accelerate the 2D convolution, we here investigate the separability of the convolution kernel. Let $r(\mathbf{h})$ denote the rank of a $N_r \times N_r$ convolution kernel $\mathbf{h}$. And $\mathbf{h}_1$, $\mathbf{h}_2$ denote two column vectors with the same length of N_r. If $r(\mathbf{h}) = 1$, we can get $\mathbf{h} = \mathbf{h}_1 * \mathbf{h}_2^T$ [8]. The advantages of separable deconvolution are, for an input vector with the length N, the computational complexity of 2D convolution (using $\mathbf{h}$) is $\mathrm{O}(N_r^2\, N)$, while the separable convolutions using $\mathbf{h}_1$, $\mathbf{h}_2$ can be greatly reduced into $\mathrm{O}(2N_r\, N)$. Meanwhile, the storage of convolution kernels is also reduced from N_r^2 to $2N_r$. Even if $r(\mathbf{h}) \neq 1$, we still want to derive $\mathbf{h}_1$ and $\mathbf{h}_2$. Since every real symmetric matrix $\mathbf{h}$ can be diagonalized, we take the EigenValue Decomposition (EVD) of of $\mathbf{h}$ as:

$$\mathbf{h} = \mathbf{U}\Lambda\mathbf{U}^T = \sum_{i=1}^{N_r} \lambda_i \mathbf{u}_i \mathbf{u}_i^T \,, \tag{7}$$

where $\mathbf{U} = [\mathbf{u}_i]_{N_r}$ is a $N_r \times N_r$ orthogonal matrix, whose columns $\mathbf{u}_i$ with $i \in [1, \cdots, N_r]$ are eigenvectors of $\mathbf{h}$; and $\Lambda = Diag[\lambda_i]_{N_r}$ is a real and diagonal matrix, with λ_i being the eigenvalues of $\mathbf{h}$, supposing $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_{N_r} \geq 0$. From Eq.(7), we can approximate $\mathbf{h}$ by using the biggest eigenvalue λ_1 and its corresponding eigenvector $\mathbf{u}_1$ as $\mathbf{h} \approx \lambda_1 \mathbf{u}_1 \mathbf{u}_1^T$. Then we define the approximating error as:

$$\varepsilon = \frac{\|\mathbf{h} - \lambda_1 \mathbf{u}_1 * \mathbf{u}_1'^T\|_2^2}{\sum_{i=1}^{N_r} \lambda_i^2} \times 100\% = \frac{\| \sum_{i=2}^{N_r} \lambda_i \mathbf{u}_i \mathbf{u}_i^T \|_2^2}{\sum_{i=1}^{N_r} \lambda_i^2} \times 100\% \,, \tag{8}$$

where the valid convolution satisfies $\mathbf{u}_1 * \mathbf{u}_1'^T = \mathbf{u}_1\, \mathbf{u}_1^T$, and $\mathbf{u}_1' = [u_{1,N_r}, \cdots, u_{1,1}]$ denotes symmetric vector of $\mathbf{u}_1 = [u_{1,1}, \cdots, u_{1,N_r}]$. Since $\mathbf{U} = [\mathbf{u}_i]_{N_r}$ is an orthogonal matrix, we then have $\mathbf{u}_i\, \mathbf{u}_j^T = \mathbf{I}$ and $|\mathbf{u}_i^T\, \mathbf{u}_j| = 1$ for $i = j$, $|\mathbf{u}_i^T\, \mathbf{u}_j| = 0$ for $i \neq j$. According to Eq.(7), $\| \sum_{i=2}^{N_r} \lambda_i \mathbf{u}_i \mathbf{u}_i^T \|_2^2 \leq \sum_{i=2}^{N_r} \lambda_i^2 \|\mathbf{u}_i\, \mathbf{u}_i^T\|_2^2 = \sum_{i=2}^{N_r} \lambda_i^2$. Finally, according to Eq.(8), the upper bound of approximating error is:

$$\varepsilon \leq \frac{\sum_{i=2}^{N_r} \lambda_i^2}{\lambda_1^2 + \sum_{i=2}^{N_r} \lambda_i^2} \times 100\% = \frac{1}{\rho + 1} \times 100\% \,, \tag{9}$$

where $\rho = \frac{\lambda_1^2}{\sum_{i=2}^{N_r} \lambda_i^2}$ denotes the separability degree. The bigger ρ is, the more separable $\mathbf{h}$ becomes. If $r(\mathbf{h}) = 1$, then $\mathbf{h} = \lambda_1 \mathbf{u}_1 * \mathbf{u}_1^T$, so that $\varepsilon = 0$ in Eq.(9) and $\rho \to \infty$. Finally, we can obtain the separable convolution as

$$\mathbf{h} \approx \lambda_1 \mathbf{u}_1 * \mathbf{u}_1'^T \,. \tag{10}$$

In conclusion, if the 2D convolution kernel $\mathbf{h}$ is a non-negative, real and symmetric matrix, it can be approximated by a separable convolution kernel which consists of the biggest eigenvalue and its corresponding eigenvector of $\mathbf{h}$. And the approximating error is relatively small, as long as the kernel separability is big enough.

4. Simulations for convolution model validation

The simulation configurations are based on the wind tunnel experiment as shown in Fig.4: there are $M = 64$ microphones locating on the vertical plane. $d = 2$m is the averaged size of microphone array. $D = 4.50$m is the distance between the microphone plane and source plane. Original source powers $\mathbf{x}^*$ are within [-10.3,3.7]dB and 14dB dynamic range. The i.i.d AWGN noise power is set $\sigma_e^2 = 0.86$ (-0.7dB), thus the averaged SNR is 0dB.

4.1. Performance of 2D invariant and separable convolutions

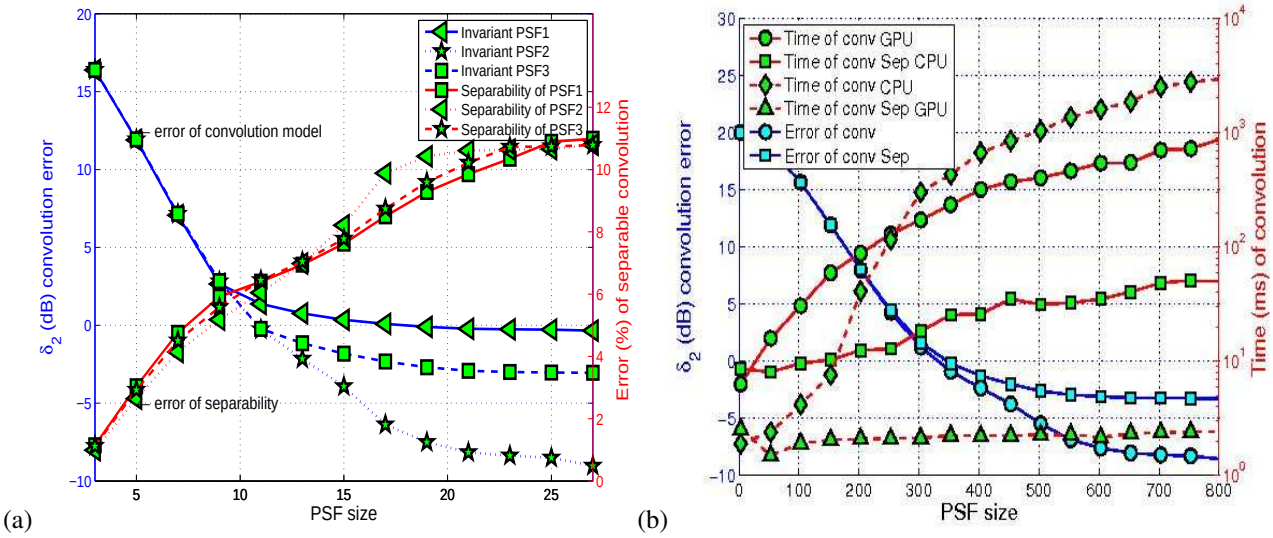

Figure 2: (a).Performance of separable convolution VS 2D invariant convolution (b).Computation performance comparisons at 2500Hz among invariant and separable convolution using CPU (3.33Hz clock) and GPU Tesla C1060, Parallel Computing Toolbox of MATLAB 2012b).

In Fig.2(a), the image size is $N_c = 27$ and $N_r = 17$, and the propagation matrix $\mathbf{C} = [c_{i,j}]_{N\times N}$ with $N = N_c \times N_r = 459$, so that we test the PSF sizes from 3 to 27. The PSF1 is obtained by the items of $c_{1,j}$ on the first line of $\mathbf{C}$; the PSF2 is from the middle line ($i = 230$) of $\mathbf{C}$; And the PSF3 is from the line of $i = 300$. For three blue curves and left Y-axis in Fig.2(a), we show the convolution approximating errors $\delta_y = \frac{\|\mathbf{y}-\hat{\mathbf{y}}\|_2^2}{\|\mathbf{y}\|_2^2} \times 100\%$ between the proposed convolutions and simulated Beamforming results. It is seen that the larger the kernel size is, the bigger the convolution approximating error becomes. But PSF2 can achieve the smallest convolution approximating errors using a relatively small kernel size (15×15). While for three red curves and right Y-axis in Fig.2(a), we show the kernel separabilities of 2D invariant kernel with respect to different PSF sizes. All of three separabilities share the same trend and are close to each other. The larger the size is, the bigger error of separability is. But separability error remains relatively small ($< 11\%$). Particularly, when kernel size is about 15×15 which is close to $N_r = 17$, the separability error is just around 5%. This is because the symmetric structures of PSFs can well meet the separable conditions in Eq.(9).

In conclusion, figure 2(a) tells that PSF2 can be a separable convolution which contributes an efficient and effective convolution to approximate the forward model of source power propagation in Eq.(3).

4.2. Convolution computational time based on CPU and GPU

In Fig.2(b), we show computation comparisons among invariant and separable convolutions using CPU and GPU respectively. Here, the size of the source power image is enlarged as the 30 times as that in Fig.2(a). Compared with CPU, one of the greatest advantages of GPU is the great number of parallel computational cores which contribute much more powerful computation capability than CPU. However, the massive data with non-parallel processing can hardly be efficient performed by GPU [9]. The structures of CPU and GPU are shown in Fig.3. For the two blue curves and left Y-axis in Fig.2(b), the convolution errors of the invariant kernel and separable kernel keep relatively small, especially when the kernel size is near to the half size of the source power image. For the four red curves and

right Y-axis, the computing time in Fig.2(b), all the red curves go up along with the large kernel size, but the separable convolution based on CPU or GPU keeps a slight increase and maintains the least computation burden, especially when the kernel size is large enough. It is seen that GPU greatly increases the computation speed.

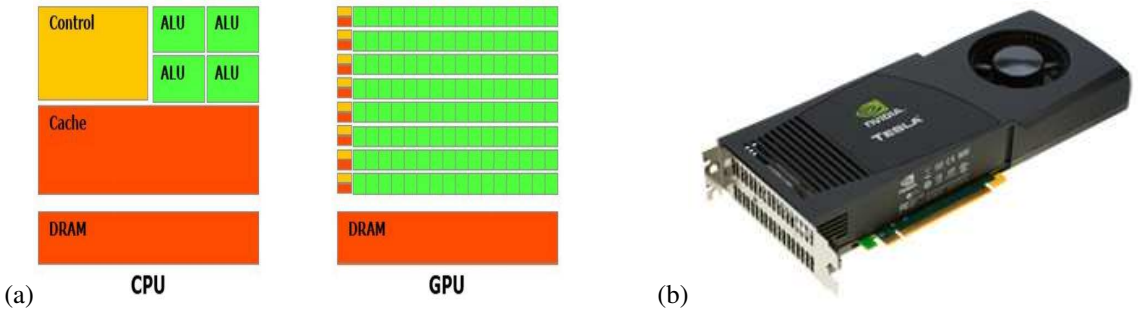

Figure 3: (a) Structures of CPU and GPU [9] (b) Tesla C1060 GPU: 240 processing cores, 1.3G Hz clock, 622 GFLOPs (Peak).

4.3. Deconvolution performance for convolution models

Table 1: Computational time of Tikhonov deconvolution via separable convolution kernels based on GPU Tesla C1060; Using Parallel Computing Toolbox of MATLAB 2012b. Time results are averaged by 20000 iterations. Reconstrution error $\delta_x = \frac{\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2}{\|\mathbf{x}\|_2^2} \times 100\%$

Power image size	17×27	255×405	527×837
Invariant kernel size	13×13	215×215	415×415
Time (ms)/iteration	1.65	7.50	38.3
Deconvolution error δ_x (%)	19.6	29.6	30.6
Speed gain on $\mathbf{C}\,\mathbf{x}$ (CPU)	0.5	314.7	822.2
Speed gain on $\mathbf{h} * \mathbf{x}$ (CPU)	0.27	7.1	9.2
Speed gain on $\mathbf{h}_1 * \mathbf{h}_2^T * \mathbf{x}$ (CPU)	0.16	5.27	7.02

In Table 1, we show the deconvolution performance for convolution models. In order to make a fair comparison, the conventional Tikhonov regularization method is used for deconvolution, since the most time-consuming operation of deconvolution mainly depends on the convolution operations. In order to show the GPU acceleration with respect to CPU, we firstly use the CPU to implement the deconvolution methods using matrix multiplication $\mathbf{C}\,\mathbf{x}$, invariant convolution $\mathbf{h} * \mathbf{x}$ and separable convolution $\mathbf{h}_1 * \mathbf{h}_2^T * \mathbf{x}$ respectively. Then the separable convolution operation $\mathbf{h}_1 * \mathbf{h}_2^T * \mathbf{x}$ is efficiently computed on GPU Tesla C1060 using the Parallel Computing Toolbox of MATLAB 2012b.

From Table 1 that for the computational time, the bigger kernel size is, the greater the computational speed gain is obtained. For the deconvolution errors, the Tikhonov regularization method does not offer very good results, but it still shows that the deconvolution for the separable convolution can be efficiently solved based on GPU. In order to improve deconvolution performance, we will use Bayesian JMAP method proposed in our paper [6] on the real data. One interesting thing is that for the small image size of 17×27, the GPU could not improve computational efficiency compared with CPU. This is because that the 240 cores of GPU Tesla C1060 can efficiently handle the large dimension of matrices, but small matrix cannot be well suited to the parallel structure of GPU. Convolution operation realized by MATLAB Parallel Toolbox could not completely use advantages of the GPU, since it still requires other operations on CPU and causes frequently data transfers between the GPU and CPU. Therefore, it is a promising work to implement the whole deconvolution algorithm (not only the convolution operation) completely based on GPU using the CUDA code library [10]. Moreover, we find out that calculating invariant convolution $\mathbf{h} * \mathbf{x}$ based on GPU merely makes use of about 14% of computational power of GPU, while separable convolution $\mathbf{h}_1 * \mathbf{h}_2^T * \mathbf{x}$ just occupies nearly 7%. So that there will be great potential to develop our own parallel separable convolution algorithm based on the GPU so as to make good use of GPU powerful peak computational capacity.

5. Wind tunnel experiments

Figure 4: Acoustic imaging on the vehicle surface in Wind tunnel S2A.

Figure 4 shows the configurations of the wind tunnel S2A [11]. Wind tunnel experiments are designed to reconstruct the positions and acoustic powers on the traveling car surface. The source plane of car side is of 1.5× 5 m^2 (31×101 pixels). On the real data, there are T=524288 samplings with the sampling frequency f_s=2.56×10^4 Hz. We separate these samplings into I=204 blocks with L=2560 samplings in each bloc. The working frequency is 2500Hz which is sensitive to human being. The image results are shown by normalized dB images with 10dB span. For the actual propagation distance $r_{n,m}$ in Eq.(1), we apply equivalent source to make refraction correction, and the mirror source signal to correct the ground reflection as discussed in author's paper[6].

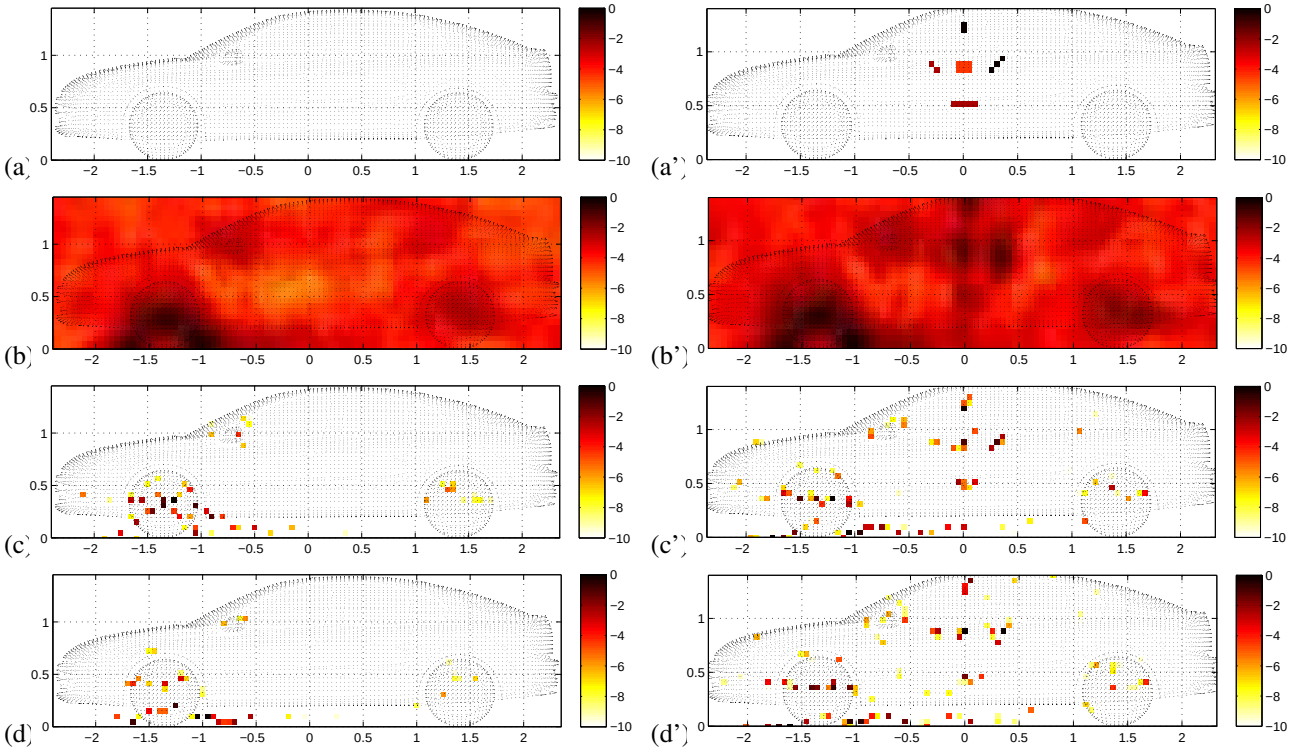

Figure 5: Left: real data at 2500Hz (a) vehicle surface (b) Beamforming (c) Bayesian JMAP via classical forward model (d) JMAP via invariant convolution model. Right: hybrid data (a') 5 simulated complex sources (b')-(d') corresponding methods.

Figure.5(a-d) illustrate the estimated power images of mentioned methods. As discussed before, the Beamforming

just gives a coarse image of strong sources. The Bayesian JMAP method via sparse prior [6] not only distinguishes the strong sources around the two wheels, rearview mirror and side window, but also successfully reconstructs the week ones on the front cover and light. The Bayesian JMAP method via proposed 2D invariant convolution models can not only achieve the acoustic localization as good as the JMAP via conventional forward model, but also successfully source power reconstruction too. Moreover, we investigate the hybrid data to further validate our proposed convolution models. In Fig.5(a'-d') the hybrid data are generated in such a way that the simulated sources are added to the real data on the positions where there are no obvious real sources locating. We suppose that if our approach can correctly detect the simulated sources, it is reasonable to effectively reconstruct the original sources on real data. As we can see, the Bayesian JMAP inference method via convolution model can successfully detect both the simulated and original source powers in the real data. In Table 2, the computing time based on the 2D invariant convolution in Eq.(6) is much less than conventional model in Eq.(3)

Table 2: Computational cost for treating real data of whole car: image 31×101 pixels, at 2500Hz, based on CPU: 3.33Hz. 'JMAP+Conv' is short for Bayesian JMAP method via 2D invariant convolution model

Methods	CB	JMAP+$\mathbf{C}\,\mathbf{x}$	JMAP+$\mathbf{H}\,\mathbf{x}$
Time (s)	1	1012	180

6. Conclusions and perspectives

We modified the forward model of source power propagation so as to accelerate the Bayesian JMAP inference applied in acoustic imaging on the car surface. We intensively discuss the forward model approximation by using 2D variant, invariant and separable convolution kernels. By various simulations, it is shown that all three convolution models can greatly reduce the calculating burden and get relatively small reconstruction errors, especially for the separable convolution. Moreover, the real data from wind tunnel S2A France shows that our proposed approximating models can successfully reconstruct the acoustic sources on the car surfaces, and cause much less consuming time than the conventional forward. Using the convolution model, the Bayesian JMAP inference method can we effectively implemented on the GPU platform. However, it is still a promising work to implement the deconvolution algorithms fully based on GPU mainly using the CUDA code library [10].

Acknowledgment

Great thank to Renault SAS for offering real data, and Mr. Xiangyang GAN's contributions during his internship.

References

[1] J. Lanslots, F. Deblauwe, K. Janssens, Selecting Sound Source Localization Techniques for Industrial Applications, Sound and Vibration 44 (6) (2010) 6–10.
[2] J. Antoni, A Bayesian approach to sound source reconstruction: optimal basis, regularization, and focusing, The Journal of the Acoustical Society of America 131 (2012) 2873–2890.
[3] J. Chen, K. Yao, R. Hudson, Source localization and beamforming, Signal Processing Magazine, IEEE 19 (2) (2002) 30–39.
[4] T. Brooks, W. Humphreys, A Deconvolution Approach for the Mapping of Acoustic Sources (DAMAS) determined from phased microphone arrays, Journal of Sound and Vibration 294 (4-5) (2006) 856–879.
[5] R. Dougherty, Extensions of DAMAS and Benefits and Limitations of Deconvolution in Beamforming, in: 11th AIAA/CEAS Aeroacoustics Conference, Monterey, CA, USA, 23-25 May, 2005, pp. 1–13.
[6] N. Chu, A. Mohammad-Djafari, J. Picheral, Robust Bayesian super-resolution approach via sparsity enforcing a priori for near-field aeroacoustic source imaging, Journal of Sound and Vibration 332 (18) (2013) 4369–4389.
[7] N. Chu, N. Gac, A. Mohammad-Djafari, J. Picheral, 2d convolution model using (in)variant kernels for fast acoustic imaging, in: Berlin Beamforming Conference 2014 (BeBeC2014), Berlin, Germany, Feb.19-20,2014, p. No.5.
[8] P. Perona, Deformable kernels for early vision, Pattern Analysis and Machine Intelligence, IEEE Transactions on 17 (5) (1995) 488–499.
[9] R. Fernando, GPU gems: programming techniques, tips, and tricks for real-time graphics, 2004.
[10] J. Nickolls, I. Buck, M. Garland, K. Skadron, Scalable parallel programming with cuda, Queue 6 (2) (2008) 40–53.
[11] A. Menoret, N. Gorilliot, J.-L. Adam, Acoustic imaging in wind tunnel S2A, in: 10th Acoustics conference (ACOUSTICS2010), Lyon, France, 2010.