AN EFFICIENT IMAGE RECONSTRUCTION ALGORITHM FOR A MULTIPLE HYDROPHONE ARRAY SYNTHETIC APERTURE SONAR

P.T. Gough, M.P. Hayes and D.R. Wilkinson

Department of Electrical and Electronic Engineering, University of Canterbury, Private Bag 4800, Christchurch, New Zealand.
email: gough@elec.canterbury.ac.nz

Single hydrophone synthetic aperture sonars (SAS) are velocity limited since the maximum distance the sonar platform can move between pings is restricted to less than half the extent of the largest transducer. By using a linear array of hydrophones combined with a single transmitting projector, the platform velocity can be increased, allowing a larger area to be covered within the same time constraints. Although it is possible to preprocess the multiple hydrophone data into the equivalent single hydrophone data, the underlying efficiencies of the image reconstruction process is often lost in the inefficiencies of the preprocessing stage.

This paper shows how data from a multiple-hydrophone SAS may be combined efficiently for use with single-hydrophone image reconstruction procedures like the wavenumber algorithm. Example code in Matlab shows the importance of program efficiency in applying this algorithm.

1 INTRODUCTION

In using side-looking sonars for sea-floor imagery, the limitation of standard sonars is that the along-track (the azimuth) resolution degrades with range. To produce reasonable imagery, either the transmitted frequency is increased (a 500kHz centre frequency is not uncommon) or the extent of the transducers is increased; both changes intended to produce a fan or pencil beam in the along-track dimension. This still does not produce a constant azimuth resolution with range but it does minimise the beam spreading and so gives reasonable imagery. However, there is another process that results in constant along-track resolution at all ranges without resorting to high frequencies or large transducers. This process is the sonic equivalent of Synthetic Aperture Radar (SAR) so it is not surprisingly named Synthetic Aperture Sonar (SAS).

With SAR/SAS, the imagery is a two-step procedure where the raw echo data is stored and must be extensively processed before the eventual image is computed. Basically instead of radiating a pencil beam, the SAS transmits a broad beam so that each reflecting object in the field of view is insonified over several pings and the echoes recorded in both amplitude and

phase. A collection of the echoes from several pings are suitably delayed and summed to form a single pixel in the final processed image.

This delay and sum procedure (often called the beam-forming algorithm) is slow -extremely slow- as it needs a different delay and sum calculation for each pixel in the image. A better way to proceed is to take a block of data and compute a block of pixels in the image. These more efficient block-processing algorithms are the range-Doppler, the wavenumber and the chirp-scaling algorithms. However they all assume there is a single projector and a single receiver. For SAR this is no restriction as it is the normal mode of operation. However SAS is different.

The difference comes from the slow speed of sound in water. For SAR/SAS to operate without along-track (azimuth) ambiguities caused by spatial undersampling along the aperture, the platform can move no further than one half of the linear extent of the largest transducer between transmitted pings. If the pulse repetition period is set by the maximum unambiguous range, the platform is restricted to quite low velocities. The common approach to the low speed/ aperture undersampling sampling problem is to provide a sonar platform with one projector and a linear array of hydrophones. By assuming the phase centre of a specific combination of projector and hydrophone bisects a line between them, a hypothetical single projector/single hydrophone set of data can be stitched up. The stitch-up is relatively straightforward if the phase centre of the rearward-most hydrophone is exactly a unit spacing from that of the forward-most hydrophone of the previous ping. However that means the platform can move at only one fixed forward velocity. Any other velocity needs a more complicated stitch-up and without careful processing, the inefficiencies of the stitch up can overwhelm the efficiencies of the block SAS algorithm. Here we have developed an efficient preprocess to combine any number of hydrophones with the platform travelling at any velocity.

2 COMBINING DATA FROM A MULTIPLE HYDROPHONE ARRAY SAS

Given there are H hydrophones in the linear array we define hydrophone zero, $h = 0$, as the most forward hydrophone from the projector. Each of the remaining hydrophones are numbered successively through to the last hydrophone at $h = H - 1$. The relative location of each hydrophone h is positioned at distance d_h from the projector, and the phase centre bisects the two at $d_h / 2$. Typically the hydrophones are spaced uniformly along the array, however this is not necessary as the combining preprocess is based on the phase centres.

2.1 “Continuous” Data

Assume the raw echoes, $ee_h(t,u)$, are recorded for each hydrophone, h, as a function of delay time, t, and along-track projector position, u, for each transmitted pulse (ping). The first step is to calculate the along-track spectral history for each hydrophone by taking a 1-D Fourier transform in time denoted by $F_t\{\}$. So for all $0 \le h \le H - 1$ we find,

$$Ee_h(\omega,u) = F_t\{ee_h(t,u)\} \tag{1}$$

Pulse compression is usually now applied by multiplication with the complex conjugate of the transmitted signal's power spectrum $P^*(\omega)$, unless use of the chirp scaling process is intended. So for all $0 \le h \le H - 1$,

$$Ss_h(\omega,u) = P^*(\omega)\cdot Ee_h(\omega,u) \tag{2}$$

If the path trajectory is known, the equivalent timing errors (sway and yaw) can be corrected by the multiplication of $Ss_h(\omega, u)$ with an appropriate phase function. The H arrays are then 1-D Fourier transformed in the along-track ordinate u into the mixed coordinate temporal frequency/wavenumber domain (ω, k_u). So for all $0 \le h \le H - 1$,

$$SS_h(\omega,k_u) = F_u\{Ss_h(\omega,u)\} \tag{3}$$

In cases where the navigation data is unknown, it is slightly more efficient to use a 2-D Fourier transform and apply the pulse compression in the following way,

$$SS_h(\omega,k_u) = P^*(\omega)\cdot F_{t,u}\{ee_h(t,u)\} \tag{4}$$

but either procedure ends up with H spectral domains $SS_h(\omega, k_u)$; one for each hydrophone. At this stage the H spectral domains can be combined into one composite domain by multiplying $SS_h(\omega, k_u)$ by an exponential phase function and summing over each spectral domain.

$$SS(\omega,k_u) = \sum_{h=0}^{H-1} SS_h(\omega,k_u)\cdot exp\left(-jk_u\frac{d_h}{2}\right) \tag{5}$$

for $-\frac{k_{us}}{2} \le k_u < \frac{k_{us}}{2}$ and the maximum spatial frequency is $k_{us} = \frac{2\pi H}{dU}$. The distance between pings, dU, is given by the along-track velocity, v, multiplied by the ping repetition period, T_r. $SS(\omega, k_u)$ may then be used as either a substitute for the first stage of the standard synthetic aperture wavenumber or the range-Doppler process for azimuth compression [1-2].

2.2 Sampled Data

Assume the raw echo data, now in time and along-track sampled form, is described by $ee_h[m,p]$ where the number of hydrophones is H, the number of pings transmitted P, and the number of time samples for each ping M. The phase centre, positioning and numbering of the hydrophones is the same as for the continuous case.

The first step is to 1-D Fourier transform the H separate arrays of raw data to compute the H along-track spectral histories for each hydrophone.

$$Ee_h[n,p] = \sum_{m=0}^{M-1} ee_h[m,p]\cdot exp\left(\frac{-j2\pi mn}{M}\right) \tag{6}$$

where the ping index is $0 \le p \le P - 1$, the time sample index is $0 \le m \le M - 1$ and the spectral index is $0 \le n \le M - 1$.

The spectral history is followed by pulse compression requiring vector/array multiplication by the conjugate of the power spectra of the transmitted signal. At this stage correction of sway and yaw errors may be performed. The next step is to compute H separate 1-D Fourier transforms over the along track index p,

$$SS_h[n,q] = \sum_{p=0}^{P-1} P^*[n]\cdot Ee_h[n,p]\cdot exp\left(\frac{-j2\pi pq}{P}\right) \tag{7}$$

where the spatial wavenumber index is $0 \le q \le P-1$.

The H spectral domains can now be combined to generate the equivalent composite domain. For $0 \le r \le P \cdot H - 1$

$$SS[n,r] = \sum_{h=0}^{H-1} SS_h[n, r \bmod P] \cdot exp\left(-j\left(\frac{r}{PH} - \frac{1}{2} \right) k_{us} \frac{d_h}{2} \right) \tag{8}$$

and as before this is used as a substitute for the equivalent *SS[n, r]* for a single hydrophone SAS.

3 IMPLEMENTATION IN MATLAB

A fast simulation has been developed in Matlab to perform the described image reconstruction algorithm[1]. As with most algorithms, the efficiency and speed of the algorithm is only realised if the program code fully uses the capabilities of the software it is run on. Since Matlab is designed for use with matrices, implementing the algorithm on a sample-by-sample basis (as the algorithm suggests) proves to be very inefficient. A significantly faster simulation results by performing the same operation as a matrix operation once over all the samples. Through efficient coding we show the dramatic decrease in execution time possible when the multiple hydrophone data is combined to the equivalent single hydrophone data.

3.1 Parameters for a Multiple Hydrophone Array SAS

To store the multiple hydrophone data in Matlab, a 3-D array is used. By allowing the third dimension to represents each hydrophone then we have a 2-D array to record the pings and time samples for that hydrophone. By convention, each ping is recorded in a successive column, while the time samples for that ping are recorded in successive rows.

The parameters used in the example code are shown in Table 1. The initial multiple hydrophone data is stored in a 3-D array called *ss_multi* and is of size $2048 \times 512 \times 4$.

Parameter	*Value*	*Unit*	*Definition*
NU	512	-	Number of pings
Nt	2048	-	Number of time samples for each ping
H	4	-	Number of hydrophones
d	[0.0 0.0614 0.1229 0.1843]	m	Relative location of hydrophones to transmitter
dU	0.1229	m	Spacing between pings
Nu	NU × H = 2048	-	Effective number of pings
du	dU/H = 0.0307	m	Effective ping spacing
kus	2π/du = 204.5	rad/m	Maximum spatial frequency of ku

Table 1: Multiple Hydrophone Array SAS Parameters

[1] The complete Matlab code for performing the image reconstruction algorithm for a multiple hydrophone SAS may be found at the web site http://www.elec.canterbury.ac.nz/research/sonar/multi_hydro.html

3.2 Combining the Multiple Hydrophone Array SAS Data

The first part of the algorithm is to combine the multiple hydrophone data to that of the equivalent data obtained from a single hydrophone. A straightforward implementation is to perform this on a sample-by-sample basis as suggested in the sample code of Fig. 1. Although this code produces the desired result it takes three and a half hours to execute.

```
1   SS = zeros(Nt, Nu);
2   for h = 1:H
3       SSm = fftshift_vert(fft2(fftshift(ss_multi(:, :, h))))*dU;
4       for r = 1:Nu
5           for n = 1:Nt
6               SS(n, r) = SS(n, r) + SSm(n, mod((r-1), NU)+1)*exp(-j*((r-1)/Nu-0.5)*kus*d(h)/2);
7           end
8       end
9   end
```

Fig. 1: Inefficient multiple hydrophone preprocess using sample by sample

Profiling this code immediately highlights the problem area. Not surprisingly the majority of the time is taken executing the innermost 'for' loop (line 6). This line of code executes 16.8 million times and consequently a slow function call such as the 'mod' function, which takes on average 0.6ms for each function call, results in a very slow overall execution time. A substantial amount of time is also taken up in the overhead necessary to set up the 'for' loops. The most obvious improvement is to try and eliminate the two innermost 'for' loops and perform the necessary operations as matrix operations rather than sample-by-sample operations. Dividing line 6 up into three parts we can examine the necessary changes to remove the inner 'for' loops.

1. Operate on the whole array, *SS*, at once instead of sample by sample, *SS(n,r)*.
2. Use of the 'mod' function to determine the appropriate sample from the array, *SSm*, may be eliminated. In matrix form this is the spectral domain, for hydrophone *h*, repeated for however many hydrophones *H* there are. So for our example, the 2048×512 array, *SSm*, is repeated four times to get one large 2048×2048 array for each hydrophone. In Matlab this repeat operation may be performed very quickly using the 'repmat' feature[2].
3. Generate a matrix containing the exponential part, which can have a very quick element-by-element multiply performed on it. The exponential part is not dependent on the time sample but only on the effective ping. A 2-D matrix can be created, by creating a row vector array that is repeated downwards using the 'repmat' function.

Fig. 2 shows the efficient code with the two inner 'for' loops removed.

Profiling the new code shows a vast speed improvement. Instead of taking over three and a half hours the equivalent efficient code takes a mere 15 seconds. The huge time gain is achieved by taking advantage of Matlab's matrix ability. The use of the 'repmat' function, which takes only 0.6s to compute, is called only eight times (twice for each of the four hydrophones) compared with the 16.8 million 'mod' and 'exp' function calls.

[2] For example the portion of code *repmat(SSm,1,H)* replicates and tiles the matrix *SSm* to produce a 1-by-H block matrix. In our case it is a 1×4 repetitions of the matrix array *SSm*.

```
1   SS = zeros(Nt,Nu);
2   for h = 1:H
3       SSm = fftshift_vert(fft2(fftshift(ss_multi(:, :, h))))*dU;
4       SS = SS + repmat(SSm,1,H).*repmat(exp(-j*[-Nu/2:Nu/2-1]/Nu*kus*d(h)/2),Nt,1);
5   end
```

Fig. 2: Efficient multiple hydrophone preprocess using block matrices

When the effective single hydrophone data is multiplied by the phase function it assumes that the DC component is in the centre of the spectrum. Since the original Fourier transformed multiple hydrophone data is replicated for as many hydrophones *H* as there are, we cannot solely use the *fftshift* function to get the DC component centred. A custom function, *fftshift_vert*, was created that instead of swapping opposite quadrants it swaps the upper half of the matrix with the lower half of the matrix. By performing the *fftshift_vert* before the replication of the multiple hydrophone data we result in the DC component being centred as illustrated in Fig. 3.

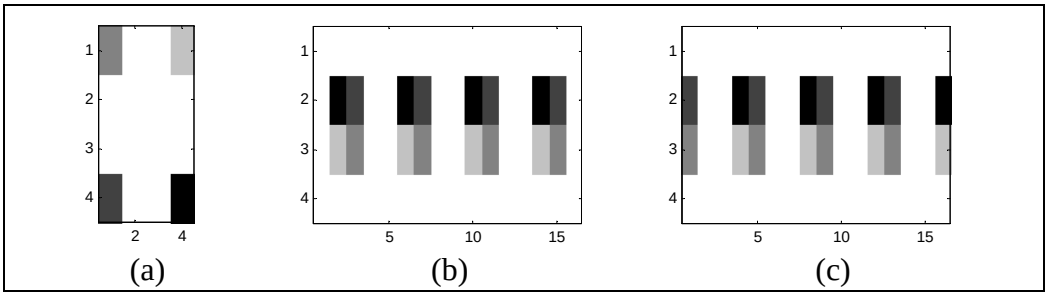

Fig. 3: (a) sample matrix, (b) repeat of matrix after fftshift (c) repeat of matrix after fftshift_vert

CONCLUSION

Because of the slow spread of acoustic propagation in water, SAS systems will always need a linear hydrophone array to achieve any reasonable mapping rate. This means that the standard single hydrophone block processing, image reconstruction procedures such as the wavenumber algorithm, cannot be applied directly and the array data must be preprocessed directly into a single hydrophone equivalent. This preprocessing step needs to be as efficient as possible to preserve the efficiency gains of the basic SAS algorithms. This can be done by including the preprocessing step as part of the overall SA reconstruction algorithm.

REFERENCE

[1] **Peter T Gough**, **David W. Hawkins**, Unified Framework for Modern Synthetic Aperture Imaging Algorithms, *International Journal of Imaging Systems and Technology*, volume (8), pp. 343-358, 1997.

[2] **David W. Hawkins**, **Peter T Gough** Efficient SAS image reconstruction algorithms, *Journal Acoustics Society of America*, volume (101), no 5, part 2, pp. 3120, 1997.