A Partially Adaptive STAP Algorithm Approach to fMRI

L. Huang[1], E. A. Thompson[2], T. M. Talavage[1]

[1]School of Electrical & Computer Engineering, Purdue University, West Lafayette, IN, United States, [2]Department of Engineering, Purdue University Fort Wayne, Fort Wayne, IN, United States

Introduction

Space-time adaptive processing (STAP), originally developed in the field of sensor array processing, has already been shown to exhibit potential for detecting cortical activations in functional magnetic resonance imaging (fMRI) (1). Previous work has focused on the fully adaptive version of STAP, which is very computationally intensive. In this study, a partially adaptive version of STAP—element-space STAP—is introduced to reduce dimensionality of the problem.

Methods

Partially adaptive STAP schemes transform a large set of input signals to a relatively small set of signals prior to assigning filter weights (2). Element-space partially adaptive STAP retains the spatial dimensionality of fully adaptive STAP but reduces the number of temporal degrees of freedom prior to adaptation. It does so by combining the time course data of the 3D data sets into several subsets prior to assigning weights. Thus, given a full M_xxM_yxN data set, where M_x and M_y are the number of sensors in the –x and –y directions, respectively, and N is the number of frames, a subset is defined comprising K successive frames. As a result, the full data set is transformed into N'=N-K+1 subsets, each consisting of the full M_xxM_y spatial dimensionality but reduced to K frames. These subsets utilize overlapping sets of frames. The reduction of the data set is accomplished via $\overrightarrow{\chi_p} = \left(J_p \otimes I_M\right)^H \vec{\chi}$, where $\otimes$ is the kronecker product, H denotes hermitian transpose, $\vec{\chi}$ is the original full dimensional data set as defined in (1), I_M is an identity matrix of size M_xxM_y, and J_p is a selection matrix defined as $J_p = [0_{p\times K}; I_K; 0_{(N-K-p)\times K}]$ which chooses a particular subset of frames. The notation 0_{pxK} refers to a p x K matrix of zeros. For each subset p, a reduced weight vector is calculated as $\overrightarrow{w_p} = R_{up}^{-1}\overrightarrow{v_p}$, here $R_{up} = E\{\overrightarrow{\chi_p}\,\overrightarrow{\chi_p}^H\}$ and $\overrightarrow{v_p} = J_p^H \vec{v}$ are reduced versions of the correlation matrix and spatial steering vector defined in (1). The individual subset weights are then combined into a single weight vector using the relation $\vec{w} = \sum_{p=0}^{N'-1} \left(J_p \overrightarrow{w_p}\right)^H \cdot \vec{v} \cdot \left(J_p \overrightarrow{w_p}\right)$(3), thus restoring full dimensionality to the filtering process. The filter output, $z = \vec{w}^H \vec{\chi}$, is subsequently obtained as using this weight vector and the original activated data signal of full dimensionality.

For purposes of comparison with the fully adaptive counterpart, performance of partially adaptive STAP was evaluated using the same data sets as in (1). Thus, specifics of imaging are the same as in that work. Performance comparison was based on amount of CPU running time required for each algorithm as well as the accuracy of each in locating the single activated pixel.

Results

For STAP, filter output values were linearly scaled to 256 levels and the threshold was given as a percentage of maximum filter output. Table 1 displays the number of true positives/false positives for four different thresholds for each of 15 trials. For the partially adaptive trials, a K value of 1 was used. Table 1 also displays the CPU running time for each algorithm. In 13 of the 15 trials, accuracy in locating the single activation was identical for fully adaptive and partially adaptive STAP. Accuracy of the two algorithms differed in only two trials (14 is better & 15 worse for partially adaptive STAP). In all 15 trials, partially adaptive STAP showed a significant improvement in CPU running time.

Figure 1 (a) displays CPU running time and (b) identifying threshold as a function of K, the number of frames in the overlapping subsets. The identifying threshold is defined as the largest threshold that ensures no false positive, and a smaller value is better. In 22 simulations utilizing the same noise environment, figure 1 illustrates that values of K less than 5 or between 295 and 300 produce relatively low identifying thresholds and CPU running times.

Conclusions

Results indicate that with selection of a suitable value of K, partially adaptive STAP can attain performance levels near those of fully adaptive STAP while decreasing the CPU running time by nearly half. Furthermore, partially adaptive STAP decreases memory requirements, enabling analysis of larger data sets.

Acknowledgement

This work was supported by NIH grant 1R21MH68267-01A1.

References

1. Thompson E, Holland S, Schmithorst V. A STAP Algorithm Approach to fMRI: A Simulation Study. J Magn Reson Imaging 2004;20:715-722.
2. Ward J. Space-time adaptive processing for airborne radar. Technical report 1015, Lincoln Laboratory, MIT, Lexington, MA. Dec. 13, 1994.
3. Yaron Seliktar, et al. Evaluation of Partially Adaptive Algorithms on the Mountain Top Data set. IEEE ICASSP 1996; vol.2:1169-1172

Trial	Algorithm	True/False at different Threshold				CPU* Running Time(s)
		60%	70%	80%	90%	
1-12	Fully	1/0	1/0	1/0	1/0	5557**
	Partially					2829**
13	Fully	1/2	1/0	1/0	1/0	5824
	Partially					2933
14	Fully	1/4	1/2	1/0	1/0	5461
	Partially	1/2	1/0	1/0	1/0	2806
15	Fully	0/1	0/1	0/0	0/0	5801
	Partially	0/1	0/1	0/1	0/1	2718

*CPU type: sparcv9+vis; CPU speed: 400MHz; Main Memory: 22 GB; OS: Sun5.9
* *Average CPU Running time of 12 trials

Table 1 True Positives/False Positives and CPU Running Time

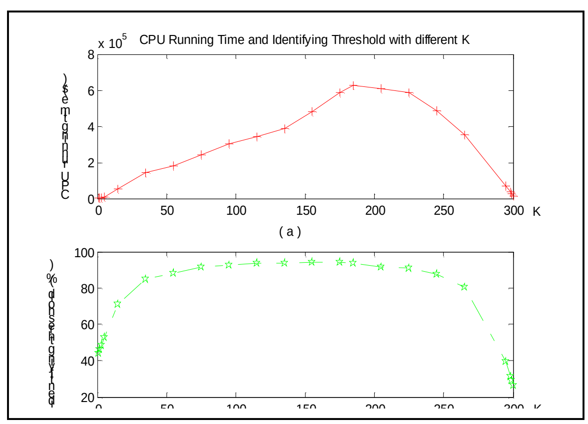

Figure 1