IJASRM International Journal of Advanced Scientific Research and Management, Volume 3 Issue 5, May 2018
www.ijasrm.com
ISSN 2455-6378

Compressive Sensing for Medical Imaging based on Greedy Algorithms

Arun Kumar giri[1], Kuldeep Yadav[2] and R.P Agarwal[1]

[1] Shobhit University, Meerut,
Uttar Pradesh, India

[2] College of Engineering Roorkee, Roorkee,
Uttarakhand, India

Abstract

Compressive sensing is a recent sampling method used to replace Nquist-Shannon Sampling technique. It has been used in various data engineering applications such as speedy reconstruction of medical scanned images based on data size to attain a high resolution image with high quality. MRI scan images were used for validation of compressive sensing algorithms. The result demonstrates a increase in processor speed and decreases in the reconstruction process time. The best resolution image has been achieved using Compressive Sampling based on Greedy algorithms Matching Pursuit (CoSaMP) on images. The lowest image recovery time.

1. Introduction

In recent years, compressive sensing is mainly used for image reconstruction and retrieval from minimal set of data. The process of reconstruction is an inverse technique in which the signal or image can be retrieved from the available minimal data set. The various methods in reconstruction of images using Compressive Sensing are examined and verified. Compressive Sensing is a significantly effective process due to its high demand for faster, efficient and economic image processing applications, but this technique needs huge storage space, computing and execution time. CS provides sampling and image compression simultaneously and evaluates the minimal set of samples that carries maximal information set. CS is the newest method which minimizes the need for acquiring and preserving the huge sample size during its implementation. CS extracts the samples from original image or signal those have significance and have minimum significance. Due to this characteristics of sparsity and incoherence, CS finds its vital role in several applications such as

COMPRESSIVE SENSING FRAMEWORK

The concept of CS is first introduced by D. Donoho, E.J Candes and T. Tao with the application in Seismology during 1970 and later Santosa and Symes suggested norm minimization techniques.

Sampling Theorem

The well known as Sampling Theorem, the Nyquist-Shannon Theorem was introduced by Shannon in 1949. The theorem defines that for any band-limited signal and of time varying can be recovered and retrieved accurately if the sampling frequency is equally or greater twice of the maximum frequency present in the signal itself. In widely used processes, before storing of the data or transmission of the signal, it must be sampled rightly at Nyquist rate before compression or modulation. The Fig. 1(a) depicts the traditional way of sampling; in which x is the signal of subject and y is the measurement of compressed vector. In the model of traditional sampling, the acquisition of data and sampling of data are performed distinctly. This method has the captured sample size that remains always greater than the information carrying samples. For the high-dimensional signals, this technique involves with complex computing, particularly, because of the demand of large memory. Analog to Digital conversion for high dimensional signals is too expensive by this approach.

Fig. 1(b) depicts the process of compressive sampling. More information is contained in the compressive structure; so there is no requirement of performing computation to the rest of the trivial samples. So Compressive Sensing can be regarded as alternative method for Nquist sampling, which does sampling and compression simultaneously.

IJASRM International Journal of Advanced Scientific Research and Management, Volume 3 Issue 5, May 2018
www.ijasrm.com
ISSN 2455-6378

Compressive Sensing

Compressive sensing on the property of the signals capable of providing the sparse domain representation with lesser numbers of nonzero coefficients. Such property is termed as sparsity of the signal. The recovery algorithm utilized along with CS determines sample size required for precise reconstruction.

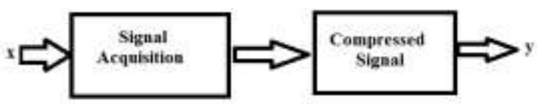


Fig. 1(a) Block diagram of Nquist Sampling Process

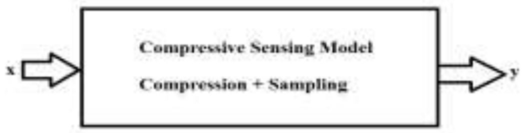


Fig. 1(b) Block diagram of Compressive sampling process

The model of reconstruction using CS depends on properties:

Model for Compressive Sensing

Simultaneous compression and sampling simultaneously is the basic operation of compressive sensing. The sparse signal for an N -dimensional original signal can be created by depicting it with any appropriate basis like DCT, Fourier Transform, and wavelet Transform. The sparse structure for the given signal can be used.

ORTHOGONAL MATCHING PURSUIT

Iterative methods used by greedy algorithms are potential in reconstructing the images. Compressive Sensing of block based favors the utilization of Greedy algorithms for reconstructing thermal images too. It operates in iteration so to reconstruct the sparse signal. Orthogonal matching pursuit (OMP), is a type of greedy algorithm, fundamentally comes as an evolved form of MP (Matching Pursuit) algorithm. Previously, the MP, an iterative greedy technique, was inducted in approximation the solution by decomposition. It recognizes involved basis and their coefficients which constructs the input signal on compounding. It begins with the presumption that entire basis are orthogonal among themselves. The values are evaluated by correlation of the original signal with the basis provides the impacts of the basis on the signal. When the basis is crucial part of the signal, the value of correlation will be high and the basis when has a trivial role, the value of correlation will be low. It operates by choosing the elements which has maximum correlation in its residual vector on iteration throughout the algorithm.

The steps involved in OMP algorithm are shown as follows:

OMP Algorithm

Initialize the index set $\Lambda_K = \Phi$ and the residue $r_0=v$ and set the counter as t=1

Find the coordinate which has its inner-product highest

$\Phi_k = argmax\ x_{1<i<N} |\langle r_{k-1} \emptyset_i \rangle|$

Pick the chosen index set and accrue by the matrix of selected atoms

$\Lambda_t = \Lambda_{t-1}\ U\ \{ \Lambda_t \}$

Calculate the least square challenge such that finding the fresh estimate of the sparse signal

$x_t = argmin\ \|y - \emptyset_t x\|_2$

Calculate the fresh update of approximation of the residue

$$a_t = \emptyset_t x_t$$
$$r_t = y - a_t$$

Increase the counter and go to step 2 if t < K

The value of the evaluated estimate provides the reconstructed signal.

Regularized Orthogonal Matching Pursuit (ROMP)

Regularized Orthogonal Matching Pursuit is also alike OMP which utilizes many vectors on iteration to construct the solution. The ROMP will utilizes the vectors which possess identical dot product with the needed vector. It utilizes entire vectors which possess dot product size half size above of largest dot product. The merit of the vectors are utilized so to have same contribution in obtaining the required vector. The process is recursively used by updating residue vector on iteration similar like in OMP.

ROMP is modification of OMP algorithm with the regularized step. The ROMP has capability to reconstruct the sparse signal using random measurements with appreciably minimized time duration. ROMP was proposed by Deanna Needell and Roman Vershynin in recovery of sparse with the merit of quicker execution. The ROMP is so precise in recovery of any kind of sparse using entire measurements matrices within the property of RI conditions. It begins sparsification of original signal of dimension N using measurement matrix Φ. The measurement vector is then obtained using $y=\Phi x$ which is utilized in recovering of the signal. The algorithm chooses the largest valued coefficients of S from the inner-product instead finding the highest. Hence by reckoning compound indices at a time and later reckoning those possessing energy greater than a specified threshold, complete execution period can be minimized appreciably.

ROMP Algorithm with mathematical notation is as follows

ROMP Algorithm
Input: Matrix A, vector b and sparsity level n **Output:** Approximation vector x or index set V Initialize and begin by setting the residual $\mathbf{r_0} = \boldsymbol{x}$, the time $t = 0$ and index set V = Φ Find and choose a set J of the n biggest coordinates in the vector $\mathbf{u} = \Phi^T \mathbf{r_t}$ (or all of its nonzero coordinates if this set is smaller) Regularize and find entire subsets J0 ⊂ J the maximum$\|u\|J_0\|_2$. Where J_0 is define $i, j \in J_0$ if $\|u_i\| \leq \|u_j\|$ and $u_i \in u|J_0$ if $k \in J_0$ Update: The index set by $\boldsymbol{V} = \boldsymbol{V} \cup J_0$ and residual by x= $\min_{c\in R^V} \|\boldsymbol{x} - \Phi\mathbf{c}\|_2$ $\mathbf{r}_{t+1=} \boldsymbol{x} - \Phi\mathbf{x}$ Check for stopping condition. If the condition is not met go to step 2.

Compressive Sampling Matching Pursuit (CoSaMP)

Compressive Sampling Matching Pursuit (CoSaMP) like ROMP is dependent on OMP and too assumes in a number of vectors to construct the approximation on iteration. CoSaMP chooses a predefined number of vectors. The vectors which generate the largest dot product. CoSaMP thus limits the approximation to the desired sparsity level by eliminating entire however with desired number of entries. CoSaMP is recent algorithm built using features of OMP and has the very interesting properties:

a. Acquires samples from a diverse sampling plans.
b. Prospers using a minimal sample size.
c. Robust even on samples are impaired with noise.
d. Supplies optimal error on every target signal.
e. Provides demonstrable effective use of resource.

A set positive integers and **x** a vector. Then

$$x|_V = \begin{cases} x_i & i \in V \\ 0 & \text{otherwise} \end{cases}$$

The support of a vector **x** notation *supp*(x) is

$$supp(x) = \{xi \neq 0\}$$

CUDA Programming

GPU Accelerated Greedy Algorithms applied to Compressed Sensing uses five greedy algorithms: Thresholding, Iterative Hard Thresholding, Normalized Iterative Hard Thresholding, Hard Thresholding Pursuit and CoSaMP. The software, coded using CUDA-C, is compiled as Matlab executable files which described three functions to run as typical Matlab functions. Being source code is CUDA-C; a developer can immediately change the root functions to produce C executables instead Matlab executables. The functionality is proficient to assume rightly from Matlab and applying on GPU to generate the result.

CoSaMP Algorithm
Input: Matrix Φ, vector x and sparsity level n **Output:** Approximation vector a Begin by defining present sample v = b, the iteration count t = 0, and initial approximation $a_0 = 0$ Fix $t = t + 1$, form a duplicate vector y = Φ^Tv and find the largest elements J = *supp* (y_{2n}) Fix V = J ∪ *supp*(a_t) and construct a new vector x described as $x|_v = \Phi^+|_{T_V}$ $x|_{v^c} = 0$ Update the approximation $a_{t+1} = x_n$ and samples vector v = x - Φ a Check for halting condition. If the condition is not met then go to step 2.

PERFORMANCE AND RESULTS

Test Environment.

The test is performed on two different high performance devices; one is CPU and other one GPU. Both devices uses same OS platform. The first device is GPU of NVIDIA GeForce GT730 ,codename GF108 with GFLOP 268.8/33.6 and 2GB memory and The second device is Intel® core i3 2100 CPU with 4GB RAM size. All programs were compiled in Matlab and CUDA Windows7 OS. The specifications of both devices are show in Table 1.

Table 1:

Property	NVIDIA GeForce GT730	Intel® core i3 2100 CPU
Graphic Bus Technology	PCI Express 2.0	NA
Memory(MB)	2048	4096
Core Clock	902MHz	3.10GHz
CUDA Core/Core	16	2
Stream Processor	384	NA
Memory Bandwidth	14.4GB	21GB

Test Data

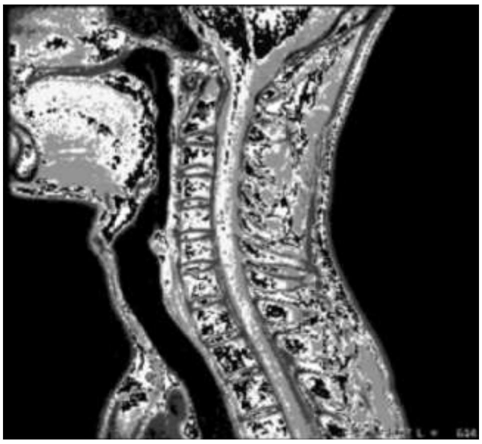
Various sparsity levels K and the acceptable measurement number M are depicted in Table 2 which are used in the experiment.

Table 2:

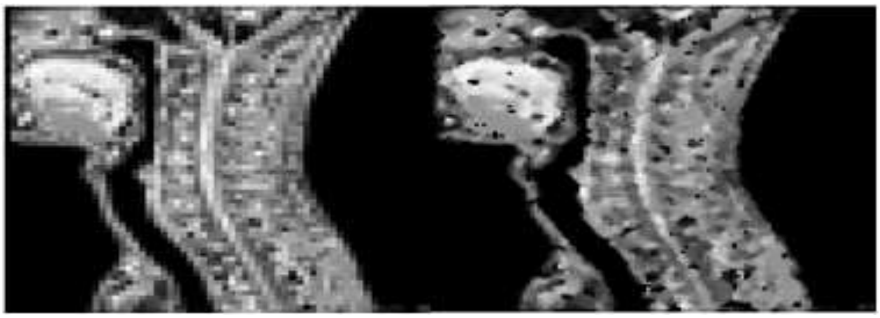
N(Size of the signal)	M(Measurement Size)	K (Sparsity level)
1024	256	64
2048	384	128
4096	896	256

The performance of ROMP and COSAMP realization using CUDA for GPU is assessed against that of CPU realization in measurement of running time.

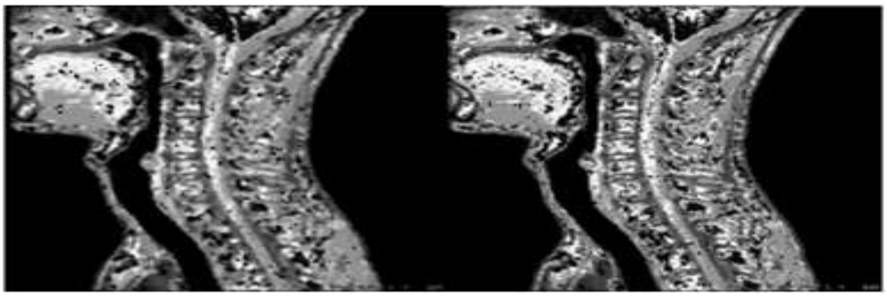
Results



(a)



(b) (c)



(d) (e)

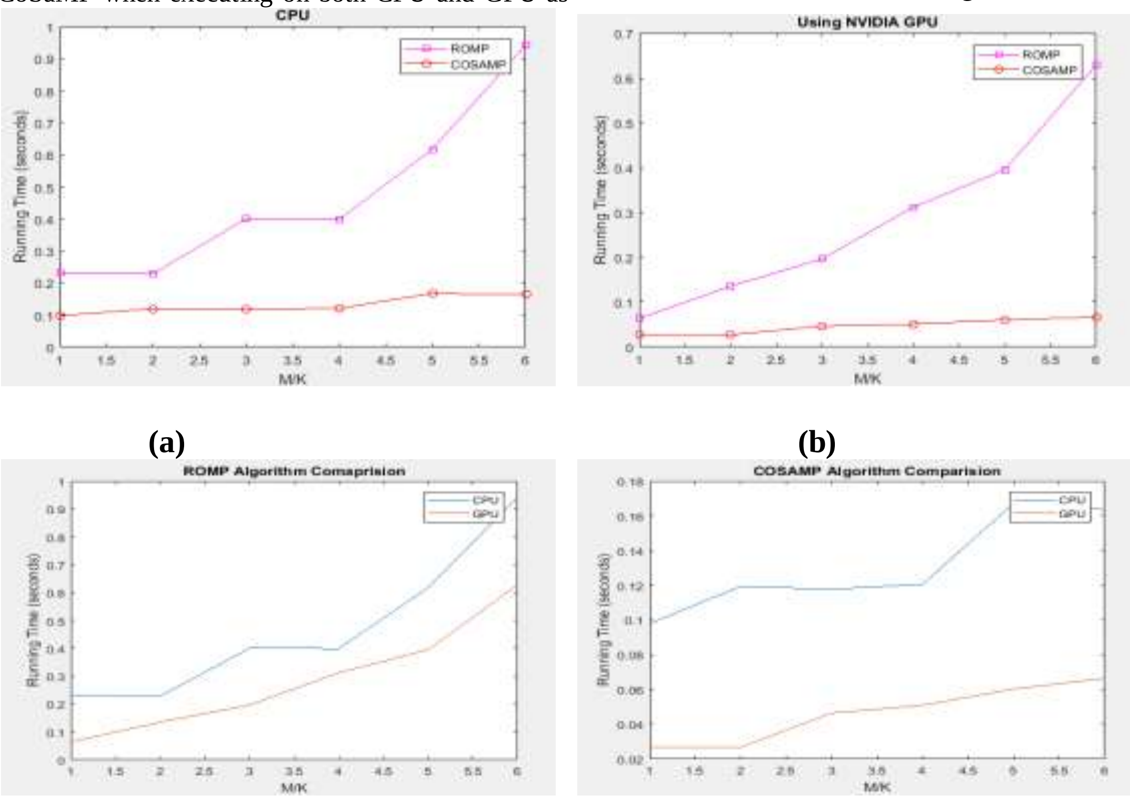
Fig 2. (a) Original Image-Cervical spine MRI Image (b) Implementation of ROMP Algorithm using CPU with M/K=3 (c) Implementation of ROMP Algorithm using GPU with M/K=3 (d) Implementation of CoSaMP Algorithm using CPU with M/K=3 (e) Implementation of COSAMP Algorithm using GPU with M/K=3

International Journal of Advanced Scientific Research and Management, Volume 3 Issue 5, May 2018
www.ijasrm.com
ISSN 2455-6378

Results and Discussion

Comparing ROMP and CoSaMP algorithms, recovery of image is much more robust using CoSaMP when executing on both CPU and GPU as observed in Fig. 2. Further, the image is recovered better with greater M/K.

Fig 3 depicts the experiment about the performance of ROMP and CoSaMP algorithms. It can be



(a) **(b)** **(c)** **(d)**

Fig. 3 (a) Running Time comparison of ROMP and CoSaMP Algorithms applied on Medical Image with size N=4096 on CPU (b) Running Time comparison of ROMP and COSAMP Algorithms applied on Medical Image with size N=4096 on GPU (c) Running Time comparison of CPU and GPU when ROMP Algorithm applied on Medical Image with size N=4096 (d) Running Time comparison of CPU and GPU when COSAMP Algorithm applied on Medical Image with size N=4096

deduced from Fig 3 (a) and (b) that CoSaMP consumes less time than ROMP in recovery of image accurately in both cases running on CPU and GPU. Further it has been observed Fig.3 (c) and (d) that GPU consumes less time than CPU for both ROMP and COSAMP Algorithms. By observing the slope of the curves of CPU and GPU of Fig. 3 (c) and (d), when M/K increases, the slope of CPU is larger than the slope of GPU, it reflects that Measurement size does affect the performance of CPU wherein GPU, it does not make huge difference

Conclusion

In this paper, compressive sensing dependent recovery algorithms for imaging are implemented. Out of several recovery algorithms, ROMP and CoSaMP have been used. Compressive Sensing based recovery algorithms utilizes fewer data samples to provide speedier computing using less memory space. ROMP and CoSaMP algorithm were realized on high performance engines for CPU and GPU. From the results it can be concluded that the speed the recovery can be enhanced, the speed in recovery of image is attained using parallel computing by choosing suitable machines. It is specifically speedier when the image size is large. It is possible to realize algorithms on GPU using CUDA.

References

[1] S. Qaisar, R. Muhammad Bilal, W. Iqbal, M. Naureen, and S. Lee, "Compressive Sensing: From Theory to Applications, a survey", *Journal of Communications and networks*, vol. 15, pp.443-456, 2013.

[2] D. Donoho, "Compressed Sensing, *IEEE Trans. on Information Theory*", Vol. 52, pp. 1289-1306, 2006.

[3] E. J. Candes, and T. Tao, "Near optimal Signal recovery from random projections: universal encoding strategies, *IEEE Trans. on Information Theory*", Vol. 52, pp. 5406-5425, 2006.

[4] E. J. Candes, and T.Tao, "Decoding by linear programming", *IEEE Trans. Inf. Theory*, vol. 51, pp. 1074-1077, 2005.

[5] U. V. Dias, and E. R. Milind "Block Based Compressive Sensed Thermal Image reconstruction using Greedy Algorithms", *I. J. Image, Graphics and Signal Processing*, vol. 6, pp. 36-42, 2014.

[6] J. A. Tropp, and A.C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit", *IEEE Trans. Inform. Theory*, Vol. 53, pp. 4655–4666, 2007.

[7] J. Zhao, Ronfang Song, Jie Zhao and Wei-Ping Zhu, "New conditions for Uniformly recovering sparse signals via orthogonal matching pursuit", *Journal of Signal Processing*, vol. 106, pp. 106-113, 2015.

[8] T. Tony andTony and Lie Wang, "Orthogonal Matching Pursuit for sparse signal recovery with noise", *IEEE Transactions on information theory*, vol. 57, pp. 4680-4688, 2011.

[9] D. Needell and R. Vershynin, "Signal Recovery from incomplete and inaccurate measurements via Regularized Orthogonal Matching Pursuit", *IEEE Journal of selected topics in signal processing*, vol. 4, pp. 310-316, 2010.

[10] J. D. Blanchard and J. Tanner," GPU accelerated greedy algorithms for compressed sensing" Math. Prog. Comp., vol.5, pp.267-304, 2013.