



Distributed Resolution Enhancement Techniques for Remotely Sensed Image

Thesis submitted to the Department of Computer Science
Faculty of Computer and Information Sciences
Ain Shams University

In partial fulfillment of the requirements for the degree
of Doctor of Philosophy (Ph.D.) in Computer and Information Sciences

By

Marwa Sayed Moustafa

National Authority for Remote Sensing and Space Science

Under the supervision of

Prof. Dr. Mohamed F. Tolba

Professor at Scientific Computing Department
Faculty of Computer and Information Sciences, Ain Shams University

Prof. Dr. Taymoor M. Nazmy

Professor at Computer science Department
Faculty of Computer and Information Sciences, Ain Shams University

Ass. Prof. Dr. Ashraf K. Helmy

Associate Professor at Receive and Analysis Data Department
National Authority for Remote sensing and space science

Ass. Prof. Dr. Hala M. Ebeid

Associate Professor at Scientific Computing Department
Faculty of Computer and Information Sciences, Ain Shams University

Acknowledgments

Praise is to ALLAH, who guided and aided me to bring-forth to light this work and by whose grace this work has been completed.

I would like to express my deep gratitude to **Prof. Dr. Mohamed F. Tolba**, Professor of Scientific Computing, Faculty of Computer and Information Sciences, Ain Shams University, for his dedicated advisor throughout my PhD studies. He provided constant guidance to my academic work. I would like to express my deep gratitude for his valuable advice and revising all items of the work.

I would also like to express my gratitude to **Prof. Dr. Tymoor Mohammad Nazmy**, Professor of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University, who supported me in my academic studies. My gratitude goes also to **Ass. Prof. Dr. Ashraf K. Helmy**, Associate Professor, National Authority for Remote Sensing, who accompanied and supported me in my work and with whom I have closely collaborated.

My greatest debt is to **Ass. Prof. Dr. Hala M. Ebeid**, Associate Professor of Scientific Computing, Faculty of Computer and Information Sciences, Ain Shams University, for supervising this work, providing all valuable suggestions, comments, scientific material and revising all items of the work.

I would like to express my gratitude to the National Authority for Remote Sensing and Space Sciences (NARSS) for providing all valuable scientific facilities and remote sensing data required for this work. I would like to express all deep thanks to my colleagues and friends at NARSS.

Finally, the warmest thanks are to my family and my sisters for their love, faith, continuous support and prayers throughout the whole period of work.

Abstract

Resolution is an important attribute for different modern military and civil applications. High-resolution images facilitate image analysis tasks performed by machines, such as scene recognition and classification. However, in practice, high-resolution images are not always readily available for different reasons, such as the deficiency of the Camera optics and lenses, Motion between the camera sensor and the scene or subject, and atmosphere.

The problem of super-resolution has already been extensively researched. Different super-resolution algorithms have been developed. In this thesis, a comprehensive investigation of super-resolution methods is conducted. There are different categories for super-resolution methods, learning-based method offers more accurate reconstruction results compared to other traditional methods. In this thesis, three different super-resolution methods are proposed.

In the first super-resolution method, support vector regression is used to learn the relation between the low-resolution -HR patches using a predefined training set. In the reconstructing, the model obtained from the training step is used to recover accurate HR patches. Specifically, the input image is first decomposed into patches. Then, for each input patch, the SVR optimal model is used to recover the high-frequency information from the low-resolution input patch. To overcome the extensive computation in the proposed method, a transition from serial implementation to an optimized parallel on GPU architecture had been proposed.

Different experiments were conducted on the syntactic and multispectral dataset. Compared to the conventional super-resolution method, the proposed method reconstructs high-resolution images with higher accuracy. The speedup achieves approximately about 10-55 times faster than their optimized serial counterparts according to the image size.

The second SR method is based on manifold learning. Acceleration for a learning based SR method using locally linear embedding (LLE) algorithm is presented. The histogram of the gradient (HoG) is adapted to generate a training set. Three main kernels were implemented to improve the speed up of the performance. The bottleneck of the LLE was the matrix manipulation. CUBLAS library was adopted for matrix inverse and the calculation of matrix-vector multiplication modules; the shared memory was utilized in the matrix-addition and subtraction kernels to enhance the performance. Different experiments are conducted using the syntactic and multi-spectral dataset to show the effectiveness of the proposed algorithm. Manifold learning SR method outperforms significantly and achieves up to $11\times$ to $163\times$ speedup compared with the state-of-art method (bicubic interpolation).

Both SVR and manifold learning are example-based methods that show superior results for upscaling factor 2, but the image quality is degraded dramatically for larger upscaling factor.

To overcome these problems, compressed sensing framework had been adopted in the third method. The K-SVD algorithm had been utilized to formulate a joint dictionary, by exploiting high-frequency information from the external image database, where the local and nonlocal. In the reconstruction, acceleration for the compressive sampling matching pursuit (CoSaMP) algorithm is proposed of HR image. This method generates a sharper high-resolution image with fewer artifacts compared with state of the art super-resolution approach. The consistency and stability of the proposed method were experimentally tested using hyperspectral dataset. Experimental results demonstrate the speedup of the proposed GPU implementation to approximately 70x times faster than the corresponding optimized CPU.

Finally, the compressed sensing paradigm was extended to exploit self-similarities existing between image patches within a single image. There is no external training set is required. Different experiments have been carried out on a multispectral

dataset. Extensive experimental were conducted to validate the effectiveness of the proposed approach. The GPU implementation accelerates the speedups compared to the CPU sequential implementation from 20× for small images to more than 40× for large image size.

Table of Contents

	Page
Acknowledgement	
Abstract	I
Table of Contents	IV
List of Figures	VII
List of Tables	XI
Abbreviations	XIII
1- Introduction	1
1.1 Preliminaries.....	3
1.1.1 Image Resolution.....	3
1.1.2 Observation Model.....	4
1.2 Super-resolution image.....	5
1.3 Motivation.....	7
1.4 Objective and Contribution of the Work.....	7
1.5 Thesis Organization.....	8
2- Super-Resolution Techniques	11
2.1 Inverse Problem	12
2.1.1 Definition of an Inverse Problem.....	12
2.1.2 Well-Posed and Ill-Posed Problems.....	13
2.1.3 Super-resolution as an Ill-Posed Inverse Problem.....	13
2.2 Taxonomy of SR Algorithms.....	14
2.3 Frequency Domain.....	16
2.3.1 Fourier Transform.....	17
2.3.2 Wavelet Transform.....	18
2.4 Spatial Domain.....	19
2.4.1 Multiple Image-based SR Algorithms.....	19
2.4.2 Single Image-based SR Algorithms	25
2.5 Super-Resolution using Sparse Representation.....	32
2.5.1 Dictionary Design	34
2.6 Summary.....	40
3- Acceleration of Super-Resolution using High Performance Computing	41
3.1 Introduction.....	42
3.2 Parallel Hardware.....	44
3.3 Trends in High Performance Computing.....	45

3.3.1	Cluster Computing.....	46
3.3.2	Grid Computing.....	47
3.3.3	Cloud Computing.....	48
3.4	General-Purpose GPU Computing.....	49
3.4.1	The Fermi Architecture.....	50
3.4.2	Compute Unified Device Architecture (CUDA)	52
3.4.3	Parallel Computing with CUDA.....	53
3.5	Performance Evaluation	55
3.6	Parallel Super-Resolution.....	56
3.7	Summary.....	61
.		
4-	Super-resolution using Example-based Techniques	62
4.1	Super-Resolution using Support Vector Machine	63
4.1.1	Support Vector Regression.....	64
4.1.2	Improved SR based Support Vector Regression.....	66
4.1.3	Parallel Implementation of Support Vector Regression	70
4.2	Super-Resolution using Neighbor Embedding	75
4.2.1	Manifold Learning.....	77
4.2.2	Enhanced SR based on Neighbor Embedding	78
4.2.3	Parallel implementation of Neighbor Embedding.....	81
4.3	Summary.....	83
5-	Super-Resolution using Compressed Sensing Framework	84
5.1	Compressive Sensing (CS): A Background.....	85
5.1.1	Sparsity	87
5.1.2	Incoherence	88
5.2	Reconstruction Algorithms in CS.....	88
5.2.1	L_1 minimization algorithm.....	89
5.2.2	Greedy Algorithms.....	89
5.2.3	Combinatorial Algorithms.....	91
5.3	Super-Resolution based Compressive Sensing using CoSaMP	92
5.3.1	Proposed Parallel Implementation of CoSaMP.....	97
5.4	Super-Resolution Method using Self-Similarity	100
5.4.1	Proposed Parallel Implementation of OMP.....	104
5.5	Summary.....	107
6-	Experimental Results	109
6.1	Dataset	110

6.2	Hardware Platforms	113
6.3	Performance Evaluation Criteria.....	113
6.4	Results	114
6.4.1	Super-Resolution based Example Methods.....	114
6.4.2	Compressed Sensing based Super-resolution	146
6.5	Summary	163
7-	Conclusions & Future Work	166
7.1	Conclusion.....	166
7.1.1	Super-Resolution using Support Vector Machine.....	166
7.1.2	Super-Resolution using Manifold Learning.....	168
7.1.3	Super-Resolution using Compressed Sensing Framework...	169
7.1.4	Super-Resolution using Compressed Sensing and Self-Similarity.....	170
7.2	Future Work.....	171
	References.....	172
	Publications.....	187
	Arabic Summary	

List of Figures

Figure 1.1	Forward super-resolution image [3].....	5
Figure 1.2	Ideal super-resolution setup. (a) Four images are taken with relative shifts of half a pixel in horizontal, vertical, and diagonal directions. (b) Their pixels can then be interleaved to generate a double resolution image.....	6
Figure 2.1	Conceptual inverse problem	13
Figure 2.2	SR algorithms taxonomy.....	15
Figure 2.3	Conceptual framework of wavelet-domain-based SR image approach	18
Figure 2.4	Conceptual framework of learning-based SR image approach.....	26
Figure 3.1	The von Neumann architecture.....	43
Figure 3.2	High performance computing trends.....	46
Figure 3.3	Fermi Architecture.....	51
Figure 3.4	CUDA programming model represent thread hierarchy, memory sharing, and communication/synchronization...	53
Figure 3.5	Transition procedure from serial to parallel algorithms...	54
Figure 4.1	Proposed super-resolution algorithm based on support vector regression	68
Figure 4.2	SVM training procedure using the LIBSVM library.....	71
Figure 4.3	SMO serial implementation.....	72
Figure 4.4	Parallel support vector regression implementation	74
Figure 4.5	Proposed super-resolution algorithm based on locally neighbor embedding.	79
Figure 4.6	Sample of LLE algorithm GUP kernels.....	82
Figure 5.1	Compressed sensing (CS) framework.....	86
Figure 5.2	Block diagram of hyperspectral super resolution a) training phase, b) reconstruction phase.....	93
Figure 5.3	Flowchart of the SR based on a CoSaMP algorithm.....	99
Figure 5.4	Sample of CoSaMP GPU kernels	100
Figure 5.5	Flow chart of the proposed super resolution method....	102
Figure 5.6	Orthogonal Matching Pursuit flowchart.....	104
Figure 5.7	Flowchart of the computation of the super resolution method on a CPU/GPU processing scheme.....	106
Figure 5.8	Gauss–Jordan matrix inverse kernels on GPU.....	107

Figure 6.1	True-color of a) SPOT-5 at 23/08/2008, b) Landsat-8 acquired at 24/07/2015.	111
Figure 6.2	EO-1 hyperspectral image cube acquired on 30 April 2002.....	112
Figure 6.3	Samples of training set consist of 30 images; each has size 512×512 pixels, 256 gray levels.....	116
Figure 6.4	The HR images obtained from a) bicubic method, b) POCS method, c) IBP method, d) CPU proposed method and e) GPU proposed method	118
Figure 6.5	Performance for SR method using PSNR (dB) for an input image of size 128×128 using for different patch size.	119
Figure 6.6	Performance for SR method using PSNR (dB) for an input image of size 128×128 for different noise rates	120
Figure 6.7	Performance for SR LLE method using PSNR (dB) for an input image of size 32×32 for different upscaling factor.....	121
Figure 6.8	Speedup of the proposed SR GPU implementation with respect to sequential implementation on a CPU for different image size.....	122
Figure 6.9	The SPOT-4 HR images using a) bicubic method, b) POCS method, c) IBP method, d) CPU proposed method and e) GPU proposed method.....	124
Figure 6.10	The Landsat-8 HR images using a) bicubic method, b) POCS method, c) IBP method, d) CPU proposed method and e) GPU proposed method	125
Figure 6.11	Performance for SR SVR method using PSNR (dB) for different input images with different noise rates.....	127
Figure 6.12	Performance for SR SVR method using PSNR (dB)for different input images at different patch size.	127
Figure 6.13	Performance for SR SVR method using PSNR (dB) for input images of size 32×32 at different magnification factor.....	128
Figure 6.14	The HR images using a) Freeman et al. [8], b) Chang et al. [9] method, c) CPU proposed method and d) GPU proposed method	131
Figure 6.15	Performance for SR based NE method using PSNR (dB) for different input images using different neighbors.....	133

Figure 6.16	The HR images using a) 5×5 , b) 7×7 , c) 9×9 , d) 11×11 and e) 15×15 -patch size.	135
Figure 6.17	Performance of the proposed SR based NE method using PSNR (dB) for different input images at different noise rates.....	136
Figure 6.18	The HR images using different input images at different noise rates. (a) $\sigma=0$, (b) $\sigma=5$, (c) $\sigma=10$, (d) $\sigma=20$ and (e) $\sigma=50$	136
Figure 6.19	SPOT-4 HR images obtained using a) Freeman et al. [96] method, b) Chang et al. [97] method, c) proposed SR based NE using CPU and d) proposed SR based NE using GPU.....	139
Figure 6.20	Landsat-8 HR images obtained using a) Freeman et al. [96] method, b) Chang et al. [97] method, c) proposed SR based NE using CPU and d) proposed SR based NE using GPU	140
Figure 6.21	Performance of the SR based NE method using PSNR (dB) for different input images using different neighbors.....	141
Figure 6.22	HR images using different patch size a) 3×3 , b) 5×5 , c) 7×7 , d) 11×11 and f) 13×13	142
Figure 6.23	Performance of the SR based NE method using PSNR (dB) for different input images using different noise.....	144
Figure 6.24	Performance of the SR based NE method using PSNR (dB) for different input images using different magnification factor.....	145
Figure 6.25	Samples of images cropped from image in Figure 6.2 used in the training phase.....	147
Figure 6.26	The high-resolution image patch dictionary trained by K-SVD atoms of size 9×9 . (b) The low-resolution image patch dictionary trained by K-SVD.....	148
Figure 6.27	(a) Window cropped from image in Figure 6.2 with size 1024×1024 pixels, and (b)– (e) images following magnification by a factor of 2 using (b) bi-cubic interpolation, (c) POCS method, (d) CPU method, and (e) GPU proposed method.....	149
Figure 6.28	(a) Window cropped from image in Figure 6.2 of size 128×128 pixels, and (b)–(e) images magnification by a factor 3 using (b) bi-cubic interpolation, (c) POCS	150

	method, (d) CPU method, and (e) GPU proposed method. The red box denotes the zoomed-in area located in the upper left corner of each image.	
Figure 6.29	(a) Window cropped from image in Figure 6.2 with size 128×128 pixels, and (b) – (e) images magnification by a factor 4 using (b) bi-cubic interpolation, (c) POCS method, (d) CPU method, and (e) GPU proposed method. The red box denotes the zoomed-in area located in the upper left corner of each image.....	151
Figure 6.30	Performance of the proposed method in terms of PSNR for varying dictionary size.....	152
Figure 6.31	Execution times (ms) comparison between CoSaMP and OMP parallel implementations for different image sizes.	153
Figure 6.32	Spectral differences between original HS images, bicubic interpolation, POCS, and proposed HS SR methods.....	155
Figure 6.33	a) Original image, SPOT-4 HR images obtained using: c) Proposed GPU for the SR method	157
Figure 6.34	a) Original image, Landsat-7 HR images obtained using: b) Bi-cubic interpolation and c) GPU proposed methods.....	158
Figure 6.35	Performance of the different four methodologies to extract the base images in the training phase.....	160

List of Tables

Table 2.1	The list of performance of previous works for super-resolution reconstructions methods for hyperspectral image.,	30
Table 2.2	Comparison between Haar method, DCT method, MOD method and K-SVD.....	38
Table 3.1	Features comparison between cluster, grid, and cloud systems.....	49
Table 3.2	Summary of CUDA Architecture Parameters.....	51
Table 4.1	Standard kernel trick.....	65
Table 6.1	The spectral bands characteristics of Spot-4 and Landsat-8 image.....	111
Table 6.2	Hardware specification used in the experiment.....	113
Table 6.3	Results of the reconstructed HR for an input image of size 128×128 using PSNR (dB).....	118
Table 6.4	Execution times (ms) of the proposed SR SVR method using CPU and GPU for different image size.....	118
Table 6.5	Performance of the proposed SR SVR method using PSNR (dB) for different SR methods.....	126
Table 6.6	Execution times (ms) of the proposed SR SVR method using CPU and GPU for different image size; SPOT-5 and Landsat-8 image.....	129
Table 6.7	Performance comparison of SR based NE methods using PNSR for ten test images for 2× magnification by different methods.....	132
Table 6.8	Performance of the proposed SR based NE method using PNSR for different patch sizes and overlapping.....	134
Table 6.9	Performance of the SR based NE method using PNSR for ten images of size 64×64 contaminated by different magnification factor.....	137
Table 6.10	Execution time (ms) of the proposed SR based NE method using GPU and CPU for different image sizes.....	138
Table 6.11	Performance of different SR based NE method using PNSR (dB) for ten test images for 2 magnification.....	141
Table 6.12	Performance of the SR based NE method using PNSR for different images using different patch sizes and overlapping.....	143

Table 6.13	Performance of the SR based NE method using sequential CPU implementation and parallel GPU implementation for different image sizes.....	146
Table 6.14	Performance comparisons of different SR methods using PSNR & SSIM for input image of size 512×512.....	149
Table 6.15	Performance of SR methods using PSNR for different magnified factor values for input image of size 128×1280...	151
Table 6.16	Execution time (ms) of the proposed SR CoSaMP method using GPU and CPU for different image sizes	153
Table 6.17	Speedups of the proposed SR CoSaMP GPU using different threads per block size with respect to sequential implementation on a CPU	154
Table 6.18	Performance comparisons of bi-cubic interpolation and the proposed method in terms of PSNR, SSIM and MSSIM....	159
Table 6.19	Performance of the bi-cubic interpolation and the proposed method in terms of PSNR for different dictionary and window sizes.....	161
Table 6.20	Robustness of the proposed method using different Gaussian noise.....	162
Table 6.21	Execution time for C++ implementation for CPU and GPU proposed method using different image sizes.....	162

Abbreviations

BP	Basis Pursuit
CCD	Charge-Coupled Device
CoSaMP	Compressive Sampling Matching Pursuit
CS	Compressive Sensing
CSNE	Clustering And Supervised Neighbor Embedding
CUDA	Compute Unified Device Architecture
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
FPGAs	Field Programmable Gate Arrays
GA	Genetic Algorithm
GD	Geometric Duality
GPGPU	General Purpose Graphics Processing Unit
GPU	Graphics Processing Unit
HH	High-High
HHSP	Heavy Hitters On Steroids Pursuit
HL	High-Low
HoG	Histogram of Oriented Gradient
HPC	High Performance Computing
HR	High- Resolution
IaaS	Infrastructure as a Service
IBP	Iterative Back-Projection
ICA	Independent Component Analysis
ILP	Instruction-Level Parallelism
IP	Isometric Property
IT	Information Technology
KM	Kernel Matrix
KPCA	Kernel Principal Component Analysis
LH	Low-High
LH-relation	Low–High Relation
LLE	Locally Linear Embedding
LL-relation	Low–Low Relation
LPC	Locality Preserving Constraints
LR	Low Resolution
MAP	Maximum A Posteriori
MCA	Morphological Component Analysis
MIMD	Multiple Instructions, Multiple Data