Super-Resolution Enhancement for Computed Tomography Imaging and Image Processing

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Abhandlung

Vorgelegt von

Kaicong Sun

aus Heilongjiang, China

Hauptberichter: Prof. Dr.-Ing. Sven Simon

Mitberichter: Prof. Dr.-Ing. Volkmar Schulz

Tag der mündlichen Prüfung: 20.12.2021

Institut für Parallele und Verteilte Systeme
der Universität Stuttgart

2021

Abstract

Computed Tomography (CT) as a computer-aided tomographic imaging technique has been widely used for medical diagnostic and becomes prevailing in industrial applications such as quality control and metrology. Comparing to natural images, CT images are usually prone to inferior spatial resolution and more severe noise due to the inherent nature of CT imaging systems. Improving the spatial resolution in the presence of noise is hence a necessary and relevant task for CT imaging.

Super-resolution (SR) is an algorithm-based approach dedicated to spatial resolution enhancement. This work focuses SR for CT from different aspects. Firstly, an iterative multi-image SR algorithm dedicated to joint resolution enhancement and noise removal has been proposed. Besides, a multi-GPU accelerated SR approach based on data parallelism has been presented and it has been seamlessly integrated into the CT system by super-resolving projections acquired by subpixel detector shift such that SR reconstruction is performed on the fly without introducing extra computation time. Additionally, a CNN-based resolution enhancement module (REM) is developed which can be easily plugged in other vision tasks and it is shown that the embedding of REM in image registration not only improves the registration accuracy but also produces resolution-enhanced images which can be used in successive analysis. Last but not least, an extremely fast hardware-efficient video SR model based on recurrent neural network (RNN) is proposed and it reveals a great potential for the application of hardware-embedded SR in fast CT acquisition.

Zusammenfassung

Die Computertomographie (CT) als computergestütztes tomographisches Bildgebungsverfahren ist sowohl in der medizinischen Diagnostik als auch in der industriellen Anwendungen wie der Qualitätskontrolle in der Produktionstechnik und Messtechnik weit verbreitet. Im Vergleich zu natürlichen Bildern haben CT-Bilder normalerweise ein stärkeres Rauschen und eine geringere Auflösung. Die Verbesserung der räumlichen Auflösung bei Rauschen ist daher eine relevante Fragestellung für die CT-Bildgebung.

Super-Resolution (SR) ist ein Verfahren für die räumliche Auflösungserhöhung. Diese Arbeit umfasst verschiedenen Fragestellungen in diesem Kontext. Zunächst wurde ein iterativer Multi-Image-SR-Algorithmus vorgeschlagen, der die Auflösung verbessert und das Rauschen reduziert. Außerdem wurde ein Multi-GPU-beschleunigter SR-Ansatz vorgestellt, der auf Datenparallelität basiert und nahtlos in das CT-System integriert wurde. Dabei werden Projektionen höher aufgelöst, die durch eine Subpixel-Detektorverschiebung erfasst wurden. Dies erfolgt während der laufenden Aufnahmen der Projektionsbilder ohne zusätzliche Rechenzeit. Darüber hinaus wurde ein CNN-basiertes Modul zur Auflösungsverbesserung (REM) entwickelt, das leicht in andere Bildverarbeitungsalgorithmen integriert werden kann. Es wird gezeigt, dass die Einbettung von REM in die Bildregistrierung nicht nur die Registrierungsgenauigkeit verbessert, sondern auch auflösungsverbesserte Bilder erzeugt. Zu guter Letzt wird ein extrem schnelles, hardware-effizientes Video-SR-Modell auf der Grundlage eines rekurrenten neuronalen Netzwerks (RNN) vorgeschlagen, das ein großes Potenzial für den Einsatz von in Hardware eingebetteter SR in den verschiedensten Anwendungen wie der industriellen Computertonograhie hat.

Contents

List of Figures

List of Tables

List of Abbreviations

ADMM	Alternating direction method of multipliers
BFGS	Broyden–Fletcher–Goldfarb–Shanno algorithm
BSWTV	Bilateral spectrum weighted total variation
BTV	Bilateral total variation
CG	Conjugate gradient
CGL	Conjugate gradient with line search
CNN	Convolutional neural network
CT	Computed tomography
DCNN	Deep convolutional neural network
GAN	Generative adversarial network
HR	High-resolution
L-BFGS	Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm
LR	Low-resolution
MISR	Multi-image super-resolution
MRF	Markov random field
PSNR	Peak sinal-to-noise ratio
RNN	Recurrent neural network
SCG	Scaled conjugate gradient
SISR	Single image super-resolution
SR	Super-resolution
SSIM	Structural similarity index
TV	Total variation

Acknowledgement

I would like to express my gratitude to my supervisor Prof. Sven Simon for offering the opportunity of doing my Ph.D. in the highly fascinating area of super-resolution at the department of Parallel Systems. I greatly appreciate his kind advice and support in my research as well as the freedom he gave me to develop my own ideas. Thank you very much!

I greatly appreciate the wonderful time at the department over the past years. In particular, let me thank my colleague Steffen Kieß for his generous support whenever I came to him. I would like also to thank my colleague Trung-Hieu Tran and my previous colleague Roman Krawtschenko. We cooperated on my first journal paper and spent a wonderful time on insightful discussions. Additionally, a big thank goes to my colleagues Dr. Zhe Wang and Timo Schweizer for their support and help on the proof-read of my work. I would like to thank all my other colleagues for the relaxed working atmosphere and wonderful leisure activities.

Besides my colleagues, I would give my big thank to my friends staying with me in Stuttgart. You have accompanied me during my hardest time and encouraged me to develop myself. It would always be my warmest memory in my heart.

I would like to acknowledge all the students that I supervised during the last years, especially Maurice Koch and Trapthi Shivaram Punja. It was a wonderful experience sharing ideas with them.

Last but not least, I would like to deeply thank my whole family and my wife. Thank you very much for the patience and emotional support within the ups and downs of the last years.

Kaicong Sun

Chapter 1

Introduction

1.1 Spatial Resolution of Digital Imaging Systems

In digital imaging systems, the object information conveyed by visible light and X-ray photons, is converted into digital signals by image sensors and recorded for processing and vision. Typical digital imaging systems include but are not limited to digital cameras, microscope, computed tomography (CT) scanners, and radars. One of the determinant quality indicators of imaging systems is the spatial resolution which characterizes the finest structure the system can distinguish. For optical imaging systems, such as a digital camera which composes of optical components and a sensor array, spatial resolution is inherently limited by diffraction due to finite aperture size [1] and discrete sampling grid caused by limited pixel size of the sensors [2]. For CT systems, the spatial resolution is mainly effected by the focal spot size of the X-ray tube and the pixel size of the X-ray sensitive detector [3].

The enhancement of spatial resolution is desired in most of the applications as technical systems have not reached the resolution of the human visual system jet. Common to both optical and CT imaging systems, pixel resolution of the sensor array plays an important role on the spatial resolution of digital imaging systems. According to the Nyquist-Shannon sampling theorem [4], the sampling frequency needs to at least double the highest frequency of the incoming signal to maintain the fidelity of the recovered signal and avoid aliasing

effect. The most straightforward way to increase the pixel resolution is to reduce the pixel size. However, the decrease of the pixel size will lead to a reduction of the collected photons in the sensors and degrade the image quality by a worse signal-to-noise ratio (SNR). In principle, there are two ways to improve the spatial resolution, instrumental-based by hardware upgradation and computational-based using image processing techniques [5]. The instrumental-based solution might cause higher cost in price and maintenance which is especially not feasible for low-end products. In this work, we focus on a hybrid hardware-software based approach which can be integrated into the current setup of imaging systems without or with limited effort on the hardware side such as shifting the detector by a fraction of a pixel.

1.2 Super-Resolution on Computed Tomography

Super-resolution (SR) is an algorithm-based image enhancement technique dedicated to improving the spatial resolution of the imaging systems by exploiting the low-resolution (LR) acquisitions. SR has been an attractive research field for decades and has been applied to different scientific disciplines [5–8]. Image SR can be grouped into two categories: single-image SR (SISR) which exploits self-similarity in the single input LR image [9–12] or employs similarity match from external example database [13–22] and multi-image SR (MISR) [23–40] which explicitly leverages the correlation existing between the reference LR image and the other neighboring input LR images. Comparing to SISR, MISR requires the relative offsets of the multiple input images, either by performing motion estimation as preprocessing or by jointly estimating motion parameters and the expected SR image. Generally, under an appropriate motion compensation, MISR outperforms SISR by exploiting the additional acquired information from neighboring LR images.

Different from natural images on which the majority of the SR algorithms are applied, CT images usually suffer from less spatial resolution and stronger noise due to the limitation of physical imaging systems such as finite focal spot size and low number of photons [41]. In the literature, the SR methods dealing with CT images can be classified into two branches: optimization-based [33–35, 42–47] and learning-based [48–54]. Usually, the optimization-based methods perform SR reconstruction using multiple LR inputs in an iterative fashion and the learning-based approaches manipulate on the single input image.

Specially, to improve the spatial resolution of CT systems, [33–35, 42, 43] perform SR in the projection domain based on the imaging model. Several work [44–47] achieves the resolution enhancement along with the CT reconstruction. Other work [48–54] performs the SR reconstruction in the CT image domain by manipulating the LR input volume. Despite the advances of the existing SR methods, super-resolving CT images contaminated with real-world strong noise could still be a challenging task in practice.

1.3 Scientific Contributions and Applications

The contributions of this work are mainly four-fold. Firstly, an iterative MISR algorithm based on mixed Poisson–Gaussian noise model and bilateral spectrum weighted total variation is proposed to cope with noisy images such as X-ray projections [33, 34]. Secondly, a real-time multi-GPU accelerated MISR method based on data parallelism is developed which runs on the fly during the CT acquisition without introducing extra computation time. Thirdly, the impact of SR on image registration is investigated and demonstrated. Lastly, an extremely fast hardware-efficient video super-resolution (VSR) model based on recurrent neural network (RNN) implemented on field-programmable gate array (FPGA) is introduced which shows a great potential for applying embedded SR module on fast CT systems. The contributions of this thesis are summarized as follows:

- MISR algorithm based on mixed Poisson–Gaussian noise model and bilateral spectrum weighted total variation. Most of the SR methods in the literature assume an additive white Gaussian noise (AWGN) model [5–7] to simplify the system model and computation complexity. However, in reality, the composition of noise in digital imaging systems is much more sophisticated, which extremely holds true for CT. There are mainly two sources of noise dominating in CT imaging: the intensity-independent readout noise and reset noise which can be modeled as an additive Gaussian noise and the intensity-dependent photon shot noise arising from the stochastic nature of the photon-counting process which obeys a Poisson distribution [55–57]. In order to better cope with noisy images, an imaging system model based on more accurate statistical noise description [33, 34] for SR is proposed and adopted. Specially, the proposed method is based on the maximum a posteriori (MAP) estimator. The likelihood function is derived from the mixed Poisson–Gaussian noise model. The

image prior is formulated based on the spectrum of the covariance matrix of the adaptively weighted image gradients dedicated to preserving sharpness and suppressing the remaining noise at the edges. The overall objective function is decomposed and solved by the modified alternating direction method of multipliers (ADMM) algorithm in a gradual-refinement-based fashion. Experiments demonstrate that the proposed SR algorithm outperforms 14 investigated SR methods by an average gain of 0.2dB in PSNR on the publicly available real-world dataset SupER [58].

- Real-time multi-GPU accelerated MISR based on data parallelism for CT. Dealing with large- scale multi-image input can be computationally expensive and time consuming especially for optimization-based MISR methods which usually suffer from the iterative manner. In order to achieve real-time SR for CT imaging, a multi-GPU accelerated MISR framework based on data parallelism [35] is presented. Specially, each GPU deals with an allocated partition of the expected SR image and the final SR image is obtained by image fusion. To synchronize the convergence rate of all the GPUs, we allow communication between GPU and CPU to unify the local variables of the scaled conjugate gradient (SCG) algorithm. In order to avoid border discontinuity, an inner-outer-border exchange mechanism is introduced and adopted. During the CT acquisition, SR is performed on-the-fly following a capture-reconstruct fashion on the projections acquired by subpixel detector shift. Experiments show that the proposed SR method can effectively improve the spatial resolution of CT systems in modulation transfer function (MTF) and visual perception. Besides, comparing to a multi-core CPU implementation, the proposed SR approach achieves a more than 50× speedup on an off-the-shelf 4-GPU system.

- SR-enhanced registration. SR is desired in most of the applications by increasing the visibility of fine structures and improving the visual quality. What would happen when connecting SR with image registration? Firstly, a novel CNN-based image registration model FDRN [59] is proposed. Secondly, a simple yet effective CNN-based resolution enhancement module (REM) is presented, which is a general purpose network and can be easily plugged into different vision model. REM is pretrained and coupled with the registration model FDRN in a cascade manner. To tighten the coupling and increase the robustness, an auxiliary loss acting on the raw LR input is introduced. In the experiments, it is shown that the cascaded network not only improves the registration accuracy but also generates resolution-enhanced images which can be used for successive diagnosis.

- Hardware-efficient RNN-based VSR. A novel residual recurrent neural network (RNN) ERVSR is proposed for real-time VSR on FPGA [60]. Unlike the existing FPGA-based VSR methods which perform SISR over the video sequence, the proposed ERVSR exploits the LR input and the temporal information of the previous frames entailed in the hidden state to reconstruct the HR frame in a residual framework. Concerning the limitation of the hardware resources, slow feature fusion is performd and a channel modulation coefficient is introduced to reduce the model parameters. In order to reduce the memory consumption, the hidden state is compressed by a statistical normalization scheme followed by a fixed-point quantization. The proposed ERVSR is evaluated on multiple public datasets and it is shown that ERVSR outperforms the other state-of-the-art FPGA-based VSR methods by an average gain of 0.28dB in PSNR and supports a real-time output of size 3860×2160 at 76 fps. By resorting to the compact and efficient network architecture, ERVSR shows a great potential for the employment of embedded SR in extreme fast CT.

1.4 Outline

This thesis consists of six chapters which includes introduction and ends with summary. Each of the remaining four chapters presents one of the contributions aforementioned.

Chapter 2 introduces the novel iterative MISR method derived from a mixed Poisson–Gaussian noise model and a bilateral spectrum weighted total variation prior. Extensive experimental results demonstrate the superior performance quantitatively on the public benchmark dataset and visually on the real-world projections.

Chapter 3 presents the real-time multi-GPU accelerated MISR method based on subpixel detector shift for CT imaging systems. Evaluation of the proposed method is conducted to illustrate both the effectiveness on resolution enhancement and the computational speedup comparing to multi-core CPU implementation. The modulation transfer function (MTF) is applied for quantitative assessment at different magnification factors based on the standard ASTM-E1695. Besides, the visual improvement is illustrated by using the QRM bar pattern phantoms.

Chapter 4 demonstrates the impact of resolution enhancement on image registration. A CNN-based image registration network FDRN, a resolution enhancement module REM, and their cascaded network ReFDRN are presented. Experiments on public medical datasets exhibit the effectiveness of REM on image registration.

Chapter 5 describes a hardware-efficient VSR model ERVSR based on the residual RNN. Extensive experiments are conducted on multiple publicly available video and image SR datasets. The proposed ERVSR establishes the state-of-the-art for FPGA-based VSR.

Chapter 6 summarizes the main findings of the thesis.

Chapter 2

Multi-Image Super-Resolution on Noisy Images

In CT imaging systems, there are mainly two sources of noise dominating in the acquisition process: the pixel-dependent photon shot noise originated from the discrete character of photons and the pixel-independent readout noise and reset noise due to the intrinsic thermal and electronic fluctuations in the sensors. The former one can be represented by a Poisson distribution [57, 61] and the latter one arising from the readout circuitry can be modeled as a Gaussian distributed noise. In this chapter, a novel MISR algorithm built on the mixed Poisson–Gaussian (MPG) noise model and bilateral spectrum weighted total variation (BSWTV) is introduced which is presented in the following publications [33, 34]. Specially, in section 2.2, the data fidelity term named MPG is derived from the mixed noise model and solved in the ADMM framework. In section 2.3, the regularization term BSWTV is proposed where an adaptive weighting map of TV is estimated based on the eigenvalues of the weighted-gradient covariance matrix. In conjunction with the data fidelity term introduced in section 2.2, the overall objective function MPG+BSWTV is decomposed and solved by the modified ADMM algorithm where the update of the weighting map is embedded into the standard ADMM framework and follows a momentum-based manner.

2.1 Previous Work of Super-Resolution

Super-resolution (SR) is an algorithmic approach dedicated to improving the spatial resolution of the imaging systems beyond the intrinsic capability of the physical devices. Consequently, object structures are visible with better sharpness without hardware upgrade of the imaging systems. This advantage makes SR extremely attractive in many applications such as remote sensing, video surveillance, smartphone camera, and medical diagnostic [5–8]. Usually, SR restores the HR image by exploring information from single LR or multiple LR images of the same scene with relative scene motions. In general, an observed LR image can be modeled as a degraded representation of the corresponding HR image by taking account of downsampling, blurring, motion effects and the exisiting noise in image acquisition. Therefore, SR reconstruction is considered as an inverse problem [5, 6, 62, 63].

Typically, the observation model can be formulated as following in the pixel domain:

$$\mathbf{y} = A\mathbf{x} + \boldsymbol{n}, \tag{2.1}$$

$\mathbf{x} \in \mathbb{R}^{n\times 1}, \mathbf{y} \in \mathbb{R}^{m\times 1}$ being respectively the latent and captured image rearranged in lexicographic order. The system matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ is usually interpreted as $\mathbf{A} = \boldsymbol{DBM}$ with $\boldsymbol{D} \in \mathbb{R}^{m\times n}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$, and $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ describing the decimation, blurring caused, e.g., by the camera lens, the detector crosstalk, atmospheric turbulence [27], and motion effects, respectively. The vector $\boldsymbol{n} \in \mathbb{R}^{m\times 1}$ indicates the additive noise during image acquisition [5, 6].

In the last three decades, SR has been intensively investigated from frequency domain [23, 64, 65] to image domain [9–22, 24–30, 30–34, 36–40, 60, 66–72], from conventional optimization-based methods [9–12, 24–30, 30–34, 67–72] to more recently emerged deep learning-based approaches [13–22, 36–40, 60, 73]. In the category of conventional methods, some most representative branches include frequency-based [23, 64, 65], projection onto convex sets (POCS) [24, 68], and maximum a posteriori (MAP) [26, 27, 29, 30, 32–34, 69–72, 74, 75]. The pioneering work of frequency-based MISR methods is proposed by Huang et al. [23]. They build the relation between the discrete Fourier transform of the LR images and the contineous Fourier transform of the HR image based on the shift property of Fourier transform. The linear equations in the frequency domain is solved by Least Squares algorithm. Kim et al. extend Huang's work by considering system noise and blur [64, 65].

The POCS-based algorithms [24, 68] address the SR problem from the perspective of set theory. Each LR image or a priori knowledge constructs a constraining convex set which embodies the latent HR image. The intersection of these constraining sets is found iteratively and considered as the reconstructed SR image. The majority of the conventional SR methods adopt the MAP framework where a regularization term derived from an image prior is incorporated with a data fidelity term. Based on different noise models [26,30,34,76], the data fidelity term comes with different formulations. The additive Gaussian noise model which describes the statistical nature of sensors results in a $\ell 2$ error norm and the additive white Laplacian noise model accounting for impulse noise (Salt & Pepper noise) arising from faulty memory locations, transmission in noisy channels, etc. [77, 78] leads to a $\ell 1$ error norm. Typically, $\ell 2$ error norm has the advantage of producing lower variance than $\ell 1$ norm. However, $\ell 2$ norm is sensitive to outliers because it penalizes the errors quadratically. In contrast, $\ell 1$ norm is robust to pixel outliers and motion errors but nondifferentiable at zero. In [29, 30, 70], adaptive norm data fidelity terms are introduced to compensate the individual drawback of the $\ell 1$ and $\ell 2$ error norms. In [26, 69], Lorentzian error norm is applied to SR reconstruction to increase the robustness to outliers by bounding the influence function for large errors. In [79], the performances of Tukey, Lorentzian and Huber norm are studied concerning outliers and compared to $\ell 1$ error norm. To describe the real-world noise more accurately, mixed Poisson–Gaussian noise model is applied to SR reconstructionin in [80–82]. Nevertheless, [80, 81] only consider single image for the SR reconstruction. Traonmilin et al. [82] investigate the acquisition strategy and reconstruction error of high dynamic range SR imaging without regularization.

In recent years, learning-based methods have achieved great success in many applications. Recently emerging learning-based SR methods [13–22, 83, 84] mainly apply deep convolutional neural networks (DCNN) which are trained with a set of LR and HR patch pairs. Particularly, Dong et al. [13] introduce a convolutional neural network (CNN) for single-frame SR. Afterwards, a series of work [14–22, 83, 84] has achieved noticeable performance based on such as residual learning, deeper structure, recursive convolution, dense connection, channel attention, and generative adversarial network (GAN). Lim et al. [17] propose a deep and compact residual network EDSR by effectively removing unnecessary modules in the conventional residual networks. Kim et al. [84] present a deeply-recursive convolutional network which repeatedly utilizes the same convolutional layer up to 16 times. It exploits a large image context due to the network deepth without introducing

more paramters. Tong et al. [84] propose a densely connected deep convolutional network which applies short connection between nonadjacent layers. The idea of dense connection originates from [85] proposed for classification tasks and it allows an efficient training even for a very deep network. In [83], the authors propose an iterative network structure by cascading three residual convolutional networks for super-resolving noisy images contaminated by an additive white Gaussian noise (AWGN). The iterative network structure is very similar to a joint network with deep supervision and skip connection. Common to all the aforementioned learning-based SR methods, the well-trained model reconstructs the HR image from a single LR image by hallucinating the missing high-frequency details using the learned relationship between LR and HR pairs exclusively from the external example database. As a matter of fact, the quality and the feasibility of the training datasets play an important role on the performance of the SR reconstruction. In order to have a better visual perception, some patterns might be fabricated in the reconstructed HR image, especially using GAN, which may result in critical issues to many applications such as metrology and non-destructive testing (NDT). On the contrary, the conventional optimization-based methods are mainly driven by the objective function and the optimization scheme explicitly using the multiple acquired LR images.

In the literature, most of the models are derived based on the assumption that the LR images are corrupted by an AWGN. However, in reality, the composition of noise in imaging systems is more sophisticated. Besides the dominant photon shot noise, readout noise, and reset noise as described at the beginning of this chapter, there is some other noise existing in complementary metal-oxide-semiconductor (CMOS) and charge-coupled device (CCD) detectors such as the Poissonian dark current shot noise which is negligible for exposure time less than one second and the quantization noise which is uniformly distributed with variance equal to $1/12$ and can be neglected compared to the readout noise except in low-illumination conditions [55, 56]. Under low-illumination conditions, the pixel-independent reset noise dominates the Poissonian photon shot noise, while in high-illumination situations, the primary noise is the photon shot noise. Therefore, neither purely Gaussian nor Poisson noise model can formulate the imaging system comprehensively. In the literature, digital image acquisition is generally modeled by mixed Poisson–Gaussian noise [55, 86–88]. To the best of our knowledge, despite the importance of adopting an accurate noise description in the imaging model, the literature on SR based on a mixed Poisson–Gaussian noise model is limited [57, 61, 82]. In this section, a multi-frame SR reconstruction model which is derived from the statistical perspective on the noise properties of imaging systems is

proposed and the performance improvement by applying a more accurate noise model on noisy image SR is demonstrated.

2.2 Super-Resolution Based on Mixed Poisson–Gaussian Noise Model

2.2.1 Observation Model

In digital imaging systems, due to the existence of pixel-dependent and pixel-independent noise, a mixed Poisson–Gaussian noise model is more appropriate to describe the system model. The imaging system can be formulated as following:

$$y_i = z_i + n_p(z_i) + n_g, \tag{2.2}$$

where y_i stands for the intensity value at the ith pixel of the observed image $\mathbf{y}$ contaminated by a mixed Poisson–Gaussian noise and z_i indicates the associated clean pixel value. $n_p(z_i)$ is an intensity-dependent noise with $(z_i + n_p(z_i))/\alpha \sim P(z_i/\alpha)$ where α is a scalar accounting for quantum efficiency and analog gain [89]. n_g represents an additive Gaussian noise with $n_g \sim N(\mu_i, \sigma_i^2)$. $n_p(z_i)$ describes mostly photon shot noise and n_g embodies mainly readout noise and reset noise. As described in Eq. (2.1), the system matrix is defined as $\mathbf{A} = \mathbf{DBM}$ with $\mathbf{z} = \mathbf{Ax}$ and $\mathbf{x} = [x_1, \ldots, x_N]$ being the vectorized latent image.

Since SR is an ill-posed problem, involving a well-defined image prior can effectively constrain the solution domain. Therefore, MAP estimator is preferably adopted for SR reconstruction. The posterior probability $P(\mathbf{x}|\mathbf{y})$ of the SR image $\mathbf{x}$ is formulated based on the Bayes' theorem:

$$P(\mathbf{x}|\mathbf{y}) = \frac{P(\mathbf{y}|\mathbf{x})P(\mathbf{x})}{P(\mathbf{y})} \tag{2.3}$$

where $P(\mathbf{y}|\mathbf{x})$ is the likelihood function and $P(\mathbf{x})$ is the image prior. The denominator $P(\mathbf{y})$ is not related to the unknown $\mathbf{x}$ and will be omitted from the optimization point of view in the latter formulations.

The likelihood function $P(\mathbf{y}|\mathbf{x})$ is derived from the mixed noise model and acts as the data fidelity term in the SR objective function. Assuming that n_p and n_g are mutually independent, the mean and the variance of the intensity of pixel i are obtained as

$$\begin{aligned} E(y_i) &= E(z_i + n_p) + E(n_g) = [\mathbf{A}]_i\mathbf{x} + \mu_i \\ Var(y_i) &= Var(z_i + n_p) + Var(n_g) = \alpha[\mathbf{A}]_i\mathbf{x} + \sigma_i^2, \end{aligned} \tag{2.4}$$

where $[\mathbf{A}]_i$ indicates the ith row of the matrix $\mathbf{A}$. It should be noted that the degradation matrix $\mathbf{A}$, the scalar α and the Gaussian noise parameters μ_i and σ_i are assumed to be known. According to the Central Limit Theorem (CLT), $P(z_i/\alpha) \simeq N(z_i/\alpha, z_i/\alpha)$ as z_i/α being sufficiently large. Hence, the observed value y_i can be approximated by a Gaussian distribution. Based on Eq. (2.4), $y_i \sim \mathcal{N}([\mathbf{A}]_i\mathbf{x} + \mu_i, \alpha[\mathbf{A}]_i\mathbf{x} + \sigma_i^2)$. The probability mass function (PMF) of y_i conditioned on the expected image $\mathbf{x}$ can be expressed as

$$P(y_i|\mathbf{x}) = \frac{1}{\sqrt{2\pi(\alpha[\mathbf{A}]_i\mathbf{x} + \sigma_i^2)}} \exp \frac{-(y_i - [\mathbf{A}]_i\mathbf{x} - \mu_i)^2}{2(\alpha[\mathbf{A}]_i\mathbf{x} + \sigma_i^2)}. \tag{2.5}$$

As the pixels in the image $\mathbf{y}$ are independent, the negative log-likelihood is formulated as

$$\begin{aligned} &-\log P(\mathbf{y}|\mathbf{x}) = -\log \prod_{i=1}^{n} P(y_i|\mathbf{x}) \\ &= \frac{1}{2}\left(\| \mathbf{y} - \mathbf{A}\mathbf{x} - \boldsymbol{\mu} \|_{\mathbf{W}}^2 + \langle \log(\alpha\mathbf{A}\mathbf{x} + \boldsymbol{\sigma}^2), 1 \rangle \right) + c, \end{aligned} \tag{2.6}$$

where $\log(\cdot)$ is the elementwise logarithm, $\langle \cdot, \cdot \rangle$ indicates the inner product and c is a constant. For the sake of brevity, the constant c is omitted in the rest of the work. The intensity-dependent diagonal weight matrix $\mathbf{W}$ is written as

$$\mathbf{W} = \operatorname{diag}\{ \frac{1}{\alpha[\mathbf{A}]_i\mathbf{x} + \sigma_i^2} \}. \tag{2.7}$$

For MISR, instead of having one observed low-resolution (LR) image $\mathbf{y}$, there are m LR images $\mathbf{y}_i$ with the individual system matrix $\mathbf{A}_i$ and additive noise $\mathbf{n}_i$. Assuming the LR images are independent, the formulation in Eq. (2.6) is extended as below:

$$
\begin{aligned}
&-\log P(\mathbf{y}_1 \ldots \mathbf{y}_m|\mathbf{x}) = -\sum_{i=1}^{m} \log P(\mathbf{y}_i|\mathbf{x}) \\
&= \frac{1}{2}\sum_{i=1}^{m} \left(\| \mathbf{y}_i - \mathbf{A}_i\mathbf{x} - \boldsymbol{\mu}_i \|^2_{\mathbf{W}_i} + \langle \log(\alpha_i \mathbf{A}_i\mathbf{x} + \boldsymbol{\sigma}_i^2), 1 \rangle \right).
\end{aligned}
\tag{2.8}
$$

With regard to the image prior $P(\mathbf{x})$, there are several representative priors such as Gaussian Markov random field (MRF), Huber MRF [90], total variation (TV) [76], and bilateral TV (BTV) [27]. MRF is constructed based on the Gibbs density function:

$$
P(\mathbf{x}) = \frac{1}{Z} exp(-\frac{1}{\lambda}\sum_{c\in \mathrm{C}} V_c(\mathbf{x})), \tag{2.9}
$$

where Z is a normalizing constant and λ denotes the "temperature" parameter. $V_c(\cdot)$ is interpreted as the potential of the configuration of $\mathbf{x}$. c represents a local group of pixels called clique and C indicates the set of all the cliques [71]. Different choices of potentials can lead to distinct image estimations.

The well-known Gaussian MRF is formulated as

$$
P(\mathbf{x}) = \frac{1}{2\pi^{\frac{N}{2}} |\mathrm{C}_\mathbf{x}|^{\frac{1}{2}}} exp(-\frac{1}{2}\mathbf{x}^T \mathrm{C}_\mathbf{x}^{-1}\mathbf{x}), \tag{2.10}
$$

where $\mathrm{C}_\mathbf{x}^{-1}$ is a symmetric positive matrix and is often defined as $\Gamma^T\Gamma$ where Γ performs some first or second derivative on image $\mathbf{x}$ which is the well-known Tikhonov regularization:

$$
\sum_{c\in \mathrm{C}} V_c(\mathbf{x}) = \|\Gamma\mathbf{x}\|_2^2 \tag{2.11}
$$

Although Gaussian MRF provides desirable results for denoising purpose, a common criticism is oversmoothing especially on edges because it tends to suppress the discontinuities.

Another popular prior is Huber MRF with the potential of Gibbs distribution as

$$
V_c(a) = \begin{cases} a^2, & |a| \le T, \\ 2T|a| - T^2, & |a| > T. \end{cases} \tag{2.12}
$$

where a is the first or second derivative of the image $\mathbf{x}$. As Huber MRF is heavier-tailed than Gaussian MRF, it performs better for preserving the sharpness at discontinuities such as edges but is non-quadratic.

TV is another widely used prior model. The isotropic TV is defined as the magnitude of the tuplewise gradient in x- and y-axis, while the anisotropic TV is formulated as

$$\sum_{c \in \mathrm{C}} V_c(\mathbf{x}) = \|\nabla \mathbf{x}\|_1 . \tag{2.13}$$

In comparison to $\ell 2$ norm, $\ell 1$ norm favors usually sparse gradients and preserves deep gradients. As shown in [76], TV prior is successfully applied on different noise models. However, it encourages artificial edges which makes the reconstructed image blocky.

Farsiu et al. [27] propose the bilateral TV (BTV) which considers an extended neighborhood of TV and weights their impacts with decaying factors $\gamma(\mathbf{d})$:

$$\sum_{c \in \mathrm{C}} V_c(\mathbf{x}) = \sum_{\mathbf{d}} \gamma(\mathbf{d}) \parallel \mathbf{x} - S_{\mathbf{d}}\mathbf{x} \parallel_1, \quad \mathbf{d} = (d_x, d_y) \tag{2.14}$$

where $\mathbf{d} \in \mathbb{N}^2$ is of size w^2 and w interprets the window size accounting for neighbors in x and y directions. $S_{\mathbf{d}}$ represents the shifting operator along x and y axis by d_x and d_y pixels. $\gamma(\mathbf{d}) := \alpha^{d_x + d_y}$ embodies the spatial decaying effect with a constant α less than one.

For this section, BTV is integrated into the objective function as the regularizer to cope with the ill-posedness of the inverse problem. Combining the data fidelity term derived from the likelihood function $P(\mathbf{y}_1 \dots \mathbf{y}_m | \mathbf{x})$ with the BTV prior model, the MAP estimator can be formulated as

$$\begin{aligned}
&\arg\max_{\mathbf{x}} P(\mathbf{x}|\mathbf{y}_1 \dots \mathbf{y}_m) \\
&= \arg\min_{\mathbf{x}} -\log\left(P(\mathbf{y}_1 \dots \mathbf{y}_m|\mathbf{x})P(\mathbf{x})\right) \\
&= \arg\min_{\mathbf{x}} J(x) \\
&= \arg\min_{\mathbf{x}} \frac{1}{2} \sum_{i=1}^{m} \left(\| \mathbf{y}_i - A_i\mathbf{x} - \mu_i \|^2_{W_i} + \langle \log(\alpha_i A_i \mathbf{x} + \sigma_i^2), 1 \rangle \right) \\
&\quad + \lambda \sum_{\mathbf{d}} \gamma(\mathbf{d}) \parallel \mathbf{x} - S_{\mathbf{d}}\mathbf{x} \parallel_1
\end{aligned} \tag{2.15}$$

with J being the objective function and λ being the weighting factor to adjust the smoothness of the estimated HR image $\mathbf{x}$.

2.2.2 Optimization Method

The objective function J in (2.15) is nonconvex and the BTV regularizer is not everywhere differentiable which makes it difficult to be directly solved using naive gradient-based algorithms. In fact, the proposed objective function J can be decomposed and the reformulated one can be attacked by means of constrained optimization, e.g., dual ascent and ADMM [91]. Since dual ascent is based on Lagrangian and usually has inferior convergence properties comparing to ADMM which benefits from the augmented Lagrangian, ADMM is adopted to solve the objective function. Particularly, the objective function can be split by

$$\begin{aligned} J(\mathbf{x},\mathbf{z}) &= \sum_{i=1}^{m+w^2} g_i(\mathbf{z}_i) \\ \text{subject to } T_i\mathbf{x}-\mathbf{z}_i &= 0, \quad \forall i \in [1, m+w^2] \end{aligned} \tag{2.16}$$

with $\mathbf{z}_i \in \mathbb{R}^N$ and T_i being a matrix:

$$T_i = \begin{cases} I_{N\times N} & ,i \in [1,m] \\ I_{N\times N} - S_{\mathbf{d}} & ,i \in [m+1, m+w^2]. \end{cases} \tag{2.17}$$

Specially, the subfunctions $g_i(\cdot)$ are defined as

$$\begin{aligned} g_i(\mathbf{z}_i) &:= \frac{1}{2}\left(||\mathbf{y}_i - A_i\mathbf{z}_i - \mu_i||^2_{W_i} + \langle \log(\alpha_i A_i \mathbf{z}_i + \sigma_i^2), 1\rangle\right), i \in [1,m], \\ g_i(\mathbf{z}_i) &:= \lambda\gamma(\mathbf{d})||\mathbf{z}_i||_1, \ i \in [m+1, m+w^2]. \end{aligned} \tag{2.18}$$

The augmented Lagrangian is formulated by integrating a quadratic penalty on the equality constraint function into the standard Lagrangian function. Mathematically, it is written

as

$$\begin{aligned}\mathscr{L}_H(\mathbf{x},\mathbf{z},\mathbf{p}) &:= \sum_{i=1}^{m+w^2} \mathscr{L}_{H_i}(\mathbf{x},\mathbf{z}_i,\mathbf{p}_i) \\ &:= \sum_{i=1}^{m+w^2} \left(g_i(\mathbf{z}_i) + \langle \mathbf{p}_i, T_i\mathbf{x}-\mathbf{z}_i\rangle + \frac{1}{2}||T_i\mathbf{x}-\mathbf{z}_i||^2_{H_i} \right)\end{aligned} \tag{2.19}$$

where $\mathbf{p}_i$ is the dual variable associated with the constraint and matrix H_i is defined as

$$H_i := \text{diag}[\rho_i,\dots,\rho_i], \quad \forall i \in [1,\cdots,m+w^2] \tag{2.20}$$

with ρ_i being some positive value acting as the update step size for the dual variable $\mathbf{p}_i$.

Unlike the method of multiplier (MM) which updates the primal variables jointly, ADMM updates the primal variables $\mathbf{x},\mathbf{z}_i$ in an alternating fashion which enables the decomposability of the objective function J when it is separable. Specially, the decomposed objective function rewritten in Eq. (2.16) can be solved in the following iterative scheme:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \sum_{i=1}^{m+w^2+1} \frac{\rho_i}{2}||T_i\mathbf{x}-\mathbf{z}_i^k+\frac{\mathbf{p}_i^k}{\rho_i}||_2^2 \tag{2.21a}$$

$$\mathbf{z}_i^{k+1} = \arg\min_{\mathbf{z}_i} g_i(\mathbf{z}_i) + \frac{\rho_i}{2}||\mathbf{z}_i - T_i\mathbf{x}^{k+1} - \frac{\mathbf{p}_i^k}{\rho_i}||_2^2 \tag{2.21b}$$

$$\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + \rho_i(T_i\mathbf{x}^{k+1}-\mathbf{z}_i^{k+1}). \tag{2.21c}$$

Since Eq. (2.21a) is quadratic and differentiable, it can be easily solved by conjugate gradient (CG) algorithm. The gradient of Eq. (2.21a) is expressed as

$$\begin{aligned}V_i(\mathbf{x}) &:= \frac{\rho_i}{2}||T_i\mathbf{x}-\mathbf{z}_i^k+\frac{\mathbf{p}_i^k}{\rho_i}||_2^2, \\ \nabla V_i(\mathbf{x}) &= \rho_i T_i^T(T_i\mathbf{x}-\mathbf{z}_i^k+\frac{\mathbf{p}_i^k}{\rho_i}).\end{aligned} \tag{2.22}$$

To update $\mathbf{z}_i^{k+1}$ with $i \in [1,m]$, $g_i(\mathbf{z}_i)$ can be solved using, e.g., Newton's method, L-

BFGS [92] and scaled conjugate gradient (SCG). For clarity, we define:

$$\begin{aligned} g_i(\mathbf{z}_i) &:= f_i(\mathbf{z}_i) + h_i(\mathbf{z}_i), \\ f_i(\mathbf{z}_i) &:= \frac{1}{2} \parallel \mathbf{y}_i - A_i\mathbf{z}_i - \mu_i \parallel^2_{W_i}, \\ h_i(\mathbf{z}_i) &:= \frac{1}{2}\langle \log(\alpha_i A_i \mathbf{z}_i + \sigma_i^2), 1\rangle, \\ u_i(\mathbf{z}_i) &:= \frac{\rho_i}{2}||\mathbf{z}_i - T_i\mathbf{x}^{k+1} - \frac{\mathbf{p}_i^k}{\rho_i}||_2^2. \end{aligned} \tag{2.23}$$

The gradient of $\mathscr{L}_{H_i}(\mathbf{x}^{k+1}, \mathbf{z}_i, \mathbf{p}_i^k)$ on $\mathbf{z}_i$ with $i \in [1,m]$ can be calculated by

$$\begin{aligned} \nabla\mathscr{L}_{H_i}(\mathbf{x}^{k+1}, \mathbf{z}_i, \mathbf{p}_i^k) &= \nabla f_i(\mathbf{z}_i) + \nabla h_i(\mathbf{z}_i) + \nabla u_i(\mathbf{z}_i) \\ &= \frac{1}{2}\Big(-2A_i^T \frac{\mathbf{y}_i - A_i\mathbf{z}_i - \mu_i}{\alpha_i A_i\mathbf{z}_i + \sigma_i^2} - \alpha_i A_i^T \frac{(\mathbf{y}_i - A_i\mathbf{z}_i - \mu_i)^2}{(\alpha_i A_i\mathbf{z}_i + \sigma_i^2)^2} \\ &\quad + \alpha_i A_i^T \frac{1}{\alpha_i A_i\mathbf{z}_i + \sigma_i^2}\Big) + \rho_i(\mathbf{z}_i - T_i\mathbf{x}^{k+1} - \frac{\mathbf{p}_i^k}{\rho_i}) \end{aligned} \tag{2.24}$$

where division and square are elementwise operations.

Newton's method or L-BFGS requires the calculation of the step size by applying line search algorithm like Wolfe conditions. In addition, Newton's method requires the calculation of the Hessian matrix of $\mathscr{L}_{H_i}(\mathbf{x}^{k+1}, \mathbf{z}_i, \mathbf{p}_i^k)$ which demands more computation time. In the following experiments, SCG is used to update $\mathbf{z}_i^{k+1}$ for $i \in [1,m]$.

For calculating $\mathbf{z}_i^{k+1}$ associated with the BTV prior, i.e., $i \in \left[m+1, m+w^2\right]$, $g_i(\mathbf{z}_i)$ can be calculated via the proximal operator of the $\ell 1$ norm:

$$\begin{aligned} \mathbf{z}_i^{k+1} &= \arg\min_{\mathbf{z}_i} \lambda\gamma(\mathbf{d})||\mathbf{z}_i||_1 + \frac{\rho_i}{2}||\mathbf{z}_i - T_i\mathbf{x}^k - \frac{\mathbf{p}_i^k}{\rho_i}||_2^2 \\ &= prox_{\lambda\gamma(\mathbf{d})(\rho_i)^{-1}||\cdot||_1}(T_i\mathbf{x}^k + \frac{\mathbf{p}_i^k}{\rho_i}) \end{aligned} \tag{2.25}$$

Cameraman Lena Coffee Text

Figure 2.1: 8-bit grayscale natural images for quantitative analysis.

$$[\mathbf{z}_i^{k+1}]_j = \begin{cases} [T_i\mathbf{x}^k + \dfrac{\mathbf{p}_i^k}{\rho_i}]_j - \dfrac{\lambda\gamma(\mathbf{d})}{\rho_i}, & [T_i\mathbf{x}^k + \dfrac{\mathbf{p}_i^k}{\rho_i}]_j \geq \dfrac{\lambda\gamma(\mathbf{d})}{\rho_i}, \\ 0, & |[T_i\mathbf{x}^k + \dfrac{\mathbf{p}_i^k}{\rho_i}]_j| \leq \dfrac{\lambda\gamma(\mathbf{d})}{\rho_i}, \\ [T_i\mathbf{x}^k + \dfrac{\mathbf{p}_i^k}{\rho_i}]_j + \dfrac{\lambda\gamma(\mathbf{d})}{\rho_i}, & [T_i\mathbf{x}^k + \dfrac{\mathbf{p}_i^k}{\rho_i}]_j \leq -\dfrac{\lambda\gamma(\mathbf{d})}{\rho_i}. \end{cases} \tag{2.26}$$

In order to improve the convergence and reduce the dependency of the initialization, the penalty parameters ρ_i can be updated adaptively at each iteration k by synchronizing the convergence of the primal residual r_i^k and the dual residual s_i^k with the scheme [91]:

$$\rho_i^{k+1} = \begin{cases} c^{inc}\rho_i^k, & ||r_i^k||_2 > c||s_i^k||_2, \\ \rho_i^k/c^{dec}, & ||s_i^k||_2 > c||r_i^k||_2, \\ \rho_i^k, & otherwise, \end{cases} \tag{2.27}$$

where c^{inc}, c^{dec}, c are constants greater than one. The primal and dual residuals r_i^k, s_i^k are calculated as

$$\begin{aligned} r_i^{k+1} &:= \mathbf{T}_i x^{k+1} - z_i^{k+1} \\ s_i^{k+1} &:= -\rho_i^k \mathbf{T}_i^T (z_i^{k+1} - z_i^k). \end{aligned} \tag{2.28}$$

The pseudocode for solving the decomposed objective function of MPGSR based on the ADMM framework is demonstrated in Algorithm 2.1.

Algorithm 2.1 MPGSR Algorithm

Initialize $\lambda, w, \mu, \sigma, \rho, \alpha, iter\_admm, c^{inc}, c^{dec}, c, \varepsilon_1, \varepsilon_2$.
Calculate $D_i B_i M_i$ system matrix.
Load LR images $\mathbf{y}_i,\ i \in [1, \cdots, m]$.
procedure Solving MPGSR by ADMM
    $\mathbf{z}_i^0 = Upscaling(y_1),\ i \in [1, m]$.
    $\mathbf{z}_i^0 = 0,\ i \in \left[m+1, m+w^2\right]$.
    $\mathbf{p}_i^0 = 0,\ i \in \left[1, m+w^2\right]$.
    $\mathbf{x}^0 = 0$.
    while $k < iter\_admm$ do
        CG($\mathbf{x}^k$) ▷ by 2.21a, 2.22
        for $i = 1$ to $m+w^2$ do
            if $i < m+1$ then
                SCG($\mathbf{z}_i^k$) ▷ by 2.21b, 2.24
            else if $i < m+w^2$ then
                Prox($\mathbf{z}_i^k$) ▷ by 2.21b, 2.25, 2.26
            else
                Prox($\mathbf{z}_i^k$) ▷ by 2.21b
            end if
            Update $\mathbf{p}_i^k$. ▷ by 2.21c
            Update ρ_i^k. ▷ by 2.27, 2.28
        end for
        if $\sum_i ||r_i^k||_2^2 < \varepsilon_1$ and $\sum_i ||s_i^k||_2^2 < \varepsilon_2$ then
            break
        end if
        $k = k+1$
    end while
    end while
    return Reconstructed HR image $\mathbf{x}$.
end procedure

2.2.3 Experiments and Results

This section demonstrates the performance of the proposed MPGSR algorithm evaluated by the synthetic 8-bit natural images under different noise levels and on the captured 16-bit X-ray images. Besides, the effect of inaccurate motion and blur estimations is evaluated.To demonstrate the merits of the proposed data term, a comparison with $\ell 1$ and $\ell 2$ data terms under Tikhonov and BTV regularizations was carried out. It is

Table 2.1: Methods and optimization parameters

Methods	Regularization	Optimizer	α	w
$\ell 1$-Tik.	Tikhonov	ADMM (CG)	-	-
$\ell 2$-Tik.	Tikhonov	ADMM (CG & SCG)	-	-
MPGSR-Tik.	Tikhonov	ADMM (CG & SCG)	-	-
$\ell 1$-BTV	BTV	ADMM (CG)	0.4	2
$\ell 2$-BTV	BTV	ADMM (CG & SCG)	0.4	2
SR-PG† [81]	TV	ADMM (CG & SCG)	-	-
DPSR [22]	SRResNet+	Adam	-	-
MPGSR	BTV	ADMM (CG & SCG)	0.4	2

necessary to note that the regularization term of the proposed MPGSR is BTV and the notation MPGSR-Tik is used to distinguish from MPGSR by replacing BTV with Tikhonov prior. Besides, the multi-frame SR-PG† was extended for comparison, which originally applies generalized Anscombe transform on the single-frame mixed Poisson–Gaussian noise model and employs TV regularization [81]. All the optimization-based methods mentioned above were implemented using the ADMM framework. Furthermore, MPGSR was compared with the state-of-the-art CNN-based plug-and-play SR method DPSR [22].

Experiments with Natural Images

1) Effect of different levels of mixed Poisson–Gaussian noise: For quantitative evaluation, four 8-bit natural images illustrated in Fig. 2.1 are considered as the reference images. RGB images were converted to YCrCb channels and solely the Y channel was used. The gray-value LR images corrupted by different levels of mixed Poisson–Gaussian noise were generated according to Eq. (2.2). In particular, each reference image was firstly rescaled to peak intensity 120 and considered as the ground truth (GT). The rescaled image was then shifted by $(0,0),(1,0),(1,1)$ and $(0,1)$ pixel to obtain four images. These four images were blurred by an isotropic 3×3 Gaussian filter and then subsampled by factor 2. Each of the degraded LR images was corrupted by a Poisson noise and then an AWGN with $\sigma = 1.2$. The same scenario was performed on generating another set of

Table 2.2: PSNR (dB) obtained by different methods (All the methods were implemented based on the ADMM algorithm except DPSR).

Image	Peak+$\boldsymbol{\sigma}$	Bic.	$\ell 1$+Tik.	$\ell 2$+Tik.	MPGSR-Tik.	$\ell 1$+BTV	$\ell 2$+BTV	SR-PG†[a] [81]	DPSR [22]	MPGSR
Camera	120+1.2	23.44	24.83	25.15	25.64	25.87	26.14	25.87	26.01	26.69
	180+1.8	23.99	25.35	25.89	26.18	26.41	26.77	25.46	26.20	27.42
Lena	120+1.2	23.43	24.83	25.16	25.64	25.92	26.21	25.87	25.78	26.84
	180+1.8	24.89	26.11	26.36	26.67	26.52	26.82	26.07	26.27	27.29
Coffee	120+1.2	23.63	24.57	24.76	24.93	24.78	24.95	24.61	24.60	25.26
	180+1.8	24.11	24.93	24.90	25.27	25.12	25.30	24.59	25.20	25.65
Text	120+1.2	23.97	26.16	26.31	27.12	26.60	27.14	27.09	27.54	27.85
	180+1.8	24.83	26.88	26.96	27.88	27.32	27.71	27.13	28.18	28.79

[a] SR-PG† denotes the implementation of the extended multi-frame SR-PG based on the ADMM framework instead of the original spectral projected gradient (SPG) algorithm for better performance.

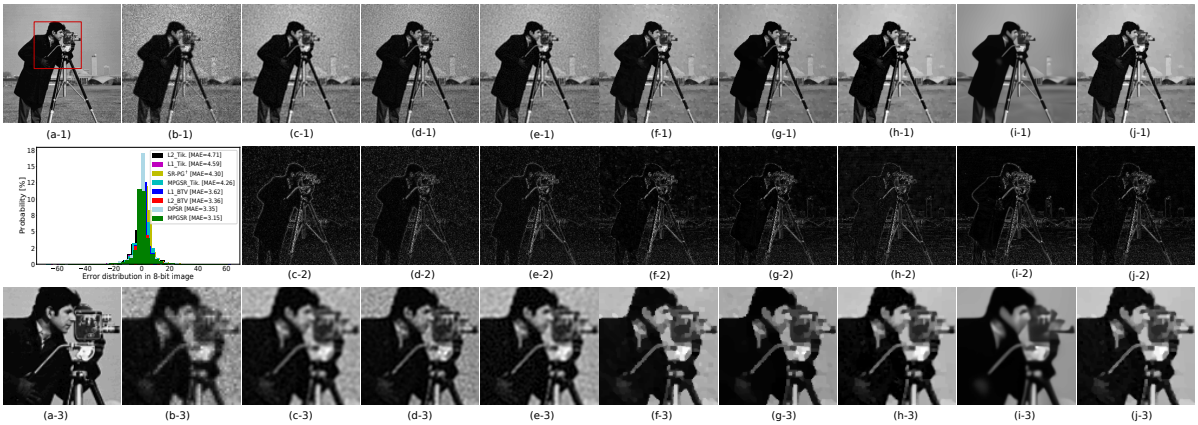


Figure 2.2: Comparison of different SR methods on Cameraman with mixed noise 120+1.2. (a-1) GT image, (b-1) LR image, (c-1) $\ell 1$+Tikhonov, (d-1) $\ell 2$+Tikhonov, (e-1) MPGSR-Tikhonov, (f-1) $\ell 1$+BTV, (g-1) $\ell 2$+BTV, (h-1) SR-PG†, (i-1) DPSR, (j-1) MPGSR, (c-2)∼(j-2) residual images and (a-3)∼(j-3) ROI.

LR images with peak intensity 180 and $\sigma = 1.8$. To quantitatively assess the estimated images, PSNR calculated by Eq. (2.29) was utilized where x^* and $\hat{x}$ denote the GT image and the estimated HR image respectively and I_{max} represents the maximum intensity of the GT image.

$$PSNR(\hat{\mathbf{x}}, \mathbf{x}^*) = 10\log_{10}\left(\frac{I_{max}^2}{MSE(\mathbf{x}^*, \hat{\mathbf{x}})}\right) \tag{2.29}$$

To implement the Tikhonov regularization, a commonly used Laplacian kernel was adopted defined by

$$\Gamma = \frac{1}{8}\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

As the update step size ρ_i of the dual variable $\mathbf{p}_i$ has a direct impact on the convergence of the ADMM algorithm. Generally, larger ρ emphasizes less on minimizing the objective function while smaller ρ penalizes less on the violation of the primal feasibility. In the following experiments, ρ was updated iteratively according to Eq. (2.27). The weighting parameter λ was chosen carefully as a constant by sweeping over $[0.01, 0.05, \cdots, 50, 100]$. Stopping criteria based on the primal and dual residuals [91] was applied and the maximum iterations was limited up to 400. A wide parameter sweep for DPSR was carried out and the parameters providing the highest PSNR were selected. The detailed implementation parameters are listed in Table 3.1

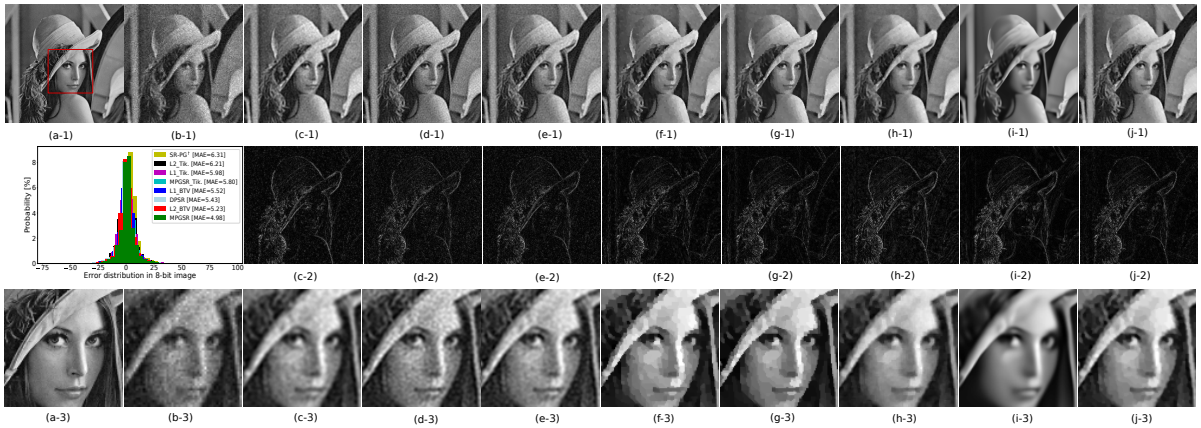


Figure 2.3: Comparison of different SR methods on Lena with mixed noise 180+1.8. (a-1) GT image, (b-1) LR image, (c-1) $\ell 1$+Tikhonov, (d-1) $\ell 2$+Tikhonov, (e-1) MPGSR-Tikhonov, (f-1) $\ell 1$+BTV, (g-1) $\ell 2$+BTV, (h-1) SR-PG†, (i-1) DPSR, (j-1) MPGSR, (c-2)$\sim$(j-2) residual images and (a-3)$\sim$(j-3) ROI.

A comparison of different methods on Cameraman with mixed noise 120+1.2 is depicted in Fig. 2.2. The residual images between the reconstructed images and the GT are shown in the second row. The error distribution and the mean absolute error (MAE) of different methods are quantitatively analysed and exhibited in the histogram. The third row focuses on the marked region of interest (ROI). As shown, SR-PG† produces a plausible HR estimation but inclines to a positive bias in the error distribution. DPSR tends to suppress the noise aggressively but compromises the sharpness of the edges in return. The proposed MPGSR outperforms the others visually and possesses a light-tailed error distribution. In the residual and ROI images, it is shown that the flat regions reconstructed by MPGSR are smoothed, while the sharpness of the edges is preserved. In Fig. 2.3, Lena with mixed noise 180+1.8 is demonstrated. DPSR seems to overreact to the noise and sacrifices the high-frequency information. The proposed MPGSR produces a clear delineation of Lena with the best error distribution. The quantitative evaluations of four test images are summarized in Table 2.2. Furthermore, aiming for analysing the robustness of the proposed method, five realizations of the mixed noise 180+1.8 on Cameraman, Lena, Coffee and Text were carried out and the performance is listed in Table 2.3. It is shown that although the performance differs over five realizations, the proposed MPGSR generates a stabilized superior performance overall.

2) Effect of inaccurate estimations of motion and blur: In order to evaluate the effect of motion uncertainty, five experiments on Cameraman contaminated with mixed noise

Table 2.3: PSNR (dB) of five realizations of Cameraman, Lena, Coffee and Text contaminated with 180+1.8 mixed Poisson–Gaussian noise.

ID	Bicubic	$\ell 1$+BTV	$\ell 2$+BTV	SR-PG†	DPSR	MPGSR
Camera #1	23.94	26.40	26.62	25.54	26.06	27.33
Camera #2	23.93	26.38	26.71	25.45	26.14	27.38
Camera #3	23.94	26.41	26.67	25.45	26.10	27.37
Camera #4	23.99	26.41	26.77	25.46	26.20	27.42
Camera #5	23.96	26.43	26.74	25.43	26.00	27.42
Mean	23.95	26.41	26.70	25.47	26.10	27.38
SEM	0.010	0.007	0.024	0.017	0.030	0.015
Lena #1	24.86	26.51	26.87	26.09	26.27	27.34
Lena #2	24.89	26.52	26.82	26.07	26.27	27.29
Lena #3	24.84	26.48	26.81	26.08	26.23	27.32
Lena #4	24.85	26.47	26.85	26.07	26.19	27.32
Lena #5	24.88	26.52	26.83	26.12	26.32	27.34
Mean	24.86	26.50	26.84	26.09	26.26	27.32
SEM	0.008	0.009	0.010	0.008	0.020	0.008
Coffee #1	24.10	25.14	25.29	24.57	25.24	25.64
Coffee #2	24.06	25.12	25.30	24.58	25.09	25.64
Coffee #3	24.11	25.12	25.30	24.59	25.20	25.65
Coffee #4	24.06	25.12	25.30	24.57	25.10	25.64
Coffee #5	24.11	25.15	25.30	24.57	25.06	25.65
Mean	24.09	25.13	25.30	24.58	25.14	25.64
SEM	0.010	0.006	0.002	0.004	0.031	0.002
Text #1	24.86	27.30	27.73	27.13	28.14	28.80
Text #2	24.85	27.33	27.70	27.13	28.10	28.77
Text #3	24.83	27.32	27.71	27.09	28.18	28.79
Text #4	24.85	27.29	27.68	27.09	28.07	28.74
Text #5	24.85	27.27	27.66	27.08	28.05	28.74
Mean	24.85	27.30	27.70	27.10	28.11	28.77
SEM	0.004	0.010	0.011	0.010	0.021	0.011

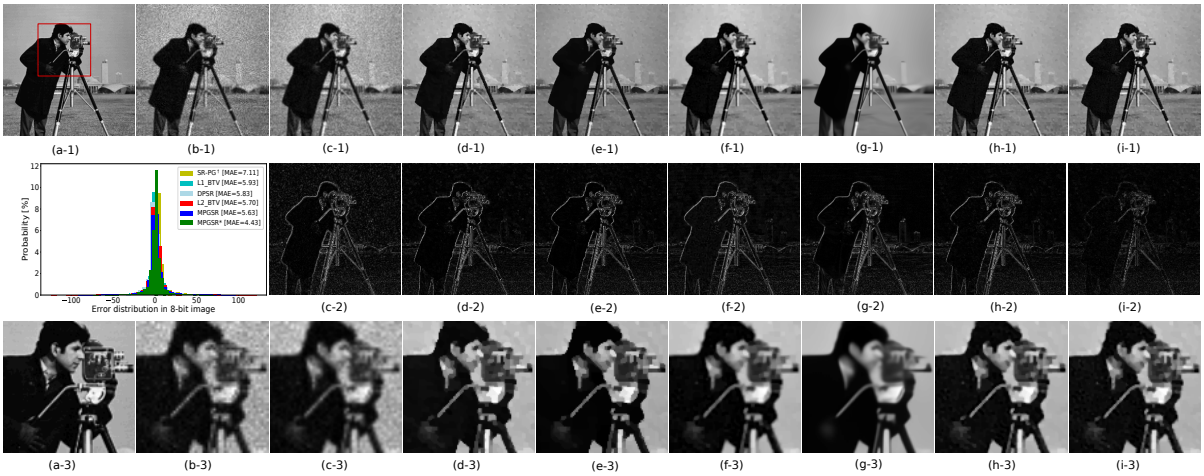


Figure 2.4: Comparison of different methods on Cameraman contaminated with mixed noise 180+1.8, inaccurate estimations of motion and blur. (a-1) GT image, (b-1) LR image, (c-1) bicubic interpolation, (d-1) $\ell 1$+BTV, (e-1) $\ell 2$+BTV, (f-1) SR-PG†, (g-1) DPSR, (h-1) MPGSR, (i-1) MPGSR* with accurate estimations of motion and blur, (c-2)$\sim$(i-2) residual images and (a-3)$\sim$(i-3) ROI.

180+1.8 and inaccurate motion were conducted. Particularly, four LR images were generated by unexpected motion with an uniformly distributed additive random offset $\mathbb{R}^2 \in [-0.1, 0.1]$. Additionally, to model the situation where an inaccurate blurring kernel $\boldsymbol{B_i}$ is applied, the synthetic LR images were blurred by a 5x5 Gaussian filter, while the HR image was reconstructed by a 3x3 Gaussian kernel. The mean and the SEM of the PSNR of five realizations are demonstrated in Table 2.4. As exhibited, the proposed MPGSR generates the highest PSNR with strong robustness against inaccurate motion. Besides, it is shown that the inaccuracy of blur kernel has a significant impact on the SR performance. The reconstructed images by different approaches are illustrated in Fig. 2.4. As shown in the residual images, all the methods generate larger deviations on the edges in contrary to the one reconstructed with the accurate estimations of motion and blur. Regardless of that, the proposed MPGSR produces the best MAE and narrowest error distribution.

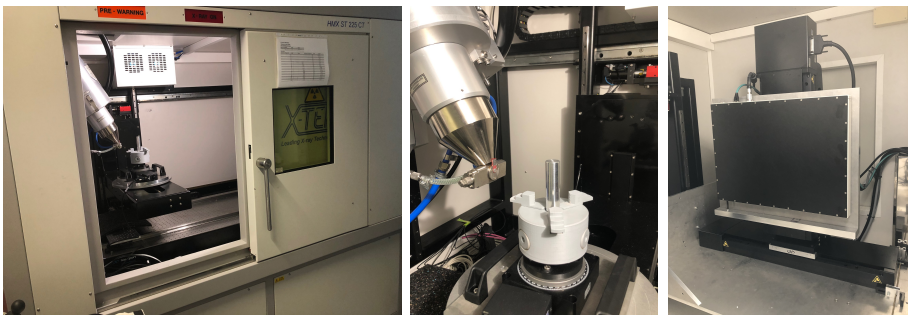
Experiments with X-ray Images

As mentioned in Section 2.1, mixed Poisson–Gaussian noise model is derived based on the statistical properties of imaging systems equipped with electronic devices, such as

Table 2.4: The mean and SEM of the PSNR(dB) obtained by different methods on Cameraman 180+1.8 with inaccurate estimations of motion and blur.

Mean & SEM	$\ell 1$+BTV	$\ell 2$+BTV	SR-PG[†]	DPSR	MPGSR	MPGSR*[a]
Mean (Inaccu. M)	26.31	26.87	25.78	25.86	27.23	27.28
SEM (Inaccu. M)	0.023	0.013	0.018	0.042	0.020	0.025
Mean (Inaccu. M&B)	23.96	24.08	24.04	24.36	24.50	27.89
SEM (Inaccu. M&B)	0.067	0.077	0.077	0.089	0.086	0.015

[a] MPGSR* denotes the results with accurate estimations of motion and blur.



a) b) c)

Figure 2.5: CT scanner equipped with mounted linear stages. a) side view; b) X-ray tube and rotatable object (aluminium cylindrical phantom); (c) X-ray detector mounted on the controllable linear stages.

CCD and CMOS sensors. In this section, the effectiveness of MPGSR on real-world X-ray images is validated. The X-ray imaging process can be described in the following steps: X-ray photons are generated by hitting the accelerated electrons on the target. A spectrum of X-ray photons with different energies, known as polychromatic beam, interact with the objects. Some of the photons are absorbed by the object and the rest arrive at the X-ray detector. For the current widely used energy-integrating detectors, the arriving X-ray photons are converted into light photons in the detector scintillator. The number of generated light photons follows a probability distribution which is proportional to the energy of the incident X-ray photons. These light photons strike the silicon photodiodes and in turn release photoelectrons. The cumulative photoelectrons are amplified and recorded by analog-to-digital (A/D) converters leading to an intensity value. Actually, the noise existing in the above mentioned process is from two sources: the additive electronic noise arisen from the X-ray detector assuming to be Gaussian distributed, and the compound Poisson distributed noise due to the nature of the polyenergetic photon-

Table 2.5: Measurement setup for capturing X-ray images.

Test objects	Voltage (kV)	Current (uA)	Exposure time (s)
Resolution target	100	60	0.33
Hard disk drive	100	70	2

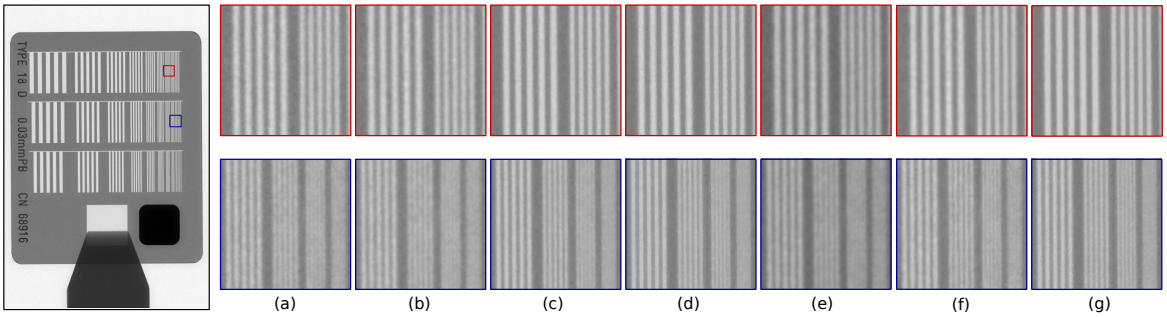


Figure 2.6: Comparison of different SR methods on the X-ray image of a resolution target. ROIs: (a) LR images, (b) bicubic interpolation, (c) $\ell 1$+BTV, (d) $\ell 2$+BTV, (e) SR-PG†, (f) DPSR and (g) MPGSR.

counting statistics [93–96]. In [94, 97], the compound Poisson can be approximated by a scaled Poisson distribution. Therefore, according to the current research, the statistical property of CT imaging can be approximately modeled by a mixed Poisson–Gaussian noise which coincides with [61, 89, 98].

The LR images were acquired by the Nikon HMX ST 225 CT scanner as shown in Fig. 2.5. The CT scanner is equipped with a flat panel Varian PaxScan@4030E detector which has a pixel size of $127\mu m \times 127\mu m$. The detector is mounted on the controllable linear stages, Newport M-IMS400CCHA and Newport M-IMS300V respectively for x- and y-positioning so that the detector can be shifted to a predefined position with movement accuracy up to $1\mu m$. A resolution target made of lead and a hard disk drive were taken as test specimens. Four 16-bit X-ray images for each specimen were captured by shifting the detector with half a pixel rightwards, downwards, leftwards, and upwards. The detailed parameter setup is depicted in Table 2.5.

The reconstructed images of the resolution target and the hard disk drive by different methods with the closeup views are demonstrated in Fig. 2.6 and Fig. 2.7, respectively. In practice even inaccurate estimations of motion and blur are not avoidable, in Fig. 2.6 it is shown that the edges of the vertical bar patterns reconstructed by the proposed MPGSR

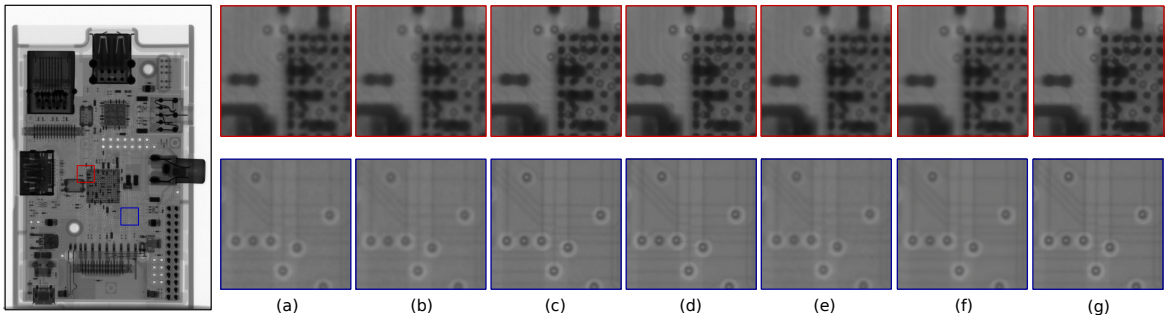


Figure 2.7: Comparison of different SR methods on the X-ray image of a hard disk drive. ROIs: (a) LR images, (b) bicubic interpolation, (c) $\ell 1$+BTV, (d) $\ell 2$+BTV, (e) SR-PG†, (f) DPSR and (g) MPGSR.

are more sharpened and meanwhile, the noisy air-regions between the bars are smoothed in both marked regions. In Fig. 2.7, the vias and stripes in the hard disk drive can be better visualized by MPGSR, while the other methods either blur the edges or emphasize the noise in the flat regions.

2.3 Super-Resolution Based on Bilateral Spectrum Weighted Total Variation

In last section, the data fidelity term based on the mixed Poisson–Gaussian noise model was presented. As X-ray images are usually much more noisy than natural images, a regularization technique is proposed, named bilateral spectrum weighted total variation (BSWTV), aiming at effectively suppressing the noise and meanwhile preserving the sharpness of fine structures for noisy image super-resolution.

2.3.1 Previous Work of Image Priors for Noise Removal

Due to the ill-posedness of the SR problems, the existing methods employ either explicitly handcrafted image priors or implicit priors. The majority of the optimization-based traditional methods utilize the handcrafted priors. Specially, Rudin et al. [76] introduce the total variation (TV) as the regularization for image denoising. In [27], bilateral total variation (BTV) is proposed by concerning photometric and geometric distance in

an extended neighborhood. Yuan et al. [28] propose a regional spatially adaptive total variation (RSATV) based on spatial information filtering and clustering which partition the image into multiple segments. However, pixels within each segment are limited to an equal weight.

Besides the above gradient-based priors, nonlocal-means (NL-means) [99] based on the self-similarity exploits the natural redundancy of image patterns aiming to average the pixels which are surrounded by similar textures. Specially, the NL-means algorithm is formulated as

$$x(i) = \sum_{j \in R_i} w(i,j)x(j), \tag{2.30}$$

where $x(i)$ is the estimated image pixel. Weight $w(i,j)$ depicts the similarity between pixel i and j with $\sum_j w(i,j) = 1, j \in R_i$ where R_i denotes the search window of pixel i. The weight $w(i,j)$ is measured by

$$w(i,j) = \frac{1}{Z(i)} exp\left(-\frac{||\mathbf{N}(i) - \mathbf{N}(j)||^2_{2,\sigma}}{\eta^2}\right). \tag{2.31}$$

$\mathbf{N}(i), \mathbf{N}(j)$ indicate respectively the square neighborhood of pixel i and j. $||\mathbf{N}(i) - \mathbf{N}(j)||^2_{2,\sigma}$ is the weighted Euclidean distance with σ being the standard deviation of the Gaussian kernel. $Z(i)$ denotes the normalizing constant $Z(i) = \sum_j exp\left(-\frac{||\mathbf{N}(i)-\mathbf{N}(j)||^2_{2,\sigma}}{\eta^2}\right)$ with η being the constant filtering parameter. In [100], Protter et al. generalize the use of NL-means in MISR without performing explicit motion estimation by applying the patch extraction operation in the data term.

To overcome the performance decline caused by contrast losses, Gilboa et al. [101] propose a variational regularization, nonlocal TV (NLTV), based on the self-similarity which can be formulated as

$$V_c(\mathbf{x}) = \sum_{\mathbf{D}=(D_x,D_y)} ||\Phi_D(\boldsymbol{S}_D - \boldsymbol{I})\mathbf{x}||_1, \tag{2.32}$$

where (D_x, D_y) indicates the shift vector with $D_x, D_y \in [-(R-1)/2, (R-1)/2]$ and R is the window size. Matrix $\boldsymbol{S}_D$ acts as the shift operator and Φ_D represents the weighting map associated with the shift vector $\mathbf{D}$ defined as

$$\Phi_D(i,j) = exp\left(-\frac{||\mathbf{N}(i,j) - \mathbf{N}(i+D_x, j+D_y)||^2_2}{\eta^2}\right). \tag{2.33}$$

$\mathbf{N}(i,j)$ denotes the neighbors of the center pixel (i,j) in the similarity patch of size r and η is the filtering parameter which controls the smoothness. However, NLTV usually suffers from the drawback of remaining noise in the surroundings of the edges especially when the image is not oversmoothed.

2.3.2 Bilateral Spectrum Weighted Total Variation

As TV performs smoothing without concerning the features, it is prone to oversmoothness, staircasing effect, and contrast losses. In order to alleviate the contrast losses, inspired by [102], the information entailed in the gradient covariance matrix is exploited to distinguish patterns. For images contaminated by a mixed Poisson–Gaussian noise, according to Proposition 2.3.1, the gradients in the flat regions follow a white isotropic Gaussian distribution with a limited variance which theoretically enables us to differentiate the mixed noise in flat areas from the edges.

Proposition 2.3.1. Let us define an observed digital image $y:\mathbb{N}_0^2\to\mathbb{R}$ contaminated by a mixed Poisson–Gaussian noise as $y=z+n_p(z)+n_g$ where $(z_i+n_p(z_i))/\alpha\sim P(z_i/\alpha)$ with z_i being the expected pixel value and α being a scalar. n_g is an independent additive Gaussian noise with $n_g(i)\sim N(\mu_i,\sigma_i^2)$. If we have a homogeneous region $\Omega\subset\mathbb{N}_0^2$, where $\forall i,j\in\Omega,\ |z_i+\mu_i-z_j-\mu_j|<\varepsilon_1,|\alpha z_i+\sigma_i^2-\alpha z_j-\sigma_j^2|<\varepsilon_2,\forall\varepsilon_1,\varepsilon_2>0$, then the gradient of each element i in Ω has the same isotropic white Gaussian distribution $\nabla_x y(i),\nabla_y y(i)\sim N(0,(\alpha z_i+\sigma_i^2)/2)$ and a collection of the gradients obeys an isotropic white Gaussian distribution.

Proof. Considering a digital image $y:\mathbb{N}_0^2\to\mathbb{R}$ contaminated by a mixed Poisson–Gaussian noise as $y=z+n_p(z)+n_g$ where $(z_{i,j}+n_p(z_{i,j}))/\alpha\sim P(z_{i,j}/\alpha)$ with $z_{i,j}$ being the noiseless intensity value at pixel (i,j) and α being a scalar. n_g is an additive Gaussian noise with $n_g(i,j)\sim N(\mu_{i,j},\sigma_{i,j}^2)$. According to the Central Limit Theorem (CLT), when $z_{i,j}/\alpha$ is sufficiently large, we have $(z_{i,j}+n_p(z_{i,j}))/\alpha\sim P(z_{i,j}/\alpha)\simeq N(z_{i,j}/\alpha,z_{i,j}/\alpha)$. Therefore, we have $z_{i,j}+n_p(z_{i,j})\sim N(z_{i,j},\alpha z_{i,j})$. As n_p and n_g are independent, we yield $y(i,j)\sim N(z_{i,j}+\mu_{i,j},\alpha z_{i,j}+\sigma_{i,j}^2)$. As element $(i+1,j)$ and $(i-1,j)$ are independent, we have $E(\nabla_x y(i,j))=(z_{i+1,j}+\mu_{i+1,j}-z_{i-1,j}-\mu_{i-1,j})/2$ and $Var(\nabla_x y(i,j))=(\alpha z_{i+1,j}+\sigma_{i+1,j}^2+\alpha z_{i-1,j}+\sigma_{i-1,j}^2)/4$. If elements $(i-1,j),(i,j),(i+1,j)\in\Omega$ where $\forall(m,n),(p,q)$ satisfying $|z_{m,n}+\mu_{m,n}-z_{p,q}-\mu_{p,q}|<\varepsilon_1,|\alpha z_{m,n}+\sigma_{m,n}^2-\alpha z_{p,q}-\sigma_{p,q}^2|<\varepsilon_2,\forall\varepsilon_1,\varepsilon_2>0$, then we have $\nabla_x y(i,j)\sim N(0,(\alpha z_{i,j}+\sigma_{i,j}^2)/2)$. The derivation holds also for $\nabla_y y(i,j)$. In the homogeneous region Ω, a collection

of the gradients with the same isotropic white Gaussian distribution can be considered as multiple realizations of an isotropic white Gaussian distributed variable at different image locations. □

The proposed BSWTV is based on the spectrum of the weighted-gradient covariance matrix as formulated in Eqs. (2.34) (2.35) (2.36). BSWTV possesses the merit that it adopts a gradually refined weighting map by introducing an inhomogeneous shrink coefficient.

$$BSWTV(\mathbf{x}) := ||\Phi\nabla\mathbf{x}||_1, \Phi = diag[\phi_1, \cdots, \phi_n] \tag{2.34}$$

The ith diagonal element of the weighting map Φ is defined as

$$\phi_i = exp(-|\lambda_{i1} - \lambda_{i2}|/\eta^2), \tag{2.35}$$

where η is the smoothing parameter which controls the dynamic range of Φ. $\lambda_{i1}, \lambda_{i2}$ are the eigenvalues of the covariance matrix of the bilateral weighted gradients G_i which is formulated as

$$\begin{aligned} G_i &= \begin{bmatrix} \omega_1 g_x^1, \ldots, \omega_j g_x^j, \ldots, \omega_q g_x^q \\ \omega_1 g_y^1, \ldots, \omega_j g_y^j, \ldots, \omega_q g_y^q \end{bmatrix}, \ \omega_j = \xi_j^{d(i,j)} \\ \text{with } \xi_j^k &= \xi_j^{k-1} \left(\gamma + \frac{(1-\gamma)}{1 + exp\left(f(\Phi_{N_j}^{k-1}) \right)} \right), \ j \in \mathbf{N}_i. \end{aligned} \tag{2.36}$$

g_j represents the gradient at pixel j and is expressed as $g_j := (g_x^j, g_y^j) = (\nabla_x x_j, \nabla_y x_j)$. The square patch centered at pixel i has the amount of $q = r^2$ pixels and is defined as $\mathbf{N}_i = \{j : |i-j| \leq (r-1)/2\}$ with r being the odd size of the patch. ω_j acts as the weight assigned to each individual neighbor in $\mathbf{N}_i$ and indicates the significance of the neighbor j to the center pixel i which depends on the distance $d(i,j) := |d_x(i,j)| + |d_y(i,j)|$ along x and y axis with $d_x, d_y \in [-(r-1)/2, (r-1)/2]$ and the local adaptive shrink coefficient ξ_j. The superscript of ξ_j^k depicts the kth iteration of the ADMM algorithm and the decay scalar $\gamma \in [0,1]$ serves for the shrinkage of the spread of the gradients within the patch. The intuition of introducing ξ and γ is to adaptively "squeeze" the gradient matrix G such that the discrepancy between eigenvalues λ_{i1} and λ_{i2} decreases as the algorithm converges and the mask of the edges in the weighting map Φ becomes thinned. Φ_{N_j} denotes the neighbors of pixel j in the weighting map Φ. f is a function of Φ_{N_j} which controls the

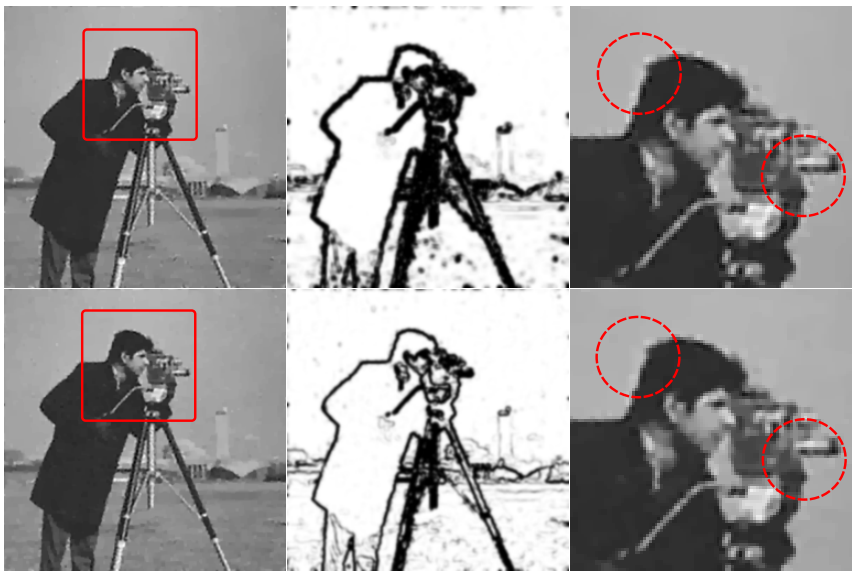


Figure 2.8: Impact of the decay parameter γ on the SR performance (2×). Top: $\gamma = 1$, PSNR = 30.35dB, SSIM = 0.8577; Bottom: $\gamma = 0.8$, PSNR = 30.47dB, SSIM = 0.8607.

whitening based on the image content and enables an inhomogeneous decay of $\boldsymbol{\xi}$. Specially, for flat regions, f is supposed to be a large positive value such that the shrink coefficient ξ_j is decreased by γ and the weighting map gets further whitened, while for fine structures, f should be a large negative value so that the shrink coefficient ξ_j is not attenuated. A simple choice of $f(\mathbf{x})$ could be an affine function $f(\mathbf{x}) := a(\bar{\mathbf{x}} - b)$ where $\bar{\mathbf{x}}$ denotes the mean of vector $\mathbf{x}$ and the positive scalars a, b are the amplitude and shift parameters, respectively. Therefore, based on the previous Φ, map $\boldsymbol{\xi}$ is inhomogeneously shrinked by factors in the range of $(\gamma, 1)$.

In Fig. 2.8, the effectiveness of leveraging the decay parameter γ is demonstrated. The top row illustrates the weighting map Φ and the SR image without decaying the shrink coefficient. The bottom row exhibits the results with $\gamma = 0.8$. As shown, the weighting map Φ in the bottom row has much thinned mask for edges than the counterpart in the top row under the same smoothing parameter η. Consequently, the SR image has much cleaner and pleasant contours without oversmoothing the fine structures. A detailed analysis of the effectiveness of the decay scalar γ, the smoothing parameter η, and the shift parameter b on the reconstruction performance is demonstrated in Section 2.3.4.

The update of the weighting map Φ is embedded in the ADMM framework as described in Algorithm 2.3 in Section 2.3.3. For the sake of suppressing the outliers and enhancing the convergence stability in the ADMM update scheme, the weighting map Φ is smoothed by a Gaussian filter with an iteratively decreased kernel width and updated in a momentum-based fashion in each ADMM iteration. A detailed description of the update of the weighting map Φ is given in Section 2.3.3.

Combining the regularization term expressed in Eq. (2.34) with the data fidelity term introduced in Eq. (2.8), the overall objective function is expressed as

$$\mathrm{J} = \frac{1}{2}\sum_{i=1}^{m}\left(\| \mathbf{y}_i - \mathbf{A}_i\mathbf{x} - \boldsymbol{\mu}_i \|^2_{\mathbf{W}_i} + \langle \log(\alpha_i\mathbf{A}_i\mathbf{x} + \boldsymbol{\sigma}_i^2), 1\rangle\right) + \lambda||\Phi\nabla\mathbf{x}||_1, \tag{2.37}$$

with λ being the weight of the regularization term. Due to the fact that the data fidelity term is derived from a mixed Poisson–Gaussian noise model, the above algorithm is named as MPG+BSWTV.

2.3.3 Optimization Method

Decomposition and ADMM

Similar to Section 2.2, the ADMM algorithm is employed to solve the decomposed objective function. Particularly, the anisotropic TV is adopted in the implementation: $||\Phi\nabla\mathbf{x}||_1 = ||\Phi(\boldsymbol{S}_x - \boldsymbol{I})\mathbf{x}||_1 + ||\Phi(\boldsymbol{S}_y - \boldsymbol{I})\mathbf{x}||_1$ where matrices $\boldsymbol{S}_x, \boldsymbol{S}_y$ perform respectively the shift operation along x and y axis by one pixel. Therefore, the overall optimization problem can be split into $m+2$ subproblems with the corresponding constraints as

$$\begin{aligned} &\arg\min_{\mathbf{x},\mathbf{z}_i} J = \sum_{i=1}^{m+2} g_i(\mathbf{z}_i) \\ &\text{subject to } \mathbf{T}_i\mathbf{x} - \mathbf{z}_i = 0, \quad \forall i \in [1, m+2] \end{aligned} \tag{2.38}$$

with $\mathbf{z}_i \in \mathbb{R}^N$ and $\mathbf{T}_i$ being a matrix:

$$\mathbf{T}_i = \begin{cases} \boldsymbol{I}_{N\times N}, & i \in [1,m], \\ \Phi(\boldsymbol{S}_x - \boldsymbol{I}_{N\times N}), & i = m+1, \\ \Phi(\boldsymbol{S}_y - \boldsymbol{I}_{N\times N}), & i = m+2. \end{cases} \tag{2.39}$$

The subfunctions $g_i(\cdot)$ are defined as following:

$$\begin{aligned} g_i(\mathbf{z}_i) &:= \frac{1}{2}\left(||\mathbf{y}_i - \mathbf{A}_i\mathbf{z}_i - \boldsymbol{\mu}_i||^2_{\mathbf{W}_i} + \langle \log(\alpha_i \mathbf{A}_i\mathbf{z}_i + \boldsymbol{\sigma}_i^2), 1\rangle\right), i \in [1,m], \\ g_i(\mathbf{z}_i) &:= \lambda||\mathbf{z}_i||_1, i \in [m+1, m+2]. \end{aligned} \tag{2.40}$$

Therefore, the augmented Lagrangian is formulated as

$$\mathscr{L}_H(\mathbf{x},\mathbf{z},\mathbf{p}) = \sum_{i=1}^{m+2} \mathscr{L}_{H_i}(\mathbf{x},\mathbf{z}_i,\mathbf{p}_i) = \sum_{i=1}^{m+2}\left(g_i(\mathbf{z}_i) + \langle \mathbf{p}_i, \mathbf{T}_i\mathbf{x} - \mathbf{z}_i\rangle + \frac{1}{2}||\mathbf{T}_i\mathbf{x} - \mathbf{z}_i||^2_{\mathbf{H}_i}\right), \tag{2.41}$$

where $\mathbf{p}_i$ is the dual variable associated with the individual constraint and matrix $\mathbf{H}_i$ is defined as

$$\mathbf{H}_i := \mathrm{diag}[\rho_i, \ldots, \rho_i], \quad \forall i \in [1, \cdots, m+2] \tag{2.42}$$

with ρ_i being a positive scalar acting as the update step size of the dual variable $\mathbf{p}_i$.

The decomposed objective function formulated in Eq. (2.38) is solved in the following iterative scheme:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \sum_{i=1}^{m+2} \frac{\rho_i}{2}||\mathbf{T}_i\mathbf{x} - \mathbf{z}_i^k + \frac{\mathbf{p}_i^k}{\rho_i}||_2^2 \tag{2.43a}$$

$$\mathbf{z}_i^{k+1} = \arg\min_{\mathbf{z}_i} g_i(\mathbf{z}_i) + \frac{\rho_i}{2}||\mathbf{z}_i - \mathbf{T}_i\mathbf{x}^{k+1} - \frac{\mathbf{p}_i^k}{\rho_i}||_2^2 \tag{2.43b}$$

$$\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + \rho_i(\mathbf{T}_i\mathbf{x}^{k+1} - \mathbf{z}_i^{k+1}). \tag{2.43c}$$

The update of $\mathbf{x}^{k+1}$ is performed by the conjugate gradient (CG) algorithm. $g_i(\mathbf{z}_i)$ with $i \in [1,m]$ is solved by scaled conjugate gradient (SCG). To update $\mathbf{z}_i^{k+1}$ associated with

the BSWTV prior, i.e., $i \in [m+1, m+2]$, the proximal operator of the L1-norm is utilized:

$$\begin{aligned} \mathbf{z}_i^{k+1} &= \arg\min_{\mathbf{z}_i} \lambda ||\mathbf{z}_i||_1 + \frac{\rho_i}{2} ||\mathbf{z}_i - \mathbf{T}_i\mathbf{x}^k - \frac{\mathbf{p}_i^k}{\rho_i}||_2^2 \\ &= prox_{\lambda(\rho_i)^{-1}||\cdot||_1}(\mathbf{T}_i\mathbf{x}^k + \frac{\mathbf{p}_i^k}{\rho_i}) \end{aligned} \tag{2.44}$$

and the closed-form solution can be obtained as

$$[\mathbf{z}_i^{k+1}]_j = \begin{cases} [\mathbf{T}_i\mathbf{x}^k + \frac{\mathbf{p}_i^k}{\rho_i}]_j - \frac{\lambda}{\rho_i}, & [\mathbf{T}_i\mathbf{x}^k + \frac{\mathbf{p}_i^k}{\rho_i}]_j \geq \frac{\lambda}{\rho_i}, \\ 0, & |[\mathbf{T}_i\mathbf{x}^k + \frac{\mathbf{p}_i^k}{\rho_i}]_j| \leq \frac{\lambda}{\rho_i}, \\ [\mathbf{T}_i\mathbf{x}^k + \frac{\mathbf{p}_i^k}{\rho_i}]_j + \frac{\lambda}{\rho_i}, & [\mathbf{T}_i\mathbf{x}^k + \frac{\mathbf{p}_i^k}{\rho_i}]_j \leq -\frac{\lambda}{\rho_i}. \end{cases} \tag{2.45}$$

The penalty parameter ρ_i is updated iteratively by means of synchronizing the convergence of the primal residual $\mathbf{r}_i^k$ and the dual residual $\mathbf{s}_i^k$ with the scheme in [91]:

$$\rho_i^{k+1} = \begin{cases} c_1\rho_i^k, & ||\mathbf{r}_i^k||_2 > c||\mathbf{s}_i^k||_2, \\ \rho_i^k/c_2, & ||\mathbf{s}_i^k||_2 > c||\mathbf{r}_i^k||_2, \\ \rho_i^k, & otherwise, \end{cases} \tag{2.46}$$

where c_1, c_2, c are constants with $c_1 > 1, c_2 > 1, c > 1$. The primal and dual residuals $\mathbf{r}_i^k$, $\mathbf{s}_i^k$ are calculated as

$$\begin{aligned} \mathbf{r}_i^{k+1} &= \mathbf{T}_i\mathbf{x}^{k+1} - \mathbf{z}_i^{k+1}, \\ \mathbf{s}_i^{k+1} &= -\rho_i^k\mathbf{T}_i^T(\mathbf{z}_i^{k+1} - \mathbf{z}_i^k). \end{aligned} \tag{2.47}$$

An early stopping criteria based on the primal and dual residuals is used as depicted in Algorithm 2.3.

Update of Weighting Map

The weighting map Φ is updated iteratively within the ADMM framework. In particular, in order to enhance the convergence stability and update efficiency, two additional steps are performed following Eqs. (2.35) and (2.36). Firstly, the weighting map Φ is smoothed by convolving with an isotropic 2D Gaussian kernel $G(\sigma_\Phi)$ to alleviate the effect of outliers on the weighting map. Secondly, the smoothed weighting map is updated in a momentum-based manner to avoid strong fluctuation of the objective during convergence. In particular, a decay scalar γ is employed to iteratively decrease the Gaussian parameter σ_Φ and the momentum coefficient β. Consequently, the mask of the edges in the weighting map is thinned and sharpened in a moderate manner and the remaining noise surrounding the edges is gradually and effectively suppressed. The update framework of the weighting map Φ in the kth ADMM iteration is formulated as following in Algorithm 2.2.

Algorithm 2.2 Update of Weighting Map

1: Initialize $\Phi, \boldsymbol{\xi}, \gamma, \eta, \beta, r, \sigma_\Phi, \sigma_{\min}$.
2: procedure Calculating Weighting Map
3: Calculate $\boldsymbol{\xi}^k(\boldsymbol{\xi}^{k-1}, \Phi^{k-1}, \gamma)$ ▷ by Eq. (2.36)
4: $\Phi^k \leftarrow$ Calculate $\Phi(\mathbf{x}^{k-1}, \boldsymbol{\xi}^k, \eta)$ ▷ by Eq. (2.35)
5: $\sigma_\Phi^k = max(\sigma_{\min}, \gamma\sigma_\Phi^{k-1})$
6: $\Phi^k = G(\sigma_\Phi^k) * \Phi^k$
7: $\beta^k = \gamma\beta^{k-1}$.
8: $\Phi^k = \beta^k\Phi^{k-1} + (1-\beta^k)\Phi^k$
9: end procedure

Overall Optimization Framework

The update of the weighting map as described in Algorithm 2.2 needs to be integrated into the ADMM framework. As the weighting map Φ is coupled with the latent image $\mathbf{x}$ as expressed in Eq. (2.43a) and is used for the update of the variables $\mathbf{z}_i$ and $\mathbf{p}_i$ for $i \in [m+1, m+2]$ as formulated in Eqs. (2.44) and (2.43c), Φ is updated with $\mathbf{x}$ being fixed as the first step of the ADMM iteration. The pseudocode for solving the overall objective function is presented in Algorithm 2.3. As depicted, there are four main steps performed in sequence to respectively update $\Phi, \mathbf{x}, \mathbf{z}$, and $\mathbf{p}$ in each ADMM iteration. Specially, the computational complexity for updating the weighting map Φ in each ADMM

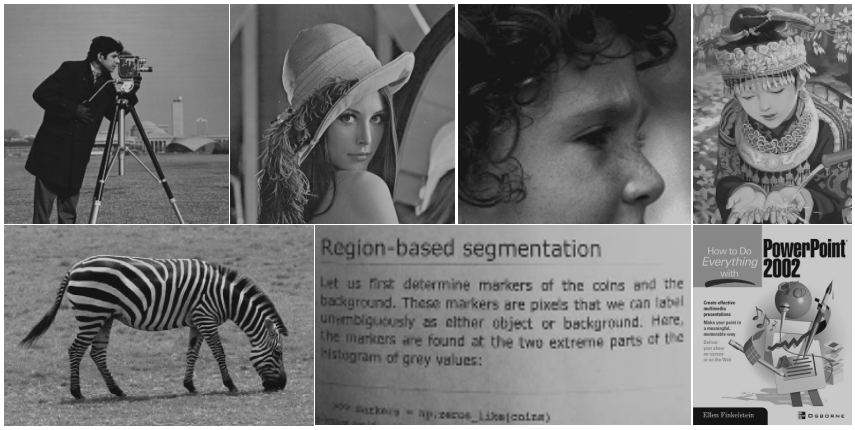


Figure 2.9: 8-bit gray-scale natural images for quantitative analysis.

iteration is $\mathcal{O}(r^2w^2N)$ with r^2 being the patch size and w^2 being the Gaussian kernel size. $\mathbf{x}$ is iteratively solved by CG which has a computational complexity of $\mathcal{O}(k_xN^2)$ with k_x being the amount of iterations of CG. For solving $\mathbf{z}_i$ with respect to the data term with $i \in [1,m]$, SCG is employed which has the computational cost of $\mathcal{O}(mk_zN^2)$ with k_z being the number of iterations for SCG. For $\mathbf{z}_i$ associated with the regularization term, the proximal operator is utilized which has the complexity of $\mathcal{O}(N^2)$. The last step is to solve the dual variables $\mathbf{p}$. For $\mathbf{p}_i$ with $i \in [1,m]$, $\mathbf{T}_i$ denotes the identity matrix and the computational cost is $\mathcal{O}(N)$ and for $i \in [m+1,m+2]$, the update of $\mathbf{p}_i$ has $\mathcal{O}(N^2)$ computational complexity. Hence, the overall computational complexity for each ADMM iteration is $\mathcal{O}((k_x+mk_z)N^2)$.

2.3.4 Experiments and Results

In this section, extensive experiments are conducted to evaluate the proposed method on the synthetic and the real-world images. The introduced approach is benchmarked with the state-of-the-art methods on the public real-world dataset SupER [58].

Super-Resolution on Synthetic Images

The proposed MPG+BSWTV is evaluated on the gray-value reference images as shown in Fig. 2.9. Specially, the system matrix $\mathbf{A}$ is assumed to be known and the scalar α is set as 1. Four LR images corrupted by a mixed Poisson–Gaussian noise with peak intensity 200 and

Algorithm 2.3 Proposed Algorithm

Initialize $\Phi, \boldsymbol{\xi}, \boldsymbol{\sigma}, \lambda, r, \eta, \beta, \gamma, \sigma_\Phi, \sigma_{min}, \rho, iter, \alpha, c, c_1, c_2$
Load observed images $\mathbf{y}_i$ $i \in [1, \cdots, m]$
procedure Solving ADMM
    $\mathbf{z}_i^0 = \mathbf{y}_1$ for denoising, Bicubic($\mathbf{y}_1$) for SR, $i \in [1, m]$
    $\mathbf{z}_i^0 = 0,\ i \in [m+1, m+2]$
    $\mathbf{p}_i^0 = 0,\ i \in [1, m+2]$
    $\mathbf{x}^0 = 0$
    while $k < iter$ do
        Update weighting map Φ ▷ by Alg. 2.2
        CG($\mathbf{x}^k$) ▷ by Eq. (2.43a)
        for $i = 1$ to $m+2$ do
            if $i \in [1, m]$ then
                SCG($\mathbf{z}_i^k$) ▷ by Eq. (2.43b)
            else
                Prox($\mathbf{z}_i^k$) ▷ by Eqs. (2.43b), (2.44), (2.45)
            end if
            Update ρ_i^k ▷ by Eqs. (2.46), (2.47)
            Update $\mathbf{p}_i^k$ ▷ by Eq. (2.43c)
        end for
        if $\sum_i(||\mathbf{r}_i^{k-1}||_2^2 - ||\mathbf{r}_i^k||_2^2)/\sum_i(||\mathbf{r}_i^{k-1}||_2^2) < \varepsilon_1$ and $\sum_i(||\mathbf{s}_i^{k-1}||_2^2 - ||\mathbf{s}_i^k||_2^2)/\sum_i(||\mathbf{s}_i^{k-1}||_2^2) < \varepsilon_2$ then
            break
        end if
        $k = k+1$
    end while
    end while
    return reconstructed image $\mathbf{x}$.
end procedure

$\sigma = 2$ were generated for each reference image as described in Section 2.2.3. The proposed method was compared with L1+BTV [27], L2+NLTV [101], MPGSR [33], L2+BSWTV, EDSR [17], RBPN [40], and DPSR [22]. Note that the notation L2+NLTV indicates the L2-norm data term in conjunction with NLTV as the regularizer and the same notation manner is employed for L1+BTV and L2+BSWTV. All the above TV-based methods were implemented by the ADMM algorithm. For a better interpretation, MPGSR is denoted as MPG+BTV in the latter formulation. EDSR and DPSR are well-known CNN-based single-frame SR methods and RBPN is one of the state-of-the-art video SR (VSR) networks

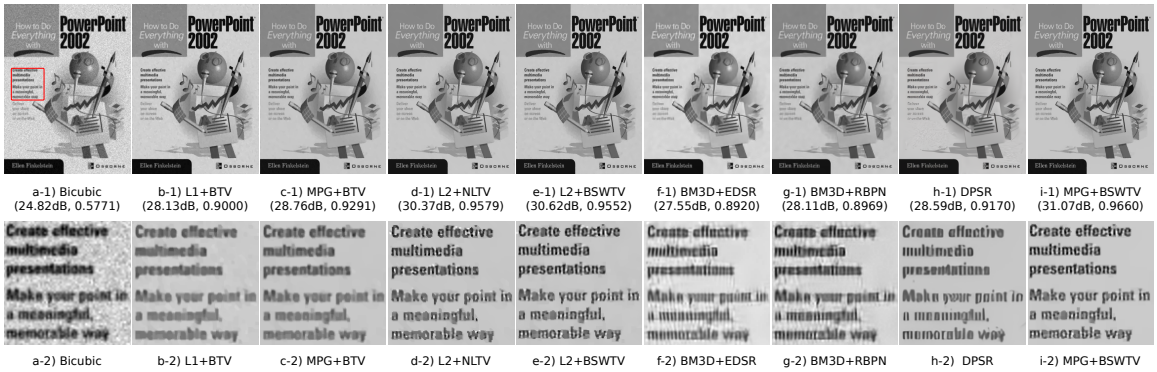


Figure 2.10: Comparison of different SR methods for 2× on PPT3 contaminated by a mixed Poisson–Gaussian noise with peak intensity 200 and $\sigma = 2$: (a) bicubic, (b) L1+BTV, (c) MPGSR, (d) L2+NLTV, (e) L2+BSWTV, (f) EDSR, (g) RBPN, (h) DPSR, and (i) MPG+BSWTV.

which uses multiple LR frames as input. Since EDSR and RBPN are originally trained on noiseless images, EDSR and RBPN were retrained using their original source code on the datasets which were contaminated by the mixed Poisson–Gaussian noise as the same noise level as the testing images. For all the investigated methods, the parameters producing the best PSNR performance were adopted. The reconstructed image PPT3 by different approaches is demonstrated in Fig. 2.10. It is shown that the proposed MPG+BSWTV provides a remarkable improvement quantitatively and qualitatively by jointly enhancing the image resolution and suppressing the residual noise surrounding the characters. The VSR method RBPN generates better result than EDSR as expected by exploiting the information entailed in the neighboring frames. The DPSR tends to suppress the noise aggressively which leads to an oversmoothness of the detailed structures. The quantitative results are summarized in Table 2.6.

Super-Resolution on Real-World Images

To validate the proposed method, experiments on the publicly available SupER dataset [58] which contains images of 14 scenes were conducted. Each of the 14 scenes is captured under multiple modes including motion types, binning factor, and compression levels. Each mode contains 40 LR images by capturing stop-motion videos. SR reconstruction was performed on images captured by binning factor of 2 under global motion which includes translation in 3D space and panning in a joint sinusoidal and circular moving trajectory.

Table 2.6: Comparison of different SR methods for 2× upscaling under a mixed Poisson–Gaussian noise with peak intensity 200, $\sigma = 2$ in PSNR (dB) and SSIM. Best: bold; second best: underline. (All TV-based methods were implemented using ADMM framework.)

	Cameraman		Lena		Page		Comic		Face		PPT3		Zebra		Average	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	26.68	0.6506	27.27	0.6934	22.80	0.5412	24.31	0.6921	31.12	0.7271	24.81	0.6115	27.04	0.7224	26.29	0.6626
L1+BTV [27]	29.15	0.8412	29.20	0.8210	23.75	0.7420	25.96	0.7914	33.40	0.8143	28.09	0.9058	29.77	0.8064	28.47	0.8174
L2+NLTV [101]	29.96	0.8591	29.41	0.8277	25.69	0.8133	26.40	0.8196	33.36	0.8080	30.35	0.9466	30.07	0.8024	29.32	0.8395
MPG+BTV [33]	29.97	0.8571	29.84	0.8410	24.70	0.7864	26.39	0.8077	33.81	0.8248	28.72	0.9336	30.45	0.8222	29.13	0.8390
L2+BSWTV	30.24	0.8622	29.75	0.8387	26.07	0.8240	26.72	0.8305	33.67	0.8172	30.60	0.9476	30.42	0.8202	29.64	0.8486
EDSR [17]	28.69	0.8404	28.84	0.8215	23.88	0.7560	24.97	0.7359	33.19	0.7956	27.75	0.9297	28.65	0.7683	28.00	0.8068
RBPN [40]	29.22	0.8593	29.53	0.8458	24.67	0.7898	25.99	0.7995	34.00	0.8292	28.17	0.9322	29.54	0.8246	28.73	0.8401
DPSR [22]	29.08	0.8404	28.72	0.8059	24.59	0.7218	25.18	0.7461	32.22	0.7652	28.53	0.9018	28.57	0.7729	28.13	0.7934
MPG+BSWTV	30.49	0.8706	29.99	0.8516	26.26	0.8401	27.00	0.8360	34.04	0.8303	31.03	0.9537	30.77	0.8465	29.94	0.8613

Following [58], a sliding window of size 5 centered at the 10th LR image was selected and other 14 SR methods [5, 13, 15, 16, 18, 27, 29, 32, 103–108] implemented in [58] were taken into comparison. Since the LR images are not severely contaminated by noise, λ was set as 0.1 for all the 14 scenes. The scalar α and the Gaussian noise parameter σ were estimated in Eq. (2.37) by [89] and the mean μ was assumed as 0. The estimated negative parameters were clamped to 10^{-6}. Besides, the decay scalar γ was set as 0.95 and the penalty parameter ρ was chosen as 10^3 for a smooth convergence over 16 iterations. The smoothing parameter η was set as 3 to make the flat regions and edges distinguishable in the weighting map. The shift parameter b was tuned as 1 so that the fine structures can be preserved. The original implementation and parameters in [58] were employed for the other 14 SR methods. The performance of the 15 SR methods is assessed by PSNR and SSIM as summarized in Fig 2.11. Single-frame and multi-frame SR methods are respectively marked by red and blue. It is shown that most of the multi-frame SR methods perform better than the single-frame ones under global motion. The proposed approach achieves considerable improvement comparing to the other methods in both PSNR and SSIM. In Fig. 2.12, the computation time of different approaches is demonstrated. It is necessary to note that all the other methods were implemented in Matlab and some of them were accelerated by C++. The proposed method was implemented in Python without C/C++ speedup. In Fig. 2.13 and Fig. 2.14, the reconstructed images of some representative methods were illustrated. As shown in Fig. 2.13, the proposed MPG+BSWTV generates more distinguishable characters and cleaner background. In Fig. 2.14, it is shown that the proposed method provides a more pleasant visual perception and resembles the GT image most.

In addition to the SupER dataset which contains 8-bit natural images, experiments on 16-bit X-ray images which were captured by the Nikon HMX ST 225 CT scanner were carried out. Two objects were taken as test specimens: a resolution target and a printed circuit board (PCB). Specially, four 16-bit X-ray images were captured by shifting the detector with a half pixel distance rightwards, downwards, and leftwards for both the specimens. The SR reconstructed images by different methods are demonstrated in Fig. 2.15 and Fig. 2.16. It is shown that the proposed method achieves better visual quality than the others by jointly sharpening the edges and suppressing the noise in the flat regions which coincides with the observations in the other SR experiments.

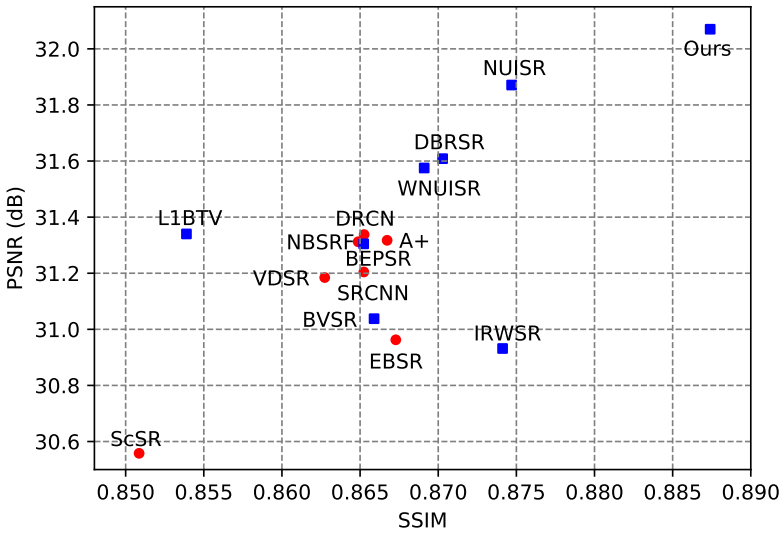


Figure 2.11: Comparison with other 14 SR methods on the SupER dataset [58] in average PSNR and SSIM for 2×. Red color map denotes the single-frame SR methods and the blue one represents the multi-frame SR methods.

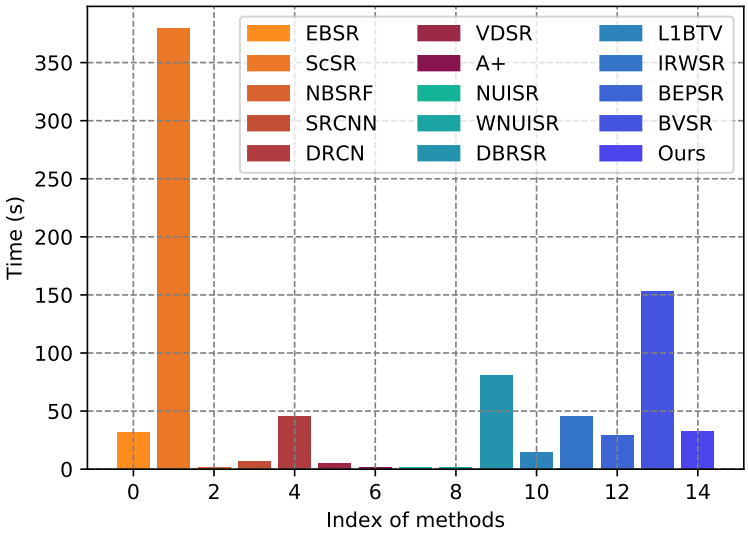


Figure 2.12: Comparison with other 14 SR methods on the SupER dataset [58] in runtime for 2×. Red color map denotes the single-frame SR methods and the blue one represents the multi-frame SR methods.

Figure 2.13: Comparison of different SR methods on the Coffee dataset (2×). Top: reconstructed SR images; Bottom: ROI

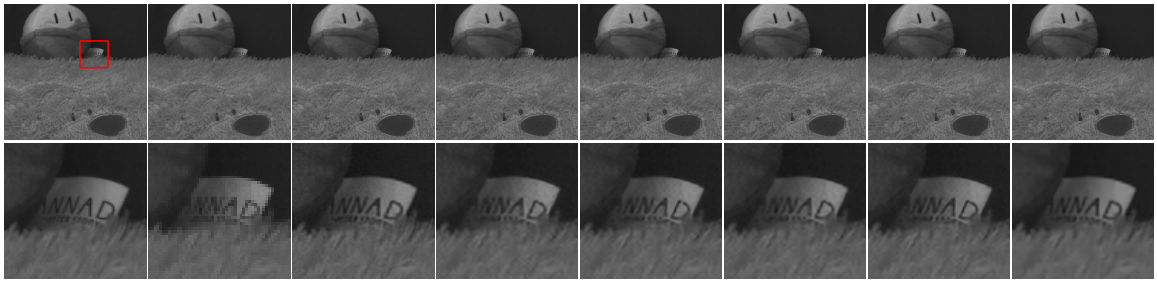


Figure 2.14: Comparison of different SR methods on the Dolls dataset (2×). Top: reconstructed SR images; Bottom: ROI

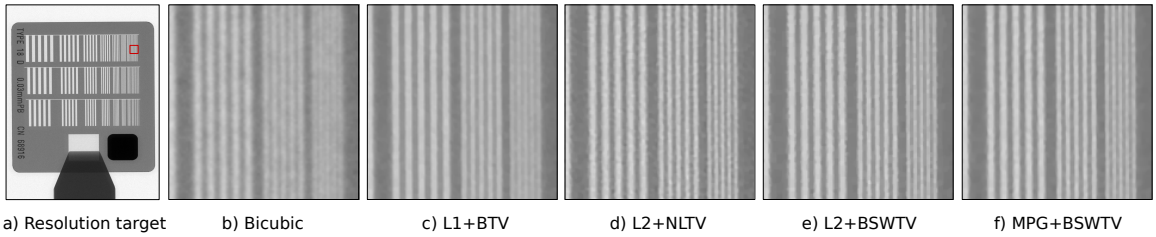


Figure 2.15: Comparison of different SR methods for 2× on the 16-bit X-ray image of a resolution target: (a) X-ray image of the resolution target, (b) bicubic, (c) L1+BTV, (d) L2+NLTV, (e) L2+BSWTV, and (f) MPG+BSWTV.

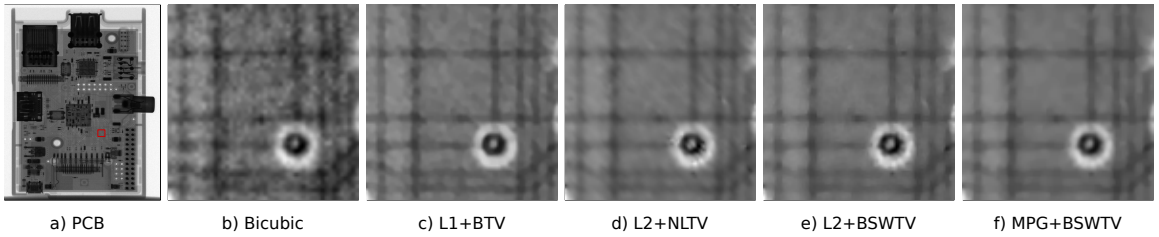


Figure 2.16: Comparison of different SR methods for 2× on the 16-bit X-ray image of a printed circuit board (PCB): (a) X-ray image of the PCB, (b) bicubic, (c) L1+BTV, (d) L2+NLTV, (e) L2+BSWTV, and (f) MPG+BSWTV.

Weighting Map and Parameter Analysis

In the following experiments, the effectiveness of the shrink coefficient on the weighting map Φ is demonstrated and the impact of the penalty parameter ρ, the decay scalar γ, the smoothing parameter η, and the shift parameter b on the performance of MPG+BSWTV is analyzed.

Refinement of weighting map: We demonstrate the refinement of the weighting map

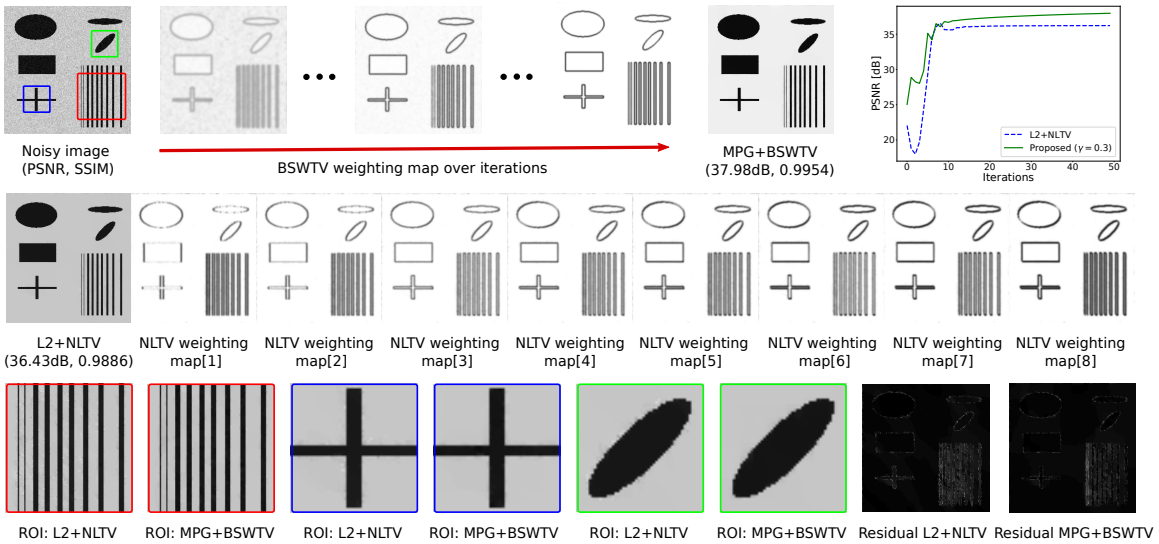


Figure 2.17: Illustration of the effectiveness of the shrink coefficient on the weighting map of BSWTV and the reconstructed image comparing to L2+NLTV by denoising an 8-bit gray-value image contaminated by a mixed Poisson–Gaussian noise with peak intensity 200 and $\sigma = 10$.

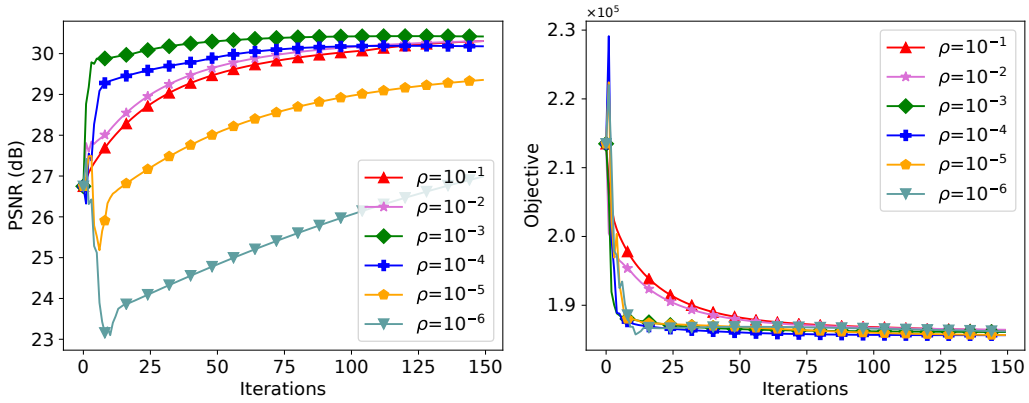


Figure 2.18: Impact of the initial penalty parameter ρ on the convergence. Left: PSNR over iterations; Right: objective over iterations.

in MPG+BSWTV over ADMM iterations and compare with the resultant weighting maps of L2+NLTV. The synthetic image contains multiple basic shapes including ellipses, rectangles, and bars as shown in Fig. 2.17. The image was contaminated by a mixed Poisson–Gaussian noise with peak intensity 200 and $\sigma = 10$. A search window of size $R = 3$ was chosen for NLTV and hence NLTV generates $R^2 - 1$ weighting maps. For a fast convergence, the decay parameter was set as $\gamma = 0.3$ and the momentum coefficient was selected as $\beta = 0.5$. The smoothing parameters η and the weighting parameters λ for both NLTV and BSWTV were tuned to achieve the best PSNR performance. As illustrated in Fig. 2.17, both approaches converge over iterations and the proposed method outperforms L2+NLTV quantitatively and qualitatively. It is shown that the mask of the edges in the weighting map of BSWTV becomes thinner and sharper along with the convergence. Consequently, the noise surrounding the edges is significantly suppressed.

Penalty parameter ρ: Experimental analysis has been conducted to study the influence of different initial ρ on the convergence of the algorithm. As shown in Fig 2.18, the magnitude of ρ has noticeable impact on the convergence rate although ρ is iteratively updated. Large ρ stabilizes the convergence and tends to slow down the convergence rate. On the contrary, small ρ accelerates the convergence but may cause overshoot of the objective function and lead to an undesirable image quality. Depending on the expected convergence rate and the noise level, an empirical choice of ρ may vary in a range of $[10^{-3}, 10^{3}]$ for 8-bit images.

Decay parameter γ: The decay parameter γ refines the weighting map Φ iteratively by

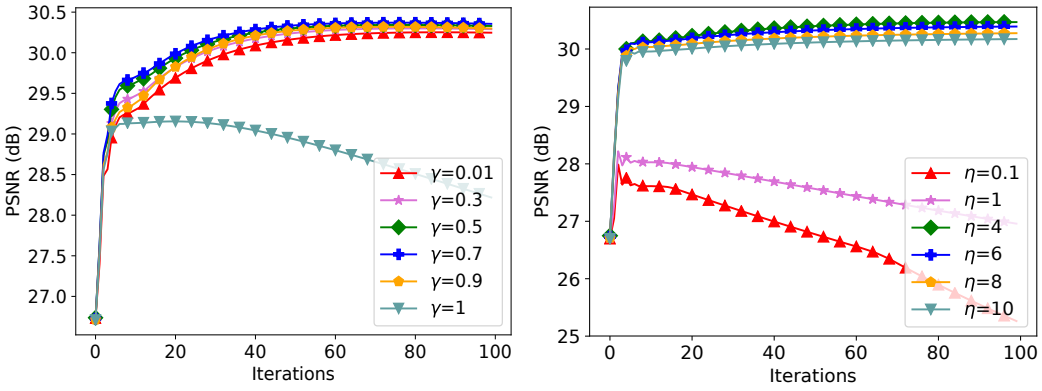


Figure 2.19: Left: impact of the decay scalar γ on the convergence; Right: impact of the smoothing parameter η on the convergence.

thinning the mask of the edges. We investigated γ in the range of $[0.01, 1]$ where $\gamma = 1$ indicates no decay. As shown in the left graph of Fig. 2.19, extremely small γ attenuates the shrink coefficient ξ aggressively so that the weighting map of regions containing fine low-contrast structures also gets whitened and the fine structures might be smoothed. In contrast, $\gamma = 1$ prevents the weighting map from whitening which limits the performance of the algorithm. To obtain gradual refinement of the weighting map, usually the decay parameter is chosen in a range of $[0.5, 0.95]$.

Smoothing parameter η: The smoothing parameter η is employed to control the impact of the eigenvalue discrepancy on the weighting map. As depicted in the right graph of Fig. 2.19, small η can not brighten the weighting map and results in a deteriorated performance. However, too large η causes saturation of the weighting map and the proposed regularization term acts as the standard TV. Depending on the dynamic range of the image, a proper choice of η for 8-bit images would be in the interval of $[2, 6]$.

Shift parameter b: The flat regions and strong edges can be easily tackled by a homogeneous shrink coefficient. The shift parameter b is introduced to deal with fine structures with relative low contrast. It behaves as a threshold and masks the fine textures in the weighting map by inhomogeneously shrinking the coefficient ξ. Specially, ξ is expected to be shrinked by γ in flat regions while maintain the same in fine-structured regions. Consequently, the fine structures are masked out in the weighting map and are not smoothed by TV. As illustrated in Fig. 2.20, b has a noticeable impact on SSIM and might have limited influence on PSNR because PSNR perfers slightly oversmoothed images.

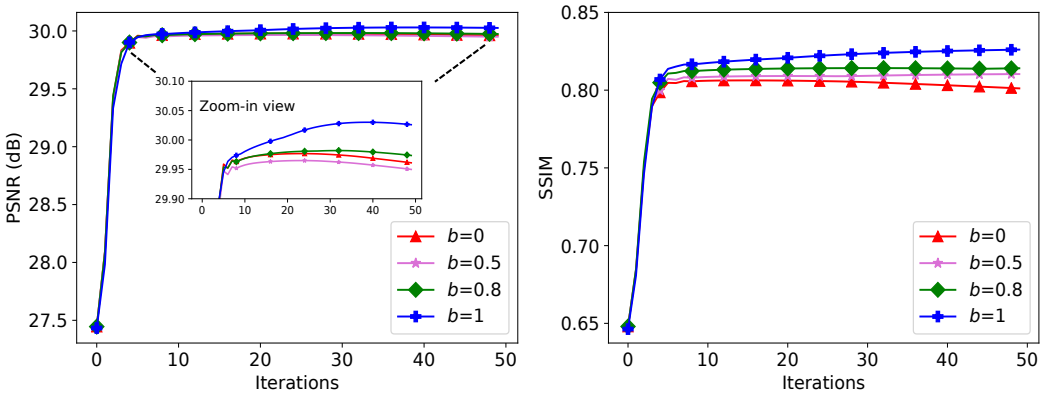


Figure 2.20: Impact of the shift parameter b on the convergence. Left: PSNR over iterations; Right: SSIM over iterations.

2.4 Conclusion

In this chapter, an objective function for noisy image super-resolution is proposed based on mixed Poisson–Gaussian noise model and the eigenvalues of the covariance matrix of the adaptively weighted image gradients. The proposed objective function is decomposed and solved based on the adapted alternating direction method of multipliers (ADMM) algorithm. Specially, the weighting map of the proposed regularizer is updated as the first step in the ADMM algorithm by considering the other variables as constants. In order to remove the outliers in the weighting map and facilitate the stability of the convergence of the objective function in the modified ADMM, the estimated weighting map is smoothed by a Gaussian filter with an iteratively decreased kernel width and updated in a momentum-based fashion. The proposed approach is benchmarked with other 14 state-of-the-art SR methods on the publicly available real-world dataset SupER. Experimental results demonstrate that the proposed MPG+BSWTV achieves an average gain of 0.2dB in PSNR and better visual perception compared to the investigated SR methods.

Chapter 3

Multi-GPU Accelerated Super-Resolution for Computed Tomography

In last chapter, an MISR algorithm based on the mixed Poisson–Gaussian noise model and bilateral spectrum weighted total variation was presented. Although it outperforms many well-known SR methods, the computational complexity might impede the deployment for real-time computing especially for Mpixel input images in practice. In this chapter, a multi-GPU framework for large-scale MISR named FL-MISR is proposed based on data parallelism, which is published in the work [35]. The presented FL-MISR has been seamlessly integrated into our CT system by super-resolving multiple projections of the same view acquired by subpixel detector shift. Since the SR reconstruction can be complete on the fly during the CT acquisition, FL-MISR achieves real-time performance. In section 3.4, experimental results are demonstrated in terms of resolution enhancement measured by MTF and computational speedup compared to the multi-core CPU implementation.

3.1 Previous Work of Fast MISR

In the literature, most of the optimization-based iterative MISR methods focus on the reconstruction accuracy and only few concern the performance in computation time [25,

27, 109, 110]. Specially, Elad et al. [25] propose a fast MISR algorithm concerning the special case of pure translation and space invariant blur. In [27], Farsiu et al. present a robust MISR method based on MAP using the L1-norm data fidelity term and the BTV regularization. Jens et al. [109] introduce a GPU-accelerated MISR approach for image-guided surgery which supports a 2× SR reconstruction from 4 LR images of size 200×200 in 60 *ms*. However, due to the GPU memory limit, their method can not handle large sized images. In [110], the authors propose a fast MISR method which composes registration, fusion, and sharpening for satellite images using high-order spline interpolation. Nevertheless, purely image fusion is performed on a GPU and the rest two steps are on the CPU which results in a degraded performance in runtime.

Comparing to the traditional iterative methods, CNN-based SR approaches focus on super-resolving single LR image by exploiting the relation learned exclusively from the LR-HR image pairs in the external example database. The learning-based MISR models are mainly proposed for video applications [36, 39, 40, 111]. Although some work is intended for real-time applications using GPU or FPGA [39, 60, 112], the video SR (VSR) performance highly relies on the fidelity of the synthesized LR-HR frame pairs and the quality of the training datasets. Furthermore, the supervised learning scheme requires the ground truth (GT) HR images during the training phase, the performance of the trained model will hence be limited by the available quality of the GT acquired in practice which is especially true for CT imaging due to the lack of publicly available high-quality HR datasets like DIV8K [113] for natural images.

To the best of our knowledge, the literature on GPU-accelerated MISR methods for large-scale images is very limited despite of its importance. In this chapter, a generalized accelerator for large-scale MISR based on data parallelism on multi-GPU systems is introduced. It is shown that the exchange of local variables and overlapped regions between neighboring GPUs has limited impact on the overall performance of runtime and leads to a consensus convergence over multi-GPUs without introducing border effects. Besides, it is shown that super-resolving four input images of size 4096×4096 by an upscaling of 2× can be achieved within 2.4*s* on a 4-GPU system.

3.2 Distributed Optimization Based on Data Parallelism

As described in Chapter 2, the SR model is usually presented as

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \tag{3.1}$$

with $\mathbf{x} \in \mathbb{R}^{n\times 1}, \mathbf{y} \in \mathbb{R}^{m\times 1}$ being respectively the latent and captured image rearranged in lexicographic order. To simplify the calculation, in this chapter $\mathbf{n}$ is assumed to be an intensity-independend additive noise and the system matrix $\mathbf{A}$ is known.

Assuming the noise $n_i \in \mathbf{n}$ in each pixel i is white Gaussian and i.i.d where $n_i \sim N(0,\sigma^2)$ and $P(n_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{n_i^2}{2\sigma^2}\right)$, the likelihood function is expressed as

$$P(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{m} P(y_i|\mathbf{x}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m \exp\left(-\frac{||\mathbf{A}\mathbf{x}-\mathbf{y}||_2^2}{2\sigma^2}\right) \tag{3.2}$$

Taking the natural logarithm, the associated negative log-likelihood can be formulated as

$$-log\left(P(\mathbf{y}|\mathbf{x})\right) = \frac{1}{2\sigma^2}||\mathbf{A}\mathbf{x}-\mathbf{y}||_2^2 + c \tag{3.3}$$

where c is a constant. For brevity, the weight $\frac{1}{2\sigma^2}$ and the constant c will be omitted in the latter formulation.

For MISR with k independent LR images $\mathbf{y}_i$ where $i \in [1\ldots k]$, the data fidelity term is hence formulated by

$$-log\left(\prod_{i=1}^{k} P(\mathbf{y}_i|\mathbf{x})\right) = \sum_{i=1}^{k} ||\mathbf{A}_i\mathbf{x}-\mathbf{y}_i||_2^2. \tag{3.4}$$

It should be noted that in case of additive white Laplacian noise which models the impulse noise (Salt & Pepper noise), it turns out to be the L1-norm data fidelity term [114]. Usually, L1-norm data term has better robustness against pixel outliers [27]. Without loss of generality, the data fidelity term can be formulated as

$$-log\left(\prod_{i=1}^{k} P(\mathbf{y}_i|\mathbf{x})\right) = \sum_{i=1}^{k} ||\mathbf{A}_i\mathbf{x}-\mathbf{y}_i||_p^p \tag{3.5}$$

with the L_p norm $1 \leq p \leq 2$.

Table 3.1: List of the consensus variables in SCG algorithm.

Param.	Description
f_c, f_{c_new}	consensus of the objective function
$\mathbf{p}_c$	consensus of the conjugate weight vector
$\mathbf{r}_c, \mathbf{r}_{c\_new}$	consensus of the steepest descent direction
σ_c, λ_c	consensus of the scalars
δ_c, μ_c	consensus of the variables in step size
α_c	consensus of the update step size

Aiming for reducing the computational complexity, BTV is leveraged as the image prior. Hence, the overall objective function based on the MAP framework is rewritten as following:

$$J(\mathbf{x}) = \sum_{i=1}^{k} ||\mathbf{A}_i\mathbf{x} - \mathbf{y}_i||_p^p + \lambda \sum_{\mathbf{d}} \gamma(\mathbf{d}) \parallel \mathbf{x} - S_{\mathbf{d}}\mathbf{x} \parallel_1 \tag{3.6}$$

where the scaling factor of the fidelity term $1/2\sigma^2$ in Eq. (3.3) is actually absorbed into the weighting parameter λ. In the experiment section 3.4, the L1-norm data term is used for a better robustness.

In order to circumvent the limitation of GPU memory resources and distribute the computational demand over multi-GPUs, data parallelism under a consensus-based convergence manner is performed to guarantee a centralized solution. The expected SR image $\mathbf{x}$ is obtained by data fusion. In particular, Eq. (3.6) can be rewritten as

$$J(\mathbf{x}) = \sum_{i=1}^{k} D_i(\mathbf{x}) + \lambda R(\mathbf{x}) \tag{3.7}$$

with D_i representing the corresponding data term and R indicating the regularization term. In this regard, the subfunction associated with the hth GPU can be expressed by

$$J_h(\mathbf{x}_h) = \sum_{i=1}^{k} D_i(\mathbf{x}_h) + \lambda R(\mathbf{x}_h), \quad s.t. \bigcup_{h=1}^{g} \mathbf{x}_h = \mathbf{x}, \tag{3.8}$$

where $\mathbf{x}_h$ is a fraction of the latent image $\mathbf{x}$ assigned to the hth GPU and g denotes the number of employed GPUs. To enforce the distributed optimization towards a centralized solution, we allow communication between the local GPU node and the host CPU for a consensus update decision. Specially, we utilize the SCG algorithm [115] to iteratively solve the subproblem described in Eq. (3.8) in each GPU. Instead of using the handcrafted

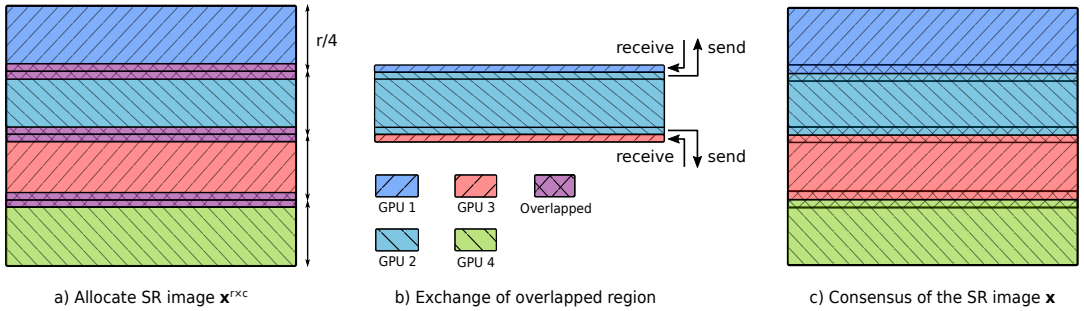


Figure 3.1: Demonstration of the exchange scheme of the overlapped regions for 4 GPU nodes.

step size or performing line search, SCG employs a step size scaling mechanism based on an adaptive scalar which achieves a faster and more robust convergence than the widely used approaches such as conjugate gradient with line search (CGL) and Broyden-Fletcher-Goldfarb-Shanno (BFGS).

Aiming for synchronizing the update of the individual $\mathbf{x}_h$ towards a centralized solution, we unify the local SCG scalar variables $\sigma, \lambda, \delta, \mu, \alpha$ by data communication. As these variables are calculated based on the inner product of vectors, we can obtain the consensus variables by the aggregate of the broadcast local ones. By means of consensus variables, the subfunctions can converge synchronically and a homogeneous resolution among multi-GPUs is guaranteed. In Table 3.1, we list the unified scalar variables and vectors (in bold) of SCG.

In addition, to avoid border discontinuity of neighboring partitions, region overlapping between neighboring GPUs is required. Instead of the naive averaging of the overlapped regions which sacrifices the sharpness and visual quality, we perform an inner-outer border exchange in each SCG iteration as shown in Fig. 3.1. A 4-GPU system is demonstrated and each GPU deals with the allocated image partition $\mathbf{x}_h$. The overlapped regions marked in violet are exchanged between neighboring GPUs. Particularly, since the inner borders can be correctly calculated only in case that the outer borders are consistent with the neighboring GPUs, the outer borders are replaced by the received ones and the inner borders are broadcast to the neighbors as exhibited in Fig. 3.1b). Consequently, an agreement in the overlapped regions is achieved as shown in Fig. 3.1c) without compromising the image sharpness. Without loss of generality, assuming g GPU nodes are employed, the architecture of the proposed multi-GPU framework for SR is illustrated in Fig. 3.2. The

local variables and overlapped regions are interchanged in each SCG iteration over the host CPU and updated in a consensus scheme.

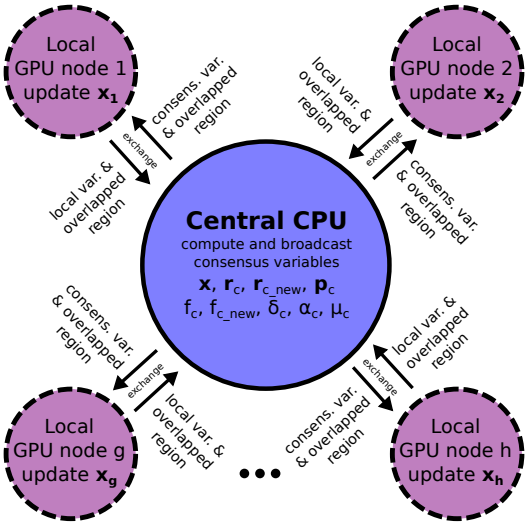


Figure 3.2: Architecture of the proposed multi-GPU framework for MISR where g GPU nodes are employed.

In Algorithm 3.1, a detailed description of the proposed distributed optimization framework is presented based on the SCG approach. The local GPU computation is marked by red and the centralized computation in the host CPU is denoted in blue. The local variables, overlapped regions, and consensus variables are exchanged after the local and central update. The SR image $\mathbf{x}$ is obtained when the predefined number of SCG iterations is complete.

In the implementation, the OpenCL framework was used. In order to optimize the data deployment on GPU memory, local memory was exploited in the kernel functions to the most extent. Sparse matrix was employed to calculate the system matrix $\mathbf{A}_i = D_i B_i M_i$ and the transpose $\mathbf{A}_i^T$ due to the sparseness of the downsampling, blurring, and motion matrices D_i, B_i, and M_i. Although memory transfer of the local variables and the overlapped regions between local GPU and host CPU is intended to hold the consensus convergence, transfer of large amounts of data is obviated during the SR reconstruction.

It is necessary to note that the proposed distributed optimization framework is based on data parallelism and SCG algorithm which is not confined to a specified objective function or a certain application.

Algorithm 3.1 Distributed SR Reconstruction

Partition and load LR images $\mathbf{y}_i, i \in [1 \dots k]$ into each GPU node $h \in [1 \dots g]$.
Calculate system matrix $\mathbf{A}_i, i \in [1 \dots k]$ of each GPU.
Initialize each GPU node with $\mathbf{A}_i, \gamma(\mathbf{d}), \lambda, f_h, f_c, \mathbf{p}_h, \mathbf{r}_h, \delta_c, \mu_c, \alpha_c, \sigma_h, n_{iter}$.
procedure Estimate latent image $\mathbf{x}$ according to Eqs. (3.6) and (3.8) using SCG [115]
    while $i_{iter} < n_{iter}$ do
        Local : Calculate $||\mathbf{p}_h||_2^2, h \in [1 \dots g]$.
        Central: Update $||\mathbf{p}_c||_2^2 = \sum_h^g ||\mathbf{p}_h||_2^2$.
        Local : Calculate $\mathbf{x}_{h\_tmp} = \mathbf{x}_h + \sigma_c \mathbf{p}_h$.
        Central: Exchange overlapped regions of $\mathbf{x}_{h\_tmp}$ with neighboring nodes.
        Local : Calculate δ_h according to SCG.
        Central: Update $\delta_c = \sum_h^g \delta_h$.
        Local : Calculate μ_h, α_h according to SCG.
        Central: Update $\mu_c = \sum_h^g \mu_h, \alpha_c = \sum_h^g \alpha_h$.
        Local : Calculate $\mathbf{x}_{h\_new} = \mathbf{x}_h + \alpha_c \mathbf{p}_h$.
        Central: Exchange overlapped regions of $\mathbf{x}_{h\_new}$ with neighboring nodes.
        Local : Calculate f_{h_new} according to Eq. (3.8).
        Central: Update $f_{c\_new} = \sum_h^g f_{h\_new}$.
        Local : Calculate $||\mathbf{r}_{h\_new}||_2^2$, inner product $\langle \mathbf{r}_h, \mathbf{r}_{h\_new} \rangle$.
        Central: Update $||\mathbf{r}_{c\_new}||_2^2 = \sum_h^g ||\mathbf{r}_{h\_new}||_2^2$, $\langle \mathbf{r}_c, \mathbf{r}_{c\_new} \rangle = \sum_h^g \langle \mathbf{r}_h, \mathbf{r}_{h\_new} \rangle$.
        Local : Update $\mathbf{p}_h$.
        Central: $i_{iter} = i_{iter} + 1$.
    end while
    end while
    Central: Fuse $\mathbf{x}$ with $\mathbf{x}_h, h \in [1 \dots g]$.
    return reconstructed image $\mathbf{x}$.
end procedure

3.3 Real-Time MISR Based on Subpixel Detector Shift

The proposed FL-MISR was applied on the Nikon HMX ST 225 CT scanner as shown in Fig. 2.5. The mechanism of enhancing the resolution by subpixel detector shift can be interpreted in Figure 3.3 and Figure 3.4. The basic idea is to displace the LR detector by subpixel level to increase the sampling rate and generate X-ray projections of the same amount of pixels as a HR detector.

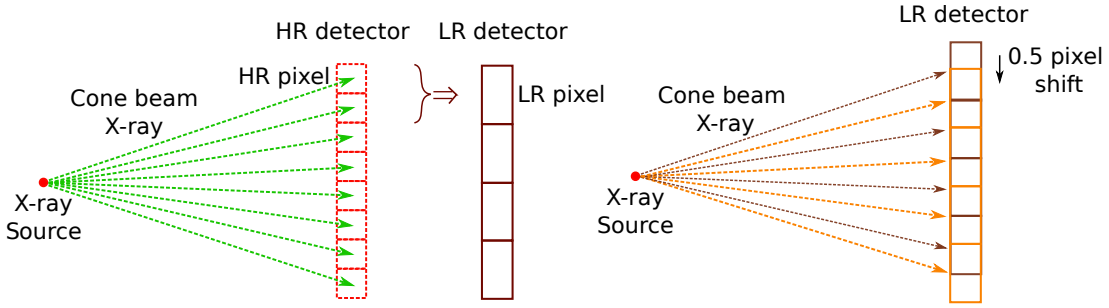


Figure 3.3: Schematic illustration of the resolution enhancement mechanism.

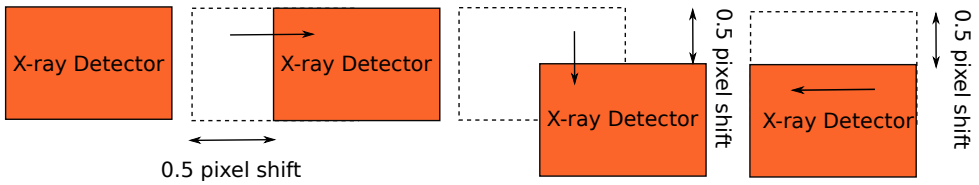


Figure 3.4: Realization of the detector shift by half a pixel.

During the CT acquisition, the object is rotated by 360° and at each rotation angle, four LR projections (X-ray images) are captured based on the detector shift rightwards, downwards, leftwards, and upwards by half a pixel as illustrated in Figs. 3.4 and 3.5. When four LR projections of the same view are collected, SR reconstruction is launched as denoted in green. The capture-reconstruct fashion repeats until the whole CT acquisition is accomplished. Since the four LR projections at different rotation angles have the same clockwise movement pattern, the system matrices A_i with $i \in [1,4]$ are calculated once at the beginning of the scan and shared by all the following projections. Due to the fact that SR reconstruction takes less time than the accumulated time of projection acquisition and table rotation as illustrated in Fig. 3.5, SR can be performed in real-time during the CT

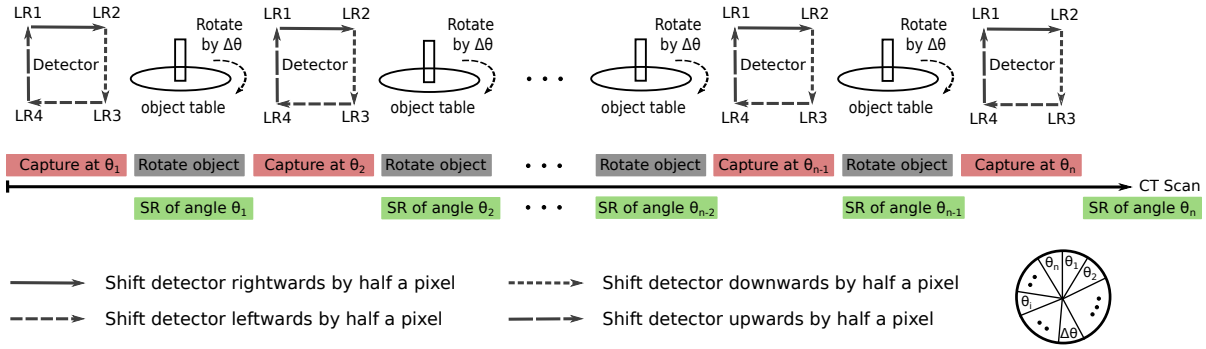


Figure 3.5: Schematic illustration of the application of FL-MISR in CT imaging based on the controlled subpixel detector shift.

acquisition without introducing additional computation time. Utilizing the super-resolved projections, the reconstructed CT volume yields an improved spatial resolution due to the increased detector sampling rate and the results are demonstrated in Section 3.4. It is worthy noting that for an upscaling of 3 or 4, 9 projections or 16 projections would be taken by shifting the detector by 1/3 or 1/4 pixel, respectively.

3.4 Experiments and Results

In this section, extensive experiments are conducted to evaluate the proposed FL-MISR for real-time SR in CT. The CT measurements were carried out on the Nikon HMX ST 225 CT scanner as shown in Fig. 2.5. The focal spot size of the tungsten X-ray tube is power dependent and for the power under 7 W which was utilized in the experiments, the effective focal spot size was measured as about 6 μm by the JIMA RT RC-04 micro chart. As the spatial resolution in CT systems depends on the magnification of the measurement (the ratio between the source-detector distance and the source-object distance), the effectiveness of FL-MISR was evaluated on the resolution enhancement of CT at magnifications of 5, 10, and 25.

The calculation of the system matrix $\mathbf{A}_i$ is thoroughly described in the previous work [33]. For an upscaling of 2× with half pixel detector shift and a 3×3 Gaussian blur B_i, a 12-row block area in the HR grid was set as the overlapped region between two neighboring GPUs. The weighting parameters λ and α were respectively set as 0.05 and 0.4 and the iteration number of SCG was selected as 20. In practice, larger λ should be opted in case

of strong noise and fewer SCG iterations, such as 5, could be used for fast CT acquisitions. The SR reconstruction was performed on a cluster of Nvidia GeForce GTX 1080 GPUs with 11GB of RAM for each and the Intel Xeon Gold 6148 CPU equipped with 56 Cores and 755GB memory. To quantify the resolution enhancement, the modulation transfer function (MTF) of the CT system was measured according to the standard ASTM-E 1695. Apart from the quantitative assessment, the QRM bar pattern phantoms were employed to visually verify the resolution improvement. Lastly, a practical case of the application of FL-MISR on a dry concrete joint is presented. The resolution-enhanced CT of the concrete joint would benefit the successive analysis such as evaluation of load capacity.

3.4.1 Evaluation of FL-MISR on Spatial Resolution Enhancement

CT scanner mainly consists of two components: the X-ray tube and X-ray sensitive detector. The spatial resolution of the CT system is primarily limited by the focal spot size of the X-ray tube and the detector pixel size. Usually, the spatial resolution of the imaging system is assessed by the MTF which is calculated as the normalized magnitude of the Fourier Transform of the point spread function (PSF). Nonstrictly speaking, MTF describes the smallest visually distinguishable line pairs per mm. The MTF of the CT system is formulated by $MTF_{sys} = MTF_{fs} \cdot MTF_{det} \cdot MTF_{others}$, where MTF_{fs} and MTF_{det} respectively denote the MTF of the focal spot and the detector. Other components such as the reconstruction algorithm, X-ray beam hardening, and display monitor are usually of less influence on the overall MTF_{sys}. In this work, subpixel detector shift is utilized to improve the MTF of the detector by increasing the sampling rate which will lead to an effective improvement of the MTF_{sys} when MTF_{det} dominates the MTF_{fs}, which is usually the case in many CT applications.

Evaluation on Synthetic CT Images

In order to analyze the effectiveness of subpixel detector shift on the spatial resolution enhancement of CT system, the impact of MTF_{det} on the MTF_{sys} is demonstrated. To simplify the system model, only the primary components are considered, namely $MTF_{sys} := MTF_{fs} \cdot MTF_{det}$. The MTF_{fs} is modeled by a Gaussian function and the MTF_{det} is represented by a $sinc$ function because of the assumed rectangular shape of the pixel.

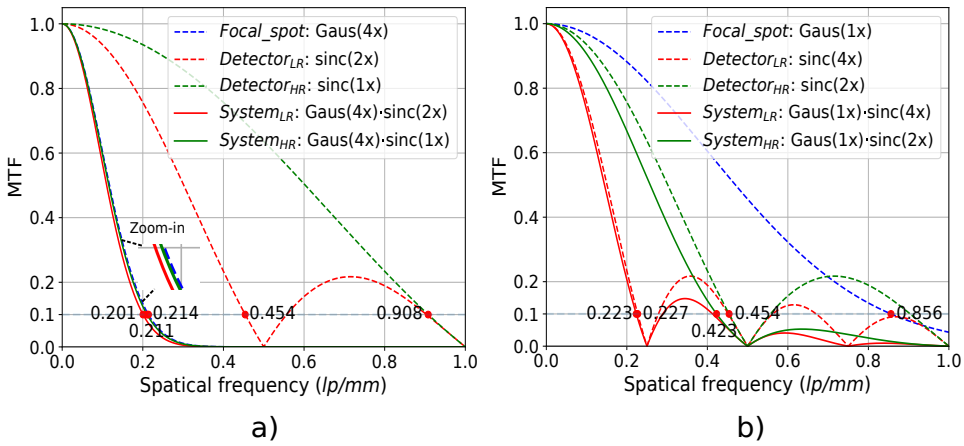


Figure 3.6: Influence of improved detector MTF on the system MTF based on one-dimensional synthetic analysis. a) when MTF_{fs} dominates, MTF_{sys} rarely improves; b) in case of MTF_{det} dominating, MTF_{sys} improves significantly.

As shown in Fig. 3.6, the left plot indicates the case where MTF_{fs} dominates MTF_{det}, for instance when the object is extremely close to the X-ray source and the right one depicts the situation where MTF_{det} dominates. The MTF of the detector with full pixel size and with half pixel size is respectively denoted as $Detector_{LR}$ and $Detector_{HR}$. Usually, the MTF at 10% is considered as the visible limit in practice and it is marked by gray dotted line. It is clearly shown that halving the detector pixel size doubles the MTF_{det} since the sampling rate is doubled by substituting $2x$ with x in the *sinc* function. Consequently, the overall MTF_{sys} is effectively improved when MTF_{det} dominates (right figure), while for the case MTF_{fs} dominates (left figure), MTF_{sys} has a negligible improvement.

Based on the analysis above, FL-MISR is evaluated on the CT images quantitatively and qualitatively. Specially, CT scans of an aluminium cylindrical phantom with a diameter of 20 *mm* were performed as shown in Fig. 2.5b) which was fixed perpendicular to the rotation table and a QRM bar pattern resolution phantom at the magnification of 10. Considering them as the ground truth (GT), four sets of 0.5× LR projections were simulated by shifting the GT projections rightwards, downwards, leftwards, and upwards by one pixel followed by a 2×2 binning. The downscaled LR projections were fused by interpolation and by FL-MISR. As the inter-image offset is assumed to be half a pixel and accurate, for interpolation-based fusion the pixel values of the LR images were inserted into the corresponding integer location in the HR grid. The super-resolved projections were then

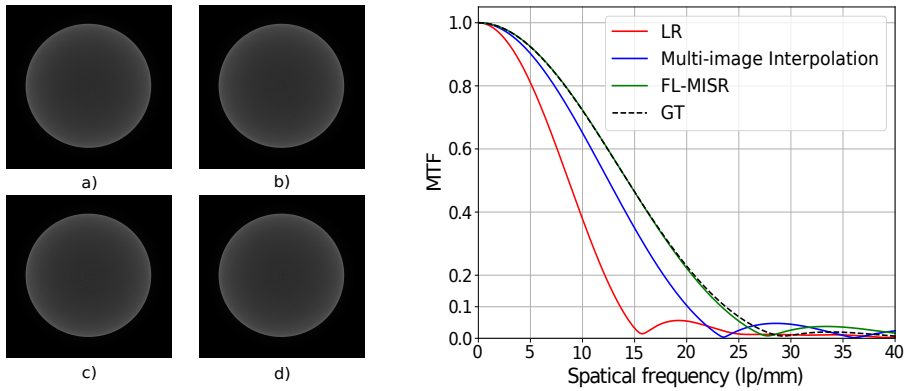


Figure 3.7: Evaluation of MTF on the CT cross section of an aluminium cylindrical phantom. Left: a) LR, b) multi-image interpolation, c) FL-MISR, d) GT; Right: MTF.

used for CT reconstruction by filter backprojection (FBP). The CT cross sections of the aluminium cylindrical phantom and the associated MTF are demonstrated in Fig. 3.7. The LR CT was reconstructed by the reference (upper left) set of the downscaled projections. As shown, the FL-MISR resembles the MTF of the GT extremely well and almost doubles the MTF of the LR image. For visual comparison, the LR images of the QRM target were generated following the same scenario as the aluminium cylindrical phantom and the CT images of the QRM bar pattern target are depicted in Fig. 3.8. It is shown that FL-MISR provides a more pleasant result with sharper structures and better visual quality.

Evaluation on Real-World CT Images

In this section, FL-MISR is evaluated on the real-world CT scans of multiple objects at different magnifications. Particularly, multiple CT measurements were conducted including the aluminium cylindrical phantoms with diameters of 10 mm and 20 mm, QRM bar pattern phantom with spatial resolution ranging from 3.3 lp/mm to 100 lp/mm, QRM bar pattern nano phantom covering resolution from 50 lp/mm to 500 lp/mm, and a cylindrical dry concrete joint with a diameter of 50 mm. The aluminium cylindrical phantoms and the QRM targets were both scanned at magnifications of 5 (voxel size of 25.4 μm), 10 (voxel size of 12.7 μm), and 25 (voxel size of 5.08 μm) and the concrete joint was acquired at magnifications of 3 (voxel size of 42.3 μm) and 5. The detailed measurement setup is listed

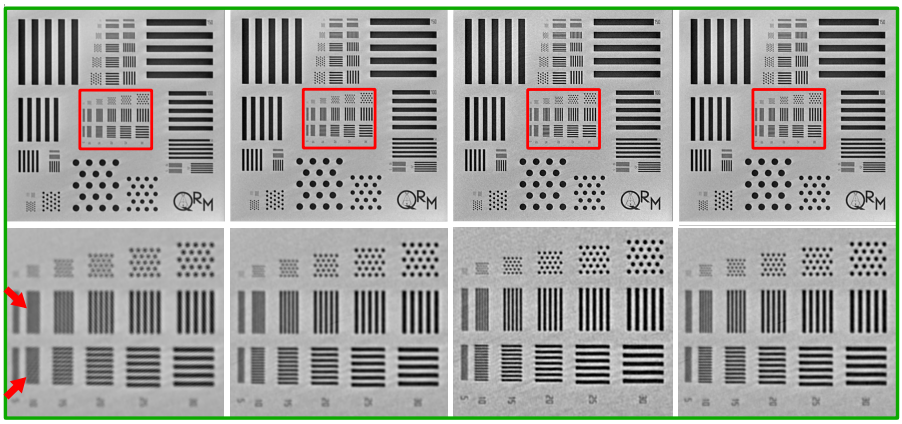


Figure 3.8: CT images of the QRM bar pattern phantom. The ROIs are marked by red rectangle and zoomed in. a) LR; b) multi-image interpolation; c) FL-MISR; d) GT.

in Table 3.2. As shown in Fig. 3.5, the X-ray detector was repeatedly displaced clockwise by half a pixel in a precisely controlled way. The projection at each detector position took 3 s, namely at each rotation angle 4×3 s was required for the acquisition. The object table rotated over 360° with 0.1 degree resolution following a stop-move manner and hence in total 4×3600 projections were taken. Besides, aluminium filters were utilized to absorb the soft X-ray beam and suppress the beam hardening artifact. FL-MISR is compared with the multi-image interpolation and with the standard CT without detector shift where the exposure time was set as 12 s which is the same as FL-MISR.

The MTF measured by the aluminium cylindrical phantoms at different magnifications is illustrated in Fig. 3.9. It it shown that FL-MISR performs significantly better than the standard CT at all the investigated magnifications covering voxel size from 25.4 μm to 5.08 μm. The multi-image interpolation behaves worse than FL-MISR as expected due to the naive manner of image fusion.

The CT images of the QRM bar pattern phantom and QRM bar pattern nano phantom

Table 3.2: Parameter setup for CT measurements.

Phantoms	Volt. (kV)	Curr. (μA)	# of Proj.	Exp. (s)	Det. Shift (px)	Filter (mm)
Al. Cylinder	200	34	3600	3	0.5	Al 2.5
QRM targets	80	86	3600	3	0.5	None

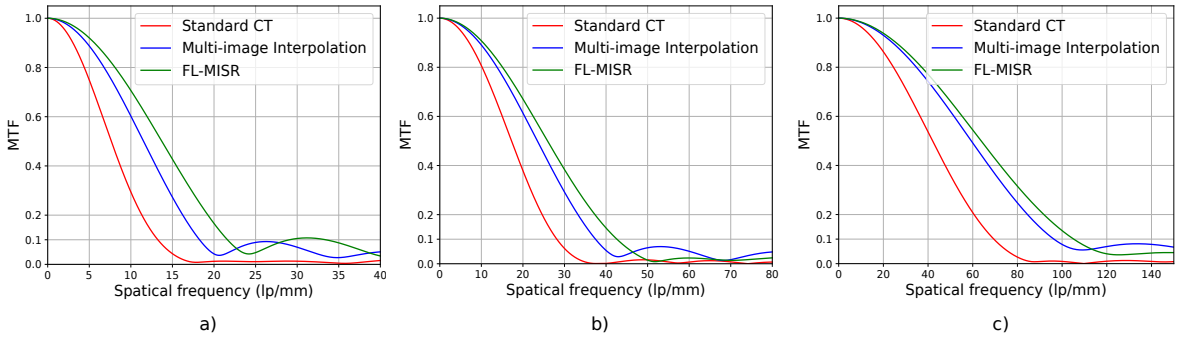


Figure 3.9: Evaluation of MTF at different magnifications. a) magnification of 5; b) magnification of 10; c) magnification of 25.

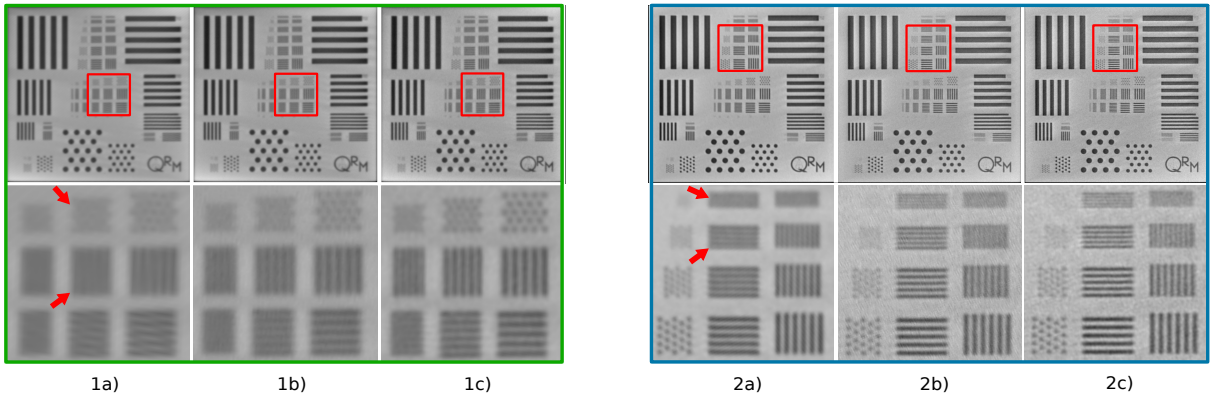


Figure 3.10: CT images of QRM bar pattern phantom. Left (marked in green): magnification of 5; Right (marked in blue): magnification of 10; a) standard CT without detector shift; b) multi-image interpolation; c) FL-MISR.

are illustrated in Fig. 3.10 and Fig. 3.11 with the corresponding closeup views. Comparing to the standard CT measurements, it is shown that FL-MISR and multi-image interpolation both improve the spatial resolution by exploiting the additional information captured via subpixel detector shift. However, multi-image interpolation is less robust than the optimization-based FL-MISR. It is shown that FL-MISR generates sharper edges than the multi-image interpolation and provides more pleasant results in visual perception. In fact, the spatial resolution estimated by the visibility of the QRM bar patterns coincides with the MTF measured by the aluminium cylindrical phantoms extremely well.

In Fig. 3.12, the CT images of a dry concrete joint are exhibited with the zoomed-in region

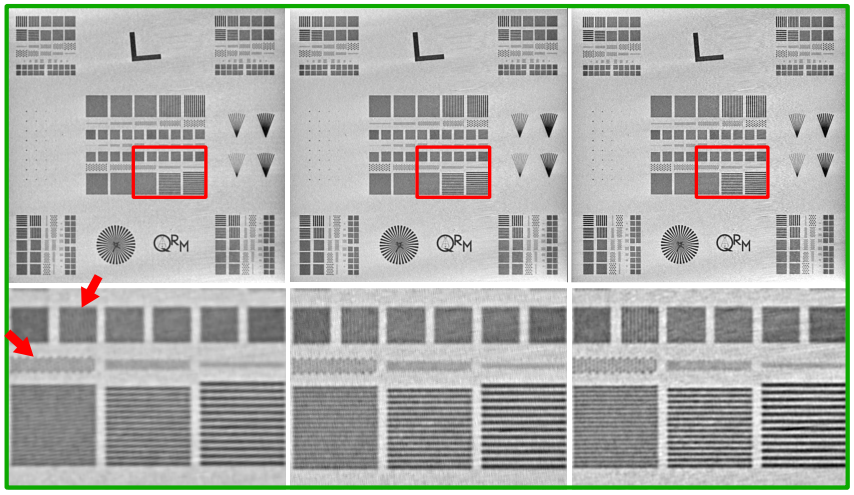


Figure 3.11: CT images of QRM bar pattern nano phantom at magnification of 25. a) standard CT without detector shift; b) multi-image interpolation; c) FL-MISR.

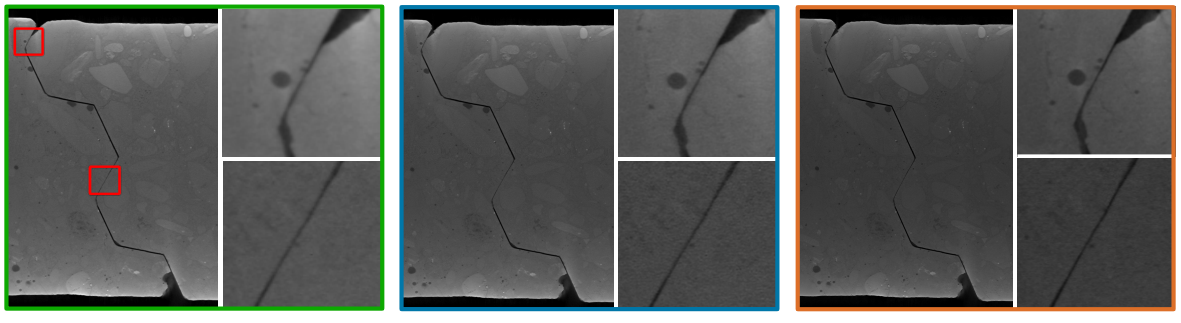


Figure 3.12: CT images of a dry concrete joint with the ROI in the closeup views. a) standard CT without detector shift at magnification of 3; b) FL-MISR with an upscaling of 2× at magnification of 3; c) standard CT without detector shift at magnification of 5.

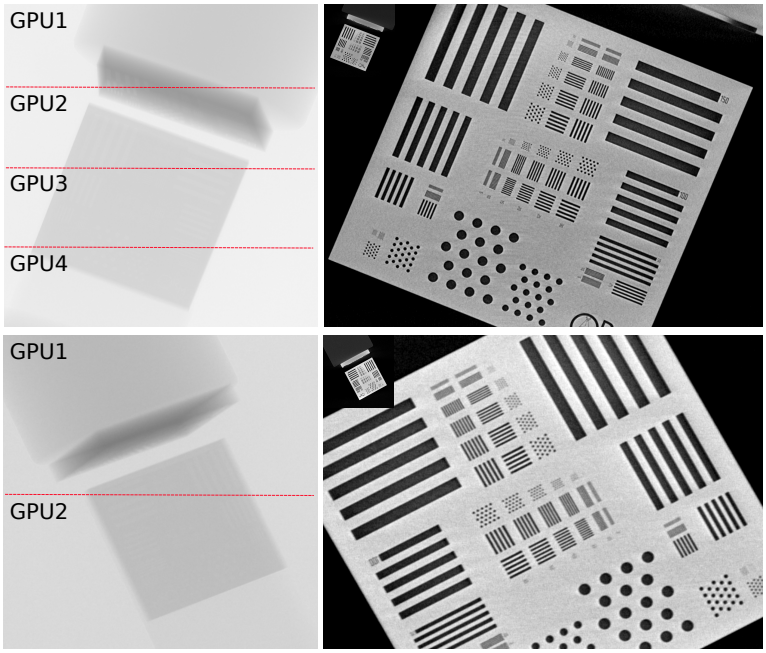


Figure 3.13: Evaluation on the border effect. First row: on the synthetic volume as utilized in Fig. 3.8; Second row: on the real-world volume as used in the middle graph of Fig. 3.10. Red dotted line marks out the border of the partitions allocated to the GPUs.

of interest (ROI) marked by red rectangles. Fig 3.12a) and Fig 3.12b) represent respectively the results of the standard CT without detector shift and FL-MISR at magnification of 3. Fig 3.12c) exhibits the results of standard CT at magnification of 5 which is considered as the reference image. It is shown that comparing to the standard CT with a voxel size of 42.3 μm at magnification of 3, FL-MISR generates sharper contours with more detailed structures which resembles the CT measurement at magnification of 5 with a voxel size of 25.4 μm.

Evaluation on Border Effect and Consensus Convergence

As explained in Fig. 3.1, we exchange the overlapped regions between neighboring GPUs to avoid border discontinuity. In Fig 3.13, we demonstrate the super-resolved projections and the associated CT images of the synthetic (top row) and the real-world measurements (bottom row). For the synthetic image, we employed four GPUs and for the real-world

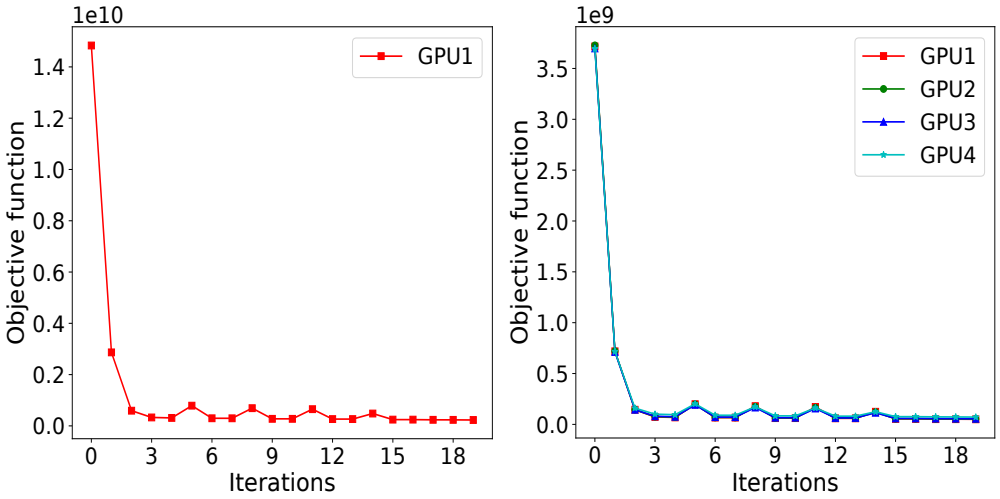


Figure 3.14: Evaluation on consensus convergence based on the objective function. Left: convergence curve obtained using single GPU; Right: convergence curves obtained using 4 GPUs.

one, two GPUs were in use. The individual $\mathbf{x}_h$ of each GPU is partitioned by the red dotted line. As we can observe that the overlapped regions, a 12-row block surrounding the borders (the red dotted lines), are of inherent sharpness without intensity discontinuity and the border effect is fundamentally obviated. Besides, in order to avoid inhomogeneous resolution in different partitions, we synchronize the update of the partitioned $\mathbf{x}_h$ among all the GPUs by exchanging the local variables of SCG. In Fig. 3.14, we illustrate the convergence curve of the centralized objective of Eq. 3.6 running on a single GPU and the distributed objective of Eq. 3.8 running on four GPUs. The consensus convergence is reflected in two aspects. First, the four GPUs have exactly the same convergence trend, where they are almost overlaid, due to the share of the SCG variables. Second, the distributed objective follows the same convergence trend as the centralized one and moreover, the sum of the four distributed objectives equals the centralized one by resorting to the scheme we adopt for the calculation of the consensus variables of SCG as described in Section 3.2. In addition, we can observe that the objective function is almost converged after 5 SCG iterations.

Table 3.3: Evaluation of computation time in terms of input image size, number of SCG iterations, and CPU/GPU platforms for the upscaling of 2× where four input images were utilized. (N/A indicates not applicable.)

Input image	SCG iterations	CPU* (s)	1 GPU (s)	4 GPU (s)
512×512	5	1.06	0.08	0.07
	10	2.24	0.13	0.12
	20	4.48	0.25	0.22
1024×1024	5	4.08	0.22	0.25
	10	8.64	0.42	0.44
	20	17.37	0.78	0.79
2048×2048	5	16.21	0.70	0.52
	10	34.89	1.30	0.76
	20	69.02	2.43	1.32
2300×3200	5	23.86	N/A	0.79
	10	50.68	N/A	1.20
	20	113.96	N/A	2.33
4096×4096	5	49.67	N/A	2.38
	10	105.71	N/A	3.02
	20	250.82	N/A	4.33

*CPU experiments were conducted on the Intel Xeon Gold 5120 CPU equipped with 56 cores.

3.4.2 Evaluation of FL-MISR on Acceleration

To demonstrate the performance of FL-MISR in computation acceleration, SR reconstructions of different sized inputs ranging from 512×512 to 4096×4096 for an upscaling factor of 2× were conducted on a multi-core CPU, single GPU, and multi-GPU systems. In particular, the CPU experiments were performed on the Intel Xeon Gold 5120 CPU which contains two nodes and each is equipped with 28 cores. The GPU experiments were carried out on the Nvidia GeForce GTX 1080 GPUs with 11GB memory. Since FL-MISR is based on the iterative SCG algorithm, the runtime is evaluated also with regard to the number of SCG iterations. The performance of different configurations was calculated based on an average of 100 runs and depicted in Table 3.3 where N/A indicates not applicable due to the large GPU memory footprint. As illustrated, comparing to the 56-core CPU variant, the single GPU implementation accelerates the computation by more than 25× for LR images of size 2048×2048 and the multi-GPU implementation which uses 4 GPUs achieves

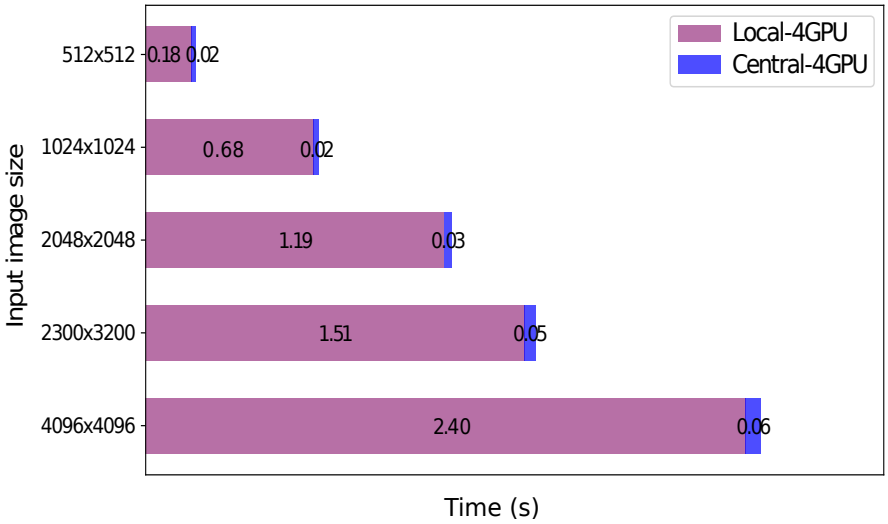


Figure 3.15: Runtime distribution for the local and centralized computation for super-resolving images of different sizes by an upscaling of 2× under 20 SCG iterations on 4 GPUs.

a speedup up to 50×. For large-scale images of size 2300×3200 and 4096×4096, FL-MISR running on 4 GPUs obtains a more than 55× speedup than the CPU implementation, while single GPU can not satisfy the memory demand. For small sized inputs like 512×512 and 1024×1024, single GPU implementation has similar performance as multi-GPU and achieves a 20× speedup comparing to the multi-CPU one.

Apart from the evaluation of the overall computation time, the runtime distribution for the local and centralized computation on a 4-GPU system is analyzed. It should be noted that the data communication time is aggregated into the centralized computation. The average runtime distribution over 100 runs for input images of different sizes is exhibited in Fig. 3.15. It is shown that the consumed time for consensus computing is almost negligible comparing to the local computation, while it is fundamentally necessary to avoid border effects between neighboring GPUs and guarantee a consensus convergence over multi-GPU systems.

3.5 Conclusion

In this chapter, a multi-GPU accelerated large-scale MISR approach based on data parallelism is presented. Specially, each GPU node accounts for a designated region of the latent HR image by applying the SCG algorithm to the distributed subproblem. The determinant variables of the local SCG algorithm are communicated and unified to impose a synchronized convergence rate among all the GPUs. The overlapped regions between neighboring GPUs are broadcast based on the inner-outer-border exchange mechanism to avoid border effects. The proposed FL-MISR is applied to our CT system by super-resolving projections captured via subpixel detector shift. The SR reconstruction is performed on the fly along with the CT acquisition so that no additional computation time is induced. Extensive experiments based on simulated data and real CT of various objects were conducted to quantitatively and qualitatively evaluate the proposed FL-MISR. Experimental results demonstrate that the spatial resolution of CT systems can be significantly improved by the application of FL-MISR. Furthermore, FL-MISR achieves a more than 50× speedup on a 4-GPU system in comparison to the multi-core CPU implementation. It is shown that the exchange of local variables and overlapped regions between neighboring GPUs has limited impact on the overall runtime.

Chapter 4

Super-Resolution for Image Registration

In previous chapters, two optimization-based MISR methods MPG+BSWTV and FL-MISR were presented from the aspects of performance in PSNR and performance in reconstruction speed, respectively. In this chapter, the application of SR on image registration is investigated and demonstrated. Specially, a novel deformable registration network FDRN [59] is proposed in Section 4.1. A light-weight resolution enhancement module (REM) based on residual CNN is introduced and evaluated in Section 4.2. REM is plugged in the registration network in a cascaded manner. The impact of REM on image registration is thoroughly evaluated on two registration networks FDRN and VoxelMorph at upscaling factors of 2× and 4× in Section 4.3.

4.1 FDRN: Fast Deformable Registration Network

4.1.1 Previous Work of Image Registration

Deformable image registration is an approach to establish dense spatial correspondence between a pair of digital images based on local structures. Deformable registration is widely applied to many medical applications such as detecting temporal anatomical changes of

individuals, analyzing variability across populations, and multi-modality fusion. In recent decades, several advancements in deformable image registration have been made [116–119]. Most of the existing conventional algorithms optimize an objective function J formulated as following:

$$J = D(F(\mathbf{X}), \phi \circ M(\mathbf{X})) + \lambda R(\phi). \tag{4.1}$$

The data term D measures the alignment between the fixed image F and the transformed moving image $\phi \circ M$ with ϕ being the dense deformation field which maps the coordinates of F to the coordinates of M and $\circ$ being the resampling operation. $\mathbf{X}$ denotes the 3D spatial coordinate in domain $\{\Omega \mid \Omega \subset \mathbb{R}^3\}$. The most commonly used data terms are, e.g., $L2$ error norm [120], mutual information [121], and cross-correlation [122]. As deformable registration is a highly ill-posed problem, regularization R is used to constrain the solution field. In general, the deformation field ϕ is modeled either by the displacement vector field (DVF) d(x) or the velocity vector field $v(\mathbf{X},t)$. The former category models the spatial transformation as a linear combination of the identity transform $\mathbf{X}$ and the DVF: $\phi = \mathbf{X} + d(\mathbf{X})$ with $d(\mathbf{X})$ being the DVF which represents the spatial offsets between the corresponding voxels in the fixed and the moving images. Particularly, [123–125] estimate the deformation based on the linear elastic models. In [126–128], the deformation field is described by cubic B-spline. Thirion proposes Demons [129] by introducing diffusion model in image registration. Generally, the DVF-based deformation model can not guarantee an inverse consistency, namely when interchanging the order of the two input images, the obtained transformation may not match the inverse of the counterpart. In contrast to the displacement-based vector field, the latter one concerns the invertibility of the transformation. Specially, deformable registration is considered as a variational problem and ϕ is formulated as an integral of a velocity vector field $v(\mathbf{X},t)$. Many variants have been proposed [120, 122, 130, 131] imposing biomedical constraints such as diffeomorphism, topology preservation, inverse consistency, and symmetry on the deformation field. In [120]. Beg et al. present the Large Deformation Diffeomorphic Metric Mapping (LDDMM) to solve a global variational problem in the space of smooth velocity vector field. Avants et al. [122] propose the symmetric normalization method (SyN) using cross-correlation as the similarity measure. However, due to the huge computational demand for volumetric medical images, tackling practical problems by conventional methods could be extremely slow.

In contrast to the traditional methods which adopt iterative updating scheme, learning-based methods are usually trained offline based on a large-scale dataset. As long as the

models are well-trained, predicting the transformation between unseen images performs solely forward propagation and consumes significantly less computation time. In other words, the learning-based methods transfer the burdens of computation to offline training and the well-trained model is dedicated to the specific application learned from the training dataset. Abundant learning-based studies on medical image registration have been conducted in the last two decades [132–138]. Particularly, Kim et al. [137] present a patch-based deformation model using sparse representation. In [136], Gutiérrez-Becker et al. formulate the prediction of the transformation parameters as supervised regression using the gradient boosted trees. More recently, deep learning has attracted increasing attention in the field of medical image registration due to the prominent capability of feature extraction [119]. Based on the supervision type, the existing deep learning models can be categorized into supervised, unsupervised/self-supervised, weakly supervised, deeply supervised, and dual supervised. Specially, [139–142] adopt deep convolutional neural network (CNN) in a supervised manner which require the ground truth of the deformation field during the training phase. However, supervised models usually suffer from the inaccuracy of the ground truth deformation field in the training datasets. In contrast to supervised learning, weak supervision employs higher-level correspondence information such as anatomical structural masks or landmark pairs which are more practical to obtain [143]. Some works [86, 144, 145] propose to optimize the similarity match between the fixed image and the transformed moving image in an unsupervised fashion by resorting to the spatial transformer network (STN) [146]. Specially, Balakrishnan et al. [145] introduce a CNN framework based on the UNet structure [147] adopting local cross-correlation as the similarity measure. In [86], Li et al. construct a multi-resolution registration model which contains three losses constraining the DVF at different spatial resolutions to maximize the similarity at different resolutions. In [148], the authors extend their previous work [145] by employing average Dice score of the segmented regions as the auxiliary loss. Fan et al. [139] present a modified UNet named BIRNet which uses multi-channel input and hierarchical loss based on dual supervision. Particularly, the loss consists of a supervised part which measures the deviation of the deformation field and an unsupervised part which drives the similarity match between the fixed and the warped moving image. However, due to the computational complexity, BIRNet requires 17.4s to register a pair of images of size $220 \times 220 \times 184$.

Autoencoder network has achieved promising performance in multiple medical image processing applications such as lesion detection, tumor segmentation, and image denoising.

By propagating the high-frequency information from the encoder path directly to the decoder path, autoencoder architecture is able to preserve local information and meanwhile obtain large receptive field. In this work, a fast deformable registration network FDRN is proposed based on the autoencoder backbone. One challenge of 3D image registration is the huge memory consumption which prohibits the deepening of the network. In order to address the issue, instead of using channel concatenation as VoxelMorph [148], an elementwise additive forwarding between the encoder and the decoder layer is utilized and the saved memory is exploited to deepen the network. In order to enhance the learning efficiency of the proposed deep model, deep supervision is leveraged at the bottom layer of smallest resolution to guide the convergence of the network and adopt skip connection in both encode and decoder stages to enable residual learning. Aiming for utilizing the available segmentation prior, a multi-label segmentation loss is proposed which improves the registration accuracy efficiently without inducing additional memory cost. Experiments show that the proposed FDRN achieves better performance than the investigated state-of-the-art registration methods for brain MR images. Although FDRN is evaluated on the brain MRI datasets, in fact, FDRN is a generalized registration model and is not limited to a particular type of image or anatomic structure.

4.1.2 Registration Method

The proposed FDRN is based on a compact encoder-decoder structure as demonstrated in Fig. 4.1. FDRN has a two-channel input which consists of a fixed and moving image pair and outputs a three-channel DVF. Based on the output $d(\mathbf{X})$, the moving image M is resampled at the transformed nonvoxel location $\phi(\mathbf{X}) = \mathbf{X} + d(\mathbf{X})$ and a similarity match between the fixed image F and the transformed moving image $\phi \circ M$ is measured and optimized. Therefore, the registration network can be trained in an unsupervised scheme regardless of the ground-truth DVF. Particularly, the encoder path extracts the features at different resolutions and meanwhile enlarges the receptive field by convolutions with stride 2. Each convolution is followed by the instance normalization and PreLU. Skip connection is utilized at each encoder and decoder stage to enable residual learning and prevent from gradient vanishing. Besides, due to the fact that the registration of low-resolution (LR) images is easier to learn, an auxiliary loss is involved at the bottom of the encoder path to punish the misalignment of the LR image pairs. In addition, the weight of the LR loss is gradually decayed along with the training and the model is fine tuned in the end fully

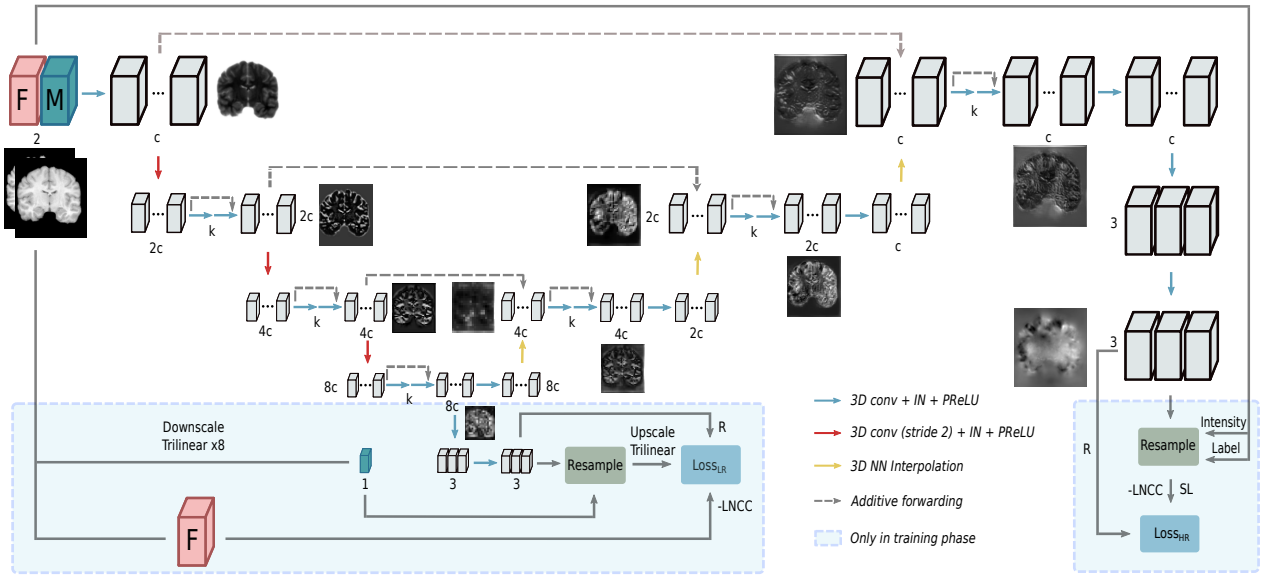


Figure 4.1: Schematic illustration of the structure of the proposed FDRN. Some feature maps are demonstrated beside the layers. Variable c depicts the amount of channels in the first layer and k represents the number of convolutions at each encoder-decoder stage. For the baseline model, $c = 8, k = 1$ and for FDRN, $c = 16, k = 2$.

based on the main loss in high-resolution (HR) grid such that the LR auxiliary loss guides the convergence of the network in the early training phase which imitates the coarse-to-fine registration strategy applied in the conventional multi-resolution registration methods. On the decoder side, the extracted features are fused and the feature maps are enlarged to restore the original image dimension. To circumvent the checkerboard artifacts induced by deconvolution, nearest neighbor (NN) interpolation is combined with 3D convolution as the upsampling operation. In order to save memory and preserve the high-frequency features, additive forwarding is performed from the encoder path to the corresponding decoder path. Last but not least, a multi-label segmentation loss (SL) is introduced to further boost the performance of the registration. Comparing to the average Dice score adopted in [148], the proposed SL does not require additional memory during the training regardless of the number of classes.

For the sake of clarity, different structure variants of the proposed architecture are indicated by c - k in the latter formulation where c represents the amount of channels in the first

layer and k denotes the number of convolutions at each encoder and decoder stage. Aggregating the loss functions of different resolutions, the overall loss is formulated as

$$L_{overall} = (1-\lambda)L_{HR} + \lambda L_{LR}. \tag{4.2}$$

The parameter λ is exponentially decreased from 0.5 to 0 so that the HR loss dominates the learning gradually in the training phase. The L_{HR} and L_{LR} are defined as

$$\begin{aligned} L_{HR} &= -D(F_{HR}(\mathbf{X}), \phi_{HR} \circ M_{HR}(\mathbf{X})) + \alpha_1 R(\phi_{HR}) + \alpha_2 SL(\phi_{HR}), \\ L_{LR} &= -D(F_{HR}(\mathbf{X}), [\phi_{LR} \circ M_{LR}(\mathbf{X})]_{\uparrow}) + \alpha_3 R(\phi_{LR}), \end{aligned} \tag{4.3}$$

where D denotes the data term, ϕ_{HR} and ϕ_{LR} indicate respectively the deformation field in the HR and LR grid. Inspired by [148], local normalized cross-correlation (LNCC) is adopted as expressed in Eq. (4.4) to quantify the similarity measure. Comparing to normalized cross correlation (NCC), it turns out that LNCC converges faster and better for large training patchsize. $[\cdot]_{\uparrow}$ represents the upscaling operation and in this work, trilinear interpolation is in use. α_1, α_3 are the weights of the regularization terms and α_2 indicates the weighting parameter of the segmentation loss.

$$D(F, \phi \circ M) = \sum_{\mathbf{X}_i}^{\Omega} \frac{\left(\sum_{\mathbf{X}}^{\Omega_i} (F(\mathbf{X}) - \bar{F}_{\Omega_i})(\phi \circ M(\mathbf{X}) - \overline{\phi \circ M}_{\Omega_i})\right)^2}{(\sum_{\mathbf{X}}^{\Omega_i} (F(\mathbf{X}) - \bar{F}_{\Omega_i})^2)(\sum_{\mathbf{X}}^{\Omega_i} (\phi \circ M(\mathbf{X}) - \overline{\phi \circ M}_{\Omega_i})^2)} \tag{4.4}$$

In the formulation of LNCC by Eq. (4.4), $\bar{F}_{\Omega_i}$ denotes the mean of the local region Ω_i centered at voxel $\mathbf{X}_i$ with size of n^3. $\overline{\phi \circ M}_{\Omega_i}$ represents the mean of the corresponding region in the transformed moving image. In this work, the window size is chosen as $n = 9$.

The regularization R imposes smoothness on the deformation field ϕ. As ϕ is a linear combination of the identity transform $\mathbf{X}$ and the expected DVF, the constraint is directly applied on the DVF $d(\mathbf{X})$ by

$$R(d(\mathbf{X})) = \sum_{S_k} ||d(\mathbf{X}) - S_k d(\mathbf{X})||_2^2, \tag{4.5}$$

where S_k indicates the shifting operator along (u, v, w) direction by vector k with $k = \{(u, v, w) \mid u, v, w \in \{0, 1\}\}$ and $||\cdot||_2^2$ represents the L2-norm. Comparing to the L1-norm

total variation (TV), L2-norm is differentiable at 0 and leads to a considerable faster convergence.

The segmentation loss SL serves to punish the misalignment between the labels of the fixed image L_F and the transformed moving image $\phi \circ L_M$. In [148], average Dice score (ADS) is adopted to regularize the DVF based on the segmentation labels. In fact, multi-label dice score is originally used for segmentation applications. Acting as the regularization for volumetric deformable registration, ADS has two major drawbacks. Firstly, the leverage of ADS for large 3D image induces noticeable additional memory cost during the training which prohibits the deepening of the network. Secondly, the memory consumed by ADS increases linearly with the number of segmentation classes. For instance, a label volume of size $160 \times 208 \times 176$ with 56 classes as brain MRI dataset LPBA40 requires 1.3GB during training. In order to tackle this issue, a multi-label SL is proposed as

$$SL(\phi) = \frac{(c_1+1)|L_F - \phi \circ L_M|_1}{|L_F|_1 + |\phi \circ L_M|_1 + c_1|L_F - \phi \circ L_M|_1 + c_2}, \tag{4.6}$$

where L_F and L_M represent respectively the labels of the fixed image and the moving image. c_1 and c_2 are nonnegative constants. c_1 weights the punishment of the inconsistency between L_F and $\phi \circ L_M$ and c_2 serves to prevent zero division. Comparing to ADS, the proposed segmentation loss does not require extra memory but depends on the value of the individual label.

For the transformation of the moving image $\phi \circ M(\mathbf{X})$, trilinear interpolation of $M(\mathbf{X})$ is performed at the transformed nonvoxel locations ϕ. Mathematically, the resampling of $M(\mathbf{X})$ at the transformed location ϕ is formulated as

$$\phi \circ M(\mathbf{X}) = \sum_{\mathbf{n} \in \mathbb{N}(\phi(\mathbf{X}))} M(\mathbf{n}) \prod_{m \in \{u,v,w\}} (1 - |\phi_m(\mathbf{X}) - \mathbf{n}_m|), \tag{4.7}$$

where $\mathbb{N}(\phi(\mathbf{X}))$ denotes the coordinates of the neighbors of the transformed nonvoxel location in the moving image $M(\mathbf{X})$ and $\mathbf{n}$ represents the coordinates of the individual neighbor. m is an indicator and iterates over the dimensions of the moving image.

4.1.3 Experiments and Results

In this section, the proposed FDRN is evaluated from different aspects. Particularly, in Section 4.1.3 FDRN is trained on the LONI LPBA40 dataset [149] which contains T1-weighted brain MR images. FDRN is compared with the state-of-the-art registration methods including the traditional method symmetric image normalization SyN [122], the deep learning-based methods Li et al. [150], and VoxelMorph [145]. In Section 4.1.3, the well-trained FDRN is evaluated on other unseen MRI datasets including CUMC12 [151], MGH10 [151], ABIDE [152] and ADNI [153]. Section 4.1.3 consists of the model analysis.

Datasets and Preprocessing

The LPBA40 dataset contains brain MR images of 40 neurologically intact nonepileptic subjects with segmentation labels for 56 brain regions. All of these MR images were firstly registered to the Montreal Neurological Institute (MNI) space using affine transformation based on the ICBM152 template [154] as preprocessing. The registered images were then cropped to the size of $160 \times 208 \times 176$. The 40 cropped images were partitioned into 30, 4, and 6 for training, validation, and testing, respectively. In addition, experiments were conducted on the unseen CUMC12, MGH10, ABIDE and ADNI MRI datasets to evaluate the generalizability of the well-trained FDRN. Specially, CUMC12 contains 12 MR images with 128 labeled regions and MGH10 consists of 10 subjects with 74 segmented regions. 10 random images were individually selected from ABIDE and ADNI which do not contain segmentation labels. The experimental results are demonstrated in Section 4.1.3 and Section 4.1.3.

Evaluation Metrics

Apart from NCC which measures the cross correlation based on intensity values, the registration performance is evaluated using Dice score [155] to quantify the overlap of labels for each segmented region by

$$Dice(A,B) = \frac{2|A \cap B|_1}{|A|_1 + |B|_1}, \tag{4.8}$$

where A and B indicate binary images which represent the individual label in the fixed and moving image, respectively. Furthermore, the perception-based metric, structural similarity index measure (SSIM) [156], is adopted to aid the assessment in visual quality in terms of luminance, contrast and structure defined by

$$SSIM(A,B) = \frac{(2\mu_A\mu_B + c_1)(2\sigma_{AB} + c_2)}{(\mu_A^2 + \mu_B^2 + c_1)(\sigma_A^2 + \sigma_B^2 + c_2)}, \tag{4.9}$$

where $\mu_A, \mu_B, \sigma_A, \sigma_B$ denote the mean and standard deviation of the image A and B. σ_{AB} indicates the covariance between image A and B. c_1, c_2 are small positive constants to stabilize the division with weak denominator. An SSIM of 1 indicates a perfect anatomical match.

Implementation Details

FDRN was implemented with a Pytorch backend. As FDRN is trained in an unsupervised manner, in order to ensure the existence of the corresponding voxels in the input pair especially at the image boarders, the network is fed with the whole image of size $160 \times 208 \times 176$. Due to the memory limitation, a mini-batch of size 1 was used. Adam with $\beta_1 = 0.9, \beta_2 = 0.999$ was used as the optimizer. The initial learning rate was set as 0.002 and multiplied by 0.9 every 1000 iterations until decreased to 0.0001 over 70 epochs. The weighting parameters α_1, α_3 for the regularization R were set as $\alpha_1 = 1 \times 10^{-8}, \alpha_3 = 8\alpha_1$. The weight λ of L_{LR} was implemented as $0.5^{(1+i/1000)}$ with i being the index of the iteration and 0.5 as the initial weight. The weight of the segmentation loss SL was tuned as $\alpha_2 = 0.2$ and the parameters in SL were set as $c_1 = 10, c_2 = 10^{-9}$. A detailed analysis of the hyperparameters α_2 and c_1 is carried out in Section 4.1.3. It is worthy noting that the above-mentioned hyperparameters were tuned in a trial-and-error manner on the validation dataset and the ones generating the best Dice score were selected. The experiments were performed on the NVIDIA GeForce GTX 1080 Ti with 11GB GDDR5X and the Intel(R) Xeon(R) E5-2650 v2 CPU.

Evaluation on LPBA40 Dataset

A comparison with the state-of-the-art deformable registration methods SyN [122], Li et al. [86], and VoxelMorph [148] on the public LPBA40 dataset was conducted. Li's model

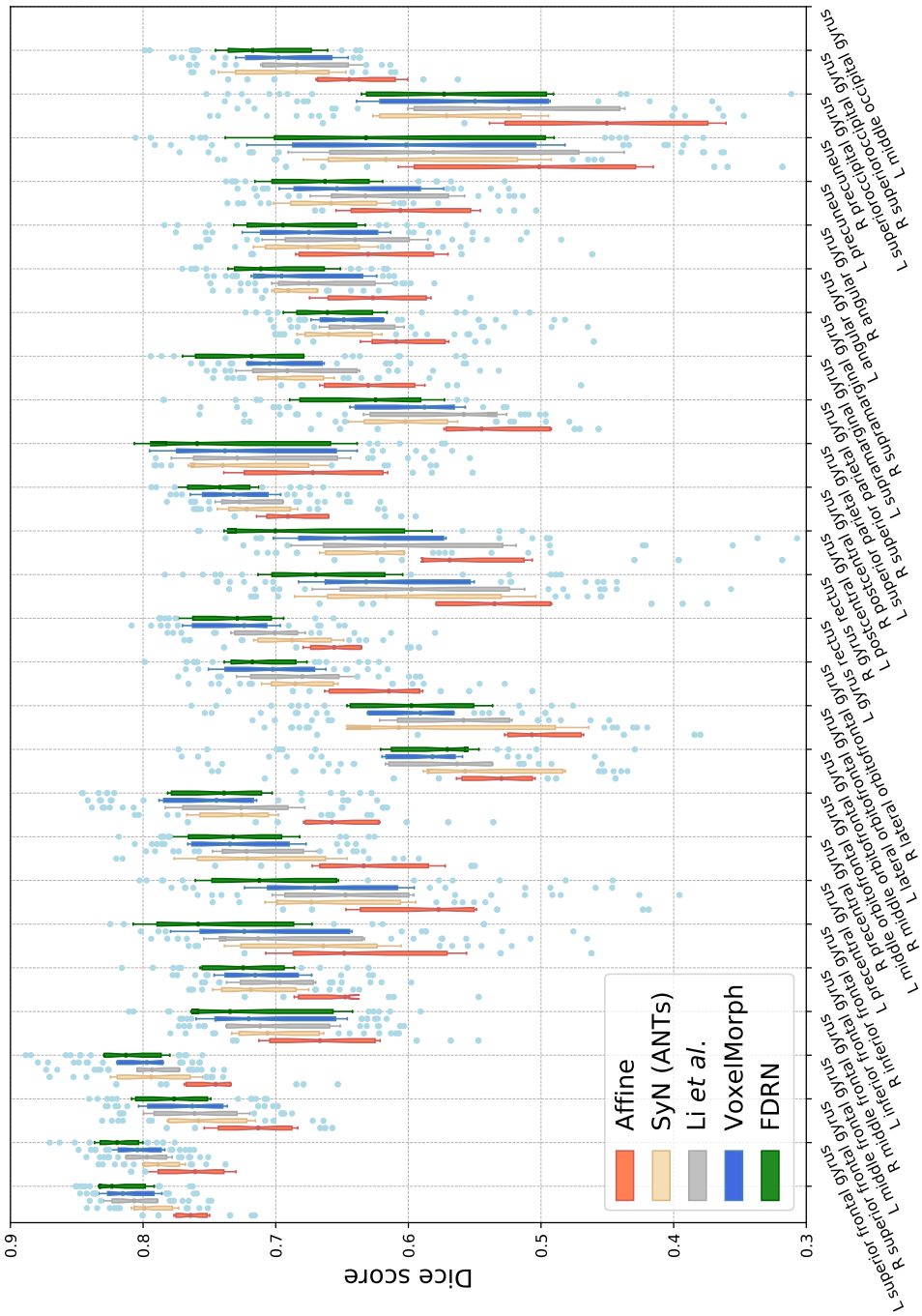


Figure 4.2: Boxplots of the average Dice scores of 54 labeled anatomical regions for the 30 testing image pairs from the publicly available LPBA40 brain MRI dataset: Part I.

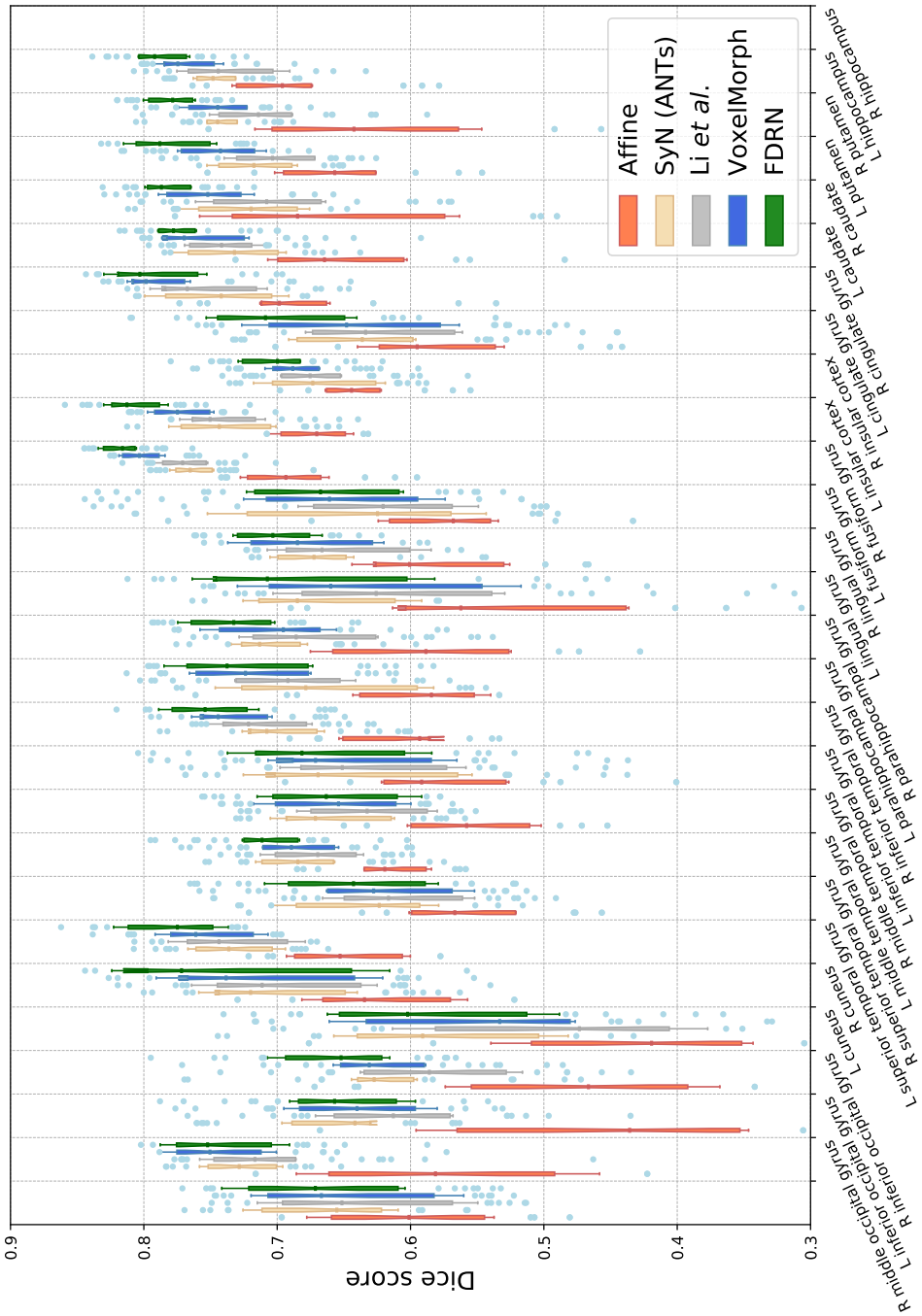


Figure 4.3: Boxplots of the average Dice scores of 54 labeled anatomical regions for the 30 testing image pairs from the publicly available LPBA40 brain MRI dataset: Part II.

and VoxelMorph were reproduced and trained in Pytorch according to their original paper. Specially, for Li's method, the full image was used as the input with a mini-batch of one and tuned the weight of the TV regularization as $\lambda = 1 \times 10^{-9}$ for best Dice performance. With regard to VoxelMorph, NLCC was utilized as the main loss, L2-norm smoothness as the regularization, and average Dice score as the auxiliary loss. The weight of the Dice loss was tuned as 0.1 and the model was trained over 70 epochs. ANTsPy, the Python wrapper for the Advanced Normalization Tools (ANTs) [157], was employed to implement SyN. Particularly, cross correlation (CC) was adopted as the similarity measure and instead of using the default iterations $(40, 20, 0)$, the iterations were set as $(100, 40, 10)$. To achieve a better accuracy, the sampling bins were set as 60 instead of the default 32. 30 MR images were used as the training data and the remaining 10 images were utilized for validation and testing. Every permutation of pairs out of the 30 images (total of 870 permutations) was used as the input during the training. In the testing phase, each of the 6 images was chosen as the fixed image and the rest 5 images were registered to it (total of 30 pairs of images). The performance of SyN, Li's method, VoxelMorph, the baseline model, and FDRN was quantified using Dice score, NCC, and runtime as summarized in Table 4.1. Particularly, VoxelMorph with and without the ADS loss were used in comparison. As depicted, all the CNN-based methods perform hundreds times faster than the traditional SyN. Comparing to VoxelMorph, the baseline model obtains comparable Dice and nearly halves the inference time by discarding the channel concatenation. FDRN extends the baseline model and improves the Dice and NCC efficiently by enlarging the network capacity. Comparing to VoxelMorph with ADS, 16-2 consumes similar training memory (about 10.8GB) and inference time and achieves a performance gain of 1.46% in Dice score. It is worth noting that the runtime is accumulated purely for the registration step without concerning the preprocessing and image loading. Additionally, the Dice score of each anatomical structure labeled in LPBA40 is depicted in the boxplots in Fig. 4.2 and Fig. 4.3. It is shown that FDRN performs best among the investigated methods for nearly all the labeled regions.

In order to evaluate FDRN visually, one image from the 6 testing images in LPBA40 was selected as the fixed image and the remaining 5 images were registered to it. The average of the 5 registered images was illustrated along with the transformed labels, the mean Dice score, NCC and SSIM of different methods in Fig. 4.4. In addition, to visualize the individual registration performance, the deformation field and the corresponding DVF of an image pair were demonstrated for Li's method, VoxelMorph, and FDRN. It is shown that

Table 4.1: Comparison of different registration methods on the testing images with size of 160×208×176 in the LPBA40 MRI dataset by average Dice score, NCC, and runtime. ADS: Average Dice score; SL: Segmentation loss. Best results are in bold.

	Affine	SyN [122]	Li et al. [86]	VoxelMorph [145]		Baseline(8-1)		FDRN(16-2)	
–	–	–	–	w/o ADS	w/ ADS	w/o SL	w/ SL	w/o SL	w/ SL
Dice	0.6079	0.6805	0.6689	0.6746	0.6897	0.6764	0.6898	0.6882	0.7043
NCC	0.9506	0.9876	0.9962	0.9973	0.9971	0.9969	0.9966	0.9978	0.9975
GPU/CPU(s)	–/–	–/5658.19	0.37/24.06	0.26/18.02		0.14/7.93		0.29/25.62	

SyN was conducted by ANTsPy and executed on the Intel(R) Xeon(R) E5-2650 v2 CPU.

the average images of affine and SyN are severely blurred which indicates an inaccurate registration and a weak robustness against different variants of the moving image. Comparing to VoxelMorph, FDRN provides a sharper average image and more reliable registered labels which resemble the labels of the fixed image better.

Evaluation on Unseen Brain MRI Datasets

The generality of the pretrained FDRN is evaluated on different unseen brain MRI datasets including CUMC12, MGH10, ABIDE and ADNI. Particularly, a standard preprocessing was conducted as mentioned in Section 4.1.3 and performed subject-to-subject registration for all the datasets where each of the images behaved as the fixed image and the remaining ones were registered to it. Experimental results are quantitatively summarized in Table 4.2. It is shown that FDRN performs best in Dice and NCC in all the unseen datasets by resorting to the large network capacity and efficient learning.

Model Analysis

Model variants: In order to analyze the network structure, experiments on different model variants in terms of model depth and width were conducted. As illustrated in Fig. 4.5, the green, blue, and magenta markers represent different variants of the proposed network

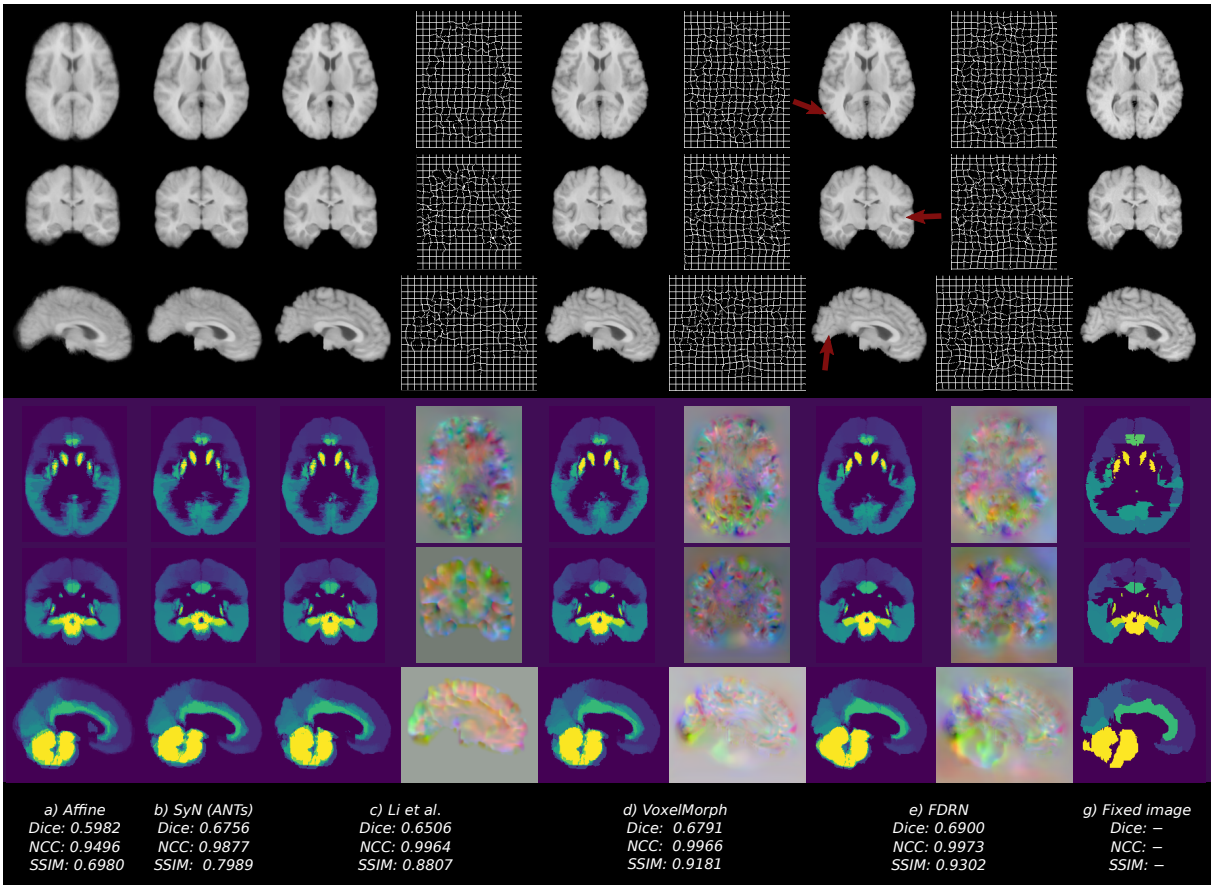


Figure 4.4: Visual evaluation of different registration methods on LPBA40 MRI dataset.

architecture. Following the same notation manner in Section 4.1.2, the VoxelMorph in Table 4.1 without and with ADS is respectively denoted as VM (16-0) and VM (16-0-ADS). The one with double features is indicated as VM(32-0) and Li's method is represented by Li (32-0). As expected, deepening and widening the network increase the model capacity and improve the Dice score. It is shown that deepening the model from 8-1 to 8-2 improves the Dice less than widening the channel to 16-1 because 16-1 has 2.4 times parameters as 8-2. Additionally, the performance of using channel concatenation (CC) instead of additive forwarding is presented. It is shown that comparing to 8-1, 8-1 (CC) indeed improves Dice but it nearly doubles the training memory (about 10.2GB) with which 16-2 (about 10.8GB) could be almost adopted, while 8-2 (CC) uses more than 13GB. Comparing to VM (16-0-ADS), the baseline model 8-1 achieves comparable Dice with nearly half of the runtime. 16-2 consumes similar runtime and training memory as VM (16-0-ADS) but contains 7.1 times model parameters. In Table 4.3, the amount of parameters required in

Table 4.2: Comparison of different deformable registration methods on the unseen MGH10, CUMC12, ABIDE and ADNI MRI datasets. Average Dice score of CUMC12 and MGH10 were calculated based on 7 segmented structures. 10 randomly chosen samples individually from ABIDE and ADNI were used for evaluation. Best results are in bold. (ADS was used in VoxelMorph.)

Dataset	Metrics	Affine	SyN [122]	Li et al. [86]	VoxelMorph [145]	FDRN
MGH10	Dice	0.6474	0.6699	0.6726	0.6678	0.6865
	NCC	0.7176	0.8615	0.8918	0.9049	0.9075
CUMC12	Dice	0.6111	0.6531	0.6617	0.6517	0.6669
	NCC	0.6603	0.7931	0.8300	0.8486	0.8603
ABIDE	NCC	0.7793	0.8739	0.9011	0.9203	0.9324
ADNI	NCC	0.8078	0.8852	0.9127	0.9279	0.9399

Table 4.3: Number of required parameters in different networks. VM: VoxelMorph; CC: Channel concatenation. (8-1: Baseline, 16-2: FDRN)

Models	8-1	8-1(CC)	8-2	8-2(CC)	8-4	16-0	16-1	16-2	Li(32-0)	VM(16-0)	VM(32-0)
#Params	285K	665K	466K	1.0M	830K	397K	1.1M	1.8M	695K	252K	1.0M

different model variants is listed.

Deep supervision: The effectiveness of deep supervision on the convergence of FDRN is evaluated. Particularly, the proposed FDRN is compared with the variant without the LR loss. As mentioned in Section 4.1.2, the LR loss is weighted by an exponentially decayed weighting factor λ so that FDRN learns a rough registration in the beginning from LR images and subsequently improves the registration accuracy fully based on the HR loss. In Fig. 4.6 a), it is shown that the convergence rate of FDRN is noticeable faster than the one without deep supervision since the early training phase. In the right figure, the impact of the deep supervision on the Dice score over epochs is demonstrated. Comparing to adopting larger learning rate, the proposed deep supervision accelerates the convergence and meanwhile improves the registration accuracy.

Segmentation loss: The impact of the weighting factor α_2 and the parameter c_1 of the segmentation loss on the Dice and NCC is evaluated. As depicted in Fig. 4.7 a), when α_2 increases, the segmentation loss gradually dominates the loss function which leads to a high Dice score. However, NCC drops severely when $\alpha_2 > 1$ which indicates that the segmentation loss might have overfitted the Dice score and result in an unrealistic DVF.

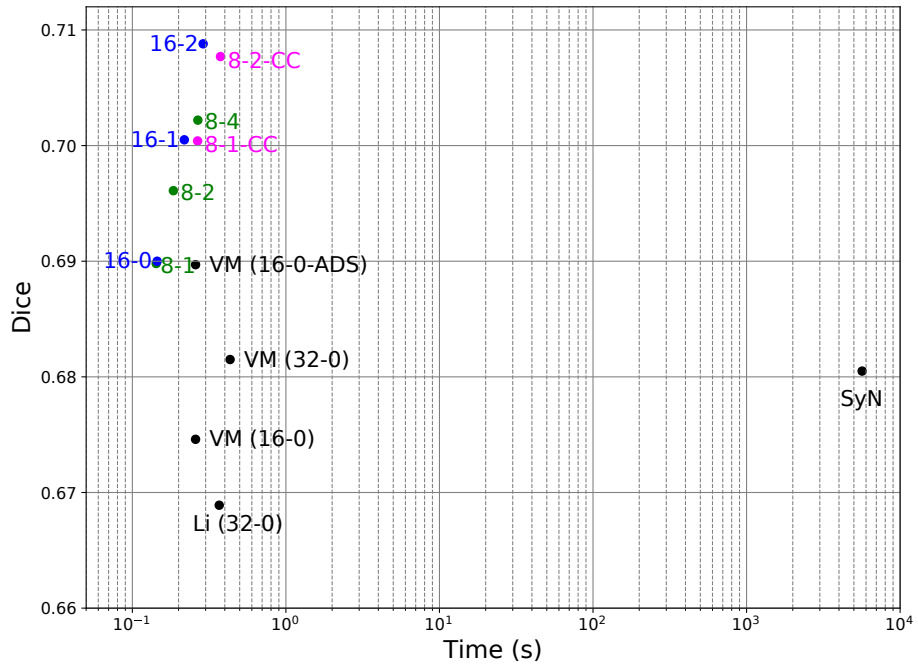


Figure 4.5: Dice performance and runtime of different model variants on LPBA40 dataset with $\alpha_2 = 0.3$. (8-1: Baseline, 16-2: FDRN)

In the experiment, α_2 was set as 0.2 for good performance of both Dice score and NCC. In the right figure, it is shown that the Dice is fine tuned by c_1 and NCC seems rarely effected.

Ablation Study: In the ablation study, the behavior of the different network variants is analyzed, including the removal of additive forwarding (AF) linking the encoder path to the decoder counterpart, the residual learning (RL) within the encoder and decoder stages, the LR loss for deep supervision (DS), and the segmentation loss SL. The experiments were performed on the LPBA40 dataset and evaluated by average Dice score and NCC as depicted in Table 4.4. It is shown that AF improves both the Dice score and NCC by directly forwarding the extracted fine features. RL seems to contribute less to the Dice score and NCC than AF but it alleviates the gradient vanishing during the convergence. DS mainly accelerates the convergence rate and has a strong impact on the Dice score especially in the early training epochs. The proposed SL improves the Dice score by 1.93%.

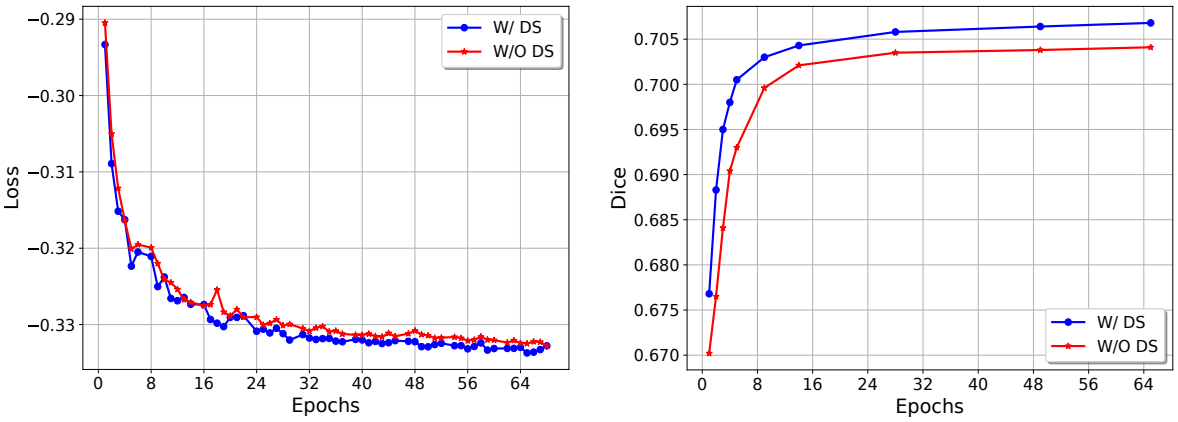


Figure 4.6: Impact of deep supervision on the model convergence ($\alpha_2 = 0.3$): a) Loss function over epochs; b) Dice score over epochs.

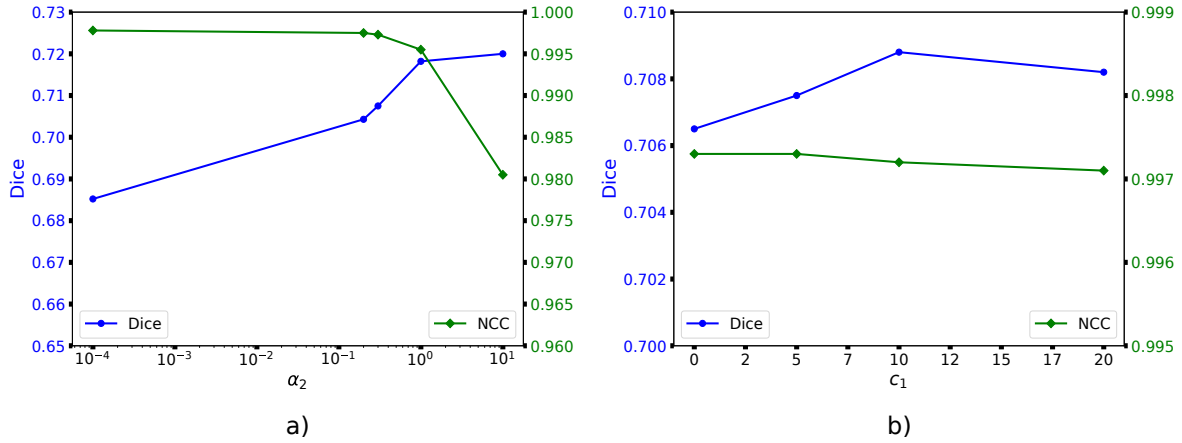


Figure 4.7: Effectiveness of the segmentation loss: a) Weight α_2 of the segmentation loss ($c_1 = 5$); b) Parameter c_1 of the segmentation loss ($\alpha_2 = 0.3$).

4.1.4 Discussion

The huge memory demand for 3D medical images limits the capacity of the registration network. In order to more efficiently exploit the memory resource, a compact deformable registration network FDRN is proposed based on the autoencoder backbone which achieves better performance in both registration accuracy and runtime comparing to the investigated state-of-the-art methods including symmetric image normalization (SyN), Li's method,

Table 4.4: Ablation study of the proposed FDRN based on average Dice score and NCC over all the segmented structures on the LPBA40 dataset with model 16-2 ($\alpha_2 = 0.3, c_1 = 5$). AF: Additive forwarding; RL: Residual learning; DS: Deep supervision; SL: Segmentation loss.

AF	✗	✓	✓	✓	✓
RL	✓	✗	✓	✓	✓
DS	✓	✓	✗	✓	✓
SL	✓	✓	✓	✗	✓
Dice/NCC	0.7001/0.9955	0.7046/0.9972	0.7048/0.9974	0.6882/0.9978	0.7075/0.9973

and VoxelMorph for brain MR images. Specially, the baseline model achieves comparable Dice as VoxelMorph and consumes nearly half of the runtime. FDRN improves the registration accuracy by enlarging the model capacity and obtains a performance gain of 1.46% in average Dice in comparison to VoxelMorph. Experiments show that the average of the registered images by FDRN contains sharper anatomical structures than the other methods and the average transformed labels resemble the labels in the fixed image most which indicate that FDRN has a better registration accuracy and strong robustness against different variants of the moving image. With regard to the computation time, the learning-based methods accomplish deformable registration of images of size 160×208×176 within 0.5s on the GPU and perform hundreds times faster than the traditional SyN on the CPU. Comparing to VoxelMorph, the baseline model halves the inference time and achieves comparable Dice. FDRN consumes similar runtime as VoxelMorph and contains 7.1 times parameters. It is necessary to mention that FDRN is a generalized model for deformable registration and is not limited to brain MR images. It can also be applied to other anatomical structures or CT images.

4.2 REM: Resolution Enhancement Module

Deep learning brings up a new generation of SR. Since the emergence of SRCNN [13], learning-based SR has been intensively studied for different applications from different aspects such as perceptual quality and upscaling factor. Many representative models [17–22,84] advance the state-of-the-art by adopting residual learning [18], dense connection [84],

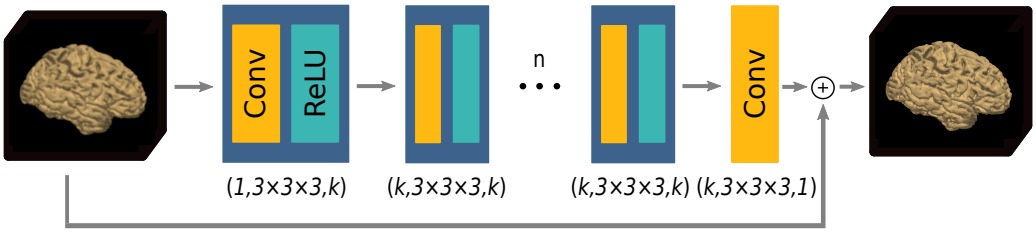


Figure 4.8: Structure of the proposed REM.

GAN [19, 21, 158], enhanced depth [17], channel attention [20], plug-and-play [22] to count a few. As a matter of fact, all the abovementioned deep learning-based SR models benefit from the extremely distinguished capability of CNN for feature extraction. Inspired by the work of Dai et al. [159], we intend to combine SR with registration task and propose a cascaded network for enhancing the performance of image registration. To this end, our resolution enhancement module (REM) serves as an auxiliary network and is desired to be a handy plug-in which is simple yet effective.

4.2.1 Architecture of REM

The structure of the proposed REM is schematically illustrated in Fig. 4.8. In order to circumvent the architecture design with a predefined upscaling factor, REM processes the input image of the same size as the output image. The neat structure contains successive 3D convolutional layers with Rectified Linear Unit (ReLU) inbetween. Residual learning is adopted to improve the learning efficiency and achieved by skip connection. The 3D kernel size is set as $3 \times 3 \times 3$. The number of intermediate convolutional layers is indicated by n and the amount of filters in convolutional layers is denoted by k. Usually, the larger n and k are, the better the performance is. To hold a compact design, based on empirical observations $n \leq 16$ and $k \leq 16$ are adopted since larger n and k bring limited performance gain for brain MR images while consume much resources. Due to the fact that REM does not alter the dimension of the input, it can be straightforwardly embedded into other vision tasks.

4.2.2 Evaluation of REM

Firstly, an experimental study of the configuration of REM is performed. Due to the extremely large memory consumption of dense connection for 3D images, the usage of dense connection is avoided and two types of residual learning are explored: residual on image and residual on features as illustrated in Fig. 4.9. REM-Variant1 draws skip connection directly from the input to the output so that the CNN model purely learns the high-frequency information, while REM-Variant2 applies residual learning on the intermediate feature maps. The same amount of convolutional filters and convolutional layers are used for both variants. In addition, for each variant, two configurations are constructed: $k8n6$ and $k12n4$. Each of the variants is well trained on the same LPBA40 dataset over 1000 epochs. The performance of each variant is evaluated quantitatively by PSNR and SSIM as summarized in Table 4.5. It is shown that variant1 performs better than variant2 in PSNR and SSIM. In fact, variant1 converges also much faster than variant2. It is worthy noting that usually the model capacity increases when the network goes deeper and wider. Depending on the performance requirement and data complexity, REM is not confined to a specific configuration.

Secondly, the performance of REM (16-8) for the upscaling of 2× (marked by blue rectangle) and upscaling of 4× (marked by green rectangle) on LPBA40 dataset is demonstrated in Fig. 4.10. Rows from top to bottom represent axial, coronal, and sagittal view, respectively. REM is compared with the trilinear interpolation. It is shown that REM improves the visual quality significantly by generating sharper contours and providing better visibility of the detailed structures. Besides, the quantitative assessment for both methods in PSNR and SSIM is depicted and REM achieves significantly better results for both upscaling factors which coinsides with the visual perception.

Thirdly, a wide parameter sweep of the configurations is performed. We set the channel

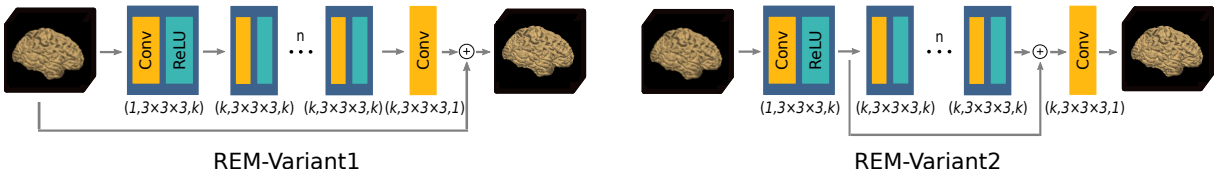


Figure 4.9: Two REM variants with the same number of model parameters. REM-Variant1: residual on image; REM-Variant2: residual on extracted features. (Best viewed in color.)

Table 4.5: Evaluation of different REM configurations in PSNR and SSIM. The notations k8n6 and k12n4 indicate the configuration of $k = 8, n = 6$ and $k = 12, n = 4$, respectively.

	REM-Variant1		REM-Variant2	
	k8n6	k12n4	k8n6	k12n4
Scale 2×	44.98/0.9951	45.19/0.9953	44.57/0.9945	44.92/0.9950
Scale 4×	37.23/0.9660	37.26/0.9661	36.99/0.9639	36.89/0.9636
# Parameters [K]	10.8	16.2	10.8	16.2
Train/Test Memory [GB]	6.30/1.69	6.85/2.05	6.30/1.69	6.85/2.05

Table 4.6: Number of network parameters and required inference memory of different variants.

Models	8-8	16-8	16-16	32-8	32-16	64-8	64-16
#Params.	14.3K	56.3K	111.7K	223.2K	444.6K	888.8K	1.7M
Test Memory [GB]	1.7	2,4	2.4	3.8	3.8	6.7	6.7

number $k = 8, 16, 32, 64$ and the number of intermediate convolutional layers $n = 8, 16$. The model variants are trained on a patch size of 64×64×64 and mini-batch size of 2. The performance of all the investigated variants is demonstrated in Fig. 4.11. It is shown that when $k > 16$, the benefit margin of PSNR and SSIM is not evidently increased. In addition, we summarize the number of parameters and the consumed inference memory of the studied configurations in Table 4.6. In practise, we have chosen $k = 16, n = 8$ as the configuration of REM to balance the SR performance and memory usage.

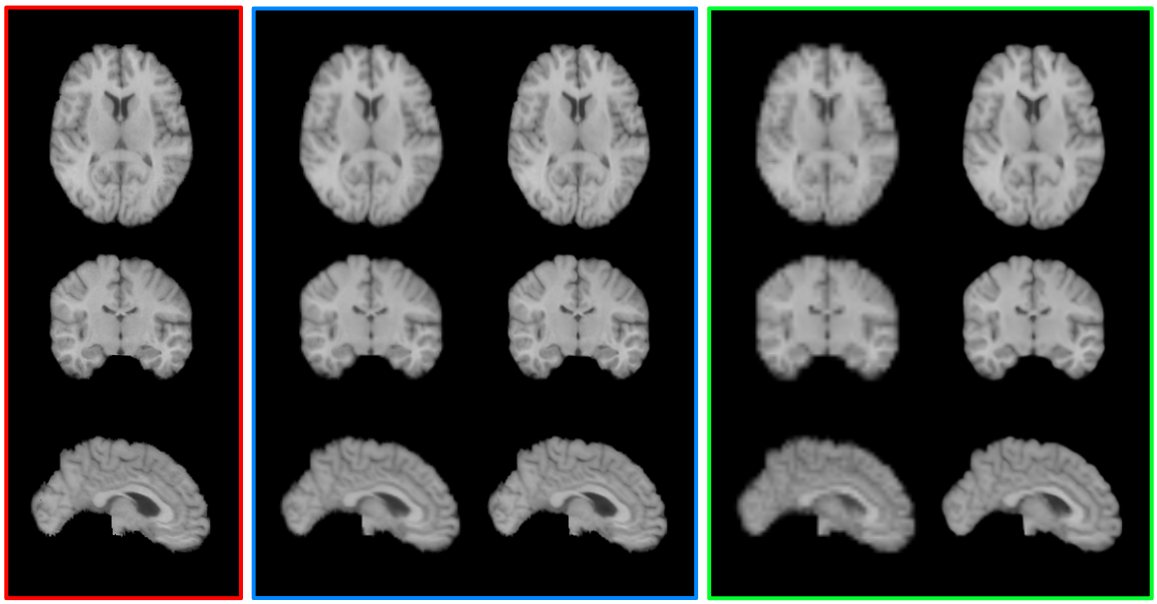


Figure 4.10: Visual evaluation of REM for upscaling factors of 2× and 4× on LPBA40 brain MRI dataset. Red: ground truth; Blue: upscale of 2×; Green: upscale of 4×

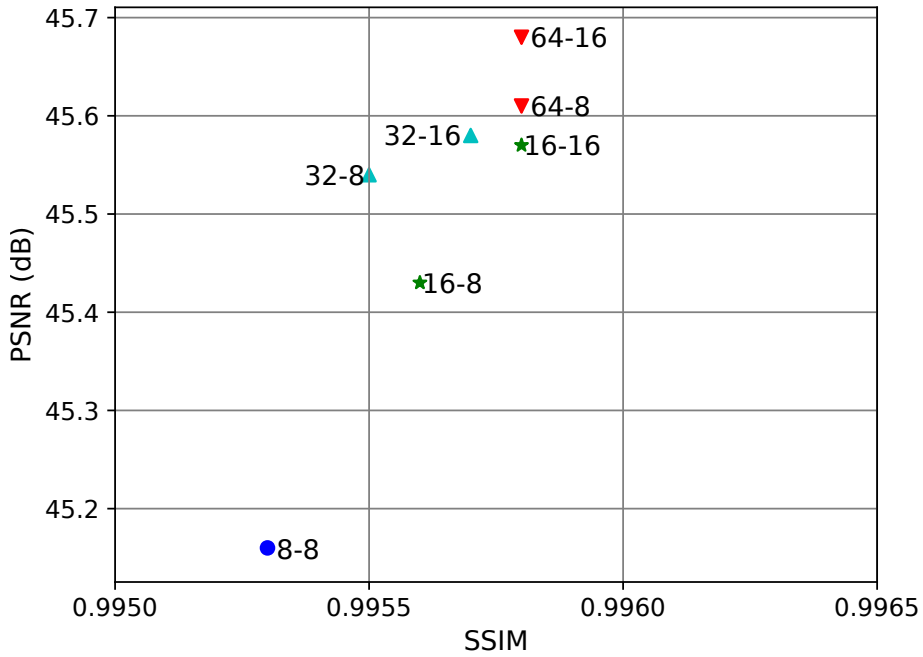


Figure 4.11: Performance evaluation of different REM configurations in PSNR and SSIM on LPBA40 MRI dataset.

4.3 ReFDRN and ReVoxelMorph: Resolution-Enhanced FDRN and VoxelMorph

SR is favored in most of the applications due to the improvement of visual quality. In [159], Dai et al. demonstrate the effectiveness of resolution enhancement by SR on other vision tasks including edge detection, semantic image segmentation, digit recognition, and scene recognition. Some works take advantage of the HR features extracted by SR for specific vision tasks such as semantic segmentation [160], face recognition [161] and pedestrian identification [162]. However, these methods require a highly specialized SR approach for the individual task which makes the embedding of SR less convenient. In this section, a general framework which combines SR with vision tasks coping with LR input is presented. To evaluate the framework, the proposed REM is applied on two registration networks FDRN [59] and VoxelMorph [148] and we denote the resolution enhanced networks as ReFDRN and ReVoxelMorph. Note that the proposed cascaded framework is not confined to image registration, it can also be applied to other vision tasks such as image segmentation, object detection, and scene recognition.

4.3.1 Architecture of ReFDRN

The architecture of the proposed ReFDRN is demonstrated in Fig. 4.12. REM is connected with FDRN in a cascaded manner. ReFDRN takes LR images as input and outputs the corresponding SR images and the DVF. The dotted lines in different colors denote the dataflow for the individual component of the loss function as formulated in Eq. 4.11. Mathematically, the SR image y and the DVF z can be expressed as

$$\begin{aligned} y &= \boldsymbol{REM}(x),\ x \in \mathbb{R}^{2\times 1\times L\times W\times H}, y \in \mathbb{R}^{2\times 1\times L\times W\times H} \\ z &= \boldsymbol{FDRN}(\tilde{y}),\ \tilde{y} \in \mathbb{R}^{1\times 2\times L\times W\times H}, z \in \mathbb{R}^{1\times 3\times L\times W\times H} \end{aligned} \tag{4.10}$$

where $\tilde{y}$ is the rearrangement of y by switching the batch size and the number of channels since REM needs to treat both input images seperately, while FDRN considers them as an image pair. The overall loss composes of three components: the main loss $\boldsymbol{Loss}_{main}$ based on LNCC, the auxiliary loss $\boldsymbol{Loss}_{aux}$ formulated in Eq. 4.13, and the regularization term

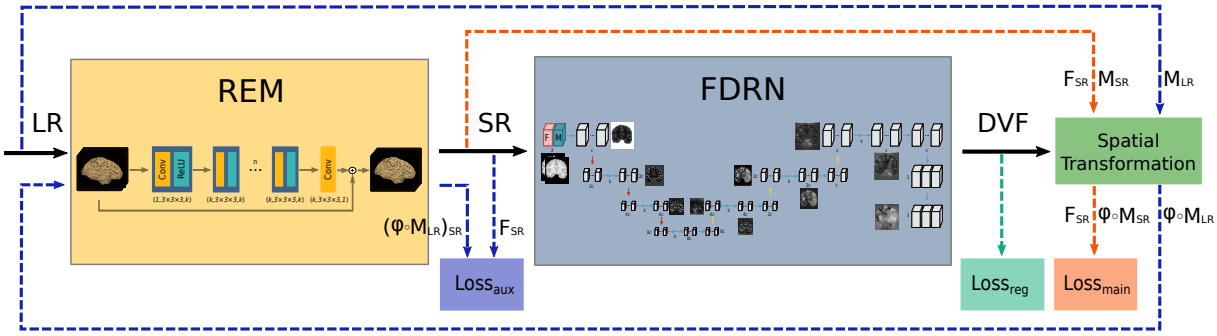


Figure 4.12: Structure of the resolution enhanced FDRN: ReFDRN. Dotted lines performs only during training. Input: LR images; Output: SR images and DVF. (Best viewed in color.)

$Loss_{reg}$ in Eq. 4.14. The weighting parameters λ_1 and λ_2 serve as a tradeoff between the main loss and the rest.

$$Loss_{total} = Loss_{main} + \lambda_1 Loss_{aux} + \lambda_2 Loss_{reg} \tag{4.11}$$

In particular, the main loss $Loss_{main}$ focuses on the original vision task, i.e., image registration, and LNCC is used as in FDRN. The difference is that instead of processing on LR input, it performs on the super-resolved SR images as formuated in Eq. 4.12.

$$Loss_{main} = -LNCC\big(\phi \circ REM(M_{LR}), REM(F_{LR})\big) \tag{4.12}$$

In order to strengthen the fidelity of the output, an auxiliary loss which imposes similarity constraint directly on the raw input is involved as expressed below based on the Huber loss.

$$Loss_{aux} = Huber\big(REM(\phi \circ M_{LR}), REM(F_{LR})\big) \tag{4.13}$$

Note that the auxiliary loss presented above is elaborately devised for unsupervised image registration to facilitate the coupling of the cascade and in fact, for other tasks if a proper design of the auxiliary loss is not available, the two networks can be straightforwardly connected by feeding the output of REM into the following one.

Besides, following the loss function of FDRN 4.3, regularization on the smoothness of the DVF is applied:

$$Loss_{reg} = \sum_{S_k} ||z - S_k z||_2^2, \tag{4.14}$$

where S_k indicates the shifting operator along (u,v,w) direction by vector k with $k = \{(u,v,w) \mid u,v,w \in \{0,1\}\}$ and $||\cdot||_2^2$ represents the L2-norm.

4.3.2 Experimental Results

In order to evaluate the effectiveness of resolution enhancement on image registration, REM with configuration of 16-8 is plugged into two investigated registration networks FDRN [59] and VoxelMorph [148]. As described in Section 4.3.1, the same cascade scenaro is performed for ReVoxelMorph by replacing FDRN with VoxelMorph. The performance of ReFDRN and ReVoxelMorph are evaluated at scaling factors of 2 and 4 on the brain MRI dataset LPBA40 [149].

Network training: In order to improve the training efficiency, the REM model is pretrained on the same LPBA40 dataset as FDRN. In particular, 30 samples out of 40 in LPBA40 are utilized for training, the rest 4 and 6 are respectively used for validation and testing. The hyperparameters including the number of convolutional layers and the amount of channels are tuned in a trial-and-error manner. The fine-tuned REM is then cascaded with the fresh FDRN. To maintain the visual fidelity of the SR images and ease the training, the weights of REM are frozen during the training of ReFDRN. The batchsize is set as one and Adam is utilized as the optimization algorithm. The learning rate is set as 0.002 and decayed by 0.9 every 1000 iterations until delined to 10^{-4}.

Results: As the proposed REM has shown great performance for spatial resolution enhancement as depicted in Section 4.2, how does the improved image quality influence the performance of image registration? The impact of REM on two registration methods FDRN [59] and VM (short for VoxelMorph [148]) are evaluated following the cascade scenario as depicted in Fig. 4.12. Specially, FDRN (16-1) and VM (16-0) as denoted in Table 4.3 are selected in this experiment. The GT volumes are firstly downscaled based on trilinear interpolation by a factor of 0.5. The image dimension is then restored by a trilinear upscaling of factor 2. The trilinearly upscaled volume is passed to FDRN and VM and denoted by $\text{FDRN}_{\downarrow\uparrow}$ and $\text{VM}_{\downarrow\uparrow}$. The same scenario is performed for the scale of 4×.

Table 4.7: Summary of performance evaluation in Dice and NCC. The subscript $\downarrow\uparrow$ denotes trilinear downsampling the input followed by the trilinear upsampling of the same scale and $\downarrow$ indicates NN downsampling without upsampling. VM is short for VoxelMorph [148].

Scale	Metrics	Affine$_{\downarrow}$	FDRN$_{\downarrow}$	VM$_{\downarrow}$	FDRN$_{\downarrow\uparrow}$	ReFDRN$_{\downarrow\uparrow}$	VM$_{\downarrow\uparrow}$	ReVM$_{\downarrow\uparrow}$
1×	Dice	0.6079	0.6810	0.6746	0.6810	–	0.6746	–
	NCC	0.9506	0.9976	0.9973	0.9976	–	0.9973	–
2×	Dice	0.6001	0.6369	0.6350	0.6797	0.6796	0.6722	0.6741
	NCC	0.9511	0.9923	0.9920	0.9963	0.9977	0.9960	0.9970
4×	Dice	0.5546	0.5536	0.5551	0.6676	0.6736	0.6593	0.6676
	NCC	0.9396	0.9693	0.9695	0.9920	0.9962	0.9916	0.9932

FDRN$_{\downarrow\uparrow}$ and VM$_{\downarrow\uparrow}$ are compared with the proposed ReFDRN$_{\downarrow\uparrow}$ and ReVM$_{\downarrow\uparrow}$. Besides, the impact of image dimension on the registration accuracy is also demonstrated. We perform image registration on the downscaled images and denote them by FDRN$_{\downarrow}$ and VM$_{\downarrow}$. The performance of the affine transformation is set as the baseline. The results are summarized in Table 4.7. Comparing the columns of FDRN$_{\downarrow\uparrow}$ and ReFDRN$_{\downarrow\uparrow}$, VM$_{\downarrow\uparrow}$ and ReVM$_{\downarrow\uparrow}$, it is obvious that image sharpness does have strong impact on registration accuracy and the phenomenon becomes more evident when the input images are more severely blurred such as for the scale of 4×. Comparing FDRN$_{\downarrow}$ with FDRN$_{\downarrow\uparrow}$ and VM$_{\downarrow}$ with VM$_{\downarrow\uparrow}$, it is shown that not only the sharpness, but also image dimension plays an important role on the registration performance.

In Fig. 4.13, we illustrate the visual comparison between the registration performance using trilinear interpolated images (marked in blue) and the ones using SR images (marked in green). Firstly, it is shown that the SR images have much sharper structures than trilinear interpolation with significant improvement in PSNR and SSIM. Since ReFDRN and ReVM produce not only the DVF but also the super-resolved images, the resolution enhanced images can be employed for successive medical diagnosis. Secondly, ReFDRN$_{\downarrow\uparrow}$ and ReVM$_{\downarrow\uparrow}$ respectively achieve better registration performance than FDRN$_{\downarrow\uparrow}$ and VM$_{\downarrow\uparrow}$ in visual perception and quantitatively in Dice and NCC. Additionally, the deformation fields of the studied methods are exhibited and we do observe noticable differences between the REM embedded methods and the ones without REM.

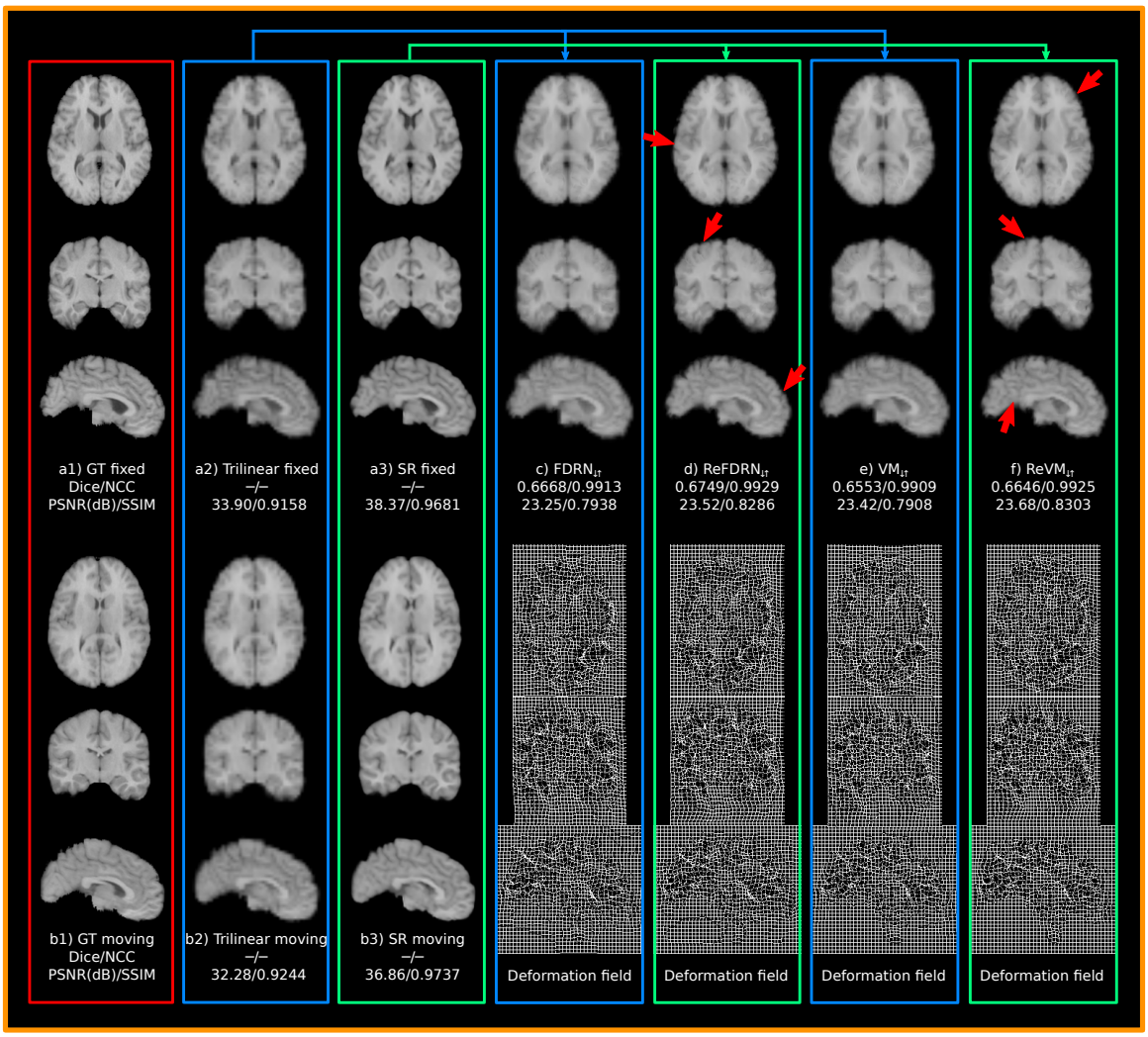


Figure 4.13: Visual evaluation of the impact of REM on registration performance for scaling factor of 4×. Red: GT; Blue: Trilinear interpolated image and the corresponding registration result; Green: SR image and the corresponding registration result.

4.4 Conclusion

In this chapter, the use of resolution enhancement by SR for image registration is demonstrated. Firstly, a state-of-the-art CNN-based deformable registration network FDRN which bases on an encoder-decoder backbone is proposed. Secondly, a light-weight resolution enhancement module (REM) is presented and evaluated. Finally, a cascade scheme for REM and the registration network is introduced. The cascaded network copes with images of undesirable resolution and provides not only the DVF but also the resolution enhanced images. In the experiments, the cascade scheme is evaluated on two registration networks FDRN and VoxelMorph at scaling factors of 2× and 4× on LPBA40 brain MRI dataset. It is shown that the embedding of REM not only provides a more accurate registration by resorting to the improved image quality, not only significantly improves the visual quality of the input images, which can be utilized for successive analysis.

Chapter 5

Real-Time RNN-based Super-Resolution

In the previous chapters, SR methods MPG+BTVSW and FL-MISR were presented based on iterative optimization and GPU acceleration. In this chapter, a hardware-accelerated deep learning-based SR model is proposed using residual recurrent neural network (RNN) implemented on field programmable gate array (FPGA) [60]. In Section 5.4.5, it is shown that the FPGA implementation performs more than 5× faster than the GPU variant. The proposed ERVSR has a compact RNN structure and supports a SR output of 3840×2160 at 76 fps which shows a great potential for the use of hardware-embedded SR in fast CT applications such as inline-CT. Specially, the proposed ERVSR leverages the input frame and the temporal information of previous frames entailed in the hidden state to reconstruct the high-resolution counterpart. To reduce the network parameters, the low-resolution input branch and the hidden state branch are convolved individually and a channel modulation coefficient is proposed to explicitly guide the network to allocate the amount of output feature channels to each branch. Additionally, in order to reduce the memory consumption, a dedicated lightweight compression of the hidden state is performed by introducing a statistical normalization scheme followed by a fixed-point quantization. Besides, group convolution and depthwise separable convolution are adopted to further compact the network. The proposed ERVSR is evaluated on multiple public datasets from different aspects. Experimental results demonstrate that ERVSR performs better than the other state-of-the-art FPGA-based VSR methods by a large margin.

5.1 Previous Work of FPGA-Based Super-Resolution

In the literature, FPGA implementations of multi-frame VSR are mainly built on the traditional iterative methods such as iterative back projection (IBP) [163,164] and L1BTV [165]. Due to the computationally expensive inter-frame registration and the iterative optimization scheme, the traditional FPGA-based VSR implementations have difficulty in challenging applications such as upscaling FHD to 4K UHD. In contrast to multi-frame VSR algorithms, single-frame interpolation-based scaling methods [166–168] reduce the computational complexity significantly but the reconstruction performance is limited by the lack of high-frequency details. To exploit the high-frequency information entailed in external database, Yang et al. [169] propose a learning-based SISR system using anchored neighborhood regression (ANR) which achieves an output resolution of 1920×1080 (FHD) at 60 fps. Kim et al. [170] introduce a hardware-friendly architecture based on the edge-orientation analysis and linear mapping which supports a real-time reconstruction of 4K UHD video streaming at 60 fps.

More recently, the deep learning-based VSR methods have achieved the state-of-the-art performance. Particularly, Manabe et al. [171] introduce the FPGA implementation of a CNN-based SR model SRCNN [13]. Afterwards, a series of works [172–174] propose CNN accelerators for SR reconstruction based on FSRCNN [14]. He et al. [172] propose a block-based SR strategy that each frame is cropped into blocks and depending on the total variation of the blocks, they are either dispatched to FSRCNN or upscaled by simple interpolation. Based on the work of [172], Shi et al. [174] propose a fast transposed convolution approach using Winograd algorithm and achieve a frame rate of 120 fps for upscaling FHD videos to 4K UHD. In [173], Chang et al. present a CNN accelerator for SISR based on FSRCNN which supports parallelization by transforming the transposed convolution to standard convolution using their proposed TDC method. Different from the aforementioned CNN accelerators which are built on SRCNN or FSRCNN, Kim et al. [112] propose a residual convolutional neural network which employs depthwise separable convolution to reduce the network parameters. Besides, instead of using 2D convolution, they adopt 1D horizontal depthwise convolution followed by pointwise convolution to save the line memories. Their hardware-efficient SR model supports upscaling from FHD to 4K UHD at 60 fps. Comparing to the FPGA implementations of the traditional iterative multi-frame VSR methods, the DCNN-based implementations achieve significantly higher PSNR and data throughput. However, due to the limited hardware resources, the existing

FPGA-based CNN accelerators perform SISR on the video sequence without taking advantage of the underlying information entailed in the neighboring frames which leads to temporal inconsistency.

In order to preserve long-term frame information and improve the temporal consistency, Sajjadi et al. [39] introduce a frame-recurrent VSR network FRVSR which utilizes the HR estimation of the previous frame to super-resolve the current frame based on motion compensation. To circumvent the computationally expensive motion estimation, Huang et al. [38] present a directional recurrent convolutional network which employs 3D feedforward convolution to capture spatio-temporal patterns for short-term fast-varying motions. Although they adopt weight-sharing in the recurrent convolutions, due to the complex network structure, there are still about 42K parameters for the variant with smallest temporal step which requires 0.48s to perform 2× upscaling for each frame of the Vid4 dataset [108]. Fuoli et al. [175] propose a recurrent latent space propagation (RLSP) algorithm which uses high-dimensional hidden states to propagate the temporal information without extra motion compensation. Particularly, multiple LR frames are concatenated with the previous HR output and the hidden state as the model input. Based on the shallow and wide network architecture, RLSP can produce 25 fps of FHD video. To the best of our knowledge, the existing RNN-based VSR approaches are implemented on the GPU due to the network complexity and have difficulty to fulfill the challenging real-time requirements of applications such as UHD video services.

Although multi-frame VSR methods have achieved promising performance, registration or explicit motion compensation of multiple LR frames is extremely resource consuming for FPGA implementation. In order to preserve the temporal consistency without compromising the data throughput, a residual recurrent convolutional neural network ERVSR is proposed based on single LR input. It is shown that by resorting to a compact recurrent network design and efficient residual learning, a hardware-friendly implementation for VSR on FPGA can exploit temporal information over 30 frames and achieve a frame rate of 76 fps.

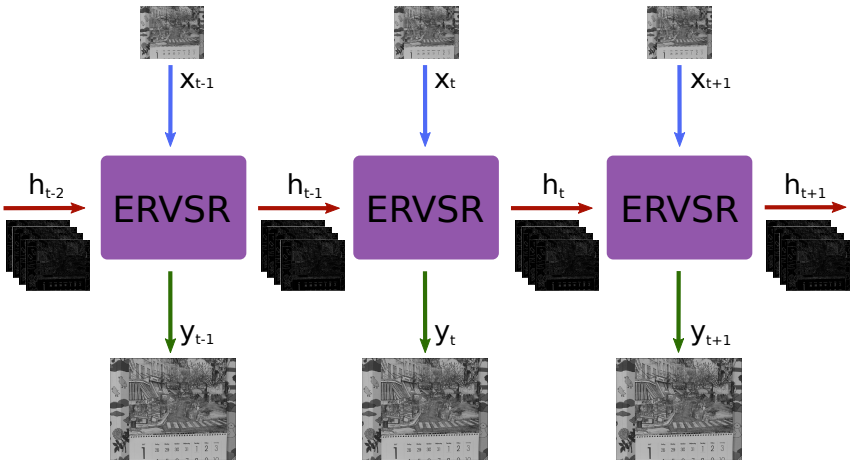


Figure 5.1: Schematic illustration of the proposed RNN model. At time t, the network ERVSR is fed with the LR frame x_t and the recurrent input h_{t-1} and outputs the HR frame y_t with the hidden state h_t.

5.2 RNN-Based Video Super-Resolution Method

The proposed hardware-efficient VSR model ERVSR is built on the residual recurrent convolutional neural network. The overview of the proposed model is illustrated in Fig. 5.1. As shown, the hidden state is propagated forward along the temporal dimension. Combining the LR frame with the recurrent input which conveys the temporal information of the previous frames, ERVSR reconstructs the HR frame along with the associated hidden state.

5.2.1 ERVSR Architecture

The proposed ERVSR is a fully convolutional recurrent network based on residual learning. A schematic illustration of ERVSR for an upscaling of r is demonstrated in Fig. 5.2. It should be noted that only the luminance channel of the LR frame is super-resolved by ERVSR and the labels above the arrows indicate the number of channels. At time t, ERVSR is fed with the LR frame x_t of dimension $W \times H \times 1$ and the HR recurrent input h_{t-1} of size $W \times H \times r^2$. A sequence of hardware-efficient operations are performed to generate the hidden state h_t and the reconstructed HR frame y_t. The network model is

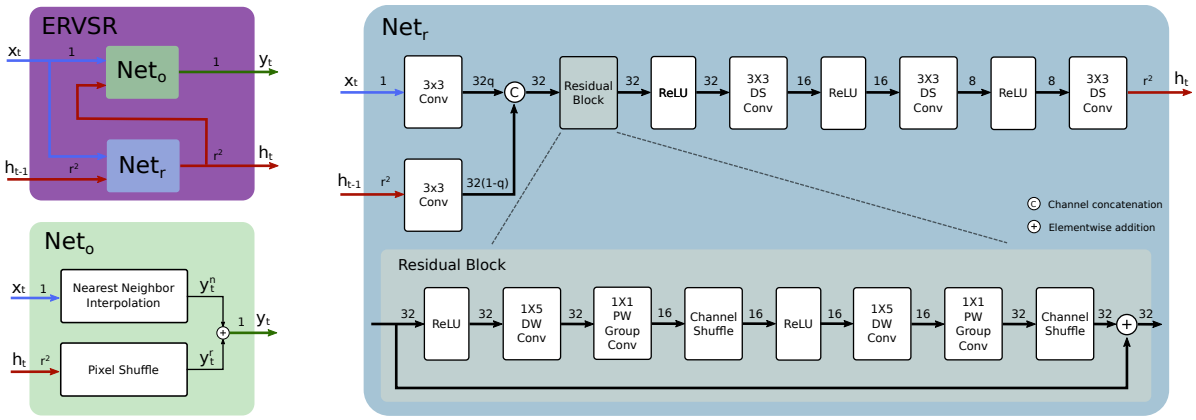


Figure 5.2: Structure of ERVSR for an upscaling factor of r. Labels above arrow connections represent the channel dimensions. The number of groups in all group convolutions is set as two. DS Conv: Depthwise separable convolution; DW Conv: Depthwise convolution; PW Conv: Pointwise convolution.

mathematically formulated as follows:

$$\begin{aligned} h_t &= Net_r(x_t, h_{t-1}) \\ y_t &= Net_o(x_t, h_t), \end{aligned} \tag{5.1}$$

where Net_r represents the hidden function of ERVSR which contains most of the functional blocks and Net_o indicates the output function. Specially, residual learning is employed to improve the learning efficiency and prevent from gradient vanishing. Based on the skip connection embedded in Net_o, the reconstructed HR frame y_t is expressed as

$$y_t = y_t^n + y_t^r. \tag{5.2}$$

y_t^r denotes the HR residual image of size $rW \times rH \times 1$ and y_t^n is the upscaled LR frame by nearest neighbor (NN) interpolation. It is worthy noting that y_t^r is the rearranged multi-channel h_t in the HR grid by pixel shuffle [176] and the upscaling factor r is set as 2.

In order to reduce the network parameters, slow fusion is applied on the two input branches of the hidden function Net_r by performing convolution individually before channel concatenation. Additionally, inspired by [112], depthwise separable convolution is adopted instead of the standard convolution. Specially, 1D horizontal depthwise convolution is performed followed by a pointwise convolution to achieve large receptive field in the horizontal dimen-

Table 5.1: Amount of parameters in RB using standard convolution and group convolution (# of Groups = 2).

	1×5 DW[a]	1×1 PW[b]	1×5 DW[c]	1×1 PW[d]	Total of MUL
Stand. Conv	(32,1,5,32)	(32,1,1,16)	(16,1,5,16)	(16,1,1,32)	1264
Group Conv	(32,1,5,32)	(16,1,1,8) (16,1,1,8)	(16,1,5,16)	(8,1,1,16) (8,1,1,16)	752

[a]: 1*st* DW in RB; [b]: 1*st* PW in RB; [c]: 2*nd* DW in RB; [d]: 2*nd* PW in RB.

sion and reduce the vertical receptive field for saving line memories. To further compact the network, instead of using standard pointwise convolution, pointwise group convolution is leveraged in the residual block (RB). Particularly, the number of groups is set as two and channel shuffle is utilized to enable crosstalk between groups. Comparing to the standard pointwise convolution, group convolution effectively reduces the amount of parameters by approximately 40% within the RB as depicted in Table 5.1.

5.2.2 Channel Modulation Coefficient

As depicted in Fig. 5.2, the hidden function Net_r consists of two input branches: the LR frame x_t and the hidden state h_{t-1}. It is intuitive to early fuse them by concatenation along the channel dimension which generates the feature maps of size $W \times H \times (r^2+1)$. However, this is computationally expensive and demands more hardware resources. We perform slow fusion and introduce a channel modulation coefficient q which addresses the above deficiency from two aspects. Firstly, the network is explicitly guided to allocate the feature channel resources to the LR frame and the hidden state branches which improves the model efficiency. Secondly, convolution is performed individually on each branch and depending on the coefficient q, a noticeable amount of parameters is reduced. Specially, the LR branch obtains $\lfloor 32q \rfloor$ feature channels and the hidden state yield the remaining $32 - \lfloor 32q \rfloor$ channels. The output feature maps are aggregated to 32 channels and propagated to RB. It is necessary to note that the channel modulation coefficient is a hyperparameter which needs to be tuned in the range of $1/32 \leq q < 1$ during the training phase. In the implementation, the channel modulation coefficient is set as $q = 0.65$ for a tradeoff between the quality of the reconstructed SR frame and the model complexity. Comparing to the naive early fusion, the number of parameters is decreased significantly

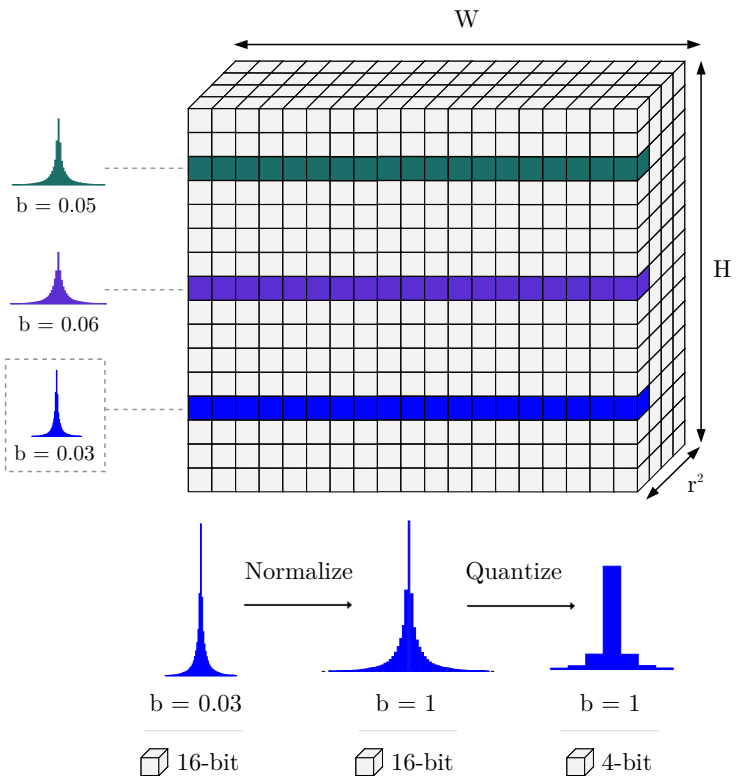


Figure 5.3: Proposed row-based compression scheme for the hidden state. An efficient normalization is performed rowwise on each channel of the hidden state based on the estimated Laplacian scale parameter b followed by a fixed-point quantization.

from 1440 to 612 for the upscale factor of 2×. It should be noted that q is in the range of $1/32 \leq q \leq 1$ because there are totally 32 output channels and at least one channel is required for the input LR image, while $q = 1$ makes the network non-recurrent which suits the cases of independent static scences such as image SR. A detailed analysis of q is shown in Section 5.4.6.

5.2.3 Hidden State Compression

In real-time applications such as UHD video services, the memory required for the hidden state is fairly considerable for the FPGA implementation. In order to reduce the memory consumption induced by the recurrent input, a dedicated lightweight compression scheme is proposed which efficiently quantizes the 16-bit fixed-point hidden state to 4-bit

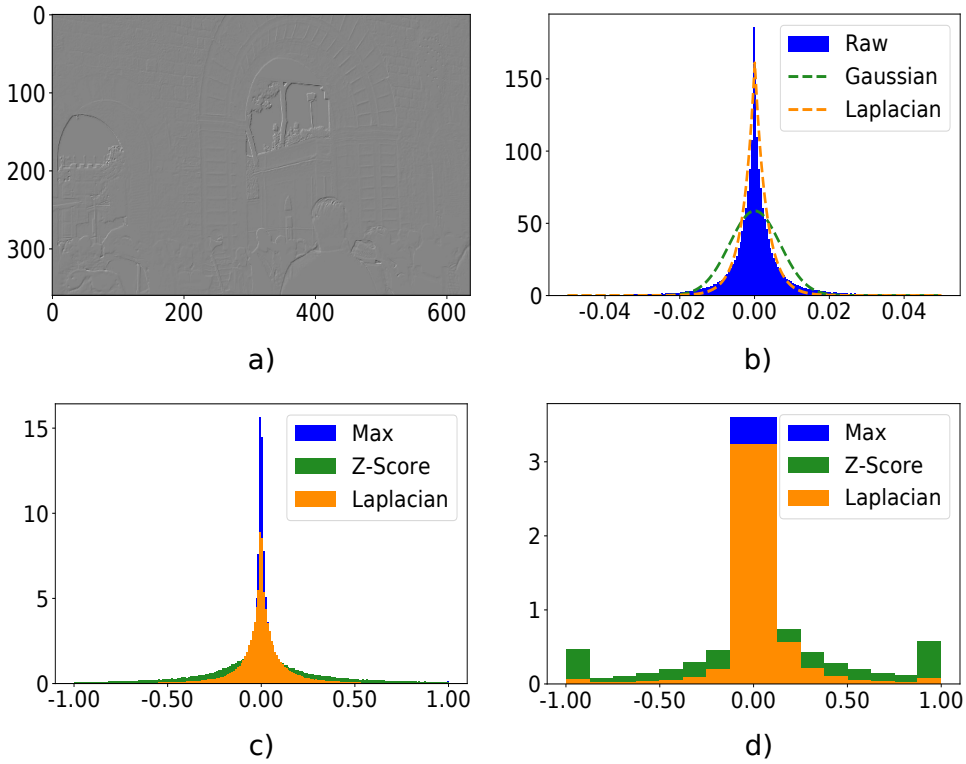


Figure 5.4: Demonstration of the effectiveness of the proposed normalization scheme on archpeople_001 from UDM10 [178]: (a) The hidden state; b) Average of the histograms of the row profiles and the average distribution fitting; (c) Average of the histograms of the normalized row profiles; d) Average of the histograms of the quantized row profiles.

representations so that the memory cost is decreased dramatically. As shown in Fig. 5.3, the proposed row-based compression strategy consists of two steps: a statistical normalization followed by a fixed-point quantization. Different from the Min-Max normalization, the proposed normalization scheme is derived from a statistical model of the hidden state which behaves more robust to outliers. Particularly, from the experimental observations it is found that the elements of each row of the hidden state approximately follow a Laplacian distribution which can be described by a location parameter μ and a positive scale parameter b. As the hidden state conveys the high-frequency information which usually centers around zero, for computational efficiency μ is set as 0. In fact, most of the μ is less than 10^{-4}. The scale parameter b_i of the ith row can be predicted by the maximum likelihood estimation (MLE) [177] as below:

$$\hat{b}_i = \frac{1}{W} \sum_{j=1}^{W} |x_{ij}|, \tag{5.3}$$

where x_{ij} denotes the jth element of the ith row in one feature map of the hidden state and W indicates the length of the row. As the scale parameter b_i is not necessarily the same for each row, it should be predicted individually. Based on the estimated $\hat{b}_i$, each element of the ith row x_{ij} is normalized by

$$x_{ij}^n = \frac{x_{ij}}{\beta \cdot \max(\hat{b}_i, b_{min})}, \tag{5.4}$$

where β is a positive scalar which controls the spread of the normalized values and b_{min} is used to prevent zero division. As 4-bit fixed-point representations of the hidden state are required, the word length (WL) is set as 4 and the integer length (IL) as 1 in the quantization step which covers the dynamic range of $[-1,1]$. In order to preserve the high-frequency information in the quantized value x_{ij}^q, β is chosen as 8 to locate the vast majority of the normalized elements in the interval $[-1,1]$. In Fig. 5.4, the effectiveness of the proposed normalization scheme is demonstrated. Specially, Fig. 5.4a shows one feature map of the hidden state. In Fig. 5.4b, the average histogram of the row profiles of the hidden state is depicted in blue and the average estimated fitting by Gaussian and Laplacian distribution in green and orange, respectively. It is shown that the average histogram of the row profiles closely matches the Laplacian distribution. In Fig. 5.4c, we illustrate the average histogram of the normalized row profiles by the modified Min-Max normalization proposed by Kim et al. [112] named Max for brevity in this chapter, the Z-Score normalization, and the proposed Laplacian normalization. It is shown that for $\beta = 8$, the proposed normalization scheme spreads all the elements naturally in the range of $[-1,1]$. Fig. 5.4d demonstrates the average histogram of the quantized row profiles with $WL = 4, IL = 1$. It is shown that the Max normalization squeezes most of the values centered at zero so that it tends to lose the data fidelity due to the quantization effect especially in the presence of outliers. On the contrary, Z-Score normalization distributes the values sparsely over a wide range which severely truncates the high-frequency details to the limit ± 1 during the quantization step and degrades the compression performance. The proposed Laplacian normalization scheme exploits the normalization potential based on the distribution of the image residuals in the hidden state and preserves the data fidelity to the best extent during the compression.

Since the estimated Laplacian parameter $\hat{b}_i$ is required to restore x_{ij} in the decompression step as formulated in Eq. (5.5), $\hat{b}_i$ is quantized and stored along with the hidden state. In the implementation, we set $WL = 12, IL = 1$ for the quantization of the parameter $\hat{b}_i$.

$$x_{ij}^d = \beta \cdot \hat{b}_i \cdot x_{ij}^q \tag{5.5}$$

5.2.4 Loss Function

In this work, Huber loss with $\delta = 1$ is adopted as the loss function because it overcomes the drawbacks of the L_1 and L_2 loss. Specially, comparing to the L_1 loss, Huber loss is differentiable at 0 and leads to a faster convergence, while in contrast to the L_2 loss, Huber loss tends to be more robust to outliers. Usually, Huber loss is formulated as

$$Loss(x, \hat{x}) = \sum_{i=1}^{N} f(x_i, \hat{x}_i), \tag{5.6}$$

with $x, \hat{x}$ being respectively the ground truth (GT) and the estimated image. N denotes the number of pixels in the image x and f is expressed by

$$f(x_i, \hat{x}_i) = \begin{cases} 0.5 \cdot (x_i - \hat{x}_i)^2 / \delta, & |x_i - \hat{x}_i| \leq \delta, \\ |x_i - \hat{x}_i| - \delta/2, & |x_i - \hat{x}_i| > \delta. \end{cases} \tag{5.7}$$

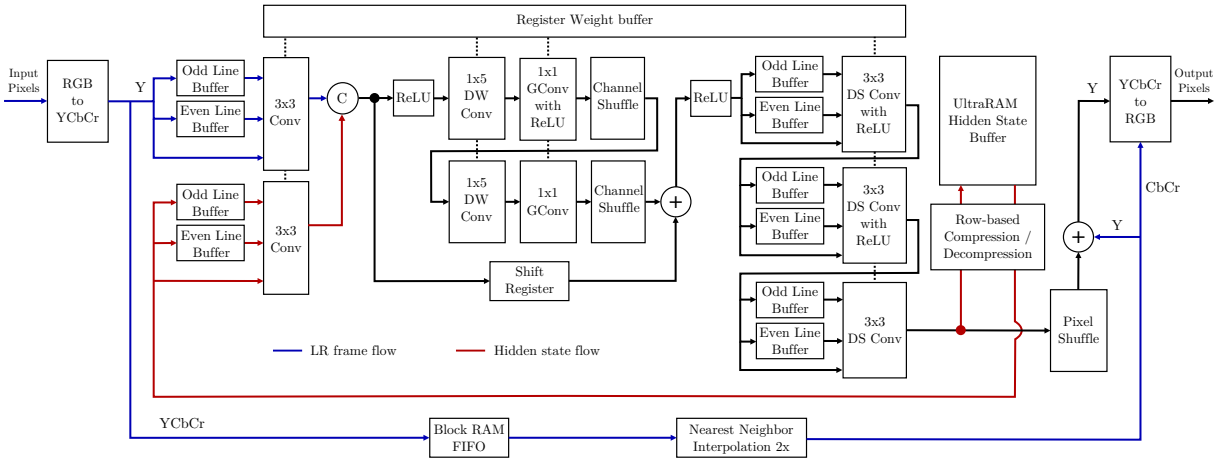


Figure 5.5: Block diagram of the proposed ERVSR. The LR input flow is depicted by blue arrows and the recurrent hidden state flow is denoted in red.

5.3 Hardware Implementation

5.3.1 Overview

In this section, the hardware implementation of ERVSR is outlined as illustrated in Fig. 5.5 where the blue and red arrows indicate respectively the LR frame flow and the recurrent hidden state flow. The proposed ERVSR architecture was implemented using Vitis HLS 2020.1. The C/RTL co-simulation is performed in HLS and place-and-route is conducted in Vivado to verify the functionality and evaluate the resource utilization and timing of the design. Particularly, the network implementation contains several main blocks including the standard convolution, depthwise convolution, pointwise convolution, rectified linear unit (ReLU), channel shuffle, pixel shuffle, NN interpolation, RGB/YCbCr, YCbCr/RGB, line buffer, weight buffer, and hidden state buffer. The incoming LR frame is processed line-by-line in a stream-based manner. Odd and even line buffers are utilized to store the coming streams for performing 3×3 convolutions. The convolution operations are parallelized along the channel dimension such that the elements at the same spatial location across feature maps in each layer can be obtained simultaneously and hence, the required amount of multiplications equals the number of network parameters. The overall design is pipelined within and between layers so that the proposed system can output r^2 reconstructed HR pixels in each clock cycle.

5.3.2 Implementation Details

Firstly, the model parameters are loaded from the register buffer. The RGB pixels are converted to YCbCr ones. To perform 3×3 convolution for the Y channel, two line buffers combining with the input stream are required. As the hidden state has four channels for $r = 2$ and each channel needs two line buffers so that ten line buffers are utilized for the two 3×3 convolutions. The output of the two branches are concatenated and propagated to RB and the other sequential blocks following Fig. 5.2. The hidden state is passed to the pixel shuffle and meanwhile compressed and stored in the UltraRAM (URAM). The NN interpolated Y channel is added to the output of the pixel shuffle to construct the super-resolved luminance channel. Combining the super-resolved Y channel with the NN interpolated CbCr channels, YCbCr pixels are converted to RGB ones. Note that the fixed-point quantization of the weights is conducted offline and the quantization of the activations is performed on the fly in the HLS implementation. Particularly, we choose $WL = 12, IL = 3$ for the weights and $WL = 16, IL = 5$ for the activations based on the simulation results shown in Section 5.4.6.

5.3.3 Buffer Allocation

The skip connection y_i^n embodied in function Net_o is implemented using a FIFO buffer in the block RAM (BRAM) and the local skip connection in RB is realized with a shift register. The quantized weights and the scale parameters b are stored in the register and BRAM, respectively. The hidden state buffer is a crucial component in the hardware design where parallel read/write access is required. Since the elements in each channel of the hidden state are read and written sequentially, r^2 FIFO buffers are used and each contains $N = W \times H$ elements. Due to the memory limitation of the BRAM in the target device, the on-chip URAM is employed for the storage of the hidden state.

5.4 Experiments and Results

In this section, the proposed ERVSR is analyzed and evaluated from different aspects. It should be noted that all the experiments employed the configuration of $q = 0.65$, namely 20 feature channels reserved for the LR input and the remaining 12 channels for the hidden

state. In particular, ERVSR is compared with the state-of-the-art FPGA-based VSR methods on the public VSR and SR datasets. In order to deal with static images using the pretrained ERVSR model, an inference strategy named self-initiation is introduced in Section 5.4.3. In Section 5.4.6, the effectiveness of the channel modulation coefficient, the statistical normalization scheme, and the recurrent hidden state are individually analyzed. Last but not least, an ablation study of the critical components of ERVSR is conducted and the results are shown in Section 5.4.6.

5.4.1 Datasets

Training Dataset: ERVSR was trained based on the publicly available VSR dataset Vimeo 90K [179] which consists of 91701 sequences with the frame resolution of 448×256 and the sequence length of 7 frames. Due to the lightweight structure of ERVSR, a reduced version of Vimeo 90K was utilized by randomly selecting 2×10^4 sequences as the GT where 1.4×10^4 sequences were employed for training and the rest were used for validation. Specially, random regions of size 128×128 were cropped from the GT sequences and downsampled the cropped regions using bicubic interpolation by a scaling factor of $r = 2$ to generate the LR counterparts. Standard data augmentation techniques such as flipping, rotation, and scaling were employed during the training to improve the generalizability of the network.

Testing Datasets: In the testing phase, multiple public VSR datasets were utilized including Vid4 [108], SPMCS30 [180], and UDM10 [178]. Particularly, Vid4 consists of four sequences and each of the sequences contains around 30 to 50 frames with the image resolution of 720×576. SPMCS30 is composed of 30 sequences with the frame size of 960×540. UDM10 is a recently published VSR dataset which contains 10 sequences with the frame resolution of 1272×720. As image SR can be considered as the specific VSR case with the sequence length of one, to explore the potential of the pretrained ERVSR for static image SR, the well-trained ERVSR was applied on the widely used SR datasets Set5 [181], Set14 [182], BSDS100 [183], Urban100 [184], and Manga109 [185] by using the introduced self-initiation strategy.

5.4.2 Training Details

During the training phase, Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$ was utilized as the optimization algorithm. The initial learning rate was set as 5×10^{-4} and multiplied by 0.2 every 10 epochs over 150 epochs until it declines to 1×10^{-6}. The mini-batch size was set as 4. The model was trained on a NVIDIA GeForce GTX1070 GPU and Intel(R) i5 2500 CPU and the training took approximately 12 hours.

5.4.3 Self-Initiation for Image SR

The proposed ERVSR is devised to make use of the high-frequency information of the previous frames to estimate the current residual based on a recurrent architecture. As proposed in the pioneering work [39,175], the recurrent input for the first frame is initialized with zero indicating no prior information. The initialization scheme performs well for VSR but seems to be inefficient for image SR reconstruction. In order to achieve promising performance for single image SR using ERVSR which is pretrained on the video dataset, an inference strategy named self-initiation which leverages the generated high-frequency information in the hidden state as the residual prior for itself is introduced. Specially, the LR image is duplicated to construct a dummy video stream of two frames. The first frame is dedicated to generating the high-frequency details which are propagated forward by the hidden state and serve as the image prior for the second frame. The output of the second frame is the estimated HR image. In fact, it is found that employing more than two frames makes no further performance improvement. In Section 5.4.6, it is demonstrated that comparing to the well-trained non-recurrent architecture dedicated to image SR, the proposed self-initiation scheme enables the pretrained ERVSR for static image SR to achieve an even improved image quality.

5.4.4 Evaluation Metrics

To quantitatively assess the image quality reconstructed by ERVSR, peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) are adopted as the evaluation metrics. Besides, the behavior of the investigated VSR methods on temporal consistency is demonstrated and evaluated based on visual perception. It should

be noted that all the quantitative evaluations are performed on the luminance channel.

5.4.5 Comparison with State-of-the-Arts

The proposed ERVSR is compared with the recently published hardware-efficient VSR methods including Lee et al. [186], Yang et al. [169], Kim et al. [170], Chang et al. [173], and Kim et al. [112] quantitatively in PSNR and SSIM, qualitatively in visual perception, and also in hardware implementations.

Quantitative Evaluation

In order to benchmark the proposed ERVSR with the existing state-of-the-art methods, experiments on multiple public VSR and SR datasets were carried out. Since the state-of-the-art FPGA-based VSR methods are only evaluated on the SR datasets in their original papers and there is no source code available, to conduct comparison on the VSR datasets, the models of Kim et al. [112] and Chang et al. [173] were reimplemented in Pytorch and their networks were trained carefully on a GPU following their papers. The results of the investigated VSR methods are summarized in Table 5.2 where their published results are marked by asterisks. It is shown that although ERVSR utilizes the recurrent architecture, due to the compact network design, it has even fewer parameters than Kim [112]. The fidelity of the reimplementations were validated and the performance of the reimplemented model of Kim [112] is demonstrated by the 32-bit floating point representations. It is shown that the reproduced model of Kim [112] generates almost the same performance as their published ones on the SR datasets. More importantly, the proposed ERVSR performs better on the VSR datasets than the other representative methods by a large margin in PSNR and SSIM. With regard to the SR datasets, by means of the proposed self-initiation scheme, ERVSR performs better especially on the recently published dataset Manga109. In fact, comparing to the floating-point performance, there is limited degradation in the hardware implementation due to the quantization effect in the fixed-point representations.

Table 5.2: Benchmark comparison on different VSR/SR datasets in average PSNR/SSIM. 32-bit Floating-point (FlP) experiments were conducted by PyTorch on a GPU device. Fixed-point (FxP) results were obtained from the Vitis HLS C simulation on a CPU device. The asterisk symbols represent the published results in their original papers.

Method	Yang [169]		Kim [170]	Kim [112]		Chang [173]		ERVSR	
# Parameters	-		-	2.56K		2.32K		2.54K	
Weight	FlP	FxP	FxP	FlP	10bit FxP	FlP	13bit FxP	FlP	12bit FxP
Activation	FlP	FxP	FxP	FlP	14bit FxP	FlP	13bit FxP	FlP	16bit FxP
VSR datasets	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM
Vid4	- -	- -	- -	30.62 0.9169	- -	30.63 0.9159	- -	30.99 0.9222	30.94 0.9209
SPMCS30	- -	- -	- -	35.16 0.9527	- -	34.98 0.9501	- -	35.30 0.9535	35.26 0.9529
UMD10	- -	- -	- -	42.08 0.9840	- -	41.67 0.9823	- -	42.40 0.9843	42.32 0.9840
SR datasets	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM
Set5	34.00* -	33.83* -	34.78* 0.9460*	36.67 0.9552	36.64* 0.9543*	36.47 0.9532	36.40* 0.9527*	36.80 0.9556	36.76 0.9553
Set14	29.97* -	29.77* -	31.63* 0.9083*	32.51 0.9076	32.47* 0.9070*	32.38 0.9059	32.21* 0.9047*	32.53 0.9079	32.51 0.9076
BSDS100	- -	- -	30.48* 0.8776*	31.33 0.8887	31.31* 0.8877*	31.23 0.8873	31.15* 0.8858*	31.32 0.8891	31.31 0.8887
Urban100	- -	- -	- -	29.30 0.8954	29.32* 0.8939*	29.05 0.8891	- -	29.32 0.8957	29.30 0.8952
Manga109	- -	- -	- -	35.37 0.9676	- -	35.03 0.9639	- -	35.83 0.9684	35.78 0.9682

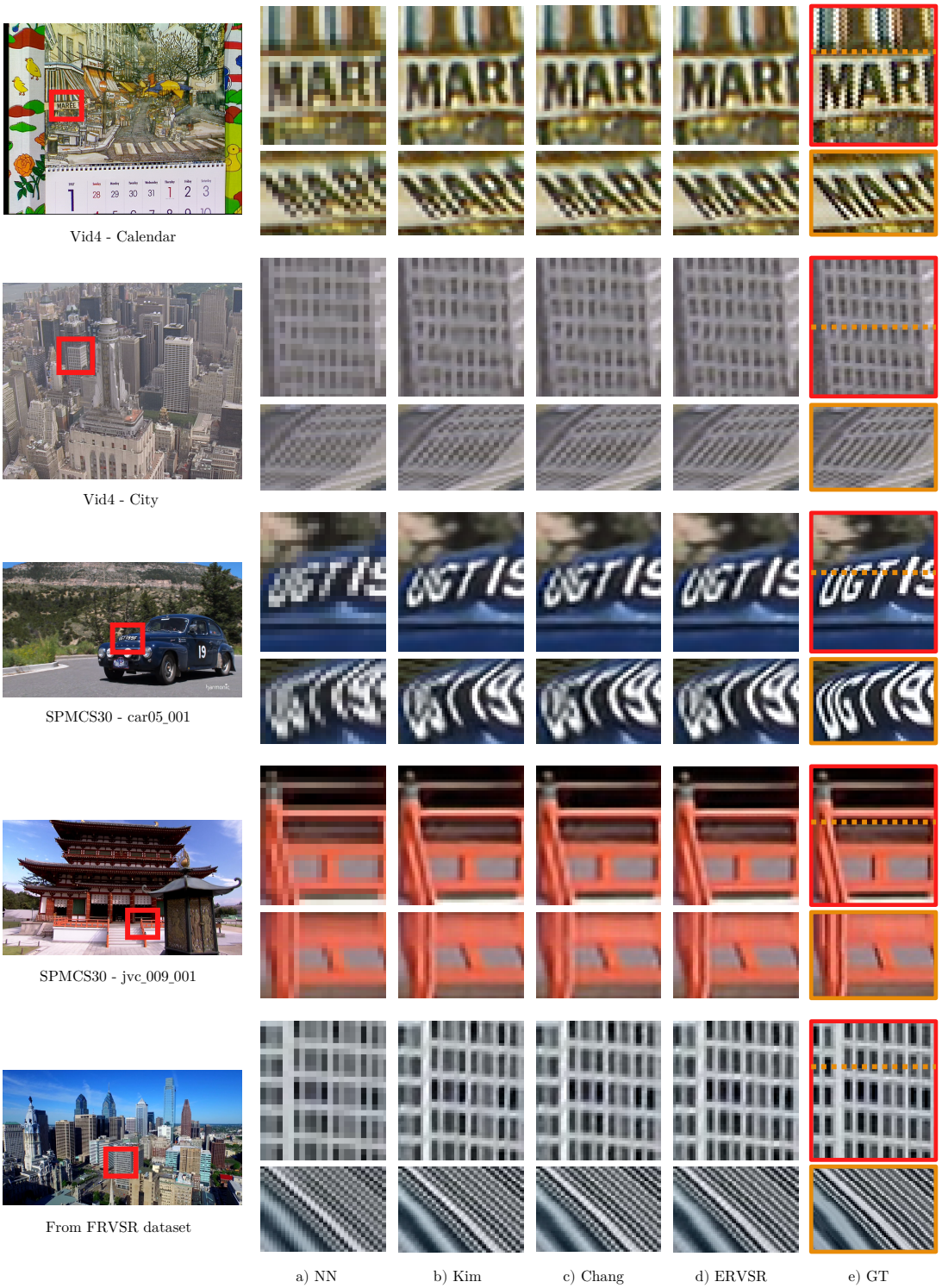


Figure 5.6: Visual comparison on different video sequences for an upscaling factor of 2. Top row: Region of interest of the reconstructed HR frames by multiple representative FPGA-based VSR methods. Bottom row: Temporal profiles. a) Nearest neighbor interpolation; b) Kim et al. [112]; c) Chang et al. [173]; d) The proposed ERVSR; e) Ground truth.

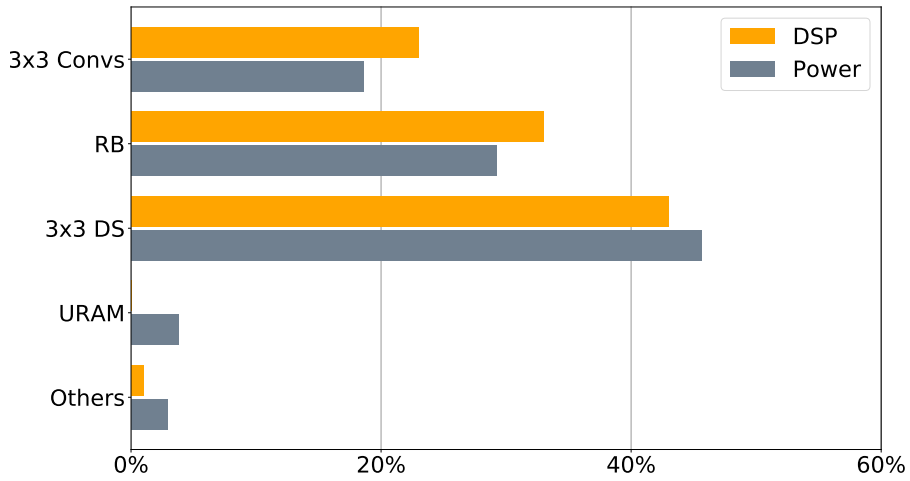


Figure 5.7: Power breakdown and DSP block usage of ERVSR.

Temporal Consistency

For VSR applications, an important evaluation criteria is the temporal consistency which significantly influences the visual perception of the video streams. In order to demonstrate the effectiveness of the recurrent architecture on the temporal consistency, The temporal profile of multiple video sequences is shown in Fig. 5.6. Specially, a row profile from a chosen region in each frame of the sequence is extracted and all the extracted profiles are stacked along the temporal dimension as depicted in the orange rectangles. Generally, flickering in video streams appears as jitter and jagged lines in the temporal profile. Sharp and continuous stacked profiles indicate good temporal consistency. It is shown that comparing to the other investigated approaches, the proposed ERVSR improves not only the temporal consistency but also provides a pleasant visual perception with sharper structures.

Hardware Efficiency

The proposed ERVSR was implemented using Vitis HLS 2020.1 targeting a Kintex UltraScale FPGA XCKU15P. The hardware configurations of all the studied methods are listed in Table 5.3. As shown, based on the efficient pipe-line design discussed in Section 5.3, ERVSR achieves a target operating frequency of 160 MHz and a throughput of 637 Mpixel/s

Table 5.3: Characteristics of hardware implementations of multiple investigated VSR methods.

	Lee [186]	Yang [169]	Kim [170]	Kim [112]	Chang [173]	ERVSR
FPGA device / CMOS Tech.	13μm	Altera EP4SGX530	Kintex XCKU040	Kintex XCKU040	Kintex XC7K410T	Kintex XCKU15P
Implementation	-	-	-	System Verilog	Vivado HLS 2016.4	Vitis HLS 2020.1
Supported Scales	2, 3	2	2	2	2, 3, 4	2
Methods	Sharpening Lagrange	ANR	HSI	CNN	CNN	RNN
Output resolution	3840x2160	1920x1080	3840x2160	3840x2160	2880x1280	3840x2160
Max. Frequency(MHz)	431	136	150	150	130	160
DSP Usage	-	-	108	1920	1512	1820
LUTs Usage	-	-	3395	110K	167K	98K
FFs Usage	-	-	1952	102K	158K	57K
Memory Size BRAM(Bytes)	-	232K	92K	392K	945K	666K
Memory Size URAM(Bytes)	-	-	-	-	-	4176K
Cycles	-	-	-	-	2074K	2083K
Power (W)	-	-	-	4.79	5.38	5.47
Throughput (Mpixels/s)	431	124	600	600	520 (S=2)	637
Power Efficiency (Mpixels/J)	-	-	-	125.2	96.3	111.0

which supports 76 fps for UHD videos. The power breakdown was evaluated using Xilinx Report Power by isolating the sub-blocks including RB, 3×3 DS Convs, 3×3 Convs, and URAM. In Fig. 5.7, the power breakdown and the DSP block usage of the sub-blocks are depicted. According to the vector (SAIF) based power estimation in Xilinx Report Power, the total on-chip power dissipation is 5.47W which is comparable to Chang et al. [173] although ERVSR contains the recurrent hidden state stored in the URAM. Specially, the power dissipation of 3×3 DS Convs and RB are respectively 44.1% and 28.3% of the overall on-chip power consumption and the URAM consumes 3.6%. Besides, it is shown

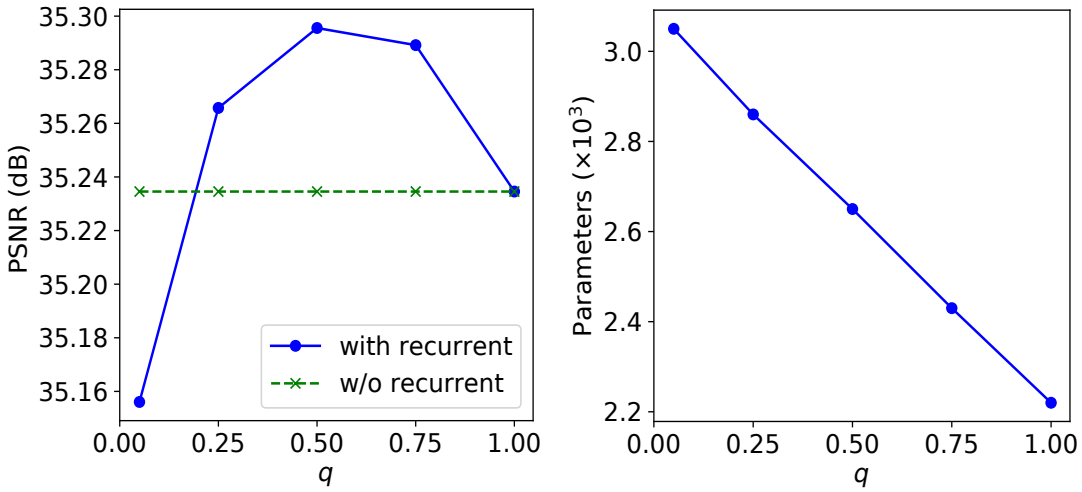


Figure 5.8: Analysis of q with $1/32 \leq q \leq 1$ on the VSR dataset SPMCS30 [180]. Left: Performance in average PSNR; Right: Number of model parameters.

that the DSP blocks are mostly utilized by the 3×3 DS Convs (42.9%), RB (33.0%), and 3×3 Convs (23.3%). Comparing to the GPU implementation of ERVSR which takes 13 ms for an UHD output on a NVIDIA GeForce GTX1070 device, the proposed FPGA implementation achieves a speed-up of more than 5×.

5.4.6 Model Analysis

Channel Modulation Coefficient

In order to evaluate the effectiveness of the proposed channel modulation coefficient q, experimental analysis on a wide spread of q in the range of $[1/32, 1]$ was performed. In Fig. 5.8, we demonstrate the impact of q on the image quality assessed in PSNR and the model complexity depicted by the number of network parameters. The blue curve represents the performance of ERVSR with different q and the green dotted line denotes the model variant without recurrent architecture ($q = 1$). As shown in the left graph, the PSNR is not monotonically related with q which indicates that a balanced allocation of the feature channels to the input frame and the hidden state is important. In the right graph, it is shown that as q increases, the overall amount of network parameters declines.

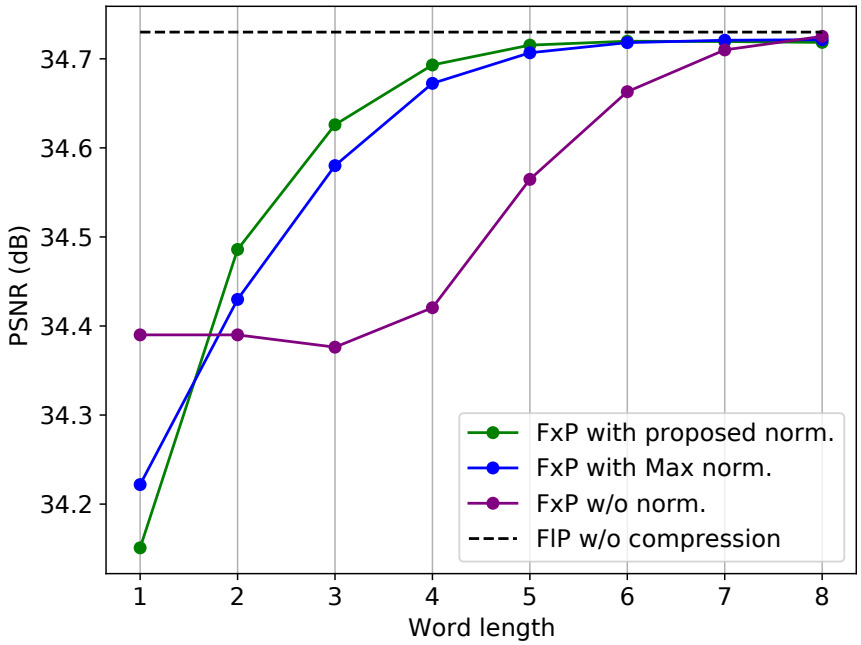


Figure 5.9: Analysis of the proposed normalization scheme under different WL with $IL = 1$ in PSNR on the sequence jvc_009_001 from SPMCS30 [180].

In the implementation, q was set as 0.65 for a tradeoff between the image quality and the model complexity.

Normalization and Quantization

An efficient compression of the hidden state plays an important role for the hardware implementation of ERVSR. In Fig. 5.9, we demonstrate the effectiveness of the proposed normalization scheme comparing to the Max normalization and without normalization. Besides, the floating-point implementation without compression is involved as the reference. It should be noted that IL was set as 1 for all the WL variants. It can be seen that normalization has a noticeable influence. The proposed normalization scheme performs better than the others especially for lower WL. For $WL \geq 6$, there is almost no deviation from the floating-point reference. In this work, WL was chosen as 4 to fit the size of the URAM in the target FPGA device. The impact of the quantization of the weights and the activations on the FPGA implementation is illustrated in Fig. 5.10. To preserve

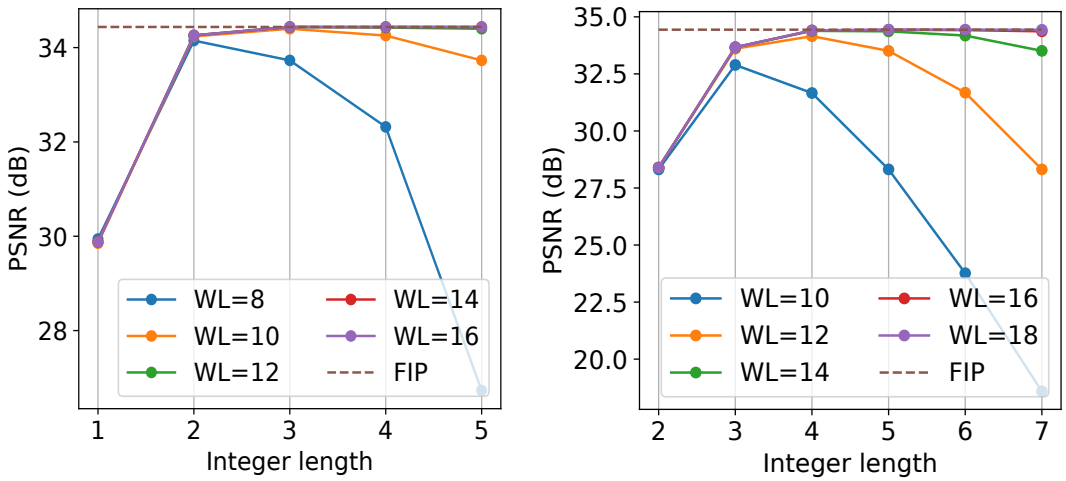


Figure 5.10: Performance of different quantization variants of the weights and the activations on the sequence jvc_009_001 from SPMCS30 [180]. Left: Quantization of the weights; Right: Quantization of the activations.

the model accuracy, $WL = 12, IL = 3$ are for the weights and $WL = 16, IL = 5$ are for the activations.

Information Flow

In order to analyze the effectiveness of the recurrent input, the information flow is demonstrated in Fig. 5.11. Particularly, the black curve and the red curve indicate the PSNR of a video sequence reconstructed from the $1st$ and the $10th$ frame, respectively. It can be clearly observed that the black curve performs better than the red one and the gap between the two curves still exists at the $30th$ frame because the black curve has accumulated temporal information over 10 more frames. From another perspective, it can be inferred that the hidden state transmits the temporal dependency over 30 frames. In fact, ERVSR improves the temporal consistency and visual perception silently without explicitly using additional LR frames and motion compensation.

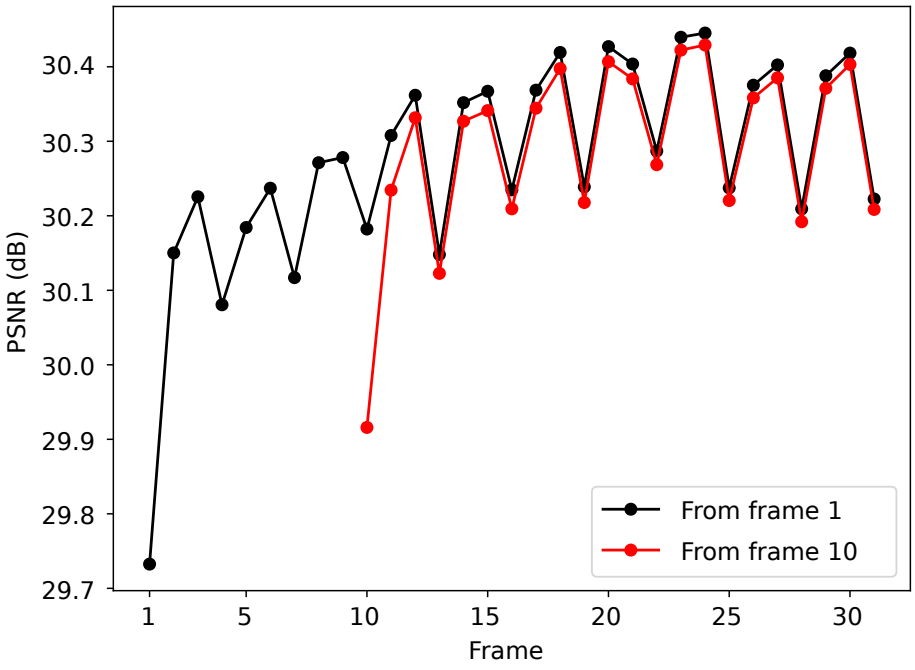


Figure 5.11: Information flow over time on the sequence hdclub_008_007 from SPMCS30 [180]. The black curve starts at the $1th$ frame and the red one begins 10 frames later.

Table 5.4: Ablation study of ERVSR on the VSR dataset SPMCS30 with $q = 0.65$, $WL = 4, IL = 1$.

	SPMCS30			
Recurrent	✗	✓	✓	✓
Normalization	✗	✗	✗	✓
Quantization	✗	✗	✓	✓
PSNR/SSIM	35.21/0.9524	35.29/0.9533	35.08/0.9515	35.26/0.9529

Table 5.5: Ablation study of ERVSR on the SR datasets BSDS100 and Manga109 with $q = 0.65$, $WL = 4, IL = 1$.

	BSDS100			Manga109		
Recurrent	✗	✓	✓	✗	✓	✓
Self-Initiation	✗	✗	✓	✗	✗	✓
PSNR/SSIM	31.30 0.8883	31.27 0.8881	31.31 0.8887	35.63 0.9673	35.53 0.9672	35.78 0.9682

Ablation Study

An ablation study was carried out to quantify the impact of the recurrent architecture and the proposed normalization scheme based on the dataset SPMCS30. The results are depicted in Table 5.4. Comparing the first two columns, it is shown that the recurrent architecture improves the performance by 0.08dB in average PSNR. Observing the last two columns, it can seen that the proposed normalization scheme achieves an average gain of 0.18dB. The second column obtains a slightly better PSNR than the last column because it employs 16-bit fixed-point representations and no data compression is performed. Additionally, the recurrent structure and the proposed self-initiation were evaluated on the SR datasets BSDS100 and Manga109. The results of both datasets are summarized in Table 5.5. From the first two columns of both datasets, it is observed that without self-initiation, the recurrent network performs slightly worse than the non-recurrent one since the recurrent input is initialized with zero. Comparing the last two columns in each dataset, it is shown that self-initiation improves the performance of ERVSR silently by estimating the HR residuals from the input frame itself. Besides, it is shown that for BSDS100, ERVSR with self-initiation achieves comparable image quality as the non-recurrent variant which is dedicated to image SR. In Manga109, ERVSR obtains an average performance gain of 0.15dB in PSNR by self-initiation.

5.5 Conclusion

In this chapter, a hardware-efficient residual recurrent neural network ERVSR for real-time VSR on FPGA is presented. Different from the current state-of-the-art FPGA-based

VSR methods which perform SISR in a sliding-window fashion over the video sequence, the proposed ERVSR exploits the input frame and the inter-frame temporal correlation encoded in the recurrent hidden state to preserve temporal consistency. ERVSR is intended to have a compact recurrent architecture to fulfill the hardware requirements. Experimental results demonstrate that ERVSR outperforms the investigated state-of-the-art FPGA-based VSR models by an average gain of 0.28dB in PSNR over multiple VSR datasets without compromising the data throughput. It is shown that ERVSR improves the temporal consistency and visual perception silently without using additional LR frames or motion compensation by exploiting the recurrent input which conveys the long-term temporal information of more than 30 previous frames.

Chapter 6

Conclusion

This thesis covers the proposal of SR algorithms which are appropriate for CT imaging and image processing mainly from the perspective of SR reconstruction in the presence of noise, SR acceleration for high-resolution images, and influence of SR on image registration.

The contribution of this thesis starts with the proposal of an iterative multi-image super-resolution (MISR) algorithm MPG+BSWTV for noisy images. To jointly super resolve the image and suppress the noise, a SR model which more accurately describes the noise characteristics is proposed and evaluated. Based on the Maximum A Posteriori (MAP) estimation, an objective function is derived from a Poisson-Gaussian noise model and an adaptive noise removal regularizer which leverages the gradual refinement mechanism. Extensive experiments show that the proposed method can effectively improve the spatial resolution of noisy images. We have benchmarked the proposed method on the public real-world dataset SupER. Comparing to the other 14 investigated optimization-based or learning-based SR methods, MPG+BSWTV achieves an average gain of 0.2dB in PSNR compared to the second best and provides better visual perception.

The contribution of the second chapter is the deployment of SR algorithm for real-time CT imaging by resorting to multi-GPU acceleration. The proposed FL-MISR approach can be seamlessly integrated into industrial CT scanners by super-resolving projections acquired by subpixel detector shift. SR reconstruction is performed in a distributed manner by data parallelism over multi-GPU systems which supports an on-the-fly resolution enhancement

of the projections without introducing extra computation time. Experiments show that FL-MISR can effectively improve the spatial resolution of CT systems. Comparing to a 56-core CPU implementation, FL-MISR achieves a more than 50× speedup on an off-the-shelf 4-GPU system. Besides, it is shown that super-resolving four input projections of size 4096×4096 by an upscaling of 2× can be achieved within 2.4s.

In addition to the above iterative MISR methods, a CNN-based light-weight resolution enhancement module (REM) is proposed to enhance the performance of other vision tasks such as image registration, semantic segmentation. Specially, REM can be easily plugged into other networks either by a straightforward cascade or by employing other coupling techniques such as auxiliary loss. REM has been applied to two registration networks FDRN and VoxelMorph at different scaling factors. Experiments on brain datasets show that the employment of REM enhances both the registration accuracy and visual quality especially when the input images suffer from poor spatial resolution.

Apart from the aforementioned optimization-based MISR methods and the CNN-based REM, which can be implemented on GPU systems, a hardware-efficient SR method ERVSR is proposed based on the residual recurrent neural network (RNN). The architecture of ERVSR is highly adapted to low-complexity hardware and supports an implementation on the field programmable gate array (FPGA). Comparing to a GPU implementation, the FPGA one achieves a speedup of more than 5×. Extensive experiments show that ERVSR performs better than the state-of-the-art FPGA-based SR methods by an average gain of 0.28dB in PSNR over multiple public datasets. In addition, ERVSR supports SR output of 8 Mpixel at 76 fps which shows a great potential for the deployment of hardware embedded SR in extreme fast CT applications such as inline-CT.

To summarize the contributions, this thesis focuses on SR enhancement for CT imaging and image processing. The acquired images of commercial CT scanners usually suffer from undesirable spatial resolution and noticable noise. Super-resolving noisy images is of indisputable significance for CT imaging. A dedicated SR algorithm concerning noisy images is presented in this thesis. Furthermore, the feasibilty of super-resolving projections acquired by subpixel detector shift for resolution enhancement of the CT system is demonstrated and a seamless integration of real-time SR into CT imaging has been achieved by multi-GPU acceleration. Additionally, it is shown that SR can be considered as a preprocessing routine for other tasks such as image registration, especially when

input images suffer from degraded resolution. Last but not least, a great potential for the employment of FPGA-based SR in CT imaging is revealed.

Bibliography

[1] O. K. Ersoy, Diffraction, Fourier optics and imaging. John Wiley & Sons, 2006, vol. 30.

[2] A. El Gamal and H. Eltoukhy, "Cmos image sensors," IEEE Circuits and Devices Magazine, vol. 21, no. 3, pp. 6–20, 2005.

[3] T. M. Buzug, Computed Tomography. Springer, Berlin, Heidelberg, 2008.

[4] C. E. Shannon, "A mathematical theory of communication," The Bell System Technical Journal, vol. 27, no. 3, pp. 379–423, 1948.

[5] S. Park, M. Park, and M. G. Kang, "Super-resolution image reconstruction: A technical overview," IEEE Signal Process. Mag., vol. 20, no. 5, pp. 21–36, 2003.

[6] K. Nasrollahi and T. B. Moeslund, "Super-resolution: A comprehensive survey," Mach. Vis. Appl., vol. 25, no. 6, pp. 1423–1468, 2014.

[7] W. Yang et al., "Deep learning for single image super-resolution: A brief review," IEEE Trans. Multimedia, vol. 21, no. 12, pp. 3106–3121, 2019.

[8] M. D. Robinson et al., Super-resolution imaging. CRC Press, 2017.

[9] C. Yang, J. Huang, and M. Yang, "Exploiting self-similarities for single frame super-resolution," in Proc. Asian Conf. Comput. Vis. Springer, 2010, pp. 497–510.

[10] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," ACM Trans. Graph., vol. 30, no. 2, pp. 1–11, 2011.

[11] E. Plenge, D. H. J. Poot, W. J. Niessen, and E. Meijering, "Super-resolution reconstruction using cross-scale self-similarity in multi-slice mri," in Proc. Int. Conf. Med. Imag. Comp. Comput. Assist. Interv. Springer, 2013, pp. 123–130.

[12] J. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 5197–5206.

[13] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in Proc. Eur. Conf. Comput. Vis., 2014, pp. 184–199.

[14] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in Proc. Eur. Conf. Comput. Vis., 2016, pp. 391–407.

[15] R. Timofte, V. D. Smet, and L. V. Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in Asian Conference on Computer Vision, 2014, pp. 111–126.

[16] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 1637–1645.

[17] B. Lima, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, 2017, pp. 136–144.

[18] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 1646–1654.

[19] C. Lediga et al., "Photo-realistic single image super-resolution using a generative adversarial network," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 105–114, [doi: 10.1109/CVPR.2017.19].

[20] Y. Zhang et al., "Image super-resolution using very deep residual channel attention networks," in Proc. Eur. Conf. Comput. Vis., 2018, pp. 286–301.

[21] X. Wang et al., "ESRGAN: Enhanced super-resolution generative adversarial networks," in Proc. Eur. Conf. Comput. Vis., 2018, pp. 1–16.

[22] K. Zhang, W. Zuo, and L. Zhang, "Deep plug-and-play super-resolution for arbitrary blur kernels," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1671–1681.

[23] T. S. Huang and R. Y. Tsan, "Multiple frame image restoration and registration," Advances in Computer Vision and Image Process., JAI Press, Inc. Greenwich, CT, pp. 317–339, 1984.

[24] H. Stark and P. Oskoui, "High resolution image recovery from image-plane arrays, using convex projections," J. Opt. Soc. Am. A., vol. 6, pp. 1715–1726, 1989.

[25] M. Elad and Y. Hel-Or, "A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur," IEEE Trans. Image Process, vol. 10, no. 8, pp. 1187–1193, 2001.

[26] V. Patanavijit and S. Jitapunkul, "A lorentzian stochastic estimation for a robust iterative multiframe super-resolution reconstruction with lorentzian-tikhonov regularization," Journal on Advances in Signal Process., vol. 2007, no. 1, pp. 1–21, 2007.

[27] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super-resolution," IEEE Trans. Image Process., vol. 13, no. 10, pp. 1327–1344, 2004.

[28] Q. Yuan, L. Zhang, and H. Shen, "Regional spatially adaptive total variation super-resolution with spatial information filtering and clustering," IEEE Trans. Image Process., vol. 22, no. 6, pp. 2327–2342, 2013.

[29] X. Zeng and L. Yang, "A robust multiframe super-resolution algorithm based on half-quadratic estimation with modified btv regularization," Digital Signal Process., vol. 23, no. 1, pp. 98–109, 2013.

[30] L. Yue, H. Shen, Q. Yuan, and L. Zhang, "A locally adaptive l1-l2 norm for multi-frame super-resolution of images with mixed noise and outliers," Signal Process., vol. 105, no. 1, pp. 156–174, 2014.

[31] T. Köhler, J. Jordan, A. Maier, and J. Hornegger, "A unified bayesian approach to multi-frame super-resolution and single-image upsampling in multi-sensor imaging," in Proc. 26th British Machine Vision Conference, 2015, pp. 1–12.

[32] T. Köhler et al., "Robust multiframe super-resolution employing iteratively re-weighted minimization," IEEE Trans. Comput. Imag., vol. 2, no. 1, pp. 42–58, 2016.

[33] K. Sun, T. Tran, R. Krawtschenko, and S. Simon, "Multi-frame super-resolution reconstruction based on mixed Poisson–Gaussian noise," Signal Process. Image Commun., vol. 82, p. 115736, 2020.

[34] K. Sun and S. Simon, "Bilateral spectrum weighted total variation for noisy-image super-resolution and image denoising," arXiv preprint arXiv:2106.00768, pp. 1–13, 2021.

[35] K. Sun, T.-H. Tran, J. Guhathakurta, and S. Simon, "FL-MISR: Fast Large-Scale Multi-Image Super-Resolution for Computed Tomography Based on Multi-GPU Acceleration," arXiv preprint arXiv: 2108.04315, pp. 1–12, 2021.

[36] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Video super-resolution with convolutional neural networks," IEEE Trans. Comput. Imag., vol. 2, no. 2, pp. 109–122, 2016.

[37] K. W. Hung, C. Qiu, and J. Jiang, "Video Super Resolution via Deep Global-Aware Network," IEEE Access, vol. 7, pp. 74 711–74 720, 2019.

[38] Y. Huang, W. Wang, and L. Wang, "Video Super-Resolution via Bidirectional Recurrent Convolutional Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 1015–1028, Apr. 2018.

[39] M. S. M. Sajjadi, R. Vemulapalli, and M. Brown, "Frame-Recurrent Video Super-Resolution," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6626–6634.

[40] M. Haris, G. Shakhnarovich, and N. Ukita, "Recurrent back-projection network for video super-resolution," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 3897–3906.

[41] J. S. Isaac and R. Kulkarni, "Super resolution techniques for medical image processing," in Proc. Int. Conf. Technol. for Sustainable Development. IEEE, 2015, pp. 1–6.

[42] M. Kachelrieß, M. Knaup, C. Penßel, and W. Kalender, "Flying focal spot (ffs) in cone-beam ct," IEEE Trans Nucl. Sci., vol. 53, no. 3, pp. 1238–1247, 2006.

[43] M. Chang, Y. Xiao, and Z. Chen, "Improve spatial resolution by modeling finite focal spot (mffs) for industrial ct reconstruction," Opt. express, vol. 22, no. 25, pp. 30 641–30 656, 2014.

[44] Y. Zhu et al., "An approach to increasing the resolution of industrial ct images based on an aperture collimator," Opt. express, vol. 21, no. 23, pp. 27 946–27 963, 2013.

[45] H. B. Thibault, K. D. Sauer, C. A. Bouman, and J. Hsieh, "A three-dimensional statistical approach to improved image quality for multislice helical ct," Med. Phys., vol. 34, no. 11, pp. 4526–4544, 2007.

[46] W. V. A. et al., "Super-resolution for computed tomography based on discrete tomography," IEEE Trans. Image Process., vol. 23, no. 3, pp. 1181–1193, 2014.

[47] G. Zang et al., "Super-resolution and sparse view ct reconstruction," in Proc. Eur. Conf. Comput. Vis., 2018, pp. 137–153.

[48] J. Park et al., "Computed tomography super-resolution using deep convolutional neural network," Phys. Med. Biol., vol. 63, no. 14, p. 145011, 2018.

[49] Y. Wang et al., "Ct-image of rock samples super resolution using 3d convolutional neural network," Computers & Geosciences, vol. 133, p. 104314, 2019.

[50] Y. Wang, S. S. Rahman, and C. H. Arns, "Super resolution reconstruction of μ-ct image of rock sample using neighbour embedding algorithm," Physica A: Statistical Mechanics and its Applications, vol. 493, pp. 177–188, 2018.

[51] Y. Zhang et al., "Reconstruction of super-resolution lung 4d-ct using patch-based sparse representation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2012, pp. 925–931.

[52] C. Jiang, Q. Zhang, R. Fan, and Z. Hu, "Super-resolution ct image reconstruction based on dictionary learning and sparse representation," Scientific reports, vol. 8, no. 1, pp. 1–10, 2018.

[53] C. You et al., "Ct super-resolution gan constrained by the identical, residual, and cycle learning ensemble (gan-circle)," IEEE Trans. Med. Imag., vol. 39, no. 1, pp. 188–203, 2019.

[54] P. Gu et al., "Low-dose computed tomography image super-resolution reconstruction via random forests," Sensors, vol. 19, no. 1, p. 207, 2019.

[55] D. L. Snyder, A. M. Hammoud, and R. L. White, "Image recovery from data acquired with a charge-coupled-device camera," J. Opt. Soc. Am. A, vol. 10, no. 5, pp. 1014–1023, 1993.

[56] C. Aguerrebere, J. Delon, Y. Gousseau, and P. Musé, "Study of the digital camera acquisition process and statistical modeling of the sensor raw data," hal-00733538v4, pp. 1–11, 2013.

[57] F. Luisier, T. Blu, and M. Unser, "Image denoising in mixed poisson-gaussian noise," IEEE Trans. Image Process., vol. 20, no. 3, pp. 696–708, 2011.

[58] T. Köhler et al., "Toward bridging the simulated-to-real gap: Benchmarking super-resolution on real data," IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 11, pp. 2944–2959, 2019.

[59] K. Sun and S. Simon, "FDRN: A fast deformable registration network for medical images," Med. Phys., vol. early access, pp. 1–11, 2021.

[60] K. Sun et al., "An FPGA-based residual recurrent neural network for real-time video super-resolution," IEEE Trans. Circuits Syst. Video Technol., vol. early access, pp. 1–12, 2021.

[61] J. Li, Z. Shen, R. Yin, and X. Zhang, "A reweighted l2 method for image restoration with poisson and mixed poisson-gaussian noise," Inverse Probl. Imaging, vol. 9, no. 3, pp. 875–894, 2015.

[62] G. Demoment, "Image reconstruction and restoration: overview of common estimation structures and problems," IEEE Trans. Acoust., Speech, Signal Process., vol. 37, no. 12, pp. 2024–2036, 1989.

[63] M. R. Banham and A. K. Katsaggelos, "Digital image restoration," IEEE Signal Process. Magazine, vol. 14, no. 2, pp. 24–41, 1997.

[64] S. P. Kim, N. K. Bose, and H. M. Valenzuela, "Recursive reconstruction of high resolution image from noisy undersampled multiframes," IEEE Trans. Acoust., Speech, Signal Process., vol. 38, no. 6, pp. 1013–1027, 1990.

[65] S. P. Kim and W. Su, "Recursive high-resolution reconstruction of blurred multiframe images," IEEE Trans. Image Process., vol. 2, no. 4, pp. 534–539, 1993.

[66] T. Komatsu, K. Aizawa, T. Igarashi, and T. Saito, "Signal-processing based method for acquiring very high resolution image with multiple cameras and its theoretical analysis," in Proc. Inst. Elec. Eng., vol. 140, no. 1, 1993, pp. 19–25.

[67] H. Ur and D. Gross, "Improved resolution from sub-pixel shifted pictures," CVGIP: Graphical Models and Image Process., vol. 54, pp. 181–186, 1992.

[68] A. M. Tekalp, M. K. Ozkan, and M. I. Sezan, "High-resolution image reconstruction from lower-resolution image sequences and space varying image restoration," in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process., vol. 3, 1992, pp. 169–172.

[69] N. A. El-Yamany and P. E. Papamichalis, "Robust color image super-resolution: An adaptive m-estimation framework," Journal on Image and Video Process., vol. 2008, no. 1, pp. 1–12, 2008.

[70] H. Song, L. Zhang, P. Wang, K. Zhang, and X. Li, "An adaptive l1-l2 hybrid error model to super-resolution," in IEEE Int. Conf. Image Process., 2010, pp. 2821–2824.

[71] R. R. Schultz and R. L. Stevenson, "A bayesian approach to image expansion for improved definition," IEEE Trans. Image Process., vol. 3, no. 3, pp. 233–242, 1994.

[72] R. Schultz and R. Stevenson, "Extraction of high-resolution frames from video sequences," IEEE Trans. Image Process., vol. 5, no. 6, pp. 996–1011, 1996.

[73] J. Zhang, C. Zhao, R. Xiong, S. Ma, and D. Zhao, "Image super-resolution via dual-dictionary learning and sparse representation," in IEEE International Symposium on Circuits and Systems, 2012, pp. 1688–1691.

[74] E. Lee and M. Kang, "Regularized adaptive high-resolution image reconstruction-considering inaccurate subpixel registration," IEEE Trans. Image Process., vol. 12, pp. 806–813, 2003.

[75] H. He and L. Kondi, "An image super-resolution algorithm for different error levels per frame," IEEE Trans. Image Process., vol. 15, pp. 592–603, 2006.

[76] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," Physica D: Nonlinear Phenomena, vol. 60, no. 1-4, pp. 259–268, 1992.

[77] M. E. Zervakis and A. N. Venetsanopoulos, "Linear and nonlinear image restoration under the presence of mixed noise," IEEE Trans. Circ. Syst., vol. 38, no. 3, pp. 258–272, 1991.

[78] J. F. Cai, R. H. Chan, and M. Nikolova, "Fast two-phase image deblurring under impulse noise," J. Math. Imaging Vis., vol. 36, pp. 46–53, 2010.

[79] A. Anastassopoulos and P. Vassilis, "Regularized super-resolution image reconstruction employing robust error norms," Optical Engineering, vol. 48, no. 11, pp. 117 004–1–117 004–14, 2009.

[80] Q. Gao et al., "Bayesian joint super-resolution, deconvolution, and denoising of images with poisson-gaussian noise," in 938-942, 2018.

[81] B. Bajić, J. Lindblad, and N. Sladoje, "Single image super-resolution reconstruction in presence of mixed poisson-gaussian noise," in Proc. IEEE Int. Conf. Image Process. Theory, Tools and Appl. (IPTA), 2016, pp. 1–6.

[82] Y. Traonmilin and C. Aguerrebere, "Simultaneous high dynamic range and superresolution imaging without regularization," SIAM Journal on Imaging Sciences, vol. 7, no. 3, pp. 1624–1644, 2014.

[83] W. Bao, X. Zhang, S. Yan, and Z. Gao, "Iterative convolutional neural network for noisy image super-resolution," in IEEE Int. Conf. Image Process., 2017, pp. 4038–4042.

[84] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 4799–4807.

[85] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., no. 9, Jul. 2017, pp. 2261–2269.

[86] X. Li, Y. Hu, X. Gao, D. Tao, and B. Ning, "A multi-frame image super-resolution method," Signal Process., vol. 90, no. 2, pp. 405–414, 2010.

[87] C. Aguerrebere, J. Delon, Y. Gousseau, and P. Musé, "Single shot high dynamic range imaging using piecewise linear estimators," in Proc. Int. Comp. Photography, 2014, pp. 1–10.

[88] D. L. Snyder, C. W. Helstrom, A. D. Lanterman, M. Faisal, and R. L. White, "Compensation for readout noise in ccd images," J. Opt. Soc. Am. A, vol. 12, no. 2, pp. 272–283, 1995.

[89] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical poissonian-gaussian noise modeling and fitting for single-image raw-data," IEEE Trans. Image Process., vol. 17, no. 10, pp. 1737–1754, 2008.

[90] P. J. Huber, "Robust estimation of a location parameter," The Annals of Mathematical Statistics, vol. 35, no. 1, pp. 73–101, mar 1964.

[91] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," Foundations and Trends® in Machine learning, vol. 3, no. 1, pp. 1–122, 2011.

[92] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," Mathematical programming, vol. 45, no. 1, pp. 503–528, 1989.

[93] I. A. Elbakri and J. A. Fessler, "Statistical image reconstruction for polyenergeticx-ray computed tomography," IEEE Trans. Med. Imag., vol. 21, no. 2, pp. 89–99, 2002.

[94] P. J. L. Riviere, J. Bian, and P. A. Vargas, "Penalized-likelihood sinogram restoration for computed tomography," IEEE Trans. Med. Imag., vol. 25, no. 8, pp. 1022–1036, 2006.

[95] I. A. Elbakri and J. A. Fessler, "Efficient and accurate likelihood for iterative image reconstruction in x-ray computed tomography," Proc. SPIE Image Process., vol. 5032, pp. 1839–1850, 2003.

[96] B. R. Whiting, "Signal statistics of x-ray computed tomography," Proc. SPIE Physics of Medical Imaging, vol. 4682, pp. 53–60, 2002.

[97] P. J. L. Riviere, "Penalized-likelihood sinogram smoothing for low-dose ct," Med. Phys., vol. 32, pp. 1676–1683, 2005.

[98] Q. Ding, Y. Long, X. Q. Zhang, and J. A. Fessler, "Statistical image reconstruction using mixed poisson-gaussian noise model for x-ray ct," in Proc. Intl. Mtg. on Image Formation in X-ray CT, 2016, pp. 399–402.

[99] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., vol. 2, 2005, pp. 60–65.

[100] M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the nonlocal-means to super-resolution reconstruction," IEEE Trans. Image Process., vol. 18, no. 1, pp. 36–51, 2008.

[101] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," Multiscale Modeling & Simulation, vol. 7, no. 3, pp. 1005–1028, 2009.

[102] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," IEEE Trans. Image Process., vol. 16, no. 2, pp. 349–366, 2007.

[103] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," IEEE Trans. Image Process., vol. 19, no. 11, pp. 2861–2873, 2010.

[104] K. K. In and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 6, pp. 1127–1133, 2010.

[105] J. Salvador and E. Perez-Pellitero, "Naive bayes super-resolution forest," in IEEE Int. Conf. Comput. Vis., 2015, pp. 325–333.

[106] M. Bätz, A. Eichenseer, and A. Kaup, "Multi-image super-resolution using a dual weighting scheme based on voronoi tessellation," in IEEE International Conference on Image Processing, 2016, pp. 2822–2826.

[107] M. Bätz, J. Koloda, A. Eichenseer, and A. Kaup, "Multi-image super-resolution using a locally adaptive denoising-based refinement," in International Workshop on Multimedia Signal Processing, 2016, pp. 1–6.

[108] C. Liu and D. Sun, "On bayesian adaptive video super resolution," IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 2, pp. 346–360, 2014.

[109] J. Wetzl et al., "GPU-accelerated time-of-flight super-resolution for image-guided surgery," in Bildverarbeitung für die Medizin 2013. Springer, 2013, pp. 21–26.

[110] J. Anger, T. Ehret, C. de Franchis, and G. Facciolo, "Fast and accurate multi-frame super-resolution of satellite images," ISPRS J. Photo. Remote Sensing, vol. 5, no. 1, pp. 1–8, 2020.

[111] J. Caballero et al., "Real-time video super-resolution with spatio-temporal networks and motion compensation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2848–2857.

[112] Y. Kim, J.-S. Choi, and M. Kim, "A Real-Time Convolutional Neural Network for Super-Resolution on FPGA With Applications to 4K UHD 60 fps Video Services," IEEE Trans. Circuits Syst. Video Technol., vol. 29, no. 8, pp. 2521–2534, Aug. 2019.

[113] S. Gu et al., "DIV8K: Diverse 8k resolution image dataset," in IEEE Int. Conf. Comput. Vis. Workshop, 2019, pp. 3512–3516.

[114] P. Rodríguez, "Total variation regularization algorithms for images corrupted with different noise models: a review," J. Electr. Comput. Eng., vol. 2013, 2013.

[115] M. F. Möller, "A scaled conjugate gradient algorithm for fast supervised learning," Neural Networks, vol. 6, no. 4, pp. 525–533, 1993.

[116] J. V. Hajnal, D. L. Hill, and D. J. Hawkes, "Medical image registration," Phys. Med. Biol., vol. 46, no. 3, 2001.

[117] B. Fischer and J. Modersitzki, "Ill-posed medicine–an introduction to image registration," Inverse Problems, vol. 24, no. 3, p. 034008, 2008.

[118] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," IEEE Trans. Med. Imag., vol. 32, no. 7, pp. 1153–1190, 2013.

[119] Y. Fu, Y. Lei, T. Wang, W. J. Curran, T. Liu, and X. Yang, "Deep learning in medical image registration: a review," Phys. Med. Biol., 2020.

[120] M. F. Beg, M. I. Miller, A. TrouvÉ, and L. Younes, "Computing large deformation metric mappings via geodesic flows of diffeomorphisms," Int. J. Comput. Vision, vol. 61, no. 2, pp. 139–157, 2005.

[121] P. Viola and W. M. W. III, "Alignment by maximization of mutual information," Int. J. Comput. Vision, vol. 24, no. 2, pp. 137–154, 1997.

[122] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee, "Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain," Med. Image Anal., vol. 12, no. 1, pp. 26–41, 2008.

[123] R. Bajcsy and S. Kovačič, "Multiresolution elastic matching," Comput. Vis., Graph., Image Process., vol. 46, no. 1, pp. 1–21, 1989.

[124] J. C. Gee and R. Bajcsy, "Elastic matching: Continuum mechanical and probabilistic analysis," Brain Warp., pp. 183–197, 1999.

[125] C. Davatzikos, "Spatial transformation and registration of brain images using elastically deformable models," Comput. Vis. Image Understand, vol. 66, no. 2, pp. 207–222, 1997.

[126] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free–form feformation: Application to breast mr images," IEEE Trans. Med. Imag., vol. 18, no. 8, pp. 712–721, 1999.

[127] J. Kybic and M. Unser, "Fast parametric elastic image registration," IEEE Trans. Image Process., vol. 12, no. 11, pp. 1427–1442, 2003.

[128] ——, "A fast nonrigid image registration with constraints on the jacobian using large scale constrained optimization," IEEE Trans. Med. Imag., vol. 27, no. 2, pp. 271–281, 2008.

[129] J. Thirion, "Image matching as a diffusion process: an analogy with maxwell's demons," Med. Image Anal., vol. 2, no. 3, pp. 243–260, 1998.

[130] M. F. Beg and A. Khan, "Symmetric data attachment terms for large deformation image registration," IEEE Trans. Med. Imag., vol. 26, no. 9, pp. 1179–1189, 2007.

[131] J. Ashburner and K. J. Friston, "Diffeomorphic registration using geodesic shooting and gauss-newton optimisation," NeuroImage, vol. 55, no. 3, pp. 954–967, 2011.

[132] D. Rueckert, A. F. Frangi, and J. A. Schnabel, "Automatic construction of 3d statistical deformation models using non-rigid registration," in Proc. Int. Conf. Med. Imag. Comp. Comput. Assist. Interv. (MICCAI), 2001, pp. 77–84.

[133] ——, "Automatic construction of 3-d statistical deformation models of the brain using nonrigid registration," IEEE Trans. Med. Imag., vol. 22, no. 8, pp. 1014–1025, 2003.

[134] J. Krebs et al., "Robust non-rigid registration through agent-based action learning," in Proc. Int. Conf. Med. Imag. Comp. Comput. Assist. Interv. (MICCAI), 2017, pp. 344–352.

[135] B. Gutiérrez-Becker, D. Mateus, L. Peter, and N. Navab, "Learning optimization updates for multimodal registration," in Proc. Int. Conf. Med. Imag. Comp. Comput. Assist. Interv. (MICCAI), 2016, pp. 19–27.

[136] ——, "Guiding multimodal registration with learned optimization updates," Med. Image Anal., vol. 41, pp. 2–17, 2017.

[137] M. Kim, G. Wu, Q. Wang, S. Lee, and D. Shen, "Improved image registration by sparse patch-based deformation estimation," NeuroImage, vol. 105, pp. 257–268, 2015.

[138] Q. Wang, M. Kim, Y. Shi, G. Wu, and D. Shen, "Predict brain mr image registration via sparse learning of appearance and transformation," Med. Image Anal., vol. 20, no. 1, pp. 61–75, 2015.

[139] J. Fan, X. Cao, P. Yap, and D. Shen, "Birnet: Brain image registration using dual-supervised fully convolutional networks," Med. Image Anal., vol. 54, pp. 193–206, 2019.

[140] H. Sokooti, B. de Vos, F. Berendsen, B. P. F. Lelieveldt, I. Išgum, and M. Staring, "Nonrigid image registration using multi-scale 3d convolutional neural networks," in Proc. Int. Conf. Med. Imag. Comp. Comput. Assist. Interv. (MICCAI), 2017, pp. 232–239.

[141] X. Yang, R. Kwitt, M. Styner, and M. Niethammer, "Quicksilver: Fast predictive image registration–a deep learning approach," NeuroImage, vol. 158, pp. 378–396, 2017.

[142] M. M. Rohé, M. Datar, T. Heimann, M. Sermesant, and X. Pennec, "Svf-net: Learning deformable image registration using shape matching," in Proc. Int. Conf. Med. Imag. Comp. Comput. Assist. Interv. (MICCAI), 2017, pp. 266–274.

[143] Y. Hu et al., "Weakly-supervised convolutional neural networks for multimodal image registration," Med. Image Anal., vol. 49, pp. 1–13, 2018.

[144] B. D. deVos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum, "End-to-end unsupervised deformable image registration with a convolutional neural network," in Proc. Deep Learning Med. Imag. Anal. Multimodal Learning for Clinical Decision Support, 2017, pp. 204–212.

[145] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "An unsupervised learning model for deformable medical image registration," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 9252–9260.

[146] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in Adv. Neural. Inf. Process. Syst., 2015, pp. 2017–2025.

[147] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in Proc. Int. Conf. Med. Imag. Comp. Comput. Assist. Interv. Springer, 2015, pp. 234–241.

[148] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "Voxelmorph: a learning framework for deformable medical image registration," IEEE Trans. Med. Imag., vol. 38, no. 8, pp. 1788–1800, 2019.

[149] D. W. Shattuck et al., "Construction of a 3d probabilistic atlas of human cortical structures," Neuroimage, vol. 39, no. 3, pp. 1064–1080, 2008.

[150] H. Li and Y. Fan, "Non-rigid image registration using self-supervised fully convolutional networks without training data," in Proc. IEEE Int. Symp. Biomed. Imaging, 2018, pp. 1–4.

[151] A. Klein et al., "Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration," Neuroimage, vol. 46, no. 3, pp. 786–802, 2009.

[152] A. D. Martino et al., "The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism," Molecular psychiatry, vol. 19, no. 6, pp. 659–667, 2014.

[153] S. G. Mueller et al., "Ways toward an early diagnosis in alzheimer's disease: the alzheimer's disease neuroimaging initiative (adni)," Alzheimer's & Dementia, vol. 1, no. 1, pp. 55–66, 2005.

[154] V. Fonov et al., "Unbiased average age-appropriate atlases for pediatric studies," Neuroimage, vol. 54, no. 1, pp. 313–327, 2011.

[155] L. R. Dice, "Measures of the amount of ecologic association between species," Ecology, vol. 26, no. 3, pp. 297–302, 1945.

[156] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, 2004.

[157] B. B. Avants, N. J. Tustison, G. Song, P. A. Cook, A. Klein, and J. C. Gee, "A reproducible evaluation of ants similarity metric performance in brain image registration," Neuroimage, vol. 54, no. 3, pp. 2033–2044, 2011.

[158] M. Fritsche, S. Gu, and R. Timofte, "Frequency separation for real-world super-resolution," in IEEE Int. Conf. Comput. Vis. Workshop. IEEE, 2019, pp. 3599–3608.

[159] D. Dai, Y. Wang, Y. Chen, and L. Van Gool, "Is image super-resolution helpful for other vision tasks?" in IEEE Winter Conf. Applications Comput. Vis., 2016, pp. 1–9.

[160] L. Wang, D. Li, Y. Zhu, L. Tian, and Y. Shan, "Dual super-resolution learning for semantic segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 3774–3783.

[161] P. H. Hennings-Yeomans, S. Baker, and B. V. Kumar, "Simultaneous super-resolution and feature extraction for recognition of low-resolution faces," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. IEEE, 2008, pp. 1–8.

[162] X. Jing et al., "Super-resolution person re-identification with semi-coupled low-rank discriminant dictionary learning," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 695–704.

[163] M. E. Angelopoulou, C. S. Bouganis, P. Y. Cheung, and G. A. Constantinides, "Robust real-time super-resolution on FPGA and an application to video enhancement," ACM Trans. Reconfigurable Technol. Syst., vol. 2, no. 4, pp. 1–29, 2009.

[164] K. Seyid, S. Blanc, and Y. Leblebici, "Hardware implementation of real-time multiple frame super-resolution," in Proc. IEEE Int. Conf. on VLSI and System-on-Chip, 2015, pp. 219–224.

[165] O. Bowen and C.-S. Bouganis, "Real-time image super resolution using an FPGA," in Proc. Int. Conf. on Field Programmable Logic and Applications, 2008, pp. 89–94.

[166] C.-H. Kim, S.-M. Seong, J.-A. Lee, and L.-S. Kim, "Winscale: An image-scaling algorithm using an area pixel model," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 6, pp. 549–553, 2003.

[167] C.-C. Lin, M.-H. Sheu, C. Liaw, and H.-K. Chiang, "Fast first-order polynomials convolution interpolation for real-time digital image reconstruction," IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 9, pp. 1260–1264, 2010.

[168] S.-L. Chen, H.-Y. Huang, and C.-H. Luo, "A low-cost high-quality adaptive scalar for real-time multimedia applications," IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 11, pp. 1600–1611, 2011.

[169] M. C. Yang, K. L. Liu, and S. Y. Chien, "A Real-Time FHD Learning-Based Super-Resolution System Without a Frame Buffer," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 64, no. 12, pp. 1407–1411, 2017.

[170] Y. Kim, J. S. Choi, and M. Kim, "2X Super-Resolution Hardware Using Edge-Orientation-Based Linear Mapping for Real-Time 4K UHD 60 fps Video Applications," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 65, no. 9, pp. 1274–1278, 2018.

[171] T. Manabe, Y. Shibata, and K. Oguri, "FPGA implementation of a real-time super-resolution system using a convolutional neural network," in Proc. Int. Conf. on Field-Programmable Technol., 2017, pp. 249–252.

[172] Z. He, H. Huang, M. Jiang, Y. Bai, and G. Luo, "FPGA-Based Real-Time Super-Resolution System for Ultra High Definition Videos," in Proc. 26th IEEE Int. Symp. on Field-Programmable Custom Computing Machines, 2018, pp. 181–188.

[173] J.-W. Chang, K.-W. Kang, and S.-J. Kang, "An Energy-Efficient FPGA-Based Deconvolutional Neural Networks Accelerator for Single Image Super-Resolution," IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 1, pp. 281–295, Jan. 2020.

[174] B. Shi, Z. Tang, G. Luo, and M. Jiang, "Winograd-Based Real-Time Super-Resolution System on FPGA," in Proc. Int. Conf. on Field-Programmable Technol., 2020, pp. 423–426.

[175] D. Fuoli, S. Gu, and R. Timofte, "Efficient Video Super-Resolution through Recurrent Latent Space Propagation," in Proc. IEEE Int. Conf. Comput. Vis. Workshops, 2019.

[176] W. Shi et al., "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 1874–1883.

[177] R. V. Hogg, J. W. McKean, and A. T. Craig, Introduction to Mathematical Statistics, 8th ed. Pearson Education Limited, 2020.

[178] P. Yi, Z. Wang, K. Jiang, J. Jiang, and J. Ma, "Progressive Fusion Video Super-Resolution Network via Exploiting Non-Local Spatio-Temporal Correlations," in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 3106–3115.

[179] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with Task-Oriented Flow," Int. J. Comput. Vis., vol. 127, no. 8, pp. 1106–1125, 2019.

[180] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-Revealing Deep Video Super-Resolution," in Proc. IEEE Int. Conf. Comput. Vis., no. 413113, 2017, pp. 4482–4490.

[181] M. Bevilacqua, A. Roumy, C. Guillemot, and M. line Alberi Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in Proc. Brit. Mach. Vis. Conf., 2012.

[182] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in Proc. Int. Conf. on Curves and Surfaces, 2012, pp. 711–730.

[183] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in Proc. IEEE Int. Conf. Comput. Vis., 2001, pp. 416–423.

[184] Y. Huang, W. Wang, and L. Wang, "Bidirectional recurrent convolutional networks for multi-frame super-resolution," Adv. Neural Inf. Process. Syst., vol. 28, pp. 235–243, 2015.

[185] K. Aizawa et al., "Building a manga dataset "manga109" with annotations for multimedia applications," IEEE MultiMedia, vol. 27, no. 2, pp. 8–18, Apr. 2020.

[186] J. Lee and I. C. Park, "High-Performance Low-Area Video Up-Scaling Architecture for 4-K UHD Video," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 64, no. 4, pp. 437–441, 2017.