



High-performance Video Super-resolution using FPGA-CPU Hybrid SoC

Author:

Reich CANLAS

Adviser:

Engr. Carlo OCHOTORENA

In partial fulfillment of the requirements

for the degree of Master of Science

in

Electronics and Communications Engineering

April 2015

Contents

Contents	i
List of Figures	iii
List of Tables	iv
Abbreviations	v
1 INTRODUCTION	1
1.1 Background of the Study	1
1.2 Statement of the Problem	2
1.3 Objectives of the Study	2
1.3.1 General Objective	2
1.3.2 Specific Objectives	2
1.4 Scope and Limitations of the Study	3
2 REVIEW OF RELATED LITERATURE	4
2.1 Image Quality Assessment (IQA)	4
2.2 Image Super-resolution	5
2.2.1 Image Observation Model	6
2.2.2 Frequency Domain	6
2.2.3 Methods Based on Sparse Representations	7
2.2.4 Dictionary Learning Methods	7
2.2.5 Computational Intelligence Methods	7
2.3 Challenges in Image SR	8
2.3.1 Image Registration	8
2.3.2 Computational Efficiency	8
2.3.3 Robustness	9
2.3.4 Edge preservation	9
2.4 Video super-resolution	9
2.4.1 Bayesian Methods	9
2.4.2 Computational Intelligence Methods	10
2.5 High Performance Computing Platforms	10

2.5.1	Graphics Processing Units	10
2.5.2	Manycore Coprocessors	11
2.5.3	Field Programmable Gate Arrays (FPGAs)	11
2.5.4	Design Considerations and Strategies	12
2.6	Comparison of CPU, GPU and FPGA	12
2.6.0.1	Use in super-resolution applications	12
2.7	Zynq-7000 System-on-a-chip	13
3	CONCEPTUAL FRAMEWORK	14
3.1	Algorithm Framework	14
3.2	Hardware Framework	14
4	METHODOLOGY	16
4.1	Benchmarking of state-of-the-art algorithms	16
4.2	Initial CPU Evaluation	17
4.3	FPGA Deployment and Comparative Tests	18
5	SUMMARY	19
A	Gantt Chart	20
B	Proposed Budget	21
	References	22

List of Figures

List of Tables

Abbreviations

GPU	G raphics P rocessing U nit
CPU	C entral P rocessing U nit
SR	S uper- R esolution
PSNR	P eak S ignal-to- N oise R atio
SSIM	S tructural S imilarity I ndex M easure
FSIM	F eature S imilarity I ndex M easure
FPGA	F ield P rogrammable G ate A rray
GPGPU	G eneral- P urpose computing on G raphics P rocessing U nits
SoC	S ystem- o n-a- C hip
IQA	I mage Q uality A ssessment
HVS	H uman V ision S ystem
HDL	H ardware D escription L anguage

Chapter 1

INTRODUCTION

1.1 Background of the Study

The search for ever higher screen resolutions led to the advent of ultra high definition television screens and monitors. However, to date, common video sources do not use the full capability of most high-resolution televisions of today.

Super-resolution is the process of rendering or recovering a larger image or video given some low-resolution source ([Dong, Loy, & He, 2014](#)). Super-resolution finds its applications in diverse fields of study. Examples include video surveillance, in [Caner, a.M. Tekalp, and Heinzelman \(2003\)](#) and [Zhang, Zhang, Shen, and Li \(2010\)](#), medical imaging in [Malczewski and Stasinski \(2008\)](#), and satellite imaging . Multi-frame image super-resolution methods use a set of LR images to construct an HR image by exploring the spatial correlations in that set ([Cheng, Hwang, Jeng, & Lin, 2013](#)). This kind of SR is applicable in laboratory settings. In other cases, single frame SR is more appropriate. These methods try to extract information from only one HR image, making the task much more difficult than multi-frame.

1.2 Statement of the Problem

In many applications such as super-HD (4K) TV, super resolution has to be performed in real time (Shen, Wu, & Deng, 2014). However, as noted by Ishizaka et al. (2013) "it is several times slower than real-time" to upscale videos using commodity hardware. Besides, power consumption is also an important issue. To integrate super-resolution processing into existing systems, there must not be a drastic increase in power footprint. A number of state-of-the-art methods of SR use GPUs and manycore CPUs, which offer a degree of performance at the expense of heavy power consumption and heat dissipation. These current solutions also have unstable frame rates (Wu, Xiang, & Lu, 2011).

We are then confronted by the problem of finding a video super-resolution system that uses a fast algorithm to generate high-quality hi-resolution videos from a low-resolution source while maintaining a relatively small power footprint.

A class of integrated circuits known as FPGAs, which are demonstrated to have high performance-per-watt ratios, is suitable for the constraints specified in this study.

1.3 Objectives of the Study

1.3.1 General Objective

This study aims to come up with a new video super-resolution algorithm and implement this for use on a hybrid FPGA (field programmable gate array).

1.3.2 Specific Objectives

Specifically, the study aims to tackle the following goals:

1. To improve on the state-of-the-art video super-resolution algorithm in terms of PSNR (peak signal-to-noise ratio) and running time.
2. To profile the video super-resolution algorithm to determine performance bottlenecks.
3. To exploit parallelizable steps in the algorithm to further enhance suitability on an FPGA.

1.4 Scope and Limitations of the Study

Since previous studies (cited in Review of Related Literature) have shown that clear upscaling is practical only at a factor of 4, it has been decided that this study should concern itself with 4x upscaling only.

Chapter 2

REVIEW OF RELATED LITERATURE

2.1 Image Quality Assessment (IQA)

Quantification of quality is a prerequisite in the development of any image or video processing algorithm. Currently, there are two primary measures of output image quality, namely, the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM). The choice of PSNR or SSIM is typically arbitrary, with a few informal arguments favoring one or the other ([Farsiu, Robinson, Elad, & Milanfar, 2004](#)). To aid in selection of a suitable metric, an analysis of both image metrics is found in [Horé and Ziou \(2010\)](#). They state that a mathematical relationship exists between the two metrics, thus making it possible to predict the PSNR from the SSIM and vice-versa. They only differ in their sensitivity to image degradations as introduced by noise, compression, and hardware limitations.

A recent addition to the list of metrics is the FSIM (Feature Similarity Index Measure) ([Zhang, Zhang, Mou, & Zhang, 2011](#)). The metric was proposed on the basis of human visual systems (HVS) understanding an image mainly due to its low-level features, such as edges and zero-crossings.

2.2 Image Super-resolution

Still-image super-resolution (SR) is the reconstruction of a high-resolution (HR) image given one, or a set of, low-resolution (LR) images. In the literature, the words "super-resolution" and "upsampling" are typically interchanged, but [Takeda, Milanfar, Protter, and Elad \(2009\)](#) clarified the distinction between the terms "upsampling" and "super-resolution". They stated that "if an algorithm which does not receive input frames that are aliased, it will still produce an output with a higher number of pixels and/or frames (i.e., 'upsampled'), but which is not necessarily 'super-resolved'". Super-resolution began as the problem of image restoration from a noisy signal ([Helstrom, 1967](#)). The first known work that directly tackled SR is that of [Tsai and Huang \(1984\)](#).

Traditionally, super-resolution of images is performed with several observed LR images, thus being termed "multi-frame SR". This is done in order to remove artifacts introduced by the low-resolution camera sensor ([Yang & Huang, 2010](#)). Applications such as medical and satellite imaging prefer the use of multi-frame SR since the minutest of features are crucial to analysis in those fields.

There is another approach which involves only a single observation or image. The limited set of data severely limits the quality obtainable, thus the SR problem becomes ill-posed ([Yang & Huang, 2010](#)). Image upscaling for information technology and entertainment is one application that relies on single-frame SR, as there is no available redundancy for the images used in those areas.

Super-resolution is necessary in the following fields of interest:

1. Surveillance video: In [Camargo, R.schultz, Wang, Fevig, and He \(2010\)](#), an SR mosaicking algorithm that stitches and super-resolves UAV (Unmanned Aerial Vehicle) video was implemented on a GPU-CPU pair. They were able to reach as high a PSNR as 41.10 dB for synthetic images. Their application is not real-time, however.
2. Satellite imaging:

3. Medical imaging: [Quan et al. \(2010\)](#) proposed a real-time algorithm that uses localization-based SR, the superior type of SR microscopy for live cell imaging. The algorithm can achieve between 30-500 fps (frames per second) depending on the speed of the camera used.
4. Video upscaling: This is the main premise of the present study, as with the rest of the papers cited in this chapter.

To the present day, super-resolution remains an active area of research, as evidenced by the wealth of literature cited in this study. The following sections present various approaches to SR that rely on several different models.

2.2.1 Image Observation Model

Several factors affect the output of a digital system, including finite aperture size and finite sensor size ([Yang & Huang, 2010](#)). The image observation model, as adopted by common imaging systems, is shown in Figure 2.1.

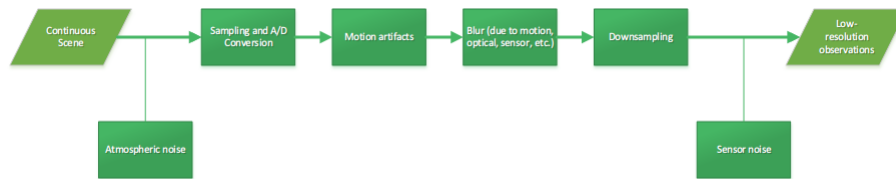


FIGURE 2.1: The Image Observation Model.

2.2.2 Frequency Domain

The first SR paper as authored by [Tsai and Huang \(1984\)](#) describes the SR process in the frequency domain. Their algorithm takes advantage of the shift and aliasing properties of the continuous and discrete Fourier transforms, given a set of multiple shifted low resolution images. A few extensions have been proposed, such as ([Yang & Huang, 2010](#))

2.2.3 Methods Based on Sparse Representations

The relative absence of data in the low resolution patches makes it reasonable to consider the LR patch space as a sparse representation of the HR patch space.

[Zeyde, Elad, and Protter \(2012\)](#) proposed an algorithm that uses the Sparseland model previously developed by [Elad and Aharon \(2006\)](#).

2.2.4 Dictionary Learning Methods

Digital signal data take up too much storage space, but most of this space does not account for the most significant components of the signal it represents. Compression and alternative representations are therefore required to reduce storage size while preserving fidelity.

Dictionary learning is the process of training a set of mutually orthogonal basis vectors in order to create a dictionary matrix. This matrix can then model any signal as a combination of its columns, better known as "atoms" ([Kreutz-Delgado et al., 2003](#)).

[Wright, Huang, Yang, and Ma \(2010\)](#) jointly trained a dictionary for low resolution and another for high resolution patches to enforce sparse representation similarity for both patch spaces. Their approach is also robust to noise, as it uses local sparse modeling. [Yang, Wang, Lin, Cohen, and Huang \(2012\)](#) similarly stressed the importance of learning two coupled dictionaries (observation dictionary and latent dictionary). However, the difference in their methods is that they used a coupled dictionary learning method for single-image SR.

2.2.5 Computational Intelligence Methods

So far, the previous methods mentioned all have solid mathematical foundations. However, it has been found out (citation here) that a great number of real-world problems cannot be modeled into well-posed mathematical problems, including super-resolution. A class of algorithms under

”computational intelligence” rely on mimicking natural systems to model and solve such kinds of problems.

[Dong et al. \(2014\)](#) used a deep convolutional neural network in order to learn a mapping between the LR and HR spaces. There are three stages involved: patch extraction and representation, non-linear mapping, and reconstruction.

2.3 Challenges in Image SR

Researchers still struggle with the following challenges, despite significant advancement in the SR literature.

2.3.1 Image Registration

Image registration is the process of mapping two images both spatially and with respect to intensity ([Brown, 1992](#)). Image registration is another ill-posed problem. Artifacts caused by registration problems are more noticeable and annoying than the blurring effect as a result of image interpolation ([Yang & Huang, 2010](#)).

2.3.2 Computational Efficiency

According to [Yang and Huang \(2010\)](#), the computational efficiency of SR is severely limited by the fact that there are a large number of unknowns and computationally-expensive matrix manipulations. To alleviate the problem, the author advocates ”massive parallel computing”. All of the so-called ”real-time SR algorithms” can only handle simple motion models, therefore, they cannot be used for real-world videos.

2.3.3 Robustness

SR algorithms are typically sensitive to signal outliers resulting from motion, blur, noise, etc. As the image degradation model parameters are difficult to estimate, researchers nowadays take into account the robustness of their approach ([Yang & Huang, 2010](#))

2.3.4 Edge preservation

It is typical in SR algorithms to lose details or edges in the output image/video. SR techniques for edge preservation have therefore been proposed. [Vishnukumar, Nair, and Wilscy \(2014\)](#) uses self-examples and high-frequency features to provide edge preservation in SR. Their PSNR goes as high as 30.77 dB, SSIM as high as 0.935, and their highest FSIM is 0.955.

2.4 Video super-resolution

Video super-resolution is the extension of image SR to moving pictures. An additional temporal dimension can now be factored in the SR process. It can generally be divided into two categories: incremental and simultaneous ([Su, Wu, & Zhou, 2011](#)). The former category is faster but less visually consistent to the human eye. [Liu and Sun \(2014\)](#) mentions that video SR is relatively more challenging than image SR which has been studied for decades, due to the presence of an additional temporal dimension.

2.4.1 Bayesian Methods

[Liu and Sun \(2014\)](#) propose a Bayesian video SR system that can simultaneously estimate the HR frame, motion flow fields, blur kernel, and noise level. Their method works best when the motion is slow and smooth, and would fail if there are significant lighting changes and occlusion. They acknowledged that aliasing both benefits and "derails" super-resolution.

2.4.2 Computational Intelligence Methods

As in image SR, video SR is a highly nonlinear task and is amenable to processing via computational intelligence. For example, [Cheng et al. \(2013\)](#) constructed an artificial neural network (ANN) for video SR. It is a four-stage algorithm, consisting of classifiers to categorize the image, motion-trace volume collection for pixel tracking, temporal adjustment for fast motions and complicated scenes, and ANN learning.

2.5 High Performance Computing Platforms

[Yang and Huang \(2010\)](#) suggest that high-performance hardware matters in tackling super-resolution problems. Typically, image SR algorithms are first developed for computer CPUs. Modern CPUs (central processing units) of computers combine high-frequency processors with a degree of parallelism to add more processing power to algorithms. Even so, the CPU is not enough to handle tasks such as SR in real-time. There are several steps in the SR process that may be implemented as parallel tasks. Following are the discussions on GPUs, manycore coprocessors, and FPGAs, three parallel platforms commonly in use today.

2.5.1 Graphics Processing Units

GPUs (Graphics Processing Units) have been favored in recent years for this task, as it offers high amounts of parallelism (due to its multiple cores) and compatibility with existing computer systems and programming paradigms. [Wu et al. \(2011\)](#) claims 6x speedup against the same algorithm implemented on a CPU.

2.5.2 Manycore Coprocessors

This class of parallel processors are based off CPU architectures but have more cores than the traditional CPU and are meant to run at a lower frequency. A host CPU passes the appropriate parallel instructions to the manycore coprocessor and subsequently fetches the results of the computation. Manycore processors offer more programmability than GPUs simply by the fact that they share the same architecture as the host CPU. The only known product in this category is that of Intel MIC (Many Integrated Core) architecture ([Intel, 2014](#)).

[Ishizaka et al. \(2013\)](#) demonstrated a power-efficient real-time SR system that uses a virtual pipeline to improve the performance as well as the utilization of both the manycore and the host processors. Their set-up was able to achieve 31.5 fps, satisfying the real-time requirement. The downside with their setup is the limited adoption of the MIC platform and the power requirement of about 300 W ([Intel, 2014](#)).

2.5.3 Field Programmable Gate Arrays (FPGAs)

FPGAs (Field Programmable Gate Arrays) are logic devices that can be reconfigured by a designer on the field after being manufactured. Since at the lowest level, logic circuits are inherently parallel and real-time, FPGAs offer optimization potential that cannot be realized when using instruction-based platforms such as CPUs and GPUs. FPGAs typically run at much lower frequencies than CPUs and GPUs, making them more power efficient. Higher-end FPGAs even offer the ability to be partially dynamically reconfigured, so that even while it is running, parts of the FPGA fabric gets their design altered ([Dye, 2012](#)). These factors makes FPGAs suitable for the most computationally-intensive real-time applications while conserving energy.

[Sirowy and Forin \(2008\)](#) investigated the reasons why an FPGA offers high speedups over sequential processing devices such as CPU, manycore, and GPU. Among these are: the removal of an instruction fetch step, hiding the control instructions, executing multiple instructions in parallel, and pipelining instructions.

2.5.4 Design Considerations and Strategies

Since the SR system of this study is to be integrated into other computing systems, it is imperative to develop an embedded system, one which consumes less energy. The more power used, the more heat is generated. According to [Anderson, Dykes, and Riedel \(2003\)](#), failure rates double for every 15 degree Celsius rise in temperature. In this light, for an embedded system, the GPU and CPU are not applicable processors. [Mittal \(2014\)](#) considered using an "unconventional core" such as an FPGA to realize lower power-consumption in an embedded system. [Struyf, Beugher, Uytsel, Kanters, and Goedem \(2014\)](#)

2.6 Comparison of CPU, GPU and FPGA

[Asano, Maruyama, and Yamaguchi \(2009\)](#) compared the performance of the CPU, GPU and FPGA in image processing applications. They noted that CPUs are consistently lagging behind the GPU and FPGA, while the GPU is best for "naive computation methods" in which processing takes place on a per pixel basis.

[Fowers, Brown, Cooke, and Stitt \(2012\)](#) compared the performance and the energy expended by FPGAs, GPUs and multicore CPUs. This paper is significant to the present study since their focus is on sliding-window algorithms, which take the data on a per-block basis instead of per-pixel. This makes computation more efficient.

2.6.0.1 Use in super-resolution applications

The following papers prove the feasibility of an FPGA in SR applications. [Angelopoulou, Bouganis, Cheung, and Constantinides \(2009\)](#) created a real-time video SR system on an FPGA that is robust against noise. It uses the iterative back projection algorithm. However, the system depends on an adaptive image sensor [Szydzik, Callico, and Nunez \(2011\)](#) constructed a high quality SR system on an FPGA. They were able to achieve 2x upscaling at 25 fps while using

only less than 37% of FPGA resources of the state-of-the-art algorithm at that time. [Bowen and Bouganis \(2008\)](#) achieved 3x speedup over equivalent software (CPU) implementations.

2.7 Zynq-7000 System-on-a-chip

The Zynq-7000 programmable System-on-a-Chip is the first-of-its-kind FPGA-CPU hybrid in the market. It combines the high-performance, low-power ARM Cortex A9 CPU, high-end computer components such as DDR3 RAM, and a programmable logic fabric similar to that of an FPGA. One advantage of using this SoC is that several peripherals are already built-in and hardwired, eliminating the need for soft IP cores for common devices. Another is that of faster sequential processing. According to Amdahl's Law, the speedup of a parallel program is limited by the time needed for the sequential fraction of the program ([Amdahl, 1967](#)). Having both a highly parallel programmable fabric and a fast sequential processor is a great help if an algorithm has to run real-time.

Chapter 3

CONCEPTUAL FRAMEWORK

3.1 Algorithm Framework

3.2 Hardware Framework

Figure 3.1 illustrates the flow of data across the hardware devices to be used in the study. A video source such as a camera or prerecorded file will be sent for super-resolution on the SoC, which will then display the result on the monitor in real-time. Initially a conventional computer

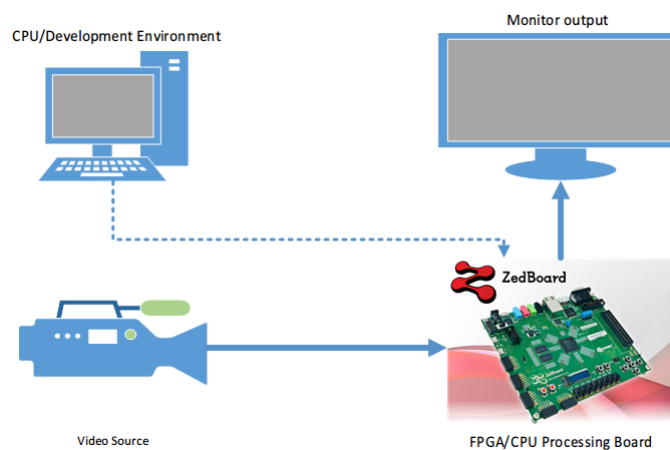


FIGURE 3.1: Framework for Hardware Implementation.

will serve as the development and evaluation environment for the SR algorithm. A revised version of the algorithm can then be sent to the SoC for further testing and fine-tuning.

Chapter 4

METHODOLOGY

4.1 Benchmarking of state-of-the-art algorithms

The metrics to be tested are the execution time, the PSNR, SSIM, and FSIM.

Given a reference image a and a test image b , both of the same size $M \times N$, the PSNR is computed using the following equation ([Horé & Ziou, 2010](#)):

$$PSNR(a, b) = 20 \log_{10} \frac{255}{\sqrt{MSE(a, b)}} \quad (4.1)$$

where

$$MSE(a, b) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (a_{ij} - b_{ij})^2 \quad (4.2)$$

The SSIM is computed using the following equation ([Horé & Ziou, 2010](#)):

$$SSIM(a, b) = l(a, b)c(a, b)s(a, b) \quad (4.3)$$

where

$$l(a, b) = \frac{2\mu_a\mu_b + C_1}{\mu_a^2 + \mu_b^2 + C_1} \quad (4.4)$$

$$c(a, b) = \frac{2\sigma_a\sigma_b + C_2}{\sigma_a^2 + \sigma_b^2 + C_2} \quad (4.5)$$

$$s(a, b) = \frac{\sigma_{ab} + C_3}{\sigma_a\sigma_b + C_3} \quad (4.6)$$

Equation 4.4 measures the similarity in luminance and is equal to 1 only if $\mu_a = \mu_b$. Similarly, equation 4.5 compares the standard deviation of the two images (which corresponds to the contrast). It will only equal to 1 if $\sigma_a = \sigma_b$. The structure comparison equation (4.6) measures the correlation between the images using the covariance σ_{ab} between them.

The FSIM is computed using the following equation (Zhang et al., 2011):

$$\frac{\sum_{x \in \Omega} S_L(\mathbf{x}) \cdot PC_m(\mathbf{x})}{\sum_{x \in \Omega} PC_m(\mathbf{x})} \quad (4.7)$$

where Ω is the whole spatial image domain. $S_L(\mathbf{x})$

4.2 Initial CPU Evaluation

Successive modifications for speed and output quality will be performed until the improvements meet the target +0.1dB increase in quality and 4x speedup. Initially, these measurements will be performed in the CPU.

The initial algorithm runs will take place on a computer with an Intel Core i7 3632QM Processor 2.2 GHz, 16GB RAM, and NVIDIA GeForce GT 730M 2GB. Profiling the algorithm for bottlenecks will be done with the assistance of MATLAB 2015a Profiler. This software package can determine the slowest segments of the algorithm and suggest necessary modifications. MATLAB will also be used to code the algorithms and will assist in the preliminary conversion into a form suitable for the Zynq through its Vision HDL Toolbox.

4.3 FPGA Deployment and Comparative Tests

Once the criteria are met in the CPU version of the algorithm, the next step is to parallelize using the available FPGA resources. The platform to be used is the Xilinx Zynq-7000 SoC (embedded CPU+FPGA) on a Digilent Zedboard. The hybrid FPGA and low-power CPU chip enables the dual advantage of quick sequential processing and fine-grained parallel computing. The Xilinx Vivado Design Suite 2014.2 software will facilitate the process of converting the CPU-based algorithm to parallel. In addition, Vivado will be used to fine-tune the algorithm and optimize resource usage in the hardware.

Chapter 5

SUMMARY

In summary, the system to be developed is a reading

Appendix A

Gantt Chart

Appendix B

Proposed Budget

References

- Amdahl, G. M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. In *Afips spring joint computer conference* (pp. 187–196).
- Anderson, D., Dykes, J., & Riedel, E. (2003). More than an interface - SCSI vs. ATA. In *Proceedings of fast '03: 2nd unix conference on file and storage technologies* (pp. 245–257).
- Angelopoulou, M. E., Bouganis, C.-S., Cheung, P. Y. K., & Constantinides, G. a. (2009). Robust Real-Time Super-Resolution on FPGA and an Application to Video Enhancement. *ACM Transactions on Reconfigurable Technology and Systems*, 2(4), 1–29. doi: 10.1145/1575779.1575782
- Asano, S., Maruyama, T., & Yamaguchi, Y. (2009). Performance comparison of FPGA, GPU and CPU in image processing. *FPL 09: 19th International Conference on Field Programmable Logic and Applications*, 126–131. doi: 10.1109/FPL.2009.5272532
- Bowen, O., & Bouganis, C. (2008). Real-time image super resolution using an FPGA. *Field Programmable Logic and ...*, 89–94. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4629913
- Brown, L. G. (1992). A survey of image registration techniques. *ACM Computing Surveys*, 24(4), 325–376. doi: 10.1145/146370.146374
- Camargo, A., R.schultz, R., Wang, Y., Fevig, R. a., & He, Q. (2010). GPU-CPU implementation for super-resolution mosaicking of Unmanned Aircraft System (UAS) surveillance video. *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, 25–28. doi: 10.1109/SSIAI.2010.5483926
- Caner, G., a.M. Tekalp, & Heinzelman, W. (2003). Super resolution recovery for multi-camera

- surveillance imaging. *2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698)*, 1. doi: 10.1109/ICME.2003.1220866
- Cheng, M. H., Hwang, K. S., Jeng, J. H., & Lin, N. W. (2013). Classification-based video super-resolution using artificial neural networks. *Signal Processing*, 93(9), 2612–2625. Retrieved from <http://dx.doi.org/10.1016/j.sigpro.2013.02.013> doi: 10.1016/j.sigpro.2013.02.013
- Dong, C., Loy, C. C., & He, K. (2014). Image Super-Resolution Using Deep Convolutional Networks. *arXiv preprint*, 1–14.
- Dye, D. (2012). *Partial Reconfiguration of Xilinx FPGAs Using ISE Design Suite* (Vol. 374).
- Elad, M., & Aharon, M. (2006). Image denoising via learned dictionaries and sparse representation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1, 895–900. doi: 10.1109/CVPR.2006.142
- Farsiu, S., Robinson, D., Elad, M., & Milanfar, P. (2004). Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology*, 14(2), 47–57. doi: 10.1002/ima.20007
- Fowers, J., Brown, G., Cooke, P., & Stitt, G. (2012). A performance and energy comparison of FPGAs, GPUs, and multicores for sliding-window applications. *Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays - FPGA '12*, 47. Retrieved from <http://dl.acm.org/citation.cfm?doid=2145694.2145704> doi: 10.1145/2145694.2145704
- Helstrom, C. W. (1967). Image Restoration by the Method of Least Squares. *Journal of the Optical Society of America*, 57(3), 297. doi: 10.1364/JOSA.57.000297
- Horé, A., & Ziou, D. (2010). Image quality metrics: PSNR vs. SSIM. *Proceedings - International Conference on Pattern Recognition*, 2366–2369. doi: 10.1109/ICPR.2010.579
- Intel. (2014). *Intel Xeon Phi™ Coprocessor - Datasheet* (Tech. Rep. No. April). Retrieved from <http://www.colfax-intl.com/nd/downloads/Xeon-Phi-Coprocessor-Datasheet.pdf>
- Ishizaka, K., Miyamoto, T., Akimoto, S., Iketani, a., Hosomi, T., & Sakai, J. (2013). Power efficient realtime super resolution by virtual pipeline technique on a server with manycore

- coprocessors. *IEEE Symposium on Low-Power and High-Speed Chips - Proceedings for 2013 COOL Chips XVI*, 2, 2–4. doi: 10.1109/CoolChips.2013.6547918
- Kreutz-Delgado, K., Murray, J. F., Rao, B. D., Engan, K., Lee, T.-W., & Sejnowski, T. J. (2003). Dictionary learning algorithms for sparse representation. *Neural computation*, 15(2), 349–396. doi: 10.1162/089976603762552951
- Liu, C., & Sun, D. (2014). On bayesian adaptive video super resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2), 346–360. doi: 10.1109/TPAMI.2013.127
- Malczewski, K., & Stasinski, R. (2008). Toeplitz-based iterative image fusion scheme for MRI. *Proceedings - International Conference on Image Processing, ICIP*, 341–344. doi: 10.1109/ICIP.2008.4711761
- Mittal, S. (2014). A Survey of Techniques For Improving Energy Efficiency in Embedded Computing Systems. *International Journal of Computer Aided Engineering and Technology*. Retrieved from <http://arxiv.org/abs/1401.0765> doi: 10.1504/IJCAET.2014.065419
- Quan, T., Li, P., Long, F., Zeng, S., Luo, Q., Hedde, P. N., ... Huang, Z.-L. (2010). Ultra-fast, high-precision image analysis for localization-based super resolution microscopy. *Optics express*, 18(11), 11867–11876. doi: 10.1364/OE.18.011867
- Shen, Y., Wu, X., & Deng, X. (2014, December). GPU-aided real-time image/video super resolution based on error feedback. In *2014 IEEE Visual Communications and Image Processing Conference* (pp. 286–290). IEEE. Retrieved from <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7051560> doi: 10.1109/VCIP.2014.7051560
- Sirowy, S., & Forin, A. (2008). Where's the Beef ? Why FPGAs Are So Fast. *Microsoft Research*(September).
- Struyf, L., Beugher, S. D., Uytsel, D. H. V., Kanters, F., & Goedem, T. (2014). The battle of the giants : a case study of GPU vs FPGA optimisation for real-time image processing. In *International conference on pervasive and embedded computing and communication systems – peccs 2014* (pp. 112–119). Lisbon, Portugal: VISIGRAPP. Retrieved from

- <http://www.peccs.org/?y=2014>
- Su, H., Wu, Y., & Zhou, J. (2011). Adaptive incremental video super-resolution with temporal consistency. *Proceedings - International Conference on Image Processing, ICIP*, 1149–1152. doi: 10.1109/ICIP.2011.6115632
- Szydzik, T., Callico, G. M., & Nunez, A. (2011). Efficient FPGA Implementation of a High-Quality Super-Resolution Algorithm with Real-Time Performance. , 57(2), 664–672.
- Takeda, H., Milanfar, P., Protter, M., & Elad, M. (2009). Super-resolution without explicit subpixel motion estimation. *IEEE Transactions on Image Processing*, 18(9), 1958–1975. doi: 10.1109/TIP.2009.2023703
- Tsai, R., & Huang, T. S. (1984). Multiframe image restoration and registration. *Advances in computer vision and Image Processing*, 1(2), 317–339.
- Vishnukumar, S., Nair, M. S., & Wilscy, M. (2014). Edge preserving single image super-resolution with improved visual quality. *Signal Processing*, 105, 283–297. doi: 10.1016/j.sigpro.2014.05.033
- Wright, J., Huang, T., Yang, J., & Ma, Y. (2010). Image Super-Resolution via Sparse Representation. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 1–13. Retrieved from [http://www.ncbi.nlm.nih.gov/pubmed/20483687\\$\\delimiter\"026E30F\\$nhhttp://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4587647](http://www.ncbi.nlm.nih.gov/pubmed/20483687$\\delimiter\) doi: 10.1109/TIP.2010.2050625
- Wu, X., Xiang, H., & Lu, P. (2011). A GPU Accelerated Algorithm for Compressive Sensing Based Image Super-Resolution. *2011 Workshop on Digital Media and Digital Content Management*(60633030), 198–202. doi: 10.1109/DMDCM.2011.10
- Yang, J., & Huang, T. (2010). Image super-resolution: Historical overview and future challenges. In *Super-resolution imaging* (pp. 3–25). Retrieved from http://books.google.com/books?hl=en&lr=&id=fjTUbMnvOkGC&oi=fnd&pg=PA1&dq=Image+super-resolution:+Historical+overview+and+future+challenges&ots=53GZConOGy&sig=oHgGJVo_57Tv19cxLV_XMtC1OPw
- Yang, J., Wang, Z., Lin, Z., Cohen, S., & Huang, T. (2012). Coupled dictionary training for

- image super-resolution. *IEEE Transactions on Image Processing*, 21(8), 3467–3478. doi: 10.1109/TIP.2012.2192127
- Zeyde, R., Elad, M., & Protter, M. (2012). On single image scale-up using sparse-representations. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6920 LNCS(1), 711–730. doi: 10.1007/978-3-642-27413-8_47
- Zhang, L., Zhang, H., Shen, H., & Li, P. (2010). A super-resolution reconstruction algorithm for surveillance images. *Signal Processing*, 90(3), 848–859. Retrieved from <http://dx.doi.org/10.1016/j.sigpro.2009.09.002> doi: 10.1016/j.sigpro.2009.09.002
- Zhang, L., Zhang, L., Mou, X., & Zhang, D. (2011). FSIM : A Feature Similarity Index for Image Quality Assessment. *IEEE Transactions on Image Processing*, 20(8), 2378–2386.