Final Year Project Inception Report
“Real-Time Face Super-Resolution”

Weng Nam Lio (wnl111) CID:683021

1 Project Specification

Real-time face super-resolution allows us to enhance information on low-resolution images of human faces for security purposes. The aim of this project is to investigate state-of-the-art single-frame[1] face super-resolution algorithms in terms of parameters optimisation, computational complexity and quality of results. In particular, a new promising algorithm called the ***Locality-Constrained Representation*** proposed by Jiang et al. [5] will be examined exhaustively along with other traditional face super-resolution (SR) techniques. The best algorithm will be chosen and implemented on an embedded system, consisting an ARM System-on-Chip (SoC) system with Field Programmable Gate Arrays (FPGA) fabric to accelerate the critical function of the algorithm.

Surveillance cameras play a key part in crime investigation as well as deterring crime and terrorism. However, the resolution of the low-cost surveillance cameras is often insufficient for law enforcement units to identify suspects or track suspicious individuals effectively. The quality of face images are further worsen by environmental factors such as underexposure, defocusing, distance of individuals from the camera, etc. As a result, identifying these blurred face images of interest is extremely difficult for humans. Face super-resolution methods provides an opportunity to improve face recognition performance and aid the law enforcement community in the conviction of offenders.

In order to obtain a enough details on a face image for recognition, the high frequency components in the Low-Resolution (LR) image can be inferred to form a High-Resolution (HR) image using face super-resolution, also known as face hallucination. However, most of the current face hallucination research (discussed in Section 2) focus on obtaining the best quality face reconstruction and disregard the computational complexity of their systems. There are little research where the face SR scheme is implemented on a real-time system, and almost none exists on an embedded platform. These time-consuming systems are difficult to realise in practical face recognition applications. Hence, this project also aims to establish an efficient mapping of the chosen algorithm on the ARM SoC system.

[1] A single low-resolution face image will be used as the input to the SR system (rather than a sequence of the same face in motion)

2 Background

2.1 Existing Face Super-Resolution Techniques

Most of the existing image hallucination techniques can be classified into two main categories: reconstruction-based and learning-based. However, reconstruction-based methods are more susceptible to ill-conditioned registration and inappropriate blurring operators[1], which leads to increasing reconstruction error. Meanwhile, Learning-based methods can give better performance and higher magnification factor with the help of a set of high- and low- resolution training face image pairs. Thus, we will focus on learning-based algorithms in this project.

2.1.1 Direct Interpolation

One simplest method to enhance low resolution face images is a direct interpolation with algorithms such as nearest-neighbours, bi-cubic interpolation or cubic-spline interpolation. Due to their simplicity, these algorithms are efficient and easy to implement. However, it is obvious that they are insufficient and produce poor quality results since they do not exploit prior knowledge of facial images. Nevertheless, direct interpolation methods can be useful as a baseline for performance comparison in this project.

2.1.2 Eigen-transformation via Principle Component Analysis

Wang et al. [8] proposed a method of hallucinating faces using Eigen-transformation. This technique reconstructs the hallucinated face from a linear combination of high resolution training images and the combination coefficients come from the low-resolution training images using the principal component analysis (PCA) method, as illustrated in Figure 1. Since PCA represents face images globally using a weighted combination of eigenfaces, the structural similarity of face image is preserved. The method also maintains high computational speed since reconstruction is performed in the global domain [6].

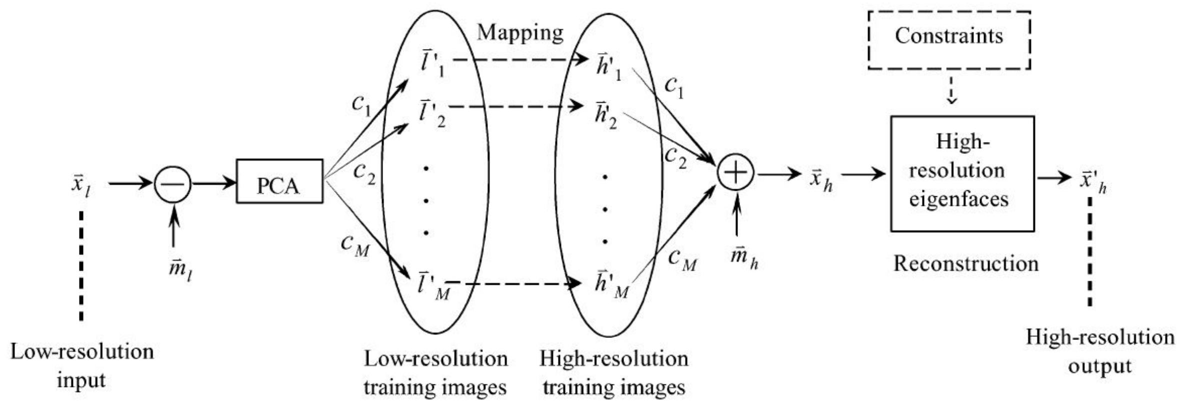

Figure 1: System diagram using eigentransformation for face hallucination [8]

2.2 Position-patch based approaches

Recently, a number of position-patch based approaches have been adopted in face hallucination research and many of them gave good empirical results [7, 6, 2, 5]. Since the human face is highly structured, position information is significant when performing reconstruction. Furthermore, position-patch based approaches are sought-after as global methods are unable to capture the fine individual characteristics of facial components.

For the following position-patch based approaches, let Y_H^m and Y_L^m denote the HR and LR training face images respectively, where $m = 1, ..., M$ where M is the size of each set. The LR input face image to be super-resoluted will be represented as X_L. Each face image in the training set and the input test image is divided into N small overlapping patch sets, with the results stored as $Y_H^m(i, j)$, $Y_L^m(i, j)$ and $X_L(i, j)$, where $i \leq U, j \leq V$ and $N = UV$.

2.2.1 Least Square Representation

Ma et al. [6] suggested the Least Square Representation model where the high-resolution image patches are synthesised from patches of the training samples at the same position (i, j). Each patch in the LR input image $X(i, j)$ can therefore be represented as a linear combination of weighted training image patches with a reconstruction error e.

$$X(i,j) = \sum_{m=1}^{M} Y^m(i,j) w_m(i,j) + e \tag{1}$$

The optimal reconstruction weight vector can then be obtained by computing the constrained least square problem.

$$w^*(i,j) = \underset{w}{\text{argmin}}(i,j) \left\| X(i,j) - \sum_{m=1}^{M} Y^m(i,j) w_m(i,j) \right\|_2^2 \tag{2}$$

However, the solution to equation 2 may not be unique when the number of training samples is much greater than the dimension of the patch. Consequently, the least square estimation can lead to biased solutions in practice.

2.2.2 Sparse Representation

To combat this instability, Jung et al. [2] transformed the computation of the optimal weights to a convex optimisation problem. By imposing a sparsity-constrained optimisation as represented in equation 3 and replacing the least squares estimation (equation 2), an unbiased solution can be obtained for the position-patch based face hallucination method based on LSR.

$$\min \|w(i,j)\|_1 \; s.t. \left\| X(i,j) - \sum_{m=1}^{M} Y^m(i,j) w_m(i,j) \right\|_2^2 \leq \epsilon \tag{3}$$

This l_1-norm minimisation gives the sparsest solution, leading to an exact solution of the optimal weight vector to the minimisation problem [5].

2.2.3 Locality-Constrained Representation

The Locality-Constrained Representation method proposed by Jiang et al. [5] takes a position-patch based approach and integrates a locality constraint into the least square inversion problem while maintaining sparsity of the weight vector.

In this scheme, the Euclidean distance between the LR test image patches $X_L(i,j)$ and each patch in the corresponding position in the LR training set $Y_L^m(i,j)$ is first calculated.

$$d_m(i,j) = \|X_L(i,j) - Y_L^m(i,j)\|_2, (1 \leq m \leq M) \tag{4}$$

The objective is to

$$\min \|d(i,j) \circ w(i,j)\|_2^2 \; s.t. \left\| X(i,j) - \sum_{m=1}^{M} Y^m(i,j) w_m(i,j) \right\|_2^2 \leq \epsilon, \sum_{m=1}^{M} w_m(i,j) = 1 \tag{5}$$

By taking the Lagrangian of equation(5), we can find the weight vector through

$$w^*(i,j) = \underset{w}{\operatorname{argmin}}(i,j) \left\{ \left\| X(i,j) - \sum_{m=1}^{M} Y^m(i,j) w_m(i,j) \right\|_2^2 + \tau \sum_{m=1}^{M} [d_m(i,j) w_m(i,j)]^2 \right\} \tag{6}$$

The first half of equation 6 measures the reconstruction error while the second half preserves locality. The regularisation parameter τ balances the contribution of the first and second term. Note that LcR becomes LSR when $\tau = 0$, i.e. locality is not taken into account.

2.3 Face Image Quality Measure

In this project, we would like to choose the 'best' face SR algorithm based on both output quality and computational complexity. Traditionally, image quality is assessed by the difference between pixels of a distorted image and the reference image. The simplest and most widely used quality metric is the mean square error (MSE), a full-reference quality metric computed from the average of the squared intensity difference. Unfortunately, this simple metric does not match perceived visual quality as illustrated in Figure 2, where images with very different distortion result in the same value of MSE.

To the human eyes, these distorted images have drastically different perceptual quality. Based on the philosophy of structural similarity, the structural similarity index (SSIM) assess the quality of an image based on the degradation of structural information [9]. Since our goal is to increase the recognisability of LR face images, SSIM will be used as the main image qualty metric in this project for optimising algorithms and parameter settings.

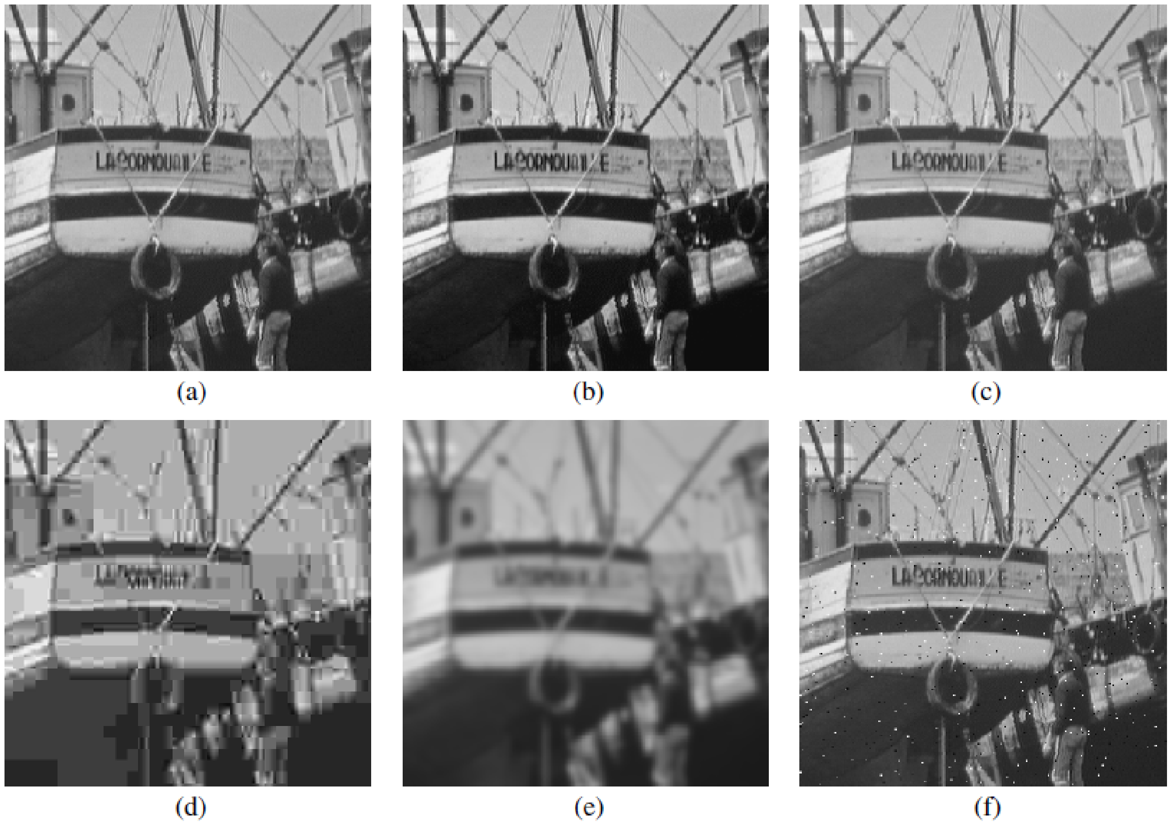

Figure 2: Comparison of "Boat" images with different types of distortions, all with MSE = 210. (a) Original image. (b) Constrast-stretched image. (c) Mean-shifted image. (d) JPEG compressed image. (e) Blurred image. (f) Salt-pepper impulsive noise contaminated image.[9]

2.4 Accelerating on an FPGA platform

In order to perform face SR in real-time, it is highly likely that we would need to accelerate the critical part of our chosen algorithm with hardware, namely an FPGA. However, developing applications on an FPGA using hardware description language is often a slow process. Higher level languages such as OpenCL can be a good option due to time constraint.

Using Altera's SDK for OpenCL, applications can be developed in a C-level environment enabling rapid FPGA designs[10]. Altera's OpenCL compiler (ACL compiler) implement kernel functions as dedicated pipelined hardware, which are then replicated to increase parallelism. SRAM Object File (SOF) containing custom hardware design is generated by the ACL compiler and can be downloaded into an FPGA when the host program is executed.

An OpenCL program consist of two parts; the host program and the kernel function. The programmer can choose to compile the kernel online or offline, i.e. either reading the kernel source file, or the pre-built kernel binaries in the host program. The ACL compiler also provides a compilation report, which contains information on the hardware configuration. This will allow us to predict the performance of their kernel without running the whole compilation, which can take a very long time.

3 Implementation Plan

Week Commencing	Duration	Project Tasks To-do/ Other commitments
20/10/2014	2 weeks	Matlab implementation of LcR Algorithm, Background Reading
03/11/2014	3 weeks	Implementation of assessment of algorithms (quality + speed), Matlab implementation of Bicubic Interpolation and PCA method
24/11/2014	2 weeks	Algorithms evaluation, Parameters tuning (patch size, overlap width, τ, num training images used, etc), Implement preLcR algorithm
08/11/2014	2 weeks	*Department of Computing exam revision*
	Christmas Break	
05/01/2015	2 weeks	Implement iterative LcR
19/01/2015	2 weeks	Optimise parameters with validation set, and evaluate the algorithms with test set, Interim report (due 02/02)
02/02/2015	4 weeks	C implementation of LcR algorithm with chosen parameters, verify correctness and performance, perform profiling and identify critical function to be mapped on to FPGA
02/03/2015	2 weeks	Set up FPGA, OpenCL implementation of critical function on FPGA
16/03/2015	2 weeks	*Department of Computing exam revision*
	Easter Break (continue with OpenCL implementation)	
	Exam on 30/04 and another one tbc	
27/04/2015	2 weeks	OpenCL implementation validation and testing, optimisation
11/05/2015	2 weeks	Real-time face SR SoC system performance evaluation
25/05/2015	2 weeks	Abstract (due 08/06), project wrap-up
08/06/2015	2 weeks	Final Report (due 17/06), Presentation (22-24/06)

Table 1: Project Schedule

There are three main parts to this project; algorithms and parameter settings exploration carried out in Matlab, mapping the chosen algorithm on an ARM SoC in C, and accelerating the critical function on an FPGA. Table 1 shows the project schedule of completed implementation since the start of Autumn Term, and the future time-line of work to be done.

Since this is a fairly implementation heavy project, we started the algorithms exploration in Matlab swifty after the project brief along with background research. The initial LcR algorithm was completed in two weeks. Over the Autumn term, Matlab implementation of other methods including bicubic interpolation and SR with eigen-transformation using PCA, initial assessment of the algorithms and parameters tuning has been performed. An improved LcR algorithm with sorting has also been implemented over the Christmas break.

Originally, we planned to complete the exploration of algorithms in Matlab in Autumn Term. Unfortunately, this stage overran because we did not take into account the Department of Computing exams in December during my initial meeting with Dr Bouganis. This has led to the decision of implementing the critical function in a higher level language instead of VHDL to reduce design time in the later stage of the project. Besides, I am positive about completing the project on-time since I only have two exams in the Summer Term.

4 Evaluation Plan

The final deliverable will consist of a C implementation of the chosen face SR algorithm running on an ARM SoC with an ARM9 processor. The critical function will be determined by profiling the C function and ported to run on an FPGA written in OpenCL.

The deliverable will be evaluated in terms of quality of the hallucinated face images, and the speed of reconstruction. A public Face Database consisting 922 faces (and their mirrored version) will be used in this project. The faces will be downsampled to HR and LR versions and split into training, validation and testing set randomly. All validation and test images must be absent in the training set. Implementation error should be easily spotted in this project since we can visualise the reconstructed face and compare that to the reference image. We can verify the correctness of the C implementation by comparing with the results given by our Matlab implementation.

As detailed in section 2.3, we chose SSIM as the reconstruction quality metric over other traditional similarity measures such as Mean Square Error (MSE) since the traditional methods are inconsistent with perceived visual quality [9]. The peak signal-to-noise ratio (pSNR) will also serve as a secondary reference to evaluate the perceived error. SSIM will also be used to optimise algorithms and parameter settings of the LcR system. An extension of this project will be to evaluate the algorithms in terms of their robustness against noise, mis-alignment and illumination.

Since we want to implement the algorithm on a real-time system, the time complexity of the system is as important as the output quality. To choose the best algorithm as well as the optimal parameter settings, the algorithms are timed with the Matlab commands `tic` and `toc`, which determines the elapsed time between the two commands. In later stages, time taken for the algorithm to run on the SoC can be obtained from profiling tools. It is also important to evaluate the cost-benefit of designating task to the accelerator and compare it with the overhead associated with the task allocation.

References

[1] Baker S, Kanade T. Limits on super-resolution and how to break them. Pattern Analysis and Machine Intelligence, IEEE Transactions on 2002;24(9) 1167-1183.

[2] Cheolkon Jung, Licheng Jiao, Bing Liu, Maoguo Gong. Position-Patch Based Face Hallucination Using Convex Optimization. Signal Processing Letters, IEEE 2011;18(6) 367-370.

[3] Hong Chang, Dit-Yan Yeung, Yimin Xiong. Super-resolution through neighbor embedding. Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, ; 2004. pp. I-I.

[4] Junjun Jiang, Ruimin Hu, Zhen Han, Tao Lu, Kebin Huang. Position-Patch Based Face Hallucination via Locality-Constrained Representation. Multimedia and Expo (ICME), 2012 IEEE International Conference on, ; 2012. pp. 212-217.

[5] Junjun Jiang, Ruimin Hu, Zhongyuan Wang, Zhen Han. Noise Robust Face Hallucination via Locality-Constrained Representation. Multimedia, IEEE Transactions on 2014;16(5) 1268-1281.

[6] Ma X, Zhang J, Qi C. Hallucinating face by position-patch. Pattern Recognition 2010;43(6) 2224-2236.

[7] Xiang Ma, Junping Zhang, Chun Qi. Position-based face hallucination method. Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on, ; 2009. pp. 290-293.

[8] Xiaogang Wang, Xiaoou Tang. Hallucinating face by eigentransformation. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 2005;35(3) 425-434.

[9] Wang Zhou, Bovik, Alan C., Sheikh, Hamid R., and Simoncelli, Eero P.Image Qualifty Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing 2004, Volume 13, Issue 4, pp. 600–612,

[10] Altera Corporation. Implementing FPGA Design with the OpenCL Standard. WP-01173-3.0. November 2013