Depth Assisted Background Modeling and Super-resolution of Depth Map

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Master of Philosophy
by

Boyuan Sun

Department of Electrical Engineering and Electronics
School of Electrical Engineering and Electronics and Computer Science
University of Liverpool

March 18, 2019

Abstract

Background modeling is one of the fundamental tasks in the computer vision, which detects the foreground objects from the images. This is used in many applications such as object tracking, traffic analysis, scene understanding and other video applications. The easiest way to model the background is to obtain background image that does not include any moving objects. However, in some environment, the background may not be available and can be changed by the surrounding conditions like illumination changes (light switch on/off), object removed from the scene and objects with constant moving pattern (waving trees). The robustness and adaptation of the background are essential to this problem.

Mixture of Gaussians (MOG) is one of the most widely used methods for background modeling using color informations, whereas the depth map provides one more dimensional information of the images that is independent of the color. In this thesis, the color only based methods such as Gaussian Mixture Models (GMM), Hidden Markov Models (HMM), Kernel Density Estimation (KDE) are thoroughly reviewed firstly. Then the algorithm that jointly uses color and depth information is proposed, which uses MOG and single Gaussian model (SGM) to represent recent observations of the color and depth respectively. And the color-depth consistency check mechanism is also incorporated into the algorithm to improve the accuracy of the extracted background.

The spatial resolution of the depth images captured from consumer depth camera is generally limited due to the element size of the senor. To overcome the this limitation, depth image super-resolution is proposed to obtain the high resolution depth image from the low resolution depth image by making the infer-

ence on high frequency components. Deep convolution neural network has been widely successfully used in various computer vision tasks like image segmentation, classification and recognitions with remarkable performance. Recently, the residual network configuration has been proposed to further improve the performance. Inspired by the this residual network, we redesign the popular deep model Super-Resolution Convolution Neural Network (SRCNN) for depth image super-resolution. Based on the idea of residual network and SRCNN structure, we proposed three neural network based approaches to address the problem of depth image super-resolution. In these approaches, we introduce the deconvolution layer into the network which enables the learning directly from original low resolution image to the desired high resolution image, instead of using conventional method like bicubic to interpolate the image before entering the network. Then in order to minimize the sharpness loss near the boundary regions, we add layers at the end of network to learn the residuals.

The main contributions of this thesis are investigating the utilization of the depth information for background modeling and proposing three approaches on depth image super-resolution. For the first part, the property of depth image is exploited and added into the commonly used background models. By doing so, the background model can be constructed more efficiently and accurately because the depth information is not affected by the color information. During the investigation, we found that the depth image usually has two problems, which are spatial resolution and accuracy, which need to be addressed. Most of the depth images either have small resolution or the accuracy is very bad. In the second part of this thesis, we investigate three methods to obtain the accurate high resolution depth image from the low resolution one.

Contents

List of Tables

List of Figures

List of Abbreviations

MOG	Mixture of Gaussians
SGM	Single Gaussian Model
DNN	Deep Neural Network
CNN	Convolution Neural Network
ResNet	Residual Network
SR	Super Resolution
BN	Batch Normalization
SGD	Stochastic Gradient Decent
NLM	Non-local Means
SVR	Support vector regression
TGV	Total generalized Variation
i.i.d.	Independent Identically Distributed
MRF	Markov Random Field
ToF	Time of Flight
CRF	Conditional Random Field
CSCN	Cascade of Sparse Coding based Network
LISTA	Learned iterative shrinkage and thresholding algorithm
MSE	Mean Squared Error
PSNR	Peak Signal to Nosie Ratio

RMSE	Root Mean Squared Error
SISR	Single Image Super Resolution
DSR	Depth Super Resolution
AR	Auto regressive
LR	Linear Regression
GMM	Gaussian Mixture Model
KDE	Kernel Density Estimation
PSF	Point Spread Function
GAN	Generative adversarial network

Acknowledgement

Foremost, I would like to express my sincere gratitude to my primary MPhil supervisor professor Tammam TILLO for his continuous support of my MPhil study and research, for his patience, guidance, encouragement, kindness, and immense knowledge. Besides my primary supervisor, I would like to thank my co-supervisors, Dr. Ming Xu and Dr. Waleed Al-Nuaimy.

I also thank my fellow students in the department, Li YU, Wenfei ZHU, Fei CHENG, Yanchun XIE, Dingyuan ZHENG, for their grate help in my daily and research life.

Last but not least, I would like to thank my family. Without their understanding and encouragement during the past few years. The completion of this MPhil study would be impossible task for me.

Chapter 1

Introduction

1.1 Background

In every second of our daily life, our brains has processed huge amount of information. Among these information, visual information occupies the major parts. We can accomplish many tasks through our powerful vision system like recognition of objects and person, understanding the scene. But how to give these abilities to the computers is a very challenge task due to different ways of perceiving the world. Let's take an example shown in Figure. 1.1. From humans perspective, this is a photo which contains river, mountain and trees. For computer, this is one large array of numbers that describes intensity at each position.

Figure 1.1: A sample photo.

In order to make computer ”see” and ”understand” the image/video, we need to find some certain patterns from array of numbers and transform them into a high level representations. These representations should be robust to the variations of objects and distinguishable from other kinds of objects. For example, representation of cat should be able to describe all kinds of cat no matter of what color or what size the cat is. Meanwhile, representations should be able to recognize the cat from a group of pets. This is a classic problem in computer vision which is a research field that aims to teach computer to see the world like humans.

The background modeling and depth image super resolution are two fundamental problems in computer vision. Background modeling algorithms are to construct background model to classify the objects into foreground and background. This procedure is crucial to the performance of many high level applications like object detection, image segmentation. Depth image super resolution is the technique that can obtain the high resolution image from low resolution image by some inference/mapping process. In many practical applications, high resolution depth images are desired. But due to the limited sensor size of Time-of-Flight (TOF) camera, the depth image resolution is highly limited compare to the color images. Although these two topics have been study for years, the challenge remains.

1.2 Challenges

A typical problem in background modeling is the similar color distributions exist in both background and foreground objects. This is a very common situation in the real life. For example, the video captured by surveillance camera in a shopping mall often contains various color distributions. To distinguish the background from foreground would be a difficult task from color space point of view.

Sudden illumination change is another difficult situation to deal with. There are many aspects that can cause these changes. The light in the office is switched on/off. This usually causes significant change in pixel intensities, which may make computer ”see” a totally different image. The most background model will make false classification in this case because the whole image has been ”changed”. A model that quickly updates the background information is needed for the case.

Based on the common definitions of background and foreground, background should be more stable than foreground. This poses anther difficulty for background modeling. Object with repetitive motions is easily misclassified as background due to its relatively stable state. The model should account for these kinds of objects in order to make the right classification.

Unlike the color images, the depth image usually contains many piece-wise smooth regions and sharp boundaries due to depth discontinuity. These characteristics make depth image super resolution more difficult than color image. The artifacts like blurred edges are often hard to minimize. The super resolution algorithm needs to take these two characteristics into the consideration.

1.3 Contributions

In this thesis, the objective is to improve the performance of the background model algorithm by introducing the depth information and to investigate the algorithm that can obtain high resolution depth images. The main contributions of the thesis are:

- **Gaussian Mixture Model for Background Modeling Using Depth Map:** Gaussian mixture model is one commonly used algorithm for background modeling. To handle the backgrounds with dynamic textures (such as waves on the river or trees shake by the wind), the intensity of each pixel is characterized by a mixture of K Gaussian. Once the background model is constructed, any newly observed pixel value will be classified based on the difference to its corresponding position in the background model and then update the background model. However, for this approach, the model will fail when the object has constant movement like rotations. To solve this problem, depth information is introduced to the model. The result shows that a more accurate background model can be achieved.

- **Directional Approaches of Depth Image Super-resolution Using Convolutional Neural Networks:** Unlike the color images that give more information on texture, depth image is more about the structures and the shapes. The essential characteristics of depth image are sharpness on boundaries and the more smooth region on the other parts. In this thesis, we propose to firstly decompose the high resolution depth image to three low resolution images based on vertical, horizontal and diagonal direction. Then three independent trainings are applied to train the network to learn how to super resolve the images along the directions. The result shows that the proposed method has better performance on the computer generated depth images.

- **Depth Image Super Resolution with Residual Learning:** Deeper neural network usually gives superior performance on the same task. But with the number of layers increasing, it becomes more difficult to train the network. Residual learning [6] is proposed to enable the training of much deeper neural network. In addition, residual block provides ability to learn the difference between the estimated high resolution depth image and the ground truth image.

- **Iterative approach for Depth Image Super Resolution:** The layers in convolutional neural network have the ability to learn their own features. We propose a network that contains three subnets instead of one end to end network for depth image super resolution. Each subnet will focus on learning features from low level to high level. We treat super resolution as a refinement process. The output of previous subnet will flow into the next subnet as input. In order to guarantee the consistency, the current subnet will keep partial structure of the previous subnet. Two different subnet structures have been constructed for better performance.

1.4 Organization of This Thesis

The thesis is organized as follows. Chapter 2 presents work on background modeling with depth map. Chapter 3 provides a general introduction to deep learning and depth image super-resolution. Then, Chapter 4 presents three neural network based methods to address the problem of depth image super resolution. Finally in chapter 5, contributions and limitations are summarized. Also, the future works are also included.

Chapter 2

Background Modeling Using Depth Map

2.1 Introduction

Stationary cameras are the most common video settings for capturing the activities at indoor or outdoor environment. The foreground objects can be obtained by making the comparison between current frame and the representation of background scene. This process is usually the first step for various computer vision tasks like object tracking, image segmentations, scene understandings.

The key to build a representation of the scene background is to find the proper features to construct a background model. Many types of features have been investigated for building an accurate background model including pixel based features (depth, color distributions, edges) and region based features (the correlation between blocks). The accuracy and adaptivity of the background model are depended on the selection of the features.

For either indoor or outdoor scene, there are changes that occur over time and may be classified as the changes in background. It is essential for the background model to adapt them since these changes could affect some parts of the background or the entire background. Based on the sources these changes can be categorized as:

Motion changes:

- Motions are already existed in the background. For example, waving tree leaves and rippling water.

- Changes are caused by camera displacement. This is common for the outdoor environment due to the strong wind.

Illumination changes:

- Sudden illumination change often occurs in the indoor situation. For example, switching the lights on or off.

- Gradual illumination change is usually caused by the changes in the position of the sun.

- Shadows cast on the background scene by objects itself.

Changes become the background: When a object moving into scene and stays long enough or permanent, this object will become the part of the new background. For example, if someone puts a chair into scene, or if a car is parked in the scene.

2.2 Related Work

There are many researchers who have proposed methods to deal with some of the problems for the background modeling. The following is a brief review of the relevant work.

Pixel intensity is one of the most common features for background modeling. One typical background model is using running Gaussian average [7] to model the background independently at each (i, j) pixel location. The model is based on fitting a Gaussian probability density function on the previous n pixels' values.

In this approach, the moving object (foreground) is modeled as a connected set of blobs. Each blob is represented by a spatial and color Gaussian distribution, and a support map that tells which pixels belong to the blob. The blob is interpreted as a Gaussian model:

$$Pr(O) = \frac{exp[-\frac{1}{2}(O-\mu)^T K^{-1}(O-\mu)]}{(2\pi)^{\frac{m}{2}} |K|^{\frac{1}{2}}} \tag{2.1}$$

where the μ and K are the spatial means and covariance matrices respectively, and O is the matrix of the blob. And the support map is defined as:

$$s_k(x,y) = \begin{cases} 1 & (x,y) \in k \\ 0 & otherwise \end{cases} \tag{2.2}$$

where s_k gives indication value for the pixel at (x, y) location in blob k.

However, this model would fail when the scene contains background motion, such as moving leaf or ripples in the water. Various researchers have proposed other temporal average filters that have a better performance than running Gaussian average. In [8], the median value of recent n frames is regarded as the background model. The main disadvantage of this kind of approach is that a buffer of recent observations is required for the computation. Moreover, the median filter does not have an accurate statistical analysis for the scene.

Non stationary backgrounds have been modeled by GMM algorithms [9–11]. The pixel intensity is modeled as a mixture of Gaussian distributions. The Gaussian mixture is weighted by frequency that matches the corresponding background pixel. The Gaussian model's parameters are updated at each new coming frame using EM algorithm to identify the changes in the scene. The drawback of this model is the adaptation speed. If the adaptation of the Gaussian model's parameters is fast, the foreground objects with slow movement will be classified as part of the backgrounds. But if the adaptation speed is slow, the model would fail to identify some fast changes of the background such as sudden illumination changes (switch on/off light).

Another approach to model the pixel intensity variation is using the discrete states. Hidden Markov models have been investigated in [12], [13]. In [12], the pixel intensity is modeled by a three state HMM for traffic monitoring system and the three states are representing foreground, shadows and background respectively. This model is constrained by the pixel intensity temporal continuity. Once a pixel classified as foreground state, it will keep this state for some period time before switching into background or shadow state. And this situation also apply to the pixel in other state. In [13], the topology of HMM is used to represent the global image intensity. The pixel intensity in each state is modeled by a single Gaussian distribution.

Kernel Density Estimation is a non-parametric approach that can be used to model a multi-modal Probability Density Function (PDF). In the work of [14], the model maintains a sample of intensity values for each pixel of the entire image. This sample is used to make the estimation of the density function of the pixel intensity distribution and then to predict the probability of each newly observed frame. Let $x_1, x_2, ..., x_N$ be a sample of one pixel's values observed from time 1 to time n. The estimated probability of the observed intensity at time t is:

$$Pr(x_t) = \frac{1}{N} \sum_{i=1}^{N} K_\sigma(x_t - x_i) \tag{2.3}$$

Where K_σ is the kernel function with bandwidth σ. For color images, the estimates can be generalized by kernel products:

$$Pr(x_t) = \frac{1}{N} \sum_{i=1}^{d} \prod_{j=1}^{d} K_{\sigma_j}(x_{t_j} - x_{i_j}) \tag{2.4}$$

where x_t is a color feature with dimension d and K_{σ_j} is a kernel function with bandwidth σ_j in jth color space dimension. The pixel will be classified as foreground if the $Pr(x_t)$ is less than predefined threshold. Despite the good performance achieved, the computational cost is very high because of density estimation process.

In [15], [16], convolutional neural network (CNN) based background subtraction is proposed. In [15], the fixed background model is obtained from temporal median operation over N frames. Then, the CNN is trained with scene specific data to build the background model. However, since the training data is scene specific, the network will have limited adaptation for the different scenes. For the new non-relevant scene, the network has to do the training again with corresponding data. In [16], they proposed to generate the background model by combining the segmentation mask from SuBSENSE algorithm [17] and the output of the flux tensor algorithm [18]. And the spatial median filter is applied to get rid of the outliers from the segmentation process.

Depth data is considered as fourth channel in GMM besides three channels of the color space such as RGB or YUV in [19, 20]. This approach gives less strict match condition for the depth data than texture data. But it does not utilize enough of the depth information and results can be further improved by more investigation on depth data. In [21], the depth information is exploited to identify the background regions which are covered by the object with reciprocal motion that the GMM fails to recover. The problem with this approach is the inaccuracy of the depth map, that may absorb part of foreground object into background especially at the transition positions between foreground and background regions.

This thesis describes an algorithm that fully utilize the depth data combined with GMM algorithm to estimate the background. It applies the adapting Gaussian mixture model with modified update mechanism and single Gaussian model to the expected background appearance and depth values respectively. The result greatly outperforms the prior GMM algorithms.

2.3 Proposed Method

2.3.1 Background model from GMM

The Gaussian Mixture Model has been widely applied to model the stable background and detect the moving objects. GMM is pixel based algorithm, where each pixel is modeled as a mixture of K Gaussian distributions (K is usually from 3 to 5) independently [10]. The probability of observing the current pixel value is

$$p(x_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(x_t, \mu_{i,t}, \sigma_{i,t}^2) \tag{2.5}$$

where K is the number of distributions, $\omega_{i,t}$,$\mu_{i,t}$ and $\sigma_{i,t}^2$ are the estimates of weight, mean value and variance of i^{th} Gaussian respectively at time t. And η is the Gaussian probability density function.

Every new pixel value x_t is checked against the existing K Gaussian distributions, until a match is found. The matched new pixel value is the pixel value that is within 2.5 standard deviations of a distribution [10]. And the parameters will be updated as following [9]

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} + \alpha \tag{2.6}$$

$$\mu_{i,t} = (1 - \rho)\mu_{i,t-1} + \rho x_t \tag{2.7}$$

$$\sigma_{i,t}^2 = (1 - \rho)\sigma_{i,t-1}^2 + \rho(x_t - \mu_{i,t})^2 \tag{2.8}$$

where α is the learning rate and ρ is

$$\rho = \alpha * \eta(x_t, \mu_{i,t}, \sigma_{i,t}^2) \tag{2.9}$$

If there is no match to K distributions for current pixel value, the least probable distribution will be replaced with a new distribution that has the current pixel value as its mean, an initially high variance, and low prior weight. The least probable distribution is defined as the distribution with the smallest ratio of ω/σ.

2.3.2 The exploitability of depth map

Based on the nature of the GMM, the background pixels are the pixels with temporal stable intensity. However, If the foreground object has reciprocal motion (e.g. rotation movement), which means the background is occluded by the foreground object in most frames, the GMM will erroneously classify this foreground object as part of the background. In this scenario, the GMM will not be able to recover the occluded background information. The typical example is shown Figure. 2.1, where the main body of the dancer is classified as background. The

GMM can not model a satisfactory background reference in this case.

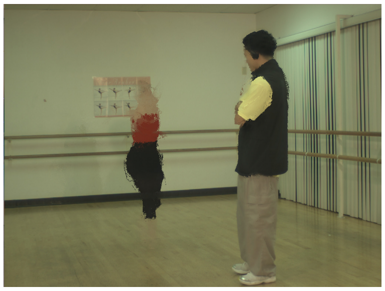


Figure 2.1: The background reference obtained using GMM after 100 frames

In this thesis, we proposed to solve this kind of problem by exploiting the depth map information. In 3D video data, the depth map measures the distance between the objects and the camera. The regions with large depth value are far away from the camera and the regions with small depth value are close to the camera. Hence, it is reasonable to assume the regions with large depth value have the high probability to be the background. Base on this assumption, by investigating the depth value of each pixel, the far field regions appear only in a small fraction of the video can be extracted and modeled as background. As illustrated in Figure. 2.3, it is clear to see that the depth value of marked area is not temporal stable. However, the depths of these areas are obviously larger compare to the depth obtained from GMM, which means these areas are highly likely to be regarded as the background.

Since the depth map is more intuitive and less complicated than texture information, we decided to applying the single Gaussian model (SGM) which is faster and requires less computations to the depth map. The SGM can be treated as a special case of GMM where only one Gaussian distribution is used for modeling the new observed depth map will be checked against the distribution of the depth model. The matched pixels will be updated using GMM. The unmatched pixels are classified into two categories. One category is the pixel with much smaller depth value, which can be coarsely regards as foreground. Another category is the pixels with larger depth value, which has the high possibility that the occluded

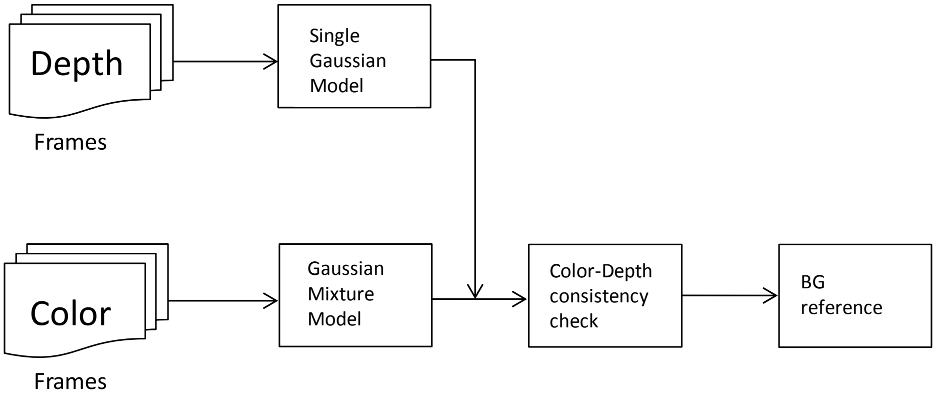


Figure 2.2: The framework of proposed algorithm

background is revealed. The general framework is shown in Figure. 2.2.

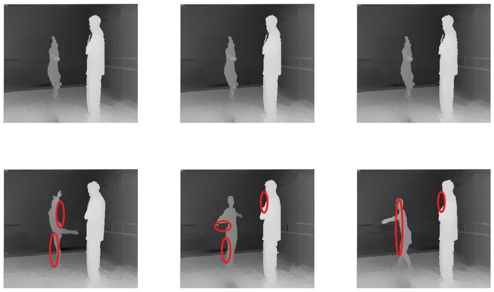


Figure 2.3: The depth map comparison between GMM result and some frames

2.3.3 Depth-color consistency check

The depth map based classification is coarse due to the accuracy limitation of the depth map. In order to refine the classification, the depth-color consistency check mechanism is required for the process, especially for the pixels could be the revealed background. These new pixels' values will be checked against the most probable Gaussian distribution. If there is no match, these new pixels' values can be regarded as part of the background. Otherwise, it shows that this kind of scenario is caused by the inaccuracy of depth map as mentioned before. One more possibility of this case is that the region with similar color and long distance is revealed, which will not affect the estimated background since there is no major change in color. For the pixels with smaller depth values, we just simply do not update those pixels since the background information is the major concern.

The flowchart for the operations on each incoming frame is illustrated in Figure. 2.4 and the details of proposed method is explained in Algorithm 1.

Algorithm 1: Gaussian Mixture Model using Depth Map

1: The empty set of models for color is initialized at the time t_0.

- Assigning the pixel value of current frame to the mean value μ_{i,t_0} of the first Gaussian model and the rest is set to 0.
- The variance σ^2_{i,t_0} of all Gaussian model is set to predefined large value, e.g., 900 in this work.
- The weight of first Gaussian model is set to 1, and the others are set to 0.
- The model of depth map is set according to single Gaussian model [7].

2: Compare the current depth observations with the existing model.

3: For the pixels that match to the existing model, the pixels' value will be updated by using GMM. The pixels with unmatched small depth value will not be updated and their corresponding mixture model remains the same. The rest pixels will go through the depth color consistency check mechanism.

4: The mixture model of the pixels with consistent depth-color change will be reset as the procedure 1. The other pixels' model will be updated by GMM.

2.4 Experimental results

In this section, we tested the proposed depth assisted GMM algorithm on three video sequences. The test sequences include: Microsoft data set Ballet (1024×768, 100 frames), Break-dancer (1024×768, 100 frames) and the MPEG-3DTV test sequence Arrive-book (1024×768 , 50 frames). And the depth maps of the

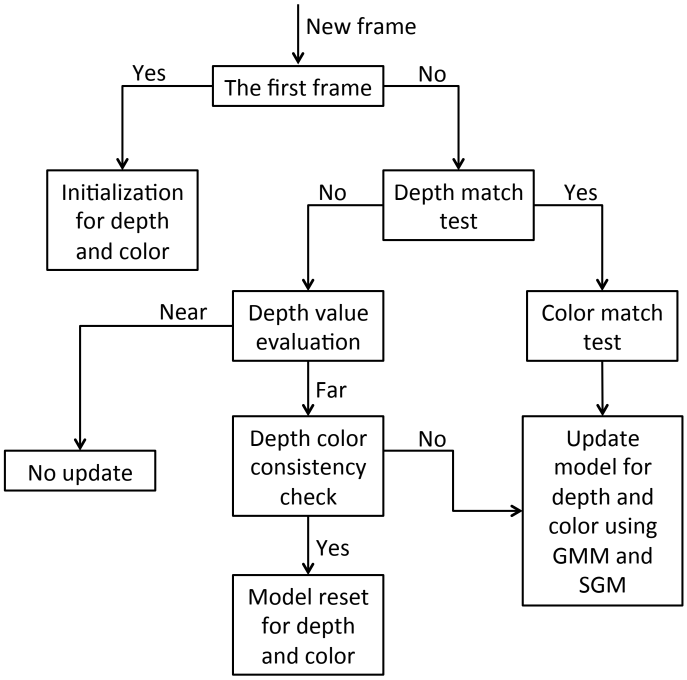

New frame
Yes
The first frame
No
Initialization for depth and color
No
Depth match test
Yes
Near
Depth value evaluation
Color match test
Far
No update
Depth color consistency check
No
Update model for depth and color using GMM and SGM
Yes
Model reset for depth and color

Figure 2.4: The detailed flowchart of the proposed method

test sequences are obtained from the MPEG depth estimation reference software (DERS) based on graph cuts [22]. The results are compared with the manually marked ground truth which is available in www.mmtlab.com/download.

The results shown in Figure. 2.5, 2.6, 2.7 for proposed method are compared with previous GMM method at a few selected video frames from test videos. In the ballet video, the female dancer is continuously rotating with small movement on the floor. The male is almost static and he only moved his body very slightly. It is obvious to see the GMM method failed to recover the background that is occluded by the dancing women. Our approach is able to recover the most occluded background region even for the small part covered by the hand of male dancer. For the arrive-book and the break-dancer video, since there is no large background region constantly occluded by the foreground objects, the proposed method still has slightly gain in subjective view. In addition, the moving objects in the scene are quite close to the static background, which makes the modeling process more challenging. As shown in Figure. 2.5 and Figure.

2.6, our method recovered the small part of background occluded by the sitting man and the most part of background covered by dancing man in the front, in arrive-book and break-dancer video respectively. The noise appears in the result of proposed method, e.g., the small black blocks on the ground in Figure. 2.7 is caused by the imperfect depth map information. For the objective assessment of the proposed method, PSNR is used. The frame by frame objective results are demonstrated in Figure. 2.9, 2.8, 2.10. The red line represents the PSNR results of our proposed method and the green dash line represents the results of GMM method. The PSNR is calculated by comparing the output of GMM and the proposed method with the ground-truth image frame by frame. The PSNR curves show that the proposed method is much better than the GMM method for ballet and arrive-book video sequences. As for break-dancer sequence, PSNR of our method is slightly less than the GMM method. This is due to the inaccurate depth map estimated from DERS. The accuracy of the depth map is decreased along with the increased degree of the object's movement. In the break-dancer sequence, the dancer made huge movement which caused the major changes in depth map especially when dancer's leg moved to the position that is very close to the ground. It is worth to mention that the depth process do not increase computational time too much. The proposed method runs at about 12fps while the GMM runs at about 15fps.

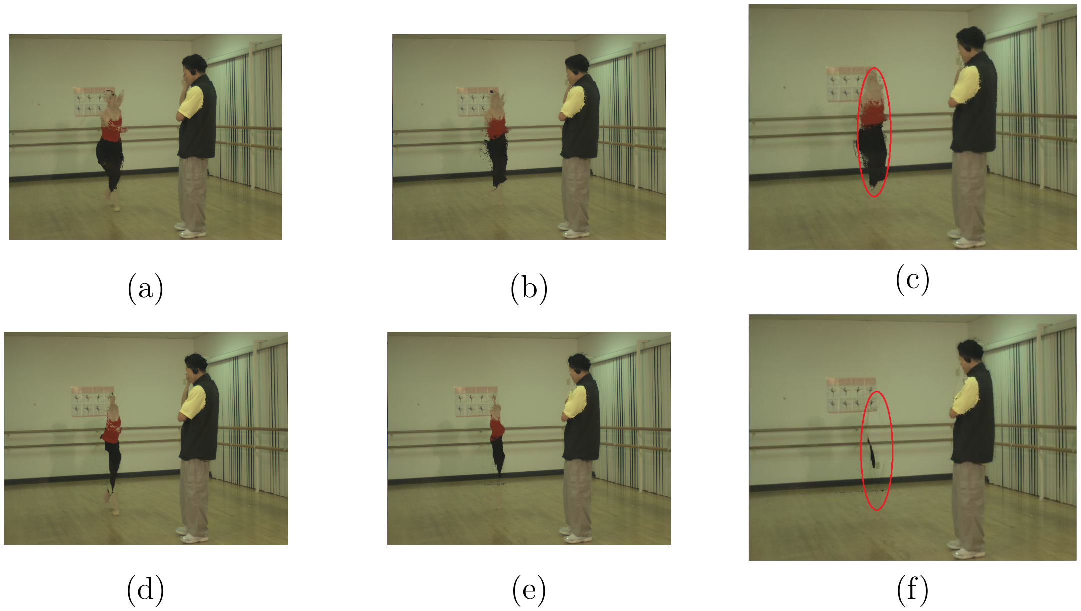
2.5 Discussion

The performance of the proposed method highly depends on the accuracy of the depth map. While we conducted experiments on the depth map based method, we have noticed that the depth map obtained from stereo matching algorithm or Microsoft kinetic camera does not have good quality (accuracy) although the spatial resolution is high. In Figure. 2.11 it shows 9 consecutive depth frames of ChairBox sequence [23], which clearly shows the discontinuity of depth. It is easy to find that the depth of many regions are changed dramatically in the consecutive frames. A more quantitative results shown in the Figure. 2.12 is the plot of the depth value of some random pixels from the first frame to the end. This phenomena encourages us to make efforts on investigating the depth map super-resolution technique.

2.6 Conclusion

In this thesis, we proposed a depth map assisted Gaussian Mixture Model approach to handle the foreground object with reciprocal motions in the video. Through the experimental evaluation, we have showed this approach has a much better performance for the object with reciprocal motion in the popular test sequences. For the future works, we will exploit the spatial correlation between the pixels for both depth and color data and then combine all the information to improve the background subtraction algorithm. In addition, the motion information of the texture will be investigated to enhance the performance.

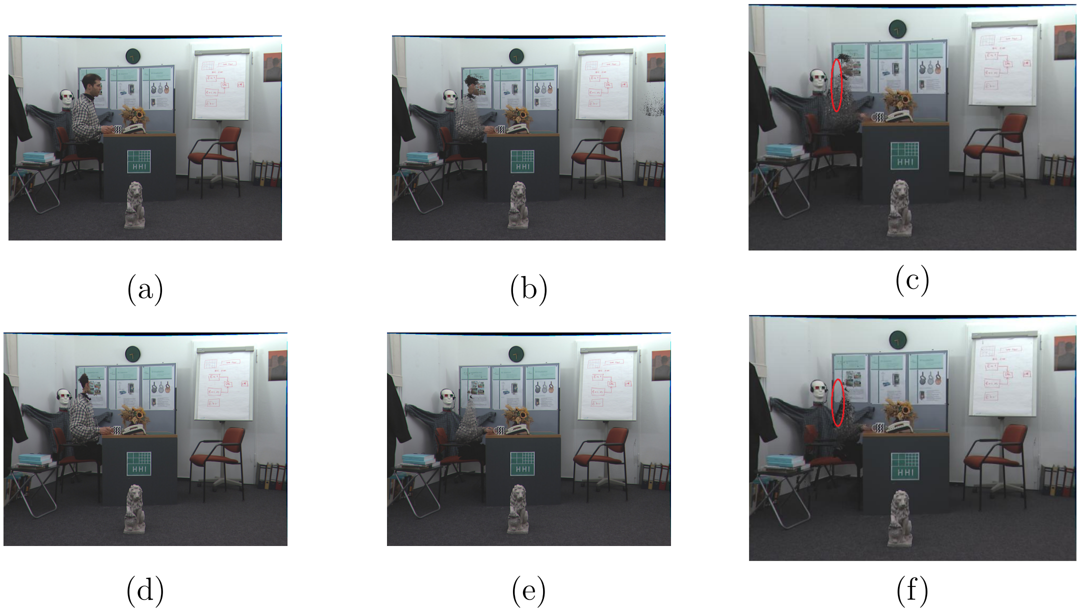
Since quality of the depth map is very important to accuracy of the proposed method and it is not easy to obtain the depth map with good quality and proper spatial resolution corresponding to the color image at the time of conducting this research. We decided to make some efforts on investigating depth map super-resolution techniques to get high resolution depth map from time of flight camera which generates high quality depth map with very low resolution. This is the reason why this thesis has two parts - one for GMM modeling and the other for the depth map super-resolution.



(a) (b) (c)

(d) (e) (f)

Figure 2.5: The experiment results for GMM (from a to c) and the proposed method (from d to f) at frame 10, 30, 50 of ballet sequence.



(a) (b) (c)

(d) (e) (f)

Figure 2.6: The experiment results for GMM (from a to c) and the proposed method (from d to f) at frame 10, 30, 50 of arrive-book sequence

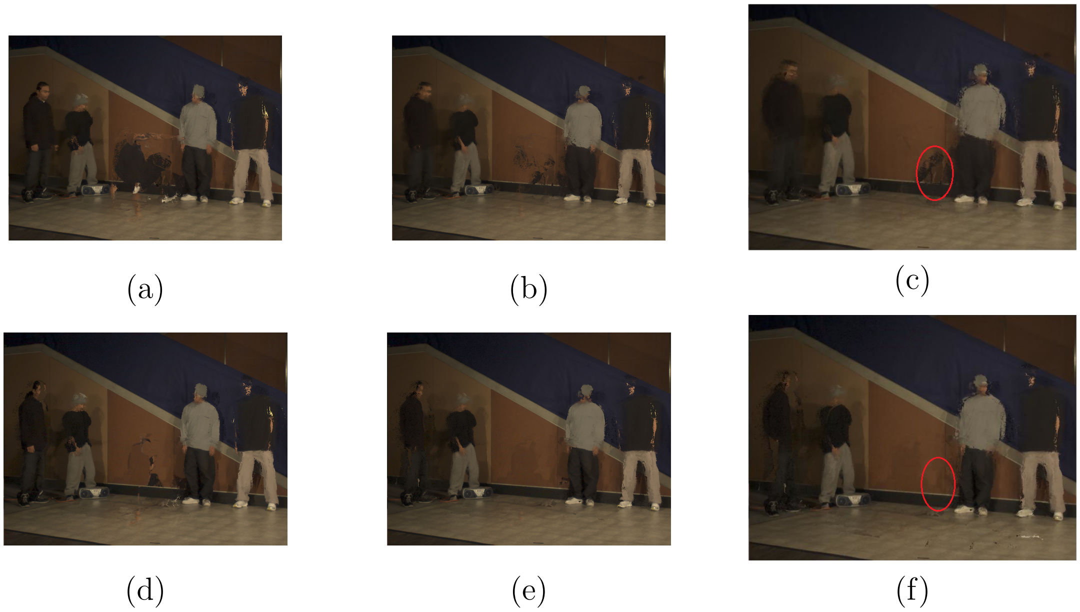


Figure 2.7: The experiment results for GMM (from a to c) and the proposed method (from d to f) at frame 10, 30, 50 of dancer sequence

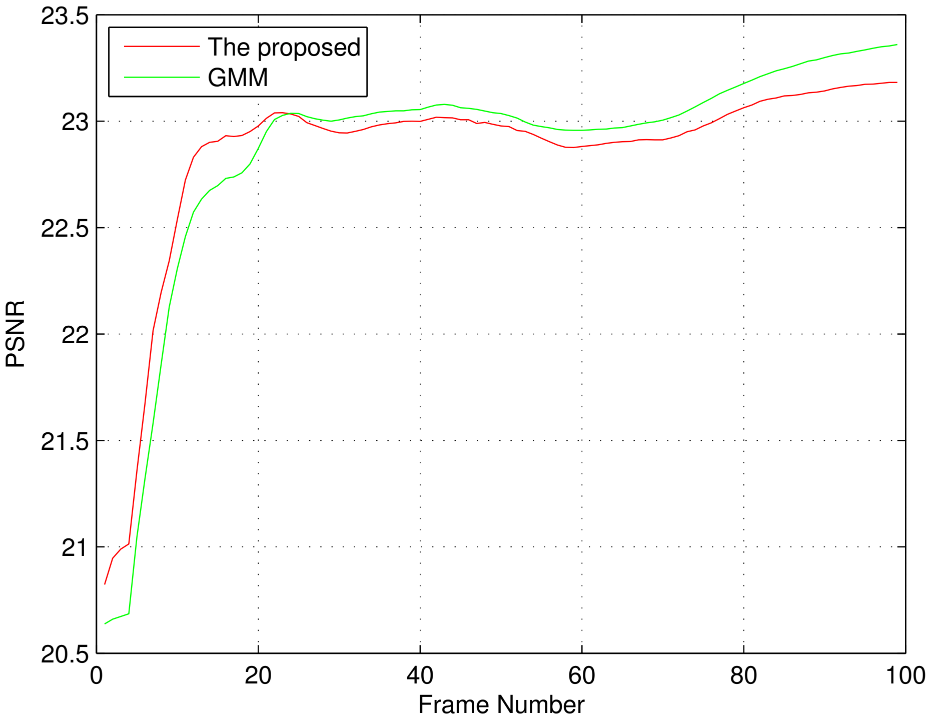


Figure 2.8: PSNR evaluation for Break dancer Sequence.

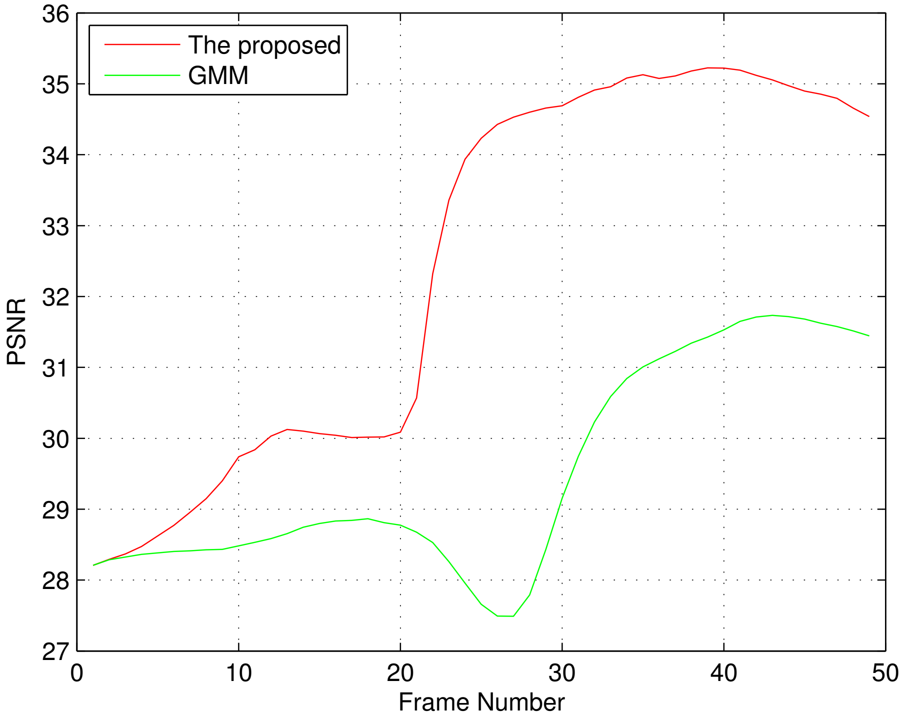


Figure 2.9: PSNR evaluation for Arrive-book Sequence.

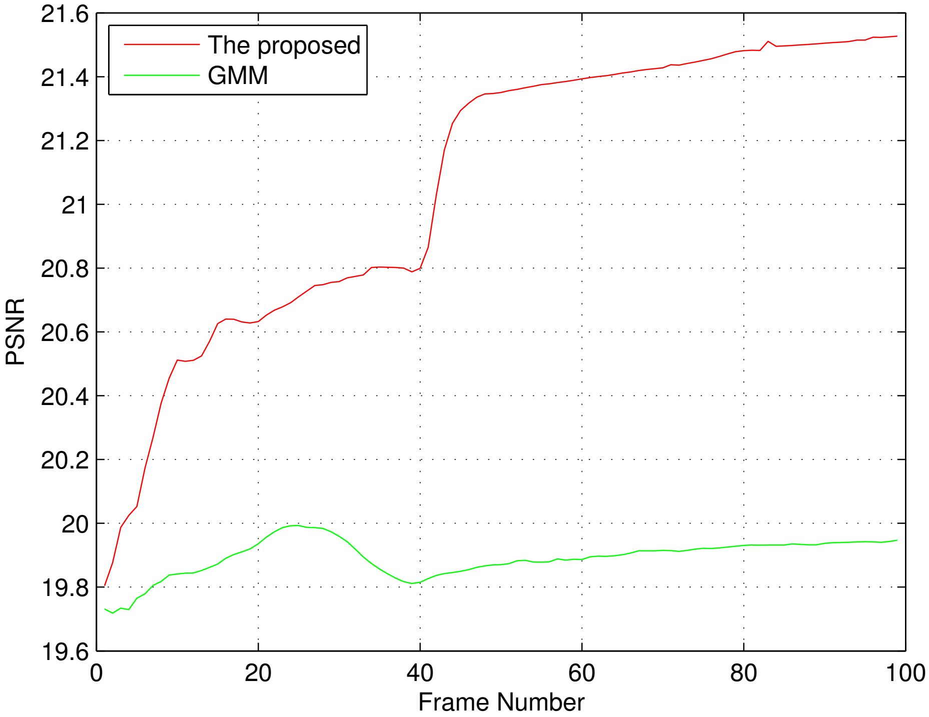
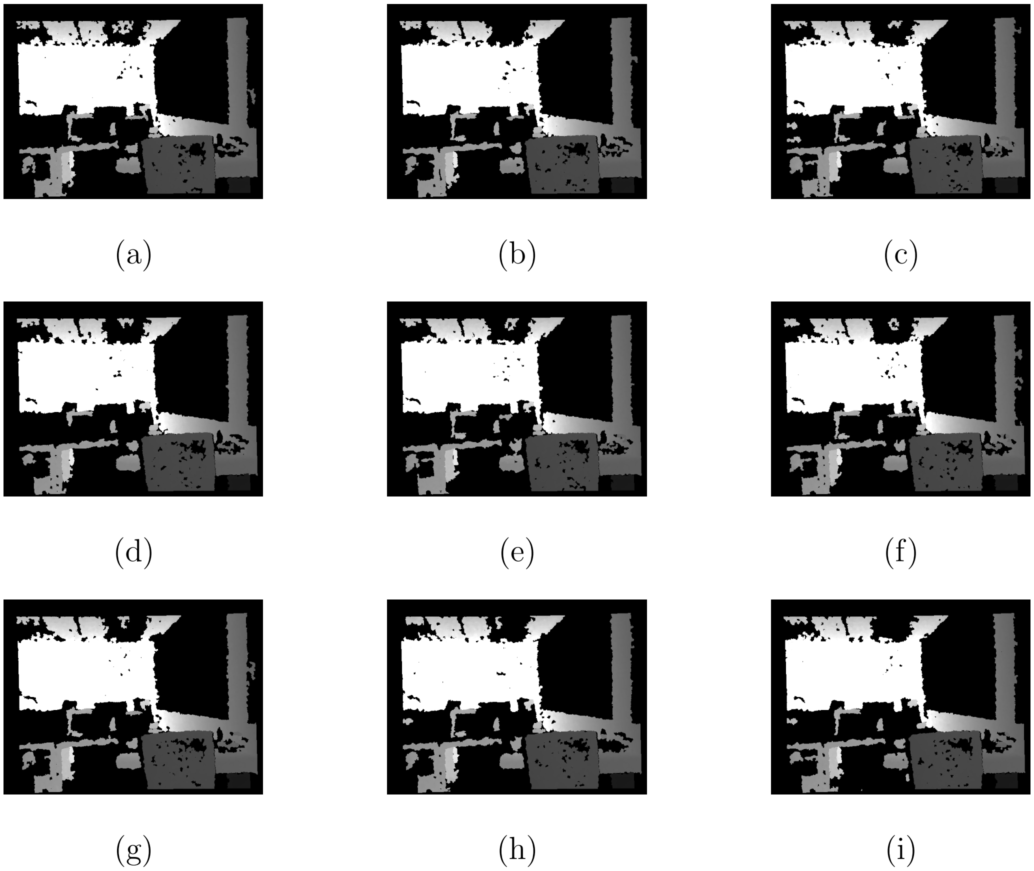


Figure 2.10: PSNR evaluation for Ballet Sequence.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Figure 2.11: The first 9 consecutive depth frames in ChairBox sequence

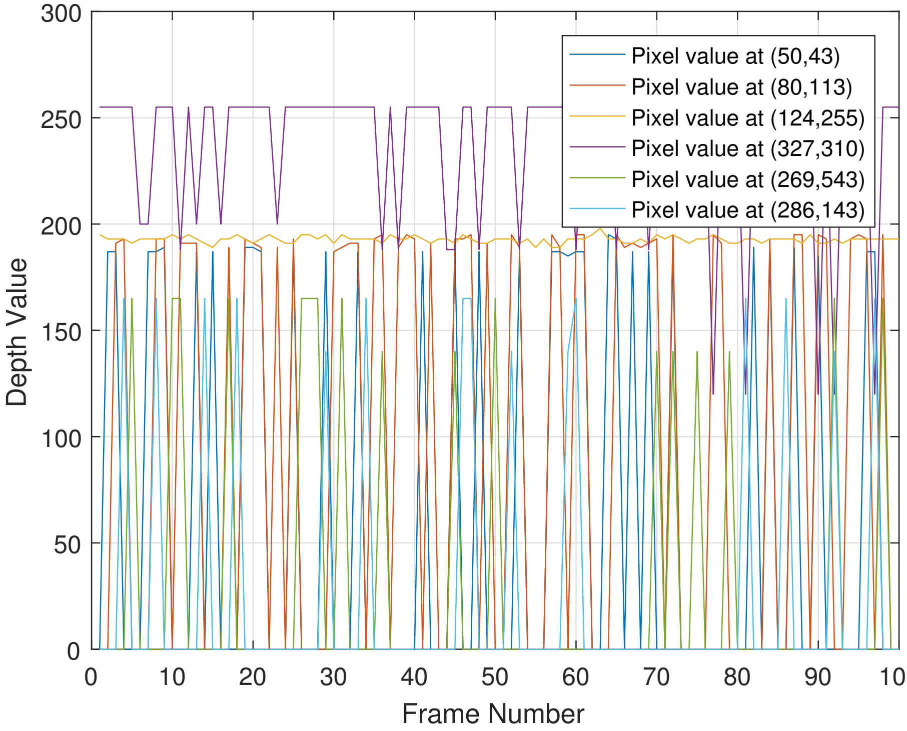


Figure 2.12: The depth values at 6 positions of first 100 frames ChairBox sequence.

Chapter 3

Introduction to Depth Image Super-resolution and Deep learning

3.1 Depth Image Super-resolution

During the past several years, consumer depth cameras like Microsoft Kinect and time-of-flight cameras have become more and more popular in many research fields human computer interaction [24], computer graphics [25] and 3D modeling [26]. However, depth images obtained from these cameras are either bad quality (low accuracy and not stable) or have very limited spatial resolution. For example Figure. 3.1 showed a typical depth map generated by Kinect. The depth image super-resolution (DSR) has gained great attention in the community. The DSR aims to restore the high resolution image from low resolution image by inferring the lost high frequency contents (image details), this makes DSR an ill-posed problem due to the insufficient knowledge.

3.2 Related Work

Single Image Super Resolution: Image super resolution is one of the most active topics in the field of low level computer vision. Single image super resolution has been studied for many years. In [27] and [28], they proposed a multi-class Markov Random Field (MRF) model to solve the super resolution problem. In

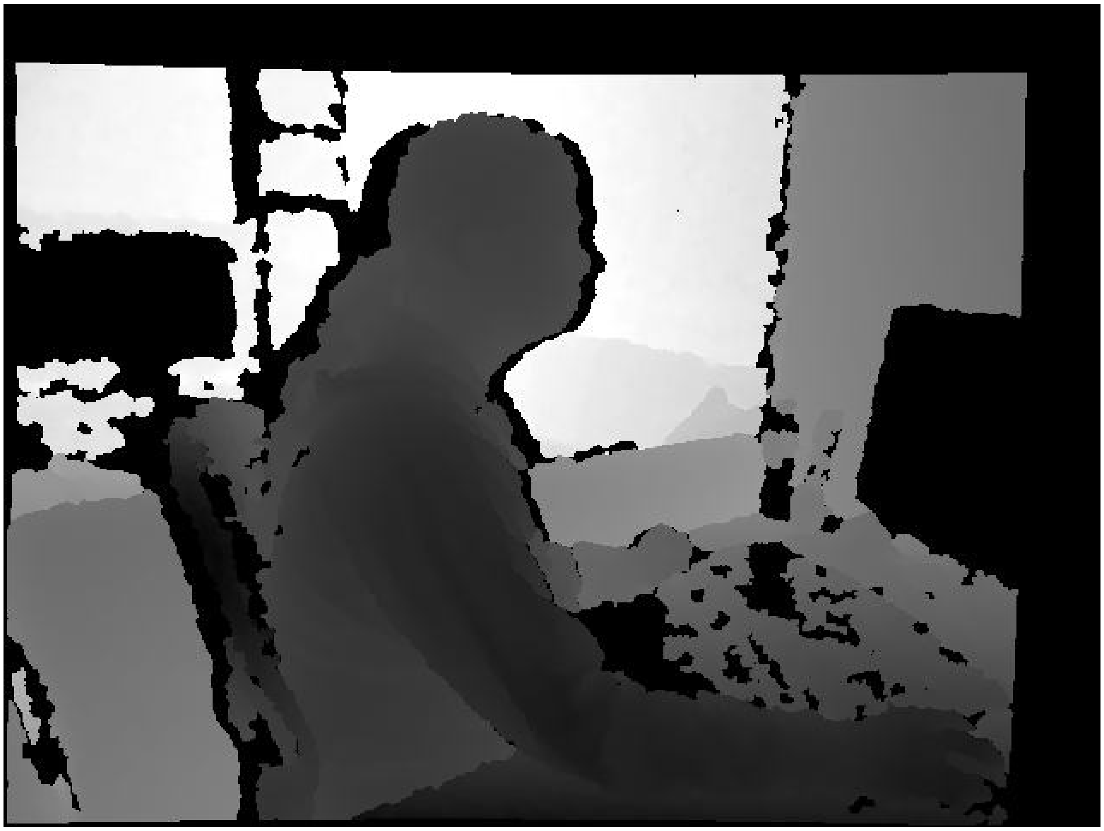


Figure 3.1: Depth map obtained from Kinect

this MRF model, each hidden node is used to represent the label of the high resolution patch. The reconstruction process largely depends on the available training examples. The performance will be degraded if there is no correspondence to be found.

Dictionary learning and sparse representation of images are also exploited to deal with image super resolution problems. Based on assumptions that low resolution and high resolution patches could share some reconstruction coefficients, [29] and [30] proposed a method that uses sparse linear combination of the learned dictionary to reconstruct the high resolution images. In [31] and [32], the high resolution image is obtained by learning a mapping from low resolution image to high resolution image with relaxed fully coupled constraint. The work in [33] proposed a self learning super resolution algorithm that employs support vector regression (SVR) with sparse representation. High resolution reconstruction is obtained from SVR model that is learned by minimizing the error function. The disadvantage of this algorithm is that data collection of low and high resolution training image and prior knowledge of the self similarity are needed. These requirements are not usually satisfied.

Self-similarity is also widely exploited for the task of image super resolution. These methods suggest that patches of natural image will re-occur within and across scales of the same image. The work of [34] proposed to super resolve the low resolution image by gathering the information of similar image patches while there is no need to prepare the training data beforehand. However, the assumption of existence of the image patch redundancy is not guaranteed. This is the key issue for which the self-similarity based methods are not very suitable for depth image which does not have much texture patterns.

Recently, the deep learning methods have showed a powerful capability for super resolution. In [2], the convolutional neural network for image super resolution is proposed (SRCNN). SRCNN has the ability to learn a non-linear mapping between low resolution and high resolution images.

Depth Super Resolution With Multiple Images: Conventional depth image super resolution is to combine the information [35–37] of the multiple complimentary low resolution images. Although good results have been achieved, the requirement for multiple available static image with small camera movement is not always satisfied for many real applications. In addition, the camera pose estimation errors have large influence on the super resolution result.

High frequency components of the pre-aligned high resolution color images can be exploited to help the depth image super resolution. In [38–42], edge information from high resolution image is utilized to perform a joint color and depth upsampling. In order to keep the detailed structure of depth image, the work of [41] proposed to use non-local means filter (NLM) for the task of super resolution.

Single Depth Image Super Resolution: Single depth image resolution posed more challenges than color images because of the sharpness on boundary and less texture information. In [43] and [44], they extended the work in [27] to depth image by applying patch based MRF model to super resolve the depth image. In addition, the bilateral filter is often used to maintain the sharp edges while reducing the noise of the depth image. The work of [45] tried to look for self-similar 3D patch correspondence through the rigid body transformation and then to construct the high resolution image patches. However, despite the good results obtained, the performance of these methods will be degraded when the patch correspondence from extra dataset or within the same image is failed to be established. In [46] and [47], sparse representation of depth image is introduced to prevent the over-fitting problem of the learning based methods.

3.3 Introduction to Deep Learning

Deep learning (Deep neural networks) has draw great attention during past several years for its outstanding performance in many challenging machine learning and computer vision tasks. Since all of the following proposed methods are based on neural network, we will provide the necessary technical background about deep learning especially the Convolutional Neural Network (CNN) in this section. For a more comprehensive introduction, please refer to the deep learning book from Goodfellow et al. [48].

3.4 Supervised Learning

For many practical problems, the learning task can be formulated as training the computer to perform a mapping $f\colon X \to Y$, where X and Y are the input and output space respectively. For image super-resolution task, X could be the space of low resolution image (LR) and Y could be the space of upscaled high resolution (HR) image. Due to the complexity of image structures, it is very difficult to explicitly write down a program that upscale a LR image to a HR image by using conventional methods. The supervised learning provides a second option by learning a mapping from LR images to HR images since it is relatively easy to have the paired examples $(x,y) \in X, Y$. In this thesis, the paired examples are datasets of LR images and their corresponding HR images.

3.4.1 Objective

Considering we have a training dataset of n examples from a data distribution D, $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ that are independent and identical distributed i.i.d. samples. The learning objective is to find the mapping $f \colon X \to Y$ by searching from a set of candidate functions and finding the one that is a best fit for the training examples. More specifically, we will choose some particular class of functions F as candidates and a loss function $L(\hat{y}, y)$ that measures the difference between the estimated label $\hat{y}_i = f(x_i)$ for some $f \in F$ and the true label y_i. Then our objective is to find $f^* \in F$ that minimize the loss over the training examples. Once the f^* is determined, we can use it to map X to Y without keeping the original training data.

3.4.2 Linear Regression

Linear regression is one of the commonly used algorithms in supervised learning. As shown in Figure. 3.2, the directed edge between two nodes shows the output of one node is fed into the another. And one node can receive (transmit) the signal from (to) multiple nodes. The signal on each edge will be multiplied by its corresponding weight. This can formulated as:

$$\hat{y_i} = \sum_{i=1}^{n} x_i w_i + b \tag{3.1}$$

where $\hat{y_i}$ is the output of linear node that takes all p incoming values multiplied by the weight of the edge, w_i is the weight of i-th edge, and b is a bias of the node. By using this linear unit, we create a simple neural network that can learn some unknown function f given a training dataset D.

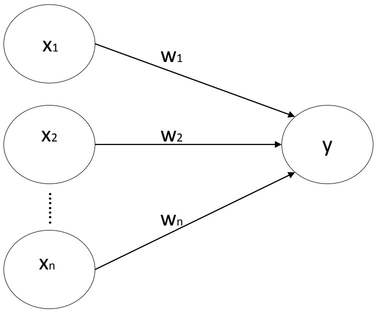


Figure 3.2: Illustration of simple linear regression network

The objective is to search for a vector of weights $w = [w_1, \ldots, w_n]^T$ that gives minimum loss on the training set D. This can be achieved by minimizing the squared error between the true output $y^{(i)}$ and the estimated output $\hat{y}_i$ with respect to w using Stochastic Gradient Descent (SGD) method:

$$\hat{w} = \arg\min_{w} \sum_{i=1}^{n} ((y_i - \hat{y}_i))^2 + \lambda R \tag{3.2}$$

where R is the regularization term, and λ is the strength of R. Regularization is method to avoid the model over-fitting to training dataset by controlling the complexity of the model.

It is easy to find that this kind of structure can mimic linear function well. When it is applied to approximate a non-linear function, this network is not expected to give an accurate approximation. This is why the deep neural network should be considered.

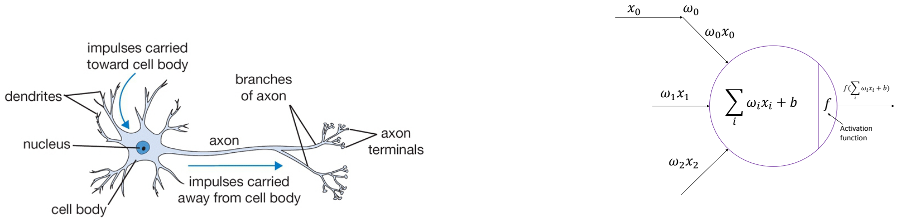


Figure 3.3: LEFT: a sketch of biological neuron, RIGHT: mathematical model of biological neuron. Figure is from [1]

3.5 Deep Neural Network

In the previous section, we have briefly introduced linear regression which can be regarded as simple neural network. Deep neural network has more layers between the input and output layer and it has non-linear activate function which gives its capability to mimic all kinds of functions especially non-linear functions. In the following section, we will give some detailed descriptions on deep neural network.

3.5.1 Model

The basic computation unit in neural network is called neuron. As show in Figure. 3.3, it gets inspiration of biological neuron. Each neural perform the computation with activation function. Figure. 3.4 shows an example of deep neural network with 2 hidden layers. Each hidden layer has four neurons. In Figure. 3.4, the leftmost layer of the network is input layer, and the rightmost layer is output layer. The 2 layers in the middle are the hidden layers.

The computation of hidden layer 2 and final in this network can be represented as following:

$$a_1^{(2)} = f(W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + W_{13}^{(1)} x_3) \tag{3.3}$$

$$a_1^{(3)} = f(W_{11}^{(2)} a_1^{(2)} + W_{12}^{(2)} a_2^{(2)} + W_{13}^{(2)} a_3^{(2)} + W_{14}^{(2)} a_4^{(2)}) \tag{3.4}$$

$$y = f(W_{11}^{(3)} a_1^{(3)} + W_{12}^{(3)} a_2^{(3)} + W_{13}^{(3)} a_3^{(3)} + W_{14}^{(3)} a_4^{(3)}) \tag{3.5}$$

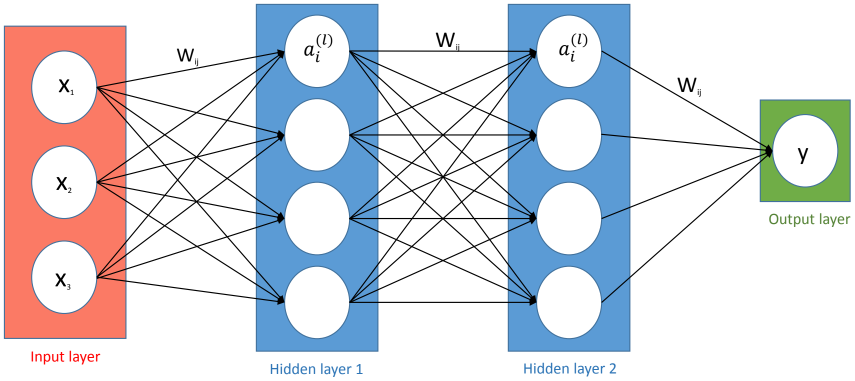


Figure 3.4: Illustration of a 3 layer neural network example: The input layer has 3 inputs, w_{ij} is the weight on the edge of i-th neuron and j-th input, $a_i^{(l)}$ is the activation of i-th neuron of l-th layer

3.5.2 Activation Function

The non-linear activation function of the deep neural network plays an essential role in learning a good unknown function approximation. There are three mainly used activation functions.

Sigmoid function The mathematic form of sigmoid function is

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.6}$$

The sigmoid function adjusts the output in the interval $(0, 1)$ as shown in Figure. 3.5 using this function is easy to obtain the gradient. It can be used as output layer. However, if the neuron is saturated at 0 or 1, the gradient vanishing problem will appear and the backpropagation algorithm will fail at updating the parameters. This could lead to some unexpected training problems.

Tanh function The *tanh* function is defined as:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.7}$$

The *tanh* function maps the input into the output interval $(-1, 1)$ as shown in

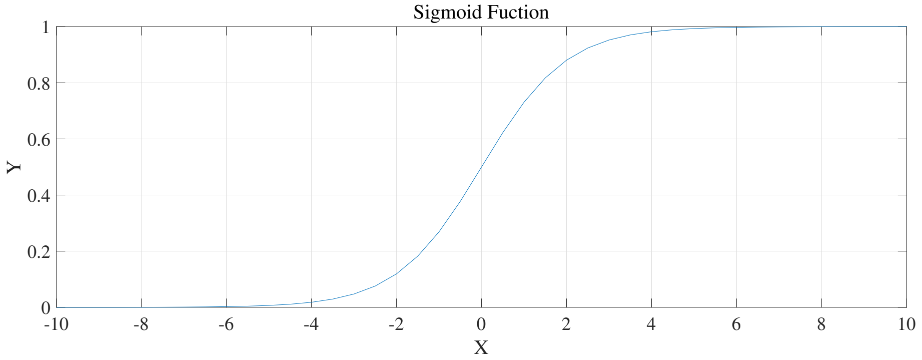


Figure 3.5: Illustration of sigmoid function

Figure. 3.6 Compared with sigmoid function, the *tanh* function has the advantage of faster convergence speed and it is centered at 0. But it still has the gradient vanishing problem when the neuron reaches saturated state just like sigmoid function.

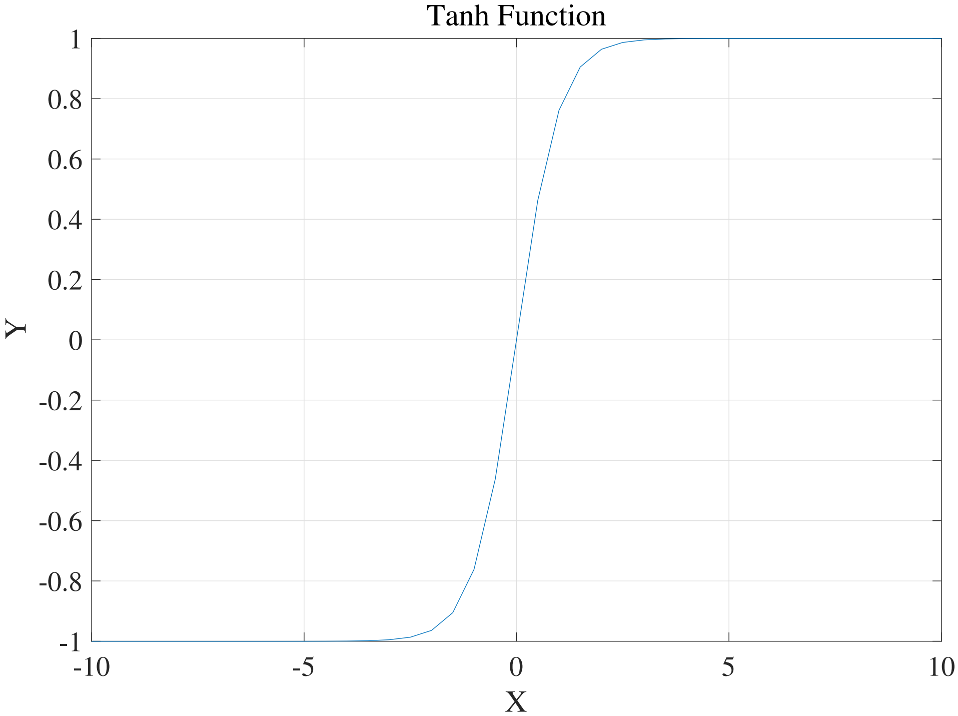


Figure 3.6: Illustration of *tanh* function

ReLU Rectified Linear Unit (ReLU)[49] is defined as

$$f(x) = \max(0, x) \tag{3.8}$$

The ReLU has become a very popular choice of activation functions in the past several years. It demonstrated its capability of accelerating the convergence speed when the neural network uses stochastic gradient descent method. In addition, unlike the sigmoid and *tanh* function, ReLU does not have the problem of vanishing gradient. It gives neural network ability for sparse representations. The drawback of ReLU is the possibility of appearing dead neuron during the training process. The weights of those dead neurons will not be updated since its gradient is always zero. In [50], Parametric ReLU is proposed. Compare to ReLU, it can adaptively learn the parameters which provides the possibility to reduce the number of dead neurons.

$$f(x_i) = \begin{cases} x_i, if x_i > 0 \\ a_i x_i, if x_{\leq} 0 \end{cases}$$

3.6 Convolutional Neural Networks

Regular neural network only has linear layers, which is not quite suitable to process the inputs with some spatial topology (e.g images, videos, characters in documents). Convolutional Neural Networks [51] is proposed to take spatial relationships into considerations.

Convolutional Layer. This is the essential building block of the convolutional neural network. It performs convolution on the input tensor with a set of filters and then generates the output tensor as shown in Figure. 3.7. For instance, images will be the input tensor. Let's take a color image with resolution of 256 by 256 as example to illustrate the process of this layer. The input tensor I will have the size of $256 \times 256 \times 3$. Assume that the size of the filter w is $5 \times 5 \times 3$ i.e. there are $5 \times 5 \times 3$ parameters to learn. The convolution is performed by sliding the filter w on every possible spatial positions of the input and the dot product of a block of I and filter w will be computed. Then the activation map is generated, which has the dimension of 251×251 if there is no zero paddings on the boarders (the activation map will have the same dimension of the input tensor when zero padding is applied). This is the process for one filter. Usually we will have a set of filters and each filter will generate its own activation map. These activation maps will be put together to produce a final output tensor. It is easy to find out each filter has the ability to learn some certain feature.

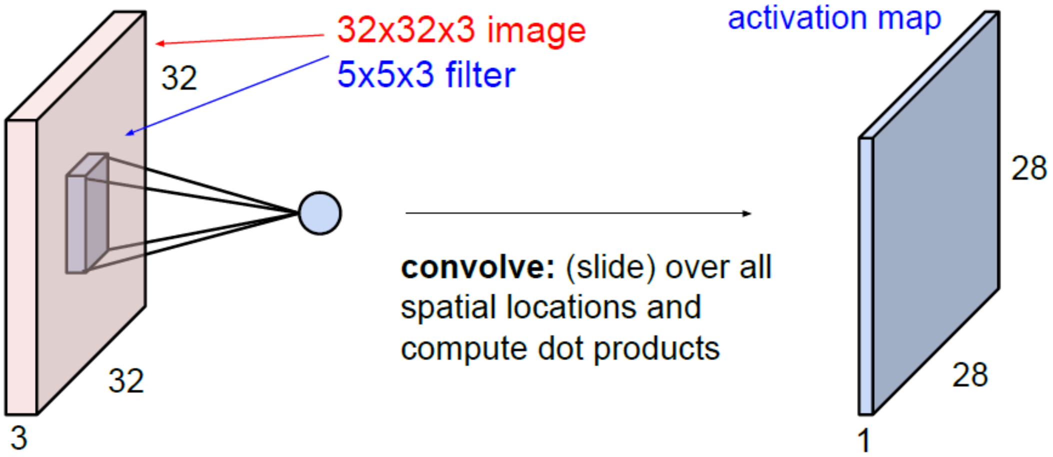
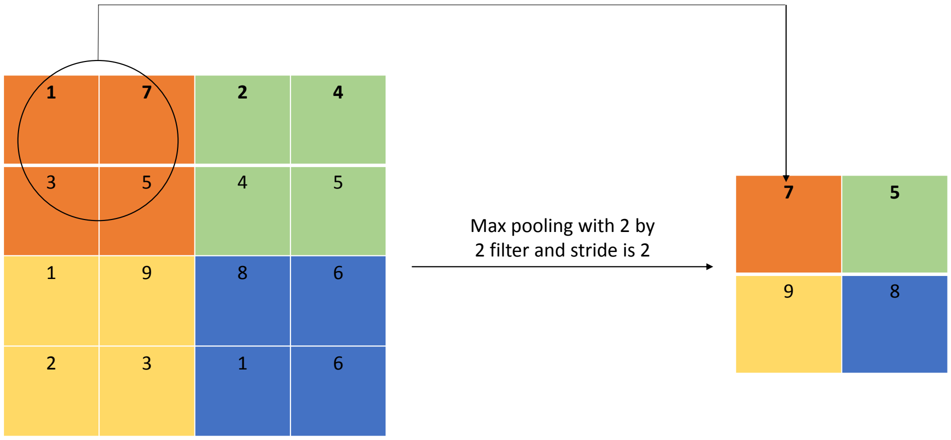

32x32x3 image
5x5x3 filter
32
32
3
activation map
28
28
1
convolve: (slide) over all spatial locations and compute dot products

Figure 3.7: Illustration of convolution of 5×5 filer on a $32 \times 32 \times 3$ image, figure is from [1]

Pooling. This layer performs downsampling process on all activation maps independently to reduce the dimension of each activation map while the most important features are retained. This procedure is helpful to control the over fitting problem. In addition, it makes network invariant to certain level of transformations, distortions and translations. One of the most common settings is to use 2×2 filters with stride of 2, where *max* operation is applied to the filter as show in Figure. 3.8.

Batch Normalization. This layer is proposed in [52] to handle the phenomenon called internal covariate shift which indicates that the parameters update will affect the input distribution of the next layer so that the weights are needed to re-initialization and reduce the learning rate. Batch normalization is to do normalization on each batch of each layer. The mini-batch mean and variance are computed first and then the feature map is normalized. By doing



1	7	2	4
3	5	4	5
1	9	8	6
2	3	1	6

7	5
9	8

Figure 3.8: Illustration of Max pooling with 2 by 2 filter and stride is 2

so, it helps the neural network to converge quickly without consuming too much computational power.

Chapter 4

CNN based Depth Image Super-resolution

4.1 Iterative Network Approach

Authors in [82] proposed to stack 2 SRCNN networks into one and then start the training process from scratch. The estimated high resolution images are closer to the ground truth images than the images estimated by single SRCNN network. Inspired by this work, we treat super resolution task as refinement process from coarse interpolated images to well interpolated images. Different from [82], we propose a super resolution network that consists of three sub-net as show in Figure. 4.1. Each sub-net consists of a set of convolution layers and one residual learning block. Many research experiments indicated that the features learned by convolution layer has the order from low to high i.e. convolution layer in the front will learn low level features and layer in the later part will learn high level features. So we make later sub-net adopt the first convolution layer of previous sub-net, which has the learning rate of zero, to keep the learned features unchanged in the first layer.

The work flow of the network training process is shown from Figure. 4.2 to Figure. 4.4. The low resolution image is firstly interpolated to the same size of ground truth image by bicubic method. There are three phases for the network training. Firstly, training is on sub-net 1. The estimated image Est_im1 is produced and the low level features are also learned and passed to the sub-net 2.

Then, the Est_im1 will be fed into sub-net 2 as input for training. Layers $L2.2$ to $L2.5$ will be tuned and the higher level features will be learned in sub-net 2. Finally, the sub-net 3 takes output of sub-net 2 Est_im2 for training along with the learned features of sub-net 1 and sub-net 2.

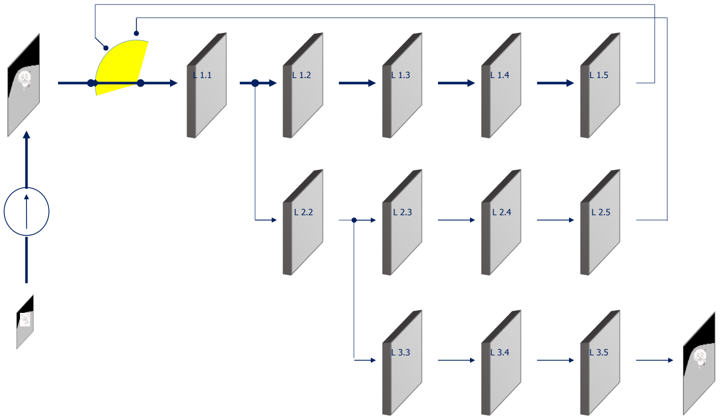


Figure 4.1: The overall structure of the iterative nets

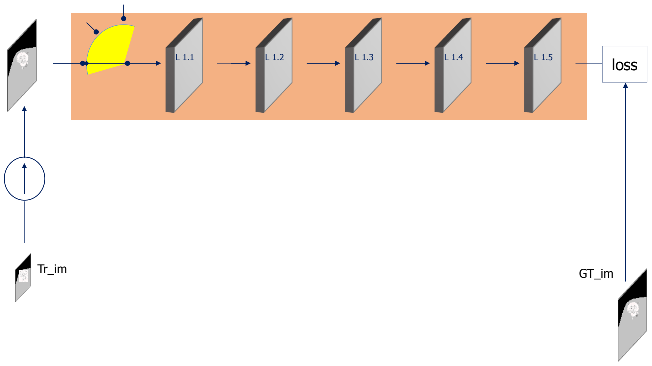


Figure 4.2: Training Phase 1

4.1.1 Subnet Structure

The unified network structures in each iteration has limited the network's ability to learn the different kinds of features of the input image patches. And the image can be treated as the composition of the low frequency and high frequency components. In this proposed network, each sub-net will have two path of training

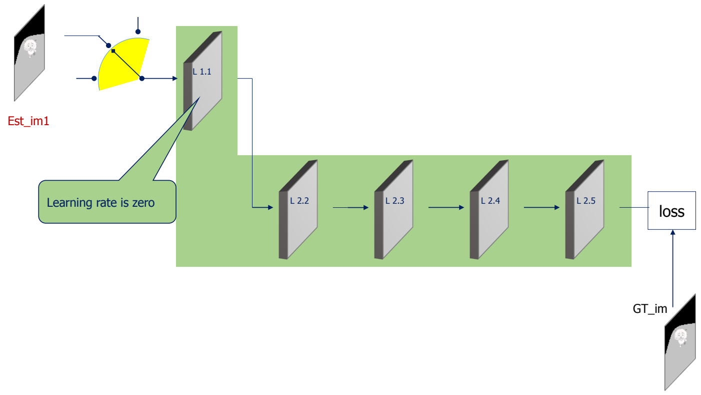


Figure 4.3: Training Phase 2

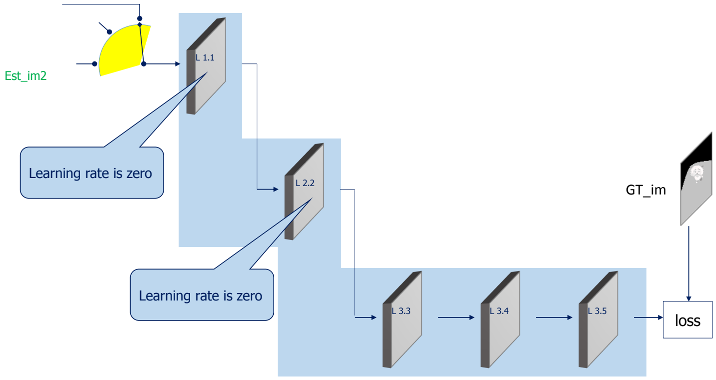


Figure 4.4: Training Phase 3

and then combining together to generate the high resolution depth image. In each sub-net, one shallow path will have a relative simple network to concentrate on training the network to learn the low frequency part. Another deep path will have the complex structures with more residual blocks to perform the identity mapping through short cut (skip the several following layers), which allows the network learn more about the high frequency part i.e image details. With multiple identity mappings in the network, the gradient flow is going to be even better since we adopt SGD as the optimization method for training the neural network. In addition, inspired by work [58, 62, 83] , multi-scale inference has been added to the later part of the network for better reconstruction performance. The working flow for one sub-net is shown in Figure. 4.5.

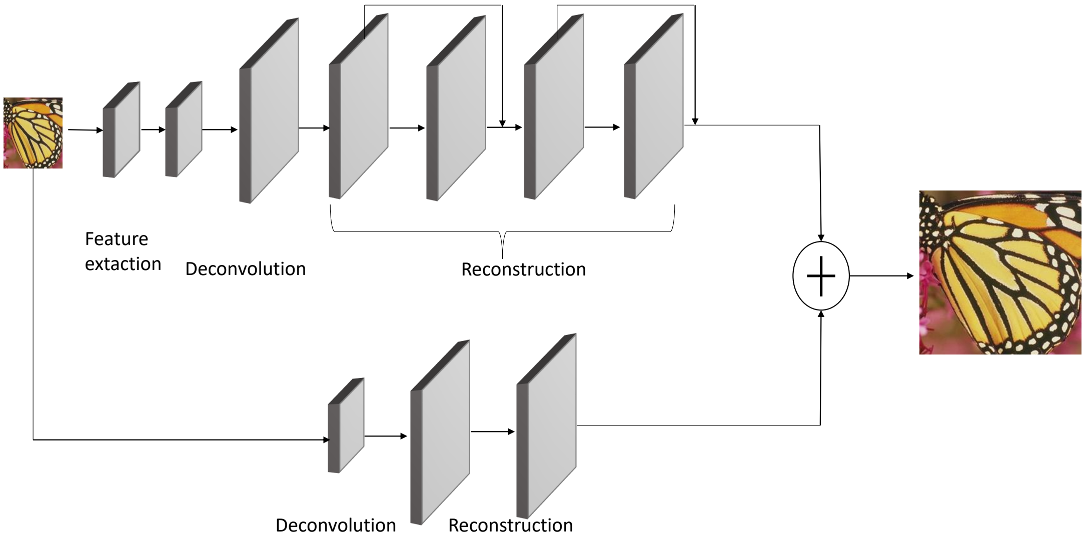


Figure 4.5: The work flow of the second propose network.

Before we come up with the idea of using two path subnet structure, the one deep path subnet structure is also tested for this iterative approach. Then network will lose the ability to learn high frequency and low frequency information separately. Theoretically, the performance will not be as good as the two path network. And the data in experiment part also agrees with this conclusion.

4.1.2 Residual Learning

Either paths of the subnet can be analyzed as one individual residual learning network. The only difference is just complexity of the network structure (number of the layers). The following parts provide detailed explanation for the residual learning network.

Blocks for Residual Learning. In SRCNN nets, the model is learned from directly mapping between interpolated image and ground truth high resolution image. A high resolution image can be treated as a composition of low frequency information (shared with low resolution image) and high frequency information (the details like edges). While the depth image usually has lots of piece-wise smooth regions (low frequency information), this makes SRCNN network difficult to properly learn the mapping for the high frequency information since the network will get saturated by learning the low frequency information. The super-resolution of a image is an interpolation process. The value of the interpolated pixel will be output of the weighted average value of surrounding pixels. When network gets saturated by learning the low frequency information, the edge pixel will not be accurately interpolated which will cause the blur on the edge regions. This is the main reason why we propose to add the residual block shown in Figure. 4.6 into the network so that the network will have more focus on learning the difference between interpolated images and ground truth high resolution images i.e. the details in high resolution image, which is helpful to reduce the blur on edge regions.

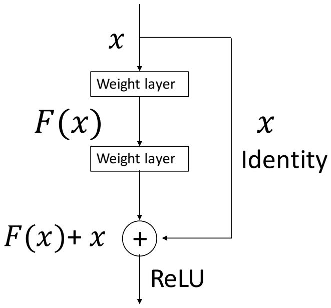
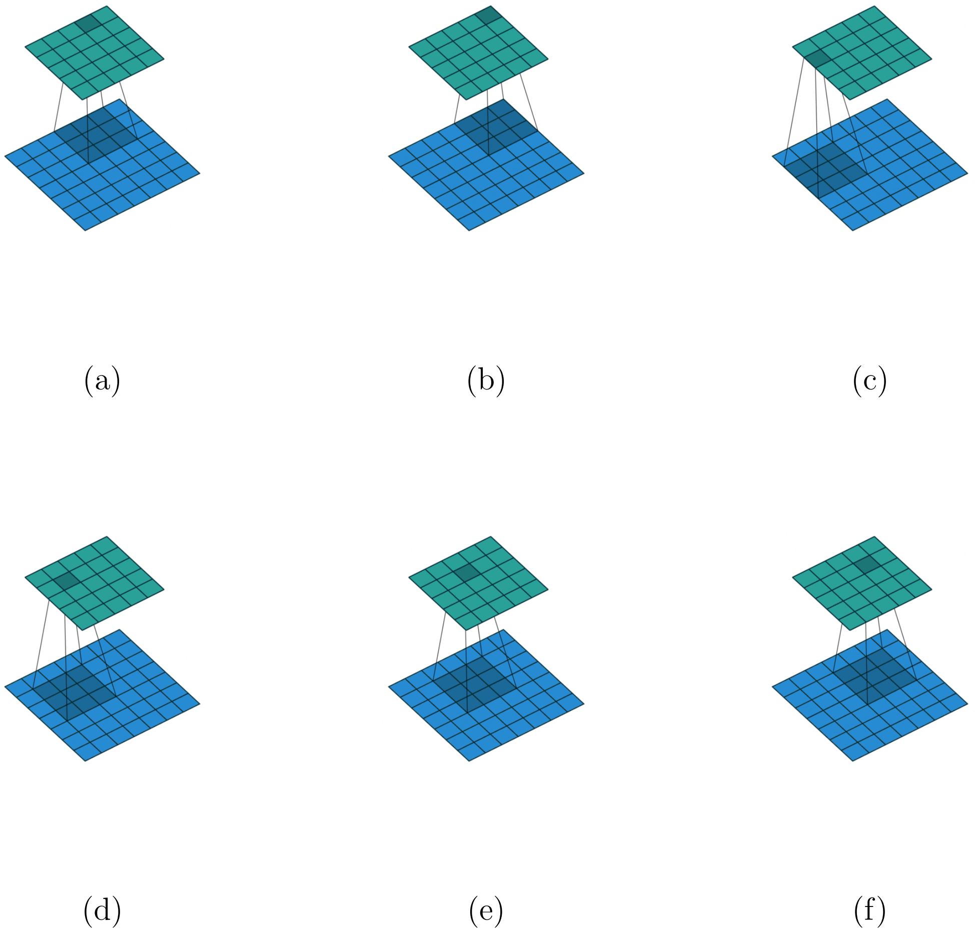


Figure 4.6: The building block for residual learning.

Deconvolution Layer. Most of the neural network based method [63, 68] will first apply the interpolation kernels (bicubic or bilinear) to the original low resolution images and then take the interpolated images as the input for training. However, the traditional interpolation method has the fixed upsampling kernels for the inputs and the parameters will not be used efficiently. The deconvolution can be regarded as reverse operation of convolution as shown in Figure. 4.7. By introducing the deconvolution layer to network, the network will have the ability to upsample and aggregate the previous feature maps by training a set of deconvolution filters through backpropagation. The upsampling kernel is no longer fixed and it gives more accuracy for the interpolation process.



(a) (b) (c)

(d) (e) (f)

Figure 4.7: Illustration of the Deconvolution

Activation Function. In the proposed network, we suggest that the Parametric Rectified Linear Unit (PReLU) should be used after each convolution layer instead of commonly used Rectified Linear Unit (ReLU). Different from ReLU, PReLU has the learnable coefficients on the negative part while the ReLU force these coefficients to be zero. Since we aim to make the network to focus on learning the difference between low resolution image and ground truth image, the PReLU could be more suitable for our network. In addition, by employing the PReLU as the activation function, the existence of dead features [69] cause by zero gradients in ReLU can be effectively avoided. So the network can fully utilize all the parameter.

Proposed Subnet Structure. The overall network structure is shown in Figure. 4.8. The description for each is presented in details as following:

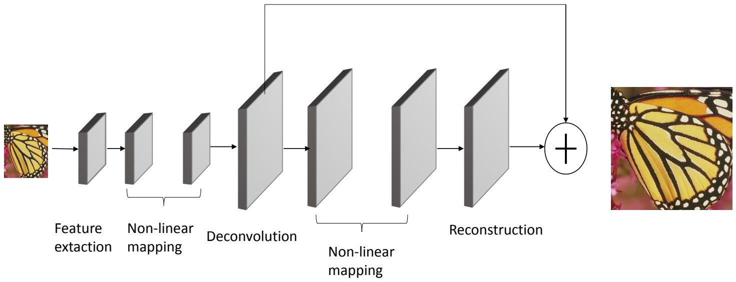


Figure 4.8: The structure of the proposed network.

1. *Feature Extraction.* This part is similar to the SRCNN network except for the size of the input images. The input images in SRCNN network are interpolated into the same size of the ground truth images while in our network the original low resolution is fed into the network directly. Then a high dimensional feature vector is produced by convolving with a set of filters. Since the input size becomes much smaller especially for lager upscale factors, we propose to use more filters with very small kernel size in order to capture more details on high frequency components of the images.

2. *Non-linear Mapping.* In the feature extraction process, the dimension of feature vector is extremely high due to the large number of filters. The computation complexity will be pretty high if we directly map the low resolution features to high resolution features. By learning from [70] where the 1×1 layers are applied to reduce the computation complexity, we suggest

to add one more layer that adopts much smaller number of filters compared to the number used in feature extraction process. In this configuration, the number of parameters are significantly reduced and the computational complexity is also maintained in a more proper level.

Non-linear mapping is still the most important part in the network, which have a large effect on the super resolution performance. There are two major influence factors for non-linear mapping layer, where they are the number of the filters in the layer and the number of used layers. In the work of [71], the author demonstrated that a good super resolution performance can be achieved by using larger filter size and employing multiple layers for non-linear mapping. In the design of our network, non-linear mapping consists of 5 layers with filter size of 5×5.

3. *Deconvolution Layer.* As explained in previous section, deconvolution layer consists of a set of trainable filters to upsample the low resolution image into high resolution image. The stride in this layer is the same as the upscale factor. The output image from deconvolution layer will have the resolution of its original size multiplied by the upscale factor.

4. *Residual Blocks.* Assume we have a training dataset x, y, our aim is to train the network to learn a mapping function f that predicts value $\hat{y} = f(x)$, where $\hat{y}$ is the predicted high resolution image. Then we apply SGD to minimize the mean squared error $\frac{1}{2}(y - f(x))^2$ over the training set. This is the common approach for CNN based methods and the vanishing gradient is the critical problem for this approach especially for deeper network. For depth images, the output of deconvolution layer will be similar to the ground truth image due to its piece-wise smooth characteristic. So we propose to employ residual learning [72] for the network.

   The residual image is defined as $r = y - x$. Since the image residual is the learning target, the loss function becomes $\frac{1}{2}(r - f(x))^2$, where the $f(x)$ is the high resolution image interpolated by the deconvolution layer. The loss layer in our network takes three inputs: image residual, interpolated image and ground truth image and the loss is the Euclidean distance between estimated image (sum of the network output and the deconvolution layer output) and ground truth image.

4.1.3 Experiments

Data generation. We generate the training dataset from MPI Sintel depth dataset [55] and RGBD images from Middlebury dataset [56, 73, 74]. The training images are divided into image patches with small spatial size and overlapping with neighbors. The training time can be reduced by this approach according to [63]. The MPI dataset is made of computer generated images with all the depth information available in the image. However, the Middlebury dataset is real depth image that is obtained from stereo matching algorithm. The depth information is not available in for all pixels in the dataset due to the occlusion phenomenon as shown in Figure. 4.9. The image patches that does not contain available depth information are excluded from the training set.

Data processing for Evaluation. The evaluation data consists of additional 10 images with different spatial resolution from Middlebury datasets. As mentioned before, the depth images from Middlebury datasets does not have all the depth information. Before we evaluate our trained model, the hole-filling process is applied to these 10 images.

Implementation details for two path subnet The network is also built on top of the $caffe$ CNN implementations [75]. The shallow path consists of 3 trainable layers followed by ReLU to introduce the non-linearity. The first one in shallow path is the deconvolution layer with large filter size (9×9) since we only want the network to learn more about the low frequency part information. The rest two both are 9×9 convolution layers which generate large depth image. The deep path contains three parts. The first part is made of two 3×3 convolution

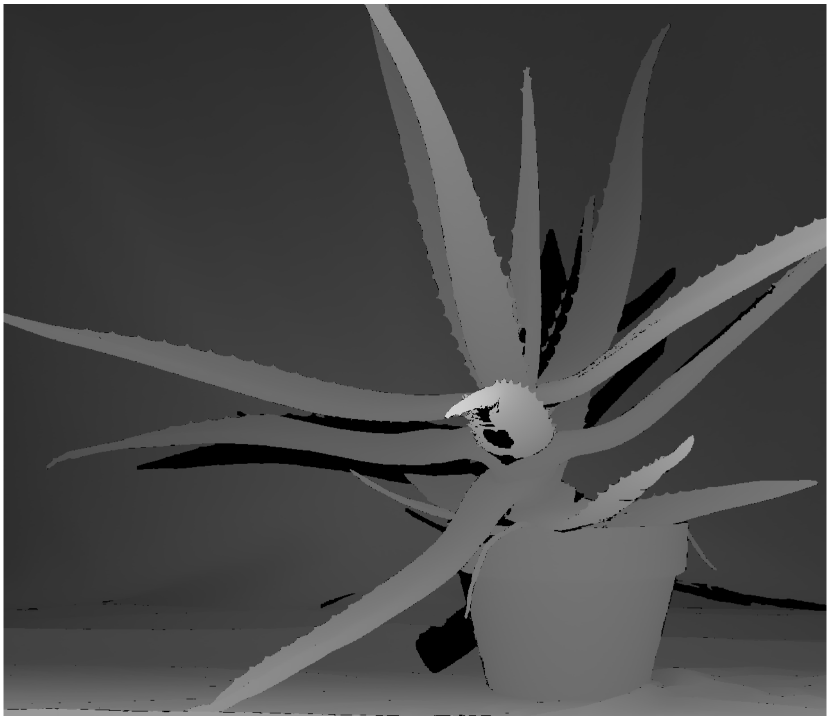


Figure 4.9: The depth information is not available in black regions.

layers for feature extractions. The middle part is made of the deconvolution layer followed by two convolution layers. This part performs upsampling task. The final part is for multi-scale reconstructions, that consists of 3 convolution layer with 3 × 3, 5 × 5 and 7 × 7 kernel size.

Implementation details for one path subnet The network is also built on top of the $caffe$ CNN implementations [75]. The number of filters in all convolution layers are set to 64. In order to capture the features in different levels, the filter size is set to 11 × 11, 7 × 7, 3 × 3 for sub-net 1, sub-net 2 and sub-net 3 respectively. The stepwise decrease learning policy is also adopted, which has 5 steps with learning rate multiplier $\gamma = 0.9$. Each sub-net will be trained for 1 million iterations.

Implementation details residual network Our network is built on top of the $caffe$ CNN implementations [75]. All the layers in the feature extraction stage have same settings for filter size and the number of the filters which are 3 × 3 and 64 respectively. In order to facilitate gradient flow, the reconstruction layer and the deconvolution layer forms a block, where the output of deconvolution layer can be added to the output of the block through a short cut with identity mapping. The 1×1 convolution layer serves as dimension reduction layer that maps the high dimension input feature map to the desired low dimension feature map. As for the non-linear mapping stage, we employ 4 convolution layers with filter size 5×5. We adopt stepwise decrease (4 steps with learning rate multiplier $\gamma = 0.9$) because this settings helps to reduce the fluctuations in convergence curve at the later part of network training. it takes one and half days to finish 2 million iterations training on GPU Titan X. The trained network needs two seconds to perform super-resolution for each frame.

Evaluations. We adopt root mean square error (RMSE) as our evaluation metric. The detailed quantitative evaluations are shown in Table 4.1 to Table 4.11. With different network structures ensemble in each sub-net, the ability of the network for making proper inference has been increased by intermediate level. This can be easily found from the RMSE evaluation on all test images with 4 different upscale factors (3 for one dataset). The subjective results are shown in Figure. 4.17.

4.2 Directional Network Approach

In this work, based on the fact that depth image contains more information about the geometric structures, we propose a directional approach with convolutional neural network to super resolve the low resolution images. Unlike the SRCNN which trains one network for perform the super-resolution on whole image, we build 3 networks to do the training for different directional components based on the structure of SRCNN and then fuse them into one large depth map.

Low resolution depth acquisition. In Figure. 4.10, we give labels to the different parts of the high resolution image based on directions i.e. label v for vertical parts, label h for horizontal parts and label x for diagonal parts. Then low resolution depth image labeled with o can be obtained by a mechanism which is equivalent to downsample a high resolution depth image as shown in Figure. 4.10. Based on the three main directions-vertical, horizontal and diagonal, the high resolution image can be decomposed into four downsampled sub-images. Lets name these sub-images as sub-image o, sub-image h, sub-image d and sub-image v respectively. Then the super resolution problem became how to find the sub-images on each direction and fuse them into high resolution image i.e. how to map the sub-image o into other sub-images.

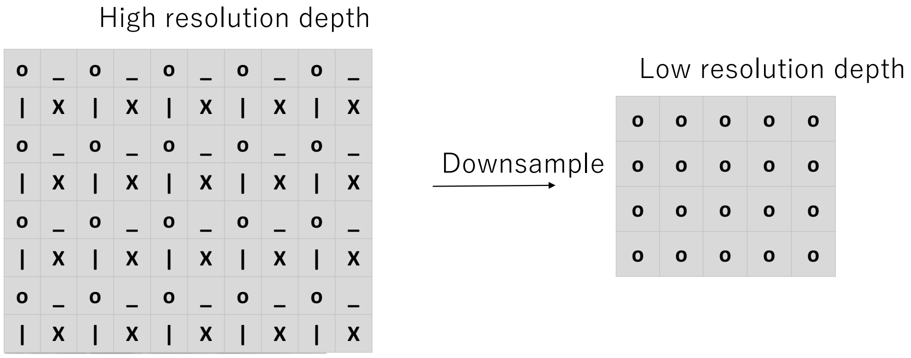


Figure 4.10: Low resolution image can be seen as downsampled version of high resolution image

CNN training for directional components. Inspired by deep learning methods, we construct three neural networks to learn the mapping relation between input low resolution image and the directional sub-image separately. We adopt the same network configuration from SRCNN [2]. As shown in Figure. 4.11, there are three main convolution layers in SRCNN settings:

1. *Patch extraction and representation.* This layer extracts image patches from the low resolution image X by convolving the image by a set of filters and output a high dimensional vector as the representation of the each patch. These vectors form a set of feature maps and the number of feature maps

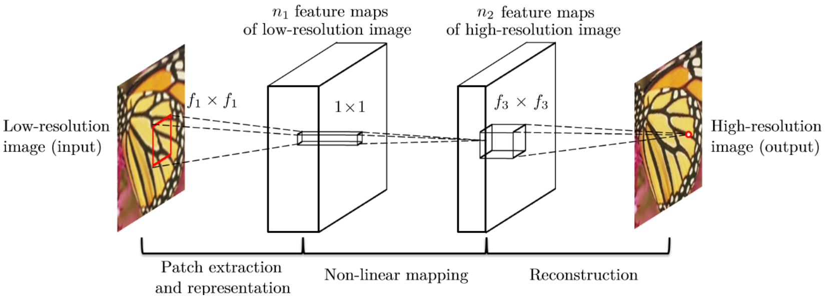

n_1 feature maps of low-resolution image
n_2 feature maps of high-resolution image
$f_1 \times f_1$
1×1
$f_3 \times f_3$
Low-resolution image (input)
High-resolution image (output)
Patch extraction and representation
Non-linear mapping
Reconstruction

Figure 4.11: The main structure of SRCNN. Figure is from [2]

equals to the dimensionality of the vectors. This layer can be formulated as an operation F_1:

$$F_1(X) = max(0, W_1 * X + B_1) \tag{4.1}$$

where W_1 represent the filters with size of $c \times f_1 \times f_1 \times n_1$. c is the number of channels of the input image and f_1 is the spatial size of the filters. The kernel size of each convolution is $c \times f_1 \times f_1$. n_1 is the number of filters and the number of feature maps of the output. B_1 represents the biases and it is n_1 dimensional vector. The ReLU is applied on the filter, which is equivalent to use the $max(0, X)$ operation.

2. *Non-linear mapping.* The previous layer extracts a n_1 dimensional feature vector for each image patch. In this layer, the n_1 dimensional vectors are mapped into n_2 dimensional ones by convolving with n_2 filters that has spatial size of 1×1. This layer can formulated as F_2:

$$F_2(X) = max(0, W_2 * F_1(X) + B_2) \tag{4.2}$$

W_2 has the size of $n_1 \times 1 \times 1 \times n_2$, and B_2 is n_2 dimensional biases. The mapped vectors are the representations of high resolution image patch and will be used for reconstruction process. Also, these mapped vectors comprise a new set of feature maps.

3. *Reconstruction.* In this layer, the mapped n_2 dimensional vector i.e. the representation of high resolution image patch generated from last layer will

go through an averaging process to finally reconstruct the high resolution image. This layer is formulated as:

$$F_3(X) = W_3 * F_2(x) + B_3 \tag{4.3}$$

W_3 has size of $n_2 \times f_3 \times f_3 \times c$, where f_3 is the spatial size of the filter in this layer. B_3 is c dimensional vector representing biases. There is no need to apply ReLU for the filter.

The mapping function F is learned from the estimation of the parameters Θ that comprise $\{W_1, W_2, W_3, B_1, B_2, B_3\}$. This is achieved by minimizing the difference between ground truth high resolution image Y and the reconstructed high resolution image $F(X; \Theta)$. For a set of training image examples-high resolution images Y_i and low resolution images X_i, the mean squared error (MSE) is introduced as loss function:

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n}(F(X_i; \theta) - Y_i)^2 \tag{4.4}$$

With SRCNN configurations, we set up three networks with same input i.e. sub-image o. The ground truth images are sub-image d, sub-image h and sub-image v because each of these sub-images contains directional details. We want the network to learn a mapping from sub-image o to other sub-images. Each network will have working flow like Figure. 4.12. Once we obtain the mapping function for each direction, the final high resolution image is reconstructed by fusing the sub-image o, estimated sub-image d, sub-image h and sub-image v as shown in Figure. 4.13.

4.3 Experiments

Dataset generation. Depth image is more about geometric structures and shapes. In order to properly train the network to learn the mapping functions, we use software BLENDER to generate a dataset which contains over 1000 depth

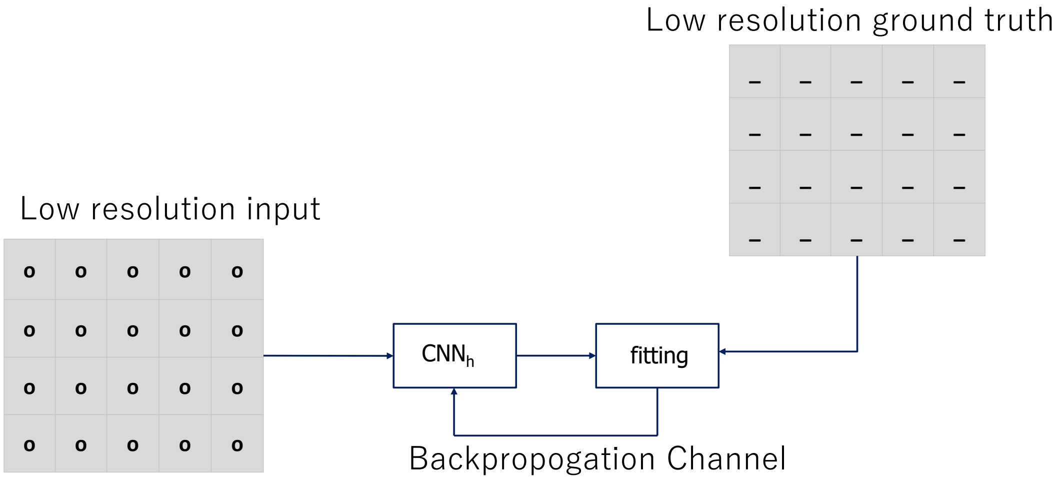


Figure 4.12: CNN work flow for the directional image details training

images. These depth images have different number of objects and the complexity also varies from one to another. Some example training images are demonstrated in Figure. 4.14.

Data Preprocessing. In order to get low resolution sub-image o and ground truth sub-images, we downsampled high resolution image without using conventional method like bicubic or bilinear because these methods usually have filtering process that changes pixel values before downsampling the high resolution. Since the pixel value of depth image keeps information of the real distance, the pixel values should not be altered.

Implementation Details. For the network setting details, we set $f_1 = 9$, $f_3 = 5$, $n_1 = 64$ and $n_2 = 32$ for evaluations. In the training stage, the ground truth images are generated with size of 32×32. And the convolution stride is set to be 3 in order to capture the fine details of the high resolution depth image. Due to the border effects in the training phase, there is no padding applied to all convolution layers. The filter weights of each layer are initialized by randomly taken from a Gaussian distribution with zero mean and standard deviation 0.001. The learning rate is set to be 10^{-4}. The training iteration is

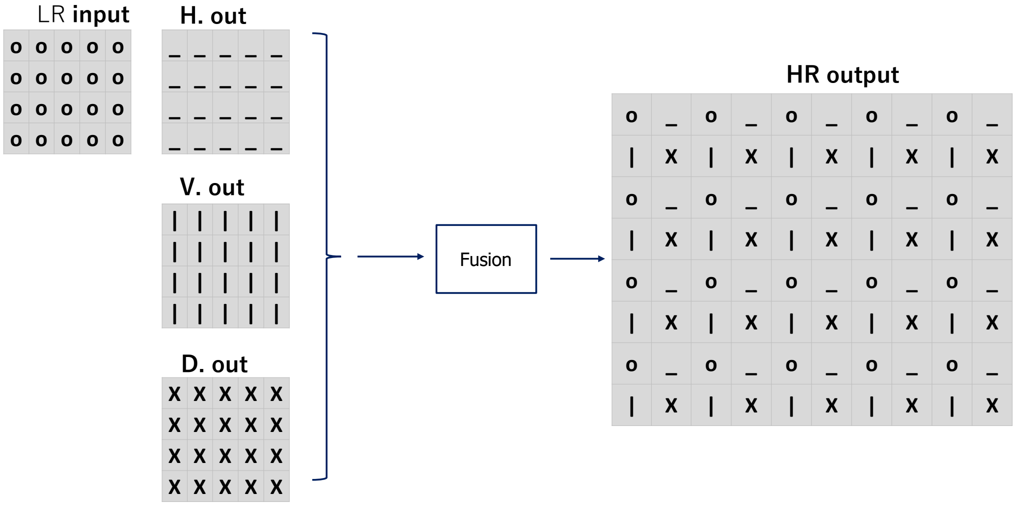


Figure 4.13: Fusion stage. H.out, V.out and D.out are the estimated version of sub-image h, sub-image v, sub-image d

set to be 10 millions which takes two days on GTX Titan X GPU. Since we network has multiple layers, the ReLU is chosen for the activation function to accelerate the convergence speed and help to avoid local optimal [53, 54]. Besides, the conventional activation functions are more likely causing vanishing gradient problem than ReLU in multilayer neural network configurations.

Quantitative Evaluation. We test our models on the computer generated depth images [55] and real depth image dataset Middlebury [56]. The evaluation on Middlebury dataset is only on the valid area because there are some region that does not contain depth information. As shown in Figure. 4.15, it is clear to see the proposed method has more superior performance compare to the bicubic method. The PSNR difference is calculated by subtracting PSNR of proposed method from PSNR of bicubic method. The difference above the 0 means the proposed method outperforms the bicubic method. However, the test on Middlebury dataset shown in Figure. 4.16 does not give good results. The proposed method fails on most images. This could be caused by the high complexity of structures of real depth image. The network is not able to learn these complex structures from computer

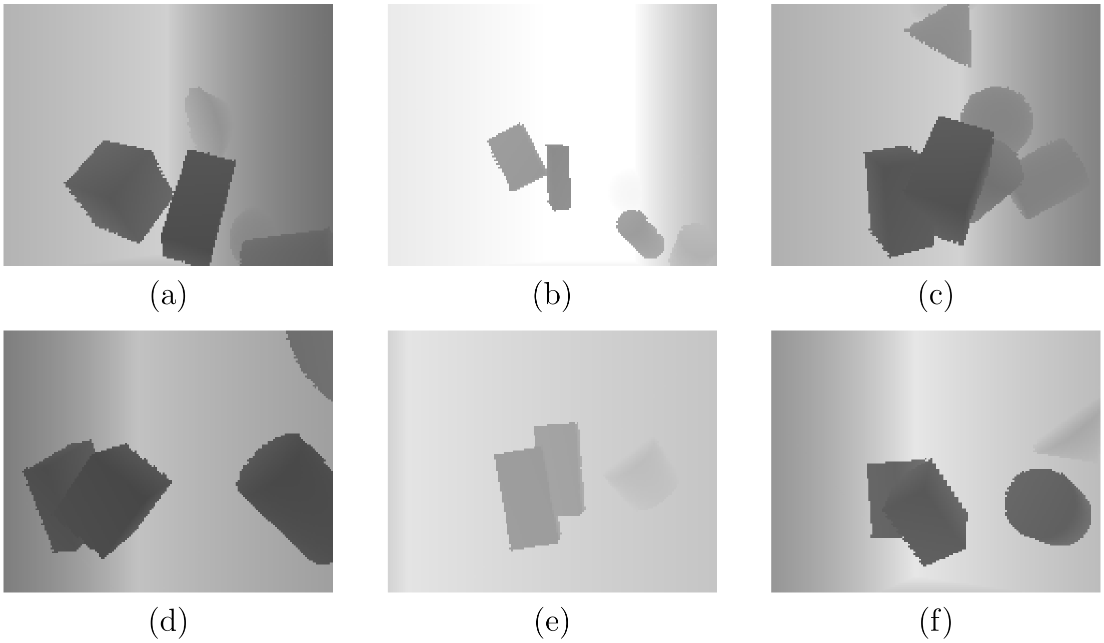

(a) (b) (c)
(d) (e) (f)

Figure 4.14: Some example training image generated from BLENDER

graphic images.

4.4 Summary

We have proposed two iterative network with different subnet configurations training approaches for depth image super resolution. The proposed network uses three sub-nets to learn the different level of features in a hierarchical way and the later sub-net takes the output of previous sub-net to continue the training process. Due to the similar subnet settings, the quantitative results show slight improvements for most test cases compare to the approach descried in the previous section. The proposed network II contains two subnets with different settings in terms of network depth, in order to learn low frequency part and high frequency part, which has more performance gain than proposed network I with only one learning path structure. However, the global optimal solution for all test images on different upscale factor remains unknown. Despite using the identity mapping through short cut, the overall training time for convergence is still more longer than non deep neural network methods.

We have also proposed an end-to-end network configurations for depth image super resolution. Unlike the SRCNN, we incorporate deconvolution layer into the network, which performs upsampling of the low resolution image directly within the network instead of using methods like bicubic to interpolate the image outside of the network. The residual block is also added into the network to learn the high frequency components of the high resolution image and to accelerate the deep network to converge. However, the sharpness at the boundaries is lost at some certain level. The edges are also getting blurred after the super resolution process. For the future work, constructing a network that can learn the high frequency components of the high resolution images could be a better approaches for the depth image super resolution task. And the corresponding color image may be added into network to guide the training process.

We have investigated a direction based network training for the depth image super resolution based on the understanding that depth image contains more information about structures and shapes instead of complex textures. The whole network consists of three separate networks that each one is responsible for one direction details. Our evaluation shows some improvement over Bicubic method over the computer generated dataset. However, the evaluation on real depth images shows that this approach also has a drawback. When we decompose the high resolution image into low resolution sub-images, the original structures in high resolution image may be damaged and the network will not be able to learn a proper mapping in these parts, especially for the real depth image which has much more complex structures or shapes than the computer generated images. And the image resolution and upscale factor will also affect the decomposition process.

	Art	Books	Moebius
MRFs	3.119	1.205	1.187
Bilateral	4.066	1.615	1.069
Park	2.833	1.088	1.064
Kiechle	1.246	0.652	0.640
Ferstl	3.032	1.290	1.129
Lu	1.133	**0.523**	**0.537**
Wang	1.670	0.668	0.641
Residual net	1.02	0.650	0.680
one_path_net	1.008	0.662	0.672
two_path_net	**0.952**	0.571	0.549

Table 4.1: Quantitative comparison in RMSE on dataset A for upscale 2 with proposed networks

	Art	Books	Moebius
MRFs	3.794	1.546	1.439
Bilateral	4.066	1.701	1.386
Park	3.498	1.530	1.349
Kiechle	2.007	0.918	0.887
Ferstl	3.785	1.603	1.458
Lu	2.017	0.935	0.913
Wang	2.525	1.098	0.979
Residual net	2.06	0.949	0.921
one_path_net	2.03	0.961	0.933
two_path_net	**1.953**	**0.839**	**0.814**

Table 4.2: Quantitative comparison in RMSE on dataset A for upscale 4 with proposed networks

	Art	Books	Moebius
MRFs	5.503	2.209	2.054
Bilateral	4.712	1.949	1.820
Park	4.165	1.994	1.804
Kiechle	**3.231**	**1.274**	1.272
Ferstl	4.787	1.992	1.914
Lu	3.829	1.726	1.579
Wang	3.957	1.646	1.459
Reisdual net	3.676	1.687	1.320
one_path_net	3.655	1.702	1.308
two_path_net	3.457	1.639	**1.228**

Table 4.3: Quantitative comparison in RMSE on dataset A for upscale 8 with proposed networks

	Art	Books	Moebius
MRFs	8.657	3.400	3.078
Bilateral	8.268	3.325	2.494
Park	6.262	2.760	2.377
Guided	7.876	3.186	2.851
Kiechle	5.744	1.927	2.128
Ferstl	7.102	2.941	2.630
Lu	7.648	3.549	3.118
Wang	6.226	**2.428**	2.202
Residual net	5.080	2.700	2.100
one_path_net	4.982	2.752	**2.076**
two_path_net	**4.461**	2.707	**1.859**

Table 4.4: Quantitative comparison in RMSE on dataset A for upscale 16 with proposed networks

	Dolls	Laundry	Rendeer
Park	0.963	1.552	1.834
Aodha	1.801	1.735	1.953
CLMF0	0.990	1.689	1.955
CLMF1	0.972	1.689	1.948
Ferstl	1.118	1.989	2.407
Kiechle	0.696	0.746	0.920
AP	1.147	1.715	1.803
Wang	0.670	1.039	**0.556**
Residual net	0.683	**0.766**	0.734
one_path_net	0.663	0.801	0.723
two_path_net	**0.621**	0.781	0.774

Table 4.5: Quantitative comparison in RMSE on dataset B for upscale 2 with proposed networks

	Dolls	Laundry	Rendeer
Park	1.301	2.132	2.407
Aodha	1.977	2.969	3.178
CLMF0	1.271	2.312	2.690
CLMF1	1.267	2.512	2.699
Ferstl	1.355	2.511	2.712
Kiechle	**0.921**	1.212	1.559
AP	1.350	2.255	2.431
Wang	0.989	1.630	1.914
Residual net	1.003	**1.200**	1.489
one_path_net	0.975	1.286	1.397
two_path_net	1.112	1.364	**1.265**

Table 4.6: Quantitative comparison in RMSE on dataset B for upscale 4 with proposed networks

	Dolls	Laundry	Rendeer
Park	1.745	2.770	2.987
CLMF0	1.878	3.084	3.417
CLMF1	1.707	2.892	3.331
Ferstl	1.859	3.757	3.789
Kiechle	**1.295**	2.077	2.583
AP	1.646	2.848	2.949
Wang	1.445	2.466	2.878
Residual net	1.650	2.05	2.314
one_path_net	1.644	2.102	2.286
two_path_net	1.741	**2.033**	**2.149**

Table 4.7: Quantitative comparison in RMSE on dataset B for upscale 8 with proposed networks

	Dolls	Laundry	Rendeer
Park	2.412	4.158	4.294
CLMF0	2.291	4.312	4.674
CLMF1	2.232	4.302	4.774
Ferstl	3.574	6.407	7.271
Kiechle	**1.736**	3.621	4.644
AP	2.323	4.656	5.249
Wang	2.107	3.834	4.526
Residual net	2.501	3.312	3.253
one_path_net	2.439	3.368	**3.207**
two_path_net	2.261	**3.294**	3.317

Table 4.8: Quantitative comparison in RMSE on dataset B for upscale 16 with proposed networks

	Tskuba	Venus	Teddy	Cones
Park	6.61	1.27	3.73	4.0
Li	8.29	2.29	2.78	3.24
Ferstl	7.2	2.151	2.71	3.5
Ferstl	5.254	1.108	1.694	2.185
kiechle	3.48	0.8	1.28	1.7
kwon	2.31	0.53	**0.83**	**0.92**
Aodha	8.993	2.175	3.233	4.262
Timofte	9.135	2.099	3.253	4.257
wang	3.979	0.828	1.368	1.856
Residual net	2.016	0.46	1.15	1.54
one_path_net	1.998	**0.446**	1.131	1.499
two_path_net	**1.791**	0.519	1.011	1.464

Table 4.9: Quantitative comparison in RMSE on dataset C for upscale 2 with proposed networks

	Tskuba	Venus	Teddy	Cones
Park	9.75	1.8	4.89	5.64
Li	11.9	3.55	4.92	6.34
Ferstl	10.3	2.52	3.3	4.45
Ferstl	7.352	1.742	2.595	4.17
kiechle	5.95	1.17	2.94	3.498
kwon	5.56	1.14	1.80	**2.13**
Aodha	12.39	2.597	4.030	5.740
Timofte	12.09	2.331	3.718	5.490
Lu	10.29	1.734	2.723	3.585
wang	6.281	1.191	2.026	3.078
Residual net	3.02	0.73	1.78	2.628
one_path_net	2.976	0.68	**1.74**	2.632
two_path_net	**2.821**	**0.592**	1.81	2.532

Table 4.10: Quantitative comparison in RMSE on dataset C for upscale 4 with proposed networks

	Tskuba	Venus	Teddy	Cones
Park	15.1	2.99	7.15	7.73
Li	15.84	5.76	7.24	8.9
Ferstl	17.2	4.04	5.39	7.14
kiechle	10.9	1.76	2.76	5.11
kwon	5.67	**1.68**	2.19	**2.37**
Lu	13.77	2.134	3.468	5.345
wang	9.589	1.786	3.015	4.865
Residual net	5.765	1.788	2.09	2.748
one_path_net	5. 682	1.763	2.03	2.744
two_path_net	**5. 572**	1.701	**1.94**	2.469

Table 4.11: Quantitative comparison in RMSE on dataset C for upscale 8 with proposed networks

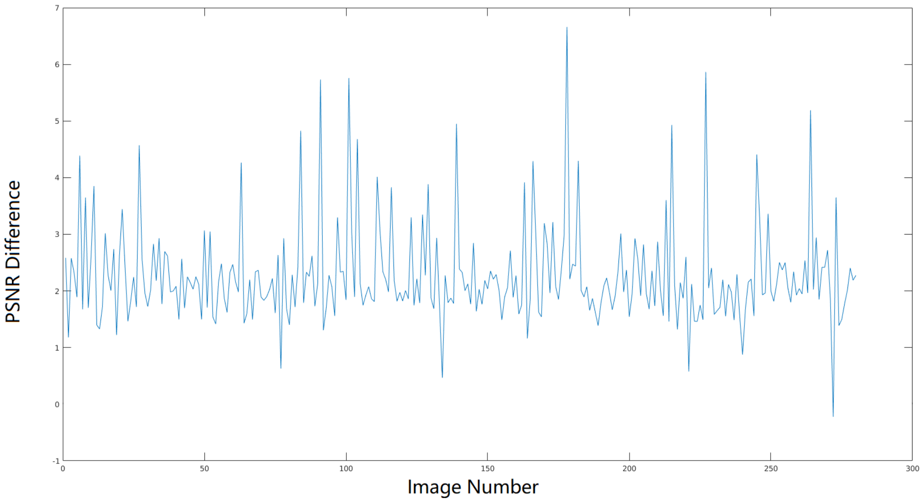


Figure 4.15: PSNR difference (PSNR (proposed) - PSNR (bicubic)) between proposed method and bicubic method

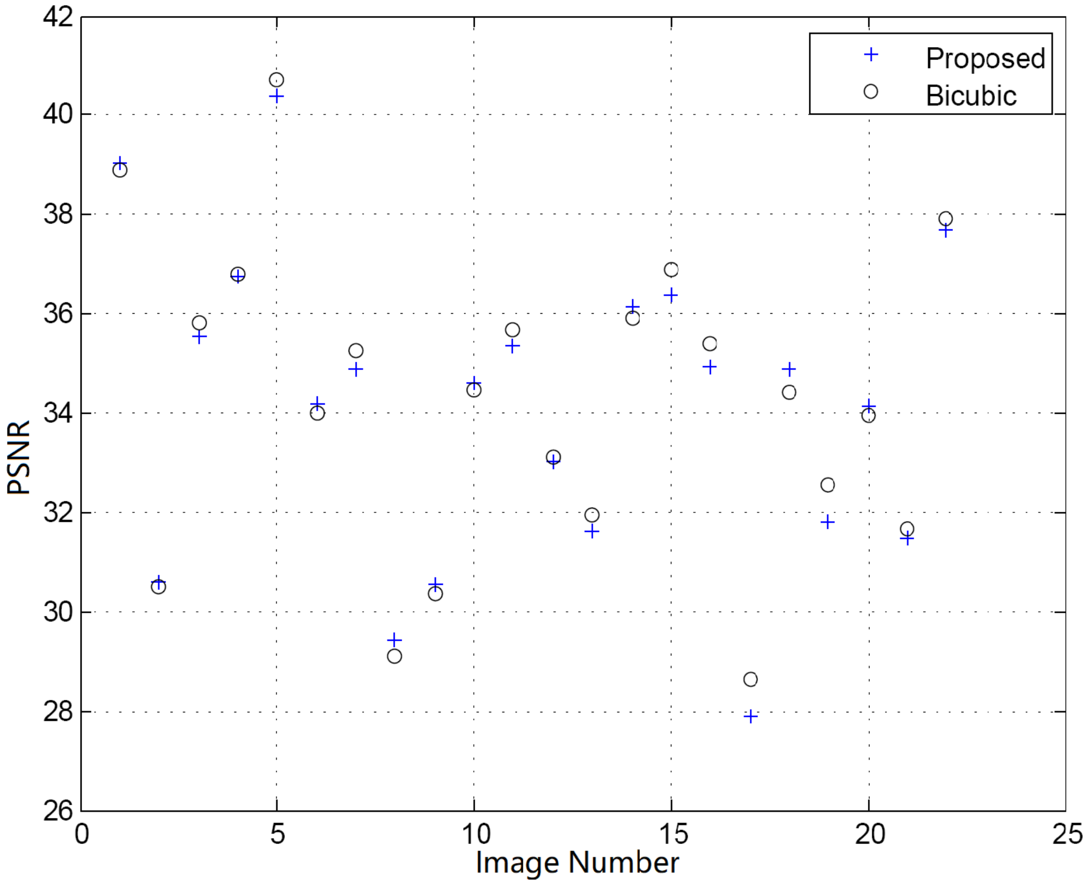


Figure 4.16: PNSR evaluation on Middelbury dataset

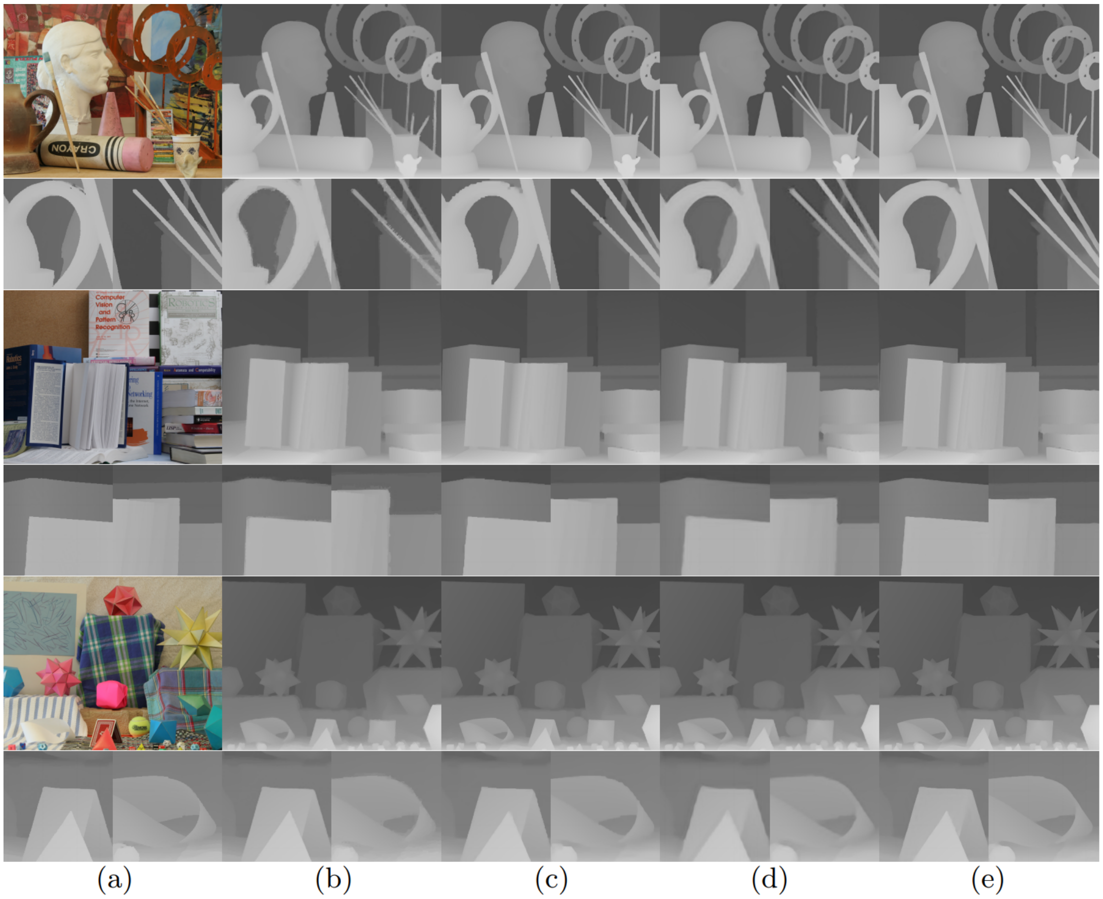


Figure 4.17: Usample results for dataset A. (a) color image and the ground truth,Upsample resluts from (b) [3], (c) [4], (d) [5]and (e) proposed

Chapter 5

Conclusion

5.1 Contributions

In this thesis, we have developed an algorithm that combines color and depth information for the task of background modeling. During the process, we found that the performance of developed algorithm largely depend on the accuracy of the depth image with proper resolution, which is usually not easily satisfied. We made investigations in image super resolution techniques that help us to obtain the accurate high resolution image from low resolution images. Main contribution of this thesis is to investigate the convolutional neural network based methods for super resolving the low resolution depth image.

Specifically, the contributions of this thesis are listed in details as following:

1. A new algorithm for background extraction using Gaussian Mixture Models (GMM) combined with depth map is presented, where the per-pixel mixture model and single Gaussian model are used to model the recent observation in color and depth space respectively. We also incorporated the color-depth consistency check mechanism into the algorithm to improve the accuracy of extracted background. Our results show much better performance than prior state of the art methodology for the background extraction task even when used for challenging scenes.

2. Based on the affect that the depth image is more about geometric structure

and shapes, the low resolution image can be obtained by a mechanism which is equivalent to decompose the high resolution image. We developed a neural network that is used to learn the mapping from low resolution image to the image decomposed from high resolution image, which contains directional details of ground truth. After we obtained estimated images on all three directions (vertical, horizontal and diagonal), all 4 images are then fused into final high resolution image.

3. To address the problem of the artifact on sharp boundary, we proposed to add residual blocks into the network for enabling the network to learn the mapping of the difference between estimated high resolution image and the ground truth high resolution image. To further improve the performance, the deconvolution layer is incorporated to make the interpolation kernel trainable and optimized through backpropagation process. And by using the ensemble structure for separate learning first then combining together, the iterative network has achieved much better performance.

5.2 Future Work

1. Deep learning has been proved to be a very powerful tool for many computer vision task. In the future work for background modeling, it could be good to build our model on the top of deep learning framework with combine information of color and depth images.

2. The generative adversarial network (GAN) [84, 85] has demonstrated a great performance on the color image super resolution. So it could be good to construct a network on the top of GAN framework. Also it could be good to continue to work on the iterative approach for depth image super resolution.

Bibliography

[1] Andrej Karpathy. Connecting images and natural language. In *PhD thesis*, pages 36–36, 2016.

[2] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199. Springer, 2014.

[3] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 993–1000, 2013.

[4] Martin Kiechle, Simon Hawe, and Martin Kleinsteuber. A joint intensity and depth co-sparse analysis model for depth map super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1545–1552, 2013.

[5] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas Huang. Deep networks for image super-resolution with sparse prior. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 370–378, 2015.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[7] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfinder: Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, 1997.

[8] Benny Ping Lai Lo and Sergio Velastin. Automatic congestion detection system for underground platforms. In *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, pages 158–161. IEEE, 2001.

[9] Pakorn KaewTraKulPong and Richard Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-Based Surveillance Systems*, pages 135–144. Springer, 2002.

[10] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2. IEEE, 1999.

[11] Dar-Shyang Lee. Effective gaussian mixture learning for video background subtraction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):827–832, 2005.

[12] Jens Rittscher, Jien Kato, Sébastien Joga, and Andrew Blake. A probabilistic background model for tracking. In *European Conference on Computer Vision*, pages 336–350. Springer, 2000.

[13] Bjoern Stenger, Visvanathan Ramesh, Nikos Paragios, Frans Coetzee, and Joachim M Buhmann. Topology free hidden markov models: Application to background modeling. In *null*, page 294. IEEE, 2001.

[14] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In *European conference on computer vision*, pages 751–767. Springer, 2000.

[15] Marc Braham and Marc Van Droogenbroeck. Deep background subtraction with scene-specific convolutional neural networks. In *IEEE International Conference on Systems, Signals and Image Processing*, pages 1–4. IEEE, 2016.

[16] Mohammadreza Babaee, Duc Tung Dinh, and Gerhard Rigoll. A deep convolutional neural network for video sequence background subtraction. *Pattern Recognition*, 76:635–649, 2018.

[17] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin. Subsense: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24(1):359–373, 2015.

[18] Sriram Varadarajan, Paul Miller, and Huiyu Zhou. Region-based mixture of gaussians modelling for foreground detection in dynamic scenes. *Pattern Recognition*, 48(11):3488–3503, 2015.

[19] G Gordon, Trevor Darrell, Michael Harville, and John Woodfill. Background estimation and removal based on range and color. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2. IEEE, 1999.

[20] Michael Harville, Gaile Gordon, and John Woodfill. Adaptive video background modeling using color and depth. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 90–93. IEEE, 2001.

[21] Chao Yao, Tammam Tillo, Yao Zhao, Jimin Xiao, Huihui Bai, and Chunyu Lin. Depth map driven hole filling algorithm exploiting temporal consistent information. *Broadcasting, IEEE Transactions on*, Accepted.

[22] M Tanimoto, T Fujii, and K Suzuki. View synthesis algorithm in view syn-

thesis reference software 2.0 (vsrs2.0)). In *ISO/IEC JTCI/SC29/WG11M*, volume 16090, 2009.

[23] Massimo Camplani, Lucia Maddalena, Gabriel Moyá Alcover, Alfredo Petrosino, and Luis Salgado. A benchmarking framework for background subtraction in rgbd videos. In *International Conference on Image Analysis and Processing*, pages 219–229. Springer, 2017.

[24] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304. Ieee, 2011.

[25] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[26] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer, 2012.

[27] William T Freeman, Thouis R Jones, and Egon C Pasztor. Example-based super-resolution. *IEEE Computer graphics and Applications*, 22(2):56–65, 2002.

[28] Yanjie Li, Tianfan Xue, Lifeng Sun, and Jianzhuang Liu. Joint example-based depth map super-resolution. In *Multimedia and Expo, 2012 IEEE International Conference on*, pages 152–157. IEEE, 2012.

[29] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[30] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010.

[31] Shenlong Wang, Lei Zhang, Yan Liang, and Quan Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2216–2223. IEEE, 2012.

[32] Kwang In Kim and Younghee Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE transactions on pattern analysis and machine intelligence*, 32(6):1127–1133, 2010.

[33] Min-Chun Yang and Yu-Chiang Frank Wang. A self-learning approach to single image super-resolution. *IEEE Transactions on Multimedia*, 15(3):498–508, 2013.

[34] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 349–356. IEEE, 2009.

[35] Sebastian Schuon, Christian Theobalt, James Davis, and Sebastian Thrun. Lidarboost: Depth superresolution for tof 3d shape scanning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 343–350. IEEE, 2009.

[36] A Rajagopalan, Arnav Bhavsar, Frank Wallhoff, and Gerhard Rigoll. Resolution enhancement of pmd range maps. *Pattern Recognition*, pages 304–313, 2008.

[37] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[38] Qingxiong Yang, Ruigang Yang, James Davis, and David Nistér. Spatial-depth super resolution for range images. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[39] Shujie Liu, PoLin Lai, Dong Tian, Cristina Gomila, and Chang Wen Chen. Joint trilateral filtering for depth map compression. In *Visual Communications and Image Processing 2010*, pages 77440F–77440F. International Society for Optics and Photonics, 2010.

[40] Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. In *ACM Transactions on Graphics (ToG)*, volume 26, page 96. ACM, 2007.

[41] Jaesik Park, Hyeongwoo Kim, Yu-Wing Tai, Michael S Brown, and Inso Kweon. High quality depth map upsampling for 3d-tof cameras. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1623–1630. IEEE, 2011.

[42] Ouk Choi and Seung-Won Jung. A consensus-driven approach for structure and texture aware depth map upsampling. *IEEE Transactions on Image Processing*, 23(8):3321–3335, 2014.

[43] Oisin Mac Aodha, Neill Campbell, Arun Nair, and Gabriel Brostow. Patch based synthesis for single depth image super-resolution. *Computer Vision–ECCV 2012*, pages 71–84, 2012.

[44] James Diebel and Sebastian Thrun. An application of markov random fields to range sensing. In *NIPS*, volume 5, pages 291–298, 2005.

[45] Michael Hornácek, Christoph Rhemann, Margrit Gelautz, and Carsten Rother. Depth super resolution by rigid body self-similarity in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1123–1130, 2013.

[46] Jun Xie, Cheng-Chuan Chou, Rogerio Feris, and Ming-Ting Sun. Single depth image super resolution and denoising via coupled dictionary learning with local constraints and shock filtering. In *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, pages 1–6. IEEE, 2014.

[47] Jun Xie, Rogerio Schmidt Feris, Shiaw-Shian Yu, and Ming-Ting Sun. Joint super resolution and denoising from a single depth image. *IEEE Transactions on Multimedia*, 17(9):1525–1537, 2015.

[48] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[49] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[51] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[52] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 448–456. JMLR. org, 2015.

[53] Kazuyuki Hara, Daisuke Saito, and Hayaru Shouno. Analysis of function of rectified linear unit used in deep learning. In *Proceedings of International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2015.

[54] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *Proceedings of International conference on acoustics, speech and signal processing*, pages 8609–8613. IEEE, 2013.

[55] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, October 2012.

[56] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition*, pages 31–42. Springer, 2014.

[57] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[58] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[59] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[60] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[61] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European conference on computer vision*, pages 346–361. Springer, 2014.

[63] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[64] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[65] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[66] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. *Efficient backprop*. Springer, 2012.

[67] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[68] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1538–1546, 2015.

[69] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[70] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *Proceedings of the IEEE Conference on Learning Representations*, 2014.

[71] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision*, pages 391–407. Springer, 2016.

[72] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016.

[73] Daniel Scharstein, Richard Szeliski, and Ramin Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Stereo and Multi-Baseline Vision, 2001.(SMBV 2001). Proceedings. IEEE Workshop on*, pages 131–140. IEEE, 2001.

[74] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[75] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convo-

lutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[76] Jiajun Lu and David Forsyth. Sparse depth super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2245–2253, 2015.

[77] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1397–1409, 2013.

[78] Jiangbo Lu, Keyang Shi, Dongbo Min, Liang Lin, and Minh N Do. Cross-based local multipoint filtering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 430–437. IEEE, 2012.

[79] Jingyu Yang, Xinchen Ye, Kun Li, Chunping Hou, and Yao Wang. Color-guided depth recovery from rgb-d data using an adaptive autoregressive model. *IEEE Transactions on Image Processing*, 23(8):3443–3458, 2014.

[80] David Ferstl, Matthias Ruther, and Horst Bischof. Variational depth super-resolution using example-based edge representations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 513–521, 2015.

[81] Radu Timofte, Vincent De Smet, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1920–1927, 2013.

[82] Xibin Song, Yuchao Dai, and Xueying Qin. Deep depth super-resolution: Learning depth super-resolution using deep convolutional neural network. In *Asian conference on computer vision*, pages 360–376. Springer, 2016.

[83] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In

Proceedings of IEEE International Conference on Computer Vision, pages 2169–2178. IEEE, 2006.

[84] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[85] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.