Depth-Map-Assisted Texture and Depth Map Super-Resolution

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

Zhi Jin

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

Supervisor team: Prof. Tammam Tillo (Xi'an Jiaotong-Liverpool University)
Dr. Waleed Al-Nuaimy (University of Liverpool)

Dec. 31, 2015

Abstract

With the development of video technology, high definition video and 3D video applications are becoming increasingly accessible to customers. The interactive and vivid 3D video experience of realistic scenes relies greatly on the amount and quality of the texture and depth map data. However, due to the limitations of video capturing hardware and transmission bandwidth, transmitted video has to be compressed which degrades, in general, the received video quality. This means that it is hard to meet the users' requirements of high definition and visual experience; it also limits development of future applications. Therefore, image/video super-resolution techniques have been proposed to address this issue.

Image super-resolution aims to reconstruct a high resolution image from single or multiple low resolution images captured of the same scene under different conditions. Based on the image type that needs to be super-resolved, image super-resolution includes texture and depth image super-resolutions. If classified based on the implementation methods, there are three main categories: interpolation-based, reconstruction-based and learning-based super-resolution algorithms. This thesis focuses on exploiting depth data in interpolation-based super-resolution algorithms for texture video and depth maps. Two novel texture and one depth super-resolution algorithms are proposed as the main contributions of this thesis.

The first texture super-resolution algorithm is carried out in the Mixed Resolution (MR) multiview video system where at least one of the views is captured at Low Resolution (LR), while the others are captured at Full Resolution (FR). In order to reduce visual uncomfortableness and adapt MR video format for free-viewpoint television, the low resolution views are super-resolved to the target full resolution by the proposed virtual view assisted super resolution algorithm. The inter-view similarity is used to determine whether to fill the missing pixels in the super-resolved frame by virtual view pixels or by spatial interpolated pixels. The decision mechanism is steered by the tex-

ture characteristics of the neighbors of each missing pixel. Thus, the proposed method can recover the details in regions with edges while maintaining good quality at smooth areas by properly exploiting the high quality virtual view pixels and the directional correlation of pixels. The second texture super-resolution algorithm is based on the Multiview Video plus Depth (MVD) system, which consists of textures and the associated per-pixel depth data. In order to further reduce the transmitted data and the quality degradation of received video, a systematical framework to downsample the original MVD data and later on to super-resolved the LR views is proposed. At the encoder side, the rows of the two adjacent views are downsampled following an interlacing and complementary fashion, whereas, at the decoder side, the discarded pixels are recovered by fusing the virtual view pixels with the directional interpolated pixels from the complementary downsampled views. Consequently, with the assistance of virtual views, the proposed approach can effectively achieve these two goals. From previous two works, we can observe that depth data has big potential to be used in 3D video enhancement. However, due to the low spatial resolution of Time-of-Flight (ToF) depth camera generated depth images, their applications have been limited. Hence, in the last contribution of this thesis, a planar-surface-based depth map super-resolution approach is presented, which interpolates depth images by exploiting the equation of each detected planar surface. Both quantitative and qualitative experimental results demonstrate the effectiveness and robustness of the proposed approach over benchmark methods.

Contents

List of Tables

List of Figures

List of Abbreviations

CCD	Charge Coupled Device
CFA	Color Filter Array
CG	Computer Graphic
CMOS	Complementary Metal Oxide Semiconductor
CS	Compressive Sensing
DDFU	Directional Data Fusion Upsampling
DIBR	Depth Image-Based Rendering
FN	False Negative
FP	False Positive
FR	Full Resolution
FRUC	Frame Rate Up-conversion
F-SFM	Factorization Structure From Motion
FTV	Free-viewpoint Television
GOP	Group of Pictures
HDTV	High Definition TV
HR	High Resolution
HVS	Human Vision System

IBP	Iterative Back Projection
JBU	Joint Bilateral Upsampling
LCC	Local Coordinate Coding
LLE	Locally Linear Embedding
LMMSE	Linear Minimum Mean Squares-error Estimation
LR	Low Resolution
LSI	Linear Space Invariant
LSV	Linear Space Variant
MAD	Mean Absolute Deviation
MAP	Maximum A Posteriori
MDL	Minimum Description Length
ML	Maximum Likelihood
MR	Mixed Resolution
MRF	Markov Random Field
MSE	Mean Square Error
MSFE	Mean Square Fitting Error
MVD	Multiview Video plus Depth
OGS	Overgrowing Surface

PAR	Piecewise Autoregressive
PCA	Principal Components Analysis
POCS	Projection Onto Convex Sets
PSD	Planar Surface Detection
PSF	Point Spread Function
PSNR	Peak Signal to Noise Ratio
QP	Quantization Parameters
RANSAC	RANdom SAmples Consensus
REoD	Relative Error of Depth
RGB	Red Green Blue
SFM	Structure From Motion
SR	Super-Resolution
S-SFM	Sequential Structure From Motion
SSIM	Structural Similarity
SSSIM	Structural Self-Similarity
SVD	Singular Value Decomposition
SVR	Support Vector Regression
ToF	Time-of-Flight
TN	True Negative
TP	True Positive

Write your injuries in dust,
your benefits in marble.

Benjamin Franklin, Statesman

Acknowledgement

The past four years of my Ph.D. research was an unforgettable and valuable experience which was full of happiness, laughter, anticipation, satisfaction, and also sadness, anxiety, disappointment, frustration. It consisted of many hardworking days and nights, and tears of success and failure. For sure, all of these would not happen without the help of numerous people, all of who I would like to remember with appreciation during my whole life.

First and most importantly I would like to express my heartfelt gratitude to my Ph.D. supervisor, Professor Tammam Tillo for his unwavering and helpful supports not only of my research but also my life, for his kindness, encouragement, spontaneous guidance, patience, comfort, understanding, tolerance, foresight from the first day I became his student, six years ago and then through every stage in this long journey. He spent many hours editing and improving the readability and presentation of my thesis as well as my academic papers. He tried his best to provide me with both economic and academic support to attend international conferences. To express all of my gratitude to him, I need many pages. In short, because of him, such kind of supervisor, I feel I am a lucky Ph.D. student. Besides my primary supervisor, I would like to thank my co-supervisor Dr. Waleed Al-Nuaimy, in University of Liverpool, UK.

My sincere thanks also go to Dr. Jimin Xiao, the senior Ph.D, now lecturer in our group, who not only discussed research with me, but also helped me modify my papers and this thesis; Professor Byeungwoo Jeon, in Sungkyunkwan University, Korea and Professor Eckehard Steinbach, in Technical University of Munich, Germany, for giving me chances studying in their research groups and learning from other outstanding Ph.D. students; Professor Yao Zhao, who supports me as the leader of our solid research partners in Beijing Jiaotong University including Yao Chao; Professor Mark Leach, in our department who helped me modify this thesis during his Christmas holiday and his kindness and carefulness touched me a lot. I am grateful to my research colleagues

in the Multimedia Technology lab and Department of Electrical and Electronics Engineering, especially Fei Cheng who provides lots of hardware supports. I thank all of my teachers, students, friends, relatives and all others who helped and inspired me directly or indirectly, and all who have wished me well in my study, research and other purposes. Special thanks to my teacher V.K. Liau who taught me how to become a valuable and useful person in the world and the truth of our life.

My eternal gratitude goes to my parents for their unconditional love, and who have always tried to make my life comfortable, always wish me for success and happiness from the moment I came to the world. They suffer from their missing in order to give me the freedom to explore the lovely world. They work hard in order to make me never worry about money. They try their best to live happily and healthily in order to make me do not worry about them. All of these are the light and power for me to walk further in my life.

Last but not least, I would like to thank Xing Luo. Without him, I could finish this thesis faster. However, without this Ph.D. journey, I will not have him in my life.

Chapter 1

Introduction

1.1 Motivation

The development of video technologies make high definition video and 3D video applications increasingly accessible to consumers through products, such as, High-Definition (HD) TVs, computer monitors, HD cameras, smart phones, and many other handheld devices. However, the demands for High Resolution(HR) and 3D video put pressure on the acquisition, storage and transmission processes, especially for bandwidth limited applications [10]. Hence, the popularity of HR or 3D video in the multimedia market still faces many challenges.

Perceived image quality relies greatly on the capture and delivery process. For image quality assessment, one essential factor is spatial resolution (or pixel density) in one image, which is affected by the camera sensor (e.g., Couple Charge Device (CCD)) [2] [3]. High resolution images can be obtained by increasing the total number of pixels on a CCD chip either via reducing pixel size or increasing chip size. Whereas, the effectiveness of the first approach is limited by shot noise which severely degrades the image quality. The second approach increases chip size, on one hand, it will lead to an increase in capacitance which will decrease the charge transfer rate. On the other hand, it will cause an additional increase in cost due to the high precision optics and image sensors required [3]. In spite of the limitations of the camera sensor, in reality, due to the capture conditions and digital camera techniques, the captured images usually cannot reflect all of the information in a scene. The image acquisition process is shown in Fig.1.1, where the atmosphere turbulence, the motion transformation caused distortion, the downsampling introduced distortion and the hardware caused additive noise can degrade the captured image quality [11]. Since the transmission of images,

especially HR images, requires much higher bitrate than text, before transmitting, high compression rates are required, resulting in annoying compression artifacts, such as, block artifacts, blurred details and ringing artifacts around edges [12]. Therefore, a new approach toward increasing spatial resolution, so as to increase the quality, is needed. One effective solution to solve these kinds of problem is the super-resolution technique, which could provide a cost-effective solution to increase image resolution.

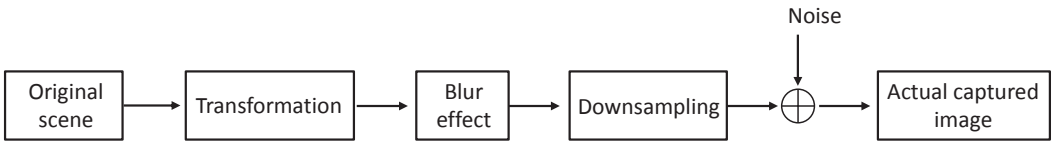


Figure 1.1: The image acquisition process.

Since Super-Resolution(SR) techniques are targeted at increasing the spatial resolution of low-cost hardware obtained LR images and limited bandwidth obtained LR images by using software, they have played an important role in many applications. For example, in surveillance video applications, SR can be used to obtain higher quality video sequences from several low resolution cameras [13] and to recognize car license plates and faces robustly and efficiently even under bad capture conditions [14] [15]. In the remote sensing and astronomy fields, due to the size and weight limitation of satellite cameras, power supply and transmission bandwidth, the obtained images are usually in low resolution. Therefore, SR techniques become essential for earth environment observation [16]. Medical imaging has become an important aid tool for medical diagnosis. Using SR to enhance the quality of medical images, the condition and position of a lesion can be well determined, so that the accuracy of diagnoses can be improved [17]. In consumer electronics, entertainment and digital communication applications, multimedia content occupies a dominant share. SR techniques can offer an improved viewing experience for customers by enlarging the resolution of perceived images/video and removing the visual artifacts caused by video compression [18].

More than 30 years ago, researchers began to work on extracting information from multiple digital images to enhance the spatial resolution of LR images [19]. In general, the SR algorithms for images can be mainly classified into 3 categories: reconstruction-based algorithms [20], learning-based algorithms [7][21] and interpolation-based algorithms [22][23]. Techniques in the first category are based on the assumption that high frequency details are available in a LR image as aliased frequencies, and these ap-

proaches can be realized in both the frequency and spatial domains [24] [25]. However, this kind of method highly relies on the choice of the regularization parameters and the number of LR images, which are not easy to obtain in reality [26]. In contrast, learning-based SR approaches assume that it is possible to predict the missing high frequency details in a single LR image by a group of LR and FR image pairs [27]. Since image information hides underlying models, it can be modeled as a mathematical function. Unfortunately, their performance largely depends on the choice of training samples, so unsuitable training samples can produce artifacts in the recovered High Resolution (HR) image [26]. Free from the suffering from the dependency problem on training data and having well time performance, especially for real time systems, interpolation-based SR algorithms are widely adopted. The interpolation-based SR algorithms are implemented based on the fact that the missing HR pixels can be estimated by using the information from neighboring LR pixels. However, the main drawback of these methods is their inability to fully exploit the scene content during the interpolation process, and consequently they are prone to blur high frequency details (edges).

1.2 Objectives

The main goal of the thesis is to propose specific and sophisticated interpolation-based 3D SR solutions for different purposes, so that to fill the gaps with respect to current 2D SR algorithms. With one more cue in depth, the SR methods designed for 3D video can achieve better results than directly applying 2D SR algorithms on 3D video. To fulfil this objective, the research work was carried out by introducing depth information into the SR process. However, the depth camera generated depth images have lower resolution than the corresponding textures which makes them cannot be used directly. Hence, the depth camera generated depth images need to be super-resolved to the same resolution as the textures. To achieve the final goal, this thesis addresses the following objectives:

- Providing a review of classic and the state-of-the-art 2D SR methods as the general background.
- Developing a 3D SR algorithm for 3D-MR video.
- Developing a 3D SR algorithm for MVD video.

- Developing a robust and accurate planar surface detection algorithm on the depth camera generated depth images.
- Developing an efficient depth image SR algorithm based on the planar surface detection results.

1.3 Overview of this thesis

1.3.1 Contribution of This Thesis

This thesis provides an investigation of efficient texture and depth image SR algorithms for different applications by using depth information. The main contributions of the thesis are:

- **Depth-map-assisted texture super-resolution for multiview mixed-resolution video system**

  A new virtual-view-assisted SR and enhancement algorithm is proposed, where the exploitation of the virtual view information and the interpolated frames offers two benefits. Firstly, the high frequency information contained in the FR views can be properly utilized to super-resolve LR views; secondly, the inter-view redundancy is used to enhance the original LR pixels in the super-resolved views and to compensate for the luminance difference between views. The experimental results show that the proposed algorithm achieves superior performance with respect to interpolation-based algorithms. This work was published in [28], and is presented in Chapter 3.

- **Depth-map-assisted texture super-resolution for multiview video plus depth system**

  In Chapter 3, the FR views generated virtual views and traditional interpolated views are used in conjunction to super-resolve the LR view in a multiview mixed-resolution video system. While, in this framework, in addition to super-resolving one LR view, the two FR views are downsampled before encoding and super-resolved after decoding by exploiting inter-view redundancy via virtual views.

  In the proposed downsampling approach, the rows of two adjacent texture views are discarded following an interlacing and complementary pattern, before compression. The aim of this downsampling approach is to systematically facilitate the super-resolution task at the decoder end, where the LR views will be super-resolved by fusing the virtual view pixels with directional interpolated pixels with the aid of pattern direction of the discarded pixels. This approach has two benefits. Firstly, the high frequency information contained in the counterpart LR view can be properly utilized to super-resolve the other LR view through the generated

virtual views. Secondly, since the virtual view quality depends on many factors, including the DIBR technique and depth map quality, it generally has low quality in areas where the corresponding depth data suffers from discontinuities. On the other hand, directional interpolation approaches can work well. Hence, by taking advantage of these two kinds of strategy, the discarded pixels can be recovered efficiently. The experimental results have shown that the proposed algorithm achieves superior performance with respect to the filter-based interpolation algorithms and state-of-the-art algorithms. This work could be regarded as an extension of the work presented in Chapter 3.

- **Super-resolution of depth map by exploiting planar surfaces**

  In the previous chapters, depth data has shown the big potential to be used to super-resolve LR views and the techniques of generating depth map become more mature and accurate. However, the ToF depth camera generated depth maps still suffer from low resolution. Therefore, in Chapter 5, this thesis focuses on depth map SR by exploiting planar surfaces on a single depth map. In this way, the super-resolved depth maps can expand the application domains of texture SR algorithm.

  Depth maps, different from common texture images due to their large homogeneous areas, are delimitated by sharp edges at the discontinuities between objects. After projecting 3D objects, they can be represented by several planar surfaces with different shapes in a 2D image, each surface will have linearly changing depth values in the corresponding depth map and the boundaries of surfaces represent the discontinuities of the depth values. If the equation of each surface can be obtained, the SR of the LR depth map can be obtained by inserting pixels based on this equation. Therefore, the whole depth map can be classified into three categories: planar surfaces, non-planar surfaces, and edges. In [29], the SR of depth map relied on the local planar hypothesis and the candidates for potential HR depth values were obtained by either linear interpolation along horizontal and vertical directions or the estimated local planar surface equations. However, since the surface equation was evaluated locally, it may be biased by noise affecting local pixels which later on will magnify the estimated error of the generated HR depth map. Therefore, to address the above problem, we propose the use

of global analytical equations of the detected surfaces in the scene. For each of these three categories a proper up-sampling approach is proposed to exploit its intrinsic properties. The related work was published in [30] [31], and is presented in Chapter 5.

1.3.2 Organization of This Thesis

This thesis is organized as follows: Chapter 2 provides general background on image acquisition and assessment. The advantages and challenges of current SR algorithms are reviewed and discussed. Then, the related concepts and techniques are also introduced.

Chapters 3 and 4 give the details of the two proposed texture SR algorithms, respectively, while Chapter 5 changes the focus to the depth map SR algorithm. Chapter 6 summarizes the whole thesis and discusses possible future research directions in the area of efficient super-resolution techniques.

Chapter 2

Background

2.1 Introduction of Texture Image

2.1.1 Texture Image Acquisition

The history of the first generation of the photographic camera dates back to the fifth century B.C. Its working principle, camera obscura, was discovered by the Chinese philosopher Mo Di [32]. However, it was not until 1826 when the first permanent photographic image from a camera obscura was captured by Joseph Nicphore Nipce on a bitumen-coated metal plate. From that time the floodgates of using photographs to record human actions were opened. Subsequently many recording techniques have been developed, themselves diverse in nature and in many cases easier to use than before. In 1888, Kodak invented a new photographic material, photographic film, and produced the first photographic film camera. The invention of photographic film greatly improved the usage of the camera. In 1975, the first digital camera was invented; formally heralding the human progression into the digital imaging era.

The working principle of the digital camera is completely different from that of the conventional camera. A conventional film camera captures the image based on chemical reactions that take place in an emulsion covering the surface of the film when it is exposed to light. The emulsion is commonly composed of silver salt, whose particles are sensitive to the quantum effect of light. The spatial variations of light intensity are captured in the salt and appear after developing the film. Instead of using a chemical approach, the digital camera relies on a physical approach, using electronic sensors to perceive spatial variations in light intensity. With the aid of some image processing algorithms, the sensor data is converted into color images and stored in a visible digital format (Fig.2.1) [2].

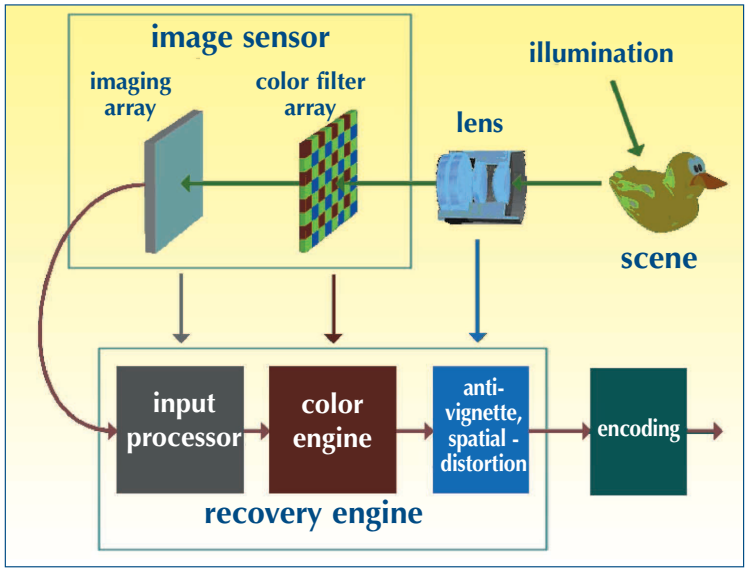
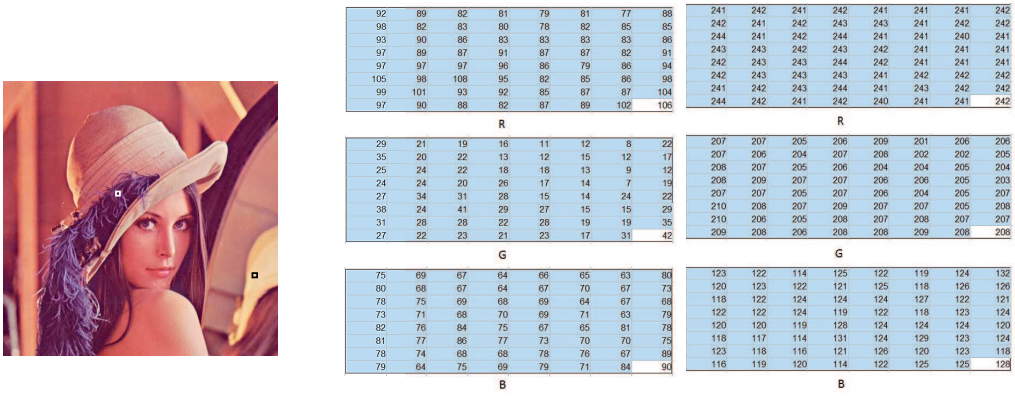


Figure 2.1: The working principle of digital camera [2].

Currently, there are two major types of imaging sensor, namely CCD (Charge Coupled Device) and CMOS (Complementary Metal Oxide Semiconductor). The CCD consists of tiny light-sensitive diodes, called photosites, which can convert photons into electrons [33]. In this way, light can be represented by electrical charge. The brighter the light on a single photosite, the greater the electrical charge that will accumulate at that photosite. The CMOS imaging chip, as a kind of active pixel sensor, is made by semiconductors. Both types of image sensor can convert light into electrons at the photosites and then an analog-to-digital converter will turn each pixel's value into a digital value. CCD sensors have the ability to accumulate the charges and extract them from the chip without distortion, therefore, CCD sensors have high fidelity and light sensitivity and also have been widely used in professional, medical, and scientific applications where high-quality image data is required. On the other hand, CMOS sensors with a more consolidated manufacturing process have a lower price and quality than that of CCD sensors. Hence, for applications with less demand on quality, such as consumer digital cameras, the CMOS sensor is popular [34].

On these sensors, each sensitive element can be called a "pixel" and the pixel is the basic unit in a digital image. In general, the number of pixels in each dimension of a rectangular image represents the spatial size of an image in that dimension (known as resolution) and the per-unit quality of captured images is determined by the number of pixels on the sensors. The more pixels the camera has, the more detail that can

be recorded in the captured image. For example, nowadays the popular HDTV (High Definition TV) which has a resolution of 1920×1080 means that each frame has 1920 and 1080 pixels along the horizontal and vertical directions, respectively. In texture images, each pixel consists of three color components, Red, Green, and Blue (RGB). The perceived color differences are caused by mixing these three components with various intensities. An example is shown in Fig.2.2, two crops from the color image Lenna have different values of the RGB components.

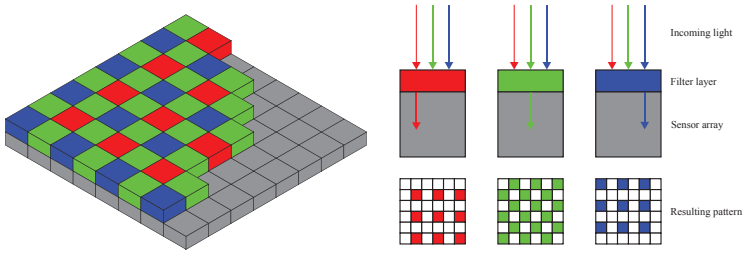


(a) (b) (c)

Figure 2.2: Testing image Lenna (a) and RGB values on the two highlighted parts. (b) the RGB values on white square region; (c) the RGB values on black square region.

In the common configuration each sensor element responds to one colour component only, hence there is a Color Filter Array (CFA) in the digital camera, located on the top of the sensor array, to decompose incoming light into the three primary colors. One common CFA arrangement pattern is called the Bayer pattern [35]. As shown in Fig.2.3, the number of green mosaics is twice the number of the red and blue mosaics. This is due to the fact that the Human Vision System (HVS) is more sensitive to the color green than the other two. At each pixel position, there is only one color intensity, hence, the values for the other two missing colors are interpolated from the adjacent corresponding colors. This process is known as "demosaicing". Each pixel is to be an RGB triplet. After some post-processing procedures, the captured images are stored in a digital storage device.

In the process of capturing a digital image, there are some factors that affect the quality of the obtained images. For example, optical distortions affect spatial resolution, limited shutter speed causes the motion blur effects and the camera sensors result in inevitable noise. Thus, in fact, the recorded image usually suffers from blur, noise, and



(a) (b)

Figure 2.3: The Bayer color filter mosaic. (a) The Bayer arrangement of color filters on the pixel array of an image sensor; (b) cross-section of sensor.

aliasing effects (Fig.2.4).

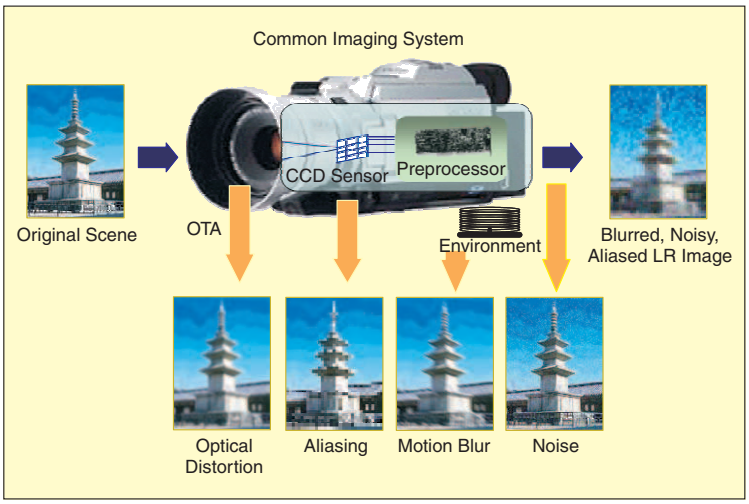


Figure 2.4: Low resolution images acquisition [3].

2.1.2 Texture Quality Assessment

In image transmission systems, between being captured and being received, the image passes through many steps and the techniques adopted in these steps may result in an aggregate degradation of the visual quality of the final received image. The SR approach is one of the image post-processing techniques, aimed at improving image quality. In order to quantify the received image quality and the efficiency of SR approaches, some quality assessment methods are required.

Assessment of the quality of an image can be carried out objectively or subjectively, each of which has its own strengths and associated applications. The objective image quality assessment is usually focused on two aspects: fidelity and intelligibility [36]. Fidelity is used to measure "how close/similar a received image is to the origi-

nal image". It mainly focuses on the detailed differences in the two images and the higher the fidelity is, the better the image quality is. While, intelligibility is used to indicate "how well the image can deliver the original information to its viewers in spite of the distortion affecting the image". It focuses on the global quality of the received images. Many researchers have worked on these two factors and tried to develop quantitative measures that can accurately describe the perceived image quality. To date, the objective quality assessment approaches can be classified into: ground truth approach which uses the available original image, and those are called the full-reference approach; the no-reference approach which uses no original image, and the reduced-reference approach which uses only part of the original image (e.g. the region of interest). For the full-reference approach, since the original image is known, it is more straightforward to measure the quality. The most widely used full-reference quality assessment metrics are the Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM). These assessment metrics are popularly used in many applications mainly due to their computational simplicity, clear physical meanings and the mathematical convenience in the context of optimization. However, in many practical applications, the reference image is not available, and the no-reference or blind quality assessment approach is desirable.

For an original image, $\mathbf{I_{GT}}$ with size $W \times H$, the quality of a received image $\mathbf{I}$ measured by MSE is:

$$MSE = \frac{\sum_{i=1}^{W}\sum_{j=1}^{H}(\mathbf{I}_{GT}(i,j) - \mathbf{I}(i,j))^2}{W \times H} \tag{2.1}$$

The quality of a received image $\mathbf{I}$ measured by PSNR can be calculated through MSE:

$$PSNR = 10\lg(\frac{B^2}{MSE}) \tag{2.2}$$

where B represents the quantization level. In general, 8-bit images have pixel values within the range $[0, 255]$, $B = 255$. Since either MSE or PSNR globally measures an image similarity to the original one by averaging the intensity differences between pixels, they do not provide assessment results consistent with the perceived visual quality. Thus, Wang *et al.* [37] proposed the SSIM matrix to assess image quality based on the image structure information at the pixel level. The quality of received image $\mathbf{I}$ measured by SSIM is:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2.3}$$

For each pixel (x, y) within the $N \times N$ windows W_x and W_y, its mean value and variance for window W_x are μ_x and σ_x^2, respectively and for window W_y, are μ_y and σ_y^2, respectively. σ_{xy} is the covariance of pixels within W_x and W_y and c_1 and c_2 are two constant values. The final SSIM result is obtained by averaging all values.

Since the received image/video is finally viewed by a human, then subjective evaluation is more "proper" to quantify visual image quality than objective evaluation. In practice, however, subjective assessment requires a human participant which makes it inconvenient, time-consuming and expensive. Hence, some advanced objective assessment approaches are needed.

2.2 Introduction of Depth Map

Depth map is a grey image with each pixel value between 0 and 255. Larger the pixel value is, closer this point to the depth camera. Hence, depth map can represent the relative distance between objects in a scene and the capturing depth camera. Due to this feature, it has been widely utilized in 3D applications to provide an immersive 3D and free-viewpoint experience for the viewer.

In this section, firstly, various depth map acquisition methods will be described, highlighting their corresponding weaknesses. The quality assessment methods and some useful depth map applications will also be introduced.

2.2.1 Depth Map Acquisition

Depth maps can be generated using software or hardware driven techniques, such as stereo or multiview matching-based methods, Structure-from-Motion (SfM), 3D laser scanner and depth-camera-based methods [38].

Since depth maps have many applications in computer vision and visual perception field, various software-based algorithms which compute correspondences from stereo or multiple views, have been proposed for the acquisition of depth map [39]. Stereo matching and SfM are two common depth map estimation methods in the computer vision field and since they do not rely on active illumination, they are regarded as passive methods.

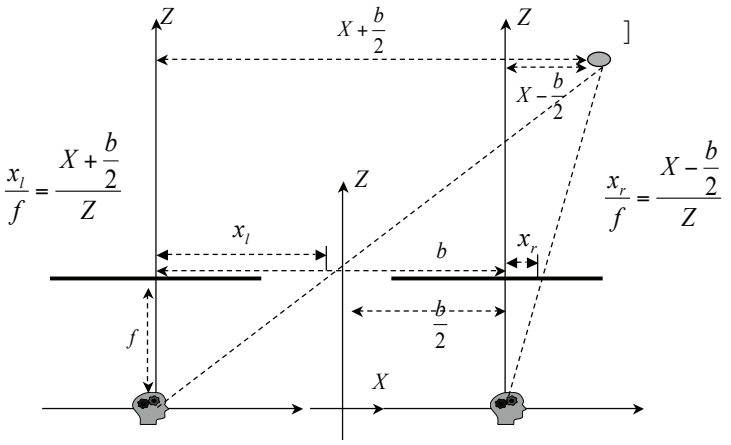


Figure 2.5: Working principle of stereo matching method.

Matching-based methods require at least two color images of the same scene captured from slightly different viewpoints. The common features and areas in these two captured images are then analyzed to extract depth information. Referring to Fig.2.5, a point P in the 3D scene is viewed from two viewpoints with the same focus length f and the line distance between the two focus center points is known as the view baseline, b. The distance of the projection point of P in the left and right view planes to the corresponding focus center on this plane are x_l and x_r, respectively. Assuming the perpendicular depth of point P to the two view planes is Z, based on the parallel line theorem, we get

$$\begin{cases} \frac{x_l}{f} = \frac{X+\frac{b}{2}}{Z} & ; \\ \frac{x_r}{f} = \frac{X-\frac{b}{2}}{Z} & ; \end{cases} \tag{2.4}$$

Then

$$Z = \frac{bf}{x_l - x_r} \tag{2.5}$$

where $x_l - x_r$ is the position difference between corresponding points in two images, called "disparity" and it is inversely proportional to the scene depth Z. Therefore, in theory, knowing the disparity of two counterpart points in the two captured images, the depth of corresponding 3D point can be obtained.

There are two approaches in stereo matching: the area-based method and the feature-based method. Area-based methods are, in general, used to obtain a dense depth map by finding the highest correlation between left and right image areas [40]. Feature-based methods are mainly used to obtain sparse depth maps. The working principle of stereo matching is straightforward. However, it is well-known that the

matching-based approaches may fail when no matching is found between some areas in the two views or textless areas. For example, some areas are occluded in one view and not in the other view. Although a considerable amount of effort has been exerted to cope with such problems, most methods are computationally expensive or iterative which make matching-based methods impractical.

SfM aims to recover both the structure of the 3D scene and the camera locations where the images where captured based on the analysis of motion of the feature points in a set of input images. Start from feature extraction, SfM matches the extracted feature points in different input images, and reconstructs the 3D structure. Hence, it can be used to estimate the depth information of the scene. Sequential methods (S-SfM) [41] and Factorization methods (F-SfM) [42] are two commonly used approaches in SfM. S-SfM works with each view sequentially, in contrast, F-SfM computes the structure of the scene and motion/calibration of the camera using all points in all views simultaneously.

Recently, with the development of sensor and lens technology, many hardware-based depth map acquisition approaches have been proposed. The 3D laser scanner is a mature 3D capturing technique and there are in general two different types of devices, Time-of-Flight (ToF) laser range finders and triangulation 3D laser scanners as shown in Fig.2.6.

(a) (b)

Figure 2.6: (a) ToF laser range finder scanner; (b) NextEngine 3D Scanner (triangulation 3D laser scanners).

Equipped with an emitter, ToF 3D laser scanners are active scanners, which probe the subject distance using laser light. The core technique is measuring the round-trip

time of a pulse of light, which is carried out by a time-of-flight laser range finder. A laser is used to emit a pulse of light and the amount of time before the reflected light seen by a detector is measured. The laser range finder can provide long-distance measurements in 1D and is capable of scanning large structures like buildings or geographic features. However, due to the difficulty of measuring the incredibly short time involved in light traveling short distances, the accuracy of the distance measurement is low.

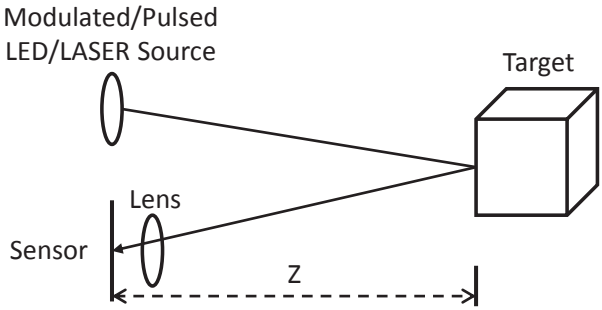


Figure 2.7: ToF detection system.

There are two kinds of ToF cameras, one is based on measuring the time of flight and the other is based on measuring the phase shift of a modulated optical signal, which can be related to measuring the time [43]. A typical ToF measuring setup consists of a modulated or pulsed light source such as a LED or a laser, a lens for focusing the light onto the sensor and an array of pixels, each capable of detecting the incoming light [44]. A sketch of the corresponding structure is shown in Fig.2.7. The measurement principle is straight-forward for the time-of-flight method. A highly accurate stopwatch begins to count the time synchronized with the light pulse emission. When the reflected light from the object surface arrives at the sensor, the count is stopped. Assuming the round-trip time of one surface point is t_i, the distance of this surface point Z_i can be obtained by the equation:

$$Z_i = \frac{c}{2} \cdot t_i \tag{2.6}$$

where c represents the speed of light propagating through the air. For phase-shift-based ToF cameras (referring to Fig.2.8), the distance is measured by the differences of the phase modulation envelop between the emitted and received light. If one pixel's phase shift is noted by $\Delta\phi_i$, its distance to the capturing camera is

$$Z_i = \frac{\lambda_m}{2} \frac{\Delta\phi_i}{2\pi} \tag{2.7}$$

where λ_m is the wave length of the modulation signal. For these two methods, af-

ter calculating the distance of each surface point, a 2D per-pixel depth map can be generated.

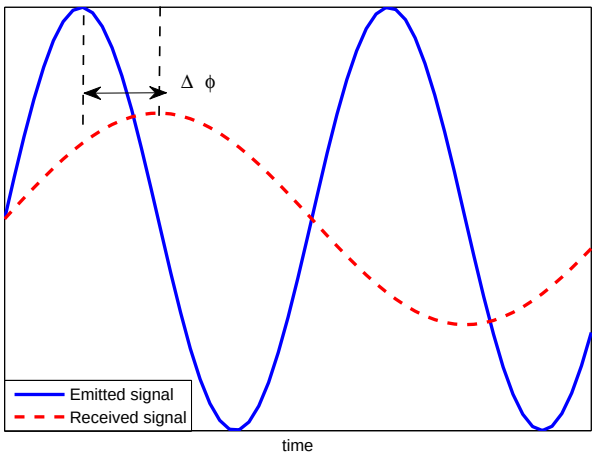


Figure 2.8: Received sinusoidally modulated input signal, sampled with 2 sampling points per modulation period T.

Similar to ToF scanners, triangulation 3D laser scanners are also active scanners. They emit a laser on the subject surface and utilize a camera to look for the location of the laser dot. Due to the varying object distances, the locations of the laser dot on the camera sensor are different. In this way, the object distance can be obtained. This technique is called triangulation because the laser dot, the camera and the laser emitter form a triangle. Compared to ToF 3D laser scanners, triangulation scanners have a limited scanning range, but the accuracy is relatively high.

Hardware-based approaches can overcome most of the shortcomings of the software-based ones. The corresponding depth maps can be generated in real-time and free from texture interference. Compared with matching-based approaches, the depth-camera-based approaches have higher accuracy. However, due to the intrinsic physical constraints of sensors and the active scanning approaches, depth-camera-generated depth maps, compared with traditional texture images, typically have low resolutions (e.g. 176×144 for SR4000 [45] and 640×480 for Kinect [46]) due to intrinsic noise and extrinsic environmental interference. Therefore, in order to successfully use depth maps in 3D applications, several depth map SR techniques have been proposed to increase the spatial resolution and the quality of depth maps.

2.2.2 Depth Map Quality Assessment

Some of the depth map assessment techniques are similar to those used for texture's, for example, PSNR and SSIM. However, in [47], a new depth map assessment, named Relative Error of Depth (REoD), has been proposed. The value of REoD can be obtained by

$$REoD = \frac{1}{H \times W} \sum_{i=1}^{W} \sum_{j=1}^{H} \frac{|D(i,j) - D_{GT}(i,j)|}{D_{GT}(i,j)} \tag{2.8}$$

where $\mathbf{D_{GT}}$ is the ground truth depth map, $\mathbf{D}$ is the assessed depth map.

Depth maps have special features in which most of the areas are homogeneous areas, sharp edges only exist between objects and they are not directly viewed by users. Therefore, some researchers argue that the assessment matrices of depth maps should not be the ones used for textures. Moreover, depth maps are usually used with 2D textures to reconstruct the 3D world, therefore, by using the DIBR technique the depth map errors often lead to object shifting or ghost artifacts on the synthesized views and these artifacts are different from the ordinary 2D distortions such as Gaussian noise, blur, and compression errors [48]. Hence, the depth map quality assessment should take the quality of the rendered view into consideration.

2.2.3 Depth Map Applications

3D video is replacing 2D video in many applications as it provides the viewers a novel spatial feeling and multiviews of a scene. Benefiting from the associated depth information, 2D-plus-depth and MVD are the two most commonly used 3D representations for real world reconstruction. Moreover, since the depth images represent three-dimensional (3D) scene information, they are commonly used for the DIBR technique to support 3D video and free-viewpoint video applications. A virtual view can be generated by the DIBR technique and its quality depends highly on the quality of depth image. Besides that, the depth information can also be applied to the navigation system of robots or walking aids for blind people.

2.3 Overview of Image/Video Super-Resolution

In sections 2.1.1 and 2.2.1, the acquisition of texture and depth maps have been introduced. In this section, firstly, the generation of observed LR images is described by a mathematical model. Secondly, a review of the methodologies from existing literature

relating to texture SR will be introduced based on the type of the input and output (image SR or video SR) as well as the implementation of SR algorithms (in the spatial and frequency domains). Next, depth map SR approaches will be presented and finally, four popular SR applications will be introduced.

Super-resolution is a process of reconstructing one or more HR images from input LR images based on the relationship model between the HR and LR images. Depending on the types of the input and output, the SR problem can be classified as a single input single output, a multiple input single output, and a multiple input multiple output spatial resolution increment problem. The first two categories have inputs of single or multiple images, which can be taken by a camera from one or several different viewpoints. The output is a single image with higher resolution than the input. They can be specified as image super-resolution problems and can be easily extrapolated to the third category, video super-resolution problems. In terms of implementation for SR problems, SR techniques can be classified into spatial and frequency domain techniques. The commonly used SR approaches, like reconstruction-based, learning-based and interpolation-based approaches, all belong to spatial domain SR techniques.

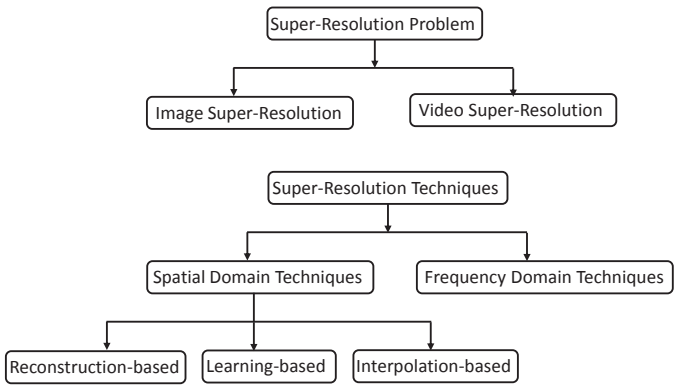


Figure 2.9: Classification of SR problems and approaches.

2.3.1 General Super-Resolution Observation Model

Fig.2.10 shows a model that describes the relationship between the observed LR images and the original HR image. Let us assume a desired or original HR image with size $L_1N_1 \times L_2N_2$ denoted as $\mathbf{x}$ and L_1 and L_2 are the down-sampling factors in the horizontal and vertical directions, respectively. Consequently, after warping, blurring, and subsampling performed on the HR image $\mathbf{x}$, the observed LR image is denoted as

$\mathbf{y}_k$ with size $N_1 \times N_2$. Corrupted by additive noise, each LR image can be represented by a mathematical model as

$$\mathbf{y}_k = \mathbf{D}\mathbf{B}_k\mathbf{M}_k\mathbf{x} + \mathbf{n}_k, 1 \leq k \leq K \tag{2.9}$$

All image variables in (2.9) are represented as column vectors composed of the pixel intensity of corresponding images in lexicographical order, thus, the transformation or effects applied to images can be represented as matrix multiplication operations. That is to say, in (2.9), the original HR image written lexicographically will be noted as the vector $x = [x_1, x_2, x_3, ..., x_N]^T$, where $N = L_1N_1 \times L_2N_2$ and the kth observed LR image is denoted as $\mathbf{y}_k$ and $y_k = [y_{k,1}, y_{k,2}, y_{k,3}, ..., y_{k,M}]^T$ where $k = 1, 2....K$ and $M = N_1 \times N_2$. $\mathbf{D}$ is a subsampling matrix with size $N_1N_2 \times L_1N_1L_2N_2$. $\mathbf{B}_k$ represents a $L_1N_1L_2N_2 \times L_1N_1L_2N_2$ blur matrix of the kth LR image which has the same size as the warping matrix $\mathbf{M}_k$ and the latter contains the motion information of the camera and the scene while capturing the images. $\mathbf{n}_k$ represents the noise matrix, and is usually assumed to be Gaussian white noise.

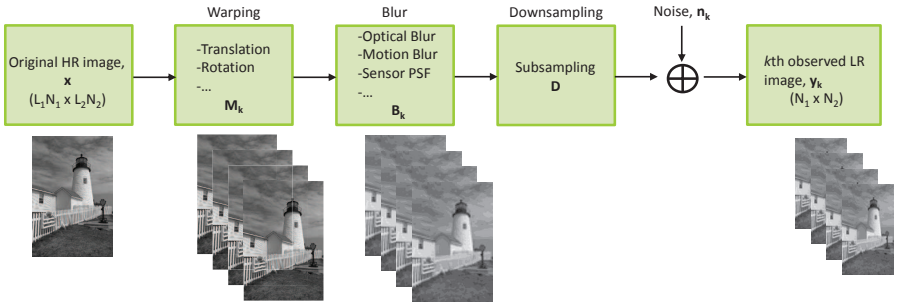


Figure 2.10: Block diagram of the SR observation model.

Since scene motion occurs during the image acquisition process, it may contain some global or local translation, rotation information, and so on. In general, this information, $\mathbf{M}_k$ is unknown. Therefore, the scene motion for each image has to be estimated by referring to one particular image. As one of the possible results of an optical system (e.g., out of focus and aberration), relative motion between the imaging system and the original scene, and the LR sensor, blurring matrix $\mathbf{B}_k$ can be either Linear Space Invariant (LSI) or Linear Space Variant (LSV). The downsampling matrix $\mathbf{D}$ results in an aliased LR image and finally, white Gaussian noise $\mathbf{n}_k$ is encountered both in the image acquisition and transmission process.

In reality, with the observed LR images, it is hard to distinguish each effect of these distortions. Hence, the model (2.9) can be unified in a simple matrix-vector form, as shown in (2.10).

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x} + \mathbf{n}_k, 1 \leq k \leq K \tag{2.10}$$

where $\mathbf{H}_k$ is the combination of the operations $\mathbf{D}$, $\mathbf{B}_k$ and $\mathbf{M}_k$. Based on this observation model, the aim of the SR image reconstruction is to solve the inverse problem and to estimate the underlying HR image $\mathbf{x}$ [49].

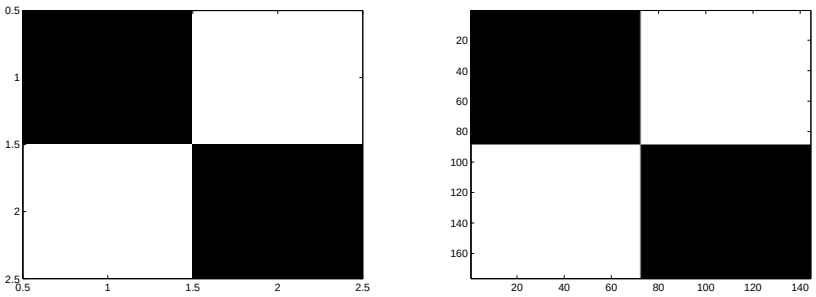
2.3.2 Texture Image Super-Resolution

Some existing SR algorithms for texture image SR are reviewed in the following subsections. Firstly, interpolation-based SR approaches that convey an intuitive comprehension of the SR image reconstruction are presented. Secondly, the reconstruction-based SR approaches are explained mainly focusing on the Iterative Back Projection (IBP) approach. Finally, one of the most popular SR trends, the learning-based SR approaches are presented.

Interpolation-based Super-Resolution

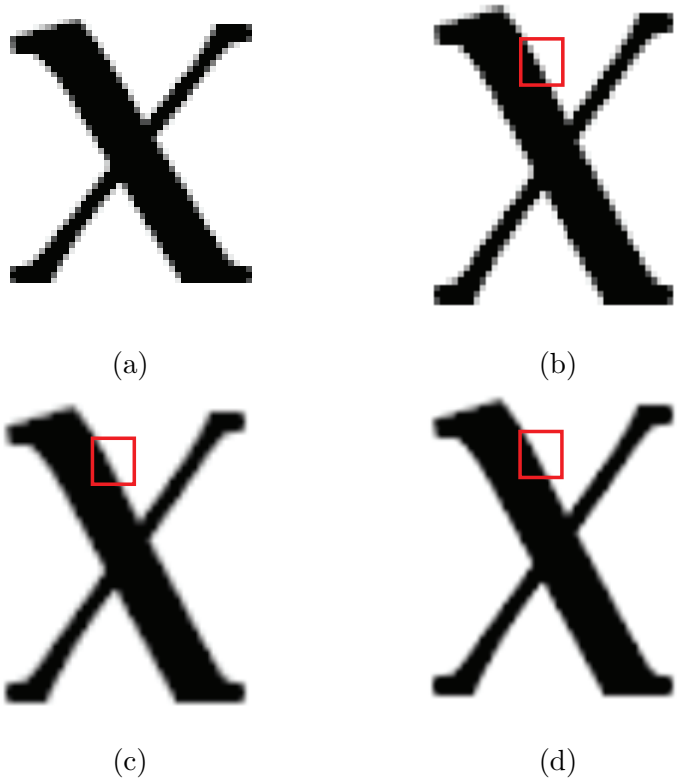
Interpolation-based SR approaches build on the image smoothness assumption, interpolating for the missing HR pixels by the surrounding LR pixels which can be achieved using a single image input. Nearest neighbor, bilinear and bicubic interpolations [50] [51] are conventional and typical interpolation-based approaches. Nearest neighbor interpolation is the simplest approach. Rather than calculating an average value by some weighting criteria or generating an intermediate value based on complicated rules, this method simply determines the "nearest" neighboring pixel, and uses this value to fill the missing pixel. As shown in Fig.2.11 below, the 2×2 checkerboard image is upsampled to a 176×144 image without any changes. Due to the simplicity of this algorithm, the operation takes little time to complete. Although the LR image is scaled by 6336 times, the HR image still has sharp horizontal and vertical edges. However, such good performance mainly exists in the integer interpolation ratio cases, and when it is applied on other patterns with non-integer ratio, it causes undesirable jaggedness. For example, in Fig.2.12 (b), the diagonal lines of the "x" in the interpolated image show the characteristic "stairway" shape.

Compared with nearest neighbor interpolation, bilinear and bicubic can reduce



(a) (b)

Figure 2.11: Using nearest neighbor to interpolate a checkerboard image. (a) the original LR image with size 2×2, (b) the interpolated HR image with size 176×144.



(a) (b)

(c) (d)

Figure 2.12: Comparison between common used interpolation methods. (a) the original LR image with size 53×49, the interpolated results of HR image with size 176×144 by using (b) nearest neighbor, (c) bilinear and (d) bicubic. In order to have a clear view of the original LR image, it has been shown in the same size as the other three images.

the visual distortion caused by the fractional interpolation ratio (Fig.2.12). Instead of copying the neighboring pixels (which often results in jaggy images), these two interpolation methods utilize the surrounding pixels to produce a smoother scaling at edges (Fig.2.12 (c) and (d)). The constructed HR pixels are generated by using 2 linear interpolations along the x and y axes, respectively. In this way, any pixel between the

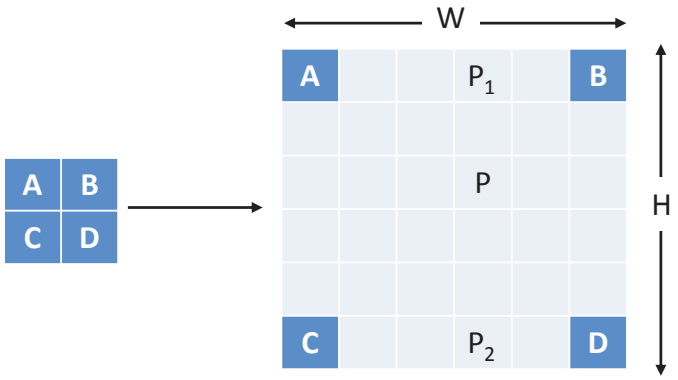


Figure 2.13: The working principle of bilinear interpolation.

LR pixels can be constructed by bilinear interpolation. Referring to Fig.2.13, A, B, C and D are four LR pixels and located at the corners of one texture area at positions $(1,1),(1,W),(H,1)$ and (H,W), respectively. P is the targeted HR pixel at position (i,j). The first linear interpolation is carried out along the x axis and the pixels P_1 and P_2 at position $(1,j)$ and (H,j) are obtained by

$$\begin{aligned} \frac{P_1-A}{j} &= \frac{B-A}{W} \\ \frac{P_2-C}{j} &= \frac{D-C}{W} \end{aligned} \tag{2.11}$$

Then, the second linear interpolation is carried out by using the pixels P_1 and P_2.

$$\frac{P-P_1}{i} = \frac{P_2-P_1}{H} \tag{2.12}$$

Substituting Eq.2.11 into 2.12 we get,

$$P = \frac{1}{HW}\left[A(W-j)(H-i) + Bj(H-i) + Ci(W-j) + Dji\right] \tag{2.13}$$

Similarly, all HR pixels among these four LR pixels can be interpolated.

In terms of performance, bicubic interpolation produces less blurring of edges and other distortion artifacts in comparison to bilinear interpolation, but it is more computationally demanding. Instead of using four pixels, bicubic interpolation fits a series of cubic polynomials to the intensity values contained in a 4×4 array of LR pixels surrounding the target HR pixel. It is also carried out in two steps. First, four cubic polynomials ($f(1)$, $f(2)$, $f(3)$ and $f(4)$) are fitted to four HR pixels along the y-direction (the choice of starting direction is arbitrary). Next, these four HR pixels are used to fit another cubic polynomial ($F(1)$) in the x-direction based on the interpolated brightness values that lie on the curves. In this way, the HR pixels at any

position can be obtained (Fig.2.14). Since the polynomial used in the bicubic interpolation algorithm can have a significant impact on the accuracy and visual quality of the interpolated image, splines as piecewise polynomial functions are often used.

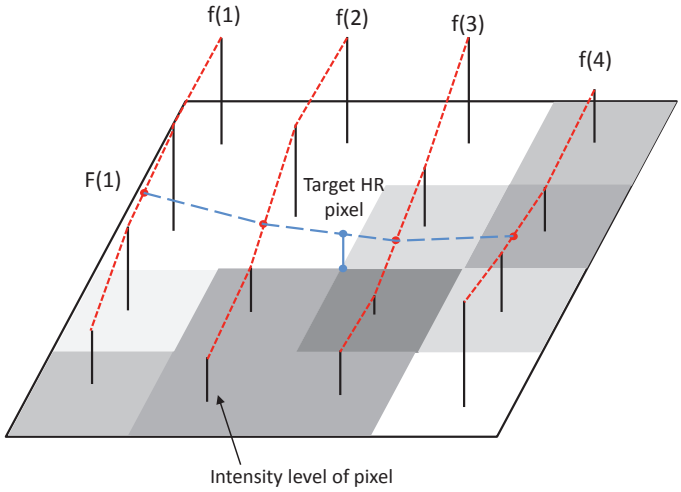


Figure 2.14: The working principle of bicubic interpolation.

Different from other types of SR methods, interpolation-based SR approaches gain their popularity in real-time applications mainly due to their computational simplicity and they have lower requirements on the number of input LR images. Therefore, interpolation-based SR approaches are suitable for applications with single input single image where only one degraded LR image is available at the input terminal. They have a good performance on smooth areas (low-frequency areas), but, work poorly on edges (high-frequency areas) [52]. This is a common drawback of the conventional interpolation methods where they cannot fully exploit the scene content during the interpolation process, and consequently they are prone to blur high frequency details (edges). In order to overcome these weaknesses, in [4], Li and Orchard proposed an edge-directed interpolation algorithm for nature images. The interpolation algorithm is composed of two steps, firstly, the local covariance coefficients of LR image are estimated and then based on the geometric duality between the LR covariance and the HR covariance, these estimated coefficients are used to steer the interpolation process. Plenty of simulation results were used to demonstrate the effectiveness of the edge-directed interpolation algorithm over conventional linear interpolations. In [5], a small-kernel bilateral filter was proposed to implement image SR based on a novel maximum posterior estimation. While maintaining local edge correlations, the global consistency

is constrained by the pixel-based soft-decision estimation [53] within a local window. Zhang *et al.* proposed a fast and effective image interpolation algorithm using a median filter. The pixels with clear directions are interpolated firstly in a non-linear iterative procedure. For the remaining pixels, a fast median-filter-based interpolation method is utilized [6]. These two filter-based methods are faster than other interpolation methods, which can be exploited for some real-time applications. In [54], Li *et al.* proposed to use the edge-direction information implicitly in the interpolation process, with the aid of a Markov Random Field (MRF) model. In this proposed algorithm, the edge directions are implicitly estimated with a statistical-based approach and represented by some weighting vectors. These weighting vectors are used to formulate geometric regularity constraints (smoothness along edges and sharpness across edges), which will be applied to the interpolated image through an MRF model. The experimental results indicate that compared with other edge-directed interpolation methods, the proposed MRF-model-based edge-directed image interpolation can improve the subjective quality of the final interpolated image, while not compromising the PSNR quality of the image. In Fig.2.15, three of these methods have been compared with the bicubic interpolation method.

Although interpolation-based single-frame SR algorithms are efficient in nature images, they have inevitable limitations. For example, the recovered information in most cases cannot represent all of the lost information and the high-frequency components that are lost or degraded during the LR sampling process. Super-resolving from a single LR image is known to be an ill-posed inverse problem due to the small number of observed LR images relative to the large number of missing HR pixels. Thus, the gain in quality in the single-frame SR approach is limited by the minimal information provided to recover missing details in the reconstructed HR signal. However, multiple-input image SR can overcome this weakness due to the multiple acquisitions of the same scene and more available inputs. In this thesis, we will try to benefit from the interpolation-based SR approaches, meanwhile, using multiple-input images/frames to generate HR images/video.

Reconstruction-based Super-Resolution

Reconstruction-based methods are usually applied on multiple input image SR and in order to regularize the ill-posed inverse problem, many approaches have been pro-

(a) (b)

(c) (d)

Figure 2.15: The interpolation results of (a) bicubic method (b) [4], (c) [5] and (d) [6] interpolate the 128×128 lena to 256×256.

posed, such as, Iterative Back Projection(IBP), Projection Onto Convex Sets (POCS), Maximum A Posteriori (MAP), and Maximum Likelihood (ML). Having more input information, multiple input SR approaches have better performance than single ones.

The Iterative Back Projection (IBP) method iteratively uses prior information of the previous results to get a better SR performance [55]. Given an initial estimated HR image, $\hat{\mathbf{x}}$ and an image quality degradation model, $\mathbf{H}$, it is possible to simulate a series of LR images, $\hat{\mathbf{y}}$, where $\hat{\mathbf{y}} = \mathbf{H}\hat{\mathbf{x}}$. The "back projection" procedure refers to backward projection of the error between the j-th simulated LR image $\hat{\mathbf{y}}_j$ and the observed LR image $\mathbf{y}$ via a backward projection operator $\mathbf{H}^{BP}$. All of the backward projected errors are added to $\hat{\mathbf{x}}$ to form the next iterative HR image. The whole process will be repeated until the stop rule is achieved, which is usually the minimum error between the stimulated LR image and the observed LR image. The IBP method can

be represented by (2.14).

$$\begin{aligned}\hat{\mathbf{x}}^{n+1} &= \hat{\mathbf{x}}^{n} + \mathbf{H}^{BP} \sum_{j=1}^{K} (\mathbf{y}_j - \hat{\mathbf{y}}_j^n) \\ &= \hat{\mathbf{x}}^{n} + \mathbf{H}^{BP} \sum_{j=1}^{K} (\mathbf{y}_j - \mathbf{H}\hat{\mathbf{x}}_j^n)\end{aligned} \tag{2.14}$$

where n is the current iteration number, K is the number of LR images, $\hat{\mathbf{y}}_j^n$ is the final version of the simulated j-th LR image after n iterations. Since it has a simple but powerful simulate-and-correct approach to reconstruct an image, many applications of the IBP method can be found in [56]. However, because SR problems are ill-posed, the solution to (2.14) is not unique and the choice of the priori image has a huge effect on the final outcome. Improper choice of the priori image constraint could result in a non-convergent or slowly convergent solution. All of these factors limit the adoption of IBP method.

Learning-based Super-Resolution

Although some reconstruction-based SR algorithms which use regularized constraints can tackle the ill-posed SR problem, the regularized constraints are usually defined as the prior smoothness knowledge of the HR image. However, when the magnification factor of the SR image increases, these constraints lead to over-smooth edges and the reconstruction model provides little useful information [57]. Moreover, the parameters of the capturing camera's point spread function have been assumed to be known in advance, which is not always a valid assumption. Therefore, in this situation, to obtain the priori knowledge from the image itself is very important and learning-based SR algorithms can overcome the weaknesses of reconstruction-based SR algorithms.

The basic implementation of learning-based SR algorithms is based on obtaining the relationship between HR and LR images by learning from the HR and LR image pairs [7]. Instead of the pre-defined priori knowledge in reconstruction-based SR algorithms, priori knowledge is obtained by learning from a huge training data set. The generation of the training set starts from a collection of many HR images and degrades each HR image to produce its corresponding LR version. Typically, an LR image with half the number of original pixels in each dimension (one-quarter the total number of pixels) of the HR image is obtained by blurring and downsampling the HR image. Then an initial analytic interpolation, such as bicubic interpolation, is applied to the LR image and generates an interpolated HR image with the same size as the original HR one but

without high frequency components. Only the high frequency component of the HR image will be stored. Moreover, passing through a high-pass filter, the high frequency component of the LR image can also be obtained. Fig.2.16 shows the training process.

Figure 2.16: Training process.

Freeman *et al.* [7] proposed a learning (example)-based scheme which was applied to generic images where the low to high resolution patch models are learned via a MRF model and loopy belief propagation is used for inference. It is worth noting that without considering the spatial neighbor information of the input LR image patch, due to insufficient geometry features, the closest LR training patches may result in different HR patches (Fig.2.17). Therefore, the spatial neighbor information should also be taken into account. As a result, two matching criteria should be met: 1) the matched LR training image patch should have similar content as the input LR image patch; 2) the corresponding HR training image patch should have content continuity with neighboring HR patches, which means they should not suffer from block artifacts (content discontinuity) in the reconstructed HR image. Although, the method can significantly preserve sharp edges and image details, it is somewhat dependent on the training set and the patch size. So the result is not stable and sometimes produces

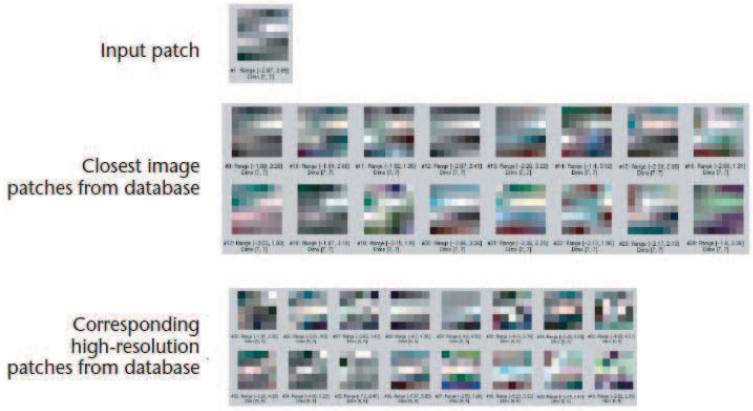


Figure 2.17: First row: an input patch; middle row: similar low-resolution patches; bottom rows: paired high-resolution patches. For many of these similar low-resolution patches, the high-resolution patches are different from each other [7].

artifacts in real applications [58].

With the aid of training data consisting of multiple LR-HR image patch pairs, Chang *et al.* [59] proposed to use Locally Linear Embedding (LLE) for single image SR purposes. For a given input LR image $\mathbf{x}$, it is broken into patches with the same size as the training LR image patches and each LR image patch $\mathbf{x}_i$ is used to search for similar patches $\mathbf{x}_i^1, \mathbf{x}_i^2, \cdots, \mathbf{x}_i^k$, from LR training images. Then the corresponding training HR patches $\mathbf{y}_i^1, \mathbf{y}_i^2, \cdots, \mathbf{y}_i^k$ linearly reconstruct the HR output image patch $\mathbf{y}_i$ with the weights $w_i^1, w_i^2, \cdots, w_i^k$ determined by LLE. The optimal weights should satisfy

$$w_i^k = \arg\min_{w_i^k} |\mathbf{x}_i - \sum_{k=1}^{K} w_i^k \mathbf{x}_i^k|^2 \tag{2.15}$$

where K is the number of neighbors searched in the training dataset. By linearly combining all the found training HR patches, the reconstructed HR image $\mathbf{y}$ can be expressed as

$$\mathbf{y} = \sum_{i=1} \sum_{k=1}^{K} w_i^k \mathbf{y}_i^k \tag{2.16}$$

However, using a fixed number of K neighbors for reconstruction often results in blurring effects, due to over- or under-fitting.

Based on the suggestion that the linear relationships among HR signals can be well recovered from their low-dimensional projections [60], Yang *et al.* [61] applied the sparse representation coefficients of LR image patches which were obtained from its over-complete dictionary to that of HR image patches to generate the target HR image. Instead of working directly with the LR-HR image patch pairs, the proposed approach

learns a compact representation for these patch pairs to capture the co-occurrence prior, and significantly improve implementation speed.

Besides utilizing one-to-one LR-HR image patch pairs, in [62] the distribution of HR patches relating to the same or similar LR counterpart(s) are adopted as the training data set. If the one-to-one correspondence between a previous LR-HR image patch pair is regarded as "hard information" in information theory, the one-to-many correspondence in this conditional distribution can be considered as "soft information". Relying on this prior knowledge, as well as the local consistency in the recovered HR image, the optimal HR counterpart can be selected among different HR image patches relating to the same given LR image patch. In this way, the ambiguity during patch mapping (Fig.2.17) can be reduced to a large extent.

In [63], the relationship between image patches are learned from different image scales rather than searching for similar patches from training image data or from different down-scaled versions of the original image. Therefore, in this way, the proposed method does not need a collection of training data or the HR image in advance and also removes the assumption of image patch self-similarity. Supported by Bayes theory, the optimal Support Vector Regression (SVR) is learned and picked up to form the SR output image. Extensive experimental results indicate the quantitative and qualitative effectiveness of the proposed self-learning SR method.

2.3.3 Texture Video Super-Resolution

Having an extra dimension of information than images, the video super-resolution problem can be implemented in two ways, one is spatial resolution enhancement, the other one is frame rate enhancement or frame rate up-conversion. As a multiple input, multiple output SR problem, the input video can be captured either from different viewpoints with the same frame rate or from the same viewpoint but with different frame rate. The former input type can generate super-solved multiview HR video (spatial resolution enhancement), while, the latter one can generate video with a high frame rate (frame rate up-conversion). It is worth noting that, the input videos can have different resolutions forming mixed-resolution inputs.

Spatial Resolution Enhancement

Since video can be regarded as many images that are captured at different times, most of the image SR problem approaches can be applied to implement video SR. Each frame of a LR video can be super-resolved to the target HR frame, and then the whole LR video has been super-resolved to the desired HR video.

Based on the LR image observed model, in [64] a Bayesian-based approach was used for adaptive video SR by simultaneously estimating the camera motion, blur kernel, and noise level while reconstructing the original HR frames. Due to the generalization of the motion and blur kernel, on one hand, it can achieve high accuracy for the estimation algorithm and high quality for the reconstructed HR frames. On the other hand, it involves many equations and many unknown parameters. Therefore, it is too complex and time consuming for real time video SR applications. Instead of using a Bayesian Maximum A Posteriori (Bayesian MAP) to determine the unknown parameters of each pixel, in [65], the Bayesian MAP is used to solve the block-based unknown parameters reducing the whole complexity while promising results are maintained. Work in [8] adopted a mixed-resolution video system where at least one of the views is captured at LR, while the others are captured at HR. Hence, in [8] the high frequency content has been extracted from the virtual view frame-by-frame and then added to the corresponding LR frame to reconstruct the HR frame. However, in this work, the high frequency content is extracted from the whole frame, thus the local characteristics of the scene are not taken into account.

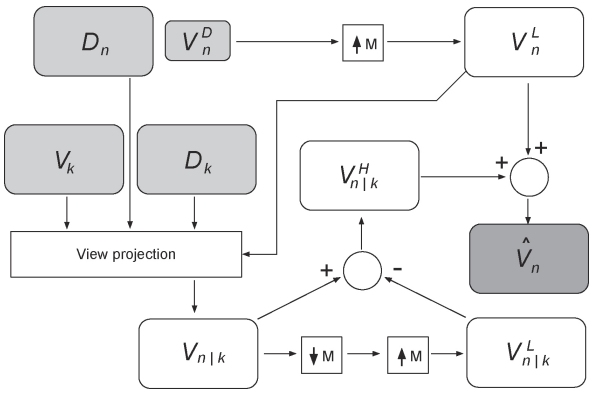


Figure 2.18: SR approach for multiview images. A super-resolved image $\hat{V}_n$ is created from its low-resolution version, V_n^D, a neighboring HR view, V_k, and the depth information for each of these views, D_n and D_k [8].

In this thesis, Chapters 3 and 4 propose two solutions to video spatial resolution enhancement. In Chapter 3, virtual views have been utilized to recover FR frames in a Mixed-Resolution Multiview Video plus Depth (MR-MVD) framework. The local similarity between the LR view and its corresponding virtual view has been used to steer the FR recovery mechanism. In Chapter 4, in addition to super-resolving one LR view, the two FR views are downsampled before encoding and super-resolved after decoding by exploiting inter-view redundancy via virtual views.

Frame Rate Up-conversion

Frame Rate Up-conversion (FRUC) is required for applications such as NTSC- PAL conversion and display on HDTV, where high frame rates are desired [66] [67]. The most commonly used frame rate up-conversion methods are frame repetition, linear interpolation, and motion compensated interpolation. Relying on temporal correlation of the original video sequence, many FRUC algorithms adopt a motion compensation technique to construct the up-sampled frame [68]. Motion compensation is bi-directionally carried out in order to take into account frames on both sides of the up-converted frame. From a theoretical point of view, the implementation of linear interpolation and motion compensation-based interpolation are simple, however, they cannot deal with temporal aliasing caused by capturing the video below the Nyquist frequency of the motion trajectory, the true motion of the object cannot be recovered even by performing ideal temporal interpolation (Fig.2.19). Hence, an insufficient frame rate will result in inaccuracy in the motion estimation by FRUC.

Spatio-temporal Resolution Enhancement

Recently, many consumer digital cameras support a dual shooting mode of both LR video and HR image. By periodically switching between the video and image modes, this type of camera makes it possible to super-resolve the LR video with the assistance of neighboring HR still images. Zhai and Wu proposed the convertion of LR video to HR video which has the same resolution as the auxiliary HR still images [69]. The target HR frames are modeled by a 2D Piecewise Autoregressive (PAR) process and the PAR model parameters are learned from these HR still images.

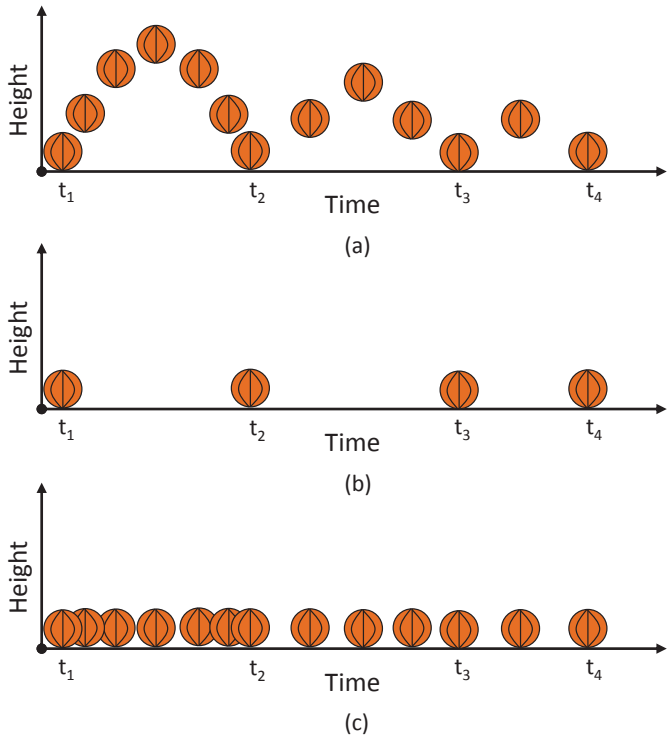


Figure 2.19: Temporal aliasing. (a) Trajectory of a ball over time. (b) Trajectory sampled over time by a low frame rate camera. Perceived trajectory is along a straight line. (c) Illustration that even with ideal temporal interpolation of (b) the true motion trajectory cannot be recovered.

2.3.4 Depth Map Super-Resolution

In the section 2.2.1, we have introduced four methods that can be applied to obtain a depth map. However, the captured depth maps have, in general, limited spatial resolution which is much lower compared to that of the corresponding color image (1280×1024). Hence, in reality, depth camera generated depth maps need to be super-resolved, so that it can be used in future applications. Based on the input source information, depth map super-resolution can be classified into two types of approach: depth map SR based on depth information and depth map SR based on texture and depth information.

The observation model of a LR texture image is also suitable for the depth camera generated depth map which contains warping, blurring, subsampling operators, and additional noise.

$$\mathbf{d}_k^L = \mathbf{D}\mathbf{B_k}\mathbf{M_k}\mathbf{d_k^H} + \mathbf{n_k}, \mathbf{1} \leq \mathbf{k} \leq \mathbf{K} \tag{2.17}$$

where $\mathbf{d}_k^H$ is the depth map of the original HR scene and $\mathbf{D}$, $\mathbf{B}_k$, $\mathbf{M}_k$ are the subsam-

pling, blurring and warping matrices, respectively. According to the above mathematical model, the super-resolution of a depth map can be realized by similar approaches as the texture SR. However, a depth map has a piecewise smooth property which is different from the texture and the depth image contains less information compared with texture. Consequently, it is difficult to directly adopt the texture image SR algorithm. Xie *et al.* [70] developed a coupled dictionary learning approach with a locality constraint for single depth image SR. Due to the piecewise smoothness of a depth map, directly using sparse features to constrain the dictionary learning will result in a dictionary over-fitting problem, i.e. similar LR patches, which are properly represented by the learned dictionary bases produce significantly different HR patches. Inspired by Local Coordinate Coding (LCC) [71], coupled dictionary learning becomes beneficial with a locality constraint and it can make sure the input depth patches are similar to the dictionary atoms with non-zero coefficients. In this way, the over-fitting problem can be prevented. An adaptively regularized Shock Filter was also applied to reduce jagged noises in the depth image while sharpening edges.

Since the information contained in a single depth map is fairly small and the same scene information contained in multiple depth maps can be complementary, when compared with a single depth map SR, fusion of multiple LR depth maps to get a HR image can achieve superior results in SR. Following the general principle of texture image SR methods, Schuon *et al.* [72] modeled the depth map SR problem as an energy minimization problem that jointly employs a data term, enforcing similarity between the input and output images, and a bilateral regularization term for edge-preserving smoothness. However, they rely on the assumptions that multiple range images are available with small camera movement and the objects motion is global or rigid, which may not be true for many practical applications. Hence, this approach may fail when large movement occurs. In order to tackle this problem, in [73] Ismaeil *et al.* put forward a novel dynamic SR algorithm that is capable of accurately super-resolving a depth sequence containing one or multiple moving objects without prior assumption of their shape or motion, and no additional post-processing is needed.

On one hand, due to the textureless nature of a depth map, with little useful depth information, the single depth map SR is still a challenging problem. On the other hand, cheaper and more easily accessible texture cameras with HR texture images have been utilized in depth map SR. Recently, a framework called the RGB and Depth

images (RGBD) system, in which a depth camera coupled with a RGB camera becomes popular in depth map SR. Fig.2.20 shows two kinds of RGBD systems. Fig.2.20 (a) is a framework with a ToF camera, SR4000 and a RGB camera. While, Fig.2.20 (b) shows a Microsoft developed camera, Kinect, which consists of one depth camera and one RGB camera. After good synchronization, depth and texture image cameras can capture the same scene at the same time. Based on RGBD systems, many methods have been

(a) (b)

Figure 2.20: (a) framework consists of a 3D-ToF camera, SR4000 and a RGB camera (b) Microsoft developed depth camera, Kinect.

proposed to exploit the co-occurrence of depth and texture image discontinuities on depth textureless areas with careful registration for the object of interest. Kopf *et al.* proposed to upsample the LR depth map by using a Joint Bilateral Upsampling (JBU) filter. Aided by the associated HR texture, the edges of the upsampled depth map can be well preserved [74]. A similar but advanced joint bilateral filtering technique was proposed in [75] which iteratively refines the input LR depth map by referring to the registered HR textures. Although the adoption of texture information can help to get sharp depth edges, the color or lighting variations on the same areas of the texture images can cause false discontinuities in HR depth maps. Hence, texture images need to be used in a more sophisticated way. By applying the MRF to super-resolve the LR depth map, Diebel *et al.* formulated the SR process as an energy minimization problem to fuse LR depth images and HR texture images. Lo *et al.* [76] proposed the incorporation of a texture-guided weighting factor into the MRF model to reduce the texture copying artifacts with the weighting factor obtained based on a learning approach. Although the demonstrated results were good, the learning-based approaches usually have a higher computational complexity, which might prevent their adoption to real-time applications.

In conclusion, there are still some challenges that need to be addressed in depth map SR problems.

2.3.5 Super-Resolution Applications

Thanks to many researchers for dedicating themselves to the development of SR techniques, SR techniques have been adopted in many applications. Among them, there are four primary applications: target recognition in the video surveillance field, remote sensing, medical imaging [77] and consumer electronics.

In video surveillance applications, the target of interest is hard to identify and recognize under tough lighting conditions and capture equipment in degraded videos and images. Therefore, given a series of LR images and video, SR can be used to reconstruct HR images and video to make the target of interest clear. He and Schultz proposed a coarse-to-fine SR algorithm in [77] to recover the HR video captured from a LR unmanned aircraft digital imaging payload in real-time. Firstly, the coarsely super-resolved video is obtained by piece-wise registration and bicubic interpolation between every additional frame and a fixed reference frame. The refined video is generated by calculating pixel-wise medians in the coarsely super-resolved video. In this way, no iterations are involved in this implementation.

Satellite remote sensing contributes to Earth observation, vegetation health, bodies of water, and climate change based on image data gathered by wireless equipments over time. However, due to the size and weight limitation of satellite camera and affections from transmission, the obtained images are usually of low resolution quality. Therefore, it limits higher spatial resolution applications (e.g., intra-urban). Pan *et al.* proposed an SR method that consists of Compressive Sensing (CS), Structural Self-Similarity (SSSIM), and dictionary learning for reconstructing remote sensing images [78]. The dictionary is formed by extracting similar structures which often exist in remote sensing images, thereby, an HR image can be reconstructed using the dictionary in the CS framework. In this work, the effectiveness of the proposed SR method is evaluated by a new SSSIM index. In order to solve the SR problem in multi-angle remote sensing images, Zhang *et al.* proposed an adaptive weighted SR reconstruction algorithm that uses different weights to determine the contributions of different multi-angle LR images [79]. Since multifractal characteristics are common in natural images and based on self-similarity, some details in one natural image can be estimated from its larger or smaller

scale version. Therefore, in [80], the presence of multifractal characteristics is firstly explored in the LR remote sensing images, and then parameters of the information transfer function and noise are estimated. Finally, a fractal coding-based denoising and downscaling method is utilized to generate a noise-free and super-resolved image.

Medical imaging has become an important tool for medical diagnosis. Due to the nature of its usage, medical images have much less tolerance for image processing artifacts than other applications [81]. Therefore, it has higher requirement on the resolution and quality. Fortunately, medical imaging systems usually work within highly controlled environments (e.g. illumination) with highly similar objects (e.g. human organs), that is to say, plenty of prior knowledge about the anatomy or biology can be used to improve medical image quality. For example, in [81] a three-stage (registration, reconstruction, and restoration) SR algorithm was proposed to address two challenges in X-ray image SR that the large amount of data associated with digital mammogram images and the limited total radiation exposure which should be less than that of a normal X-ray image dosage. In [82], a SR technique was used to increase the resolution of coronal images. For further information, a comprehensive literature review of SR applications in medical imaging can be found in [83].

Entertainment and digital communication applications occupy a huge part in consumer electronics. In 2015, Samsung Display said they were planning to develop an $11K$ super-resolution display in the next five years together with industry and education. However, the development of capturing cameras can not reach the $11K$ frame size, therefore, SR techniques can offer a good solution by enlarging the resolution of perceived images/video and removing the visual artifacts caused by video compression [18]. In order to extend the appeal and usefulness of the broadcast service, meanwhile, reducing the unnecessary cost of HR video delivery, Boon *et al.* proposed a fast SR approach, which is based on recent sparse recovery SR techniques. In this way, the mobile terminals can show a much enhanced version of the broadcast video on nearby high-resolution devices without further cost [84].

2.3.6 Summary

In this Chapter, the basic concepts of texture and depth map have been introduced via the acquisition and assessment. In terms of color information, texture images contain more information than depth maps, therefore,

texture are usually compressed before transmission. Due to the high visual requirement from viewers, the low quality views need to be super-resolved. Thus, in this Chapter, the working principles, advantages and disadvantages of different texture and depth map SR approaches are also discussed. Among three mainly used SR methods which are reconstruction-based, learning-based and interpolation-based methods, due to easy implementation and well time performance, interpolation-based methods are prevailing in texture SR. However, most of the methods are still based on 2D video systems which may not take the depth information into account. Hence, directly applying these methods on 3D video systems does not fully exploit the 3D information. Motivated by this fact, in this thesis the depth information is introduced into the SR process in 3D video systems. Finally, the four most frequently used SR applications are investigated.

Chapter 3

Depth-Map-Assisted Texture Super-Resolution for Mixed-Resolution System

3.1 Introduction

In recent years, 3D video technology has drawn significant attention with more and more products and services becoming available on the consumer markets. They can provide viewers the perception of real-world scenes relying on large amounts of texture and depth map data captured from various viewpoints. Hence, this puts pressure on the acquisition, storage and transmission processes, especially for limited bandwidth applications [10]. One effective solution, for such kind of problem, has been proposed in [85] [86] that uses MR video, in which at least one of the views is captured at LR, while the others are captured at FR. The MR video in comparison with FR video significantly reduces the amount of captured, transmitted, and stored data as well as processing time, which is the bottleneck in real-time applications. Nevertheless, in order to meet the requirements of high definition, reduce visual uncomfortableness and make the video format more suitable for FTV, the LR video needs to be super-resolved to FR size using SR techniques at the decoder side [85]. Therefore, in a MR video system, the final quality will depend on the performance of the SR algorithm.

In general, image SR algorithms can be classified into three categories: reconstruction-based SR algorithms [87] [88], learning-based SR algorithms [7] [89] and interpolation-based SR algorithms [90] [54]. Compared with the previous two kinds of SR methods, interpolation-based SR methods have gained more popularity in real-time applications mainly due to their computational simplicity. However, the main drawback of this kind

of method is the inability to fully exploit the scene content during the interpolation process, and consequently they are prone to blurring high frequency details (edges). In order to overcome this weakness, Zhang *et al.* proposed to adaptively fuse the LR pixels on the two diagonal directions according to the Linear Minimum Mean Squares-error Estimation (LMMSE) technique [90]. Garcia *et al.* in [8] proposed the use of high frequency content from neighboring FR views and the corresponding depth information to recover the high frequency content in the LR view.

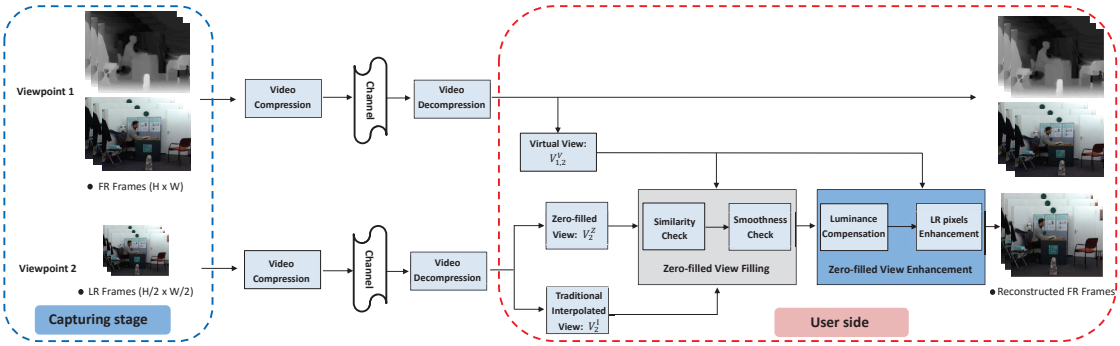


Figure 3.1: The framework of the proposed super-resolution method.

In this chapter, a new depth-map-assisted SR and enhancement algorithm is proposed where virtual view information and interpolated frames are exploited to provide two benefits. Firstly, the high frequency information contained in FR views can be properly utilized to super-resolve LR views. Secondly, the inter-view redundancy will be used to enhance the original LR pixels in the super-resolved views and to compensate the luminance difference between views. Experimental results have shown that the proposed algorithm achieves superior performance with respect to interpolation-based algorithms.

The rest of this chapter is organized as follows. Details of the proposed SR method will be introduced in Section 3.2. Several algorithms and explanation for choosing the thresholds are given in Section 3.3. Generalization of the proposed method is presented in Section 3.4, and experimental results are presented in Section 3.5. Section 3.6 concludes this chapter and discusses future work based on this current work.

3.2 Proposed Super-Resolution Method

In [91] and [92] it has been shown that a comfortable viewing of MR format could be achieved when the resolution of the FR view is twice as much as the resolution of the

LR view in both horizontal and vertical directions, whereas, higher ratios of the FR to LR resolutions will result in unacceptable subjective quality. In the following this ratio will be dubbed the *resolution factor* for brevity. This work will only address resolution factor two based on the findings in [91] and [92]. The framework of the proposed virtual view assisted interpolation-based SR algorithm is depicted in Fig.3.1. At viewpoint 1 the FR textures and associated depth maps with frame size $W \times H$ are compressed and transmitted to the receiver side. Meanwhile, the texture at viewpoint 2 has half the resolution of the FR view in both horizontal and vertical directions. The FR decoded textures and depth maps will be denoted by $\mathbf{V}_1^F$ and $\mathbf{D}_1^F$, respectively, while, the decoded LR texture sequence will be denoted by $\mathbf{V}_2^L$. At the decoder side, the decompressed LR view is used to generate two intermediate FR versions at viewpoint 2. The first version ($\mathbf{V}_2^I$) is obtained by using an interpolation method, such as bilinear or bicubic. The second version ($\mathbf{V}_2^Z$) is the zero-fill version of the LR view, where the original LR pixels placed at positions with indices $(2i-1, 2j-1)$ are separated by inserted zeros. This version will be used as a basis to generate the final super-resolved FR version at viewpoint 2. The zero-inserted positions in this frame will be replaced by pixels from either the interpolated view, $\mathbf{V}_2^I$, or from the FR view generated virtual view at viewpoint 2 using the 1D DIBR process [93] from one reference view to another without any post-processing (i.e., no hole filling). This virtual view will be referred to $\mathbf{V}_{1,2}^V$ in the following sections. Deciding which pixels to use to replace the inserted zeros will be driven by a similarity and smoothness check mechanism which will be explained in Section 3.2.1. To further improve the quality of the super-resolved FR frames at viewpoint 2, the enhancement methods will be proposed in Section 3.2.2.

3.2.1 Zero-filled View Filling

To generate an FR frame from the corresponding LR version and recover most of the lost high frequency information in the capturing stage, both the virtual view and the interpolated frame are used as candidates in this work. Since the virtual view is only synthesized from the neighboring FR view, the inter-view redundancy and the high frequency component of the FR frame, can be exploited in the proposed SR approach. However, the virtual view might be affected by holes and cracks due to the wrapping process and inaccurate depth map. Therefore, the similarity between the original LR pixels in $\mathbf{V}_2^Z$ and the corresponding pixels in the virtual view is measured to ensure

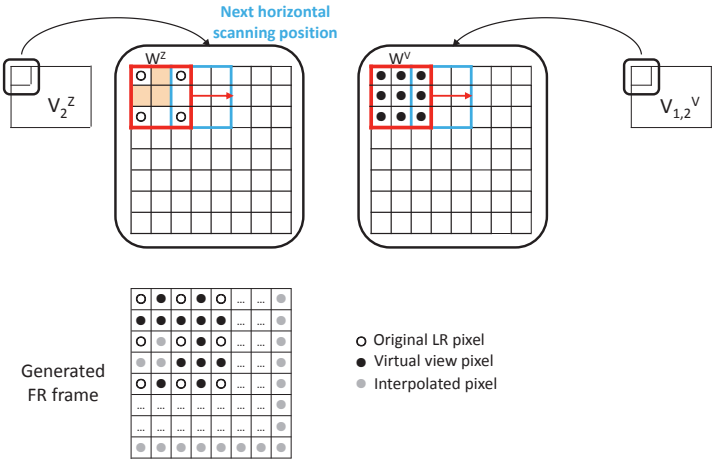


Figure 3.2: A pictorial representation of the similarity check process and the generation of FR frame.

that only proper virtual view pixels are selected to replace the zero-filled pixels in $\mathbf{V}_2^Z$. This process will minimize the probability of copying holes from the virtual view into $\mathbf{V}_2^Z$. The similarity check mechanism consists of two 3×3 scanning windows W^Z and W^V which synchronously scan the zero-filled view and the virtual view, respectively. A pictorial representation of this process is shown in Fig.3.2. The centers of these windows are used as the origin of their coordinate systems, thus for example $W^Z(-1,-1)$ stands for the upper left corner pixel in the window W^Z. The two windows move in a raster-scan mode by sliding two pixels at a time, so as to be always centered at the zero-filled pixels, i.e., $(2i, 2j)$ with $1 \le i \le H/2$ and $1 \le j \le W/2$. This ensures that for the zero-filled view, there are four LR pixels at the corners of the window W^Z to measure the local texture similarity between the zero-filled view and the virtual view. In this work, the Sum of Absolute Difference (SAD) is used for this purpose as below[1]:

$$D_{SC} = \sum_{\eta \in \{-1,1\}; \theta \in \{-1,1\}} |W^Z(\eta,\theta) - W^V(\eta,\theta)| \tag{3.1}$$

In this case, a hole due to the DIBR process in any corner of W^V will lead, in general, to a large SAD value. Therefore, this will be used as an indication that the local virtual view pixels in the current window, W^V, are not appropriate for filling the corresponding zero positions in the zero-filled view. Consequently, the zero-filled pixel $(2i, 2j)$ and its two causal neighbors, i.e., $(2i-1, 2j)$ and $(2i, 2j-1)$, will be filled by the corresponding interpolated pixels from $\mathbf{V}_2^I$ if the SAD value is larger than a threshold

[1]The SAD and Euclidean distance in this case will almost lead to the same results.

T_{si}, as shown in the following equation:

$$V_2^Z(\eta,\theta) = V_2^I(\eta,\theta)\ ;\ D_{SC} \geq T_{si} \tag{3.2}$$

where $(\eta,\theta) \in \mathcal{C}$ and $\mathcal{C} = \{(2i,2j),(2i-1,2j),(2i,2j-1)\}$. Hence, except for pixel-size holes located at the zero-filled positions, this mechanism will minimize the possibility of mistakenly copying hole pixels from W^V into W^Z.

For the case when the SAD measure is smaller than T_{si}, which indicates that the diagonal pixels in the two windows are relatively similar, a further check is carried out to determine the proper approach to fill the zero-filled positions in V_2^Z. If the area encompassing W^Z is smooth then interpolation algorithms could be better than the virtual view to estimate the zero-filled pixels. This is because chromatic discrepancies among different viewpoints make the obtained virtual view pixels less accurate than the interpolated pixels in representing the missed information for smooth areas. The chromatic discrepancies phenomenon happens due to the scene illumination difference, camera calibration and jitter speed, even if the capturing cameras have been adjusted to the same configuration [94]. Hence, based on this fact zero-filled pixels in smooth areas will be replaced by their counterparts from $\mathbf{V}_2^I$. On the other hand, for non-smooth areas, such as edges, interpolation algorithms intrinsically fail to estimate proper values for the zero-filled pixels, whereas, the virtual view generated from the FR view carries significant amount of information related to those non-smooth areas. Thus, for this kind of areas the zero-filled pixels will be replaced by their counterparts in the virtual view $\mathbf{V}_{1,2}^V$.

The previous paradigm is implemented in the second step, where the smoothness of a 3×3 area, W^I, centered at the pixel $(2i,2j)$ in $\mathbf{V}_2^I$ is checked; in this work, this has been done by measuring the standard deviation, σ^s. The motivation behind using the window W^I to measure the local smoothness, is that a non-trivial interpolator uses more than 8-connected neighbors in the estimation process to preserve the local regularity [95]. Consequently, the five estimated pixels along with the four corners of W^I carry more information about the local smoothness of the area, than the four LR pixels at the corners of the W^Z window. The outcomes of the smoothness check stage could be summarized by the following equation:

$$V_2^Z(\eta,\theta) = \begin{cases} V_2^I(\eta,\theta) & ;\ \sigma^s < T_{sm} \\ V_{1,2}^V(\eta,\theta) & ;\ \sigma^s \geq T_{sm} \end{cases} \tag{3.3}$$

where $(\eta, \theta) \in \mathcal{C}$. In Eq.(3.3), T_{sm} is a threshold to determine whether an area surrounding the pixel $(2i, 2j)$ has smooth or non-smooth texture. A flowchart depicting the similarity check and smoothness check stages is shown in Fig.3.3. As for the boundary pixels, they are copied from the interpolated view directly.

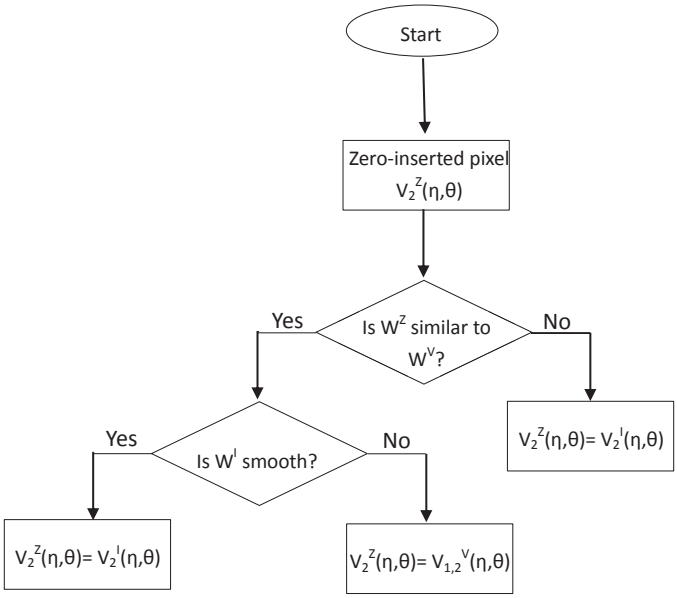


Figure 3.3: Flowchart of the Zero-filled View Filling stage.

3.2.2 Zero-filled View Enhancement

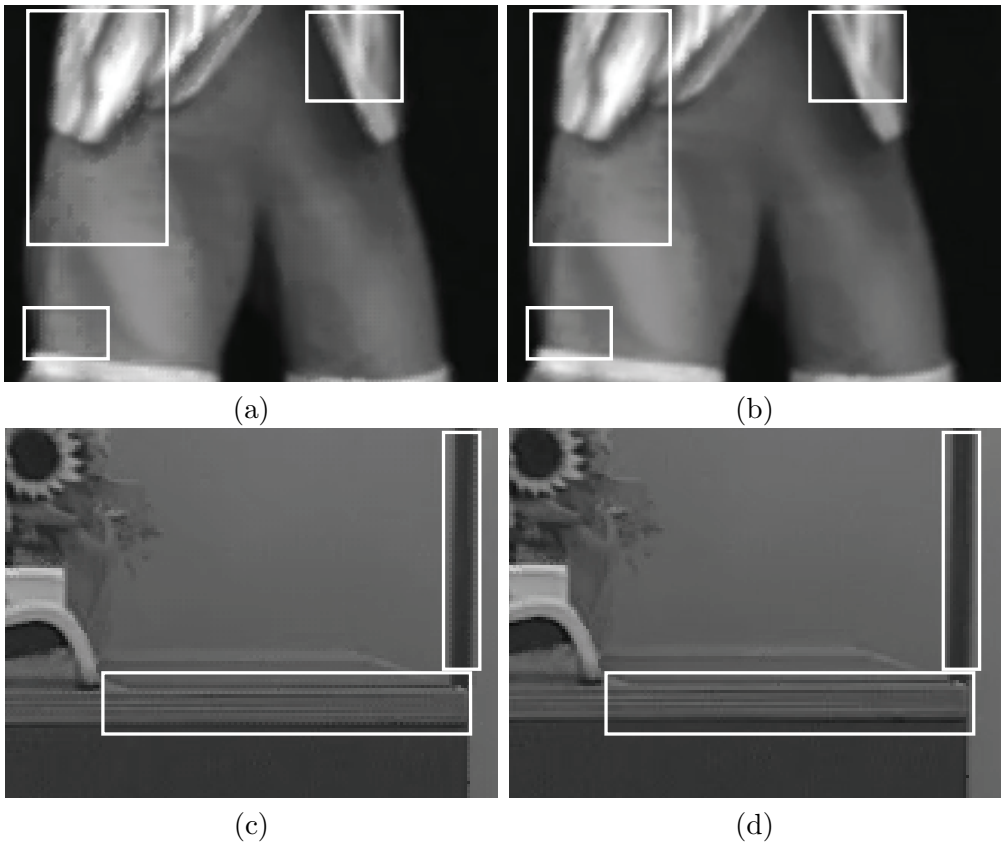
In the previous stage, all of the zero-filled positions will be filled by either virtual view pixels or interpolated pixels. However, the recovered FR frame will be affected by compression distortion, virtual view introduced distortion and interpolation induced distortion, therefore, in this work two methods are proposed to reduce the overall distortion, and enhance the final quality of the generated FR view.

Luminance compensation

In real video capturing scenarios, different views will have slightly different luminance. In addition, the LR and FR views will have different quality after compression, especially at large Quantization Parameters (QP), this is demonstrated in Table 3.1 for four sequences. This is because the LR view has more details in each macroblock than the FR view, thus even with the same QP the quality of its compressed version will be lower than its counterpart in the FR view. All of these factors cause jagged edges in the reconstructed FR frames when using the virtual view to recover the zero-filled positions

Table 3.1: The PSNR differences (dB) between the FR and LR views using H.264 for: (a) Bookarrival; (b) Doorflower; (c) Laptop; (d) Champagne with QP = 22, 27, 32, 37, 42, 47

Seq.	Bookarrival	Doorflower	Laptop	Champagne
QP	ΔPSNR(dB)	ΔPSNR(dB)	ΔPSNR(dB)	ΔPSNR(dB)
22	1.00	0.78	0.83	0.65
27	1.48	1.15	1.22	1.28
32	2.07	1.88	1.82	2.23
37	2.55	2.42	2.31	2.75
42	2.69	2.68	2.56	2.80
47	2.22	2.20	2.19	3.05



(a) (b)

(c) (d)

Figure 3.4: Comparison of the effect of luminance compensation on the first frame of "Pantomime" and "Bookarrival" sequences. The two images on the left show the artifacts in the super-resolved frames without luminance compensation and the two images on the right show the visual effects of same frame but after luminance compensation (better perception could be achieved by viewing the images at their full resolutions, which are 620×775 for (a) and (b); 620×884 for (c) and (d)).

in $\mathbf{V}_2^Z$, and an example of this artifact is shown in the highlighted areas in Fig.3.4 (a) and (c). Hence, a luminance compensation mechanism is proposed, which adjusts the brightness of the copied pixels from the virtual view into $\mathbf{V}_2^Z$ (i.e., the virtual-view-

based recovered pixels) to have harmonious brightness with the surrounding LR pixels. If the set of pixels $\mathcal{C}$ of the current pixel $(2i, 2j)$ is recovered from the virtual view, the average luminance difference between the two sliding windows W^Z and W^V centered at $(2i, 2j)$ will be evaluated as:

$$D_{LC} = \frac{1}{4} \sum_{\eta \in \{-1,1\}; \theta \in \{-1,1\}} [W^Z(\eta, \theta) - W^V(\eta, \theta)] \tag{3.4}$$

If the absolute value of D_{LC} is larger than a threshold T_l, the compensation process will be used to update the intensity of the pixels $\mathcal{C}$. The reason behind using the threshold T_l is to eliminate the effect of compression distortion on the luminance compensation process. In fact, given that a small number of pixels are used to estimate D_{LC}, it will be highly likely that this estimated value is biased by the amount of compression distortion affecting W^Z and W^V. Nevertheless, the use of the threshold will ensure that mainly luminance differences get compensated, and the small window size will ensure that luminance compensation is performed locally. Once the luminance compensation process is invoked for a set $\mathcal{C}$, then for each of its three pixels the proper amount of compensation will be determined by using the closest available neighbors for each of the pixels in $\mathcal{C}$ to avoid blurring edges. For example, for the pixel $(2i - 1, 2j)$ its two horizontal neighbors $(2i - 1, 2j - 1)$ and $(2i - 1, 2j + 1)$ will be used to evaluate its compensation value ΔY^h:

$$\Delta Y^h = \frac{1}{2} [\Delta W(-1, -1) + \Delta W(-1, +1)] \tag{3.5}$$

where $\Delta W(\eta, \theta) = W^Z(\eta, \theta) - W^V(\eta, \theta)$. Now the position $(2i - 1, 2j)$ in the zero-filled view will be filled by $V^V_{1,2}(2i - 1, 2j) + \Delta Y^h$ instead of $V^V_{1,2}(2i - 1, 2j)$. Similarly, the compensation value for the pixel $(2i, 2j - 1)$ will be computed starting from its two vertical neighbors as $\Delta Y^v = \frac{1}{2} [\Delta W(-1, -1) + \Delta W(+1, -1)]$ and consequently $V^Z_2(2i, 2j-1) = V^V_{1,2}(2i, 2j-1) + \Delta Y^v$. As for the center pixel, $(2i, 2j)$, it will be updated as $V^Z_2(2i, 2j) = V^V_{1,2}(2i, 2j) + \Delta Y^c$. However, given that the pixel $(2i, 2j)$ is at equal distance from the four corners its compensation value will be evaluated as:

$$\begin{aligned} \Delta Y^c = \frac{1}{4} [&\Delta W(-1, -1) + \Delta W(-1, +1) \\ &+ \Delta W(+1, -1) + \Delta W(+1, +1)] \end{aligned} \tag{3.6}$$

Some luminance compensation results are shown in Fig.3.4 (b) and (d).

LR pixels enhancement in the super-resolved view

The previous subsection proposes an enhancement mechanism for the virtual-view-based recovered pixels, whereas, in this subsection a mechanism to enhance the quality of the other pixels in $\mathbf{V}_2^Z$, i.e., the original LR pixels, is proposed. This is particularly important given that these pixels suffer from more compression distortion than their counterparts in the FR view, as shown in Table 3.1. The proposed enhancement method in this subsection exploits the inter-view correlation to achieve its objective. In fact, in a multiview system adjacent views have a large similarity, so the same content may appear in two different positions in two adjacent views. Hence, if the same content is separately encoded in the two views then their compression distortions could be partially canceled out. To show why the proposed mechanism improves performance, and to show its principle, let us consider a point in the scene $\hat{v}$ which is viewed from two viewing points, which means it is not occluded in either of these two views. Let us denote the projection of this point into viewpoint 1 and viewpoint 2 by $\hat{V}_1^F(\mu,\nu)$ and $\hat{V}_2^L(i,j)$, respectively. Apart from small differences, due to the nature and relative position of the lighting source in the scene, the previous two values could be regarded as similar, i.e., $\hat{V}_1^F(\mu,\nu) \approx \hat{V}_2^L(i,j) = \hat{v}$, the smaller the baseline is the more correct this assumption is[2]. Hence, in the following sections, we will assume that $\hat{V}_1^F(\mu,\nu) = \hat{V}_2^L(i,j) = \hat{v}$. These two projections will be compressed separately in the two viewpoints, so they become: $V_1^F(\mu,\nu) = \hat{v} + d_1$ and $V_2^L(i,j) = \hat{v} + d_2$, where d_1 and d_2 are the distortion caused by video compression on view 1 and 2, respectively. Since the compression can be treated as a random process with the mean value being $E\{d_1\} = E\{d_2\} = 0$, the variance of the distortion affecting view 1 and 2 will be $\sigma_1^2 = E\{d_1^2\}$ and $\sigma_2^2 = E\{d_2^2\}$, respectively. Then at the decoder side, and as explained previously, the zero-filled view is obtained from the LR view by inserting zeros in between its pixels, thus:

$$V_2^Z(2i-1, 2j-1) = V_2^L(i,j) = \hat{v} + d_2 \tag{3.7}$$

At this point, assume that the wrapping process works accurately and maps $V_1^F(\mu,\nu)$ into $V_{1,2}^V(2i-1, 2j-1)$ without introducing tangible wrapping distortion. This assumption implies that the depth information is accurate, in this case:

$$V_{1,2}^V(2i-1, 2j-1) = \hat{v} + d_1 \tag{3.8}$$

[2] Although the coordinate system in the two views are related, they are different due to the fact that the two views are with different resolutions.

If at the decoder side the pixel at position $(2i-1, 2j-1)$ in the zero-filled view is replaced by the average of $V_2^Z(2i-1, 2j-1)$ and $V_{1,2}^V(2i-1, 2j-1)$ then the expected compression distortion could be evaluated by using (3.12) and (3.13) as:

$$\begin{aligned} \sigma^2 &= E\left\{\left(\frac{\hat{v}+d_2+\hat{v}+d_1}{2} - \hat{v}\right)^2\right\} \\ &= E\left\{\left(\frac{d_2+d_1}{2}\right)^2\right\} \end{aligned} \tag{3.9}$$

Since view 1 and 2 are separately compressed, $E\{d_1 d_2\} = 0$. In the general case $d_1 \leq d_2$ even when using the same QP for the LR and FR views, consequently, $\frac{\sigma_2^2}{4} \leq \sigma^2 \leq \frac{\sigma_2^2}{2}$. This means that the equivalent distortion of the pixels at (2i-1, 2j-1), where $1 \leq i \leq H/2$ and $1 \leq j \leq W/2$ in the zero-filled view will be reduced.

It is worth noting that the averaging process can only be applied to those pixels in $\mathbf{V}_2^Z$ which have equivalent pixels in $\mathbf{V}_{1,2}^V$, thus holes and occluded areas need to be excluded from this process. To ensure this, the similarity and smoothness check mechanism proposed in section 3.2.1 will also be used here. Since the sliding window used in this process moves in a raster scan fashion, then except for some border pixels, each LR pixel will appear in four different windows. Therefore, only when the LR pixel is regarded similar to its counterpart virtual view pixel in four measurements, then it will be replaced by $\frac{V_2^Z(2i-1,2j-1)+V_{1,2}^V(2i-1,2j-1)}{2}$.

3.3 Thresholds Evaluation

In the proposed SR approach, both the virtual view and interpolated view are utilized to generate the FR frames, and two post-processing enhancement operations are exploited to further improve the quality of the generated FR view. In this whole process, three thresholds are required. An experimental-based approach to determine the values of these thresholds could be used at the encoder side by using an analysis-by-synthesis approach. Since these three thresholds are intertwined, the choice of one will have impacts on the others. Therefore, to obtain the best thresholds the encoder needs to test different combinations of them using three nestlike loops, and then send values to the decoder. If the complexity of estimating one threshold is $\mathcal{O}(n)$, then the complexity of this exhaustive approach is $\mathcal{O}(n^3)$. A simplified approach is proposed, where the value for each threshold is obtained in a successive approach and the complexity can be consequently, reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$. Some experiments were conducted on the "Doorflower" and "Pantomime" sequences at $QP = \{22, 27, 32, 37, 42, 47\}$ to compare

the performance of exhaustive and successive approaches; the corresponding results are shown in Fig.3.5 [3]. The results indicate that the proposed simplified approach can significantly reduce complexity without large quality degradation and also indicate that although these three thresholds are intertwined, there are some other factors that have more influence on their values. It is shown in 3.6 that T_{si} and T_l could be represented by

$$T_{si} = \alpha\sqrt{\sigma_1^2 + \sigma_2^2} \tag{3.10}$$

$$T_l = \beta\sqrt{\sigma_1^2 + \sigma_2^2} \tag{3.11}$$

where σ_1 and σ_2 are the standard deviations of the compression distortion affecting view 1 (FR view) and view 2 (LR view), respectively; α and β are two parameters which depend on the sequence content.

In order to explain the process of deriving thresholds T_{si} and T_l, it should be clear that the approach used stems from the idea that the LR and FR frames are affected by compression distortion, and consequently the thresholds should take this distortion into account. To derive the threshold T_{si} which is used to qualitatively indicate the local texture similarity, let us suppose that a point is projected into a FR viewpoint and LR viewpoint as $\hat{V}_1^F(\mu,\nu)$ and $\hat{V}_2^L(i,j)$, respectively. As we did in Section 3.2.2, let us suppose that the previous two values could be regarded similar $\hat{V}_1^F(\mu,\nu) = \hat{V}_2^L(i,j) = \hat{v}$, which means the original point is not occluded in any of the two viewing points. Then after compression $\hat{V}_1^F(\mu,\nu)$ and $\hat{V}_2^L(i,j)$ will become: $V_1^F(\mu,\nu) = \hat{v} + d_1$ and $V_2^L(i,j) = \hat{v} + d_2$, where d_1 and d_2 are distortions caused by video compression on view 1 and 2, respectively. The mean and variance of d_1 and d_2 are $E\{d_1\} = 0$, $\sigma_1^2 = E\{d_1^2\}$ and $E\{d_2\} = 0$, $\sigma_2^2 = E\{d_2^2\}$, respectively. At the decoder side, the zero-filled view is obtained from the LR view by inserting zeros in between its pixels, thus:

$$V_2^Z(2i-1, 2j-1) = V_2^L(i,j) = \hat{v} + d_2 \tag{3.12}$$

Assuming that the wrapping process works accurately and maps $V_1^F(\mu,\nu)$ into $V_{1,2}^V(2i-1, 2j-1)$ without introducing tangible wrapping distortion. In this case:

$$V_{1,2}^V(2i-1, 2j-1) = \hat{v} + d_1 \tag{3.13}$$

[3]Similar results, not reported here for briefness, have been obtained from other sequences.

Therefore, at this stage the variance of the difference between $V_{1,2}^V$ and V_2^Z, which will be used to measure the local texture similarity, could be evaluated as:

$$\begin{aligned} \sigma_d^2 &= E\left\{\left(V_{1,2}^V(2i-1,2j-1) - V_2^Z(2i-1,2j-1)\right)^2\right\} \\ &= E\left\{(d_1-d_2)^2\right\} \end{aligned} \tag{3.14}$$

Due to the fact that d_1 and d_2 are uncorrelated, $E\left\{(d_1-d_2)^2\right\} = E\{\sigma_1^2\} + E\{\sigma_2^2\}$. So when measuring the local similarity, the threshold T_{si} should be selected to mask the distortion induced dissimilarity, thus

$$T_{si} = \alpha \times \sigma_d = \alpha\sqrt{\sigma_1^2 + \sigma_2^2} \tag{3.15}$$

where α is a parameter which depends on the sequence content.

The derivation of the luminance compensation threshold, T_l, follows a similar approach to the one used for T_{si}. The luminance compensation process is carried forward for the virtual-view-based recovered pixels. For this to happen it requires that the similarity condition between W^Z and W^V be satisfied. Thus, we could use the same approach we used to evaluate σ_d^2 in (3.14) to evaluate the variance of ΔY^h as:

$$\sigma_h^2 = \frac{\sigma_1^2 + \sigma_2^2}{2} \tag{3.16}$$

For the vertical compensation item ΔY^v the variance could be evaluated as

$$\sigma_v^2 = \frac{\sigma_1^2 + \sigma_2^2}{2} \tag{3.17}$$

Finally, for the center compensation item ΔY^c we have

$$\sigma_c^2 = \frac{\sigma_1^2 + \sigma_2^2}{4} \tag{3.18}$$

Hence, if we want to use threshold T_l to ensure that mainly luminance differences get compensated and not the differences due to compression of the two views, then T_l should be selected to be larger than σ_h, σ_v and σ_c.

Therefore,

$$T_l = \beta\sqrt{\sigma_1^2 + \sigma_2^2} \tag{3.19}$$

where β is a factor that depends on the sequence.

In the successive approach, the three thresholds(or equivalently, α, β and T_{sm}) could be either determined at the encoder frame-by-frame and sent to the decoder side, or be just determined for the first frame and then applied to the following frames. These two

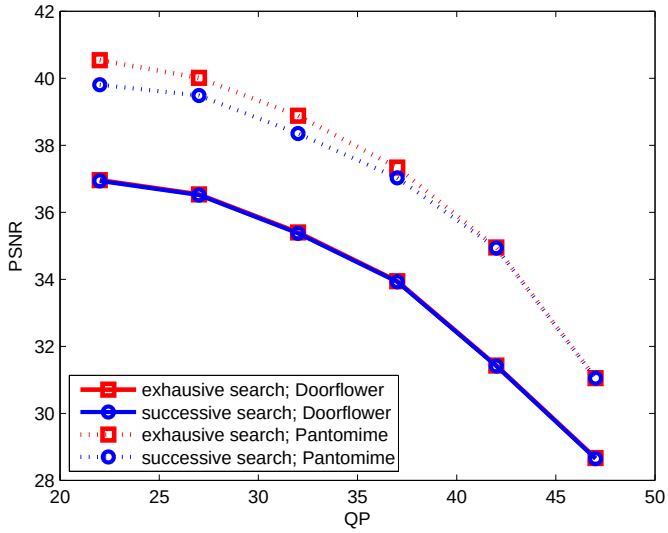


Figure 3.5: The comparison of exhaustive and successive approaches for thresholds determination on "Doorflower" and "Pantomime" sequences.

approaches have been tested and the results are shown in Fig.3.6. In addition, there is a third approach which uses user-defined value for both α and β; the corresponding results are also shown in Fig.3.6. In this approach, it is reasonable to assume α and β are larger than three based on Chebyshev's inequality. From the figure, it is obvious that the successive-search approach which estimates α, β and T_{sm} based on the first frame and then use these values for the following frames is almost as good as the frame-by-frame approach, and consequently, all the following experiments were conducted using this approach.

3.4 Proposed Method on Multiview Video

The proposed virtual view assisted SR algorithm can also be applied to multiview multi-resolution systems. Since in these kinds of systems more neighboring FR views and the corresponding depth maps are available, at a given viewpoint, more virtual view versions can be utilized. With the aid of these virtual views, the quality of the final generated FR views can be considerably improved. As depicted in Fig.3.7, $V^V_{q,k}$ ($q = 1, ..., m$ and $k = 1, ..., n$) is the virtual view generated from one of the adjacent FR views at viewpoint q to one of the LR views at viewpoint k . In this case, the zero-filled pixels in the zero-filled view are replaced by selecting from the available virtual views the one which best satisfies the similarity condition. Subsequently, the two enhancement

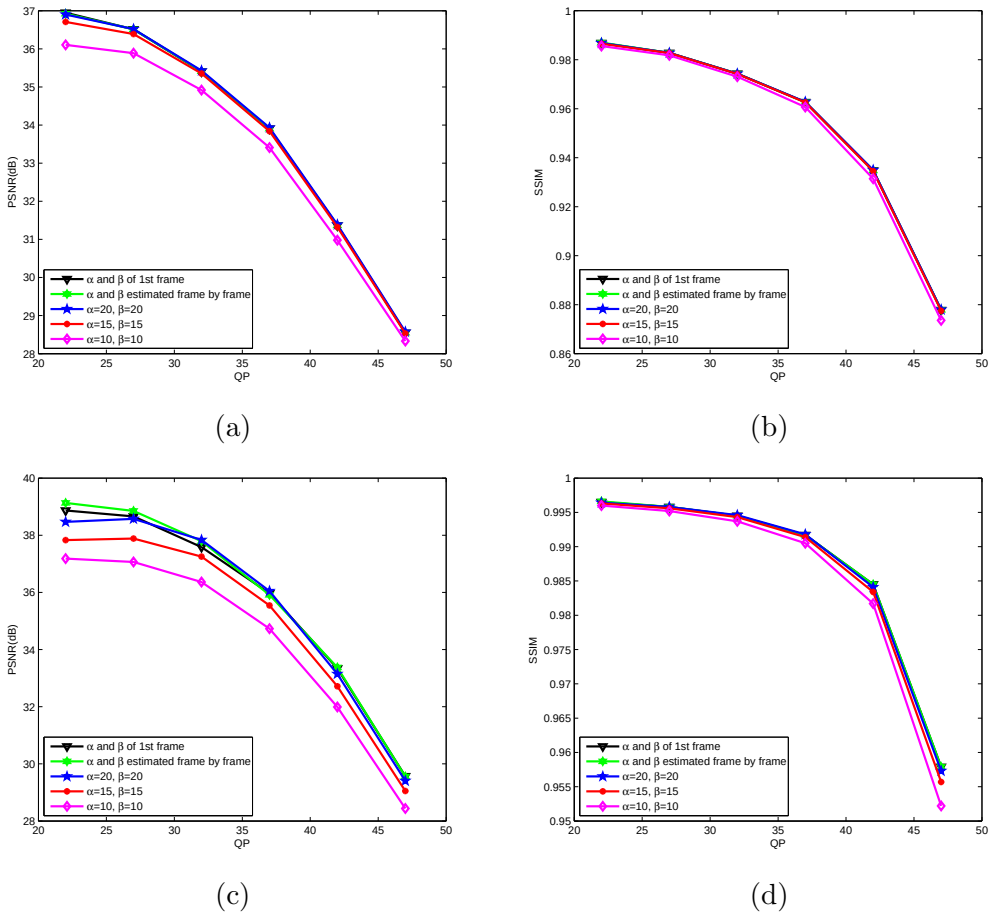


Figure 3.6: PSNR and SSIM comparisons of different approaches for the evaluation of α and β; (a) and (b) are results of "Doorflower"; (c) and (d) are results of "Pantomime".

methods are performed step-by-step, in this way, the proposed algorithm effectively super-resolves the LR views.

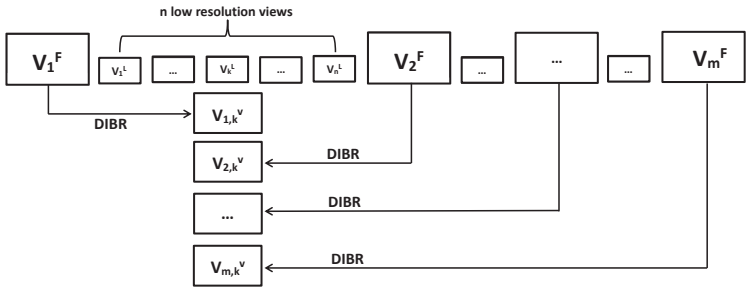


Figure 3.7: The proposed algorithm for multiview multi-resolution system.

Table 3.2: The parameters and characteristics for each used sequence

Name	Size	FR	LR	Content's Motion	Frame rate (fps)
Doorflower	1024×768	View10	View08	Moderate	16.67
Bookarrival	1024×768	View08	View06	Moderate	16.67
Leavelaptop	1024×768	View06	View07	Moderate	16.67
Pantomime	1280×960	View39	View40	Medium complex	30
Champagne	1280×960	View37	View38	Complex	30
Dog	1280×960	View38	View39	Medium complex	30
Kendo	1024×768	View03	View04	Complex	30

3.5 Experimental Results

In order to objectively and subjectively evaluate the performance of the proposed algorithm, several experiments were conducted with MPEG 3-D video sequences, including "*Doorflower*", "*Bookarrival*", "*Leavelaptop*", "*Pantomime*", "*Champagne*", "*Dog*", and "*Kendo*". Since the proposed SR method is targeted for MR paradigm, and due to the lack of MR MVD sequences. Test sequences have been generated by downsampling at least one of the FR views to LR for each of the sequences. The downsampling factor is 2 in both the horizontal and vertical directions. The original FR views are considered as ground-truth views for objective assessments. The DIBR technique is employed to render the virtual views, and the H.264/AVC reference software JM17.0 [96] is used to implement the coding process. The IPPP coding structure is used and one second of each sequence is tested. The QP values are $\{22, 27, 32, 37, 42, 47\}$ for both the texture and depth maps and for both FR and LR views. In the following experiment, the 6-tap *Lanczos* interpolation filter is used as a benchmark method. The other most used interpolation method *Bicubic* is also tested [4]. Finally, Peak Signal-to-Noise-Ratio (PSNR) and Structural Similarity Index Measurement (SSIM) [97] are employed to assess the objective performance.

In the experiments, firstly the effectiveness of the proposed approach on stereoscopic sequences is evaluated, and then its performance is compared with other approaches. Secondly, the effectiveness of each stage of the proposed method is verified, the results of each stage are also reported. Lastly, the proposed algorithm is applied to MVD

[4]These two methods enables other researchers to "indirectly" compare their works with ours by directly comparing their results with these two common interpolation methods. Moreover, the proposed approach belongs to the interpolation-based SR category, so it is reasonable to compare it with Bicubic and Lanczos which are also interpolation methods.

sequences.

3.5.1 Performance Evaluation on Stereo Video

Unless otherwise noted, the characteristics and two chosen viewpoints for each test sequence have been listed in Table 3.2. All the PSNR and SSIM results for the luminance component of the three interpolation-based approaches are shown in Table 3.3, where "Lan" = "Lanczos", "Bic"= "Bicubic", "Pro"= "Proposed". Results of the state-of-the-art single image super-resolution approach via Sparse Coding (SC) [61] are also reported in the table, where the parameters for the publicly available code[5] are set according to [61]. It is clear that the proposed method outperforms the benchmark method and Bicubic method over all QPs both in terms of PSNR and SSIM. For most of the cases, the proposed method is also better than [61]. Table 3.3 also presents the PSNR and SSIM gains over the benchmark method which are indicated by ΔPSNR and ΔSSIM, respectively. This reveals that the PSNR gains increase with a decrease of QP values, while, the SSIM gains increase with an increase of QP values. The highest PSNR gain obtained by the proposed method is 3.85dB on the sequence "*Bookarrival*" when PQ=22, while, the average PSNR gain over all sequences and QPs is 2.11dB. Although, in terms of SSIM the gains are not as obvious as the PSNR ones, the SSIM gains still indicate an improvement in the objective quality compared with the benchmark method, especially when QP is very large.

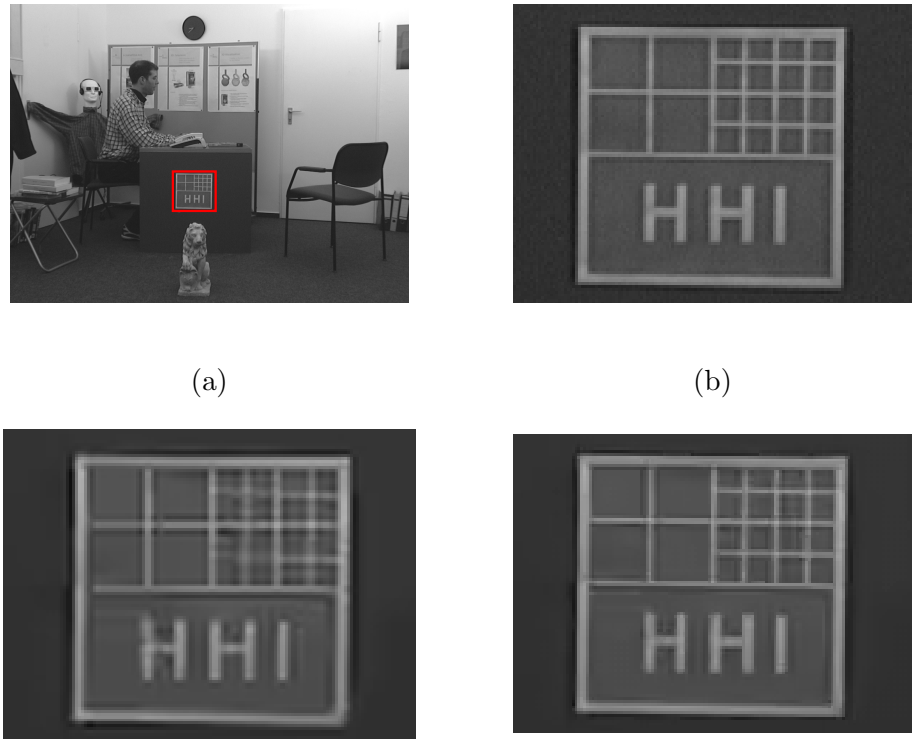
To further evaluate the effectiveness of the proposed method, comparisons with the method proposed in [1] also have been carried out by adopting the same test sequences with the same resolution and the same way of generating the test MR sequences with a resolution factor of 2. The results of these comparisons are shown in Table 3.4.

In the following we compare our proposed approach with [8]. The proposed method was tested under disadvantageous condition with respect to [8], where in the latter approach the uncompressed sequence "*Pantomime*" and "*Dog*" with a resolution factor of 2 was used, and the reported gains were 2.57dB and 1.06dB over the *Lanczos* method, respectively, while, in the proposed method, even with video compression (QP=22) the gains are 3.62dB and 1.22dB, respectively. The average PSNR gain in [8] at $QP = \{22, 27, 32, 37\}$ on these two sequences with respect to the *Lanczos* method are 1.39dB and -0.16dB, respectively. While, the average PSNR gains of the proposed method in

[5] `http://www.ifp.illinois.edu/~jyang29/ScSR.htm`

Table 3.3: The Luminance PSNR (dB) and SSIM results of proposed method in comparison with other methods and corresponding gains of the proposed method over Lanczos method

QP	Sequence		Doorflower	Bookarrival	Leavelaptop	Pantomime	Champagne	Dog	Kendo
22	PSNR	Lan	33.12	32.98	33.31	35.50	33.95	34.96	37.56
		Cub	33.20	33.05	33.40	35.54	34.00	35.04	37.60
		SC[61]	32.83	32.57	32.94	36.31	34.79	35.43	38.05
		Pro	**36.95**	**36.83**	**36.05**	**38.87**	**36.25**	**36.05**	**39.36**
	ΔPSNR		3.83	3.85	2.74	3.37	2.30	1.09	1.80
	SSIM	Lan	0.970	0.969	0.970	0.994	0.992	0.983	0.986
		Cub	0.971	0.970	0.970	0.994	0.992	0.983	0.987
		SC[61]	0.945	0.938	0.936	0.985	0.984	0.967	0.970
		Pro	**0.987**	**0.987**	**0.984**	**0.997**	**0.995**	**0.989**	**0.990**
	ΔSSIM		0.017	0.018	0.014	0.003	0.003	0.006	0.004
27	PSNR	Lan	32.89	32.68	33.03	35.21	33.62	34.27	36.85
		Cub	32.95	32.73	33.09	35.25	33.67	34.33	36.89
		SC[61]	32.76	32.44	32.82	36.10	34.47	34.83	37.49
		Pro	**36.51**	**36.23**	**35.70**	**38.66**	**35.53**	**35.10**	**38.74**
	ΔPSNR		3.62	3.55	2.67	3.45	1.91	0.83	1.89
	SSIM	Lan	0.966	0.963	0.964	0.993	0.990	0.976	0.981
		Cub	0.967	0.963	0.964	0.993	0.990	0.976	0.981
		SC[61]	0.940	0.930	0.928	0.982	0.980	0.955	0.962
		Pro	**0.983**	**0.981**	**0.979**	**0.996**	**0.993**	**0.982**	**0.986**
	ΔSSIM		0.017	0.018	0.015	0.003	0.003	0.006	0.005
32	PSNR	Lan	32.33	32.04	32.45	34.48	32.84	32.93	35.48
		Cub	32.37	32.08	32.49	34.51	32.89	32.99	35.51
		SC[61]	32.47	32.08	32.55	35.48	33.71	33.61	36.26
		Pro	**35.41**	**35.12**	**34.81**	**37.59**	**35.00**	**33.49**	**37.45**
	ΔPSNR		3.08	3.08	2.36	3.11	2.16	0.56	1.97
	SSIM	Lan	0.957	0.951	0.953	0.991	0.985	0.961	0.972
		Cub	0.957	0.951	0.953	0.991	0.985	0.961	0.972
		SC[61]	0.931	0.916	0.917	0.977	0.972	0.937	0.950
		Pro	**0.974**	**0.972**	**0.970**	**0.995**	**0.991**	**0.967**	**0.980**
	ΔSSIM		0.017	0.021	0.017	0.004	0.006	0.006	0.008
37	PSNR	Lan	31.26	30.89	31.26	33.07	31.56	31.04	33.48
		Cub	31.30	30.92	31.30	33.12	31.62	31.10	33.51
		SC[61]	31.67	31.29	31.66	34.18	32.41	**31.89**	34.37
		Pro	**33.91**	**33.49**	**33.40**	**35.97**	**33.33**	31.36	**35.25**
	ΔPSNR		2.65	2.60	2.14	2.90	1.77	0.32	1.77
	SSIM	Lan	0.941	0.930	0.931	0.986	0.977	0.931	0.956
		Cub	0.941	0.930	0.931	0.986	0.977	0.931	0.956
		SC[61]	0.915	0.897	0.896	0.967	0.957	0.905	0.932
		Pro	**0.963**	**0.958**	**0.956**	**0.992**	**0.983**	**0.937**	**0.969**
	ΔSSIM		0.022	0.028	0.025	0.006	0.006	0.006	0.013
42	PSNR	Lan	29.46	28.99	29.49	30.74	29.50	28.42	30.87
		Cub	29.49	29.01	29.51	30.78	29.56	28.45	30.89
		SC[61]	30.35	29.92	30.41	31.95	30.55	**29.48**	31.86
		Pro	**31.36**	**31.08**	**31.31**	**33.34**	**31.05**	28.62	**32.51**
	ΔPSNR		1.90	2.09	1.82	2.60	1.55	0.20	1.64
	SSIM	Lan	0.904	0.882	0.888	0.975	0.960	0.859	0.927
		Cub	0.904	0.882	0.888	0.975	0.960	0.859	0.927
		SC[61]	0.882	0.854	0.859	0.955	0.942	0.834	0.901
		Pro	**0.935**	**0.923**	**0.924**	**0.985**	**0.971**	**0.866**	**0.949**
	ΔSSIM		0.031	0.041	0.036	0.010	0.011	0.007	0.022
47	PSNR	Lan	27.29	26.83	27.17	27.50	26.90	25.80	27.86
		Cub	27.30	26.84	27.18	27.53	26.96	25.82	27.88
		SC[61]	28.50	28.02	28.40	28.83	28.07	**27.00**	28.94
		Pro	**28.55**	**28.36**	**28.57**	**29.57**	**28.25**	25.91	**29.55**
	ΔPSNR		1.26	1.53	1.40	2.07	1.35	0.11	1.69
	SSIM	Lan	0.838	0.811	0.817	0.939	0.922	0.732	0.877
		Cub	0.838	0.811	0.817	0.939	0.922	0.732	0.878
		SC[61]	0.820	0.790	0.796	0.922	0.900	0.712	0.853
		Pro	**0.878**	**0.862**	**0.863**	**0.958**	**0.941**	**0.740**	**0.912**
	ΔSSIM		0.040	0.051	0.046	0.019	0.019	0.008	0.035



(a) (b)

(c) (d)

Figure 3.8: (a) the reference FR frame; (b) cropped portion of the FR frame; the results at QP=32 for: (c) benchmark interpolation method; (d) proposed method; full resolution of the cropped portion is 620×884.

comparison with the *Lanczos* method at the same QP values are 3.21dB and 0.74dB, respectively.

The subjective comparisons are shown in Fig.3.8. The reference FR frame at $QP = 32$ is shown in Fig.3.8 (a), whereas, Fig.3.8 (b) shows a cropped portion of it, the same areas processed by the benchmark and the proposed method are shown in Fig.3.8 (c) and Fig.3.8 (d), respectively. In contrast to the *Lanczos* method, our method preserves the edges and obtains a satisfactory result, due to the elimination of the aliasing artifacts and blurring caused by only adopting the interpolation process.

3.5.2 Performance of Each Stage of the Proposed Method

In this subsection, several experiments have been conducted to validate the necessity and effectiveness of each stage in the proposed algorithm. Hence, the PSNR and SSIM improvements of each stage are listed in Table 3.5 for the zero-filled view filling stage and the enhancement stage, these two stages will be respectively denoted as "zfvf" and

Table 3.4: PSNR (dB) and SSIM of SR results obtained by the proposed method and reference method in[1]

Sequence		Book	Doorflower	Laptop
PSNR	[1]	33.04	33.27	33.93
	Pro	**34.57**	**33.76**	**34.38**
ΔPSNR		1.53	0.49	0.45
SSIM	[1]	0.941	0.941	0.945
	Pro	**0.986**	**0.983**	**0.986**
ΔSSIM		0.045	0.042	0.041

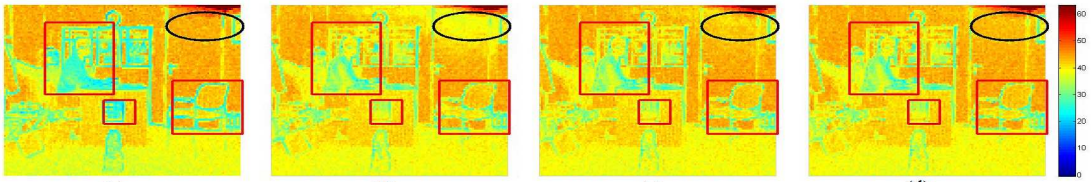


Figure 3.9: The PSNR value for each 8×8 block evaluated on the luminance component of the first frame of "Doorflower" (shown in Fig.3.8 (a)) at QP=22. (a) benchmark interpolation method; proposed method: (b) after similarity check, (c) after smoothness check, and (d) after enhancement stage.

"zfve" for short. As can be seen from the table, each stage contributes some gains, except for some very small losses. For the majority of the tested sequences, the first stage provides significant gains, nevertheless, the second stage also offers contributions by doing local improvement, when QP is large. Similarly, the first stage provides more SSIM gains than the second stage, except for in the sequence "*Champagne*".

To further investigate the effectiveness of each stage the sequence "*Doorflower*" is taken as an example in Fig.3.9. The PSNR value for each 8×8 block has been shown for the *Lanczos* method in Fig.3.9 (a) (the original tested frame is shown in Fig.3.8 (a)). It is worth noticing that it has a PSNR value over 50dB in the top-right corner of the frame and also high PSNR values in smooth areas. However, areas with complex texture and edges suffer low PSNR values, some of these areas are indicated by red squares in the figure. This observation emphases the importance of recovering high frequency information and the weakness of approaches that solely rely on the LR pixels to generate the FR frame. By referring to Fig.3.9 (b) which shows the PSNR distribution after replacing the zero-filled pixels in the zero-filled view, it can be seen that the parts, especially in the highlighted red squared areas, the edges and areas with

Table 3.5: The Luminance PSNR gain (dB) and SSIM gain for each stage of the proposed approach;"zfvf" and "zfve" stand for zero-filled view filling stage and the enhancement stage, respectively

QP			22	27	32	37	42	47
Doorflower	ΔPSNR	zfvf	3.28	3.08	2.58	2.16	1.52	0.95
		zfve	0.55	0.55	0.50	0.49	0.38	0.31
	ΔSSIM	zfvf	0.016	0.017	0.018	0.023	0.034	0.044
		zfve	0.001	0.000	0.000	-0.001	-0.004	-0.005
Bookarrival	ΔPSNR	zfvf	3.04	2.81	2.44	2.06	1.68	1.27
		zfve	0.81	0.74	0.64	0.55	0.40	0.27
	ΔSSIM	zfvf	0.017	0.018	0.022	0.030	0.047	0.060
		zfve	0.001	0.000	-0.001	-0.002	-0.006	-0.009
Leavelaptop	ΔPSNR	zfvf	1.83	1.74	1.55	1.41	1.28	1.10
		zfve	0.90	0.94	0.81	0.74	0.54	0.30
	ΔSSIM	zfvf	0.013	0.014	0.017	0.025	0.040	0.053
		zfve	0.002	0.001	0.001	0.000	-0.004	-0.007
Pantomime	ΔPSNR	zfvf	3.40	3.46	3.09	2.87	2.61	2.08
		zfve	1.13	1.24	1.17	1.15	1.02	0.68
	ΔSSIM	zfvf	0.002	0.002	0.003	0.006	0.010	0.019
		zfve	0.000	0.000	0.000	0.000	0.000	-0.001
Champagne	ΔPSNR	zfvf	1.16	0.97	1.04	0.84	0.68	0.78
		zfve	1.14	0.95	1.12	0.94	0.88	0.58
	ΔSSIM	zfvf	0.000	0.000	0.003	0.004	0.005	0.023
		zfve	0.004	0.003	0.003	0.003	0.006	-0.003
Dog	ΔPSNR	zfvf	0.82	0.64	0.41	0.23	0.10	0.03
		zfve	0.28	0.20	0.15	0.09	0.10	0.08
	ΔSSIM	zfvf	0.006	0.006	0.006	0.006	0.007	0.014
		zfve	0.000	0.000	0.000	0.000	-0.001	-0.005
Kendo	ΔPSNR	zfvf	0.81	0.69	0.69	0.87	1.03	1.16
		zfve	1.00	1.20	1.28	0.89	0.61	0.53
	ΔSSIM	zfvf	0.002	0.003	0.007	0.012	0.022	0.033
		zfve	0.002	0.002	0.002	0.001	0.000	0.002

complex texture have been improved significantly. However, at the same time the area highlighted by the black ellipse, which is a flat background area in the scene, endures quality degradation due to this process. This indicates that if most of the pixels are copied from the virtual view, the information around edges and strong details can be recovered well, but the flat area might be degraded with respect to the interpolated frame. Hence, there is an overall trade-off between these two choices. The FR frame after the smoothness check is shown in Fig.3.9 (c), from this figure we could appreciate how the quality of flat areas is improved (the highlighted ellipse in Fig.3.9 (c)). Actually, the area highlighted by the ellipse in the scene is closer to the light source, so it has a higher possibility of being affected by the imbalanced light distribution. Therefore, improvement is achieved after the luminance compensation process, as shown in Fig.3.9 (d), and the already improved areas in the previous stage are still preserved well.

3.5.3 Performance Evaluation on Multiview Video

When testing on multiview video, for "*Doorflower*", the LR version of View10 is super-resolved with the aid of View12 and View08. For "*Pantomime*", the LR version of View40 is super-resolved with the aid of View39 and View41. For "*Champagne*", the LR version of View38 is super-resolved with the aid of View37 and View39. For "*Dog*", the LR version of View39 is super-resolved with the aid of View38 and View40. For "*Kendo*", the LR version of View04 is super-resolved with the aid of View03 and View05. Table 3.6 reveals all the results of these simulations and it shows that the proposed SR method can also work well in multiview video system. In this case, the highest PSNR gain can be up to 4.6dB for "*Pantomime*" sequence. Compared with the two-view video case, the PSNR gains of multiview video become higher especially when the QP is small (QP=22), and the average gain over the tested sequences for all QPs is 2.16dB which is 0.20dB higher than the obtained results for stereoscopic video. These gains are obtained due to the availability of multiple virtual view candidates, which ensures that the more suitable virtual view pixels are copied into the zero-filled view.

3.6 Conclusions

In this chapter, a novel interpolation-based virtual view assisted super-resolution method for mixed-resolution multiview video has been proposed. The low resolution views in

Table 3.6: The Luminance PSNR (dB) and SSIM values and gains over the benchmark method for multiview video

Doorflower							
QP		22	27	32	37	42	47
PSNR	Lan	33.40	33.15	32.55	31.46	29.62	27.43
	Pro	**37.48**	**36.89**	**35.62**	**33.80**	**31.49**	**28.70**
ΔPSNR		4.08	3.74	3.07	2.34	1.87	1.27
SSIM	Lan	0.972	0.967	0.957	0.941	0.904	0.840
	Pro	**0.988**	**0.982**	**0.973**	**0.960**	**0.932**	**0.876**
ΔSSIM		0.015	0.015	0.016	0.019	0.028	0.036
Pantomime							
QP		22	27	32	37	42	47
PSNR	Lan	35.56	35.27	34.53	33.13	30.80	27.54
	Pro	**40.16**	**39.71**	**38.11**	**35.41**	**31.91**	**28.00**
ΔPSNR		4.60	4.44	3.58	2.28	1.11	0.46
SSIM	Lan	0.994	0.993	0.991	0.986	0.975	0.939
	Pro	**0.997**	**0.996**	**0.994**	**0.989**	**0.978**	**0.942**
ΔSSIM		0.003	0.003	0.003	0.003	0.003	0.003
Champagne							
QP		22	27	32	37	42	47
PSNR	Lan	34.01	33.67	32.90	31.62	29.55	26.95
	Pro	**37.75**	**36.88**	**35.48**	**33.17**	**30.42**	**27.31**
ΔPSNR		3.74	3.21	2.58	1.55	0.87	0.36
SSIM	Lan	0.992	0.990	0.985	0.977	0.961	0.923
	Pro	**0.996**	**0.994**	**0.989**	**0.981**	**0.965**	**0.926**
ΔSSIM		0.004	0.004	0.004	0.004	0.004	0.003
Dog							
QP		22	27	32	37	42	47
PSNR	Lan	34.99	34.29	32.95	31.07	28.45	25.82
	Pro	**36.81**	**35.77**	**33.97**	**31.71**	**28.76**	**26.00**
ΔPSNR		1.82	1.48	1.02	0.64	0.31	0.18
SSIM	Lan	0.983	0.977	0.962	0.933	0.862	0.733
	Pro	**0.991**	**0.984**	**0.970**	**0.940**	**0.869**	**0.743**
ΔSSIM		0.008	0.007	0.007	0.007	0.007	0.010
Kendo							
QP		22	27	32	37	42	47
PSNR	Lan	37.56	36.85	35.48	33.49	30.87	27.86
	Pro	**41.05**	**39.90**	**38.01**	**35.58**	**32.58**	**29.35**
ΔPSNR		3.49	3.05	2.53	2.09	1.71	1.49
SSIM	Lan	0.986	0.981	0.972	0.956	0.927	0.877
	Pro	**0.990**	**0.986**	**0.980**	**0.968**	**0.946**	**0.907**
ΔSSIM		0.004	0.005	0.008	0.012	0.020	0.030

the MR multiview video are super-resolved to full resolution size using a two-step process. In the first stage, the similarity between the LR pixels and their counterparts in the virtual view is measured. Then if necessary, a smoothness check will be carried out to determine whether to use virtual view pixels or interpolated pixels to fill the zero-

filled pixels. Subsequently, the quality of the virtual-view-based pixels is enhanced by compensating the intrinsic luminance difference between the two views. Furthermore, the inter-view correlation is exploited to enhance the LR pixels in the super-resolved frame by reducing their compression distortion. Therefore, different from the previous interpolation-based SR algorithms, the advantages of virtual views have been exploited by the proposed method at different stages. Moreover, it has been shown that the proposed algorithm achieves superior performance with respect to both benchmark and state-of-the-art approaches. Future work will be devoted to combining temporal correlations with inter-view correlation to improve the exploitation of the virtual views.

It is worth reporting that the work reported in this section has led to the following publication: Zhi Jin, Tammam Tillo, Jimin Xiao, Chao Yao and Yao Zhao, Virtual View Assisted Video Super-Resolution and Enhancement, Circuits and Systems for Video Technology, IEEE Transactions on (Volume:PP , Issue: 99), doi:10.1109/TCSVT.2015.2412791.

Chapter 4

Depth-Map-Assisted Texture Super-Resolution for Multiview Video Plus Depth

4.1 Introduction

The MVD format [98], as one popular representation format for 3D multiview data, consists of textures and the associated per-pixel depth data. Although this format allows any intermediate view within a certain range to be generated, referring to 2D video systems, the required transmitted data of 3D multiview video is still very large.

Compared with texture, depth maps require less transmission bitrate [99], however, the quality of the decoded depth maps highly affects the quality of the DIBR generated synthesized views. Hence, much research has only focused on reducing the amount of transmitted texture data. In [100], Garcia *et al.* proposed a mixed-resolution-based coding approach where all of the frames were divided into groups of N FR frames and M LR frames. In each group, the smallest encoding resolution for the M LR frames was obtained by iteratively comparing the N-th reconstructed FR frame with its original version. Since the best downsampling ratio was obtained based on only one FR frame and then applied to the remaining LR frames, the obtained results may not be optimal. Moreover, if the resolution estimation analysis was done on a large scale, the computational complexity would hugely increase. While, Zhang *et al.* in [9] proposed a content adaptive downsampling on both views of a stereoscopic video. However, the downsampling mechanism needs information regarding the interpolation mechanism used.

One common problem of downsampling-based coding approaches is that they gen-

erally improve the rate-distortion performance at low bitrate. However, it may result in subjective quality degradation. In order to overcome this quality degradation, such as, block artifacts, blurred details and ringing artifacts around the edges [90], some sophisticated edge-guided upsampling and super-resolution methods have been proposed. In [101], a sharp HR gradient field was obtained from the LR image by an adaptive self-interpolation algorithm, and then this HR gradient was used as an additional edge-preserving constraint to refine the FR image. In [8] and [102], virtual views have been utilized to recover FR frames in a MR-MVD framework, so in [8] the high frequency content has been extracted from the virtual view and then added to the LR frame to reconstruct the FR frame. However, in this work, the high frequency content is extracted from the whole frame, thus the local characteristics of the scene are not taken into account. While, in [102], the local similarity between the LR frames and their corresponding virtual view has been used to steer the FR recovery mechanism. Consequently, for similar areas, virtual view pixels are used to generate the FR frame, whereas, for non-similar areas a conventional interpolation method is used. The main weakness of this approach is that it does not provide a mechanism to jointly fuse these two kinds of pixel. Different from the previously described paradigms, in [103] and [9], an optimized down/upsampling framework is proposed. In this approach, in order to minimize interpolation errors, the image downsampling pattern is evaluated as a function of the interpolation method. Unfortunately, this paradigm is not suitable for video applications, because the downsampling pattern is frame dependent which means that temporal redundancy cannot be efficiently removed by the video encoder. Thus, for video applications there is a need to use temporally static downsampling patterns.

In this chapter a *systematical* framework to downsample and upsample MVD data is proposed. In the proposed downsampling approach, the rows of two adjacent texture views are downsampled following an interlacing and complementary pattern, before compression. The aim of this downsampling approach is to facilitate the upsampling at the decoder side, where the LR views will be upsampled by fusing the virtual view pixels with directional interpolated pixels with the aid of pattern direction of the discarded pixels. This approach has two benefits. Firstly, the high frequency information contained in counterpart LR view can be properly utilized to upsample the other LR view through the generated virtual views. Secondly, since the virtual view quality depends on many factors, e.g. DIBR technique and depth map quality, it generally has

low quality on depth discontinuous areas, where on the other hand, directional interpolation approaches can work well. Hence, by taking advantages of these two strategies, the discarded pixels can be recovered efficiently. Experimental results have shown that the proposed algorithm achieves superior performance with respect to the filter-based interpolation algorithms and other state-of-the-art algorithms. The proposed upsampling approach is named Directional Data Fusion Upsampling (DDFU) throughout this chapter.

The rest of this chapter is organized as follows. Section 4.2 describes the details of the proposed down/upsampling algorithm. Experimental results are presented in Section 4.3 and the conclusion is in Section 4.4.

4.2 Proposed Down/Upsampling Paradigm

A proper downsampling approach for multiview video needs to take into account the fact that different views cover almost the same scene, with a considerable amount of inter-view redundancy. Thus, in this work, by taking this feature into account, an interlacing-and-complementary-row-downsampling method is proposed, as shown in Fig.4.1. The benefits of this downsampling approach will be explained in the following section.

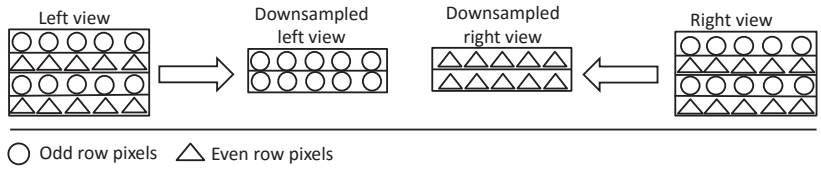


Figure 4.1: The proposed interlacing-and-complementary-row-downsampling process for a stereo video.

4.2.1 Interlacing and Complementary Row Downsampling

Due to inter-view redundancy, interlacing-and-complementary downsampling approaches could maintain more information than the non-interlacing-and-complementary ones. Hence, in the following sections and aided with a graphical example, three downsampling approaches will be compared with the assumption that two cameras in a parallel configuration setting are used to record an uneven bars structure (similar to the artistic gymnastics apparatus), as shown in Fig.4.2 (d). Fig.4.2 (a), (b) and (c) show the front, side, and top view of the stereoscopic orthographic projection of the scene. The

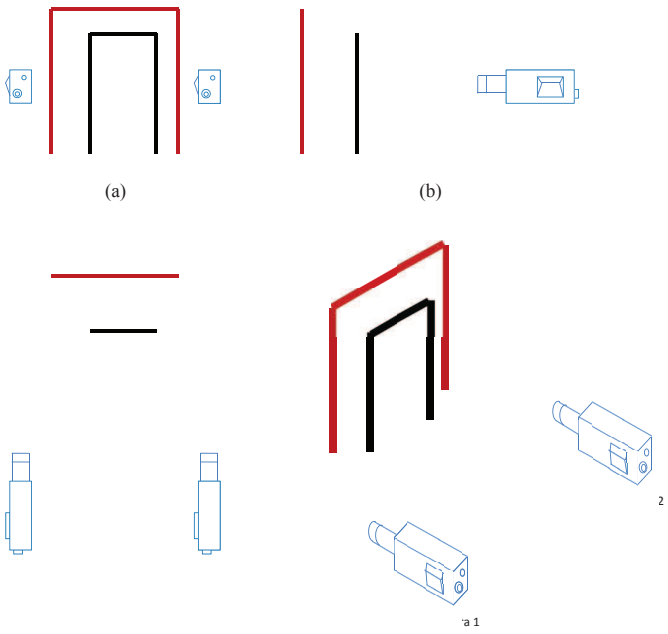


Figure 4.2: (a), (b) and (c) show the front, side, and top view of the stereoscopic orthographic projection of uneven bars structure viewed by two cameras in a parallel configuration setting, as shown in (d).

viewed scene of the first and second cameras is shown in Fig.4.3 (a). The output of the vertical interlacing-and-complementary downsampling approach (i.e., column-wise downsampling) is shown in Fig.4.3 (b), where the grey areas indicate the "discarded areas" during downsampling process. It is possible to see that the left black bar of the uneven bars structure is missing in both views. Hence, neither intra-view or inter-view interpolation can help to recover this part. This happens because the column-wise downsampling approach causes some "blind areas", where objects can not be seen in any of the two views. Referring to Fig.4.4, the top view of the prospective projection of a scene with two pinhole cameras, the area enclosed by red lines could be viewed by both cameras. Whereas, the yellow and blue bands indicate discarded areas in view 1 and view 2, respectively, due to the column-wise downsampling. Some areas (indicated by black) inevitably end up being discarded in both views, thus any object falling in any of these areas cannot be recovered by inter-view interpolation and consequently, these areas are called "blind areas".

Compared with column-wise downsampling, the output of row-wise downsampling is shown in Fig.4.3 (c). It indicates that the interlacing-and-complementary-row-downsampling will always guarantee that an object could be seen in the rows of one

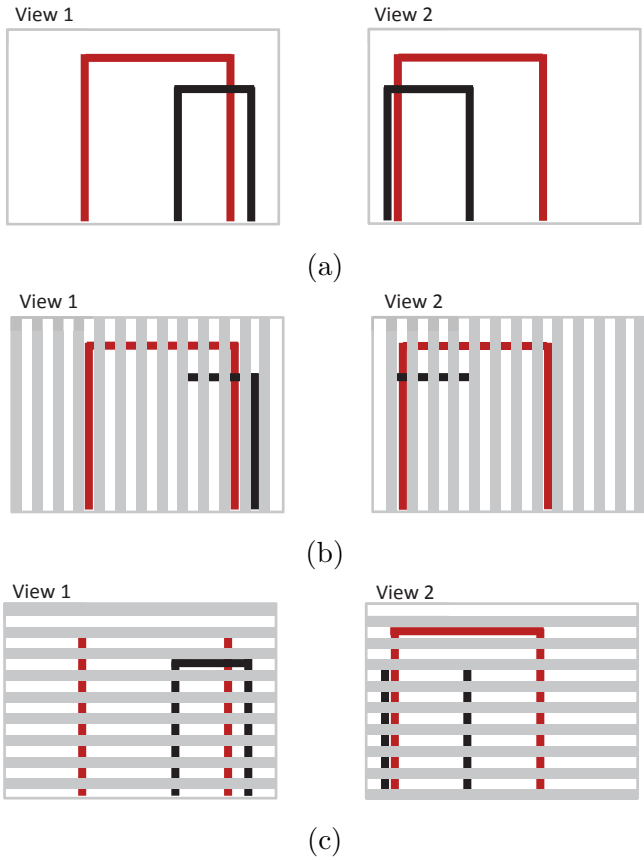


Figure 4.3: (a) the left side and right side of each frame shown the captured scene by the corresponding cameras, respectively; (b) the output of the vertical downsampling approach (i.e., column-wise downsampling); (c) the output of the interlacing and complementary row-wise downsampling.

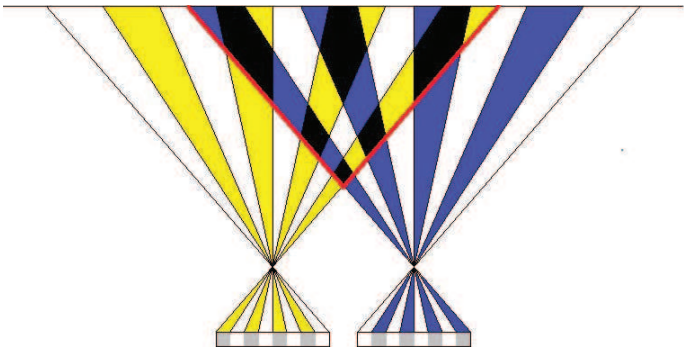


Figure 4.4: The top view of the prospective projection of a scene using a pinhole camera model for the column-wise downsampling approach.

of the two views, except for some small objects with a one-pixel-width projection size in the camera plane, these might not be captured in either of the two views. Nevertheless, the probability of this happening is low and could similarly happen for the column-wise downsampling approach. Moreover, row-wise downsampling could better exploit the warping feature of the DIBR technique and consequently, could enhance the upsampling performance.

The chessboard downsampling approach can be regarded as the combination of row- and column-wise downsampling. It is able to achieve the highest upsampling performance, since each to-be-filled pixel has four adjacent pixels in both horizontal and vertical directions which provide more information during interpolation. However, the chessboard pattern usually requires a comparatively higher bitrate due to low spatial and temporal correlations. Furthermore, for each row of the chessboard downsampled views, it is possible to notice that the top view of prospective projection of a scene is similar to the one shown in Fig.4.4. Therefore, it could be conjectured that the chessboard downsampling approach also suffers from "blind areas", thus its performance is better than the column-wise approach while being worse than the row-wise approach.

4.2.2 Virtual View-assisted Directional Data Fusion Upsampling

In order to reduce the required resources and the amount of transmitted data, downsampling of the texture sequences is performed before the compression stage. In this chapter, motivated by the findings in Section 4.2.1, the downsampled texture frames are generated by discarding the even rows in the left view and the odd rows in the right view of the stereo video, respectively. That is to say, after downsampling, the left view only has odd rows and the right view only has even rows. Let the left and right FR frames be defined as $\mathbf{V}_f^l$ and $\mathbf{V}_f^r$, respectively, with size $W \times H$, and the downsampled left and right LR frames as $\mathbf{V}_l^l$ and $\mathbf{V}_l^r$, respectively, with size $W \times H/2$. Fig.4.5 shows the main stages of the proposed FR recovery mechanism. The downsampled views are expanded to their original size with the positions of the discarded pixels left empty (this stage is indicated by ① in Fig.4.5). The expanded left view is represented by $\mathbf{V}_e^l$ where $V_e^l(2n, m) = 0, 1 \leq n \leq H/2, 1 \leq m \leq W$, whereas, the expanded right view is represented by $\mathbf{V}_e^r$ where $V_e^r(2n-1, m) = 0, 1 \leq n \leq H/2, 1 \leq m \leq W$. Then in the second stage indicated by ② in Fig.4.5, based on the direction estimation results, a directional interpolation method is used to generate the corresponding interpolated frames, and these

are denoted by $\mathbf{V}_i^l$ and $\mathbf{V}_i^r$ for the left and right view, respectively. Meanwhile, in the third stage indicated by ③ in Fig.4.5, the DIBR technique is applied on the expanded frames using the corresponding depth maps in order to generate the virtual views at the counterpart viewpoints, i.e. the left side virtual view $\mathbf{V}_v^l$ is generated by the right side expanded view. As a consequence, all the even rows in the left virtual view are warped from the even rows in the right view, and these warped rows are, to some extent, equivalent to the rows discarded during the downsampling process. Similarly, for the right virtual view, all the odd rows are warped from the odd rows in the left view. Therefore, based on the above design which aims to make the recovery of discarded pixels work in synergy with the downsampling stage, the virtual view becomes a potential source of information to efficiently recover the discarded pixels. So, two parallel stages are used to recover the discarded information due to the downsampling process, and the outputs of these two stages are fused to generated the final FR frames. The fusion process is driven by the pattern direction of the texture around each of the discarded pixels, so as to exploit the potential of stages ② and ③.

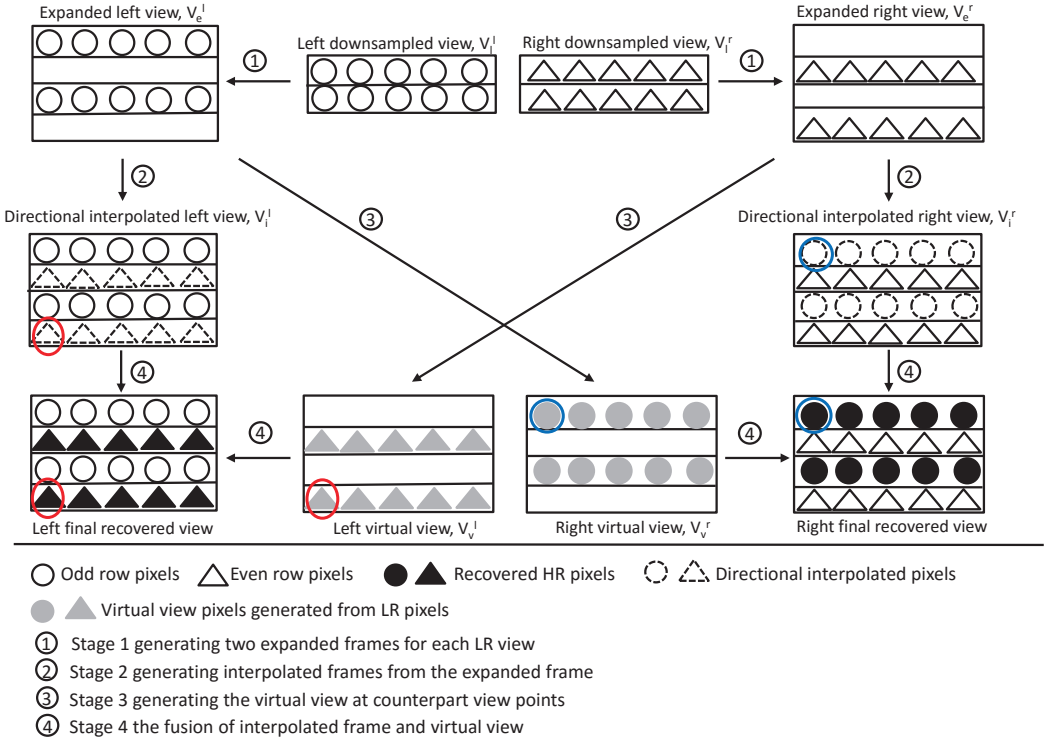


Figure 4.5: The proposed discarded pixels recovery process.

PCA-based pattern direction estimation

As mentioned in the previous section the fusion process is driven by the local pattern direction of the texture around each discarded pixel. Since knowing the dominant direction for each discarded pixel allows better exploitation of the virtual and interpolated frames in recovering the discarded pixels. For example, texture patterns with horizontal edges usually cannot be accurately estimated from their vertical neighbors, and due to the missing horizontal neighbors, they cannot be directly interpolated, either. Hence, in this case exploiting the pixels of the virtual view could greatly help recover these types of patterns.

To get the pattern direction, in this work, a Principal Components Analysis (PCA) [104] based method will be used. This approach evaluates the gradients of the surrounding pixels for each discarded pixel, and then the dominant direction of the texture is determined by PCA [105], where PCA can be obtained by evaluating the Singular Value Decomposition (SVD) [106] of the data.

In general, the gradient at $V(x,y)$ can be obtained by $\nabla V(x,y) = [\partial V(x,y)/\partial x, \partial V(x,y)/\partial y]^T$, and this could be approximated for discrete applications as:

$$\nabla V(x,y) \approx \begin{pmatrix} \frac{1}{2}(V(x+\Delta,y) - V(x-\Delta,y)) \\ \frac{1}{2}(V(x,y+\Delta) - V(x,y-\Delta)) \end{pmatrix} \tag{4.1}$$

$\Delta = 1$ offers the best approximation, however, taking into account that half of the rows are discarded in the LR frames, then Δ needs to be 2 while evaluating the gradients of the surrounding pixels of a discarded pixel. This ensures that $V(x+\Delta,y)$, $V(x-\Delta,y)$, $V(x,y+\Delta)$, and $V(x,y-\Delta)$ are available[1].

It is worth noticing that the horizontal neighbors of the discarded pixels are unavailable, therefore, the dominant direction for each discarded pixel will be inferred from the four corner pixels of a 3×3 overlapping window centered at the discarded pixel. For example the discarded pixel p_5, in Fig.4.6, has two discarded neighbors, namely p_4 and p_6, so in order to maintain an equivalent number of neighbors and symmetric structure around p_5, the two pixels p_2 and p_8 will not be taken into account while evaluating the dominant pattern direction. In other words only the gradients of the corner pixels p_1, p_3, p_7 and p_9 will be evaluated[2]. The gradients of the surrounding pixels of the

[1] The pixels on the boarder of the frame will be filled by filter-based interpolation without estimating their pattern directions.

[2] Although using p_2 and p_8 may seem beneficial, the lack of p_4 and p_6 will negatively affect direction estimation due to the non-symmetric set of pixels. Nevertheless, p_2 and p_8 will be used in pattern estimation of the following discarded pixel, p_6.

discarded pixel at position (x,y) will be then arranged into a 4×2 matrix $\mathbf{G}$ [106] , as follows

$$\mathbf{G} = \begin{bmatrix} \nabla V(x-1, y-1)^T \\ \nabla V(x-1, y+1)^T \\ \nabla V(x+1, y-1)^T \\ \nabla V(x+1, y+1)^T \end{bmatrix} \tag{4.2}$$

The SVD of the matrix $\mathbf{G}$ will be computed as $\mathbf{G} = \mathbf{USV^T}$, where $\mathbf{S}$ is a 4×2 diagonal matrix and the ratio of the diagonal elements in $\mathbf{S}$ (i.e., S_{11}/S_{22}) represents the energy of the dominant gradient. Both $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices with size 4×4 and 2×2, respectively, and the first column of $\mathbf{V}$ (i.e., $[\nu_{11}\ \nu_{21}]^T$) represents the orientation of the dominant gradient, whose angle is $\theta = \arctan(\nu_{21}/\nu_{11})$. For the remarkably dominant gradient (i.e. $S_{11}/S_{22} \geq Th$ where Th is a threshold to define the remarkably dominant gradient), this angle will be used to determine the pattern directions of the discarded pixel, which are horizontal, 45° diagonal, vertical and 135° diagonal directions as shown in Fig.4.6. For the pixels from texture uniform areas whose energy in all four directions is almost equal, there is no remarkably dominant directional pattern (i.e. $S_{11}/S_{22} < Th$), will be classified into the "undefined" direction category. This process will be carried forward for each discarded pixel in the left and right LR views at both the encoder and decoder sides.

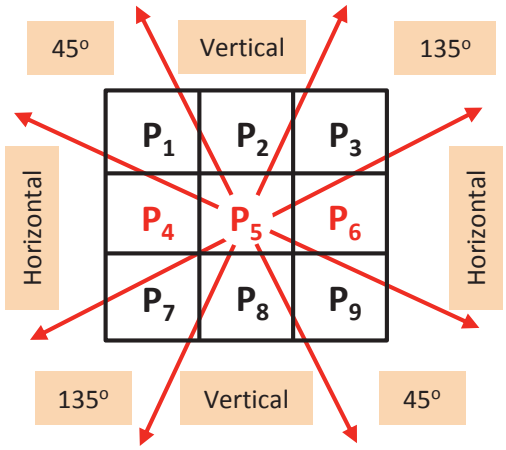


Figure 4.6: The overlapping window centered at the discarded pixel p_5. The dominant pattern direction will be categorized into five groups. In this figure only the remarkably dominant patterns are shown which are horizontal, 45° diagonal, vertical and 135° diagonal directions.

Directional data fusion

Since all of the discarded pixels, referring to the texture pattern of their surrounding pixels, are classified into five categories: horizontal, 45° diagonal, vertical, 135° diagonal and undefined direction, the directional interpolated frames $\mathbf{V}_i^l$ and $\mathbf{V}_i^r$ are generated based on this classification. That is to say, for the discarded pixel with vertical direction, it is interpolated by averaging its nearest upper and lower pixels; for the pixel with 45° or 135° diagonal, it is filled by averaging the corresponding nearest diagonal pixels; for the pixel with horizontal direction, it is the average of four nearest corner pixels. However, the undefined directional pixels could be easily recovered by vertical interpolation, since the vertical neighbors are the closest to the discarded pixels. Whereas, some high frequency components (e.g. edges) get recovered by exploiting inter-view redundancy from the counterpart view. Therefore, in the fourth stage of the proposed upsampling algorithm, the discarded pixels are recovered by fusing the interpolated pixels with the virtual view pixels in order to exploit the advantages of both types of approach and to compensate the compression distortion.

To reduce the compression effect, at the fusion stage, each discarded pixel is filled by a weighted average of the counterpart pixels in V_v and V_i as shown:

$$\hat{V}^l(2n,m) = \eta^l V_i^l(2n,m) + (1-\eta^l) V_v^l(2n,m) \tag{4.3}$$

The value of the weighting coefficient, η^l, is in the range $[0,1]$. This value, in theory, should be evaluated for each missing pixel and it determines the relative contribution of the directional interpolated pixel with respect to the virtual view pixel. The fusing coefficients could be obtained by minimizing the L2 distance between the recovered pixels and their counterpart original pixels, as follows.

$$\sum_{m=1}^{W} \sum_{n=1}^{H/2} \left(\hat{V}^l(2n,m) - V_f^l(2n,m) \right)^2 \tag{4.4}$$

Holes and occluded areas in the virtual views are excluded during the fusion process and in these areas, the discarded pixels are directly recovered by directional interpolation. Since the original FR frame is only available at the encoder side, this means that all the fusing coefficients need to be transmitted for each frame to the decoder side, obviously, this makes the pixel-by-pixel estimation of the weighting coefficient impractical.

In this chapter, a pattern direction based weighting coefficient estimation is proposed which can hugely reduce the transmitted side information. In this approach,

at both the encoder and decoder side the fusion stage classifies all of the discarded pixels, based on the texture pattern of their surrounding pixels into five categories, and these five categories will be respectively represented by five binary masks $\mathbf{M_h}$, $\mathbf{M_{45}}$, $\mathbf{M_v}$, $\mathbf{M_{135}}$ and $\mathbf{M_{ud}}$. That is to say, the binary value "1" in $\mathbf{M_h}$ indicates that the discarded pixel in that position has a horizontal texture pattern, in this case the same position in $\mathbf{M_v}$, $\mathbf{M_{45}}$, $\mathbf{M_{135}}$ and $\mathbf{M_{ud}}$ will have "0" binary value. For each directional mask, one weighting coefficient will be estimated by using (4.4). Therefore, equation (4.3) could be rewritten in matrix format, while taking into account the five pattern categories, as follows:

$$\begin{aligned}\hat{\mathbf{V}}^l &= \eta_h^l \mathbf{M_h^l}. * \mathbf{V}_i^l + (1-\eta_h^l)\mathbf{M_h^l}. * \mathbf{V}_v^l \\ &+\eta_{45}^l \mathbf{M_{45}^l}. * \mathbf{V}_i^l + (1-\eta_{45}^l)\mathbf{M_{45}^l}. * \mathbf{V}_v^l \\ &+\eta_v^l \mathbf{M_v^l}. * \mathbf{V}_i^l + (1-\eta_v^l)\mathbf{M_v^l}. * \mathbf{V}_v^l \\ &+\eta_{135}^l \mathbf{M_{135}^l}. * \mathbf{V}_i^l + (1-\eta_{135}^l)\mathbf{M_{135}^l}. * \mathbf{V}_v^l \\ &+\eta_{ud}^l \mathbf{M_{ud}^l}. * \mathbf{V}_i^l + (1-\eta_{ud}^l)\mathbf{M_{ud}^l}. * \mathbf{V}_v^l\end{aligned} \tag{4.5}$$

where $\hat{\mathbf{V}}^l$ denotes the recovered image. The operation $.*$ represents the element-by-element multiplication of two matrixes. A graphic representation of the proposed data fusion process is shown in Fig.4.7.

Given that the encoder and decoder work on the same set of data to estimate the pattern direction, there is no need to transmit the masks $\mathbf{M_h}$, $\mathbf{M_{45}}$, $\mathbf{M_v}$, $\mathbf{M_{135}}$ and $\mathbf{M_{ud}}$ from the encoder to the decoder side and only the directional weighting coefficients for the left view (i.e. η_h^l, η_{45}^l, η_v^l, η_{135}^l and η_{ud}^l), and for the right view, need to be estimated at the encoder side and transmitted to the decoder side. Obviously, the overhead rate of transmitting the weighting coefficients is negligible in comparison to the texture and depth map bit rate. Moreover, it is worth indicating that the pattern direction estimation stage is much less complex than the video encoding stage. In the experimental results section the term DDFU will be used to refer to this full version scheme.

In addition, DDFU can be simplified to only transmit the weighting coefficients of the first frame to the decoder side and to use them later on for the fusion of all other frames. This simplification is possible because the content of each frame does not change significantly, especially for sequences with slow motion. Based on this observation, the simplified approach can further reduce the amount of transmitted side information with little quality degradation. In the experimental section the term DDFU (first frame η) will be used to refer to this simplified scheme.

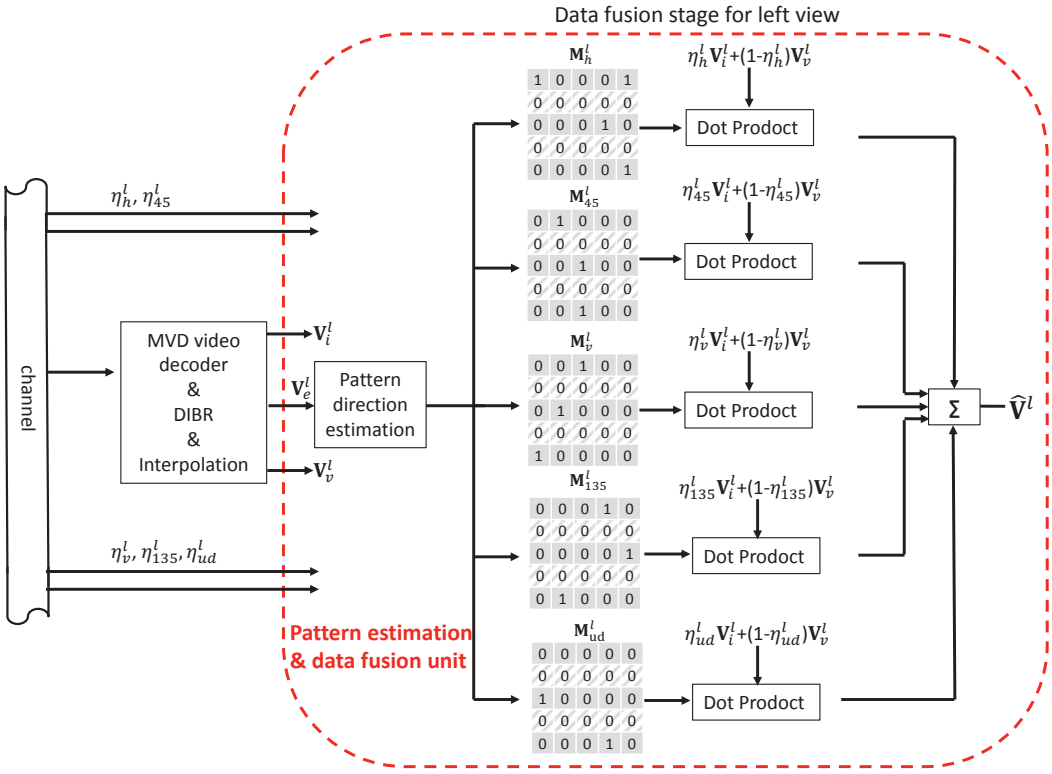


Figure 4.7: The process of data fusion by directional weighting coefficients and corresponding directional binary masks

4.3 Experimental results

To objectively evaluate the performance of the proposed method, several experiments were conducted with the following 3D video sequences "*Doorflower*', "*Kendo*", "*Dog*", "*Balloons*", "*Newspaper*" and "*Undo-Dancer*". Some parameters and content characteristics of the testing sequences are listed in Table 4.1 for reference. The depth maps of "*Doorflower*" were estimated by depth estimation reference software (DERS) 5.0 [107]. The depth maps of all of the sequences from Nagoya University were computed by the

Table 4.1: The parameters and characteristics of each used sequence

	Size	Camera	Left	Right	Content's Motion
Doorflower	1024 × 768	Fixed	View10	View08	Moderate
Kendo	1024 × 768	Moving	View03	View05	Complex
Dog	1280 × 960	Fixed	View38	View39	Medium
Balloons	1024 × 768	Moving	View03	View05	Complex
Newspaper	1024 × 768	Fixed	View04	View06	Simple
Undo-Dancer	1920 × 1088	Moving	View02	View05	Complex

depth estimation software provided by Nagoya University Tanimoto Laboratory [108] and the depth maps of "*Undo-Dancer*" were generated by computer graphics. For each sequence both the left and right views had been interlacing-and-complementary-row downsampled with a factor 2 before encoding. JMVC 8.5 [109] was used for compression, and eight different QPs, namely $28, 31, 34, 37, 40, 43, 46, 49$, were used to code the texture and depth map sequences. The temporal GOP size and the total number of encoded frames was 8 and 80, respectively, while the delta QP and the differential QP between the base layer and sublayer in hierarchical-B picture structure was set to zero in all layers. The virtual views at the decoder side were rendered using a 1D DIBR technique from one reference view to another view without any post-processing (i.e., no hole filling).

The first set of simulations aim to evaluate the effectiveness of the proposed approach by comparing the rate-distortion performance with FR video coding and the 6-tap Lanczos filter approach as well as state-of-the-art approach [9]. In the first part of comparison, the 6-tap Lanczos filter has been used at both encoder and decoder sides and the results are reported in Fig.4.8 for all of the tested sequences. In the following experiments, this matched-filter-based approach will be treated as a benchmark in this chapter.

From the results in Fig.4.8 the effectiveness of the proposed systematic down/upsampling approach over the matched filter approach and FR coding at low bit rate could be appreciated for all testing sequences. The coding performance improvement of the proposed method and benchmark method over FR coding performance is due to the adoption of the down/upsampling processes. The proposed method, in comparison with the benchmark method, can gain up to 1.14dB and 1.03dB on the sequences "*Kendo*" and "*Doorflower*", respectively. This is due to the high quality depth map which makes the contribution of the generated virtual view pixels significant. The sequence "*Kendo*" and "*Doorflower*" have more gain than the sequence "*Dog*". By fusing directional interpolated pixels with virtual view pixels, edges can be well preserved. The matched filter approach has good coding performance on the smooth areas, therefore, for the sequences (e.g. "*Dog*" and "*Undo-Dancer*") containing more smooth areas, the benchmark method is comparable to the proposed method. Moreover, the average PSNR gains across different bitrates for all the sequences range from 0.17dB to 0.68dB.

To further evaluate the effectiveness of the proposed method, comparisons with the

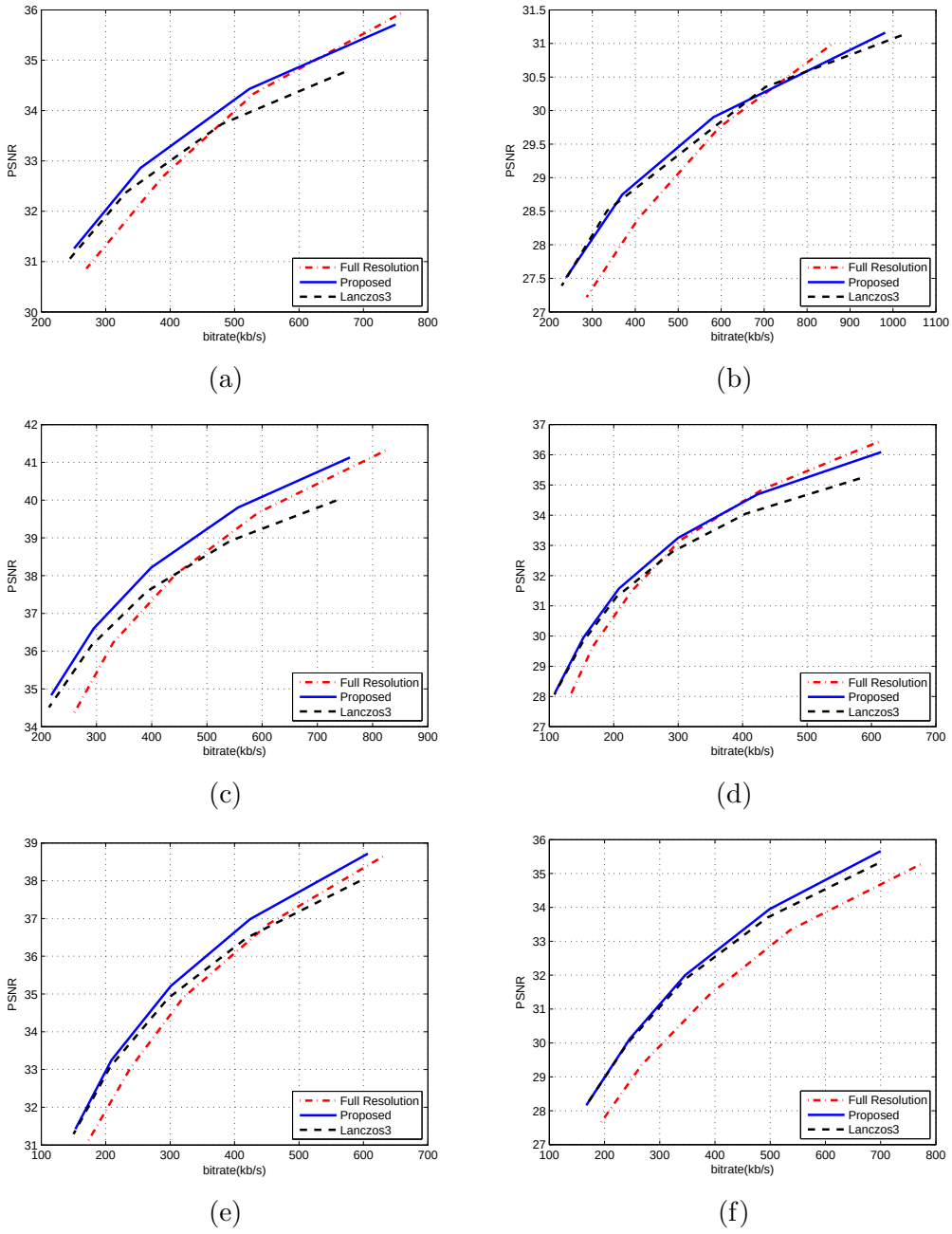
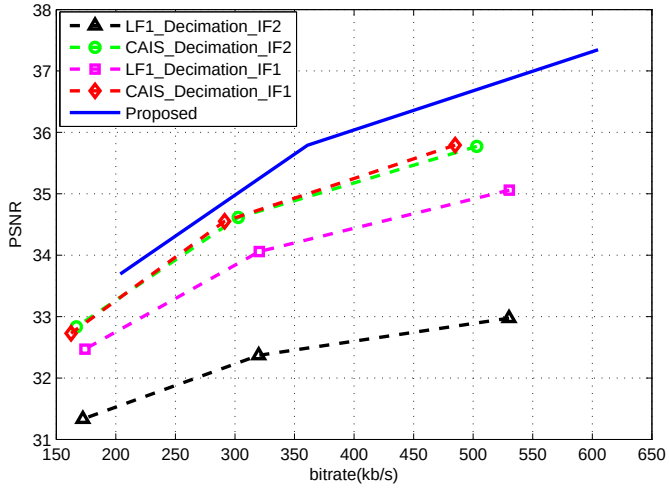


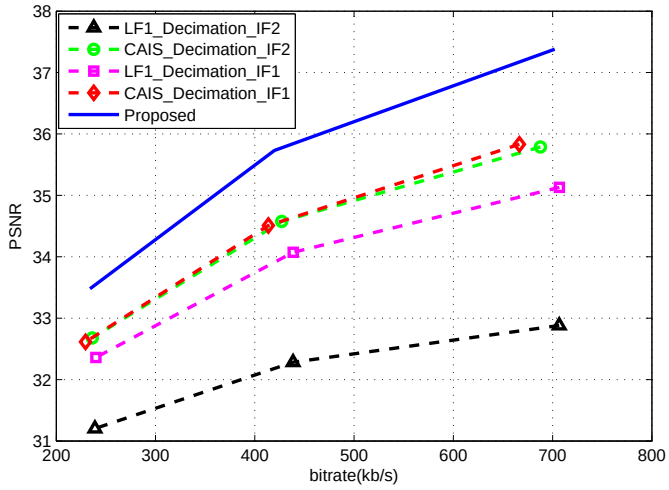
Figure 4.8: The rate-distortion curves for the testing sequences (a) Doorflower; (b) Dancer; (c) Kendo ; (d) Newspaper; (e) Balloons; (f) Dog.

method [9][3] have also been carried out by using the same test sequences ("*Doorflower*" and "*Laptop*") with the same resolution (512×384) and the same coding standard, i.e. H.264/AVC with the same coding parameters. The results of these comparisons are shown in Fig.4.9, where LF1 represents the direct downsampling (i.e. even rows in both left and right views are discarded without low pass filtering), CAIS repre-

[3]All the results of [9] have been obtained from the authors and their paper.



(a)



(b)

Figure 4.9: The rate-distortion curves for the testing sequences (a) Doorflower; (b) Laptop, for the proposed approach and [9].

sents the proposed method in [9], IF1 and IF2 are two interpolation filters with coefficients $\{1, -5, 20, 20, -5, 1\}/32$ and $\{-3, 28, 8, -1\}/32$, respectively, as proposed in [9]. The depth sequences used in the proposed DDFU are generated by the method given in [110] and their bitrates have been included in the results. Indicated by these results, the gain of the proposed method is larger than that of [9].

The visual results of zoomed-in parts of the sequences "*Doorflower*" and "*Undodancer*" are shown in Fig.4.10. It is possible to note that the edges recovered by DDFU are sharper than those that are recovered by the benchmark method. Although the proposed DDFU recovered frame also has some blurred areas, nevertheless, it still

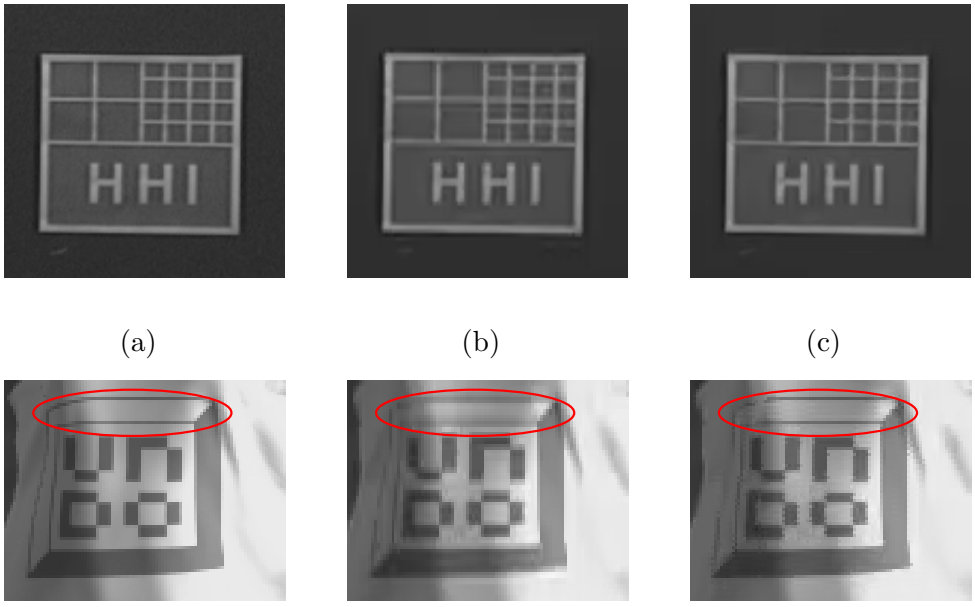


Figure 4.10: Comparison between proposed DDFU method and benchmark method. (a)-(c) are the results of Original, Benchmark and DDFU on zoomed-in part of the sequence Doorflower; (d)-(f) are the results of Original, Benchmark and DDFU on zoomed-in part of the sequence Undo-Dancer.

achieves a better visual quality than matched-filter-interpolated frame. Fig.4.10 (d) shows a portion of the original left view of "*Undo-Dancer*", and its recovered versions using the matched-filter-based approach and the proposed approach is shown in Fig.4.10 (e) and Fig.4.10 (f), respectively. Since the one-pixel-wide edge is difficult to recover properly using only the surrounding pixels, the advantage of the DDFU method is more obvious in the highlighted areas by red ellipse in Fig.4.10 (e) and Fig.4.10 (f). From this comparison, it can be seen that the proposed approach can recover the one-pixel-wide edge without blurring.

Since, the category of the to-be-filled pixels is determined by the estimated texture pattern, accurate pattern direction estimation plays an important role in the fusion process. Therefore, to verify its effectiveness, Fig.4.11 (b) shows the pattern estimation result on the uncompressed frame, whereas, Fig.4.11 (c) and Fig.4.11 (d) show the estimation results on the compressed frame with QP = 34 and QP = 40, respectively. For reference, Fig.4.11 (a) shows the original uncompressed texture frame from the "*Doorflower*" sequence with three highlighted parts containing clear patterns. Different colors are used to distinguish the five directions, so the colors dark red, red, orange,

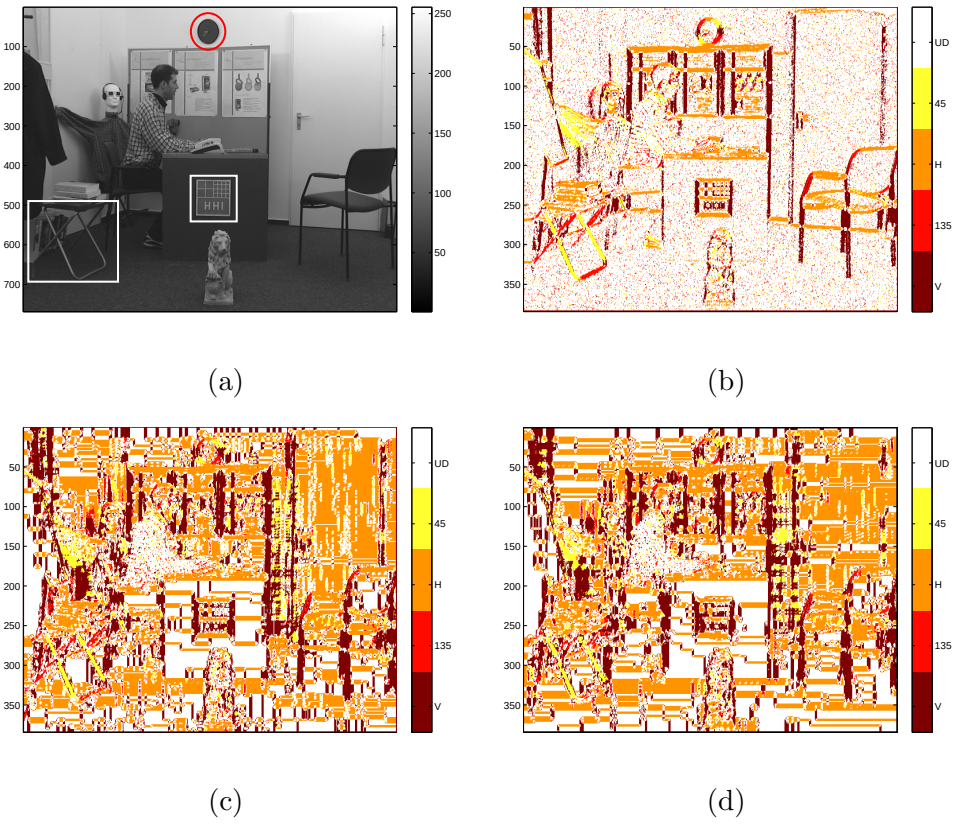


Figure 4.11: (a) original texture; the pattern direction estimation results on: (b) original uncompressed texture; (c) compressed texture with QP=34; (d) compressed texture with QP=40; the color: dark red, red, orange, yellow and white represent vertical, 135° diagonal, horizontal, 45° diagonal and undefined direction pixels, respectively. (For clearness, the directional estimation results on the discarded pixels are scaled up to the same size as the original texture; their real height is shown on the y axis of each figure).

yellow and white are used to represent vertical, 135 diagonal, horizontal, 45 diagonal edges and the undefined direction areas, respectively. In this chapter, pixels are regarded as undefined pattern pixels when $S_{11}/S_{22} \leq Th$ where $Th = 4$. The accuracy of the adopted pattern detection algorithm could be appreciated from Fig.4.11 (b) and Fig.4.11 (c). By comparing these two figures, the direction estimation results of the three highlighted parts are almost the same, which could also demonstrate that the accuracy of the pattern estimation is barely affected by the compression distortion.

To show the level of contribution of the virtual views in the fusion stage and how the texture pattern direction influences the fusing process, the fusion coefficients η_h, η_v, η_{45}, η_{135} and η_{ud} are reported in Table 4.2 for the six testing sequences and for different QPs. The smaller the value of η is, the more important the virtual view pixels are for the recovery of the discarded pixels. Obviously, in the fusion stage, the contribution of

Table 4.2: The values of η_h, η_v, η_{45}, η_{135} and η_{ud} for each sequence and for different QPs

Doorflower									
QP		28	31	34	37	40	43	46	49
Leftview	η_h	0.18	0.20	0.20	0.22	0.26	0.37	0.47	0.60
	η_v	1.00	0.99	0.98	0.96	0.92	0.90	0.89	0.89
	η_{45}	0.75	0.76	0.76	0.75	0.79	0.78	0.78	0.78
	η_{135}	0.79	0.80	0.80	0.80	0.80	0.84	0.89	0.92
	η_{ud}	0.55	0.55	0.57	0.60	0.62	0.68	0.70	0.75
Rightview	η_h	0.28	0.28	0.26	0.27	0.34	0.38	0.47	0.49
	η_v	0.99	0.98	0.97	0.94	0.89	0.87	0.82	0.75
	η_{45}	0.72	0.76	0.77	0.77	0.78	0.76	0.72	0.72
	η_{135}	0.78	0.80	0.79	0.80	0.79	0.77	0.75	0.66
	η_{ud}	0.59	0.59	0.59	0.62	0.63	0.62	0.58	0.50
Undo-Dancer									
QP		28	31	34	37	40	43	46	49
Leftview	η_h	0.06	0.08	0.12	0.16	0.24	0.39	0.61	0.79
	η_v	0.90	0.95	1.00	1.00	1.00	0.99	1.00	0.98
	η_{45}	0.41	0.52	0.64	0.75	0.82	0.87	0.87	0.87
	η_{135}	0.39	0.48	0.62	0.73	0.80	0.84	0.85	0.84
	η_{ud}	0.12	0.14	0.19	0.30	0.43	0.60	0.77	0.85
Rightview	η_h	0.04	0.06	0.10	0.16	0.22	0.32	0.39	0.36
	η_v	0.90	0.93	0.95	0.96	0.96	0.93	0.90	0.88
	η_{45}	0.41	0.51	0.61	0.70	0.77	0.82	0.79	0.73
	η_{135}	0.42	0.50	0.62	0.72	0.78	0.80	0.77	0.71
	η_{ud}	0.12	0.13	0.19	0.27	0.38	$0.\eta_{45}$	0.48	0.47
Kendo									
QP		28	31	34	37	40	43	46	49
Leftview	η_h	0.79	0.76	0.76	0.76	0.74	0.73	0.72	0.73
	η_v	0.90	0.90	0.90	0.87	0.85	0.82	0.83	0.85
	η_{45}	0.95	0.93	0.92	0.90	0.88	0.86	0.83	0.81
	η_{135}	0.90	0.89	0.86	0.82	0.80	0.77	0.75	0.73
	η_{ud}	0.84	0.81	0.77	0.77	0.77	0.79	0.79	0.80
Rightview	η_h	0.91	0.90	0.89	0.86	0.81	0.73	0.64	0.54
	η_v	1.00	1.00	0.99	0.91	0.88	0.79	0.72	0.63
	η_{45}	1.00	0.99	0.96	0.91	0.89	0.82	0.71	0.61
	η_{135}	1.00	1.00	1.00	0.98	0.94	0.90	0.85	0.77
	η_{ud}	0.99	0.99	0.95	0.89	0.81	0.70	0.60	0.51
Newspaper									
QP		28	31	34	37	40	43	46	49
Leftview	η_h	0.89	0.90	0.87	0.88	0.83	0.82	0.81	0.79
	η_v	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99
	η_{45}	1.00	1.00	1.00	1.00	1.00	0.99	0.98	0.97
	η_{135}	1.00	1.00	1.00	1.00	1.00	0.95	0.92	0.90
	η_{ud}	1.00	0.99	0.98	0.98	0.98	0.94	0.94	0.94
Rightview	η_h	0.90	0.90	0.90	0.90	0.90	0.87	0.81	0.75
	η_v	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.93
	η_{45}	1.00	1.00	1.00	1.00	0.99	0.97	0.90	0.88
	η_{135}	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97
	η_{ud}	1.00	1.00	0.99	0.99	0.99	0.93	0.90	0.84
Balloons									
QP		28	31	34	37	40	43	46	49
Leftview	η_h	0.85	0.81	0.80	0.80	0.77	0.75	0.77	0.83
	η_v	0.94	0.90	0.90	0.89	0.89	0.88	0.89	0.89
	η_{45}	0.97	0.92	0.90	0.90	0.89	0.88	0.86	0.86
	η_{135}	0.96	0.92	0.90	0.90	0.89	0.83	0.80	0.80
	η_{ud}	0.90	0.90	0.87	0.86	0.86	0.85	0.87	0.89
Rightview	η_h	0.96	0.94	0.93	0.90	0.87	0.79	0.68	0.46
	η_v	1.00	1.00	1.00	1.00	1.00	0.99	0.93	0.84
	η_{45}	1.00	1.00	1.00	1.00	0.98	0.92	0.88	0.78
	η_{135}	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.93
	η_{ud}	1.00	1.00	1.00	1.00	0.97	0.91	0.82	0.67
Dog									
QP		28	31	34	37	40	43	46	49
Leftview	η_h	0.90	0.90	0.90	0.89	0.88	0.84	0.80	0.69
	η_v	1.00	1.00	0.99	0.97	0.91	0.90	0.88	0.84
	η_{45}	1.00	1.00	0.98	0.93	0.90	0.89	0.81	0.74
	η_{135}	1.00	0.99	0.98	0.93	0.90	0.89	0.82	0.80
	η_{ud}	1.00	1.00	0.98	0.95	0.90	0.90	0.90	0.82
Rightview	η_h	1.00	1.00	1.00	1.00	0.97	0.94	0.91	0.85
	η_v	1.00	1.00	1.00	1.00	1.00	0.98	0.90	0.77
	η_{45}	1.00	1.00	1.00	0.99	0.97	0.91	0.88	0.82
	η_{135}	1.00	1.00	1.00	1.00	1.00	0.97	0.90	0.76
	η_{ud}	1.00	1.00	1.00	1.00	0.99	0.93	0.90	0.81

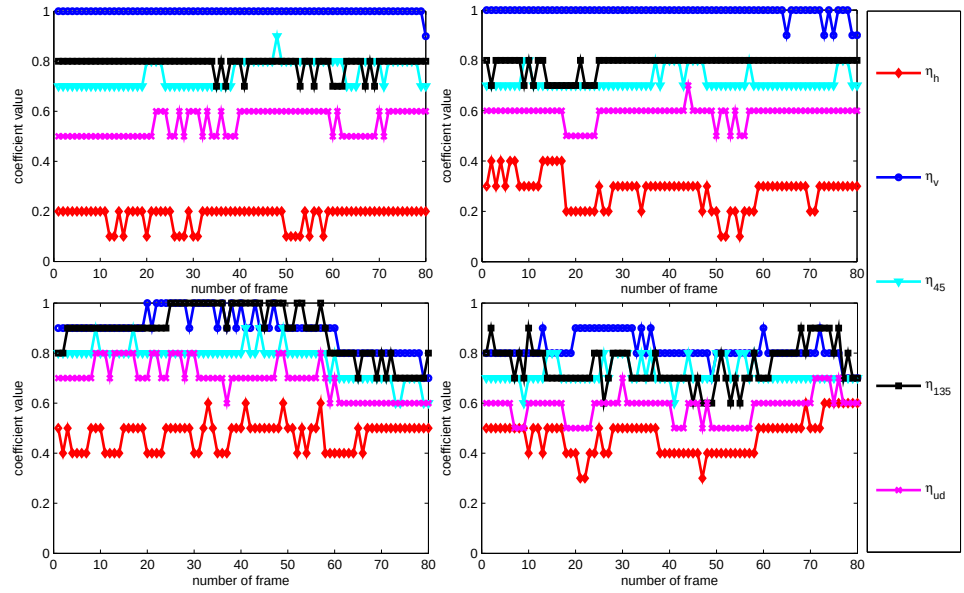
virtual view depends on several factors, such as, the adopted DIBR technique and depth map quality. It is worth noticing that even with advanced rendering techniques, the generated virtual view may still face a problem in generating high quality and aligned texture around depth discontinuous areas, where the adopted directional interpolation can work well. From this table it can be seen that virtual view pixels are more important to recover the pixels with horizontal pattern than other directions. On the other hand, the directional interpolated frame is more important in recovering the pixels with vertical pattern, for example η_h for the left view of the "*Undo-Dancer*" sequence at QP = 28 is 0.06 versus η_v = 0.9. The η_{45}, η_{135} and η_{ud} values for the two diagonal patterns and undefined pattern lay in between the horizontal and vertical cases, as for instance the "*Undo-Dancer*" sequence at QP = 28 these are η_{45} = 0.41, η_{135} = 0.39 and η_{ud} = 0.12. Moreover, it should be noted that for the sequence "*Undo-Dancer*" which is a computer graphic sequence, and consequently has an accurate depth map, the virtual view pixels provide the greatest contribution to the final recovered FR frames in all five directions, with respect to other sequences. As expected, this contribution is remarkably higher for the horizontal pattern.

The upsampling performances of the proposed approach and the 6-tap Lanczos filter are shown in Table 4.3 and direct downsampling (i.e. even rows in both left and right views are discarded without low pass filtering) is adopted. Table 4.3 depicts that for all the testing sequences with different QP values, the PSNR and SSIM [97] results of the proposed upsampling approach are higher than that of the Lanczos filter. The average PSNR gain for all sequences ranges between 0.33dB to 0.55dB. By comparing the upsampling performance at the decoder side, the importance of pattern direction information and data fusion can be appreciated. Moreover, by using direct downsampling rather than the proposed interlacing-and-complementary-row-downsampling, the biggest drop in PSNR gain is up to 0.3dB.

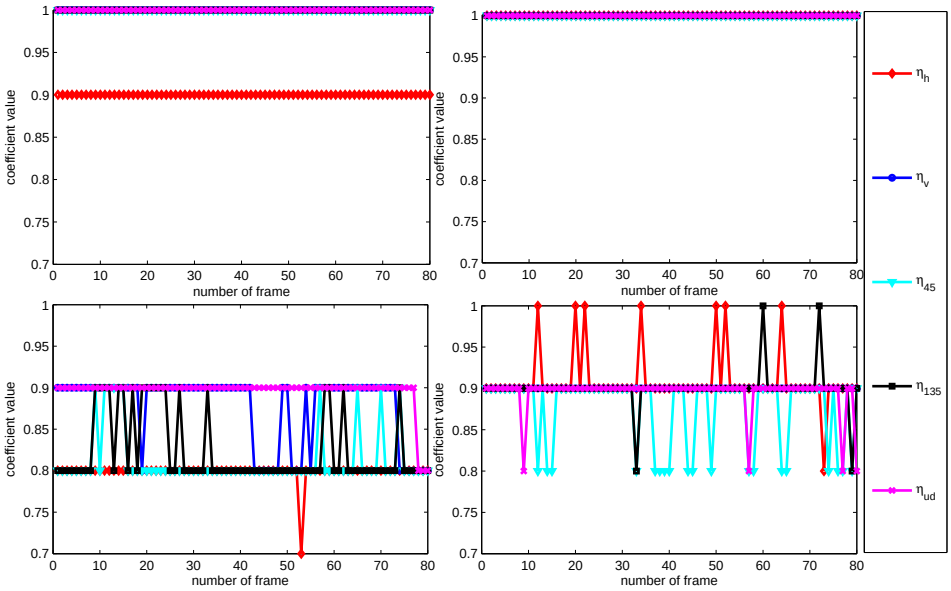
In the basic implementation of the DDFU algorithm, the encoder needs to transmit the five weighting coefficients, η, for each frame and each view, and obviously it needs to evaluate them by minimizing (4.4). However, given that in most cases there are no major changes in the scene content, it is reasonable to assume that those coefficients do not change very much from frame to frame, hence it is not necessary to evaluate them for each frame. This assumption could be verified by Fig.4.12 which shows the trend of the weighting coefficients versus frame number. Thus, one way to reduce the

Table 4.3: The upsampling performance on PSNR (dB) and SSIM comparison by discarding even rows directly downsampling

QP			28	31	34	37	40	43	46	49
Bitrate(kb/s)			1562	1070	722	505	346	245	167	111
Doorflower	PSNR	Lanc	35.76	35.25	34.53	33.59	32.35	31.01	29.31	27.77
		Pro	36.87	36.18	35.25	34.13	32.71	31.24	29.46	27.87
	ΔPSNR		**1.11**	**0.93**	**0.72**	**0.54**	**0.36**	**0.23**	**0.15**	**0.10**
	SSIM	Lanc	0.979	0.974	0.968	0.958	0.942	0.920	0.883	0.838
		Pro	0.981	0.976	0.969	0.960	0.944	0.923	0.888	0.845
	ΔSSIM		**0.002**	**0.002**	**0.001**	**0.002**	**0.002**	**0.003**	**0.004**	**0.007**
Bitrate(kb/s)			4123	2526	1482	918	563	367	240	164
Undo-Dancer	PSNR	Lanc	32.11	31.67	31.07	30.40	29.55	28.60	27.46	26.43
		Pro	32.92	32.29	31.51	30.69	29.73	28.72	27.54	26.50
	ΔPSNR		**0.81**	**0.63**	**0.44**	**0.29**	**0.18**	**0.11**	**0.08**	**0.06**
	SSIM	Lanc	0.978	0.969	0.955	0.939	0.921	0.898	0.862	0.824
		Pro	0.980	0.971	0.957	0.941	0.923	0.901	0.866	0.830
	ΔSSIM		**0.002**	**0.001**	**0.001**	**0.002**	**0.002**	**0.003**	**0.005**	**0.006**
Bitrate(kb/s)			752	550	396	293	216	166	123	94
Kendo	PSNR	Lanc	39.93	38.88	37.56	36.13	34.49	32.76	30.88	29.03
		Pro	41.07	39.75	38.18	36.57	34.81	33.02	31.10	29.24
	ΔPSNR		**1.15**	**0.87**	**0.62**	**0.44**	**0.32**	**0.25**	**0.22**	**0.21**
	SSIM	Lanc	0.988	0.985	0.980	0.973	0.962	0.947	0.925	0.897
		Pro	0.989	0.986	0.982	0.976	0.966	0.953	0.933	0.907
	ΔSSIM		**0.001**	**0.001**	**0.002**	**0.003**	**0.004**	**0.006**	**0.008**	**0.010**
Bitrate(kb/s)			864	608	418	297	207	152	109	83
Newspaper	PSNR	Lanc	35.90	35.05	33.92	32.70	31.23	29.71	27.97	26.32
		Pro	37.20	36.06	34.66	33.22	31.57	29.94	28.12	26.42
	ΔPSNR		**1.29**	**1.01**	**0.74**	**0.52**	**0.33**	**0.23**	**0.15**	**0.10**
	SSIM	Lanc	0.982	0.976	0.967	0.955	0.937	0.912	0.873	0.826
		Pro	0.983	0.978	0.969	0.957	0.939	0.916	0.880	0.836
	ΔSSIM		**0.001**	**0.002**	**0.002**	**0.002**	**0.003**	**0.004**	**0.007**	**0.011**
Bitrate(kb/s)			830	602	421	298	208	151	108	83
Balloons	PSNR	Lanc	39.08	37.94	36.47	34.86	33.04	31.28	29.45	27.74
		Pro	40.15	38.72	36.99	35.22	33.27	31.45	29.57	27.84
	ΔPSNR		**1.07**	**0.78**	**0.53**	**0.36**	**0.23**	**0.17**	**0.11**	**0.11**
	SSIM	Lanc	0.986	0.981	0.973	0.960	0.939	0.911	0.874	0.832
		Pro	0.987	0.983	0.975	0.963	0.943	0.917	0.883	0.847
	ΔSSIM		**0.001**	**0.002**	**0.002**	**0.003**	**0.004**	**0.006**	**0.009**	**0.015**
Bitrate(kb/s)			1428	1010	697	496	344	244	165	115
Dog	PSNR	Lanc	37.78	36.64	35.21	33.63	31.80	29.98	28.07	26.39
		Pro	38.66	37.28	35.64	33.92	31.98	30.11	28.16	26.47
	ΔPSNR		**0.88**	**0.64**	**0.43**	**0.29**	**0.18**	**0.13**	**0.09**	**0.08**
	SSIM	Lanc	0.989	0.983	0.974	0.961	0.938	0.903	0.846	0.781
		Pro	0.990	0.984	0.976	0.964	0.942	0.909	0.853	0.788
	ΔSSIM		**0.001**	**0.001**	**0.002**	**0.003**	**0.004**	**0.006**	**0.007**	**0.007**



(a)



(b)

Figure 4.12: (a) and (b) panes show the four coefficients for the sequence "Doorflowers" and "Dog", respectively. The top Left and Right figures of each pane are: the weighting coefficients of Left and Right view, respectively, when QP=28; The bottom Left and Right figures of each pane are: the weighting coefficients of Left and Right view, respectively, when QP=46.

complexity of the proposed approach could be achieved by using directional weighting coefficients of the first frame for the whole sequence (called DDFU (first frame η)). In this approach, the weighting coefficients are only estimated for the first frame and then used for the whole sequence. To verify the effectiveness of this simplified approach, its

performance has been compared with the DDFU approach and DDFU (user defined η), the latter adopts user defined coefficients at the decoder side for the whole sequence. The pre-set values for the DDFU (user defined η) used are η_v = 1, η_h = 0, η_{45} = 0.5, η_{135} = 0.5 and η_{ud} = 1 for the left and right view which means that all vertical edges and undefined pattern areas are recovered by the directional interpolation algorithm. All recovered horizontal edges are obtained from the virtual view pixels, and the two diagonal direction pixels are obtained by equally fusing the directional interpolated pixels with the virtual view pixels. The results of this comparison are listed in Table 4.4.

Table 4.4: The PSNR (dB) comparison between: deriving η values for each frame, using the η values of first frame and user defined η values for the whole sequence

Sequences	Methods	QP							
		28	31	34	37	40	43	46	49
Doorflower	DDFU	37.80	36.92	35.79	34.49	32.90	31.29	29.48	27.86
	DDFU(first frame η)	37.79	36.91	35.78	34.48	32.89	31.29	29.47	27.85
	DDFU(user defined η)	37.38	36.57	35.52	34.29	32.74	31.17	29.37	27.76
Undo-Dancer	DDFU	35.07	33.75	32.38	31.16	29.90	28.75	27.51	26.46
	DDFU(first frame η)	35.04	33.74	32.37	31.15	29.89	28.75	27.51	26.46
	DDFU(user defined η)	33.20	32.49	31.63	30.74	29.71	28.65	27.44	26.39
Kendo	DDFU	41.13	39.80	38.21	36.60	34.83	33.05	31.13	29.27
	DDFU(first frame η)	41.10	39.79	38.21	36.59	34.83	33.04	31.12	29.26
	DDFU(user defined η)	39.43	38.51	37.31	35.98	34.43	32.74	30.89	29.07
Newspaper	DDFU	37.24	36.09	34.69	33.25	31.57	29.95	28.13	26.42
	DDFU(first frame η)	37.23	36.09	34.69	33.25	31.57	29.95	28.13	26.42
	DDFU(user defined η)	34.65	34.01	33.15	32.11	30.83	29.44	27.78	26.18
Balloon	DDFU	40.16	38.72	36.98	35.21	33.25	31.42	29.54	27.80
	DDFU(first frame η)	40.16	38.72	36.98	35.21	33.25	31.42	29.54	27.79
	DDFU(user defined η)	38.61	37.59	36.23	34.71	32.93	31.21	29.40	27.68
Dog	DDFU	38.67	37.31	35.66	33.95	32.00	30.13	28.16	26.47
	DDFU(first frame η)	38.67	37.31	35.66	33.95	32.00	30.12	28.16	26.47
	DDFU(user defined η)	37.09	36.09	34.80	33.37	31.64	29.90	28.02	26.40

From Table 4.4 it can be noticed that DDFU and DDFU (first frame η) have almost similar performance for all sequences, which demonstrates the validity and effectiveness of the simplified approach. By comparing the results of DDFU and DDFU (user defined η) the importance of adapting the coefficients to the scene content can be appreciated. The results in Table 4.4 show that the performance of DDFU (first frame η) are better

than that of DDFU (user defined η). This is due to the fact that the η values for the DDFU (first frame η) are based on the content of the testing sequence, if the content of the sequence does not vary hugely frame-by-frame, neither does the value of η. While, the values of the predetermined η are user defined values, which means they do not take the content of the sequence into account. The performance of DDFU (user defined η) highly depends on how close the predetermined values are to the frame-by-frame evaluated coefficients.

4.4 Conclusions

In this chapter, an interlacing-and-complementary-row-downsampling method is employed on the two adjacent views of a multiview video at the encoder side to reduce the transmitted data and cost bit-rate. This downsampling method allows, the proposed Directional Data Fusion Unsampling (DDFU) algorithm, to recover the discarded pixels by exploiting the information of the downsampled views and the corresponding virtual views. In the proposed upsampling approach, edge directions around the discarded pixels are estimated by principal components analysis. This information is subsequently used to steer the fusion of the virtual view with the directional interpolated pixels. The aim behind this is to exploit the inter-view redundancy to minimize the overall system distortion, which is a combination of the compression distortion and the distortion introduced by the downsampling process. Therefore, different from conventional interpolation algorithms, the advantages of virtual views have been exploited by the proposed method. Moreover, it has been shown that the proposed algorithm achieves superior performance in comparison with conventional interpolation algorithm, Lanczos and the state-of-the-art algorithm, CAIS. The future work will be to exploit the temporal correlation, in video sequences, to control the fusion process.

Chapter 5

Depth Map Super-Resolution by Exploiting Planar Surfaces

In this chapter, an unsupervised Planar Surface Detection (PSD) algorithm employing the depth map is presented. The proposed **Depth map based Planar Surface Detection (DPSD)** method detects planar surfaces by adopting a dynamic seed growing approach. The valid seed patches are used sequentially according to their level of "planarity", which means the more flat the seed is, the earlier the seed will be used in the growing stage. An equation estimating each planar surface is used to steer its growing mechanism. This equation gets refined at each growing stage, by using the newly englobed neighboring pixels, so as to enhance the accuracy of the estimated plane equation. Each growing surface grows to a maximum extent until the next surface get detected. Furthermore, two post-processing methods are proposed to correct the problem of overgrowing surfaces and to merge over-segmented surfaces, thus making the proposed approach resilient to depth noise. According to the global analytical equations of the detected surfaces in the scene, a proper depth map SR approach is proposed with three different categories: planar surfaces, non-planar surfaces and edges.

The rest of this chapter is organized as follows. In Section 5.1 the details of the closely related works are presented; Section 5.2 and 5.3 describe the proposed planar surface detection method and the proposed depth map SR approach, respectively. The experimental results are presented in Section 5.4. Section 5.5 concludes the chapter and also contains ideas for future work.

5.1 Related Work

5.1.1 Planar Surface Detection

Many different PSD methods have been proposed, and they can be classified into three main categories according to their working principles [111] [112]:

Iterative plane fitting methods

Iterative plane fitting or iterative initial estimates refining is a common approach used for planar surface detection and the typical representative is the RANdom SAmples Consensus (RANSAC) algorithm [113]. RANSAC is an iteratively randomized model fitting process and the initial fitting model is obtained based on several randomly selected points. For each remaining point from the whole data set, its distance to the model is evaluated and if the distance is smaller than the predefined error tolerance, the point will be regarded as an inlier in this model. Subsequently, another fitting model is set up and the remaining points are checked again. The number of inliers of one model indicates how well the model fits for the remaining points. The whole process starting from seed selection, model generation and the finalization stage including finding the maximum number of inliers is repeated until the best model is found and then all the inliers for the best fitting model are removed from data set. Next the model finding and fitting process begins. When the fitting error of remaining points from the whole data set is smaller than a predefined error tolerance, these points become inliers to this model until maximum number of points have been involved. The number of inliers of one model indicates how the model fits well for the remaining points.

RANSAC is efficient in detecting large planes and robust to noisy data, however, it has a high computational cost. Meanwhile, it tends to over-simplify complex planar structures. That is to say, the separated segments will be merged using the RANSAC method if they share a common orientation and distance to the origin. For example, the steps in a stair-case structure are often detected as one plane aligned with the steps. Hence, RANSAC is usually combined with other detection or refinement methods to detect planar surfaces. In [114], RANSAC and the Minimum Description Length (MDL) principle have been integrated to detect planes in point cloud data. Firstly, all points were partitioned into small rectangular blocks and RANSAC was carried out in each block. Then after detecting all possible planes, MDL was utilized to reduce the over-

fitting caused by RANSAC. Targeted to tackle the over-simplified problem of RANSAC, in [115], data point normal coherence checking was applied on all of the inlier patches within one fitted plane and the points with contradictory normal direction with respect to the fitting plane were removed. Later, the separated inlier patches were clustered recursively until all planes were extracted.

Hough Transform-based methods

The Hough Transform is well-known for parameterized object detection, typically for detecting lines and circles in 2D data sets [116]. Aiming to extend its usage to 3D space and also to reduce its computational cost, numerous variations have been proposed.

The 3D Hough Transform proposed by Hulik *et al.* [111] describes each plane by its slope along x and y axes and the distance to the origin of the coordinate system. Hence, each point in the corresponding 3D Hough space (θ, ϕ, ρ) represents one plane in 3D space and each point in 3D space represents one sinusoid curved surface in 3D Hough space. Therefore, a plane in 3D space could be represented by the intersection point of all corresponding sinusoid curved surfaces in 3D Hough space. In order to find this intersection point, each data point in 3D space casts its vote in the Hough Transform parameter space. The accumulator cells with the largest number of votes, which represent the Hough Transform parameters for one fitting model are identified as the parameters for the final optimal model. However, this kind of voting means that the Hough Transform method suffers from a high computational cost in finding the parameters of one fitting model when a large data set is input as well as sensitivity to the accumulator design. Different from the classic Hough Transform, the Randomized Hough Transform avoids the high computational cost of the voting process, instead, for every pixel in the image, it calculates the model parameters in a probabilistic way. Based on the properties of the Kinect depth camera, Dube *et al.* presented a PSD method by applying a Randomized Hough Transform on the depth map [117], which makes the plane detection realized in real time. For a more comprehensive review of the Hough-based methods on plane detection refer to [118].

Region growing based methods

Compared with the RANSAC and Hough Transform methods, region growing methods are faster, especially in the presence of many planes. Similarly, region growing based

PSD methods are also robust to noise and can efficiently detect large planes.

In [119] and [120], a two-point-seed growing algorithm was proposed to detect planar surfaces. The algorithm starts from a region G which consists of a random point p and its one nearest-neighbor from the point cloud data. Then the region G extends outwards by adding its neighboring point p_n to G if it satisfies that 1) the distance between p_n and the region G; 2) the plane-fitting error of p_n to G; 3) the distance between p_n and the new formed plane of $G \cup p_n$ are less than three corresponding thresholds, respectively. This region-growing process continues until no more points may be added to G. By taking the centroid and covariance matrix of the previous growing region into account, the plane parameters are incrementally updated. Following a similar growing approach, in [121] instead of incrementally computing the covariance matrix to derive a plane normal from it, the normals for all points and local surfaces are computed directly to obtain an estimate of the plane's normal. Therefore, after every growth only the centroid of the growing region is updated and stored in normal space, which further reduces the computation in comparison to [119]. Xiao *et al.* proposed a cached-octree region-growing algorithm to segment each point cloud into planar segments [122] [123]. Since the method stops region-growing merely based on distance information, over-extraction may occur at the intersection of two planes. An accurate and fast region growing algorithm for detecting regions was presented in [124], where the normal of each point in the noisy point-cloud data was used to select the seed with highest local planarity. Then the 26 voxel neighboring points were checked during the growing stage.

5.1.2 Depth Map Super-Resolution

To successfully adopt depth maps in 3D applications, several kinds of depth map SR techniques have been proposed aimed at increasing the spatial resolution of the depth maps. They can be summarized into three categories: filter-based methods, MRF-based methods and planar-surfaces-based methods.

Kopf *et al.* proposed upsampling the LR depth map by using a Joint Bilateral Upsampling (JBU) filter. Aided by the associated HR texture, the edges of the upsampled depth map can be well preserved [74]. A similar but advanced joint bilateral filtering technique was proposed in [75], which iteratively refines the input LR depth map by referring to the registered HR textures. On one hand, the adoption of texture information can help to obtain sharp depth edges. However, on the other hand, the color or lighting

variations on the same areas of the texture images can cause false discontinuities in HR depth maps. Hence, the texture images need to be used in a more sophisticated way. By applying the MRF to super-resolve the LR depth map, Diebel *et al.* formulated the SR process as an energy minimization problem to fuse LR depth images and HR texture images. Different from Diebel's work, in [125], a NLM term was used in the MRF to preserve the edges. However, during the optimization process of the MRF-based methods, the estimation errors are easily propagated into the obtained HR depth map. Lo *et al.* [76] proposed the incorporation of a texture-guided weighting factor into the MRF model to reduce the texture copying artifacts and the weighting factor was obtained based on a learning approach. Although the demonstrated results were good, the learning-based approaches usually have a higher computational complexity, which might prevent their adoption for real-time applications.

Since after projection, the objects in a 3D scene can be represented by several planar surfaces with different shapes in a 2D image, each planar surface will have linearly changing depth values in the corresponding depth map and the boundaries of surfaces indicate the discontinuities of the depth values. If an equation can be obtained for each surface, the SR of LR depth map can be obtained by inserting pixels based on this equation. Therefore, the whole depth map can be classified into three categories: planar surfaces, non-planar surfaces, and edges. In the work in [126], the SR of depth map relied on the local planar hypothesis and the candidates of potential HR depth values were obtained by either linear interpolation along horizontal and vertical directions or the estimated local planar surface equations. However, since the surface equation was evaluated locally, it may be biased by the noise contained in the local pixels which later on will magnify the estimated error in the generated HR depth map. Therefore, to address the above problem, in this chapter, the global analytical equations of the detected surfaces are used and for each of these three categories a proper up-sampling approach is proposed to exploit its intrinsic properties.

5.2 Proposed Planar Surface Detection Method

The proposed indoor PSD method consists of two stages. The first stage of the proposed indoor planar surface detection method aims to detect valid seed patches over the whole depth map. This is an important step since the depth data may have holes or anomalous points. Then, the valid seed patches according to their planarity will be used as the

starting elements of the growing process. When no more new neighbor points fit into the current planar surface, the current growing process stops and next seed patch begins to grow. Finally, two post-processing approaches are proposed to tackle the overgrowing surface problem and to merge separated surfaces. These two post-processing stages make the proposed method robust to various testing conditions.

Fig.5.1 shows the framework of the proposed DPSD.

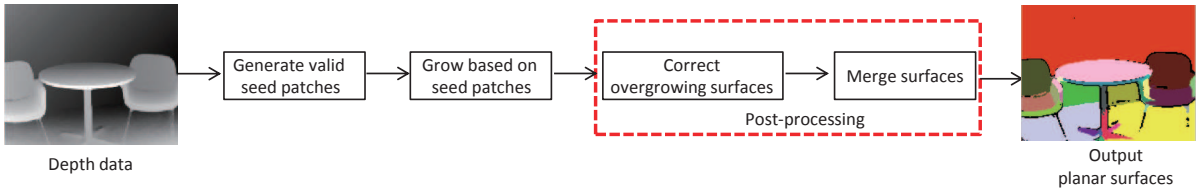


Figure 5.1: The framework of the proposed depth map based planar surface detection method.

5.2.1 Generating Valid Seed Patches

Firstly some notations about plane representation are given. In the 3D space, a plane can be defined as $a_m x + b_m y + c_m z + d_m = 0$, where (a_m, b_m, c_m) defines the normal vector $\hat{\mathbf{n}}_m$ of the plane and d_m is the distance from the 3D space origin. According to this notation, the Hessian form of the plane, S_m, could be written as $\hat{\mathbf{n}}_m \cdot \mathbf{p} + d_m = 0$, where $\mathbf{p} = (x, y, z)$ indicates an arbitrary point on the plane, and the operation " $\cdot$ " stands for the dot-product of two vectors. The distance between this surface and a point $\mathbf{p}_i$ (or the fitting error) could be evaluated as [127]:

$$e(\mathbf{p}_i) = |\hat{\mathbf{n}}_m \cdot \mathbf{p}_i + d_m| \tag{5.1}$$

Furthermore, the mean square fitting error of a set of points S_k with respect to the surface S_m is defined as:

$$\delta_{m,k} = \sqrt{\frac{1}{|S_k|} \sum_{\forall \mathbf{p}_i \in S_k} (\hat{\mathbf{n}}_m \cdot \mathbf{p}_i + d_m)^2} = \sqrt{\frac{1}{|S_k|} \sum_{\forall \mathbf{p}_i \in S_k} e^2(\mathbf{p}_i)} \tag{5.2}$$

To ensure that the proposed growing process is based on reliable seed patches, a sliding $L \times L$ square window is moved in raster-scan fashion over the whole depth map, and at each position all covered pixels will be checked. Patches with no anomalous measurements and no holes will be regarded as *valid seed patches*. Then, the linear least squares plane fitting approach is applied to each valid seed patch to find the best fitting plane that could represent it.

Each valid seed patch is denoted by $\psi_m(f_m, \delta_{m,m})$, or for brevity ψ_m, with m being the seed index. The function f_m is the estimated equation representing the plane that best fits the seed patch, and it is given by $\hat{\mathbf{n}}_m$ and d_m. Whereas, $\delta_{m,m}$ is the mean square fitting error of the seed's points with respect to the estimated plane, and it can be evaluated using (5.2), to simplify the notation this will be represented by δ_m.

5.2.2 Growing Process

This subsection explains the iterative growing process of a planar surface starting from a seed patch. Firstly, some notations that will be used need to be introduced. The index j will be used as superscript to indicate the iteration index of the iterative growing process, so for example the planar surface i at stage j of the growing process will be represented by S_i^j or $S_i^j(f_i^j, \delta_i^j)$. In the latter form the function f_i^j is the estimated equation of the plane at the end of the j-th stage of the growing process, and it is given by $\hat{\mathbf{n}}_i^j$ and d_i^j. The mean square fitting error of the surface's points with respect to the estimated fitting surface is δ_i^j, and it could be evaluated using (5.2). The ground-truth surface of the i-th surface will be represent by $\bar{S}_i$.

Different to the RANSAC method whose seeds are selected randomly, in the proposed growing stage, the previously obtained seed patch candidates will be initially arranged in ascending order of their mean square fitting error in a *growing seed list* Ψ_1, thus $\Psi_1 = \{\forall \psi_n, \psi_m \in \Psi_1 : \delta_n \leq \delta_m; n < m\}$. The first seed patch appearing in the list will be used to initiate the first surface. Once this surface reaches its maximum extent then its growing process will stop and the growing seed list will be updated to Ψ_2 by eliminating all the seed patches that are enclosed within the first detected planar surface. This updating process will be carried out at the end of the growing process for each planar surface, S_i^j, to generate a new seed growing list $\Psi_{i+1} = \{\forall \psi_m \in \Psi_i : \psi_m \notin S_i\}$. This ensures that only non-incorporated seed patches will be used in the subsequent growing of other planar surfaces.

At this point the growing process of the plane i will be described. This plane at its initial stage is merely defined by its seed patch, thus $S_i^0(f_i^0, \delta_i^0) = \psi_m(f_m, \delta_m)$, with ψ_m being the first seed patch in Ψ_i. At the j-th iteration stage of the growing process the neighbors $\bar{N}_i^j$ of the surface S_i^j will be firstly identified. Then points belonging to other planar surfaces will be excluded from $\bar{N}_i^j$ to obtain a new set $N_i^j = \{\mathbf{p} : \mathbf{p} \notin S_m, m < i\}$. This set of points will be plugged into the current plane equation, i.e. f_i^j, to evaluate

their fitness to this plane. So the points with fitting error larger than the threshold, T^j, will be regarded as outliers to this planar surface. Otherwise, they will be enclosed within the current plane to form S_i^{j+1}. This could be summarized by:

$$S_i^{j+1} \backslash S_i^j = \left\{ \forall \ \mathbf{p} \ \in N_i^j : e(\mathbf{p}) \leq T^j \right\} \tag{5.3}$$

Fig.5.2 shows an example of the first two growing steps (i.e., $j = 1$ and $j = 2$) of the plane i and its N_i^1 and N_i^2 neighboring sets. After each growing stage, the plane equation f_i^j will be refined to f_i^{j+1} by using the linear least square plane fitting approach over the whole set of pixels of the newly updated surface S_i^{j+1}.

The growing process for the surface i will be iteratively repeated until one of the following two halt conditions is met: a) the set N_i^j is empty, b) no point in the neighboring set N_i^j fits well into the current planar surface. These two conditions indicate that the i-th surface has grown to its maximum extent. Once the growing process stops then the surface i will be finalized and it will be represented hereinafter by $S_i = S_i(f_i^{k_i}, \delta_i^{k_i})$, where k_i represents the index of the last growing stage of this planar surface. $f_i^{k_i}$ is the final estimated equation of the surface and it is given by $\hat{\mathbf{n}}_i^{k_i}$ and $d_i^{k_i}$.

As previously described at the end of the growing process of the i-th planar surface, a new growing seed list, Ψ_{i+1}, will be generated and a new growing process will be initiated by using the first-ranked seed patch in the list. This growing process will be repeated until the updated seed list is empty.

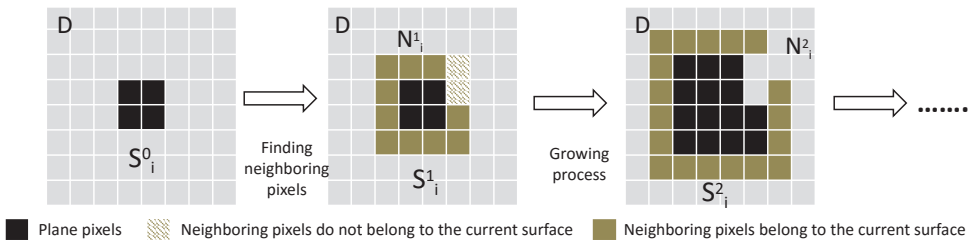


Figure 5.2: An example of the growing process of a planar surface; D is depth map, S_i^j is the current surface and N_i^j is current neighboring pixels.

In addition, it is worth noticing that the threshold value, T^j, which is used to determine the fitness of a pixel for the current growing surface, will be dynamically updated after each iteration of the growing process by increasing its value. That is to say, at beginning the requirement for enclosing neighboring pixels is very strict since these points will become the base of the following growing stages. With the growing of

the plane, more points need to be checked, so if the threshold is still small, then with high probability this will lead to over-segment the whole depth map into many small planar surfaces. In this work, the following equation for T^j has been adopted

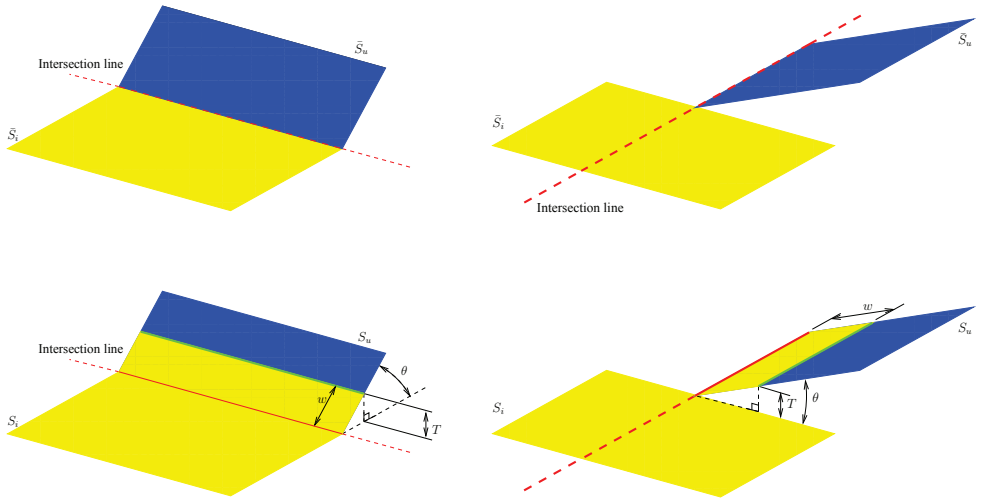
$$T^j = \tau(1 - e^{-j/\lambda}) \tag{5.4}$$

where τ is the maximum allowed "roughness" of the planar surface and λ is the changing speed of the threshold.

5.2.3 Post-processing of Detected Surfaces

Some of the detected surfaces suffer from two major problems. The first one is intersecting-surfaces-caused overgrowing problem. The second one is overgrowing-caused surface separation problem. So in the following work two post-processing stages have been proposed to tackle both problems.

Overgrowing surfaces correction



(a) (b)

Figure 5.3: Two typical cases of overgrowing surfaces: (a) lateral-OGS; (b) axial-OGS; the lateral points and medial points of the OGS are shown in green and red, respectively.

The proposed planar surface detection method uses a growing-based approach, which causes intersecting surfaces to experience "overgrowing" problem. This happens because from geometric point of view, the intersection line belongs to both intersecting surfaces. Thus the Overgrowing Surface (OGS) problem can occur: (a) along the lateral side of the intersection line (noted as lateral-OGS in this chapter), and/or (b) along the

(a) (b)

Figure 5.4: (a) and (b) are the examples of daily life scenes with OGS problem; red lines show the intersection lines of the two hashed surfaces.

intersection line (noted as axial-OGS in this chapter). Some graphic examples of the former and latter cases are shown in Fig.5.3 (a) and (b) respectively and two examples of daily life scenes which causes the OGS problem are shown in Fig.5.4. Practically, the extent of this problem depends on several factors, such as the geometry of the scene, the angle between the two intersecting surfaces, the accuracy of the depth data, the order of seeds used, and finally the threshold value T^j. In the following, suppose that the ground truth surface $\bar{S}_i$ and $\bar{S}_u$ intersect in a line segment (as shown in Fig.5.5 (a)), with the red line representing the intersecting line. Furthermore, the symbol S_o will be used to represent the subsurface defined by the neighboring points of the intersection line, which equally fits well into both surfaces. To have a formal representation of S_o, assume for simplicity, that the threshold given in (5.4) does not change with j, i.e, $T^j \approx T$, and that the depth data is accurate, consequently $\hat{\mathbf{n}}_i \approx \hat{\mathbf{n}}_i^{k_i}$, $d_i \approx d_i^{k_i}$, $\hat{\mathbf{n}}_u \approx \hat{\mathbf{n}}_u^{k_u}$ and $d_u \approx d_u^{k_u}$. In this case the subsurface S_o can be described as:

$$S_o = \left\{ \forall\ \mathbf{p}\ \in S_o : \left(\left|\hat{\mathbf{n}}_i^{k_i} \cdot \mathbf{p} + d_i^{k_i}\right| \leq T\right) \bigwedge \left(\left|\hat{\mathbf{n}}_u^{k_u} \cdot \mathbf{p} + d_u^{k_u}\right| \leq T\right) \right\} \tag{5.5}$$

Thus S_o could be enclosed within S_i or S_u. So, for example if the first surface to grow was S_i then S_o will be incorporated into it, and vice versa. For a scene with a large number of intersecting surfaces, it is reasonable to expect that half of the subsurfaces will be mistakenly assigned to the wrong surfaces. The overgrowing surfaces have negative impacts on the results obtained by the growing-based planar surface detection, as can be seen in Fig.5.5 (b), where the overgrowth of S_i ends up splitting the wall into two surfaces. By referring to Fig.5.3 it is possible to prove that the width of the OGS

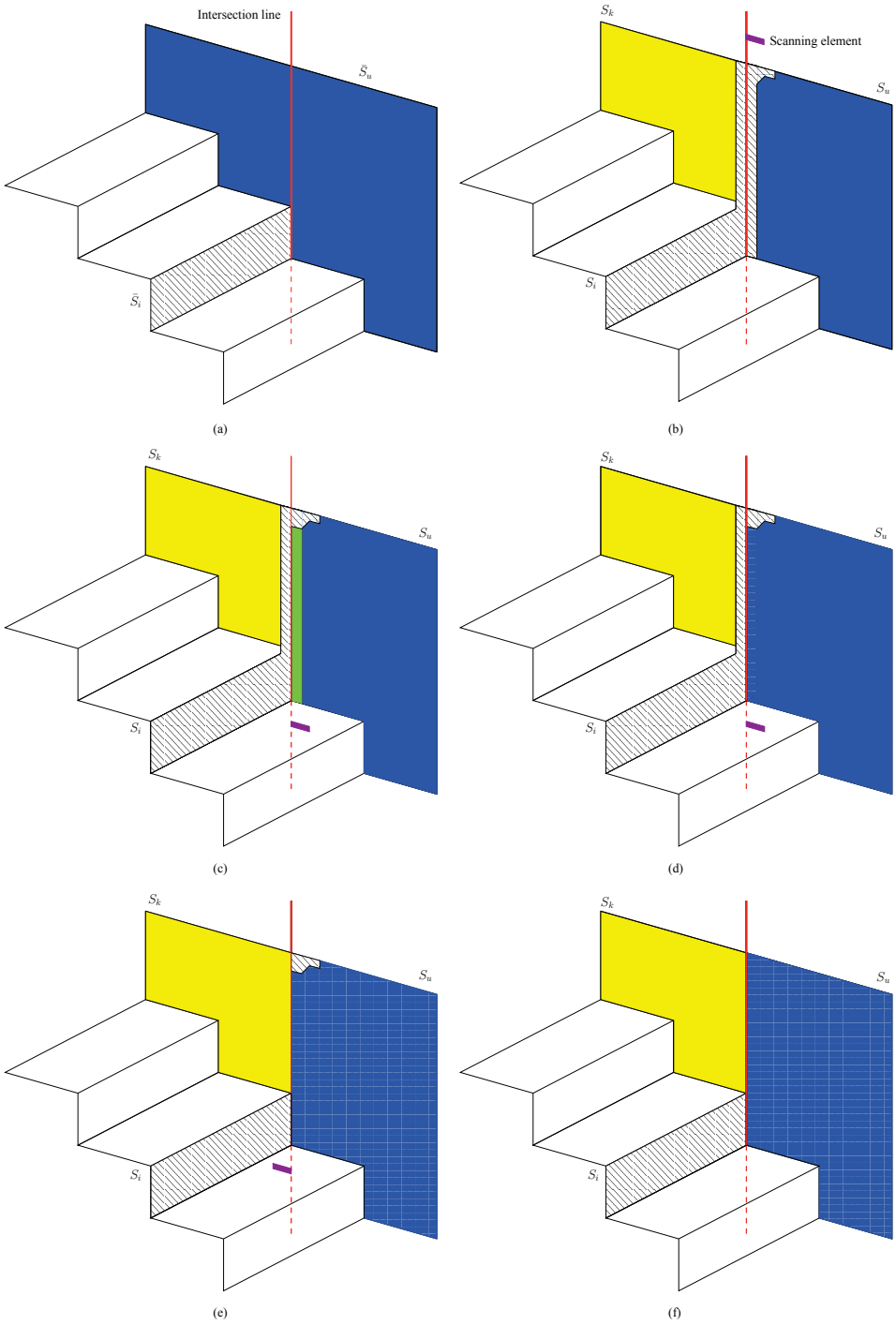


Figure 5.5: (a) shows two intersecting surfaces $\bar{S}_i$ and $\bar{S}_u$; (b) the overgrowing surface S_i splitting $\bar{S}_u$ into two surfaces; the scanning element is shown in violet; (c) the detected shared surface (S_o) is shown in green; (d) the outcome of relocating the shared surface when processing S_i and S_u; (e) the outcome of relocating the shared surface when processing S_i and S_k; (f) the outcome of fragmented element relocation and finalizing surface S_i.

is given by

$$w = \frac{T}{\sin\left(\arccos(\hat{\mathbf{n}}_i^{k_i} \cdot \hat{\mathbf{n}}_u^{k_u})\right)} \tag{5.6}$$

Consequently, the smaller the angle between the two surfaces is, and the larger the threshold is, the wider the OGS is. When the two surfaces are perpendicular then $w = T$, which is the smallest width of the OGS.

To achieve the goal of the first post-processing stage the detected surfaces will be processed sequentially. So assuming that the current surface to be processed is S_i (noted as the primary surface), then all its neighbors (noted as secondary surfaces) which intersect with it need to be determined. If the intersection segment between S_i and its neighbor, say S_u, is within the boundary of the image, then it is possible that S_i overgrows into S_u, thus this case needs to be investigated. Moreover, if the intersection segment lies outside of the image boundary then S_i cannot overgrow into S_u, consequently this case will be skipped and the next neighbor will be examined. The intersection line between S_i and S_u should be evaluated to determine which of these two cases happened. Thus, the intersection line could be written in a parameterized form with respect to t using the equations of the two planes[1], as [128]:

$$\mathbf{p} = (\hat{\mathbf{n}}_i^{k_i} \times \hat{\mathbf{n}}_u^{k_u})\, t + \mathbf{p}_0 \tag{5.7}$$

where the operation "×" stands for the cross-product of two vectors. Whereas the point $\mathbf{p}_0$ on the intersection line is obtained by

$$\mathbf{p}_0 = \frac{\left(d_u^{k_u}\ \hat{\mathbf{n}}_i^{k_i} - d_i^{k_i}\ \hat{\mathbf{n}}_u^{k_u}\right) \times \left(\hat{\mathbf{n}}_i^{k_i} \times \hat{\mathbf{n}}_u^{k_u}\right)}{|\hat{\mathbf{n}}_i^{k_i} \times \hat{\mathbf{n}}_u^{k_u}|^2} \tag{5.8}$$

At this point it is possible to determine the location of the intersection segment, which will be assumed for the surfaces S_i and S_u to be within the boundary of the image.

Since the OGS lies on the intersection line, and its shape depends on several factors, in order to accurately detect it, a line-shape scanning element is proposed to perpendicularly scan the intersecting surfaces along their intersection line, an example of a scanning element is shown in violet in Fig.5.5 (b). If S_i overgrows into S_u then, in general, the elongated OGS will have a width w given by (5.6), furthermore, it will be surrounded on at least one side by S_u. These two properties will be used to check for shared subsurfaces S_o when using the proposed line-shape scanning approach. Consequently, the length of this scanning element will be chosen to be $\lceil w \rceil (1 + \epsilon)$, with $\lceil x \rceil$

[1] Although the detected planar surfaces are shown in a 2D image, their estimated equations represent them in 3D space.

is the nearest integer to x that's not smaller than x, and $\epsilon > 0$ so as to ensure that the scanning element is longer than the width of the shared subsurface, thus it could cover pixels at both S_o and S_u. The scanning starts at one extreme of the intersection line within the image boundary. When the scanning element first encounters an S_i with S_u neighbor then a new S_o is initiated. This area will continue to grow as pixels belonging to S_i and surrounded by S_u are found. The growing of S_o will halt once the scanning element no longer covers the two surfaces. The detected S_o in addition to the scanning element are shown in green and violet, respectively, in Fig.5.5 (c).

If the subsurface S_o has been mistakenly enclosed within S_i instead of S_u then its medial points (i.e., the points along the intersection line) will theoretically have zero fitting error with respect to S_i and S_u. Consequently, these points cannot be used to judge where S_o should be allocated. In contrast, the lateral points of S_o, being the furthest from the intersecting line, will have a large fitting error with respect to the S_i surface, and small with respect to S_u (the lateral points and medial points of S_o are shown in green and red, respectively in Fig.5.3). Denoting S_L as the lateral points of S_o, the following criteria will be used to reallocate it:

$$S_o \subset \begin{cases} S_u & ; \quad \delta_{u,L} < \delta_{i,L} \\ S_i & ; \quad \delta_{u,L} \geq \delta_{i,L} \end{cases} \tag{5.9}$$

The surface which minimizes the mean square fitting error of the lateral points of S_o will end up enclosing it; in this example the outcome of the shared surface relocation is shown in Fig.5.5 (d). Then the next neighboring surface of S_i, say S_k, will be checked for a valid intersection, and the process described for S_u will be repeated for S_k. An example of the outcome of this is shown in Fig.5.5 (e). It is worth noting that, at the end of the relocation process, the surface S_i may end up becoming fragmented and scattered into a main body and isolated areas. Thus, each isolated area will be compared with all of its neighbors to determine which one is more suitable to enclose it. The outcome of this is shown in Fig.5.5 (f). The minimization of the mean square fitting error is the criteria to decide where the isolated area should be enclosed. At the end of this step all equations representing the affected surfaces will be updated, and the surface S_i will be finalized. Then another surface will be regarded as the primary surface, and the previously described post-processing procedure will be carried out, until all surfaces are checked.

Fig.5.6 shows the flowchart of the proposed approach for OGS detection and relo-

cation.

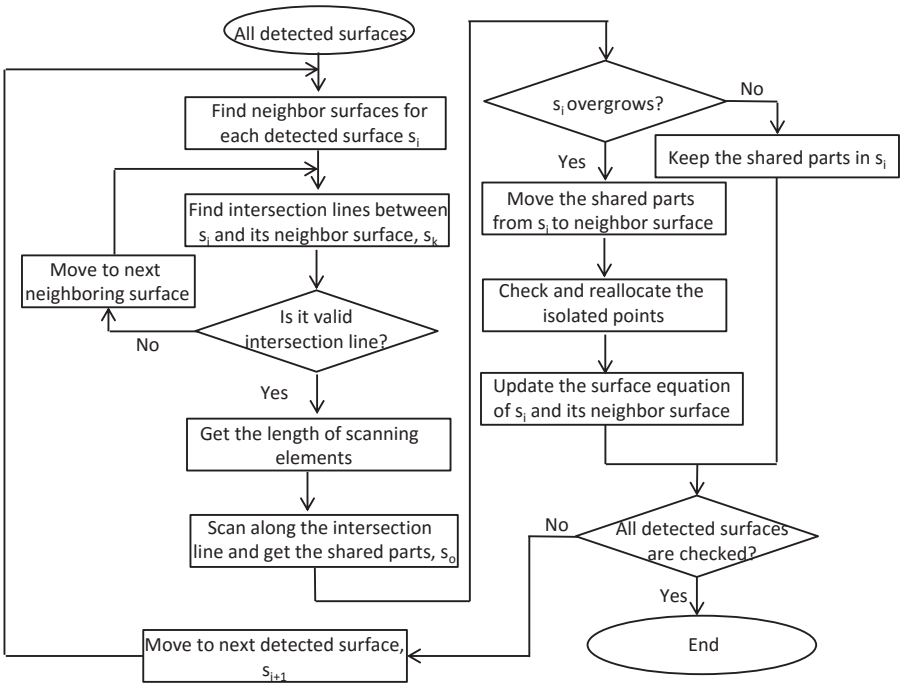


Figure 5.6: The flowchart of the proposed OGS algorithm.

Surfaces merge

In the proposed approach some surfaces end up being wrongly separated into two or more different parts. This problem can be caused by several issues. The first cause is the OGS problem, where the intersection area separates one surface into several parts, as shown in Fig.5.5 (f). The second is due to the barrel distortion phenomenon affecting the depth data, from the capturing optics of the depth camera.

In order to tackle the over segmentation problem the second post-processing stage is proposed, which exploits the estimated equations of the planar surfaces to merge the neighboring surfaces that are on the same plane. Two neighboring surfaces are on the same plane if they are parallel and at equal distance from the origin, these two conditions will be used in the proposed post-processing approach. If two surfaces, say S_i and S_u, are parallel then $\arccos(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_u) = 0^\circ$. To take into account the estimation error of the surface equation the previous condition will be relaxed, consequently the surfaces S_u could be regarded as parallel to S_i if:

$$\theta_{i,u} < \Delta\theta_i \tag{5.10}$$

with $\theta_{i,u} = |\arccos(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_u)|$, i.e., the angle between the surface S_i and S_u, and $\Delta\theta_i$ is the angle between the norm of the surface S_i, i.e., $\hat{\mathbf{n}}_i$, and its worst estimate:

$$\Delta\theta_i = \underset{\left\{\hat{\mathbf{n}}_i^j : j < k_i\right\}}{\operatorname{argmax}}\left(\left|\arccos(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_i^j)\right|\right) \tag{5.11}$$

with $\hat{\mathbf{n}}_i^j$ being the estimated norm of the plane at the end of the j-th iteration of the growing process. As described in Section 5.2.2, at each iteration of the surface growing process, the plane equation will be refined to better fit all the enclosed points, hence, it is reasonable to assume that the final estimated plane equation is the most accurate, thus (5.11) will be rewritten with the assumption that $\hat{\mathbf{n}}_i \approx \hat{\mathbf{n}}_i^{k_i}$. The bigger the surface is, the more accurate the previous assumption is. Consequently, the first step of the proposed merging stage is to arrange all of the surfaces in a surface list $\mathbb{S}$ in descending order according to their size. The list will be $\mathbb{S} = \{S_i; 1 \leq i \leq s_N\}$, with N being the total number of detected surfaces.

Then all of the combinations of the angle $\theta_{i,u}$ will be evaluated and arranged in the upper triangular combinational matrix Θ as follows:

$$\Theta = \begin{pmatrix} 0 & \theta_{1,2} & \theta_{1,3} & \theta_{1,4} & \cdots \\ 0 & 0 & \theta_{2,3} & \theta_{2,4} & \\ 0 & 0 & 0 & \theta_{3,4} & \\ 0 & 0 & 0 & 0 & \\ \vdots & & & & \ddots \end{pmatrix} \tag{5.12}$$

At this point each entry in the i-th row, say $\theta_{i,j}$, will be compared with $\Delta\theta_i$ according to (5.10) to determine whether the surface S_j is parallel to S_i. All of the surfaces that are parallel will be grouped into the same surface group. It is worth noticing that the rows and columns of Θ are arranged in descending order with respect to the surface size, this ensures that smaller surfaces will be checked against bigger ones.

To verify that two surfaces can be merged we need to ensure that they are on the same plane. To do this assume that S_i and S_j form a pair of parallel planes, and that the size of S_i is smaller than S_j. Then, the mean square fitting error of S_i with respect to the equation of the surface S_j is calculated using (5.13) as $\delta_{j,i}$. If this value is smaller than the mean square fitting error of the surface S_j then this pair of surfaces, i.e., S_i and S_j, will be regarded as being on the same plane. Furthermore, if these two surfaces are also contiguous, then S_i will be merged into S_j surface. At the end of the merging step the equation representing enlarged S_j surface will be updated.

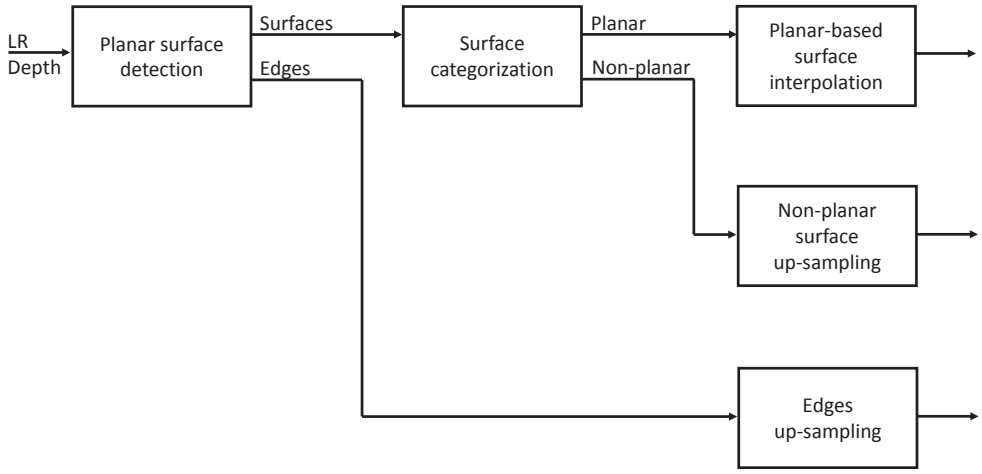


Figure 5.7: The framework of the proposed depth map super-resolution method.

5.3 The Proposed Depth Map SR Method

In this section the proposed depth map SR will be explained. If the analytical equation of a surface in the scene is known then it could be used to up-sample this surface, by plugging the coordinate of each missing pixel (x, y) in the surface equation to find its corresponding depth value z. So aided with this paradigm, in the proposed approach, the surfaces in the depth-scene will be categorized into three groups, namely: planar surfaces, non-planar surfaces, and finally edges. For each of these three categories a proper SR approach will be devised to better exploit its intrinsic properties.

Since the output of the DPSD method is mainly surfaces and the surrounding edges, the edges category will simply be the one outputted by the DPSD method, whereas, in the second stage of the proposed approach the detected surfaces will be categorized into planar surfaces and non-planar surfaces. The block diagram of the proposed approach is shown in Fig. 5.7. The idea behind the proposed categorization mechanism is to check if the estimated planar equation fits the measured depth values of the surface well. So the pixels of each detected surface $\mathcal{S}_m$ will be plugged into the surface equation so as to evaluate their estimated depth values (i.e., $\hat{z} = \frac{-1}{c_m}(a_m x + b_m y + d_m)$, where $\mathbf{p} = (x, y)$ is a pixel belonging to the surface $\mathcal{S}_m$. Then the Mean Square Fitting Error (MSFE) is evaluated for this surface as:

$$\delta_m = \sqrt{\frac{1}{|S_m|} \sum_{\forall \mathbf{p} \in S_m} (z - \hat{z})^2} = \sqrt{\frac{1}{|S_m|} \sum_{\forall \mathbf{p} \in S_m} (z + \frac{1}{c_m}(a_m x + b_m y + d_m))^2} \qquad (5.13)$$

This value will be compared against a threshold related to the maximum roughness

value used in the planar surface detection unit. If the MSFE is larger than the threshold then this indicates that this surface is non-planar and/or the estimated surface equation is not accurate. In this case, this surface will be classified as a non-planar surface. Otherwise it will be classified as planar. The details of the SR approach for each of the three categories of pixels will be explained hereinafter.

5.3.1 Super-Resolution Process

For each set of pixels belonging to a planar surface the surface equation will be used to estimate the values of the pixels to-be-filled. Although a planar surface could be easily up-sampled by using a first-order linear interpolator, given that the measured depth data is affected by measurement noise, this will affect the accuracy of the interpolator-based up-sampled pixels. Thus, it is more accurate to use the estimated equation of the plane to up-sample it. To prove this property, firstly the analysis is simplified by modeling the high resolution version of the depth data using a one-dimensional vector $\bar{\mathbf{Z}}_h = [\bar{z}_0, \ldots, \bar{z}_i, \ldots, \bar{z}_{N-1}]^T$, and let us consider the case where N is even. The depth data measured by the depth-camera will be represented by the vector $\mathbf{Z}_l = \bar{\mathbf{Z}}_l + \mathbf{N}$ where $\bar{\mathbf{Z}}_l = [\bar{z}_0, \ldots, \bar{z}_{2i} \ldots, \bar{z}_{N-2}]^T$ is the low resolution, or in other words the down-sampled version of $\bar{\mathbf{Z}}_h$ where the downsampling factor is 2. The vector $\mathbf{N} = [n_0, \ldots, n_{2i}, \ldots, n_{N-2}]^T$ is a zero mean iid random process representing the measurement noise affecting depth data, and its variance is σ_n^2.

Let $\mathbf{Z}_z$ represent the zero-filled version of $\mathbf{Z}_l$ where the measured samples in $\mathbf{Z}_l$ have been separated by inserted zeros, i.e., $\mathbf{Z}_z = [z_0, 0, \ldots, 0, z_{2i}, 0, \ldots, z_{N-2}, 0]^T$. This version will be used as the basis to generate the final super-resolved version of the vector $\mathbf{Z}_l$ by filling the zeros using super-resolved values. If a first order linear estimator is used to estimate the zero-fill position in $\mathbf{Z}_z$ starting from its two-side neighbors then it could be written that:

$$\hat{\mathbf{Z}}_h = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ \alpha_1 & 0 & \beta_1 & 0 & 0 & \\ 0 & 0 & 1 & 0 & 0 & \\ 0 & 0 & \alpha_3 & 0 & \beta_3 & \\ \vdots & & & & & \ddots \end{pmatrix} \mathbf{Z}_z \tag{5.14}$$

where $\hat{\mathbf{Z}}_h$ is the super-resolved version of the vector $\mathbf{Z}_l$. The variance of the estimation

error of, for example, z_{2i+1} can be evaluated as:

$$\sigma_e^2(2i+1) = E\left\{(\alpha_{2i+1}z_{2i} + \beta_{2i+1}z_{2i+2} - \bar{z}_{2i+1})^2\right\} \quad (5.15)$$

$$= E\left\{\left((\alpha_{2i+1}\bar{z}_{2i} + \beta_{2i+1}\bar{z}_{2i+2}) + (\alpha_{2i+1}n_{2i} + \beta_{2i+1}n_{2i+2}) - \bar{z}_{2i+1}\right)^2\right\} \quad (5.16)$$

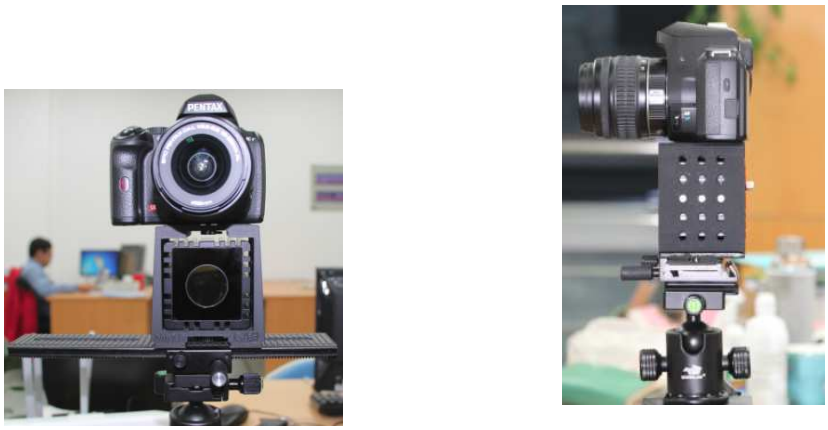
if it is supposed that $\bar{z}_{2i}$, $\bar{z}_{2i+1}$ and $\bar{z}_{2i+2}$ are the depth distances from a planar surface then in this case $\bar{z}_{2i+1} = \frac{\bar{z}_{2i}+\bar{z}_{2i+2}}{2}$, which means the best estimate of $\bar{z}_{2i+1}$ can be obtained when $\alpha_{2i+1} = \beta_{2i+1} = 1/2$. In this case, if we take into account the assumption that the noise $\mathbf{N}$ is a random iid process with zero mean value then (5.15) can be simplified as:

$$\sigma_e^2(2i+1) = \frac{1}{4}E\left\{(n_{2i} + n_{2i+2})^2\right\} = \frac{1}{2}\sigma_n^2 \quad (5.17)$$

Obviously the accuracy of the up-sampling process depends on the accuracy of the depth measurement, and the error shown in (5.17) cannot be minimized by using traditional interpolators (such as linear, bicubic, etc.). On the other hand, by using the planar surface estimated equation for up-sampling the interpolation error is reduced. This is because the depth measurement noise is canceled out during the estimation of the planar surface equation.

After super-resolving all planar surfaces the non-planar surfaces will be up-sampled by exploiting the local structure by using a traditional interpolator. In this chapter, the Bicubic [129] interpolator is used for this task. Nevertheless, it is worth noticing that more advanced interpolators, such as a directional-based interpolator could be used to estimate the values of the pixels to-be-filled.

Once all planar and non-planar surfaces are up-sampled then edges and the remaining non-filled pixels are up-sampled. For each non-filled pixel its N_8 neighbors [130] and the surfaces that they belong to will be firstly identified. Then the missing pixel will be estimated by taking into account the surface category of each of its neighbours. To simplify the description of the proposed approach an example will be used hereinafter. Suppose that (x, y) are the coordinates of one of the non-filled pixels, and suppose that one of its neighbors, $\mathbf{p}_i$, belongs to a planar surface $\mathcal{S}_k$, then the surface equation of this surface will be used to estimate the depth value at (x, y), as follows $\hat{z}_i = \frac{-1}{c_k}(a_k x + b_k y + d_k)$. If $\mathbf{p}_i$ belongs to a non-planar surface $\mathcal{S}_n$ then the same traditional approach which was used to up-sample $\mathcal{S}_n$ will be used to extrapolate it and evaluate the depth value at (x, y). After obtaining, for each neighbor of the pixel (x, y) an estimated version of the to-be-filled pixel, these estimated versions will be fused by using a weighted average approach.



(a) (b)

Figure 5.8: The capturing texture and depth camera platform: (a) front view; (b) side view

5.4 Experimental Results

Since the performance of the proposed depth map upsampling approach is affected by the accuracy of planar surface detection, in this section both of these two methods are evaluated.

To evaluate the performance of the proposed DPSD approach, different indoor scenes were tested[2]. For each scenario, the texture and its corresponding depth map are captured by using a PENTAX K-R camera [131] and the SwissRanger SR4000 camera [45]; the composite camera structure is shown in Fig.5.8. Although the texture and depth data are slightly misaligned due to the intrinsic structure of the capturing platform, the texture data is only used to visually assess the planar surface detection results. In the following experiment, the RANSAC planar surface detection approach is used as a benchmark. It is worth noticing that the results of RANSAC are affected by the initially and randomly selected points, thus for the following comparisons, the comparatively best RANSAC result for each scene has been reported. The tested texture images and their associated depth maps are shown in the top left and right images, respectively, of each pane in Fig.5.9. From this figure it can be seed that the four scenes have various levels of texture and depth complexity. For example, the first scene, i.e.,

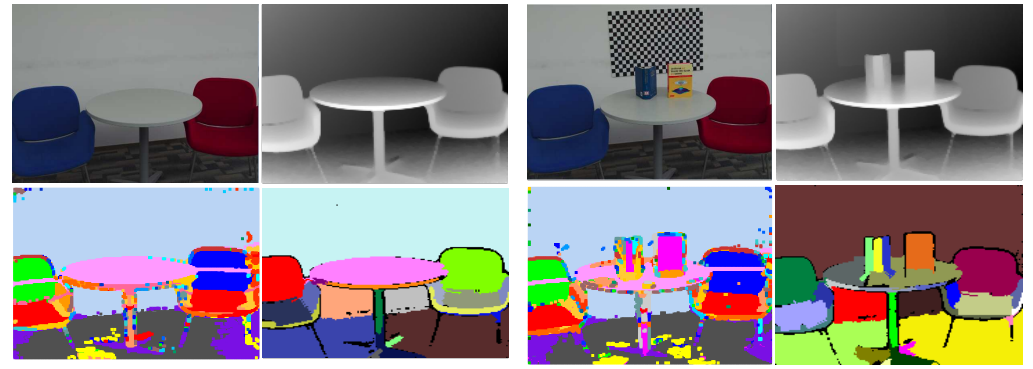
[2]All the testing images with their setting information are available at http://www.mmtlab.com/DDPSD.ashx.

Chairs & Table 1, has a table and wall with uniform textures. More complex than the scene *Chairs & Table 1*, in *Chairs & Table 2* and *Seat & Table*, there are some books on the table, and various kinds of chair. The aim of this type of arrangement is to assess the ability of the proposed DPSD method to distinguish different planar surfaces forming complicated objects. It is worth noticing that in *Chairs & Table 2*, a chessboard is hanging on the wall, thus if texture-based PSD is used then it will have a problem in detecting the wall's surface. On the contrary, this problem can be avoided by using depth-based PSD. Finally, the scene *Cabinet* is used to test a more challenging scenario, due to the presence of many small objects, transparent glass and complex combinations of vertical and horizontal surfaces. The scenes *Stairs 1* and *Stairs 2* are the front and side view of a stair-step structure.

For the four scenes, square seeds of 4×4 pixels were used, as for the threshold parameters the maximum allowed roughness, i.e., τ, was set to 3 and the threshold change speed, λ was set to 1. As for the benchmark method, the maximum number of iterations per surface was set to 5000 which is sufficient for the detection process [111] and the threshold was set at 7.26.

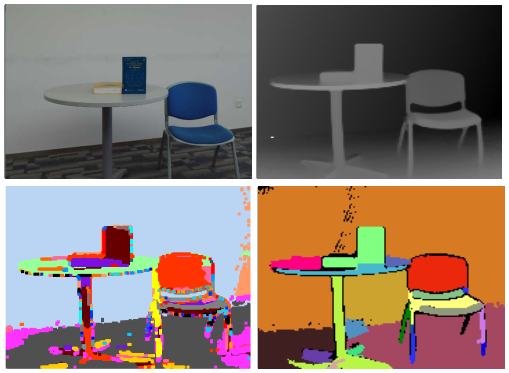
The bottom left and right image of each pane in Fig.5.9 shows the obtained results of the benchmark method and the proposed DPSD method, respectively, for each of the four tested scenes. The detected planar surfaces are represented with different colors. It is easy to notice that the proposed method can detect the majority of planar surfaces, and more accurately than the benchmark approach. The proposed method can even distinguish each planar surface of objects which consist of multiple planar surfaces, for example the chairs in *Chairs & Table 1*, the blue book in *Chairs & Table 2* and the office seat in *Seat & Table*. Furthermore, most of the planar surfaces in the complex structure in *Cabinet* are successfully detected by the proposed DPSD, without being over-simplified. It is worth noticing that even some small objects on the cabinet table have been detected, which demonstrates the effectiveness of the proposed approach.

The threshold updating mechanism allows the estimated equations of the detected surfaces to be refined. This in turn allows the proposed method to correctly detect curved surfaces with limited curvature, for example, the top-board side surface of the table in *Chairs & Table 1* and *Seat & Table*. However, for the chairs' base in Scene *Chairs & Table 1* and *Chairs & Table 2* with large curvature, they will be detected as combined multiple surfaces rather than one curved surface.

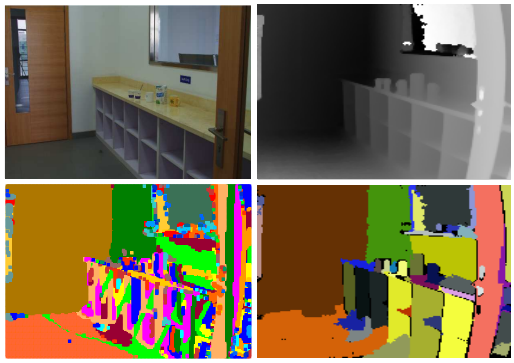


(a) Chairs & Table 1

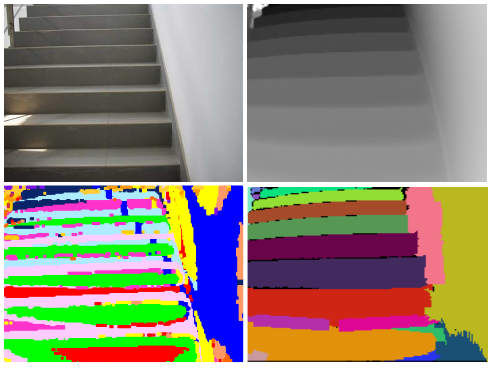
(b) Chairs & Table 2



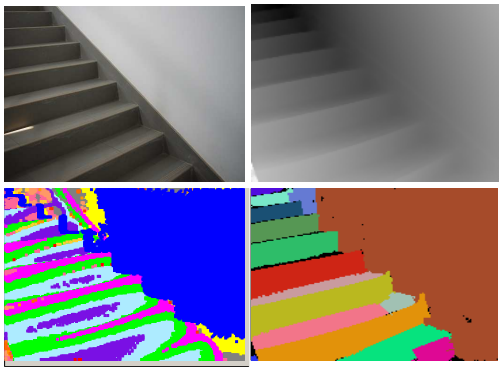
(c) Seat & Table



(d) Cabinet

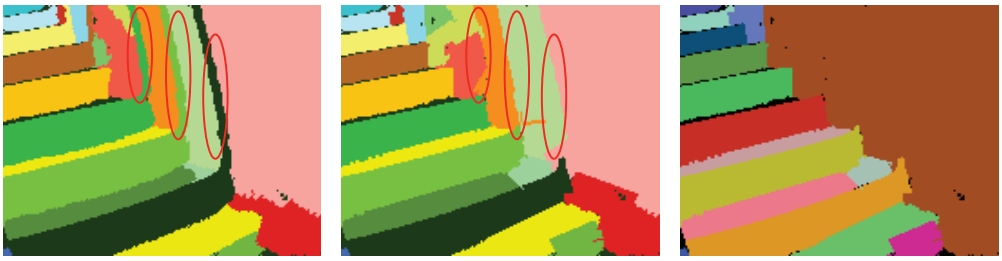


(e) Stairs 1



(f) Stairs 2

Figure 5.9: Detection comparison of the proposed DPSD method and benchmark method for several scenes. Each of the four panes is as follows: Top Left: the original texture of the scene. Top Right: the corresponding depth map. Bottom Left: detection results of benchmark method. Bottom Right: detection results of proposed method.

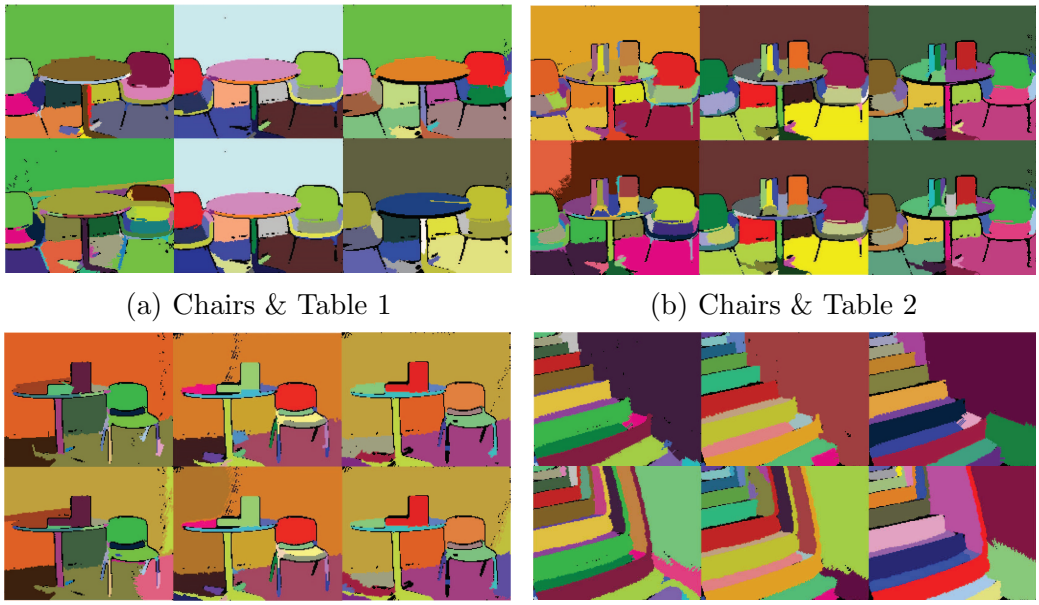


(a) (b) (c)

Figure 5.10: The outcome of the proposed approach: (a) without any of the two post-processing stages; (b) with only the OGS post-processing stage; (c) with the two post-processing stages.

The effectiveness of the proposed post-processing stages has been verified by comparing the results of the proposed approach with and without these stages. The results of this comparison are shown in Fig.5.10, where Fig.5.10 (a) shows the outcome of the proposed approach without the post-processing stages. It is evident from this result that the vertical side of the steps overgrows into the wall, thus ending up separating it into many segments. Meanwhile, Fig.5.10 (b) shows the output of the proposed approach with only the first post-processing stage which mitigates the OGS problem by reallocating the overgrowing surfaces. The output of the first post-processing stage gets refined by the second stage, which merges the parallel subsurfaces, the outcome of this is shown in Fig.5.10 (c). From this result it is evident how the over segmented wall in Fig.5.10 (a) has been correctly detected by the two post-processing stages.

The effect of seed size on the proposed approach could be appreciated from Fig.5.11. The results of each tested scene are shown in one pane in Fig.5.11, each pane has 2×3 images, the columns from left to right show the results for 3×3, 4×4, and 5×5 seed patch. Whereas, the upper and lower row of each pane shows the output of the proposed approach with and without post-processing stages. From the reported results it can be seen that using different sizes of seed patch will lead to slightly different detection results, however, the differences become more pronounced without using the post-processing stages. Consequently, the proposed post-processing methods increase the overall robustness of the proposed algorithm with respect to the chosen parameters. It is worth noticing that the proposed surface merging method only merges the surfaces that are aligned and contiguous, therefore, in the first three testing scenes, the wall surfaces isolated by the table leg and the table top-surface will not be merged with the larger part of the wall.



(a) Chairs & Table 1 (b) Chairs & Table 2

(c) Seat & Table (d) Stairs 2

Figure 5.11: The results of each tested scene are shown in one pane; the columns from left to right in each pane show the results for 3×3, 4×4, and 5×5 seed patch; the upper and lower row of each pane show the output of the proposed approach with and without post-processing stages, respectively.

To objectively assess the accuracy of the proposed DPSD method two methodologies were used, the first one measures the Receiver Operating Characteristic (ROC) of several detected surfaces. The second one checks the angle between some pairs of surfaces based on their estimated equations, and compares these angles with their ground-truth values. Due to the difficulty in generating the ground truth of planar surfaces in captured depth images, one Computer Graphic (CG) depth image was generated along with the ground truth of some surfaces. This image was used to objectively assess the proposed approach.

Fig.5.12 shows a 3D saw-tooth structure, with each "tooth" having different height, thus the angle of each tooth is shown on its top in Fig.5.12 (b). The detected planar surface by the proposed algorithm with and without the post-processing stages are shown, respectively, in Fig.5.13 (a) and Fig.5.13 (b). Furthermore, the actual angle of each tooth, the corresponding measured values by the proposed approach with and without the post-processing stages, and the measurement errors are reported in Table. 5.1. The following equation was used to measure the angle of a tooth $\arccos(\hat{\mathbf{n}}_i^{k_i} \cdot \hat{\mathbf{n}}_u^{k_u})$ where $\hat{\mathbf{n}}_i^{k_i}$ and $\hat{\mathbf{n}}_u^{k_u}$ are the estimated norms of the two surfaces which define that tooth. From these results the accuracy of the proposed approach can be appreciated, and the impor-

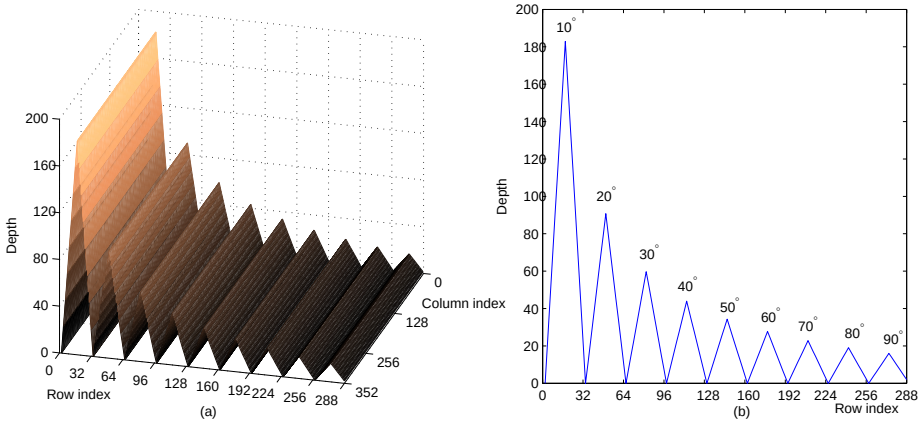
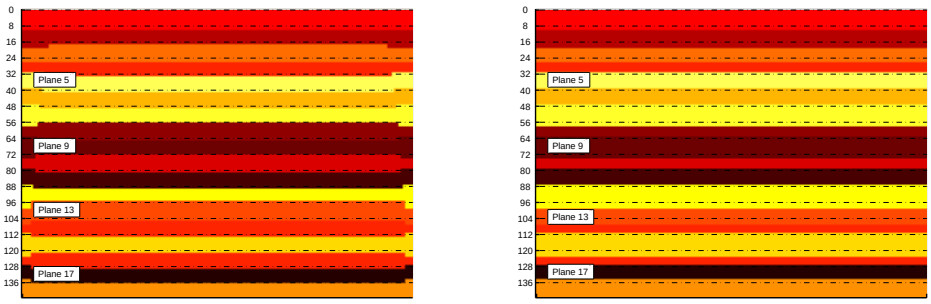


Figure 5.12: (a) The 3D saw-tooth structure, each "tooth" has different height; (b) the profile of the saw-tooth structure with the angle of each tooth is shown on its top.

Table 5.1: The measured angle of each tooth in the 3D saw-tooth structure by the proposed approach with and without the post-processing (PP) stages; the ground-truth (GT) angles, and the measurement errors

GT angle	With PP		Without PP	
	measured angle	Δ	measured angle	Δ
10	9.92	-0.08	11.29	1.29
20	20.03	0.03	20.03	0.03
30	29.87	-0.13	29.87	-0.13
40	39.84	-0.16	39.84	-0.16
50	48.29	-1.71	48.29	-1.71
60	61.07	1.07	60.08	0.08
70	70.50	0.5	86.23	6.23
80	76.73	-3.27	88.23	8.23
90	90.00	0	99.63	9.63



(a) (b)

Figure 5.13: The detected planner surfaces by the proposed algorithm: (a) with the post-processing stages; (b) without the post-processing stages. The actual intersection lines between each two surfaces are shown as dashed lines.

tance of the post-processing stages show that in all but one case, the proposed approach is worse than its version without the post-processing stages. Furthermore, it can be seen that the smaller the angle is the more accurate the measurement is. This is due to the fact that smaller angles in the tested image correspond to bigger planar surfaces, which means that the estimated surfaces' equations are more accurate.

The ROC of the proposed algorithm with and without the post-processing stages for the $5, 9, 13, 17$-th surface are reported in Table 5.2. In this table "TP" (True Positive) counts the points that have been successfully detected as inliers of the surface and "TN" (True Negative) counts the non-belonging points that have been successfully detected as outliers of the surfaces. Whereas, "FN" (False Negative) and "FP" (False Positive) count the points which were wrongly classified as not belonging and belonging to the surface, respectively. This table also shows the detection sensitivity which is the ratio of the correctly detected inliers to the total number of inliers pixels, furthermore, the specificity, which is the ratio of the correctly detected outliers to the total number of outliers pixels.

A demo video that shows the growing process and the detected planar surfaces could be found at `http://v.youku.com/v_show/id_XOTMyODI5MjI4.html`.

Table 5.2: The ROC of the proposed algorithm with and without the post-processing (PP) stages for some planar surfaces of a 3D saw-tooth structure; "TP", "TN", "FN", and "FP" stand for True Positive, True Negative, False Negative and False Positive, respectively

Plane index		Plane 5	Plane 9	Plane 13	Plane 17
TP	With PP	1234	1258	1568	1224
	Without PP	1232	1584	1056	1056
FN	With PP	18	0	8	16
	Without PP	176	0	352	352
TN	With PP	24072	24060	26720	24096
	Without PP	23760	23408	23584	23760
FP	With PP	20	26	184	8
	Without PP	176	352	352	176
Sensitivity TP/(TP + FN)	With PP	98.6%	100.0%	99.50%	98.70%
	Without PP	87.5%	100.0%	81.7%	75.0%
Specificity TN/(TN + FP)	With PP	99.9%	99.9%	99.3%	100.0%
	Without PP	99.3%	98.5%	98.5%	99.3%

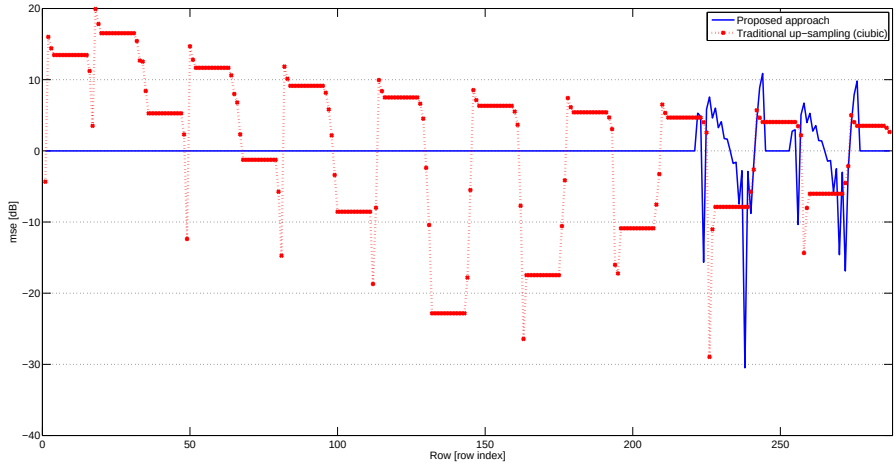
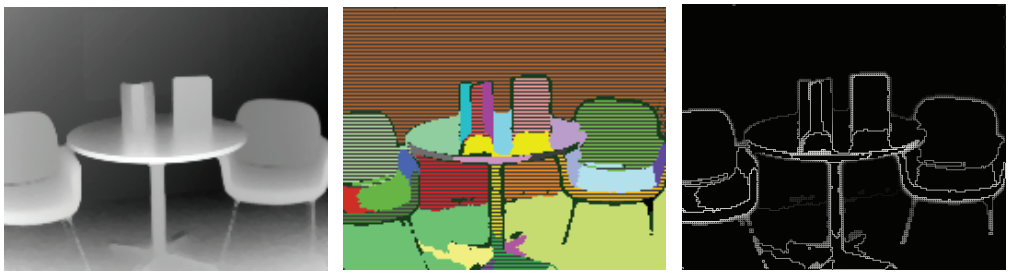


Figure 5.14: The row-by-row MSE for the up-sampled saw-tooth image with respect to the HR ground truth versus the row index.

The MSE was evaluated row-by-row for the up-sampled saw-tooth image with respect to the HR ground truth as reported in Fig.5.14 versus the row index. Two different approaches for up-sampling are shown, namely: the proposed approach, and the traditional cubic approach. The traditional cubic approach was chosen as the benchmark to make future comparisons with the proposed method straightforward. From the reported results the effectiveness of the proposed approach in recovering planar surfaces and edges can be seen. However, for small surfaces, such as those at the right side of Fig.5.12 (a) the DPSD has some problems in estimating their equations, consequently, their HR version will have a high MSE. It is worth reporting that the PSNR of the proposed SR approach is 47.35 dB versus 39.91 dB for the traditional cubic approach, a matched down/up-sampling cubic approach is also tested and its PSNR is 46.54 dB, which also confirms the superior performance of the proposed approach.

Furthermore, to visually appreciate the performance of the proposed approach, different indoor scenes were captured by using the SwissRanger SR4000 depth camera and then tested. In the following due to the limited space the results for only one scene are reported. Fig.5.15 (a) shows the 176×144 original depth image, the output of the surface categorization on this image is shown in Fig.5.15 (b), in this figure different colors are used to represent different detected surfaces, and a horizontal hatch pattern is used to show the surfaces which were categorized as planar surfaces. From this image it can be seen that the proposed surface categorization approach works well, in fact



(a) (b) (c)

Figure 5.15: Image (a) shows the original LR 176×144 depth image; the output of the surface categorization is shown in (b), where horizontal hatch pattern shows planar surfaces; the edges and the isolated non-filled pixels are shown in (c).

non-planar surfaces such as some parts of the chair, and the room floor which although flat however are not smooth, were well identified. In Fig.5.15 (c) the edges and the isolated non-filled pixels are shown.

The image shown in Fig.5.15 (a) has been super-resolved using the proposed approach and a cubic interpolation approach, the results are shown respectively in Fig.5.16 (a) and Fig.5.15 (b). The area delimitated by a red box in Fig.5.16 (a) and Fig.5.15 (b) is blown-up in Fig.5.16 (c) and Fig.5.15 (d), respectively. From these cropped images the superiority of the proposed approach in recovering edges details can be observed.

5.5 Conclusions

In this chapter, a dynamic seed growing mechanism to detect planar and semi-planar surfaces using depth map images and a planar-surface-based depth map super-resolution approach are proposed. The performance of the proposed method was assessed by visual and objective comparisons with benchmark methods on some typical indoor scenes. To tackle the overgrowing surface problem and to merge separated surfaces two post processing methods were proposed which exploit the estimated equations of the detected surfaces. Referring to these equations, all the surfaces will be categorized into three groups: planar, non-planar surfaces and edges. Then, for each of these three categories, a proper up-sampling approach is applied to estimate the to-be-filled pixels. For the planar surfaces, they are upsampled by using the analytical equations, while, for the non-planar surfaces, the bicubic interpolator is used. Finally, a combination of the planar-surface and bicubic approaches is used to upsample the edges. Moreover, the reported results indicate that firstly, the DPSD method can detect planar and

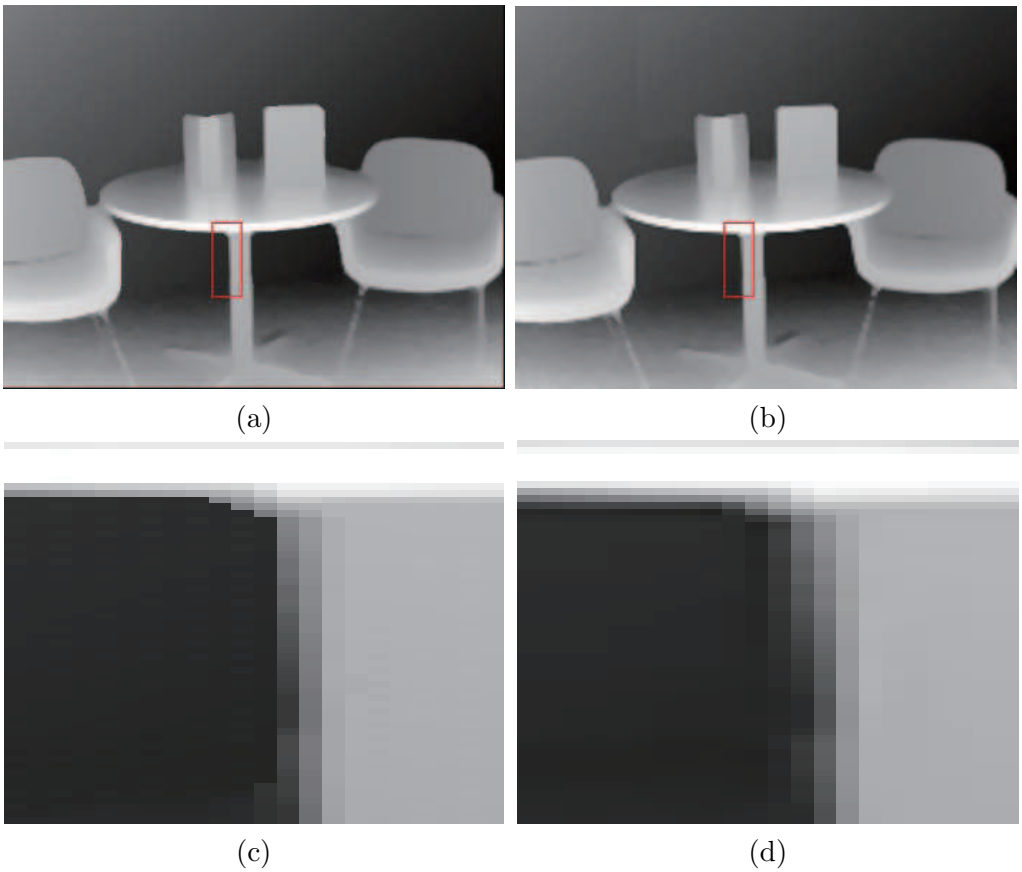


Figure 5.16: The super-resolved image using: (a) proposed approach; (b) traditional interpolation approach; The delimitated area by a red box in (a) and (b) is blown-up in (c) and (d), respectively.

semi-planar surfaces with high sensitivity and the post-processing methods increase its robustness with respect to the chosen parameters; secondly, the proposed super-resolution approach achieves superior performance in comparison with a traditional interpolation one.

It is worth reporting that the work reported in this chapter has led to the following publication: Zhi Jin, Tammam Tillo, Fei Cheng, Depth-map Driven Planar Surfaces Detection, IEEE Visual Communications and Image Processing (VCIP) Conference, 2014.

Zhi Jin, Tammam Tillo, Fei Cheng, Planar Surfaces Detection on Depth Map Using Patch Based Approach, IEEE 3rd Global Conference on Consumer Electronics (GCCE), 2014.

Zhi Jin, Tammam Tillo, Fei Cheng, (Demo) Accurate Planar Surfaces Detection Using Depth Map, European Conference on Computer Vision (ECCV), 2014.

Tammam Tillo, Zhi Jin and Fei Cheng, Super-resolution of depth map exploiting planar surfaces, Pacific-Rim Conference on Multimedia (PCM), 2015.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Image/video SR provides an efficient solution for the quality enhancement and has been widely used in video surveillance, medical image process, daily life entertainment and so on. In this thesis, a comprehensive review of classic and state-of-the-art SR techniques has been provided in Chapter 2. From the review, it can be observed that many super resolution algorithms are proposed based on image or 2D video systems, however, few of them are based on 3D video. With one more cue in depth, the SR methods designed for 3D video can reach better results than directly applying 2D SR algorithms on 3D video. Therefore, in order to fill the gap with respect to current SR algorithms, we proposed to introduce depth information into the texture SR process by referring the characteristics of 3D video. In general, the SR algorithms for texture images can be mainly classified into 3 categories: reconstruction-based, learning-based and interpolation-based algorithms. The performance of reconstruction and learning-based algorithms highly rely on the choice of corresponding parameters and training samples, respectively. Hence, compared with previous two methods, the interpolation-based SR algorithms are easy implemented and efficient which is suitable to real time system. Starting from this point, in this thesis, two interpolation-based SR algorithms are proposed with the assist of corresponding depth images in Chapters 3 and Chapters 4.

In Chapter 3, a novel virtual view assisted super-resolution method for MR multiview video has been presented where the low resolution views in the MR multiview video are super-resolved to full resolution size in two stages. In the first stage, the similarity between the LR pixels and their counterparts in the virtual view are measured.

A smoothness check is carried out to determine whether using virtual view pixels or interpolated pixels to fill the zero-filled pixels. Subsequently, the quality of the virtual-view-based pixels is enhanced by compensating the intrinsic luminance difference between the views. Furthermore, the inter-view correlation is exploited to enhance the LR pixels in the super-resolved frame by reducing their compression distortion. Therefore, different from the state-of-the-art interpolation-based SR algorithms, the advantages of virtual views are exploited by the proposed method at different stages. The experimental results demonstrate the effectiveness of the proposed approach with a PSNR gain of up to 3.85dB.

Except MR multiview video, the MVD video is also a popular representation of 3D data. Unlike MR multiview video, each view of the MVD video has the same resolution and is associated with corresponding depth map. Therefore, in this thesis, a novel virtual view assisted SR method for MVD video has been presented as the extension of the work in Chapter 3. This SR method aims to super-resolved the LR views that are generated by employing interlacing-and-complementary-row-downsampling method on the two adjacent texture views at the encoder side to reduce the transmitted data. In the proposed approach, edge directions around the discarded pixels are estimated by principal components analysis. This information is subsequently used to steer the fusion of the virtual view with the directional interpolated pixels so as to exploit the inter-view redundancy and to minimize the overall system distortion. The performance improvement of this work can be confirmed by exhaustive computer simulations.

The depth maps in MVD video are generated by software so that they have the same resolution as the textures. However, for the hardware, i.e.ToF depth camera, generated depth maps, they have lower resolution compared with general texture resolution which means they are hardly used in the MVD video. Hence, in order to solve this problem, in this thesis, a depth map SR algorithm has been proposed where all the planar surfaces in the depth map are detected with corresponding surface equations firstly. Then, for the planar surfaces, they are super-resolved by using the analytical equations. For the rest parts of depth map, i.e. the non-planar surfaces and edges, the bicubic interpolator and a more sophisticated approach are used, respectively. The performance of the proposed method was assessed by visual and objective comparisons with benchmark methods on some typical indoor scenes and the reported results indicate that the proposed SR approach achieves superior performance in comparison with traditional interpolation

methods.

To conclude, the proposed depth SR algorithm boosts the adoption of the proposed two texture SR algorithms and the proposed texture SR algorithms increase the usage of depth information.

6.2 Future Work

The future researches will be focused on the following aspects:

- Exploitation of temporal correlation

  Both of these two proposed texture SR methods mainly adopt inter-view information to super-resolve the LR texture video. In other words, they are frame-based texture SR. In fact, the previous and later frames with respect to the current frame contain highly redundant information which can be used in SR algorithms. Especially for the second proposed texture SR algorithm, if the previous and current frame also adopt the interlacing-and-complementary-row-downsampling method, which means in one view, the odd frame discards odd rows and even frame discards even rows, the generated virtual view from the counterpart view can also be involved in the fusion step. Therefore, one future work will be devoted to combining temporal correlation with inter-view correlation to improve the exploitation of the virtual views and to enhance the performance of texture video SR.

- Adoption of human-involved subjective assessment

  Since at the decoder side, the received video is finally viewed by human, subjective evaluation is more"proper" to quantify visual quality than objective evaluation. So far, although some of the assessments for proposed SR methods are carried out subjectively, no human-involved subjective assessment is used. This is caused by the inconvenience, high requirement of time and money, and the testing equipment and environment. However, added by the future possible collaborations with corresponding research groups, this aim may be achieved. Referring to the results of human-involved subjective assessment, the proposed SR algorithms could be further improved.

- Further applications of proposed planar surface detection method

The proposed planar surface detection method in Chapter 5, besides being used in depth SR technique, also can be used in depth map enhancement, for example, large size hole filling problems in Kinect. With the analytic surface equations, the big holes in planar surfaces can be filled. In addition, this method can be modified in order to work on the 3D point cloud data for point cloud segmentation.

Appendix A

List of publications

1. Zhi Jin, Tammam Tillo, Chao Yao, Jimin Xiao, and Yao Zhao, Virtual View Assisted Video Super-Resolution and Enhancement, IEEE Transactions on Circuits and Systems for Video Technology, Volume:PP , Issue: 99, doi:10.1109/TCSVT.2015.2412791

2. Tammam Tillo, Zhi Jin and Fei Cheng, Super-resolution of depth map exploiting planar surfaces, Pacific-Rim Conference on Multimedia, PCM 2015

3. Zhi Jin, Tammam Tillo, and Lei Luo, Quality Enhancement of Quality-asymmetric Multiview Plus Fepth Video By Using Virtual View, IEEE International Conference on Multimedia and Expo, ICME 2015

4. Zhi Jin, Tammam Tillo, and Fei Cheng, Depth-map Driven Planar Surfaces Detection, IEEE Visual Communications and Image Processing, VCIP 2014

5. Zhi Jin, Tammam Tillo, and Fei Cheng, Planar Surfaces Detection on Depth Map Using Patch Based Approach, IEEE 3rd Global Conference on Consumer Electronics, GCCE 2014

6. Zhi Jin, Tammam Tillo, and Fei Cheng, Accurate Planar Surfaces Detection Using Depth Map, European Conference on Computer Vision, ECCV 2014

7. Zhi Jin,Tammam Tillo, EngGee Lim, Zhao Wang and Jimin Xiao, Novel Wireless Capsule Endoscopy Diagnosis System with Adaptive Image Capturing Rate, In Proceedings of the International Conferenceon Computer Vision Theoryand Applications, VISAPP 2013

8. Zhi Jin, Jimin Xiao, Tammam Tillo and Fei Cheng, 3D Video Depth map Quantization based on Lloyds Algorithm, 11th IEEE Signal Processing Society IVMSP

Workshop, 2013

9. Fei Cheng, Jimin Xiao, Zhi Jin and Tammam Tillo, Video Error Concealment of P-frame Using Packets of the Following Frames, The 8th International Conference on Signal Image Technology and Internet Based Systems, 2013

Bibliography

[1] J. Zhang, Y. Cao, and Z. Wang. A simultaneous method for 3d video super-resolution and high-quality depth estimation. In *ICIP*, pages 1346–1350, 2013.

[2] M. Mancuso and S. Battiato. An introduction to the digital still camera technology. *Image*, 2(2), 2001.

[3] S.C. Park, M.K.Park, and M.G. Kang. Super-resolution image reconstruction: a technical overview. *Signal Processing Magazine, IEEE*, 20(3):21–36, May 2003.

[4] X. Li and M.T. Orchard. New edge-directed interpolation. *Image Processing, IEEE Transactions on*, 10(10):1521 –1527, Oct. 2001.

[5] K.-W. Hung and W.-C. Siu. Fast image interpolation using the bilateral filter. *Image Processing, IET*, 6(7):877–890, 2012.

[6] J. Zhang, S. Ma, and D. Zhang, Y.and Zhao. Fast and effective interpolation using median filter. 5879:1174–1184, 2009.

[7] W.T. Freeman, T.R. Jones, and E.C. Pasztor. Example-based super-resolution. *Computer Graphics and Applications, IEEE*, 22(2):56–65, 2002.

[8] D.C. Garcia, C. Dorea, and R.L. de Queiroz. Super resolution for multiview images using depth information. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(9):1249 –1256, Sep. 2012.

[9] Y. Zhang, X.Ji, H. Wang, and Q. Dai. Stereo interleaving video coding with content adaptive image subsampling. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(7):1097–1108, Jul. 2013.

[10] J. Xiao, M.M. Hannuksela, T. Tillo, M. Gabbouj, C. Zhu, and Y. Zhao. Scalable bit allocation between texture and depth views for 3-d video streaming over het-

erogeneous networks. *Circuits and Systems for Video Technology, IEEE Transactions on*, 25(1):139–152, Jan. 2015.

[11] A.C. Yau, N.K. Bose, and M.K. Ng. An efficient algorithm for superresolution in medium field imaging. *Multidimensional Systems and Signal Processing*, 18(2-3):173–188, 2007.

[12] Y. Wang, J. Ostermann, and Y.Q. Zhang. *Video processing and communications*, volume 5. Prentice Hall Upper Saddle River, 2002.

[13] G. Caner, W.D. Heinzelman, et al. Super resolution recovery for multi-camera surveillance imaging. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 1, pages I–109. IEEE, 2003.

[14] X. Chen and C. Qi. A super-resolution method for recognition of license plate character using lbp and rbf. In *Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on*, pages 1–5. IEEE, 2011.

[15] B. K. Gunturk, A. U. Batur, Y. Altunbasak, M.H. Hayes, and R.M. Mersereau. Eigenface-domain super-resolution for face recognition. *Image Processing, IEEE Transactions on*, 12(5):597–606, 2003.

[16] A. Boucher, P.C. Kyriakidis, and C. Cronkite-Ratcliff. Geostatistical solutions for super-resolution land cover mapping. *Geoscience and Remote Sensing, IEEE Transactions on*, 46(1):272–283, 2008.

[17] J. Kennedy, O. Israel, A. Frenkel, R. Bar-Shalom, H. Azhari, et al. Super-resolution in pet imaging. *Medical Imaging, IEEE Transactions on*, 25(2):137–147, 2006.

[18] X. Huang, H. Li, and S. Forchhammer. A multi-frame post-processing approach to improved decoding of h.264/avc video. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 4, pages IV – 381–IV – 384, Sep. 2007.

[19] M. Singh. *THEORY AND METHODS FOR EFFICIENT SPATIO-TEMPORAL SUPER-RESOLUTION IMAGING.* PhD thesis, Department of Electrical and Computer Engineering,University of Alberta, 395 Wellington Street, OttawaONK1A0N4,Canada, 2009.

[20] S.S. Panda, M.S.R. Prasad, and G. Jena. Pocs based super-resolution image reconstruction using an adaptive regularization parameter. *arXiv preprint arXiv:1112.1484*, 2011.

[21] X. Gao, K. Zhang, D. Tao, and X. Li. Joint learning for single-image super-resolution via a coupled constraint. *Image Processing, IEEE Transactions on*, 21(2):469–480, 2012.

[22] D. Zhang and X. Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. *Image Processing, IEEE Transactions on*, 15(8):2226–2238, 2006.

[23] M. Li and T.Q. Nguyen. Markov random field model-based edge-directed image interpolation. *Image Processing, IEEE Transactions on*, 17(7):1121–1128, 2008.

[24] X. Li, Y. Hu, X. Gao, D. Tao, and B. Ning. A multi-frame image super-resolution method. *Signal Process.*, 90(2):405–414, 2010.

[25] X. Gao, Q. Wang, X. Li, D. Tao, and K. Zhang. Zernike-moment-based image super resolution. *Image Processing, IEEE Transactions on*, 20(10):2738–2747, 2011.

[26] K. Zhang, X. Gao, D. Tao, and X. Li. Single image super-resolution with non-local means and steering kernel regression. *Image Processing, IEEE Transactions on*, 21(11):4544–4556, 2012.

[27] H. Su, L. Tang, Y. Wu, D. Tretter, and J. Zhou. Spatially adaptive block-based super-resolution. *Image Processing, IEEE Transactions on*, 21(3):1031 –1045, Mar. 2012.

[28] Z. Jin, T. Tillo, C. Yao, J. Xiao, and Y. Zhao. Virtual view assisted video super-resolution and enhancement. *Circuits and Systems for Video Technology, IEEE Transactions on*, PP(99):1–1, 2015.

[29] G. Zhong, L. Yu, and P. Zhou. Edge-preserving single depth image interpolation. In *Visual Communications and Image Processing (VCIP), 2013*, pages 1–6. IEEE, 2013.

[30] Z. Jin, T. Tillo, and F. Cheng. Depth-map driven planar surfaces detection. In *Visual Communications and Image Processing Conference, 2014 IEEE*, pages 514–517, Dec. 2014.

[31] Z. Jin, T. Tillo, and F. Cheng. Planar surfaces detection on depth map using patch based approach. In *Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference on*, pages 227–229, Oct. 2014.

[32] G. Batchen. *Burning with Desire: The Conception of Photography*. Cambridge, MA: MIT Press., 2002.

[33] W.S. Boyle and G.E. Smith. Charge coupled semiconductor devices. *Bell System Technical Journal*, 49(4):587–593, 1970.

[34] A.G. Dickinson, E.I. Eid, and D.A. Inglis. Active pixel sensor and imaging system having differential mode, May 20 1997. US Patent 5,631,704.

[35] J. Nakamura. *Image sensors and signal processing for digital still cameras*. CRC press, 2005.

[36] H.R. Sheikh, A.C. Bovik, and G. De Veciana. An information fidelity criterion for image quality assessment using natural scene statistics. *Image Processing, IEEE Transactions on*, 14(12):2117–2128, 2005.

[37] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[38] C. Wang, Z.-Y. Zhu, S.-C. Chan, and H.-Y. Shum. Real-time depth image acquisition and restoration for image based rendering and processing systems. *Journal of Signal Processing Systems*, 79(1):1–18, 2013.

[39] F. Schaffalitzky, A. Zisserman, R.I. Hartley, and P. HS Torr. A six point solution for structure and motion. In *Computer Vision-ECCV 2000*, pages 632–648. Springer, 2000.

[40] S.D. Cochran and G. Medioni. 3-d surface description from binocular stereo. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10):981–994, 1992.

[41] D.P. Robertson and R. Cipolla. Building architectural models from many views using map constraints. In *Computer VisionECCV 2002*, pages 155–169. Springer, 2002.

[42] B. Triggs. Factorization methods for projective structure and motion. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 845–851. IEEE, 1996.

[43] F. Remondino and D. Stoppa. *TOF range-imaging cameras*, volume 68121. Springer, 2013.

[44] S.B. Gokturk, H. Yalcin, and C. Bamji. A time-of-flight depth sensor-system description, issues and solutions. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 35–35. IEEE, 2004.

[45] http://www.mesa-imaging.ch/home/ (accessed: May.1st, 2014).internest, 2014.

[46] http://www.microsoft.com/en-us/kinectforwindows/ (accessed: May.1st, 2014).internest, 2014.

[47] J. Zhang, Y. Cao, Z. Zheng, C. Chen, and Z. Wang. A new closed loop method of super-resolution for multi-view images. *Machine Vision and Applications*, 25(7):1685–1695, 2014.

[48] C.-T. Tsai and H.-M. Hang. Quality assessment of 3d synthesized views with depth map distortion. In *Visual Communications and Image Processing (VCIP), 2013*, pages 1–6, Nov. 2013.

[49] S.S. Qureshi, X. Li, and T. Ahmad. Investigating image super resolution techniques: What to choose? In *Advanced Communication Technology (ICACT), 2012 14th International Conference on*, pages 642–647, Feb. 2012.

[50] R. Keys. Cubic convolution interpolation for digital image processing. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 29(6):1153–1160, Dec. 1981.

[51] H.S. Hou and H. Andrews. Cubic splines for image interpolation and digital filtering. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(6):508–517, Dec. 1978.

[52] T.M. Lehmann, C. Gonner, and K. Spitzer. Survey: interpolation methods in medical image processing. *Medical Imaging, IEEE Transactions on*, 18(11):1049–1075, Nov. 1999.

[53] X. Zhang and X. Wu. Image interpolation by adaptive 2-d autoregressive modeling and soft-decision estimation. *Image Processing, IEEE Transactions on*, 17(6):887–896, 2008.

[54] M. Li and T.Q. Nguyen. Markov random field model-based edge-directed image interpolation. *Image Processing, IEEE Transactions on*, 17(7):1121–1128, 2008.

[55] S. Borman and R.L. Stevenson. Super-resolution from image sequences-a review. In *Circuits and Systems, 1998. Proceedings. 1998 Midwest Symposium on*, pages 374–378, Aug. 1998.

[56] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324 – 335, 1993.

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

[58] C.M. Bishop, A. Blake, and B. Marthi. Super-resolution enhancement of video. In *Proc. Artificial Intelligence and Statistics*, volume 2. Key West, FL, USA, 2003.

[59] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2004.

[60] D.L. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[61] J. Yang, J. Wright, T.S. Huang, and Y. Ma. Image super-resolution via sparse representation. *Image Processing, IEEE Transactions on*, 19(11):2861–2873, 2010.

[62] Z. Xiong, D. Xu, and F. Sun, X.and Wu. Example-based super-resolution with

soft information and decision. *Multimedia, IEEE Transactions on*, 15(6):1458–1465, 2013.

[63] M.-C. Yang and Y.-C.F. Wang. A self-learning approach to single image super-resolution. *Multimedia, IEEE Transactions on*, 15(3):498–508, 2013.

[64] C. Liu and D. Sun. On bayesian adaptive video super resolution. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(2):346–360, Feb. 2014.

[65] B.-T. Cao, D.-H. Tuan, L.-T. Thuong, and N.-D. Hoang. An efficient approach based on bayesian map for video super-resolution. In *Advanced Technologies for Communications (ATC), 2014 International Conference on*, pages 522–527, Oct. 2014.

[66] A.M. Tourapis, H.-Y. Cheong, M. Liou, and O.C. Au. Temporal interpolation of video sequences using zonal based algorithms. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 895–898. IEEE, 2001.

[67] H.A. Karim, M. Bister, and M.U. Siddiqi. Low rate video frame interpolation-challenges and solution. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–117. IEEE, 2003.

[68] T. Thaipanich, P. Wu, and CC.J. Kuo. Low complexity algorithm for robust video frame rate up-conversion (fruc) technique. *Consumer Electronics, IEEE Transactions on*, 55(1):220–228, 2009.

[69] G. Zhai and X. Wu. Video super-resolution for dual-mode digital cameras via scene-matched learning. In *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*, pages 438–442, Oct. 2010.

[70] J. Xie, C.-C. Chou, R. Feris, and M.-T. 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.

[71] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. In *Advances in neural information processing systems*, pages 2223–2231, 2009.

[72] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. High-quality scanning using time-of-flight depth superresolution. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–7. IEEE, 2008.

[73] K. Al Ismaeil, D. Aouada, B. Mirbach, and B. Ottersten. Dynamic super resolution of depth sequences with non-rigid motions. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 660–664. IEEE, 2013.

[74] J. Kopf, M.F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. In *ACM Transactions on Graphics (TOG)*, volume 26, page 96. ACM, 2007.

[75] Q. Yang, R. Yang, J. Davis, and D. 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.

[76] K.-H. Lo, K.-L. Hua, and Y.-CF. Wang. Depth map super-resolution via markov random fields without texture-copying artifacts. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 1414–1418. IEEE, 2013.

[77] Q. He and R. Schultz. *Super-resolution reconstruction by image fusion and application to surveillance videos captured by small unmanned aircraft systems*. INTECH Open Access Publisher, 2010.

[78] Z. Pan, J. Yu, H. Huang, S. Hu, A. Zhang, H. Ma, and W. Sun. Super-resolution based on compressive sensing and structural self-similarity for remote sensing images. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(9):4864–4876, Sep. 2013.

[79] H. Zhang, Z. Yang, L. Zhang, and H. Shen. Super-resolution reconstruction for multi-angle remote sensing images considering resolution differences. *Remote Sensing*, 6(1):637, 2014.

[80] M.-G. Hu, J.-F. Wang, and Y. Ge. Super-resolution reconstruction of remote sensing images using multifractal analysis. *Sensors*, 9(11):8669–8683, 2009.

[81] M.D. Robinson, S.J. Chiu, J.Y. Lo, C.A. Toth, J.A. Izatt, and S. Farsiu. *New applications of super-resolution in medical imaging.* CRC Press, 2010.

[82] J.A. Kennedy, O. Israel, A. Frenkel, R. Bar-Shalom, and H. Azhari. Super-resolution in pet imaging. *Medical Imaging, IEEE Transactions on*, 25(2):137–147, Feb. 2006.

[83] H. Greenspan. Super-resolution in medical imaging. *The Computer Journal*, 52(1):43–63, 2009.

[84] C.S. Boon, O.G. Guleryuz, T. Kawahara, and Y. Suzuki. Sparse super-resolution reconstructions of video from mobile devices in digital tv broadcast applications. In *SPIE Optics+ Photonics*, pages 63120M–63120M. International Society for Optics and Photonics, 2006.

[85] H.S. Sawhney, Y. Guo, K. Hanna, and R. Kumar. Hybrid stereo camera: an ibr approach for synthesis of very high resolution stereoscopic image sequences. In *In SIGGRAPH*, pages 451–460. Press, 2001.

[86] P. Aflaki, M.M. Hannuksela, J. Hakkinen, P. Lindroos, and M. Gabbouj. Subjective study on compressed asymmetric stereoscopic video. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 4021 –4024, Sep. 2010.

[87] X. Li, Y. Hu, X. Gao, D. Tao, and B. Ning. A multi-frame image super-resolution method. *Signal Process.*, 90(2):405–414, 2010.

[88] X. Gao, Q. Wang, X. Li, D. Tao, and K. Zhang. Zernike-moment-based image super resolution. *Image Processing, IEEE Transactions on*, 20(10):2738–2747, 2011.

[89] X. Gao, K. Zhang, D. Tao, and X. Li. Joint learning for single-image super-resolution via a coupled constraint. *Image Processing, IEEE Transactions on*, 21(2):469–480, 2012.

[90] D. Zhang and X. Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. *Image Processing, IEEE Transactions on*, 15(8):2226–2238, 2006.

[91] P. Aflaki, M.M. Hannuksela, J. Hakkinen, P. Lindroos, and M. Gabbouj. Impact of downsampling ratio in mixed-resolution stereoscopic video. In *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2010*, pages 1 –4, Jun. 2010.

[92] L. Stelmach, Wa James Tam, D. Meegan, and A. Vincent. Stereo image quality: effects of mixed spatio-temporal resolution. *Circuits and Systems for Video Technology, IEEE Transactions on*, 10(2):188 –193, Mar. 2000.

[93] Jr.L. McMillan. *An image-based approach to three-dimensional computer graphics*. PhD thesis, Chapel Hill, NC, USA, 1997. UMI Order No. GAX97-30561.

[94] B. Shi, Y. Li, L. Liu, and C. Xu. Color correction and compression for multiview video using h.264 features. In *Computer Vision–ACCV 2009*, pages 43–52. Springer, 2010.

[95] J.L. Véhel, P. Legrand, et al. Hölderian regularity-based image interpolation. In *ICASSP 06, International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 852–855, 2006.

[96] M.T. Alexis, S. Karsten, and S. Gary. H.264/14496-10 avc reference software manual. In *document:JVT-AE010 ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6,London, UK*, 2009.

[97] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[98] P. Merkle, A. Smolic, K. Muller, and T. Wiegand. Efficient prediction structures for multiview video coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(11):1461–1473, Nov.2007.

[99] C. Fehn. A 3d-tv approach using depth-image-based rendering (dibr). Sep. 2003.

[100] D.C. Garcia, T.A. da Fonseca, and R.L. de Queiroz. Video compression complexity reduction with adaptive down-sampling. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 745–748, Sep. 2011.

[101] L. Wang, S. Xiang, G. Meng, H. Wu, and C. Pan. Edge-directed single-image super-resolution via adaptive gradient magnitude self-interpolation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(8):1289–1299, Aug. 2013.

[102] Z. Jin, T. Tillo, C. Yao, J. Xiao, and Y. Zhao. Virtual view assisted video super-resolution and enhancement. *Circuits and Systems for Video Technology, IEEE Transactions on*, PP(99):1–1, 2015.

[103] Y. Zhang, D. Zhao, J. Zhang, R. Xiong, and W. Gao. Interpolation-dependent image downsampling. *Image Processing, IEEE Transactions on*, 20(11):3291 – 3296, Nov. 2011.

[104] I.T. Jolliffe. *Principal Component Analysis*. Springer, NY, 2002.

[105] B. Yang, Z. Gao, and X. Zhang. Principal components analysis-based edge-directed image interpolation. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 580–585, 2012.

[106] G. W. Stewart. On the early history of the singular value decomposition. *SIAM Rev.*, 35(4):551–566, 1993.

[107] S.Olgierd, W. Krzysztof, and D. Marek. First version of depth maps for poznan 3d/ftv test sequences. In *document MPEG 2010/M17176,ISO/IEC JTC1/SC29/WG11,Kyoto,Japan*, Jan. 2010.

[108] Nagoya University. Ftv test sequences. `[Online]Available:http://www.tanimoto.nuee.nagoya-u.ac.jp/mpeg/mpeg_ftv.html`.

[109] Y. Chen, P. Pandit, S. Yea, and C. Lim. Draft reference software for mvc. In *Joint Video Team (JVT) of ISO/IEC/MPEG and ITU-T/VCEG, Doc. JVT-AE207,London, U.K.*, 2009.

[110] Z. Wang and Z. Zheng. A region based stereo matching algorithm using cooperative optimization. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[111] R. Hulik, M. Spanel, P. Smrz, and Z. Materna. Continuous plane detection in point-cloud data based on 3d hough transform. *Journal of Visual Communica-*

tion and Image Representation, 25(1):86 – 97, 2014. Visual Understanding and Applications with RGB-D Cameras.

[112] A. Hoover, G. Jean-Baptiste, X. Jiang, P.J. Flynn, H. Bunke, D.B. Goldgof, K. Bowyer, D.W. Eggert, A. Fitzgibbon, and R.B. Fisher. An experimental comparison of range image segmentation algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(7):673–689, Jul. 1996.

[113] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[114] M.Y. Yang and W. Förstner. Plane detection in point cloud data. In *Proceedings of the 2nd int conf on machine control guidance, Bonn*, volume 1, pages 95–104, 2010.

[115] X. Qian and C. Ye. Ncc-ransac: A fast plane extraction method for 3-d range data segmentation. *IEEE transactions on cybernetics*, 2014.

[116] P. VC. Hough. Method and means for recognizing complex patterns, 18 1962. US Patent 3,069,654.

[117] D. Dube and A. Zell. Real-time plane extraction from depth images with the randomized hough transform. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1084–1091. IEEE, 2011.

[118] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter. The 3d hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Research*, 2(2):1–13, 2011.

[119] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak. Fast plane detection and polygonalization in noisy 3d range images. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3378–3383. IEEE, 2008.

[120] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga. Fast registration based on noisy planes with unknown correspondences for 3-d mapping. *Robotics, IEEE Transactions on*, 26(3):424–441, Jun. 2010.

[121] D. Holz and S. Behnke. Fast range image segmentation and smoothing using approximate surface reconstruction and region growing. In *Intelligent Autonomous Systems 12*, pages 61–73. Springer, 2013.

[122] J. Xiao, B. Adler, J. Zhang, and H. Zhang. Planar segment based three-dimensional point cloud registration in outdoor environments. *Journal of Field Robotics*, 30(4):552–582, 2013.

[123] J. Xiao, J. Zhang, J. Zhang, H. Zhang, and H. P. Hildre. Fast plane detection for slam from noisy range images in both structured and unstructured environments. In *Mechatronics and Automation (ICMA), 2011 International Conference on*, pages 1768–1773. IEEE, 2011.

[124] J.-E. Deschaud and F. Goulette. A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing. In *Proceedings of 3D Processing, Visualization and Transmission Conference (3DPVT2010)*, 2010.

[125] J. Park, H. Kim, Y.-W. Tai, M.S. Brown, and I. Kweon. High quality depth map upsampling for 3d-tof cameras. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1623–1630. IEEE, 2011.

[126] G. Zhong, L. Yu, and P. Zhou. Edge-preserving single depth image interpolation. In *Visual Communications and Image Processing (VCIP), 2013*, pages 1–6. IEEE, 2013.

[127] http://mathworld.wolfram.com/point-planedistance.html (accessed: April.3rd, 2016).internest, 2016.

[128] http://mathworld.wolfram.com/plane-planeintersection.html (accessed: April.3rd, 2016).internest, 2016.

[129] R. Keys. Cubic convolution interpolation for digital image processing. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 29(6):1153–1160, 1981.

[130] E.W. Gonzalez, R.C.and Richard. *Digital Image Processing*. Pearson Education, 2009.

[131] Japan RICOH. http://www.us.ricoh-imaging.com/ (accessed: April.1st, 2015).internest, 2015.