WestVirginiaUniversity
THE RESEARCH REPOSITORY @ WVU

Graduate Theses, Dissertations, and Problem Reports

2012

Selected Topics in Bayesian Image/Video Processing

Qiang Hao
West Virginia University

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Recommended Citation

Hao, Qiang, "Selected Topics in Bayesian Image/Video Processing" (2012). *Graduate Theses, Dissertations, and Problem Reports*. 3584.
https://researchrepository.wvu.edu/etd/3584

This Dissertation is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Dissertation has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Selected Topics in Bayesian Image/Video Processing

Qiang Hao

Dissertation submitted to the
Benjamin M.Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Electrical Engineering

Xin Li, Ph.D., Chair
Arun. A. Ross, Ph.D.
Natalia A. Schmid, Ph.D.
Donald A. Adjeroh, Ph.D.
Erdogan Gunel, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2012

Keywords: data-driven, non-blind deconvolution, visual content insertion, image statistics, pre- and post- processing, encoder and decoder, sparse representation

Copyright 2012 Qiang Hao

Abstract

Selected Topics in Bayesian Image/Video Processing

Qiang Hao

In this dissertation, three problems in image deblurring, inpainting and virtual content insertion are solved in a Bayesian framework.

Camera shake, motion or defocus during exposure leads to image blur. Single image deblurring has achieved remarkable results by solving a MAP problem, but there is no perfect solution due to inaccurate image prior and estimator. In the first part, a new non-blind deconvolution algorithm is proposed. The image prior is represented by a Gaussian Scale Mixture(GSM) model, which is estimated from non-blurry images as training data. Our experimental results on a total twelve natural images have shown that more details are restored than previous deblurring algorithms.

In augmented reality, it is a challenging problem to insert virtual content in video streams by blending it with spatial and temporal information. A generic virtual content insertion(VCI) system is introduced in the second part. To the best of my knowledge, it is the first successful system to insert content on the building facades from street view video streams. Without knowing camera positions, the geometry model of a building facade is established by using a detection and tracking combined strategy. Moreover, motion stabilization, dynamic registration and color harmonization contribute to the excellent augmented performance in this automatic VCI system.

Coding efficiency is an important objective in video coding. In recent years, video coding standards have been developing by adding new tools. However, it costs numerous modifications in the complex coding systems. Therefore, it is desirable to consider alternative standard-compliant approaches without modifying the codec structures. In the third part, an exemplar-based data pruning video compression scheme for intra frame is introduced. Data pruning is used as a pre-processing tool to remove part of video data before they are encoded. At the decoder, missing data is reconstructed by a sparse linear combination of similar patches. The novelty is to create a patch library to exploit similarity of patches. The scheme achieves an average 4% bit rate reduction on some high definition videos.

Acknowledgements

I have been fortunate to be advised by Dr.Xin Li. He has taught me much about image and video processing and how to get to the fundamental part of research problems. I am grateful for the time that he has spent discussing ideas and helping me get out of difficulties in my research.

I appreciate very much that Dr.Natalia Schmid, Dr.Arun Ross, Dr.Donald Adjeroh and Dr.Erdogan Gunel be my thesis committee members and spend time in reading it carefully and make suggestions for its improvements.

I appreciate very much that Dr.Dong-Qing Zhang provided me an internship at Technicolor Research Center in summer 2010. His encouragement gave me some confidence when I was at the most difficult times of my PhD studies.

I would also thank Dr.Yu Huang and Dr.Heather Yu who provided me an internship at Huawei US Technologies in fall 2010. I learned things from their insights of computer vision. I would also thank Dr.Ali Tabatabai and Dr.Jun Xu who gave me an intern opportunity at Sony US Research Center in summer 2011. It was a good research training on video compression.

I would thank Dr.Jinyu Zuo for his selfless help over the years. I would also thank all my friends who companied and encouraged me along this journey.

Many thanks to my parents and my parents-in-law for their tremendous supports, both emotionally and financially.

My greatest thanks go to my family. My three-year old son Steven indeed challenged my research work and life, however, he has shaped my graduate experience more than anything else. My husband, Dr.Bo Zhao, has continually loved and supported me. He always encouraged me to pursuit my dreams and helped me to develop. I am fortunate to have such an amazing parter for life and the eternities.

Contents

List of Figures

List of Tables

Notation

We use the following notation and symbols throughout this thesis.

μ	:	Mean of Gaussian Distribution
σ^2	:	Variance of Gaussian distribution
$(\cdot)^T$	:	Transpose matrix
$(\cdot)^{-1}$	:	Inverse matrix
$\|\cdot\|_p$	:	p norm operator
$\bullet$	:	Inner product
$\|\cdot\|$	:	Absolution value
$[a]_\times b$	:	The line passing through the point a and b
$\vec{a}$	:	Vector a
$\langle\cdot,\cdot\rangle$	:	Inner product
$\otimes$	:	Convolution operation
$p(x\|y)$	:	Conditional probability of x given y
$N(x\|\mu,\sigma^2)$	:	Gaussian distribution of x
$\prod$	:	Multiple multiplication
∂	:	Derivation
$\mathcal{F}$	:	Fourier transform
$\mathcal{F}^{-1}$	:	Inverse fourier transform
$\overline{(\cdot)}$	:	Conjugate operator
$\circ$	:	Array multiplication

Chapter 1

Introduction

From Bayesian perspectives [3], the prior knowledge of image and video signals plays an important role in solving problems in the image and video processing field. In this dissertation, I will solve three problems such as image deblurring, inpainting and virtual content insertion in a Bayesian framework.

Image deblurring is investigated as a long-standing problem in photography. During the image acquisition process, a blurry image is obtained due to camera shake, motion blur or defocus of lens. The task of image deblurring algorithms is to remove blur from blurry images. In the proposed algorithm, a MAP problem is generated from the linear blur model and natural image statistics. Then the optimization problem is solved by applying a data-driven approach to estimate the image prior from unblurry images.

Augmented reality (AR) is getting closer to our lives as the cloud computing services come true. The real world and the cyberspace are integrated by GPS and compass data from mobile phones and tablet computers. The enhanced content is overloaded to the devices and then users will better comprehend and enjoy the real scene when they are visiting a place of interest, watching a sports broadcasting or working in an office.

In augmented reality, it is a challenging problem to insert virtual content in video streams by blending it with spatial and temporal information. In chapter 3, an image or a movie is inserted in the specific region of tennis video streams and street view video streams. A novel detection and tracking combined approach is proposed to analyze and estimate the camera parameters in each frame. The inserted positions are determined from a Bayesian

perspective with the prior knowledge of standard tennis court model or the 3D geometry model of buildings.

The quality of video streams is related to the video compression techniques. Analog broadcast *TV* and *VCR* movies have been replaced by *HDTV*, *DVD* and *Blu-ray*. Furthermore, the ultra-high definition video is emerging on the way. Therefore, video coding standards have been developing rapidly to follow the trend of high resolution demands. Coding efficiency has been improved tremendously in the past decades from MPEG2, H.264/AVC to the on-going HEVC by adopting new tools in the codec structures.

However, it costs numerous modifications in the complex coding systems. Therefore, it is desirable to consider alternative standard-compliant approaches without modifying the codec structures. In chapter 4, an exemplar-based data pruning video compression scheme for intra frame is introduced. Data pruning is used as a pre-processing tool to remove part of video data before they are encoded. At the decoder, missing data is reconstructed by a sparse linear combination of similar patches. This process is regarded as an inpainting problem, which is solved in a Bayesian framework with a sparseness image prior.

1.1 Prior Work

1.1.1 Image Deblurring

Single image deblurring has been extensively studied in the image processing and computer vision field. A blurry image comes from the real image acquisition process due to the camera shake, object motion or defocus of lens. One way to eliminate camera shake is using a tripod or other specific devices such as hardware in [4, 5], but it's inconvenient to carry them around especially when smartphones and small cameras are favored these days. By contrast, there is another way to remove blur by applying the image deblurring algorithms.

The task of image deblurring is to remove blur from the blurry image and recover the sharp latent image. When blur kernel is spatial invariant, the deblurring problem is equivalent to the deconvolution problem. It is divided into non-blind deconvolution when the blur kernel is known and blind deconvolution when the blur kernel is unknown. In blind deconvo-

lution, the estimation of blur kernel and latent image are often separated. Therefore, blind deconvolution is converted to the non-blind deconvolution when blur kernel is estimated. In this dissertation, I will discuss the non-blind deconvolution problem.

Various non-blind deconvolution approaches for single images have been proposed in the literature. Classical methods such as Wiener filtering [6], Kalman filtering [7] and Richardson-Lucy method [8] were widely used, but they yielded severe ringing artifacts because they didn't take image statistics into account.

In order to suppress ringing effects and recover the image details, an optimization model in a Bayesian framework has been proposed as the main approach to attack this problem. There were several algorithms, such as the total variation (TV) regularization [9] and sparse image prior method [5, 10]. They were different in terms of image priors and regularization methods. The ringing artifacts were suppressed well when the observed image contained only a small amount of noise which was approximated as a Gaussian distribution. However, these methods had limited restoration performances because the Gaussian noise model was failed when a large amount of noise existed.

Another non-blind deconvolution method by Yuan [11] was built on the bilateral Richardson-Lucy (BRL) filter which could recover the edge details and handle a large blur. A progressively solution was proposed to recover the image from coarse to fine. The ringing artifacts were largely reduced from their experimental results.

Blind deconvolution is more difficult than non-blind deconvolution due to the uncertainty of both kernel and latent image. The maximum a posteriori (MAP) approach is usually established at the assumptions that the image and kernel prior obey some specific distributions. Several blind deconvolution algorithms had achieved a tremendous improvement of deblurring quality in the past years.

The first camera shake removal algorithm is Fergus's deblurring algorithm [12]. It adopted a Gaussian Scale Mixture model as the image prior in a MAP framework to estimate the blur kernel. It assumed that gradients of natural images typically obeyed a heavy-tailed distribution. Then a variational Bayesian approach was proposed to achieve a good blur kernel estimation. Unfortunately, the latent image was recovered by the RL method, so it still had some space to improve. This approach also required users to manually select a small

patch with rich edge structures, which was not a completely automatical method.

Shan et al. [13] proposed a unified probabilistic model to estimate both the blur kernel and latent image. A noise model of spatial randomness and local smoothness constraints were proposed. Remarkable results were obtained when the kernel was small but the ringing effects will showed up again for a large blur kernel. Furthermore, the image prior didn't represent the actual statistics of natural images. Our non-blind deconvolution algorithm is partially based on this work.

In [14], Levin evaluated these two blind deconvolution algorithms and some derived ones theoretically and experimentally. Fergus's method achieved best deblurring results on the test images. The conclusion that both image prior and optimization algorithm were important shed some light for the future research.

How to describe the characteristics of natural images is a very important issue in deblurring algorithms. Most algorithms assume a single prior for the whole image and it usually works well. However, the heavy-tailed distribution of gradients sometimes doesn't describe the textures well. Cho in [15] proposed a content-aware image prior to better deal with different structures.

The blur kernel is often regarded as spatial invariant to make the deblurring easier to solve, but removing the partial blur is more common in practice. Levin in [16] used a sparse image prior to remove motion blur. Dai in [17, 18] proposed a novel recovery technique to simultaneously recover both foreground and background layers with the help of the matting technique and user assistance.

In all the above deconvolution algorithms, the generation of blurry image is usually regarded as a linear blur model. The assumption may not hold in practice due to various types of outliers including pixel saturation and non-Gaussian noise in [19, 20]. Cho et al. [20] proposed a novel nonlinear blur model that explicitly taken these outliers into account and it efficiently reduced the visual artifacts caused by outliers.

1.1.2 Augmented Reality

Augmented reality has been shown industrial potentials in recent years. It is related to a more general concept called mediated reality in which a view of reality is modified (possibly even diminished rather than augmented) by computer vision techniques. Augmented reality allows us to learn more about the buildings, people or other things we are looking at, or to see alternative views of the world. For example, arrows will float over roads when we use *GPS* to find the destination. Price of property for sale will display on it when we review the real properties on smart phones. Links to a person's Twitter or Facebook account will float above people's head when we google people.

Virtual ads insertion or more generally virtual content insertion is an application of augmented reality. The basic concept consists of identifying specific places in the real scene, tracking them and augmenting the scene with virtual content. Specific region detection relies on scene analysis and understanding. The challenging issues are how to less intrusively insert the contextually relevant content (what) at the right place (where) and the right time (when) with the attractive representation (how) in videos.

Some researchers attacked these issues of virtual content insertion in the past years. For example, H.Liu proposed a general VCI system in [21] which performed attention analysis to detect the higher attentive shot as the insertion time and lower attention region as the insertion place. Unfortunately, the structure information of each frame was not utilized to find the meaningful object for content augmentation. Moreover, their insertion is only occurred in images and it had inevitaly annoying warp noises. In [22], an approach to insert virtual ads on the billboard of a video was proposed. They tried to automatically find and segment the planar surfaces in the scene as the insertion places. It had a poor performance because even the state-of-art segmentation techniques couldn't prove good performance in planar surface recognition.

Inserting an image in sports videos has been studied for its large commercial profits. The general way to add virtual content in a live video is manually inserting it by professional editors. One example is using the inserted lines or images to denote the starting positions in the football broadcastings. However, it is very labor-intensive and inefficient for broadcasting

real-time sports videos in this way. Compared to the general system [21, 22], automatic content insertion in sports videos looks easier because of the known domain knowledge. For example, Wan et al. in [23] selected the region above the goalmouth bar or the soccer central ellipse for placing virtual ads. It only worked for still images and the detection was not reliable. Chang et al. in [24] applied the tennis court model fitting and tracking to dynamically insert ads, in which visual acuity was analyzed and color harmonization was performed to reduce disturbance of virtual ads to viewers.

Similar to insertions in sports videos, inserting an image into street view videos also shows potential values especially with the cloud computing development. However, it is more challenging to implement it in street scenes due to difficulties of detecting the building facades. Especially, building or street recognition in [25] provides a clue for relevant content selection. If we know the (registered) user's information who is watching or capturing (via a mobile device) the video, targeted ads is enabled to be much more appealing. Alvarez et al. in [26] proposed an interactive system that allowed inserting a virtual picture in the architecture or extracting a poster from a certain building facade. This tool needed a few user interactions such as selecting two groups of parallel lines for vanishing points identification and camera calibration. It didn't address how to insert new elements in videos, which is obviously more difficult.

Extracting regions in a 2D image that corresponded to a very rich class of regular patterns on a planar surface in 3D world is an interesting topic in computer vision. One work to address this problem is Zhang's work in [27]. The basic idea of their approach is to view each image as a matrix and seek a transformation that gives rise to a low-rank matrix subject to sparse errors. By solving a ℓ^0 norm regularization problem, the homography matrix is estimated without estimating the vanishing points which may cause many outliers. More specifically, the rectangular structure detection on the building facade is investigated in [28, 29]. The corner and dominant direction verification are proposed by Zhang in [28], which achieves a robust and plausible result. The rectangular verification in my proposed VCI system adopts similar approaches to delete invalid rectangulars.

1.1.3 Video Compression

G.Sullivan in [30] introduced the recent developments in video coding standardization organized by a new Joint Collaborative Team on Video Voding (*JCT-VC*) formed by ITU-T VCEG and ISO/IEC MPEG. The on-going High Efficiency Video Coding (*HEVC*) aims to reduce the bit rate by half with equal subjective quality compared to the *AVC* high profile. HEVC is targeted at the next-generation *HDTV* display and content capture systems which feature progressive scanned frame rates and display resolutions from *QVGA*(320×240) at the low end up to 2560×1600 cropped from 4K×2K Ultra *HDTV* at the high end.

HEVC has released the *HEVC* Test Model (*HM*) software [31] as the basic framework and expect to complete the first version in 2012 or 2013. There are several new features in current *HM5.0* version. One of the most beneficial elements for higher compression performance comes from the introduction of larger block structures with flexible mechanisms of sub-partitioning. Instead of macroblock structure in previous video coding standards, *coding units* (*CUs*) which define a sub-partitioning of a picture into rectangular regions with variable sizes are adopted. CUs contain several *prediction units* (*PUs*) and *transform units* (*TUs*). Either the intra-picture or inter-picture prediction is selected at the level of PU. An integer spatial transform is adopted with a selectable block size ranging from 4×4 to 64×64 for inter modes. A new *mode-dependent directional transform* (*MDDT*) with block sizes 4×4 and 8×8, and a *rotational transform* for block sizes larger than 8×8 are adopted for intra modes at the level of TU. In addition, two context-adaptive entropy coding schemes for low-complexity and high-complexity cases are used to accommodate different complexity requirements.

Currently HEVC achieved at least 40% bit rate reduction with equal subjective quality as reported in [30] and the coding efficiency will be further improved in the near future. Adding new and feasible tools in video coding structures is one way to achieve the goal of bit rate reduction, but it costs numerous modifications in the complex coding system. Therefore, it is desirable to use the processing techniques while maintaining the codec structures unchangeable.

There have been some prior efforts in using pruning and recovering tools to achieve such

goal. For example, a texture replacement based method was used to remove texture regions in a movie before sending it to the encoder, and re-synthesize the texture regions after decoding the bitstream in [32, 33]. Compression efficiency was enhanced because only synthesis parameters were sent to the decoder, which were smaller than the regular transformation coefficients of those regions. However, the texture synthesis was too weak to obtain good reconstruction results. In [34, 35, 36], a texture synthesis and edge-based inpainting scheme was proposed to remove some regions at the encoder side. The removed content was recovered at the decoder with the help of region masks. The idea that separating the edges from textures was a good attempt to combine high-level and low-level vision problems together, but the representation of edges was too tricky to obtain gains.

Liu in [37] proposed a patch-based inpainting image compression scheme. Some patches were removed at encoder and recovered at decoder. The differences of similar patches were sent to the decoder too. It achieved a better performance than JPEG. However, encoder and decoder syntax were modified so as to code the non-removed regions correctly. Therefore, it was not an exact out-of-loop approach and difficult to be implemented in practice. In [38, 39], a line removal based method was proposed to rescale a video to a smaller size by selectively removing some horizontal and vertical lines in a least-square minimization framework. It was an out-of-loop approach, and did not require modifying codec structures. However, completely removing certain lines may damage the natural image characteristics and it failed to achieve compression gain according to their experimental results.

All the above schemes only had limited sources to recover the missing data and hence constrained coding efficiency improvement. Technicolor conducted some research on data pruning for video compression. For example, in [40], Zhang proposed a data pruning scheme using sampling-based super-resolution techniques. The full resolution frame was sampled into several smaller-sized frames, and hence the resolution was reduced. At the decoder, the high-resolution frame was re-synthesized from downsampled frames with the help of metadata received from the encoder side. In [41], an exemplar-based video compression scheme with super-resolution method was proposed. A representative patch library was trained from the original video. Afterwards, the video was downsized to a smaller size. The downsized video and the patch library were both sent to the decoder. The full resolution video was derived

from the downsized video by exemplar-based super-resolution [42] using the patch library. However, there was little gain from experiments, because a substantial redundancy existed in the patch library and downsized frames.

1.2 Contributions of This Dissertation

In this dissertation, three work will be introduced on non-blind deconvolution, augmented reality and video compression from a Bayesian perspective. A non-blind deconvolution algorithm is proposed to remove blur from a blurry image by solving a MAP problem. A generic virtual content insertion system is proposed to insert an image or a movie on a specific region of sports and street view video streams. The inserted positions are estimated through line detection and vanishing points extraction in a Bayesian framework. An exemplar-based data pruning video compression scheme for intra frame is proposed to improve the video coding efficiency without modifying the codec structures. In pruned data recovery process, a ℓ^1 norm regularization problem is established to reconstruct the missing data.

The contributions of this dissertation are applying the Bayesian perspective to solve some problems in image and video processing.

- A data-driven non-blind deconvolution algorithm is proposed. In the Bayesian framework, the image prior is an important part. It is represented by the Gaussian Scale Mixture (GSM) model and learned from related unblurry images in the proposed algorithm. The GSM model represents the natural image statistics well, and therefore more image details are recovered.

- Inserting virtual content in sports video streams is quite mature. However, the insertion was usually controlled by professional editors and only lasted for several frames. More importantly, cameras were setup by themselves and then camera calibration was not a problem. By contrast, in my VCI system, virtual content is inserted in the video streams from the Internet or digital camcorders. The difficulty is that camera positions are totally unknown. Therefore, camera calibration is automatically estimated frame by frame with a detection and tracking combined approach, and virtual content is

analyzed and inserted in feasible regions with a standard court prior in a Bayesian framework.

- Compared to sports videos, virtual content insertion in street view videos is much more challenging, because there is no standard model of building facades. I attack the problem to estimate the inserted rectangular in a Bayesian framework with the 3D geometry model of building facades through vanishing points extraction. Moreover, motion stabilization is applied to remove the jittering effects. To the best of my knowledge, this is the first successful system to automatically insert an image or a movie in street view video streams.

- Adding new tools in video coding standards is a normal way to improve the coding efficiency, but it's very complex to modify the codec structures. Therefore, I try to use the pre- and post- processing techniques to achieve such goal without modifying codec structures. In the proposed exemplar-based data pruning (*EDP*) scheme for intra frame, some video data is removed and replaced with a constant value before encoding. A patch library is created from original video data and it helps to determine the pruning decision. Assistant metadata containing the pruning information is sent to the decoder as well. At the decoder, the pruned data is reconstructed by a sparse linear combination of similar patches from the established patch library with non-pruned decoded video data. It benefits from the sparseness prior to reconstruct textures.

1.3 Roadmap

In chapter 2, a data-driven non-blind deconvolution algorithm is introduced in a Bayesian framework. How to represent the image prior from related unblurry images is discussed in details. The first-order derivative of natural images obeys a heavy-tailed distribution. The GSM model fits it well and serves as the image prior. Therefore, an optimization problem is derived from likelihood and image prior, and is solved in an iterative approach.

In chapter 3, a novel automatic virtual content insertion system for sports and street view video streams is proposed. The system starts from detecting specific regions, such as

the soccer goalmouth, tennis court or building facades at the first meaningful frame. Visual tracking are then executed to calibrate camera in the following frames. Inserted positions are estimated in a Bayesian framework with the standard court model and 3D geometry model as the image prior. Then virtual content is warped and blended into detected regions. Two typical VCI systems are discussed respectively for sports and street view video streams to insert an image or a movie above the soccer goalmouth bar, on tennis court and on the modern building facades.

In chapter 4, an EDP video compression scheme for intra frame is introduced. In this scheme, data pruning is used as a pre-processing technology to remove part of video data before they are encoded. A pruning strategy based on rate-distortion optimization is introduced. The generation of metadata containing the pruning decision and position information is also explained.

At the decoder, pruned data is recovered by inferring from the generated patch library with non-pruned decoded data and the assisted metadata. Instead of copying the most similar patch from the patch library, a novel post-processing method is proposed. The patch to be filled is represented by a sparse linear combination of candidate patches under some constraints. Linear combination coefficients are obtained through solving a ℓ^1 norm regularization by a novel Bregman iteration algorithm [43].

Finally in chatper 5, I will make some conclusions on the data-driven non-blind deconvolution algorithm, the automatic VCI system and the EDP video compression scheme for intra frame. In addition, limitations and future work are discussed.

1.4 Notes

Chapter 3 is implemented during my internship at Huawei US Technology (Bridgewater, NJ). Part of its content is from the patent [44] and the ICIP paper [45].

The pre-processing part in chapter 4 is implemented during my internship at Technicolor Research Center (Princeton, NJ). Part of its content is from the patent [46].

Chapter 2 and the post-processing part of chapter 4 are implemented during my PhD studies. Two manuscripts are prepared for submission.

Chapter 2

Non-blind Deconvolution

Image deblurring has been extensively studied in the image processing and computer vision field for decades. During image acquisition, camera shake, object motion or defocus of lens all cause blurry images. The elimination of camera shake may be obtained by a tripod or other specific devices such as hardware in [4], but it's inconvenient to carry a tripod especially when smartphones and small cameras are favored these days. Therefore, it's a usual way to apply image deblurring algorithms on blurry images. The task is to remove blur from a blurry image and recover the sharp latent image. For example, the left image in Figure 2.1 is a blurry image captured by a camera. The right image is the restored result with the image deblurring algorithm in [13].

The generation of blurry images is often regarded as a linear blur model, that is, a blurry image is obtained through a convolution operation between the latent image and blur kernel. Therefore, the deblurring problem is equivalent to the deconvolution problem. Image deconvolution is divided into non-blind deconvolution when the blur kernel is known and blind deconvolution when the blur kernel is unknown. In a blind deconvolution algorithm, the estimation of blur kernel and latent image are often separated. Therefore, blind deconvolution is converted to the non-blind deconvolution when blur kernel is estimated, and hence non-blind deconvolution serves as an important step in blind deconvolution algorithms. In this chapter, only non-blind deconvolution is discussed.

Single image deconvolution has been investigated for several years. However, there is no perfect solution due to inaccurate image prior and optimization algorithms. In this chapter,

(a) *Blurry image* **(b)** *Recovered image*

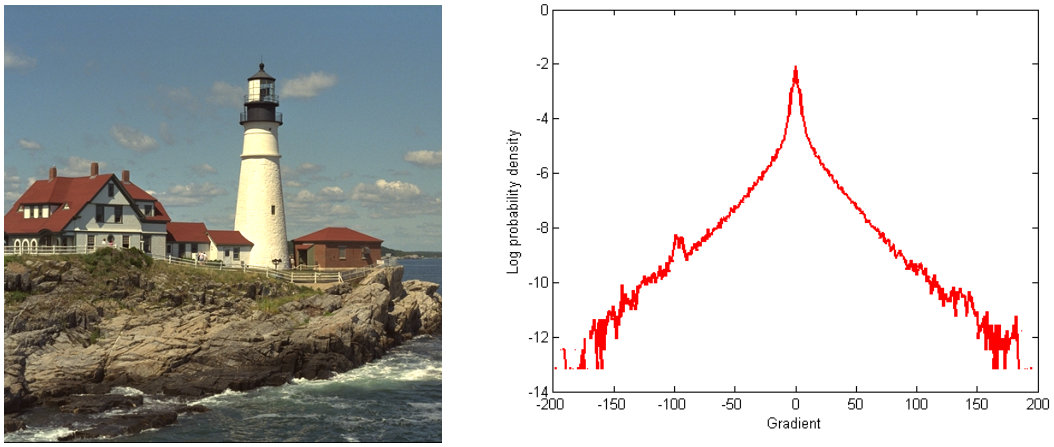
Figure 2.1: *Image deblurring is to remove blur from (a) to recover (b)*

a data-driven non-blind deconvolution is proposed in a MAP model. The image prior is represented by the GSM model learned from related unblurry images. For example, the clear left view image will help to remove blur from the blurry right view image, since the GSM model extracted from it represents the image statistics of the right view latent image. Details of generating and solving the MAP problem are introduced in the following sections.

2.1 Natural Images Characteristics

In low level vision problems such as image denoising, deblurring, interpolation and inpainting, the statistics of natural images are extensively studied as an important factor to solve these problems. The distribution of a natural image processed by a band-pass filter has a distinctive form across many different images, that is, the distribution has a sharp peak at the zero and a heavy tail at large values. Figure 2.2 shows that the Log histogram of the first order derivative of an image is a heavy-tailed distribution. It reflects the fact that intensities of neighboring pixels are usually similar in smooth area while different in structural area in natural images.

This consistent property makes it invaluable to the low level vision applications as a statistical prior of images. Roth [47] proposed a Field of Experts (FOE) framework in image denoising and inpainting. Tappen [48, 49] used a sparse image prior in super-resolution and



(a) *Natural Image* **(b)** *Distribution of gradients*

Figure 2.2: *Distribution of gradients of an image. Right: The y-axis has a logarithmic scale to show the heavy tails of the distribution. The x-axis is gradient values*

demosaicing. Simoncelli [50] used this sparse prior as the basis of denoising algorithm and also applied this statistical model to compression. Levin [51] used an exponential family model of image statistics in image inpainting. Apostoloff [52] applied the image statistics in video matting.

The statistical model of images has been used in many deblurring algorithms. Shan in [13] used a simple two piece-wise function to estimate the distribution of gradients. Levin in [16] used the statistics of derivative filters in blurry images to remove the motion blur from different part of a blurry image. Fergus in [12] used the Gaussian Scale Mixture (GSM) model as the image prior to estimate the blur kernel in a Bayesian framework. Cho in [15] proposed a content-aware image prior to better deal with the texture area because the heavy-tailed distribution is not a good model for textures.

2.1.1 Gaussian Scale Mixture Model

The heavy-tailed distribution of natural image gradients is obviously not a Gaussian distribution, but it is well fit by the GSM model in [53]. In addition, the student T distribution used in the successful FOE model [47] is a variation of GSM. In my non-blind deconvolution algorithm, the image prior is represented by the GSM model in the MAP generation. A

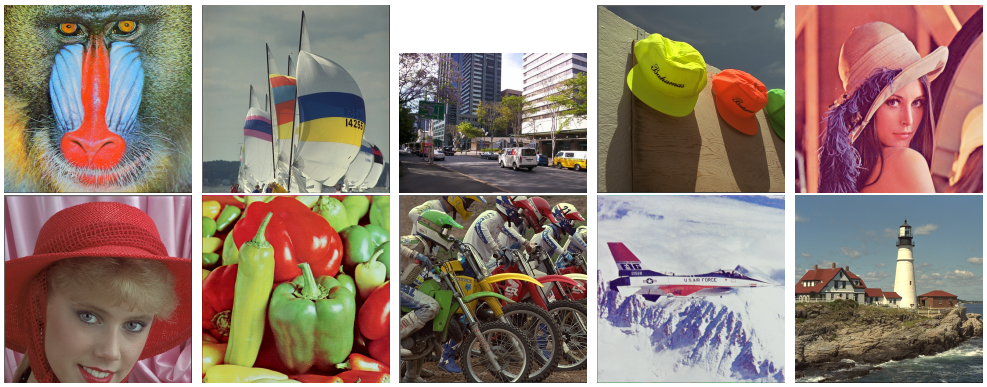


Figure 2.3: *Natural images for GSM model learning*

GSM potential is expressed by a mixture of zero mean Gaussians:

$$\Psi(x) = \sum_{j=1}^{J} \frac{c_j}{\sqrt{2\pi}\sigma_j} \exp(-\frac{x^2}{2\sigma_j^2}) \tag{2.1}$$

The GSM model is composed of multiplications of GSM potentials as follows:

$$Pr(x; \{\omega_k\}) = \frac{1}{Z(\{\omega_k\})} \prod_{i,k} \Psi(\omega_{ik}^T x) \tag{2.2}$$

where $\{\omega_k\}$ is the filters passing through x. $Z(\{\omega_k\})$ is a normalization term. Although this term is intractable, it's usually ignored in the MAP problem since it is a constant value in posterior comparisons.

2.1.2 GSM Parameters Learning

Parameters of a GSM potential $\Theta = \{c_j, \sigma_j\}_{j=1}^{J}$ are estimated from data set $\{x_1, x_2, \cdots, x_N\}$ by the Expectation-Maximization(EM) algorithm.

EM algorithm is a general iterative algorithm for parameter estimation by maximum likelihood when some of the random variables involved are not observed, which is considered as missing or incomplete. EM algorithm is based on the intuitive idea that repeating replacing missing data by estimated values until convergence will result parameter estimation. There are two steps in EM algorithm. E-Step uses estimated parameters as true values to calculate the expectation of a conditional probability and M-Step is to maximize the expectation from E-Step. The details are found in [54].

Each x_n is generated from one of the J hidden states with the corresponding gaussian probability. Let h_n denotes the state of x_n and $h_n \in \{1, \cdots, J\}$. We get the following equation.

$$logP(\mathcal{X}|\Theta) = log\prod_{n=1}^{N} P(x_n|\Theta) = \sum_{n=1}^{N} log\sum_{j=1}^{J} c_j N(x_n|0, \sigma_j^2) \tag{2.3}$$

Given values of $\mathcal{H}$:

$$logP(\mathcal{X}, \mathcal{H}|\Theta) = \sum_{n=1}^{N} log(P(x_n|h_n)P(h_n)) = \sum_{n=1}^{N} log(c_{h_n} N(x_n|0, \sigma_{h_n}^2)) \tag{2.4}$$

- **E-step**

  In the E-step, the conditional expectation $E_{\mathcal{H}|\mathcal{X},\Theta}\{logP(\mathcal{X}, \mathcal{H}|\Theta)\}$ is determined.

$$\begin{aligned} Q(\Theta|\Theta^t) &= \sum_{h\in\mathcal{H}} logP(\mathcal{X}, \mathcal{H}|\Theta)P(h|\mathcal{X}, \Theta^t) \\ &= \sum_{j=1}^{J}\sum_{n=1}^{N} log(c_j N(x_n|0, \sigma_j^2))P(j|x_n, \Theta^t) \\ &= \sum_{j=1}^{J}\sum_{n=1}^{N} log(c_j)P(j|x_n, \Theta^t) + \sum_{j=1}^{J}\sum_{n=1}^{N} log(N(x_n|0, \sigma_j^2))P(j|x_n, \Theta^t) \end{aligned} \tag{2.5}$$

  where $P(j|x_n, \Theta^t)$ is denoted by ω_j^n as:

$$\begin{aligned} \omega_j^n &= P(j|x_n, \Theta^t) \\ &= \frac{P(x_n|j, \Theta^t)P(j|\Theta^t)}{P(x_n|\Theta^t)} \\ &= \frac{P(x_n|j, \Theta^t)P(j|\Theta^t)}{\sum_{k=1}^{J} P(x_n|k, \Theta^t)P(k|\Theta^t)} \\ &= \frac{c_j^t N(x_n|0, {\sigma_j^t}^2)}{\sum_{k=1}^{J} c_k^t N(x_n|0, {\sigma_k^t}^2)} \end{aligned} \tag{2.6}$$

  In conclusion, the E-step is given by:

$$Q(\Theta|\Theta^t) = \sum_{j=1}^{J}\sum_{n=1}^{N} log(c_j)\omega_j^n + \sum_{j=1}^{J}\sum_{n=1}^{N} log(N(x_n|0, \sigma_j^2))\omega_j^n \tag{2.7}$$

- **M-step**

  In the M-step, the parameters c_j and σ_j^2 are obtained respectively by maximizing this Equation (2.7) with respect to Θ.

  (1) M-step for c_j:

$$\arg\max_{\{c_j\}} \sum_{j=1}^{J}\sum_{n=1}^{N} log(c_j)\omega_j^n, \quad s.t. \sum_{j=1}^{J} c_j = 1; \tag{2.8}$$

  By differentiating the above equation with a Lagrange multiplier λ, the weight c_j is obtained as:

$$\frac{\partial}{\partial c_j}\left[\sum_{j=1}^{J}\sum_{n=1}^{N} log(c_j)\omega_j^n + \lambda(1 - \sum_{j=1}^{J} c_j)\right] = 0 \tag{2.9}$$

$$c_j = \frac{1}{\lambda}\sum_{n=1}^{N}\omega_j^n \tag{2.10}$$

  The solution is given by Equation (2.10). λ is obtained by normalizing $\sum_{j=1}^{J} c_j = 1$, then $\frac{1}{\lambda}\sum_{n=1}^{N} 1 = 1$, $\lambda = N$. Therefore, weight c_j is obtained by

$$c_j = \frac{1}{N}\sum_{n=1}^{N}\omega_j^n \tag{2.11}$$

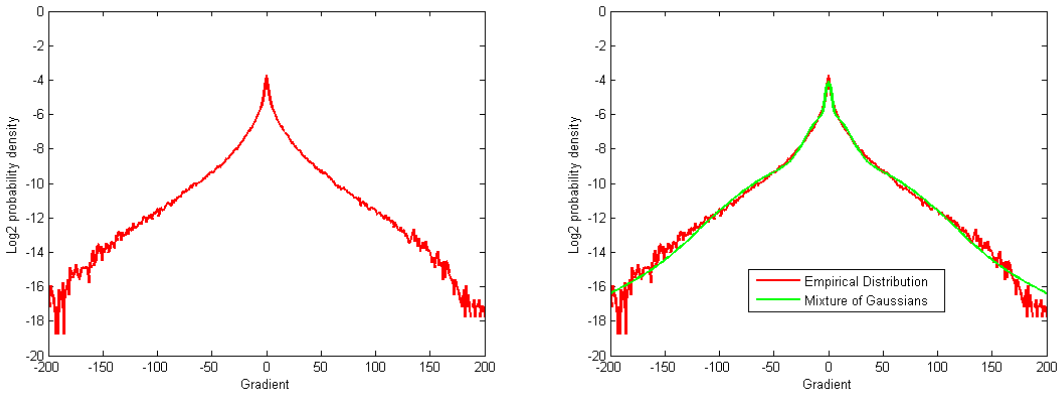
  (2) M-step for σ_j^2:

$$\arg\min_{\{\sigma_j^2\}} \sum_{n=1}^{N}\left(\frac{1}{2} log\sigma_j^2 + \frac{x_n^2}{2\sigma_j^2}\right)\omega_j^n \tag{2.12}$$

  Taking the derivative with respect to σ_j^2 and equating it to zero, the variances are obtained as:

$$\sum_{n=1}^{N}\left[\frac{1}{\sigma_j^2} - \frac{1}{(\sigma_j^2)^2}x_n^2\right]\omega_j^n = 0$$

$$\sigma_j^2 = \frac{\sum_{n=1}^{N} x_n^2\omega_j^n}{\sum_{n=1}^{N}\omega_j^n} \tag{2.13}$$



(a) *Empirical distribution* **(b)** *Estimated GSM model*

Figure 2.4: *Illustrations of empirical distribution in red and estimated GSM model in green: GSM model can well fit the empirical distribution*

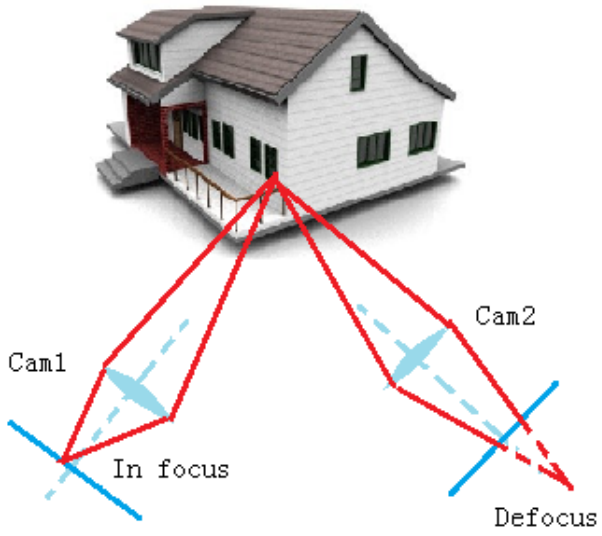
The first-order derivative of natural images in both horizontal and vertical directions $\{\partial_x I, \partial_y I\}$ is treated as the input data $\{x\}$. Then the GSM model is obtained by EM algorithm from the above steps. The empirical distribution from ten test images (shown in Figure 2.3) is well fit by the GSM model. In Figure 2.4, the empirical distribution and the estimated GSM model are denoted by the red and green curves respectively. From experimental data, GSM is a good description of natural images statistics.

2.2 Non-blind Deconvolution Formulation

During natural image acquisition, camera shake, object moving or defocus of lens all cause a degraded photograph. In the math modeling, the blurry image B is regarded as coming from a convolution process by the latent image L and a point spread function (PSF) f. Due to ubiquitous noises in camera systems, some noises are also involved in this process. In conclusion, the blurry image generation is modeled as:

$$B = L \otimes f + n \tag{2.14}$$

According to this blur model, the non-blind deconvolution is to recover L when B and f are known; while the blind deconvolution is to estimate both L and f when only B is known. In this chapter, the non-blind deconvolution is discussed under this blur model. The



(a) *Stereo images generation*

(b) *Left view image*

(c) *Right view image*

Figure 2.5: *Illustrations of stereo images generation that the left view image (b) is in focus while the right view image (c) is blurry. Removing blur from a blurry image will benefite from the related unblurry images*

blur kernel f is assumed as a known shift-invariant value. However, it's not true in practice and f is varied with regions. How to deal with the shift variant blur kernel is a topic in the future research.

2.3 Proposed Algorithm

Single image deconvolution has been extensively studied, but no perfect solution exists due to inaccurate image prior and optimization algorithm. Model-driven and data-driven approaches are two main methods for solving image restoration problems. In order to represent the image prior accurately, a data-driven non-blind deconvolution is proposed. A lot of related images are available in real world, which makes the data-driven approach feasible in removing blur from a blurry image. For example, stereo images are getting more attentions now as there are many 3D applications emerging on smartphones and 3DTV. Two view

images have some common scenes. Therefore, if the left view image is captured clear but the right view image is blurry, the left view image will help in deblurring the right view image.

The above scenario is similar to the amblyopia in medicine science. Amblyopia or "lazy eye" is a disorder of the visual system that one eye has a vision deficiency but the other eye acts normally. Although it needs treatments, a person with amblyopia could still view and understand the scenes without any problem, because human beings have an ability to automatically correct the disordered eye with the help of the other eye. It aspired us to adopt a data-driven approach to utilize the related image pairs.

Similar to most of image deblurring algorithms, a Maximum A Posterior (MAP) model is applied to estimate the latent image L when the blur kernel f and the clear related image I_L are given. In Equation (2.15), the posterior $p(L|B, f)$ is maximized, where $p(B|L, f)$ represents the likelihood and $p(L)$ is the image prior. Therefore, how to describe the likelihood and image prior and how to solve this optimization problem are fundamental components of the proposed algorithm. The details are introduced in the following sections.

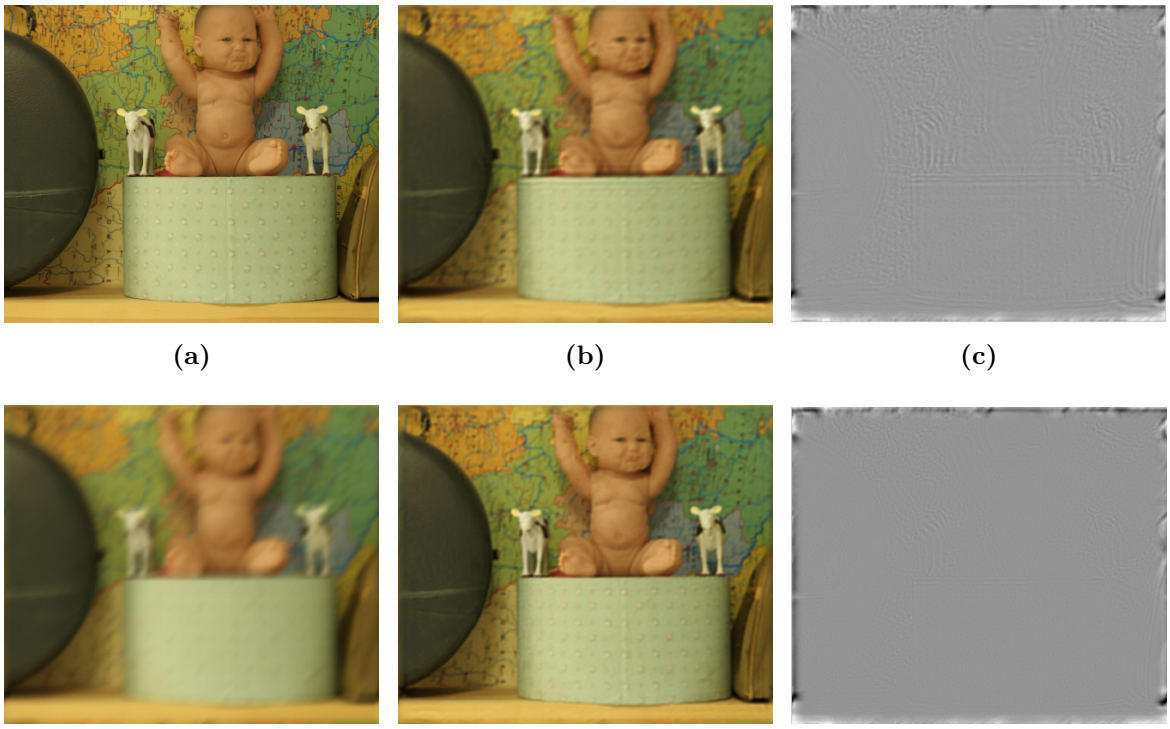
$$p(L|B, f) \propto p(B|L, f)p(L) \tag{2.15}$$

2.3.1 Likelihood Representation

The likelihood $p(B|L, f)$ is determined by the noise model, because $n = B - L \otimes f$ and hence $p(B|L, f) = p(n)$. In the previous work such as Fergus [12], image noise n was usually treated as the white Gaussian noise, but such model didn't match the actual noises and caused serious ringing effects. In Shan's paper [13], higher-order derivatives were used to represent the noise model, in order to capture the spatial randomness of noises. I use the same noise model as Shan's in my algorithm.

Image noise n is regarded as an independent and identically distributed (i.i.d) Gaussian variable in each pixel. The white Gaussian model $\prod_i N(n_i|0, \zeta_0)$, the first-order $\prod_i N(\nabla n_i|0, \zeta_1)$ and second-order $\prod_i N(\nabla^2 n_i|0, \zeta_2)$ derivative of noises are combined to represent the spatial random noises.

Derivatives are defined in the horizontal and vertical directions. For example, the first-order derivative is expressed by $\partial_x n$ and $\partial_y n$ respectively. Since n is a Gaussian variable in



(a) **(b)** **(c)**

(d) **(e)** **(f)**

Figure 2.6: *Effects of likelihood (a) Ground truth image. (d) Blurry image. (b) Estimated latent image using Gaussian noise model. (e) Estimated latent image using mixed-order Gaussian noise model. (c) Noise from (b). (f) Noise from (e)*

each pixel, $\partial_x n$ and $\partial_y n$ also follow the i.i.d Gaussian distribution.

$$\begin{aligned} \partial_x n(x,y) &= n(x+1,y) - n(x,y) \\ \partial_y n(x,y) &= n(x,y+1) - n(x,y) \end{aligned} \tag{2.16}$$

The second-order derivative is represented by $\partial_{xx} n$, $\partial_{xy} n$ and $\partial_{yy} n$. They are also the i.i.d Gaussian variables and the definitions are as follows:

$$\begin{aligned} \partial_{xx} n(x,y) &= n(x+2,y) - 2n(x+1,y) + n(x,y) \\ \partial_{xy} n(x,y) &= n(x+2,y) - n(x+1,y) - n(x,y+1) + n(x,y) \\ \partial_{yy} n(x,y) &= n(x,y+2) - 2n(x,y+1) + n(x,y) \end{aligned} \tag{2.17}$$

Assume the standard variance (SD) of n as ζ_0, then the SD of ∂n and $\partial^2 n$ are $\zeta_1 = \sqrt{2}\zeta_0$, $\zeta_2 = \sqrt{2}\zeta_1 = \sqrt{2^2}\zeta_0$.

In order to explicitly express the higher-order derivatives of noises, ∂^* denotes the operator of any partial derivative and $\kappa(\partial^*)$ denotes the order. For instance, $\partial^* = \partial_{xx}$ and $\kappa(\partial^*) = 2$. If $\kappa(\partial^*) = q$, the SD of gaussian variable $\partial^* n$ is $\zeta_q = \sqrt{2^q}\zeta_0$.

In conclusion, the noise model is represented by a combination of i.i.d Gaussian variables $\partial^* n$ of zero-order, first-order and second-order derivative. The likelihood is denoted as:

$$\begin{aligned} p(B|L,f) &= \prod_i \prod_{\partial^* \in \Theta} N\left(\partial^* n_i | 0, \zeta_{\kappa(\partial^*)}\right) \\ &= \prod_i \prod_{\partial^* \in \Theta} N\left(\partial^* B_i | \partial^* (L \otimes f)_i, \zeta_{\kappa(\partial^*)}\right) \end{aligned} \tag{2.18}$$

where $B_i \in B$ denotes pixel intensity in the blurry image. $\Theta = \{\partial^0, \partial_x, \partial_y, \partial_{xx}, \partial_{xy}, \partial_{yy}\}$ is the set of all the derivative operators from zero-order to second-order. $\partial^0 n_i = n_i$ denotes the white Gaussian variable.

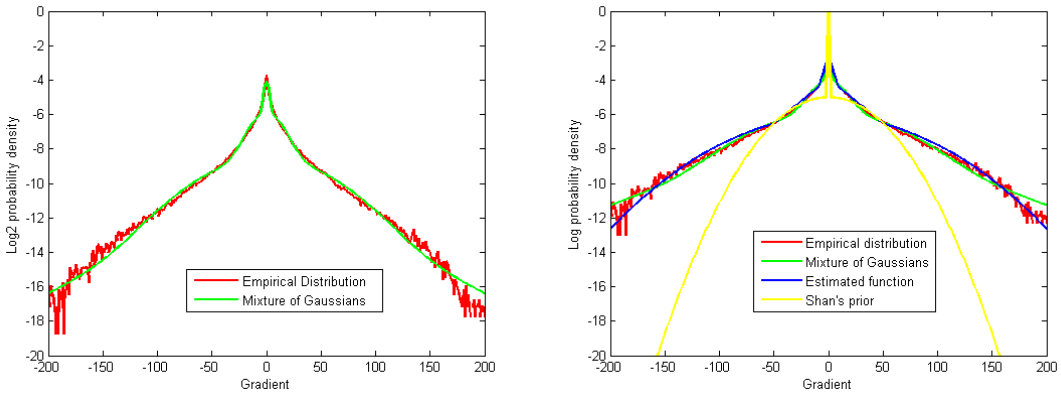
By comparing the estimated noise $n = B - L \otimes f$, the deconvolution algorithm with less structure information exceeds the one with more structures. The ideal noises should have spatial randomness. Figure (2.6) illustrates the effects of different likelihood. (a) and (d) are the ground truth and blurry image. (b) is the recovered image using the Gaussian noise model $\prod_i N(n_i|0,\zeta_0)$. (e) is recovered image in higher-order noise model in my algorithm. As we can see, the noise $n = B - L \otimes f$ in (f) has less structure than in (e). Therefore, the likelihood in proposed algorithm is better than the simple white Gaussian noise model.

2.3.2 Image Prior Representation

The image prior $p(L)$ describes the natural image statistics and determines the details of the recovered latent image. In section (2.1), the heavy-tailed distribution of natural image gradients is well fit by the GSM model. Therefore, the image prior is represented by the GSM model in the proposed algorithm.

$$P(L) = \prod_i \sum_{j=1}^{J} c_j N\left(\partial L_i | 0, \sigma_j^2\right) \tag{2.19}$$

When only the blurry image exists, parameters of GSM model $\{c_j, \sigma_j^2\}$ must be learned from training images. However, such prior represents an average image statistic and it may not describe a certain image accurately. By contrast, when related unblurry images exist,



(a) **(b)**

Figure 2.7: *Illustrations of different image priors. (a) Empirical distribution and fitted GSM model; (b) Proposed image prior and Shan's image prior*

they share some common scenes and thus the image gradients follow a similar distribution. Therefore, the image prior learned from the clear images is also a good description of this blurry image. In my algorithm, the GSM model is extracted from the related unblurry images and it serves as the image prior in the MAP model. In all, $J = 4$ mixtures of Gaussian model are adopted in the GSM model and parameters are obtained by EM algorithm shown in section (2.1.2).

In the MAP method, the log-prior $log \prod_i \sum_{j=1}^{J} c_j N\left(\partial L_i | 0, \sigma_j^2\right)$ is usually taken. However, it's very difficult to find the optimized values when using this term because $\sum$ is in the logarithm. In Shan's method, a simple two piece-wise function is modeled to fit the empirical distribution, which is not accurate enough to describe the latent image statistic. In order to attack the problem, a convex function $\Phi(x)$ is used to fit the GSM model in the proposed algorithm.

$$\sum_{j=1}^{J} c_j N\left(\partial L_i | 0, \sigma_j^2\right) = e^{\Phi(\partial L_i)} \tag{2.20}$$

Now the logarithm of prior is represented by $\sum_i \Phi(\partial L_i)$, so that this MAP problem has a closed-form solution. The image prior is represented by the following expression:

$$p(L) = \prod_i e^{\Phi(\partial L_i)} \tag{2.21}$$

The convex function $\Phi(x)$ is a three piece-wise function and the parameters are calculated from the corresponding GSM model.

$$\Phi(x) = \begin{cases} -k_1|x| - c_1 & |x| \leq l_1 \\ -k_2|x| - c_2 & l_1 < |x| \leq l_2 \\ -(ax^2 + b) & |x| > l_2 \end{cases} \tag{2.22}$$

For example, the GSM model and empirical distribution from ten natural images are illustrated in Figure 2.7 (a). The x-axis is the gradient values ranging from $[-200, 200]$ and y-axis is the logarithm scale of the GSM model $log\ \Psi(\partial L)$ and probability density function $(pdf)\ log\ p(\partial L)$. The fitted convex function $\Phi(x)$ and the image prior used in Shan's method are also shown in (b). It clearly shows that Shan's image prior didn't match the heavy-tailed empirical distribution. On the other hand, the proposed $\Phi(x)$ fits the GSM model well and represents the natural image statistics accurately.

2.4 Algorithm Description

We analyzed the likelihood and image prior in above sections. The next step is to solve the MAP problem $p(L|B, f)$. By taking the logarithm operation, it is transformed to an energy minimization problem $E(L) = -\log(p(L|B, f))$. An optimization problem is obtained by substituting the likelihood and prior into above definitions.

$$E(L) \propto (\sum_{\partial^* \in \Theta} \omega_{\kappa(\partial^*)} \|\partial^* L \otimes f - \partial^* B\|_2^2) + \lambda_1 \|\Phi(\partial_x L) + \Phi(\partial_y L)\|_1 \tag{2.23}$$

The parameters are defined as follows:

$$\omega_{\kappa(\partial^*)} = \frac{1}{\zeta_{\kappa(\partial^*)}^2 \tau}, \quad \lambda_1 = \frac{1}{\tau} \tag{2.24}$$

Since $\kappa(\partial^*) = q > 0$, $\omega_{\kappa(\partial^*)} = 1/(\zeta_0^2 \cdot \tau \cdot 2^q)$. In my implementation, $1/(\zeta_0^2 \cdot \tau) = 50$. λ_1 is defined by users and the default value is 0.01. It will be automatically updated during the iterative process.

This MAP problem could be solved using gradient descent method, but it's very slow and sometimes doesn't converge to the global minimum. Shan proposed a novel method

which divided this problem into tractable sub-problems in their blind deconvolution algorithm. I use a similar strategy to estimate the latent image L in my non-blind deconvolution algorithm.

The energy minimization problem $E(L)$ is a highly non-convex function including high-dimension unknown variables. More importantly, it consists of several convolution operations, which makes it difficult to be solved directly. In order to separate the complex convolutions from the other terms, a variable substitution scheme is adopted. A set of auxiliary variables $\Psi = (\Psi_x, \Psi_y)$ is added to take place of the first-order derivative $\partial_L = (\partial_x L, \partial_y L)$. At the same time, a new term $\Psi \approx \partial L$ is added to guarantee that the substitution doesn't change the minimum value of $E(L)$. Therefore, this problem is equivalent to the following one:

$$\begin{aligned} E(L) \propto \quad & (\sum_{\partial^* \in \Theta} \omega_{\kappa(\partial^*)} \|\partial^* L \otimes f - \partial^* B\|_2^2) + \lambda_1 \|\Phi(\Psi_x) + \Phi(\Psi_y)\|_1 + \\ & \gamma(\|\Psi_x - \partial_x L\|_2^2 + \|\Psi_y - \partial_y L\|_2^2) \end{aligned} \tag{2.25}$$

where γ is a weight whose value will be iteratively increased from 2 until it is very large when $\{\Psi_x, \Psi_y\}$ converges to their global minimum. Therefore, in the final iteration step, $\Psi = \partial L$, which guarantees that an optimized L is the solution of $E(L)$ minimization problem.

The non-blind deconvolution algorithm is executed in iterations. When λ_1 is a certain value, optimization is iterated between Ψ and L. Firstly, optimize $\{\Psi_x, \Psi_y\}$ when L is fixed. Secondly, optimize L when $\{\Psi_x, \Psi_y\}$ are fixed. This process will end when the values converge to the global minimum. It usually converges within 16 iterations. Then λ_1 is updated to a smaller value and such iteration process will execute again until L converging. In all, this algorithm is a fast and efficient method since each iteration can be executed efficiently by separating the convolution operations apart. Table(2.1) is descriptions of this iterative optimization algorithm.

- **Updating Ψ**

  Fix the values of L and $\partial^* L$, update Ψ. The energy minimization problem (2.25) is converted as:

$$E_\Psi^t = \lambda_1 \|\Phi(\Psi_x) + \Phi(\Psi_y)\|_1 + \gamma \|\Psi_x - \partial_x L\|_2^2 + \gamma \|\Psi_y - \partial_y L\|_2^2 \tag{2.26}$$

Algorithm: Non-blind deconvolution of stereo images
Input: Blurry image B, a related clear image L_L and blur kernel f
Learn the image prior from L_L
Initialize $L \Leftarrow B$
Repeat optimizing L with updated λ_1
Repeat optimizing L
Update Ψ by minimizing the energy defined in (2.26).
Calculate L in (2.28)
until $\|\Delta L\|_2 < 1 \times 10^{-5}$ and $\|\Delta \Psi\|_2 < 1 \times 10^{-5}$ or 16 iterations performed
Update λ_1
until $\|\Delta L\|_2 < 1 \times 10^{-5}$ or 36 iterations performed
Output: L

Table 2.1: *Descriptions of the proposed algorithm*

Decompose Ψ to all elements $\psi_{i,x}$ and $\psi_{i,y}$, then E^t_Ψ is sum of sub-energy terms.

$$E^t_\Psi = \sum_i \left(E^t_{\psi_{i,x}} + E^t_{\psi_{i,y}} \right) \tag{2.27}$$
$$E^t_{\psi_{i,v}} = \lambda_1 |\Phi(\psi_{i,v})| + \gamma(\psi_{i,v} - \partial_v L_i)^2$$

Each $E^t_{\psi_{i,v}},\ v \in \{x, y\}$ is a quadratic differential function in (2.28). Therefore, Ψ is obtained by getting the closed-form solution for each pixel independently. For example, when $0 < \psi_{i,v} < l_1$, the above equation becomes $\lambda_1 |k_1 \psi_{i,v} + c1| + \gamma(\psi_{i,v} - \partial_v L_i)^2$, it has a closed-form optimized solution $\psi_{i,v} = \gamma \times \partial_v L_i - \lambda_1 k_1/(2\gamma)$. Similarly, the other cases are also solved in an efficient way. In conclusion, this step is completed quickly and result in a global minimum of $\{\Psi_x, \Psi_y\}$.

- **Updating** L

  Once $\{\Psi_x, \Psi_y\}$ are obtained, the energy minimization problem is represented as follows:

$$E^t_L = (\sum_{\partial^* \in \Theta} \omega_{\kappa(\partial^*)} \|\partial^* L \otimes f - \partial^* B\|_2^2) + \gamma \|\Psi_x - \partial_x L\|_2^2 + \gamma \|\Psi_y - \partial_y L\|_2^2 \tag{2.28}$$

  There are several convolution operations in this equation, so it's difficult to solve it directly. It's a natural way to convert it to Fourier domain because a convolution in spatial domain will convert to a multiplication in Fourier domain. Moreover, according to Plancherel's theorem [55], sum of the square of a function equals the sum of the

(a) *Blurry image* **(b)** *1st iteration* **(c)** *6th iteration* **(d)** *16th iteration*

Figure 2.8: *Illustrations of iterative optimization process. (a) blurry image (b)-(d) recovered latent image in the 1st, 6th and 16th iteration. The details are recovered from coarse to fine during this process.*

square of its Fourier transform. Therefore, E_L^t is converted to its Fourier expression:

$$\begin{aligned} E_{\mathcal{F}(L)}^t = \; & (\sum_{\partial^* \in \Theta} \omega_{\kappa(\partial^*)} \|\mathcal{F}(\partial^*) \circ \mathcal{F}(L) \circ \mathcal{F}(f) - \mathcal{F}(\partial^*) \circ \mathcal{F}(B)\|_2^2) + \\ & \gamma \|\mathcal{F}(\Psi_x) - \mathcal{F}(L) \circ \mathcal{F}(\partial_x)\|_2^2 + \gamma \|\mathcal{F}(\Psi_y) - \mathcal{F}(L) \circ \mathcal{F}(\partial_y)\|_2^2 \end{aligned} \tag{2.29}$$

By minimizing $E_{\mathcal{F}(L)}^t$, the optimized value $\mathcal{F}(L^*)$ is the Fourier transformation of L^* to minimize E_L^t.

$$L^* = \mathcal{F}^{-1}\left(\arg\min_{\mathcal{F}(L)} E_{\mathcal{F}(L)}^t\right) \tag{2.30}$$

$\mathcal{F}(L^*)$ is easily obtained by setting the first-order derivative of $E_{\mathcal{F}(L)}^t$ to zero. The closed-form solution is represented by:

$$L^* = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(f)} \cdot \mathcal{F}(B) \cdot \Delta + \gamma \overline{\mathcal{F}(\partial_x)} \cdot \mathcal{F}(\Psi_x) + \gamma \overline{\mathcal{F}(\partial_y)} \cdot \mathcal{F}(\Psi_y)}{\overline{\mathcal{F}(f)} \cdot \mathcal{F}(f) \cdot \Delta + \gamma \overline{\mathcal{F}(\partial_x)} \cdot \mathcal{F}(\partial_x) + \gamma \overline{\mathcal{F}(\partial_y)} \cdot \mathcal{F}(\partial_y)}\right) \tag{2.31}$$

where $\Delta = \sum_{\partial^* \in \Theta} \omega_{\kappa(\partial^*)} \overline{\mathcal{F}(\partial^*)} \cdot \mathcal{F}(\partial^*)$

In order to iteratively update Ψ and L, γ is automatically increased to control the similarity of Ψ and ∂L. If γ is a large value, the convergence will be quite slow. If γ is set to a small value, the convergence doesn't hold. In current implementation, the initial value is 2. The value of λ_1 is also updated and hence the details are recovered gradually. Figure 2.8 is an example to show that blur is removed and latent image is recovered gradually during the iterative process.

Last but not least, how to deal with the boundary area is also important to the restoration quality. Because this algorithm needs operations in Fourier domain, it will cause ringing

Image	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6
Blurry	0.621	0.727	0.546	0.301	0.493	0.382
RL	0.889	0.833	0.812	0.563	0.692	0.692
Shan	0.788	0.647	0.664	0.160	0.395	0.291
Proposed	0.921	0.887	0.882	0.720	0.742	0.740

Table 2.2: *Objective comparisons of three algorithms in terms of SSIM (Part I)*

effects in the boundary. The Matlab command "edgetaper" is used to avoid the ringing effects in the boundary. It will smooth the boundary by calculating weights of neighboring patches before the Fourier transformation.

2.5 Experiments

In the proposed non-blind deconvolution algorithm, the image prior is represented by the GSM model, which is extracted from the related clear image. Then an energy minimization problem is established and solved with an iterative approach. This iterative restoration process will recover the latent image from coarse to fine until this optimization problem converges. Although the iterative approach is similar as the one in Shan's blind deconvolution method, a different image prior and energy minimization problem are used in my algorithm.

2.5.1 Different Algorithms Comparisons

I test the proposed algorithm on twelve stereo images. The blurry image is generated from ground truth right view image using four commonly used blur kernel respectively. The Gaussian white noise is added with different variance such as $\sigma = 0.01,\ 0.05,\ 0.1$. The conventional RL non-blind deconvolution method is tested with 20 iterations, which also serves as the image restoration in Fergus's blind deconvolution method. Moreover, the non-blind deconvolution part from Shan's blind deconvolution algorithm is also tested using their released executable file. The results are obtained using the best parameters as I can. In my algorithm, the initial λ_1 is defined by users such as $[0.01, 0.2]$ and all the other parameters use the default values.

Both objective and subjective measurements are evaluated. Since the ground-truth im-

Image	Image 7	Image 8	Image 9	Image 10	Image 11	Image 12
Blurry	0.476	0.688	0.615	0.610	0.556	0.780
RL	0.851	0.911	0.805	0.881	0.752	0.903
Shan	0.627	0.822	0.532	0.700	0.517	0.823
Proposed	0.913	0.942	0.862	0.922	0.749	0.918

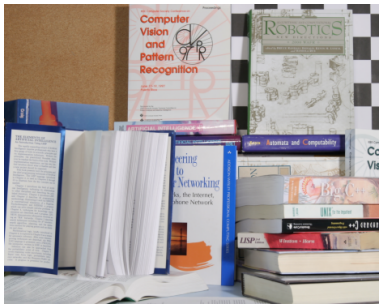
Table 2.3: *Objective comparisons of three algorithms in terms of SSIM (Part II)*

ages are available, errors between original and restored images are represented by MSE or SSIM [56]. Table (2.2, 2.3) list the objective comparisons of three algorithms in term of SSIM for 12 test images. The proposed result is the best of three for all the images except the Image 11. However, the visual quality of my result is the best for all the test images even for Image 11. As seen in Figure 2.19, the proposed result has the most pleasant restoration quality especially for characters and ropes.

In Figure (2.9, 2.10, 2.11), the blurry image is generated from the same kernel (size 27×19) with variance 0.01, 0.05 and 0.1 respectively. Compared with RL method, the proposed result is much better since ringing effects are largely suppressed and more details are restored. Compared with Shan's method, the proposed result still achieves better restoration quality even though their method works very well. For example, the enlarged introduction of the book in Figure 2.9 (g-i) clearly shows that the proposed result consists of more details and less ringing effects. The characters on the egg box are recovered best in my algorithm as well. The enlarged part is shown in Figure 2.10 (g-i). In Figure 2.11, my result has the best restoration result among three methods, especially for the doodle on the wall. In Figure (2.12, 2.13, 2.14), the blurry image is generated from the same kernel (size 27×27) with variance 0.01, 0.01 and 0.05 respectively. My algorithm exceeds RL and Shan's method when removing blur from texture regions. As we see, the dots, cloth texture and wood grain are recovered with more details in my algorithm.

In Figure (2.15, 2.16, 2.17), the blurry image is generated from the same kernel (size 25×25) with variance 0.01, 0.01 and 0.05 respectively. Although my result in Figure 2.15 has a little more ringing effects than Shan's result, more details are recovered such as on the statue and cone. The proposed algorithm is much better than the other two methods in Figure 2.16 and 2.17.

(a) *Blurry image*



(b) *Related clear image*

(c) *Blur kernel*

(d) *RL result*

(e) *Shan's result*

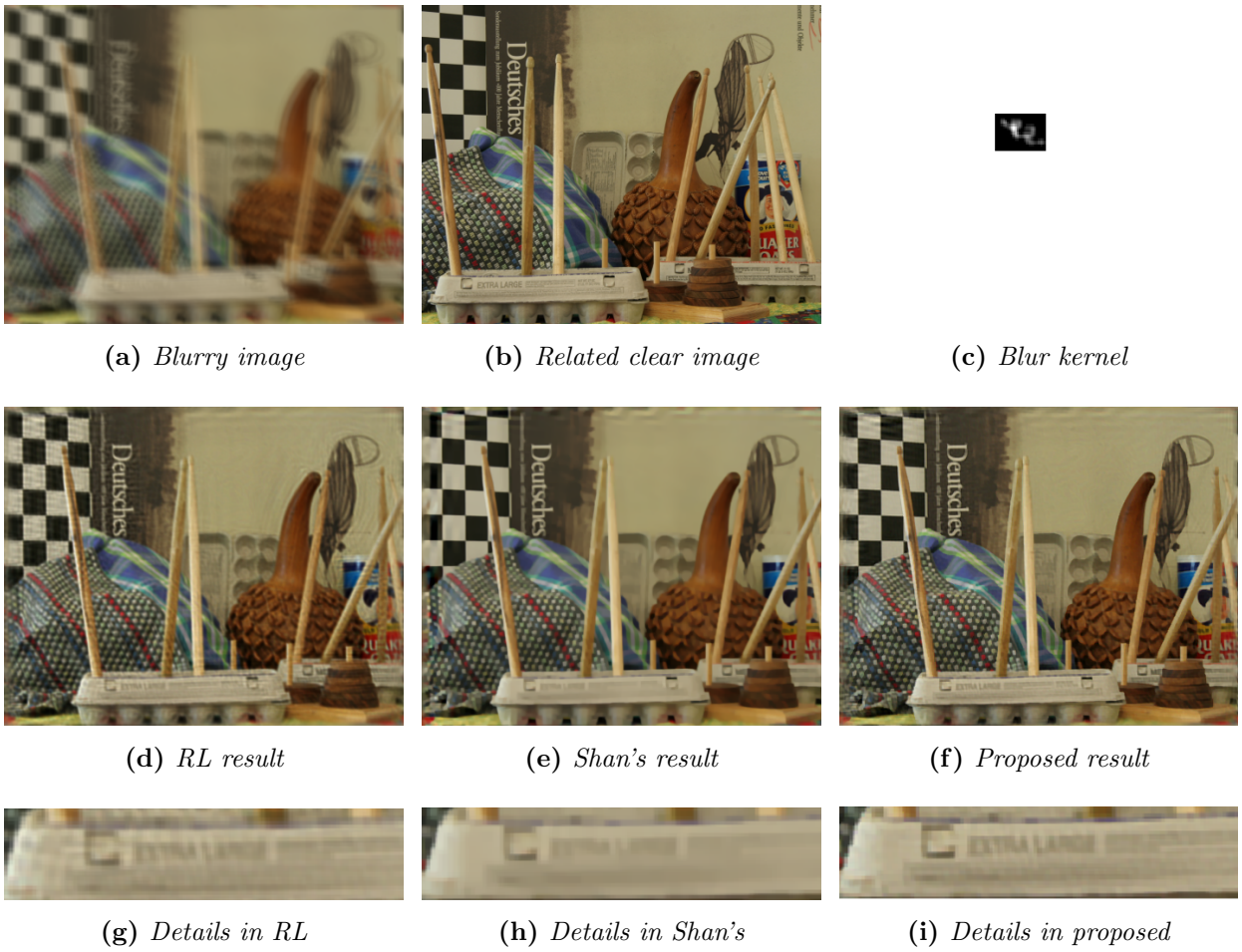
(f) *Proposed result*

(g) *Details in RL*

(h) *Details in Shan*

(i) *Details in proposed*

Figure 2.9: *Comparisons of non-blind deconvolution for Image 1 (a) Blurry image. (b) Related clear image (c) Blur kernel. (d)-(f) Recovered images by RL, Shan's and proposed algorithm. (g)-(i) Enlarged details in the three methods respectively*



(a) *Blurry image* **(b)** *Related clear image* **(c)** *Blur kernel*

(d) *RL result* **(e)** *Shan's result* **(f)** *Proposed result*

(g) *Details in RL* **(h)** *Details in Shan's* **(i)** *Details in proposed*

Figure 2.10: *Comparisons of non-blind deconvolution for Image 2 (a) Blurry image. (b) Related clear image (c) Blur kernel. (d)-(f) Recovered images by RL, Shan's and the proposed algorithm. (g)-(i) Enlarged details in the three methods respectively*

The proposed algorithm achieves a very good restoration quality on the above test images. However, those blur kernels are rather small when compared to the image size. What about the deblurring quality with serious blur? In Figure (2.18, 2.19, 2.20), the blurry image is generated from the same kernel (size 99×99) with variance 0.01, 0.01 and 0.05 respectively. RL's restoration is the worst with serious ringing effects. In Figure 2.18, details such as keyboard in Shan's method are a little better than ours, but some part such as multimeter is recovered with saturated values and it looks like distorted in Shan's result. In Figure 2.19, Shan's result is too smooth to lose details and RL's method has serious ringing and ghost effects. The proposed result is the best among three methods for these two images.

(a) *Blurry image*

(b) *Related clear image*

(c) *Blur kernel*

(d) *RL result*

(e) *Shan's result*

(f) *Proposed result*

Figure 2.11: *Comparisons of non-blind deconvolution for Image 3*

(a) *Blurry image*

(b) *Related clear image*

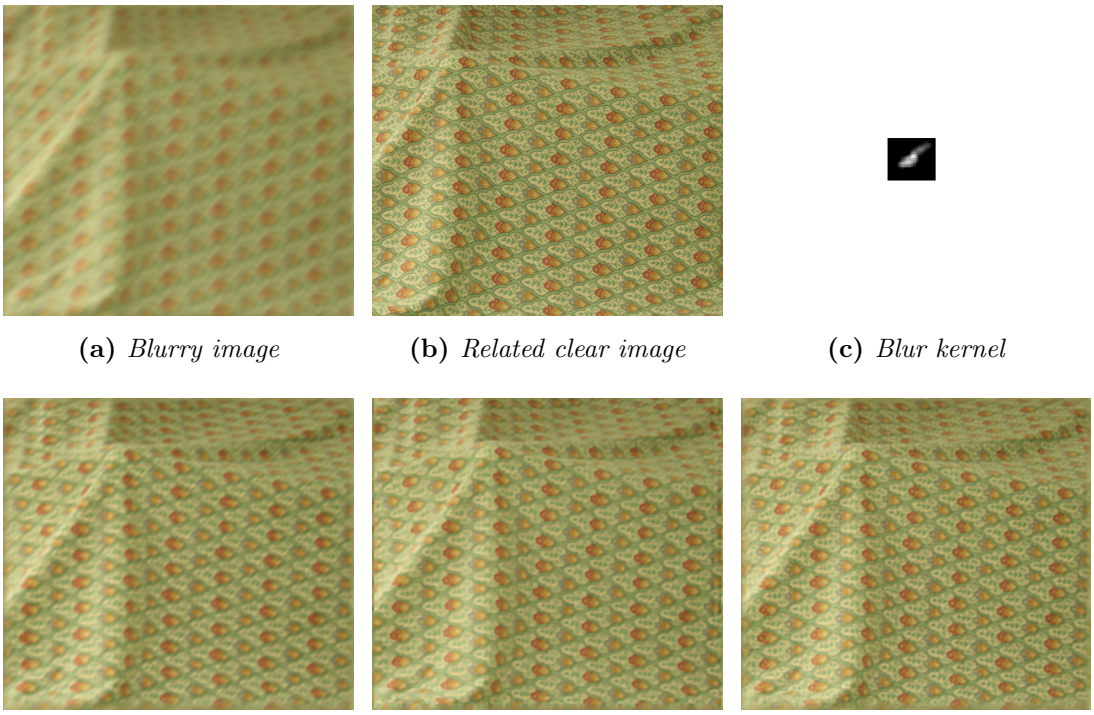
(c) *Blur kernel*

(d) *RL result*

(e) *Shan's result*

(f) *Proposed result*

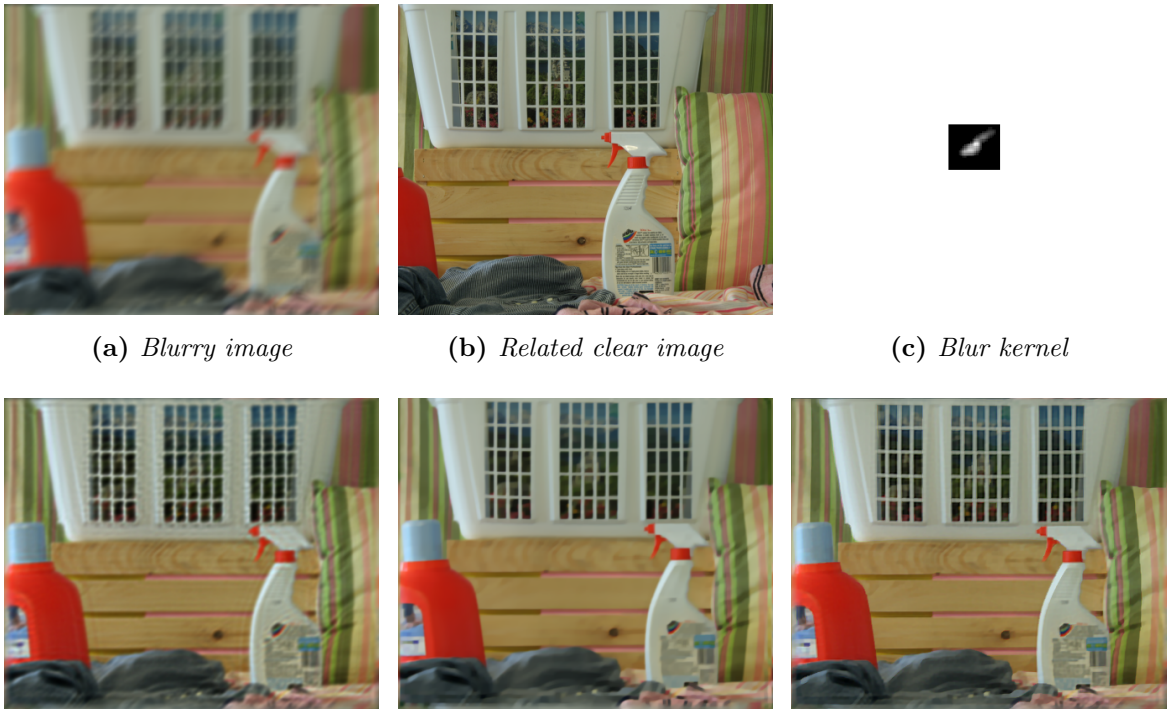
Figure 2.12: *Comparisons of non-blind deconvolution for Image 4*



(a) *Blurry image* **(b)** *Related clear image* **(c)** *Blur kernel*

(d) *RL result* **(e)** *Shan's result* **(f)** *Proposed result*

Figure 2.13: *Comparisons of non-blind deconvolution for Image 5*



(a) *Blurry image* **(b)** *Related clear image* **(c)** *Blur kernel*

(d) *RL result* **(e)** *Shan's result* **(f)** *Proposed result*

Figure 2.14: *Comparisons of non-blind deconvolution for Image 6*

(a) *Blurry image*

(b) *Related clear image*

(c) *Blur kernel*

(d) *RL result*

(e) *Shan's result*

(f) *Proposed result*

Figure 2.15: *Comparisons of non-blind deconvolution for Image 7*

(a) *Blurry image*

(b) *Related clear image*

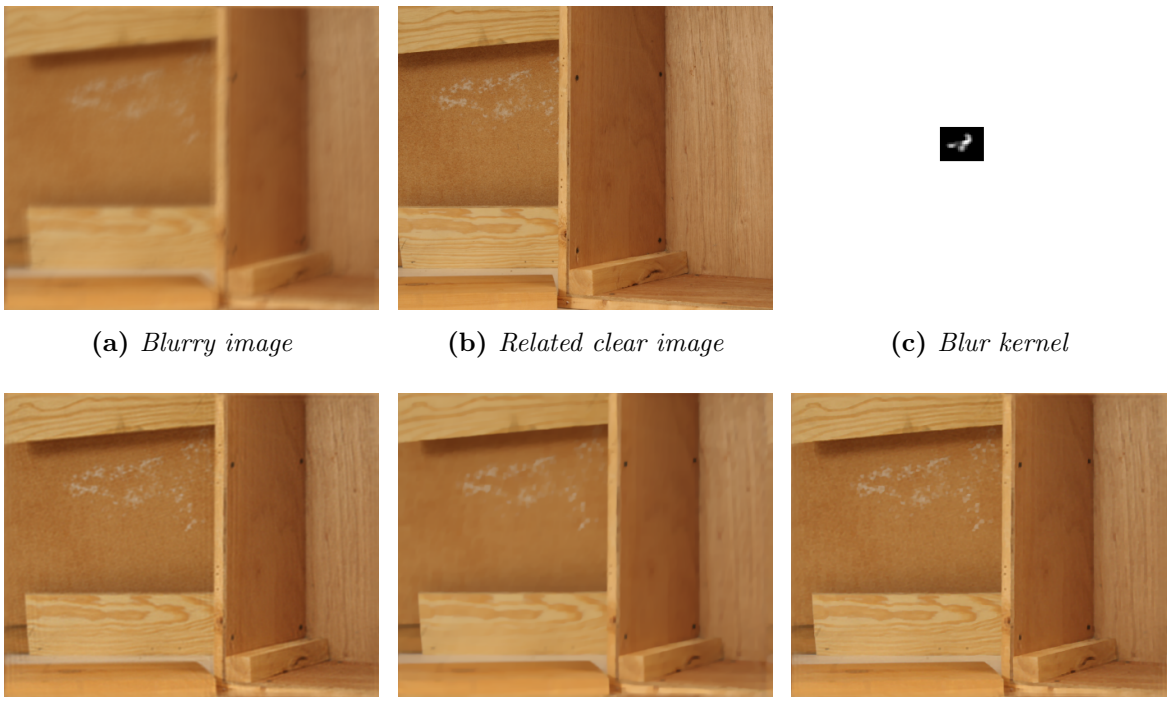
(c) *Blur kernel*

(d) *RL result*

(e) *Shan's result*

(f) *Proposed result*

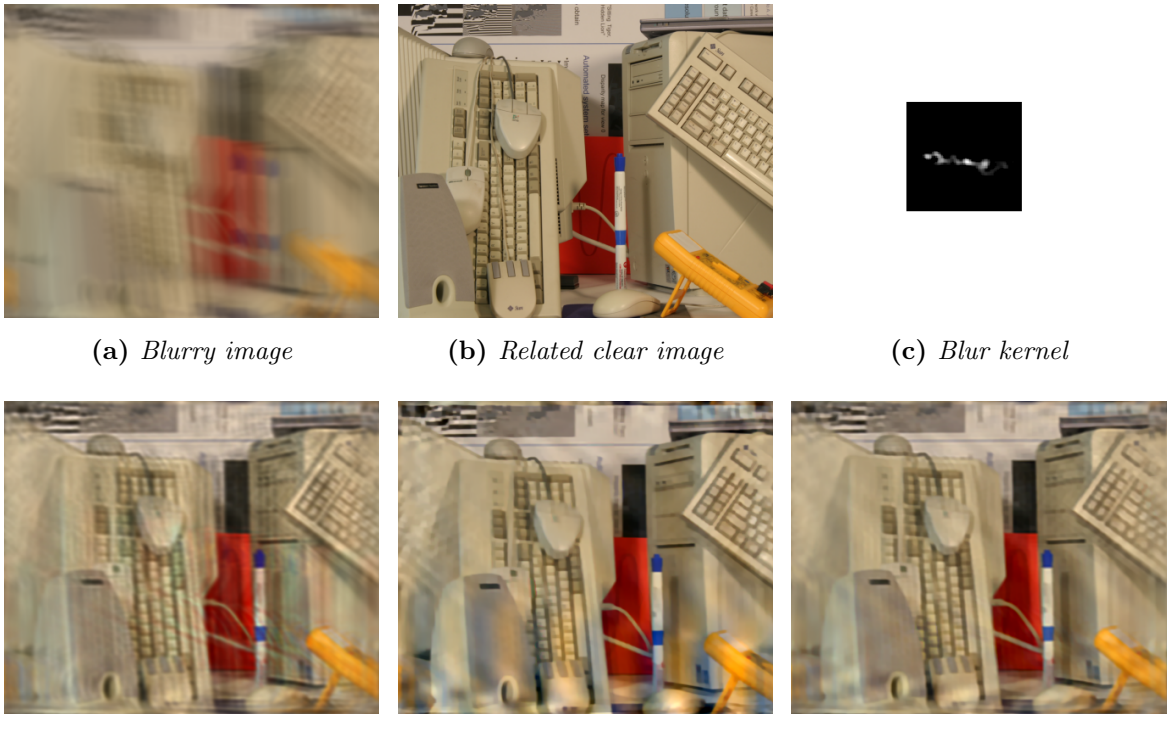
Figure 2.16: *Comparisons of non-blind deconvolution for Image 8*



(a) *Blurry image* **(b)** *Related clear image* **(c)** *Blur kernel*

(d) *RL result* **(e)** *Shan's result* **(f)** *Proposed result*

Figure 2.17: *Comparisons of non-blind deconvolution for Image 9*



(a) *Blurry image* **(b)** *Related clear image* **(c)** *Blur kernel*

(d) *RL result* **(e)** *Shan's result* **(f)** *Proposed result*

Figure 2.18: *Comparisons of non-blind deconvolution for Image 10*

(a) *Blurry image*

(b) *Related clear image*

(c) *Blur kernel*

(d) *RL result*

(e) *Shan's result*

(f) *Proposed result*

Figure 2.19: *Comparisons of non-blind deconvolution for Image 11*

(a) *Blurry image*

(b) *Related clear image*

(c) *Blur kernel*

(d) *RL result*

(e) *Shan's result*

(f) *Proposed result*

Figure 2.20: *Comparisons of non-blind deconvolution for Image 12*

2.5.2 Complexity Analysis

In the proposed algorithm, only λ_1 is defined by users from $[0.005, 0.02]$ and the default value is 0.01. The other parameters are assigned default values. The maximum iteration number is 16 when updating the Ψ in step 1. The overall iteration number of step 1 and step 2 are determined by the convergent speed of estimated latent image L. It usually doesn't exceed 30 times.

The proposed algorithm and RL method are implemented in Matlab 2011a Version. Shan's method is executed from their released exe file in C++ language. The results are compared on Intel 2.2GHz duo CPU. For example, it took 130s and 96s to deblur the Image 1 (size463×370) in RL and proposed algorithm. Since Shan's executed file is an optimized code in C++, it's fast and took 32s to deblur the same size image. However, by comparing the actual operations of algorithms, the proposed algorithm is more efficient than Shan's method. The two algorithms adopt the similar likelihood representation and iterative approach, but my minimization problem contains two terms while Shan's contains three terms. Moreover, a binary mask is needed and updated in every iteration in Shan's method. Therefore, the proposed algorithm is an efficient method and the computational time is acceptable.

2.6 Summary

In this chapter, a data-driven non-blind deconvolution algorithm is proposed. A MAP problem is established to estimate the latent image. In this inverse problem, the likelihood with mixed-order Gaussian noise model is used to describe the spatial randomness of noises. In addition, the GSM model is used as an accurate image prior of the blurry image. The GSM model is estimated from the related clear images. By separating the convolution operations with other terms, the energy minimization problem is solved in an efficient iterative approach. More details are recovered in the proposed algorithm from experimental results. Moreover, compared to Richardson-Lucy and Shan's algorithm, ringing effects are suppressed better in proposed algorithm. In conclusion, the proposed algorithm is an efficient non-blind deconvolution method with high restoration quality.

Chapter 3

Augmented Reality

Augmented reality (AR) is a new and promising application of video processing and computer vision techniques [57, 58]. The overall revenue generated by AR on smartphones is predicted to exceed $1.5 billion by 2015 in [59]. Some applications on innovative cameras and compasses are beginning to prove that AR is coming into reality and changing the way we view the world. For example, Google unveiled the "Project Glass" in April 2012, which applies the AR technology to take all the functionality of a smartphone. The augmented content will appear on a wearable device according to environment and demands.

There are many AR apps on the smartphones in recent years. For example, an AR app for iPhone uses Apple GPS compass to tell you where you parked your car and also provides an alert of due time. Wikipedia has an AR app which lets you view the wikipedia entries on an object or landmark simply by pointing your phone at it. When you walked by a famous building and was interested in the history of the building, you could point your phone at it and then found out the answer promptly without an Internet connection.

AR blends the virtual information with the real scene and hence improves the visual experiences of consumers. In this chapter, a generic virtual content insertion (VCI) system is introduced as an example of AR. It will analyze and insert an image or a movie on specific regions of broadcast sports or street view video streams. The challenging issues of VCI are how to less intrusively insert the contextually relevant content (what) at the right place (where) and the right time (when) with the attractive representation (how) in videos. In the proposed VCI system for sport and street view videos, (what) problem is not addressed. An

Figure 3.1: *VCI examples in football broadcasting*

image or a movie is randomly chosen and inserted as the appropriate virtual content.

3.1 Challenges of Virtual Content Insertion

Implementing the virtual content insertion in video streams is not an easy task in terms of the following aspects:

- Although inserting virtual content in sport videos is quite mature and already applied in soccer and football broadcasting, insertion in the compressed video streams from the Internet or digital camcorders is rather difficult. For example, there are usually inserted lines or images to denote the starting positions in the football broadcastings which are shown in Figure 3.1. In this case, the insertion usually lasts for several frames and involves user interactions. Moreover, the cameras are setup by the broadcasting companies and then the camera calibration is not a problem. By contrast, video streams are from the Internet or camcorders in the proposed system. Therefore, we don't know anything about the camera positions and have to estimate the camera parameters.

- Inserting virtual content in street view videos is more challenging than in sport videos. In sport videos, the standard court model is served as the image prior to estimate the inserted positions. By contrast, in street view videos, there is no such prior knowledge of buildings. In addition, detecting the building facade is also a difficult task. There was an interactive system that allowed inserting a virtual picture in the architecture

or extracting a poster from a certain building facade. This tool required a few user interactions such as selecting two groups of parallel lines for vanishing points identification and camera calibration. On the contrary, the proposed system doesn't involve user interactions and it will automatically analyze and implement the insertion on the building facades.

- Insertion a virtual content on a still image is not that difficult, but it's more challenging to do it in a video stream because video contains both spatial and temporal information. Therefore, it not only needs to calibrate the camera for one frame, but also needs to track the motions of camera in a video sequence. Then the inserted image seems to be pasted as it's real and should move with the camera.

- There are several other factors affecting the performance of this system. For instance, the inserted image should not be pasted on the players. At the same time, it should not be inserted in such frames which are the close-up of players and audiences or even advertisements. How to reduce the warping noise when converting the processed result to the current view also affects this system. Jittering effects caused by the estimation errors should be reduced as much as possible.

In this chapter, I propose a generic VCI system that satisfies all the requirements. A novel detection and tracking strategy is proposed to process the insertion task with a high efficiency.

In a sport game (soccer, tennis, baseball, volleyball etc.), a playfield constrains the players action region and also makes it easier to find a good place for virtual content insertion, so playfield modeling is applied to extract the court area. The standard court model is used as the image prior to detect the specific region as the right place, like soccer center circle and goalmouth bar, tennis and volleyball court etc. In a street view video, the building facade looks appropriate to post an image. The modern building shows structured visual elements, like parallel straight lines with repeated window patterns. Therefore, it makes sense to determine the orientation of the architecture by estimating vanishing points. More importantly, the 3D geometry model derived from vanishing points extraction is served as the image prior to estimate inserted positions. Then a valid rectangular is determined. If it

also lasts for several frames, virtual content are inserted in these frames passing through a Wiener filter to suppress jittering effects.

It is important to accurately align the virtual content with the real scene by registration. The visual tracking method is either feature-based or region-based, extensively discussed in the computer vision field. Sometimes GPS or information from other sensors (inertial for the camera) is used together to make tracking much more robust. The failure in tracking may cause jittering and drifting which cause bad viewing impression for reviewers. The virtual-real blending should take into account the differences of contrast, color and resolution between inserted image and current frame. Therefore, color harmonization is adopted to make inserted content coordinated with the background.

3.2 Framework of VCI System

Figure 3.2 shows the generic diagram of the proposed VCI system in sports videos or in street building videos. The task is finding an appropriate region for insertion, such as the goalmouth bar in soccer videos and play court in tennis videos. It applies a detection and tracking combined strategy. In the first frame or frame when specific region is not tracked, VCI system tries to detect the specific regions. Inserted regions are estimated in a Bayesian framework. Different targets are detected in sports and street view video streams, because different image priors are used in these two scenes. For example, in sports videos, lines composing the tennis court are detected. Then the standard tennis court is served as the image prior. In street view videos, a 3D geometry model of building facade is established through vanishing points estimation and is used as the image prior to estimate the inserted region.

Once the estimated inserted region is verified through the model fitting at a certain frame, camera calibration is processed to describe the homography transformation between the 3D planar and front-view planar. In the following frames, object tracking is executed and new camera parameters are obtained by model refining, such as the playfield model verification in sports videos, or the corner and dominant direction verification in street view videos. In order to suppress jittering, motion stabilization is processed before the virtual content is blended

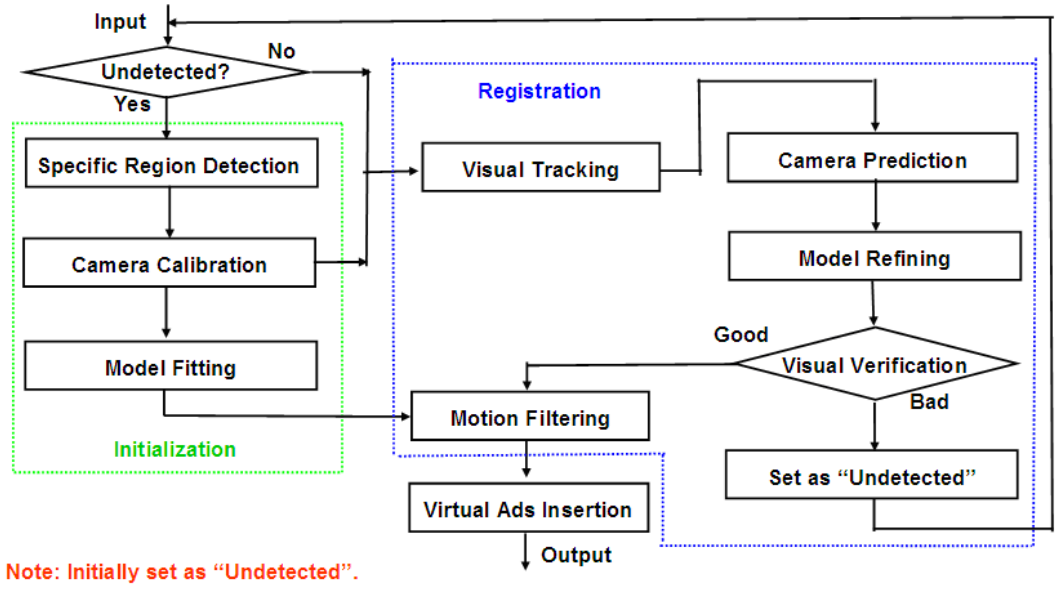


Figure 3.2: *Diagram of generic virtual content insertion system.*

with the background. Then in the last step, the inserted content is warped into the specific regions by the estimated homography transformation. In addition, color harmonization is applied to reduce disturbance to the viewers.

3.3 Implementation for Sports Video Streams

In this section, a VCI system is proposed for sports scenes such as broadcast soccer and tennis matches. Soccer goalmouth is typically assumed to be composed by two vertical and two horizontal white lines. By contrast, tennis court is more complex. It is regarded as a planar and composed by five horizontal and five vertical white lines. Although some intersections of lines do not exist in the real world, I still use these virtual intersection points to estimate the homography transformation. Figure 3.3 illustrates the corresponding lines of the real scene and court models from [1]. There are two virtual intersection points in this example.

In Figure 3.4, a detailed description of VCI system for sports videos is illustrated. Line detection is an important part and it starts from identifying the white pixels because the actual lines are usually white in the real world. However, sometimes white pixels also appear

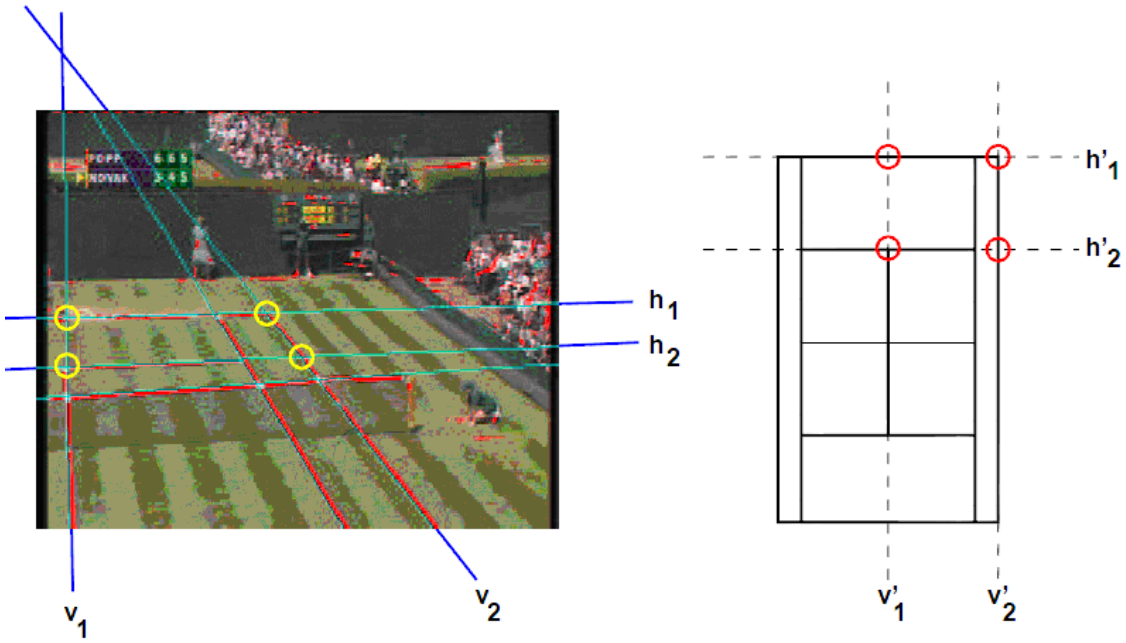


Figure 3.3: *Corresponding lines between the real scene and court model in a tennis video (images courtesy of Farin et al. in [1])*

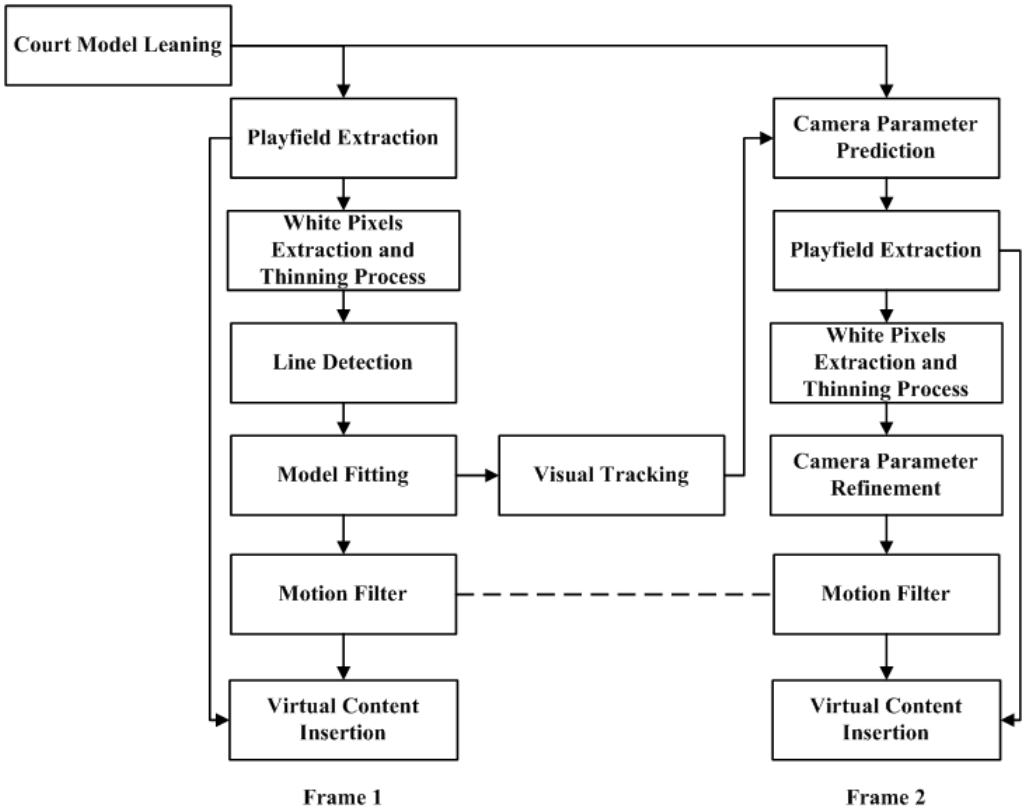


Figure 3.4: *Detailed diagram of VCI system for sports videos*

on other objects such as player uniforms or advertisements. In order to detect lines correctly, only white pixels within the playfield are kept. Therefore, the first step of this system is to extract the playfield through pre-learned Gaussian models. White pixels are then extracted within the playfield and lines are obtained by Hough line transform.

The inserted positions are estimated in a Bayesian framework. The image prior is the standard tennis court model. Among all the horizontal and vertical detected lines, the combination which achieves the minimum error through homogaphy transform with image prior is regarded as the best case to calibrate the camera in the model fitting process. A homography transform which maps the world coordinate system to the image coordinate system is determined by four point-correspondences of detected lines to the ones in the court model [60]. The virtual content is inserted into the specific region by perspective transformation. Now specific detection in the first frame is finished.

In the following frames, the plane with goalmouth or the court plane is tracked by optical flow method in [61] or keypoint tracking KLT method by Shi [62]. The homography matrix is updated from the tracking process. On one hand, the playfield and white pixels will be detected more accurately based on the estimated homography matrix. On the other hand, homography matrix is refined by fitting the lines with goalmouth or tennis court model in a Bayesian framework. The virtual content is inserted with the new estimated camera parameters.

In a broadcast sports video, there are always some frames with the players close-up or shots with audiences or even advertisements. These frames should be ignored in order to avoid inserting content in false positions. If a playfield is not detected or the detected lines don't fit the goalmouth or court model in a certain frame, this frame is not processed. In order to let the insertion last for several frames, a buffer is set to store some continuous frames and utilize Wiener filter to remove high frequency noise and reduce jitters, which will be discussed in section 3.4.3.

(a) *soccer* **(b)** *tennis*

Figure 3.5: *Examples of playfield models learning*

3.3.1 Playfield Extraction

The mean and variance in the Gaussian models of playfield RGB values are learned in advance by manually choosing the playfield region frame by frame in the training videos. Figure 3.5 is an example of RGB model learning in the playfield of a soccer and tennis video. The manually chosen regions for learning purpose are denoted by red rectangles. There are four typical tennis courts from four grand slams, namely, US open, French open, Australian open and Wimbledon. In US open and Australian open, the colors of inner and outer of the court are different, therefore in these two cases, there are two Gaussian models for each part.

Assume the R, G, B values of pixel (x, y) in the Image $I(x, y)$ is $L_{(x,y)} = \{R_{(x,y)}, G_{(x,y)}, B_{(x,y)}\}$, The mean and variance of pixels in the selected region S are calculated as:

$$\mu = \frac{1}{N} \sum_{(x,y)\in S} L_{(x,y)}, \sigma^2 = \frac{1}{N} \sum_{(x,y)\in S} (L_{(x,y)} - \mu)^2, \tag{3.1}$$

By comparing the RGB values of each pixel with the RGB Gaussian models, a playfield mask is obtained. The following criteria classifies a pixel (x, y) with the RGB value $L_{(x,y)}$ into 0 or 1. t is scaling factor in the range $[1.0, 3.0]$:

$$G_{(x,y)} = \begin{cases} 1 & \text{if } |L_{(x,y)} - \mu| < t\sigma \\ 0 & \text{if otherwise} \end{cases} \tag{3.2}$$

A typical RGB model is listed in Table 3.1 for soccer videos and Table 3.2 for tennis videos. Figure 3.6 shows the 1^{st} frame and its extracted playfield in a soccer and tennis video. The playfield remains the same but the non-playfield region is represented in black .

	R	G	B
Mean	123.67	142.366	89.916
Variance	47.285	26.110	39.976

Table 3.1: *Gaussian models of RGB values for soccer videos*

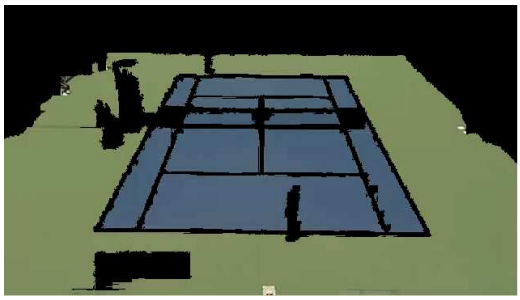
Court Type	R_Mean	G_Mean	B_Mean	R_Variance	G_Variance	B_Variance
French	170.73	97.82	70.21	36.85	34.99	29.31
US inner	76.62	98.71	123.11	16.25	16.25	14.33
US outer	125.61	141.82	93.01	14.25	16.80	15.77
Aus inner	88.26	151.41	172.30	24.08	12.44	14.52
Aus outer	60.82	149.02	209.57	29.35	25.85	28.87
wimbledon	126.68	134.09	115.72	43.56	26.25	45.14

Table 3.2: *Gaussian models of RGB values for tennis videos*

(a) *original frame*

(b) *extracted playfield*

(c) *original frame*



(d) *extracted playfield*

Figure 3.6: *Playfield extraction in the* 1^{st} *frame of a test soccer and tennis video*

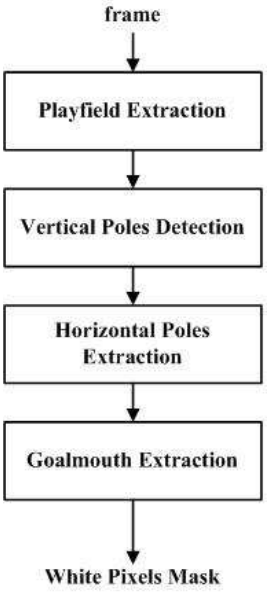
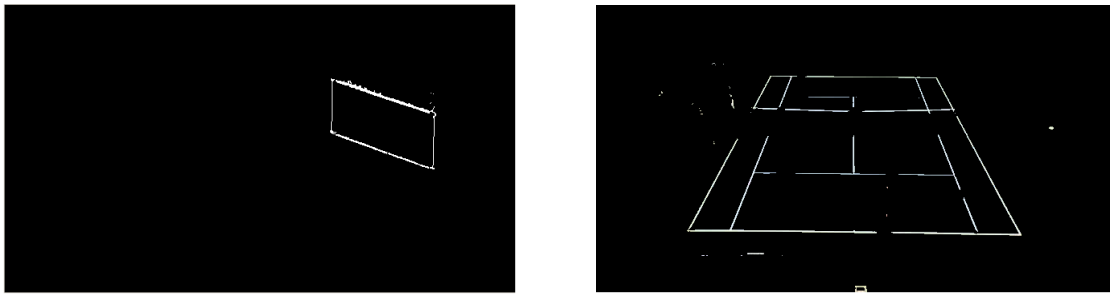


Figure 3.7: *Flowchart of white pixels extraction in the goalmouth scenario*

3.3.2 Line Detection

Line detection determines the accuracy of VCI systems. It is executed as follows. Firstly, white pixels are obtained within the playfield by setting a RGB threshold. For instance, the threshold is $(200, 200, 200)$ in soccer videos and $(140, 140, 140)$ in tennis videos. Secondly, these white pixels will be thinned to reduce the errors in hough line detection. Finally, lines are obtained from Hough transform.

I propose a novel method to obtain the white pixels in the soccer scene, as shown in Figure 3.7. Firstly, detect the vertical poles in the playfield. Then detect the horizontal bar between vertical poles in the non-playfield region. Since horizontal lines should have similar directions, parallel lines in the playfield are detected. They are also intersected with two vertical poles. Finally two binary masks are obtained for goalmouth and playground. Although virtual content is only inserted above the goalmouth bar in the demo, it is also possible to insert the image in the ground since a binary image of the penalty area is also obtained. Then the lines are detected and corresponded to the real penalty model. The binary image which illustrates the white pixels extraction from a soccer and tennis video is shown in Fig 3.8.



(a) *soccer goalmouth* **(b)** *tennis court*

Figure 3.8: *White pixels extraction in the* 1^{st} *frame of a soccer and tennis video*

Lines are detected by Hough transform in these binary images. Not only the desired lines but also some other close-by lines are detected by applying the Hough line transform directly. Therefore, initial results are refined by non-maximal suppression. Denote a line with normal $\vec{n} = (n_x, n_y)^T$ ($||\vec{n}|| = 1$) and distance d to the origin. Then a points set L containing all the white pixels (denoted as $l(x, y) = 1$) near this line is defined as follows:

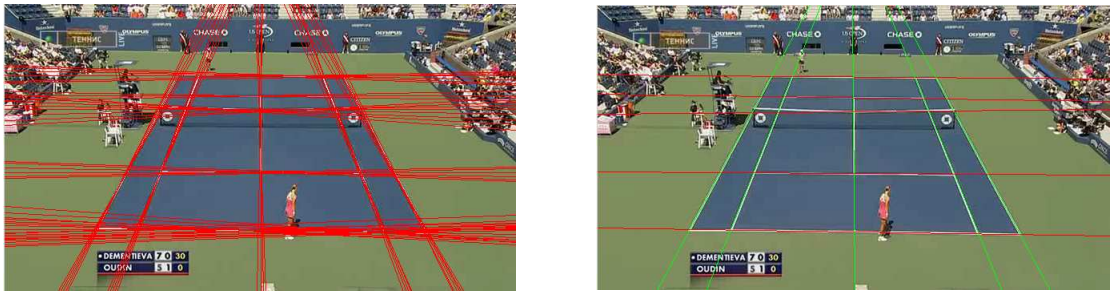
$$L = \{p = (x, y)^T | l(x, y) = 1, |(n_x, n_y, -d) \bullet p| < \sigma_r\} \tag{3.3}$$

By solving the LMS problem of Equation (3.4), a robust line parameters (n_x, n_y, d) are obtained.

$$\begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{pmatrix} \begin{pmatrix} m_x \\ m_y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

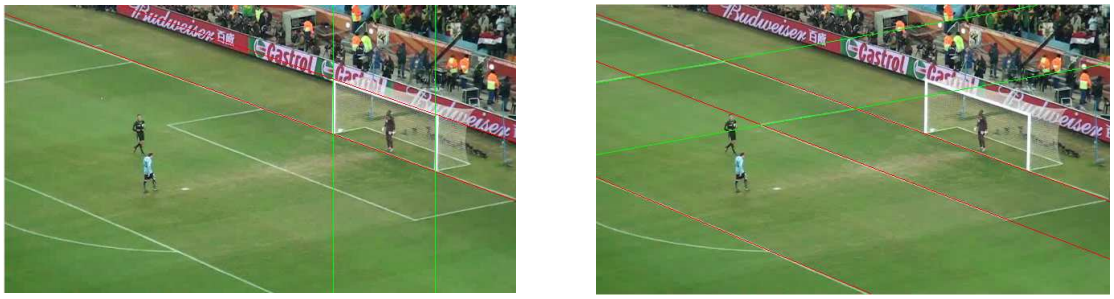
$$d = \frac{1}{\sqrt{m_x^2 + m_y^2}}, n_x = m_x d, n_y = m_y d, \tag{3.4}$$

All the candidate lines obtained through the above steps are classified into the horizontal line set if $|arctan(n_y/n_x)| < 25^\circ$ and the vertical line set otherwise. In the front-view of goalmouth or tennis court, a valid plane is composed by two horizontal and two vertical lines. Therefore, any two lines from the horizontal and vertical line set will map to the corresponding lines in the goalmouth or tennis court model. Figure 3.9 illustrates the initial and refined line detection results from the 1^{st} frame of a tennis video. Meanwhile, Figure



(a) *Initial lines* **(b)** *refined lines*

Figure 3.9: *Examples of initial and refined line detection on a tennis court*



(a) *goalmouth* **(b)** *penalty region*

Figure 3.10: *Final line detection in the goalmouth and penalty region*

3.10 shows the final line detection on the goalmouth and also on the penalty region from the 1^{st} frame of a test soccer video. The red lines stand for the horizontal lines and green lines are for the vertical lines.

3.3.3 Camera Calibration

The task of VCI system is looking for an appropriate region for virtual content insertion. It is estimated in a Bayesian framework with the standard court model as the image prior. For a combination of detected lines, the goal is to minimize the errors between the transformations of lines and standard court model. This homography transformation is defined as the mapping from a planar of the real world to the image and denoted as H. It is a eight-parameter perspective transformation, mapping a position $p^{'}$ in the real coordinate system

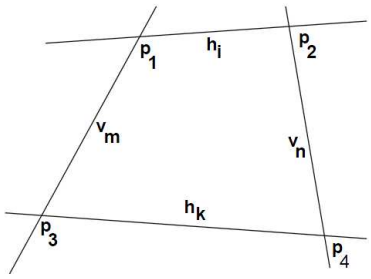


Figure 3.11: *Intersections of two horizontal and two vertical lines*

to image coordinate p. These positions are represented in the homogeneous coordinates as $(x, y, w)^T$, and the transformation $p = Hp^{'}$ is rewritten as Equation (3.5).

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x^{'} \\ y^{'} \\ w^{'} \end{pmatrix} \tag{3.5}$$

Homogeneous coordinates are scaling invariant and then the degrees of freedom in H are reduced from nine to eight. A four-corners correspondence between the current and front view is enough to determine the eight parameters. Assume the two horizontal lines are h_i, h_j and two vertical lines are v_m, v_n. The four intersections p_1, p_2, p_3, p_4 shown in Figure 3.11 are represented as the following:

$$p_1 = h_i \times v_m, \quad p_2 = h_i \times v_n, \quad p_3 = h_j \times v_m, \quad p_4 = h_j \times v_n, \tag{3.6}$$

Therefore, if the four intersection points from a line combination in the current view could find their corresponding points in the front view of court model, a homography matrix describing the transformation between two views is obtained. In order to suppress estimation errors, the popular random sample consensus ($RANSAC$) method in [63] is applied to achieve a good estimation of this homography matrix.

3.3.4 Model Fitting

Model fitting is an important part to ensure the accuracy of camera calibration. In the goalmouth scenario, only one case exists if two horizontal and two vertical lines are detected.

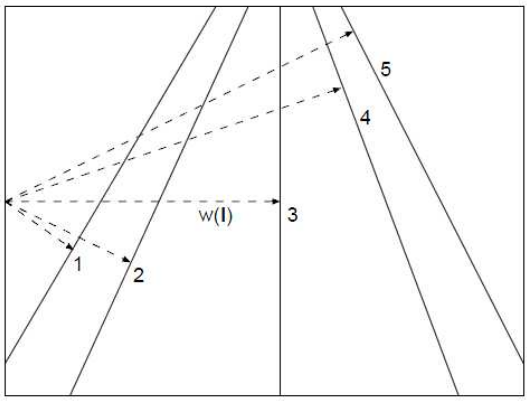
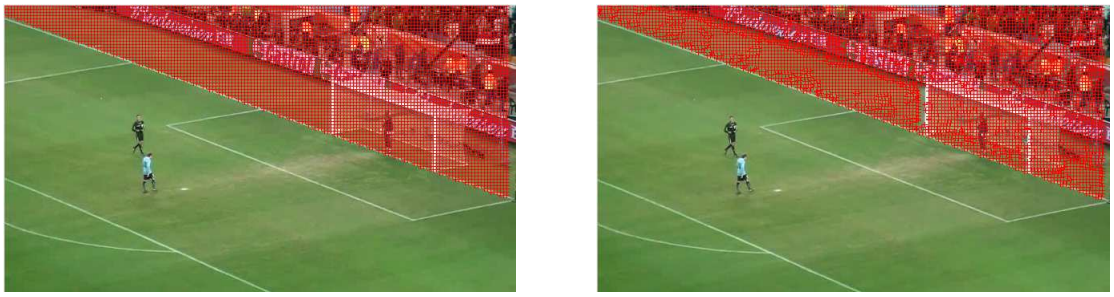


Figure 3.12: *Vertical lines ordering(images courtesy of Farin et al. in [1])*

However, the tennis scenario is more complex. If C_H horizontal lines and C_V vertical lines are detected, the number of possible line combinations will be $C_H C_V (C_H - 1)(C_V - 1)/4$. Randomly choosing two lines from each line set, this line combination will determine a homography transformation between this planar and the standard court. The calculation of the homography matrix H is similar to the algorithm described in 3.3.3. Among all the combinations, there exists a combination which fit the model court best. In order to enumerate all the combinations efficiently, the set of vertical lines are ordered from left to right and the set of horizontal lines are from top to bottom. They are sorted by comparing the distances from a point in the middle of the left border or the top border to each line. Figure 3.12 shows that the vertical lines are sorted by the distances of lines to the point in the middle of the left border.

In the evaluation process, all the line segments from the standard court in the front view are transformed to the current view with the guessed homography matrix H by $p = Hp^{'}$. Each model line $p_1^{'}p_2^{'}$ is transformed into the image coordinates $p_1 p_2$. Pixels are sampled at discrete positions along the line. Then the evaluation value is increased by one if the pixel is a white pixel on the candidate court lines, or decreased by 0.5 if it is not. Pixels outside the image are ignored. Eventually, each parameter set is rated by computing the score in Equation (3.7). The homography matrix with the highest score is used as the camera



(a) 5^{th} *frame* **(b)** 6^{th} *frame*

Figure 3.13: *Visual tracking between consecutive frames in a soccer video*

parameters for later insertion task.

$$Score = \sum_{p_1 p_2} \sum_{(x,y)} \begin{cases} 1 & \text{if } l(x,y) = 1 \\ -0.5 & \text{if } l(x,y) = 0 \\ 0 & \text{if others} \end{cases} \tag{3.7}$$

3.3.5 Visual Tracking

The fundamental part of VCI system is to estimate the inserted positions. In the first frame, it is achieved by line detection and model fitting in a Bayesian framework. The following frames could be processed in the same way, but it will be extremely time-consuming. Since the temporal redundancy is common in a video sequence, object tracking will locate the new positions of object if camera motion is small. Therefore, a detection and tracking combined approach is adopted.

In the proposed VCI system, specific regions such as tennis court and building facades are detected every frame until they are detected in a certain frame. Then in the following frames, camera parameters are updated by visual tracking without detection, because it captures the motion of camera in continuous several frames when motion is small. If the estimated camera parameters through visual tracking can't represent the actual homography transformation between 2D image and 3D planar, detection process is invoked again. Therefore, each frame is processed efficiently with the detection and tracking combined strategy.

Assuming the homography matrix in the frame where tennis court is detected as H, the homography matrix H_t which describes the camera motion in consecutive frames is

obtained by the optical flow method [61] or KLT tracking method [62]. Then the estimated homography matrix of the following frame is $\hat{H} = HH_t$. Therefore, the model fitting process will be much simpler and we only need to find the best matching score within a small amount of line combinations because the estimated homography constraints possible lines positions.

In the proposed VCI system in sport videos, the optical flow method is applied. The motion vectors of a pixel are calculated within its neighborhoods. Figure 3.13 illustrates the points chosen from the 5^{th} frame and their tracking results in the 6^{th} frame. Points which are in the same planar of goalmouth are chosen and denoted in red.

3.4 Implementation for Street View Video Streams

In this section, virtual content insertion in a street view video stream is described. The modern building facade is usually regarded as a planar and hence suitable for inserting purpose. However, due to the large variability of building orientations, it is more difficult to implement the VCI system in street view videos than in sports videos. Generally speaking, there is no such standard court model when inserting the virtual content on a building facade. Nevertheless, a 3D geometry model through vanishing points detection is established as the image prior in the Bayesian framework to estimate inserted positions on a building facade. It still adopts the detection and tracking combined strategy.

In the detection process, firstly, vanishing points are extracted and the detected lines are classified into groups corresponding to the different vanishing points. Secondly, similar to the sports scene, inserted positions are estimated in a Bayesian framework. Among all the detected lines, the best line combination should represent the mapping from the 3D planar to the 2D image. Although there are usually more lines in a building scene than in the tennis scenario, not every line combination is enumerated due to complexity. The task is to find the largest rectangle on the facade which passes both corners and dominant directions verifications. Then the virtual content is inserted within this rectangle.

In the tracking process, the KLT tracking method is applied to pursue the corner feature points in consecutive frames. Then the homography matrix of next frame is estimated by multiplying the tracking matrix and homography matrix in current frame. Moreover, in order

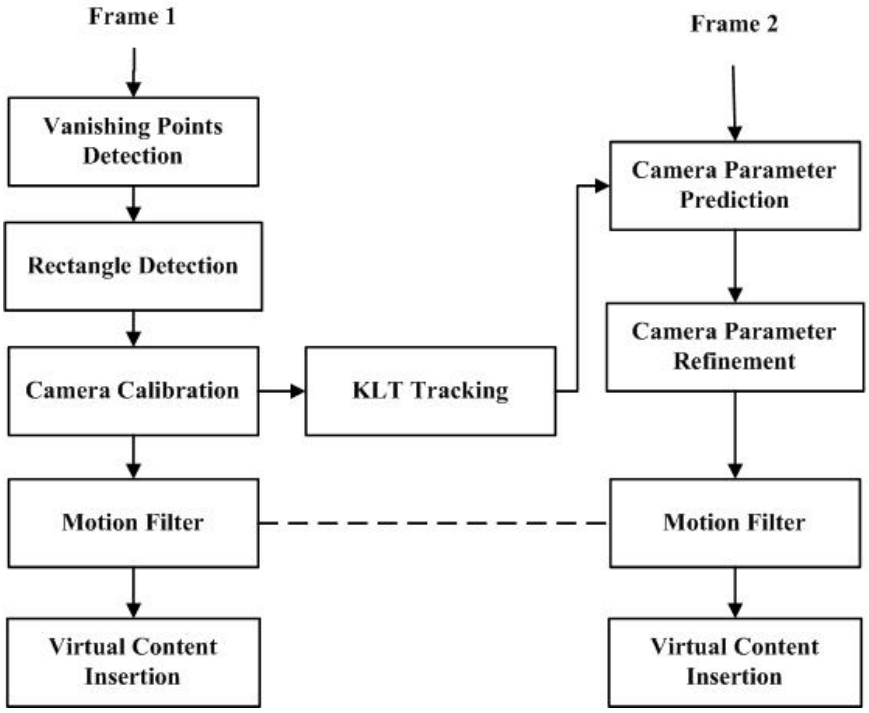


Figure 3.14: *Flowchart of VCI system in street view video streams*

to avoid jitters, a buffer is created to store the latest several (thirty, for instance) frames and a Wiener filter is applied to remove the high frequency noise in the motion trajectory. Color harmonization is also applied to blend the inserted image with the background. Figure 3.14 illustrates the diagram of VCI system in street view videos. The vanishing point detection, rectangular extraction, motion stabilization and color harmonization are described in details in this section. The algorithms such as camera calibration, visual tracking and virtual content insertion are found in section 3.3.

3.4.1 3D Geometry Model Establishment

In VCI system in street view videos, inserted positions are estimated in a Bayesian framework. The image prior is a 3D geometry model extracted from the current view through vanishing points detection, which describes the geometry properties of the building facades. A non-iterative approach by Tardif [2] was applied with a slight modification to detect the vanishing points. This method didn't need to represent edges on the Gaussian sphere; instead

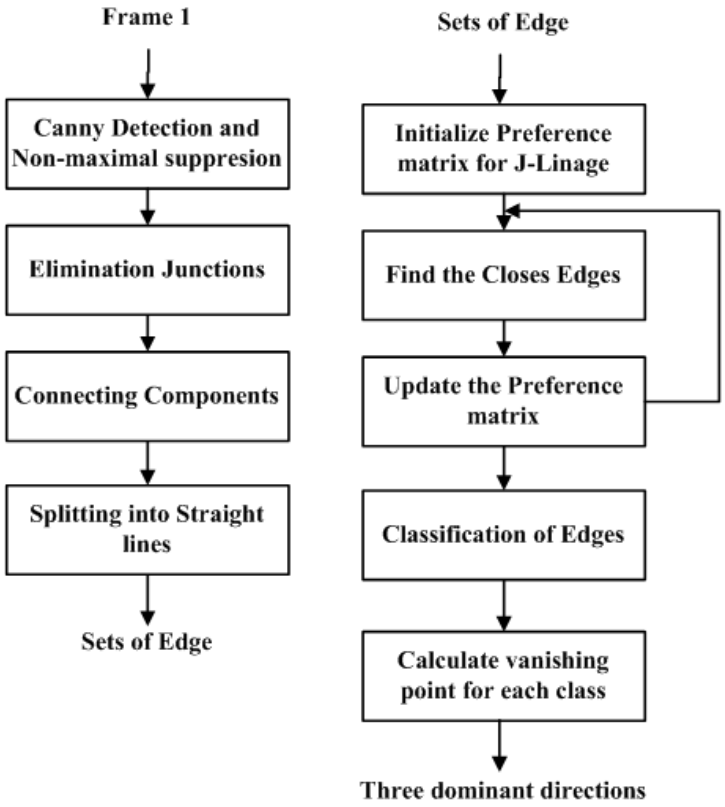


Figure 3.15: *Flowchart of vanishing points detection*

it directly labeled the edges. Figure 3.15 shows the procedures of this method.

In the frame where a building facade is the purpose of detection, this algorithm is executed as follows. It starts from obtaining the edges by the Canny detector. Then the non-maximal suppression is followed and a map of one pixel thick edges is obtained. Moreover, junctions are eliminated and connected components are linked using flood-fill. Afterwards, each component is divided into straight lines by browsing the list of coordinates. It will split when the standard deviation of a line is larger than a one pixel. The separated short segments which lie on the same line are merged, in order to reduce errors and also reduce computation complexity in later classification process. Figure 3.16 is an example of canny detection and the refined edge map is denoted in red.

The vanishing point estimation algorithm is based on the J-Linkage which requires a measurement of consistency between hypothesized vanishing points and edges that are geometrically meaningful. It treats all vanishing points equally and does not need to distinguish

(a) *Edges from Canny Detection*

(b) *Refined Edges*

Figure 3.16: *Canny and refined edge map of a building facade*

Entities	Definition
ε_n	Edge index n
e_n^1,e_n^2	The two end points of ε_n
$\bar{e}_n$	Centroid of the end points
l_n	Implicit line passing by ε_n
S_m	Subset of edges of ε
$\|S_m\|$	Size of the set S_m

Table 3.3: *Detonation of detected edges*

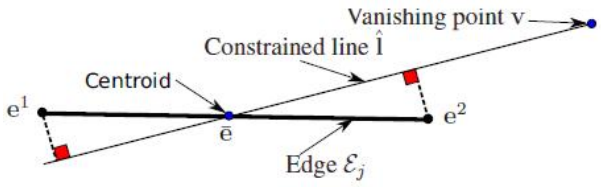


Figure 3.17: *Illustrations of a vanishing point and an edge (images courtesy of Tardif in [2])*

the finite and infinite vanishing points.

In this part, the notations to present the straight lines are listed in Table 3.3. Besides, a function denoted as $D(v, \varepsilon_j)$ provides a measurement of the consistency between a vanishing point v and an edge ε_j. The function is represented by Equation (3.8).

$$D(v, \varepsilon_j) = dist(e_j^1, \vec{l}) \tag{3.8}$$

where $\vec{l} = [\bar{e}_j]_\times v$ defines a line passing through the vanishing point and the centroid of the detected edge. The orthogonal distance from a point p to the line l (illustrated in Figure 3.17 from [2]) is defined as

$$dist(p, l) = \frac{|l^T p|}{\sqrt{l_1^2 + l_2^2}} \tag{3.9}$$

Assuming a set of N edges is obtained, and the consistency measurement is also defined as Equation (3.8). The next step is to estimate the vanishing points, and classify each edge such as assigning it to a certain vanishing point or marking as an outlier. The classification relies on the J-Linkage algorithm. A brief overview of the J-Linkage algorithm in the context

of vanishing point detection is given as follows. The initial parameters are the consensus threshold ϕ and the number of vanishing point hypotheses M ($\phi = 2$, $M = 500$ for example).

The first step is to randomly choose M minimal sample sets of two edges $S_1, S_2, \ldots, S_M$ and to compute a vanishing point hypothesis $v_m = V(S_m, \vec{1})$ for each of them ($\vec{1}$ is a vector of ones, i.e. the weights are equal). The second step is to construct the preference matrix P which is a $N \times M$ Boolean matrix. Each row corresponds to an edge ε_n and each column to a hypothesis v_m. The consensus set of each hypothesis is computed and copied to the m^{th} column of P. Each row of P is called the *characteristic* function of preference set of edge ε_n. $P(n, m) = 1$ if v_m and ε_n are consistent, that is, $D(v, \varepsilon_n) \leq \phi$. Otherwise $P(n, m) = 0$.

The J-Linkage algorithm is based on the assumption that edges corresponding to the same vanishing point tend to have similar preference sets. It is true that any non-degenerate choice of two edges corresponding to the same vanishing point should yield solutions with similar, if not identical, consensus sets. The algorithm represents the edges by their preference sets and clusters them if they have similar preference sets.

The preference set of a cluster of edges is defined as the intersection of preference sets of its members. It uses the Jaccard distance between two clusters by Equation (3.10)

$$d_j(A, B) = \frac{|A \bigcup B| - |A \bigcap B|}{|A \bigcup B|} \tag{3.10}$$

where A and B are the preference sets of each of them. It equals 0 if the sets are identical and 1 if they are disjoint. The algorithm proceeds by placing each edge in its own cluster. At each iteration, the two clusters with minimal Jaccard distance are merged together. The operation is repeated until the Jaccard distance among all the clusters equals to 1. Once the clusters of edges are determined, vanishing points are calculated for each of them. Outlier edges often appear in the very small clusters, typically of two edges. If no refinement is performed, small clusters are defined as outliers and ignored.

Vanishing points are calculated for each cluster. In addition, they are refined using the EM algorithm. An optimal problem is written as Equation (3.11).

$$\hat{v} = \arg\min_{v} \sum_{s_j \in S} w_j^2 dist^2([\bar{e}_j]_\times v, e_j^1) \tag{3.11}$$

which is solved by the Lvenberg-Marquardt minimization algorithm by reference [64]. The

Figure 3.18: *An example of vanishing point detection*

function $V(S, w)$ where w is a vector of weights which defines the generation of vanishing points. A vanishing point is computed from a set of edges S by Equation (3.12).

$$V(S, m) = \begin{cases} l_1 \times l_2 & \text{if } S \text{ contains 2 edges} \\ \hat{v} & \text{if others} \end{cases} \tag{3.12}$$

The top three vanishing points are chosen by the number of associated edges in the clusters. These three either finite or infinite vanishing points typically represent the 3D geometry of this building. An example of detected finite vanishing points in an image is illustrated in Figure 3.18. The estimated vanishing point is in the position $(559, 357)$ and all the corresponding edges are marked in green.

3.4.2 Rectangular Planar Extraction

The fundamental component of VCI system in street view videos is to detect a valid rectangle on the building facade as the inserted region. The rectangular structure is regarded as composed of two pairs of parallel lines on the same plane, because these two sets of parallel lines correspond to the orthogonal vanishing points in the real world. This assumption works for most man-made environments, like the building facades in the street view scenes.

(a) *Before line refinement*

(b) *After line refinement*

Figure 3.19: *Detected horizontal lines on a building facade*

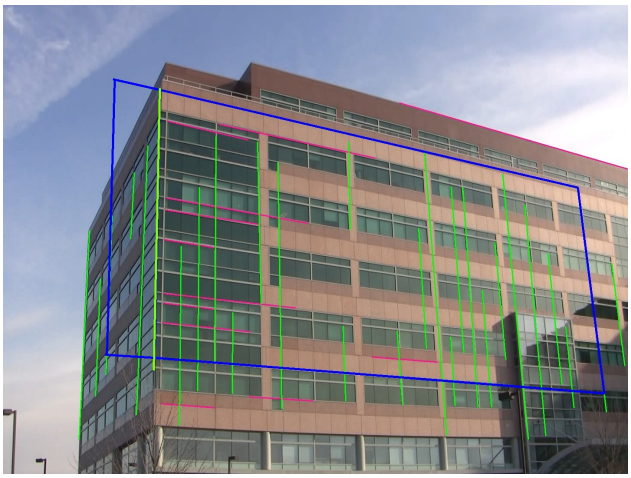


Figure 3.20: *An example of rectangle failing the corner verification*

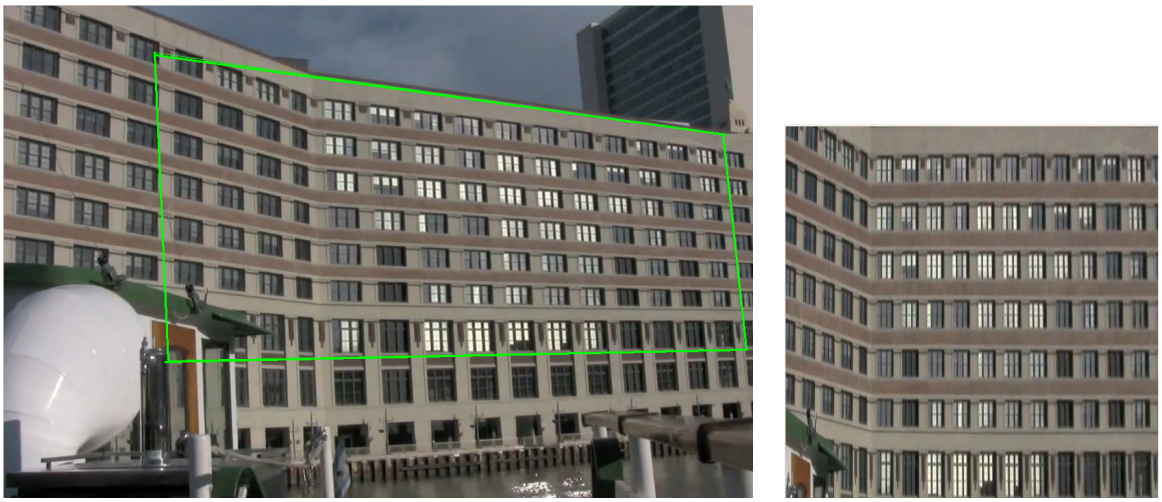
However, there are many possible combinations of two pairs of lines corresponding to the orthogonal vanishing points. An improved approach based on the method of verifying the rectangular structure hypothesis in [28] is applied in our VCI system to determine the biggest rectangular region for insertion purpose.

Firstly, the separated segments lying on the same line are merged and lines which are either close-by or too short are removed. Figure 3.19 is the detected horizontal lines denoted in green before and after the line refinement process. It clearly illustrates that some close lines are merged and short lines are removed.

Secondly, the valid rectangular planar must pass the model verification such as the corner points and dominant orientations verification. Similar to the camera calibration in sports scenes in section 3.3.3, two lines from the "horizontal" line set and two others from the "vertical" line set are combined together to obtain the homography matrix between the current rectangular planar and the front view unit model. The "horizontal" and "vertical" denote the directions associated with two out of three orthogonal vanishing points in the real world.

A rectangle is generated from this line combination, but not every one lies on a building facade. Two observation truth are used for rectangular structure hypotheses verifications.

- *Corner points verification*: The four corner points of this rectangle must be actual



(a) *A rectangle in green*

(b) *Front-view of the rectangular patch*

Figure 3.21: *An example of chosen rectangle failing the dominant directions verification*

corners in the building facade. Using this criteria, the case that intersections of lines are in the sky is deleted, as shown in Figure 3.20.

- *Dominant directions verification*: Since the rectangle is on a building facade, the front-view of this rectangular patch must only contain the horizontal and vertical directions. By recalling that any planar mapping between the 3D world planar and the 2D image plane is represented by Equation (3.5), a normalized front-view of this rectangular patch is obtained. Therefore, if the dominant directions of it are mainly horizontal and vertical, this rectangle will pass the second verification. The dominant directions are estimated by calculating the gradient histogram of canny edges. Figure 3.21 is an example of rectangle failing the dominant directions verification. The front-view rectangular patch not only contains horizontal and vertical edges, but also edges in other directions.

Similar to the lines ordering in the tennis scenario, all the "vertical" lines are sorted from left to right and the "horizontal lines" are sorted from top to bottom. In order to find the biggest valid rectangle efficiently, lines are chosen from outer to inner and such process is repeated until the composed rectangle passes both verifications. Figure 3.22 shows the

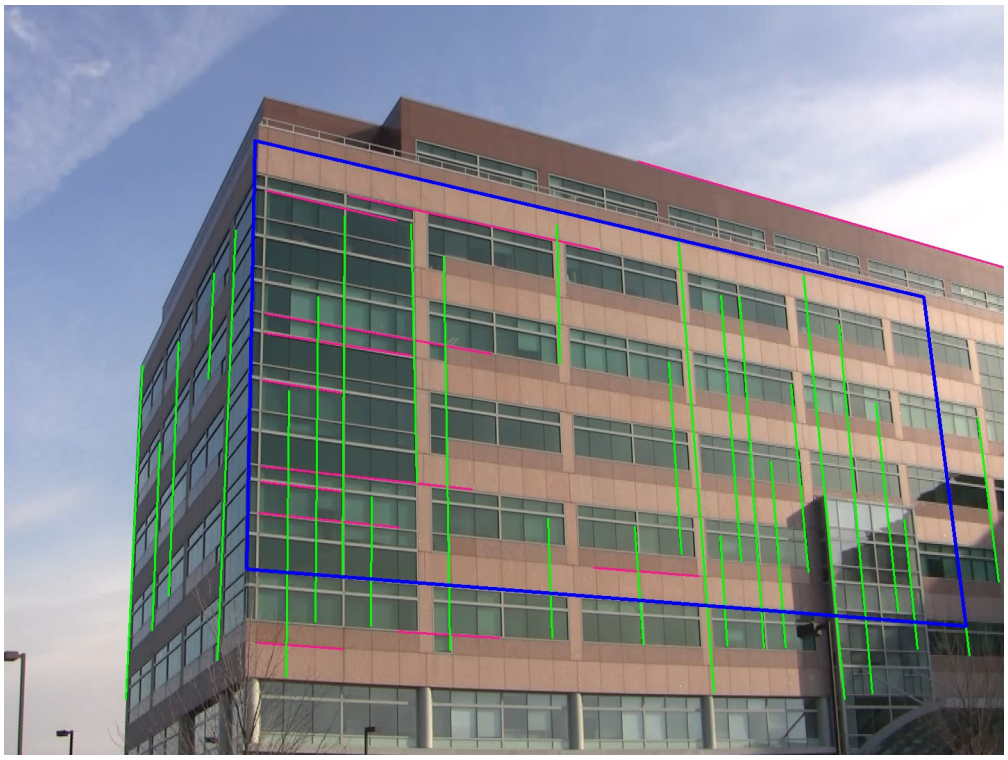


Figure 3.22: *Detected valid rectangle in a building scene*

detected rectangular region (in blue) with horizontal edges (in pink) and vertical edges (in green). It also coordinates with the visual observation.

3.4.3 Motion Stabilization

Dynamic registration is a challenging problem in VCI system. Although a closed-loop detection and tracking strategy is proposed and camera parameters are estimated quite accurately, there still exists the high frequency noise among the motion trajectory due to errors of homography matrix estimation. Moreover, a tiny error in KLT tracking may cause frequent switching between detection and tracking. An annoying jittering effect is observed in the experiments. Therefore, motion stabilization is required to reduce the jittering effect.

In the proposed VCI system, a buffer is created to store the processed results of several frames and the Wiener filter is applied to smooth the inserted positions similar to [65]. Assume the inserted positions in current frame are the linear combination of those in the previous N and following N frames. Let p_i^j denote the four corners of inserted region in the

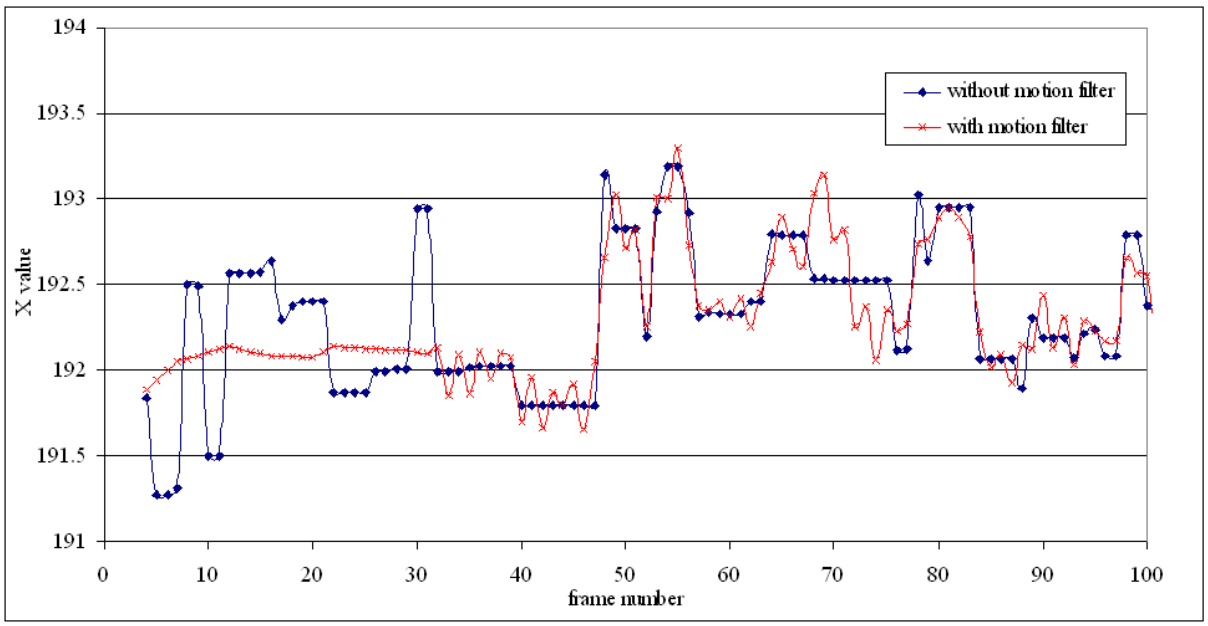


Figure 3.23: *X values of the first corner using Wiener filter*

i^{th} frame $(1 \leq j \leq 4)$:

$$\hat{p}_i^j = \sum_{k=-N}^{N} \alpha_{i+k} p_{i+k}^j \tag{3.13}$$

The $2N+1$ coefficients are estimated from training samples. For example, the number of buffer is M, then the training samples are $M-2N$. If the $2N+1$ neighbors of each sample are packed into a $1 \times (2N+1)$ row vector, a data matrix C with size $(M-2N) \times (2N+1)$ and a sample vector $\vec{P}$ with size $(M-2N) \times 1$ are generated. The optimal coefficients $\vec{\alpha}$ from a Least Square formulation to minimize $||\vec{P} - C\vec{\alpha}||^2$ have the closed-form solution given by Equation (3.14).

$$\vec{\alpha} = (C^T C)^{-1} C^T \vec{P} \tag{3.14}$$

Then the smoothed positions are calculated by Equation (3.13). An estimated homography matrix is calculated through camera calibration. Figure 3.23 shows the values of top-left position in the first 100 frames of a test tennis video when the Wiener filter is used or not.

(a) *without color harmonization*

(b) *with color harmonization*

(c) *without color harmonization*

(d) *with color harmonization*

Figure 3.24: *Virtual content insertion with/without color harmonization*

It demonstrates that the positions are indeed smoothed and also reflect the changes of data. In the experiments, the size of buffer and filter tap is defined as $M = 50$, $N = 9$.

3.4.4 Virtual Content Insertion

Virtual content (an image or a movie) is inserted seamlessly with the best homography matrix. In sports videos, the insertion region is determined by users. For example, we can choose one eighth of the goalmouth height above the goalmouth bar as the inserted region, or the region formed by two bottom horizontal and two left vertical lines in tennis videos. In street view videos, it is the biggest valid rectangular region. For a position $P(x, y)$ in the inserted region, the corresponding position $P^{'}(x, y)$ in the model coordinate system is determined by $P^{'} = H^{-1}P$. If the calculated coordinates are not integer, the bilinear

interpolation method is used to obtain the intensity from neighboring pixels with integer coordinates. Moreover, a simple segmentation is applied considering the player positions in tennis videos in order not to occlude the players. A simple playfield mask obtained from playfield extraction can help us to handle the pixels in the players.

Besides how to warp the virtual content in specific regions, how to less disturbs the viewers is also very important. One way is to make the virtual content blending with the background to reduce the sharp contrast of inserted content and background. A color harmonization algorithm based on the color model of Chang [24] is introduced in the VCI system.

Let $I(x,y)$, $I_{Ad}(x,y)$ and $\hat{I}(x,y)$ are the original image value, virtual content value and the estimated blending value at pixel (x,y). The playfield mask is $I_M(x,y)$, in which it is 1 if (x,y) in the playfield region Φ and 0 if not. Thus we have

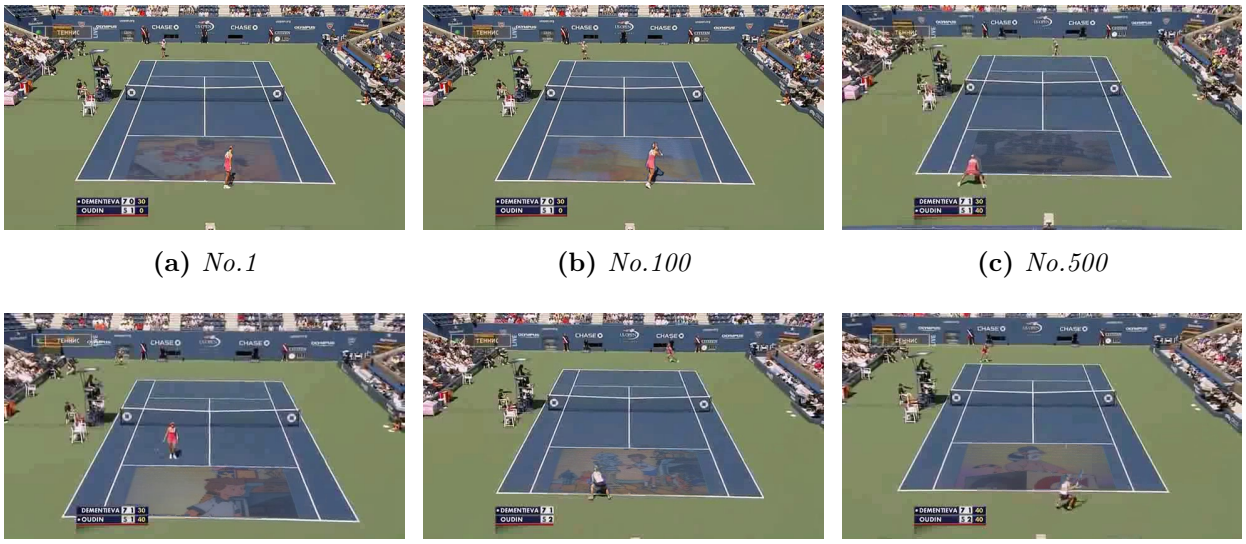
$$I_M(x,y) = \begin{cases} 0 & \text{if } (x,y) \in \Phi \\ 1 & \text{if others} \end{cases}$$

$$\hat{I}(x,y) = (1 - \alpha I_M(x,y))I(x,y) + \alpha I_M(x,y)I_{Ad}(x,y) \tag{3.15}$$

Based on the contrast sensitivity function, parameter α (normalized opacity) is estimated by Equation (3.16) [24]:

$$\begin{aligned} \alpha &= A \exp(f_0 \cdot f \cdot \frac{-\hat{\theta}_e(p,p_f)}{\theta_0}) \\ \hat{\theta}_e(p,p_f) &= \max(0, \theta_e(p,p_f) - \theta_f) \\ \theta_e(p,p_f) &= \arctan(\frac{||p - p_f||}{D_v}) \end{aligned} \tag{3.16}$$

where A is the amplitude tuner. f_0 is the spatial frequency decay constant (in degrees), f is the spatial frequency of the contrast sensitivity function (cycles per degree), $\hat{\theta}_e(p,p_f)$ is the general eccentricity (in degrees), $\theta_e(p,p_f)$ is the eccentricity, p is the given point in an image, p_f is the fixation point (for example, the player in tennis videos), θ_0 is the half resolution eccentricity constant, θ_f is the full resolution eccentricity (in degrees) and D_v is the viewing distance in pixels. We use the following values in our system: $A = 0.8$, $f_0 = 0.106$, $f = 5$, $\theta_f = 0.5$, $\theta_0 = 2.3$, D_v is approximated as 2.6 times of the image width. The parameter α controls the contrast of inserted content and background. Figure 3.24 illustrates the inserted



(a) *No.1* **(b)** *No.100* **(c)** *No.500*

(d) *No.1000* **(e)** *No.1500* **(f)** *No.2000*

Figure 3.25: *Inserting a movie in a US open video*

performance with and without color harmonization. The result with color harmonization is preferred and the blended insertion will reduce the disturbances of inserted content.

3.5 Experiments

3.5.1 VCI System in Sports Videos

I have downloaded several 640 × 360 tennis videos from US open, Wimbledon and Australia Open from *YouTube* for test purpose. Not only an image or a logo, but also a movie is inserted in the proposed VCI system. Figure 3.25 is the insertion result of a cartoon at frame # 1, 100, 500, 1000, 1500, 2000 in a US open video (2009 US Open Lena Vs Melanie). Figure 3.26 is another example of inserting a logo in a Wimblendon video (2010 Wimbledon Nadal Vs Soderling) at frame # 1, 100, 500. The frames with players close-up will not be processed in this system.

3.5.2 VCI System in Street View Videos

I captured some HD (1440 × 1080) street view videos with a hand held camcorder for test. In Figure 3.27, a poster of "mission impossible" is inserted at frame # 0, 49, 99, 149,

(a) *No.1* **(b)** *No.100* **(c)** *No.500*

Figure 3.26: *Inserting a logo in a Wimbledon video*

199 and 249 in Video A and in Figure 3.28 a logo of Huawei company is inserted at frame # 99, 199, 299, 599, 749 and 899 in Video B.

3.5.3 Complexity Analysis

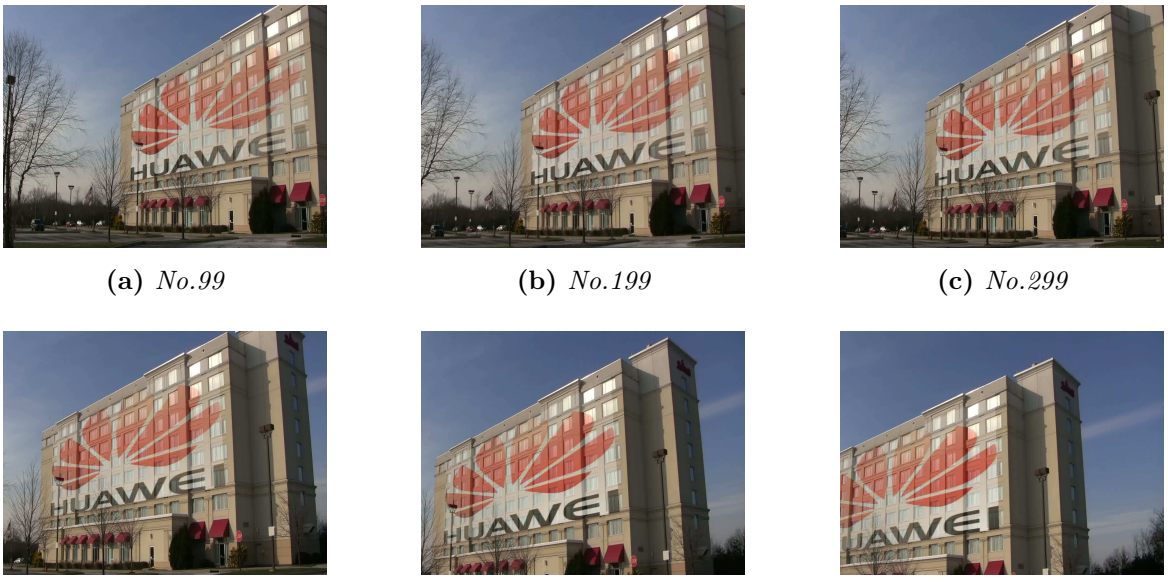
The proposed VCI system is implemented in C++ programming language based on OpenCV library. From the experimental results, the virtual to real registration and visual acuity looks satisfying, although there is a slight handshaking in street view videos and the tennis videos downloaded from Internet are not in a high definition format. The inserted content looks like it is pasted on the tennis court or on the building facade.

This VCI system is not a real time system, because motion stabilization is used to remove the jitters and some frames are stored in the buffer. However, each frame is processed efficiently in this system. The most time consuming part is model fitting in sports videos and rectangle detection in building videos. It usually takes an average one second to process each frame on Intel 2.2GHz duo CPU. It will automatically analyze the frame content and ignore those frames with players close-ups or advertisements. Each frame is determined automatically to be processed in detection or tracking. The percentage of tracking relies on the content of video sequences, and the difficulty of detecting lines and vanishing points. For example, the percentage of frames in tracking is 82% and it took about $4min$ to process the whole 300 frames in Video A. Therefore, the overall computational time is acceptable.

(a) *No.0* **(b)** *No.49* **(c)** *No.99*

(d) *No.149* **(e)** *No.199* **(f)** *N0.249*

Figure 3.27: *Inserting an image in a building scene*



(a) *No.99* **(b)** *No.199* **(c)** *No.299*

(d) *No.399* **(e)** *No.499* **(f)** *N0.569*

Figure 3.28: *Inserting a logo in a building scene*

3.6 Summary

In the proposed VCI system, where, when and how to insert the content so as not to disturb the viewers are challenging issues. In this chapter, a novel and generic VCI system for AR is introduced. To achieve a seamless virtual to real blending, a closed-loop detection and tracking combined approach is proposed to implement insertion in an efficient way. In the detection process, inserted positions are estimated in a Bayesian framework. In sports videos, the image prior is represented by the standard court model; While in street view videos, it is the 3D geometry model learned from the vanishing points extraction. Moreover, motion stabilization, dynamic registration and color harmonization are used to reduce the disturbance of inserted content.

The proposed VCI system in tennis video streams achieves a satisfying and robust performance. It works on four typical tennis courts such as US open, Australia open, French open and Wimbledon. In addition, it achieve a good insertion performance in some simple building scenes. To the best of my knowledge, it is the first successful system to insert content in street view video streams.

Chapter 4

Video Compression

Coding efficiency is an important objective of video codec design. The comparison of two coding systems is typically represented by the percentage savings in bit rate at equal subjective quality. Many efforts for improving coding efficiency have been made in the image/video compression field in the past decades. As the development of compression techniques, the service providers could send more television channels over the same bandwidth and users could enjoy more higher quality videos from online broadcasting. In addition, as more images and videos are transmitted and viewed on smartphones, it also motivates the improvement of compression techniques.

Video coding standards have been developed from MPEG2, H.264/AVC to the on-going HEVC, which achieves at least 40% bit rate deduction than H.264 high profile [66, 67] now. The development benefits from adding new algorithms and powerful tools in the codec structures. However, it costs numerous modifications in the complex coding system. Therefore, there is an alternative approach to use processing techniques while maintaining the encoder and decoder unchangeable.

In this chapter, an exemplar-based data pruning (*EDP*) video compression scheme for intra frame is introduced, which falls into the latter category to improve coding efficiency. It applies the pre- and post- processing techniques without modifying encoder and decoder. Before encoding, some video data is removed and replaced with a constant value based on an optimized pruning strategy. A patch library is established to describe the similarity of patches and guarantee that missing data can be reconstructed from available similar patches.

In addition, metadata which describes the pruning information is also transmitted to the decoder.

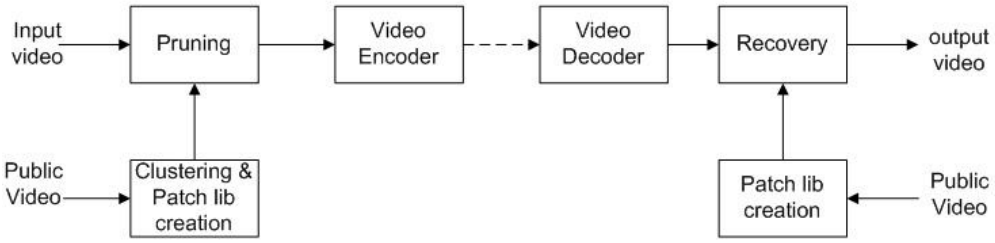
At the decoder, missing data recovery is regarded as an inpainting problem, although pruned blocks are replaced with constant values which are not completely missing and the decoded metadata also provides a side information about the pruning. Inspired by the latest exemplar-based approach in image inpainting, a post-processing method is proposed to reconstruct the pruned blocks. The patch to be filled is represented by a sparse linear combination of candidate patches under some constraints such as the neighboring pixels of target should be the known values. Linear combination coefficients are computed from a ℓ^1 norm regularization problem by the novel Bregman iteration algorithm [43]. This recovery process is a non-parametric approach with a sparseness image prior in the Bayesian framework.

4.1 EDP Video Compression Scheme

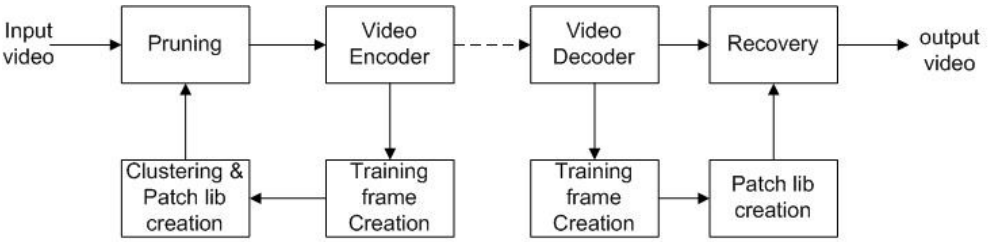
The idea of EDP video compression scheme comes from the consideration that redundancy exists in an image both locally and globally, especially when a repeated pattern or texture exists. According to the resource of training frame, the EDP video compression scheme is divided into two scenarios, as illustrated in Figure 4.1.

One scenario shown as Figure 4.1(a) occurs when the encoder and decoder are able to access to the same video database. This is a general video coding framework and has industrial potentials. For example, as a large amount of video data available on the Internet such as videos from *YouTube*, and the popular cloud computing coming into our lives as shown in Figure 4.2, the same patch libraries are established from the same training frames available at both encoder and decoder. Some video data is removed before encoding process. However, they are reconstructed from the same patch library at the decoder. This "reference patches" scheme breaks the limitations of hybrid block-based video coding since more shared resources are available at the encoder and decoder. Therefore, it will be an interesting and promising topic in the future, although it doesn't satisfy the current video coding constraints.

The other scenario is that training frames are from the previous decoded frames of a video sequence, as shown in Figure 4.1(b). The patch library is generated from the previous



(a) *EDP scheme when training frames from public resources*



(b) *EDP scheme when training frames from decoded frames*

Figure 4.1: *Diagrams of EDP video compression schemes when (a) training frames from the same resources and (b) training frames from current video sequence*

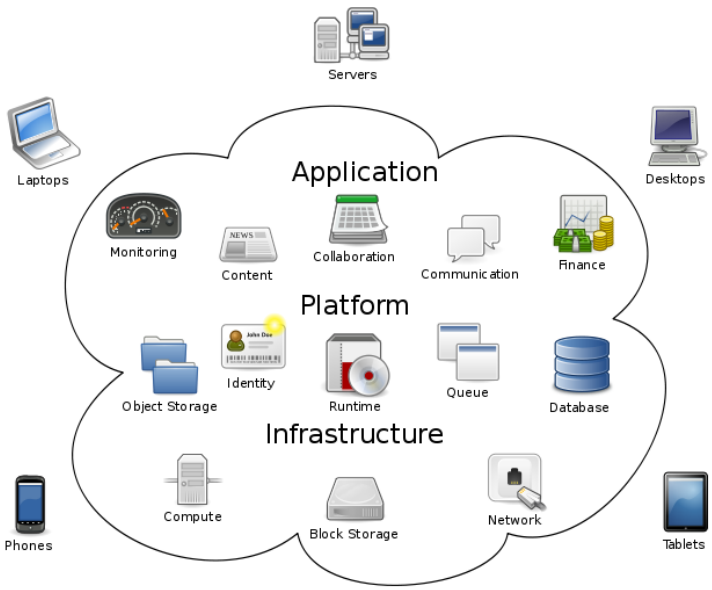


Figure 4.2: *Illustration of cloud computing which shares resources and provides information to computers and other devices as a utility over a network (image courtesy of* Wikipedia*)*

decoded frames. In the pre-processing stage, some blocks are removed and replaced with constant values in the pruning part. The pruning decision is made based on whether the pruned block could find similar patches in the patch library. The incomplete video sequence is passed through the encoder and decoder. There are two components in the post-processing stage: patch library creation and recovery. An exactly the same patch library is generated to recover the missing video data in the recovery process. Then output the reconstructed decoded video sequence. This scheme satisfies the current video coding constraints that original video data is unknown at the decoder. The proposed EDP scheme belongs to this scenario.

4.1.1 EDP Scheme for Intra Frame

In the proposed EDP video compression scheme, training frames for patch library generation are from the decoded frames. The first frame of a video sequence is usually encoded as the Intra frame, which tries to reduce the spatial redundancy. The following frames are often encoded as the Inter frame, which tries to reduce the temporal redundancy. Inter frame often needs less bit rates than Intra frame at equal subjective quality.

In this chapter, EDP video compression scheme for Intra frame is discussed. If pruning occurs in the Inter frame, metadata which tells the decoder pruning information should contains more information such as which frame the best similar patch of a pruned block is from. Moreover, the generated inter frames which have some missing data would damage the temporal redundancy and cause bit burdens when they are encoded. Therefore, how to apply the pruning technique to Inter frames needs further studies and experiments. This chapter will only discuss the EDP scheme in Intra frame.

The pre- and post- processing tools are applied in the proposed EDP scheme. In the pre-processing stage, in order to prune some video data in Intra frame, a patch library is established from patches in the original frame before encoding. Although the similar patch founded in the decoded frame by the assisted metadata is not the same patch in the original frame, there is an assumption that two similar patches in the original frame are still similar for their decoded patches. Then some blocks are removed and replaced with the average value of

that block. Encode and decode this synthesized frame and also the metadata which contains the pruning information. In the post-processing stage, pruned blocks are reconstructed by inferring from the patch library with metadata. The sparseness of patches is utilized to represent the pruned patch by a sparse linear combination of other similar patches. This recovery process is an inpainting problem solved by a non-parametric approach in a Bayesian framework. In all, the proposed EDP video compression scheme for Intra frame is described as follows. Assume the pruned and non-pruned part are p and f in the first frame I. The reconstructed image $\hat{I}$ is composed of decoded $\hat{f}$ and inpainted $\hat{p}$. The assisted metadata is θ. The overall coding performance is measured by RD (Rate-distortion) cost.

$$J = D(f, \hat{f}) + D(p, \hat{p}) + \lambda(R_f + R_p + R_\theta) \tag{4.1}$$

The smaller RD cost is, the better coding efficiency it achieves. However, it's unfortunately that this RD cost function can't be used directly as a guide to determine whether a block should be pruned or not, because it's very difficult to estimate the afterwards encoding performance and inpainting quality. Therefore, pruning before encoder and recovery after decoder are processed separately. The proposed recovery process is better than the method simply copying the best similar patch included in metadata, so separating the two part still achieves a global minimum RD cost.

4.2 Pre-Processing Techniques Before Encoder

In this section, pre-processing techniques before the encoder are introduced in details. At the encoder, a patch library is created from the original frame to describe the similarity of patches. Then part of video data in the Intra frame is removed by a pruning strategy and replaced with the average value of that block. Finally the synthesized picture which is a mixture of pruned and unpruned blocks is encoded by any video encoder and transmitted to decoder. In addition, the metadata including pruning information is also encoded and transmitted to decoder in order to assist the recovery of missing data. Figure 4.3 illustrates the whole process at the encoder. Patch library generation, pruning strategy, pruning process and metadata transmission are presented in the following part.

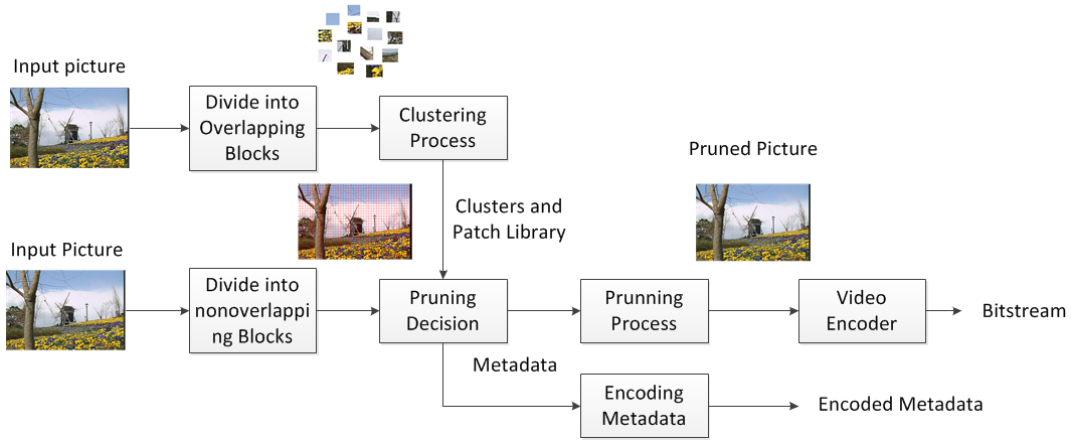


Figure 4.3: *Illustration of pre-processing part at the encoder side*

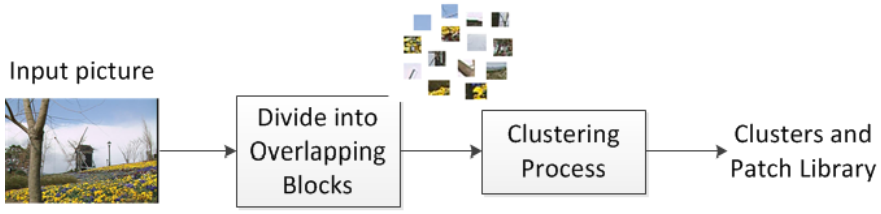


Figure 4.4: *Illustration of patch library generation*

4.2.1 Patch Library Generation

The novelty of this EDP scheme is to establish a patch library which describes the similarity of small patches. The generation process is as follows. The first frame of a video sequence is divided into overlapping blocks with a certain overlapping step. Then a large amount of small patches are obtained. The patch library is a pool composed of these small patches. During the pruning process, the best similar patch of each block will be located and hence patch library has achieved its task by linking similar patches together. However, in order to better organize small patches and find the best similar patch of each block efficiently, a clustering process is necessary. Figure 4.4 shows the generation of patch library as the first step of EDP scheme.

In clustering process, similar patches are grouped together. There are two fundamental issues in any clustering algorithm: 1) What is the definition of similarity? 2) What is the feature vector composed of? In order to exceed the clustering process, the downsized version of patches is used. In my implementation, the size of patches in the original picture is

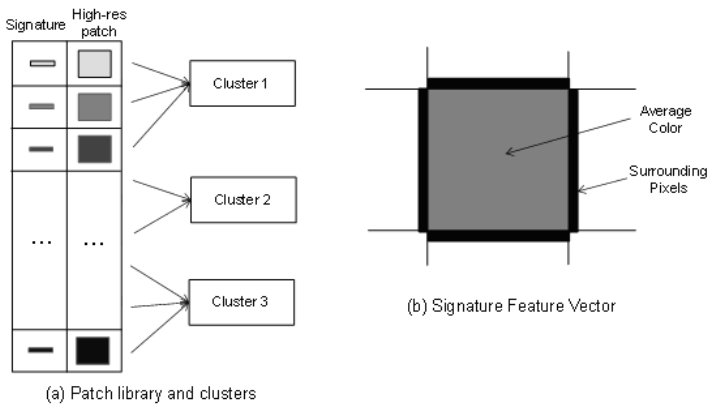


Figure 4.5: *Illustrations of signature vectors in a patch library*

16×16 and hence the size of downsized patches is 4×4. Assume there are N 16×16 patches $\{P_1, P_2, \cdots, P_N\}$ and corresponding 4×4 patches are denoted as $\{p_1, p_2, \cdots, p_N\}$. The 16 pixel values in p_i form the feature vector $X_i = \{x_1, x_2, \cdots, x_{16}\}$.

First of all, blocks with a small variance are ignored, because they are regarded as flat regions and pruning these blocks usually will not achieve gain in terms of bit rate reduction. Then a modified K-means clustering algorithm [40] is applied to classify the non-flat patches. K-means algorithm is one of the simplest unsupervised learning algorithms to solve the clustering problem. The details are found in [68]. In order to speed up the clustering process, the overlapping patches are generated in different phases and K-means is executed in each phase.

Assume the overlapping step is (bh, bw) in the vertical and horizontal direction, then there are $(16/bh) \times (16/bw)$ phases. In each phase $(i, j), (0 \leq i < 16/bh, 0 \leq j < 16/bw)$, an image with the left-top corner at $(1 + i \times bh, 1 + j \times bw)$ is obtained if the original left-top position is $(1, 1)$. The clustering is executed as follows. Firstly, clustering is executed in the image with phase $(0, 0)$. Non-overlapping blocks are divided and regarded as the input data set of K-means algorithm. The blocks will be classified into K groups and the objective function is minimized as:

$$J = \sum_{j=1}^{K} \sum_{i=1}^{N} \|X_i^j - C_j\|^2 \tag{4.2}$$

where C_j is the centroid of group j. J is minimized when all the centroids are not changed.

Algorithm: Clustering the overlapping blocks
Input: Image I
Initialize K centroids and downsample I
Repeat processing the image with phase (i, j)
  If ($i == 0$ and $j == 0$)
    Cluster non-overlapping blocks into K groups with K-means algorithm
  else
    Repeat classifying non-overlapping blocks
      Classify a block and update its centroid
    Until all blocks are processed
Until all phases are processed
Output: K centroids and K clusters

Table 4.1: *Descriptions of clustering algorithm*

Secondly, clustering is executed in other phases with the new left-top corner at other positions. For example, in phase $(1, 1)$, the new position is $(1 + bh, 1 + bw)$ with the the step size (bh, bw). Every non-overlapping block in current image is inserted into the already classified groups. It will belong to the group with the minimum distance to its centroid and then the centroid is updated among all the items. This process is continually executed until all the blocks are processed and the objective function goes to the minimum. The clustering algorithm is listed in Table (4.1).

Fast K-means algorithm is utilized and hence the above clustering process is executed in an efficient way. The patch library is composed of K groups represented by centroids. Furthermore, each patch is represented by a signature vector. It consists of the average color of the patch and the surrounding pixels of the patch, as shown in Figure 4.5. On one hand, usage of signature vectors makes searching the best similar patch more efficiently. On the other hand, it can make the metadata encoding process more efficiently, which is introduced in section (4.2.4).

Once the patch library is established, the best similar patch of every non-overlapping block is obtained through the searching process, as illustrated in Figure 4.6. First of all, the block is compared with the centroids by calculating the Euclidean distance of feature vectors in the downsized version. The top M matched clusters are selected. Currently, M is determined empirically as 25 in my implementation. In principle, M should be determined

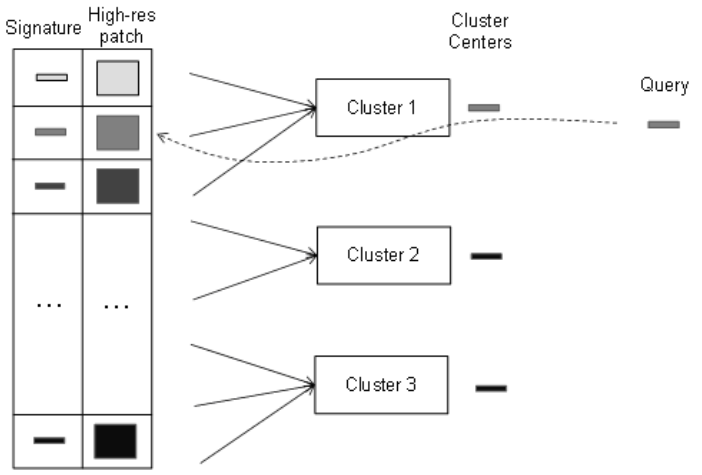


Figure 4.6: *Best similar patch searching process*

by the error bound of the clusters. Secondly, the searching process is continued within the M candidate clusters until the Euclidean distance between feature vectors of current block and candidate patch comes to the minimum value. At last, a picture composing of best similar patch of each non-overlapping block is obtained as an input of pruning strategy module.

4.2.2 Optimized Pruning Strategy

Pruning strategy makes a decision of pruning a block or not. It's critical to this EDP video compression scheme. There are several strategies, for example, compare the distortion of original block and the best similar patch with a threshold. It may achieve some improvements at large bit rate, but it usually have large distortion at low bit rate. Therefore, we propose a decision strategy considering both distortion and bit rate to guarantee that there will not be a loss within a large range of bit rate.

From rate distortion theory [67], the goal of an encoder is to optimize its overall fidelity, that is, minimize the distortion D subject to a constraint Rc on the number of bits R. This constrained problem is solved by using the Lagrangian optimization method. The Lagrangian formulation is given by equation 4.3.

$$\min J = D + \lambda R \tag{4.3}$$

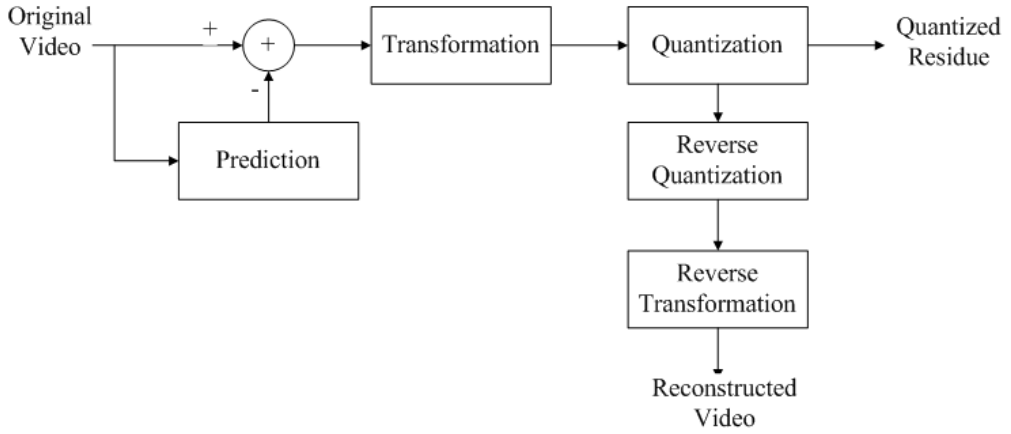
The pruning decision algorithm is shown in Table (4.2). The first frame is divided into

Algorithm: Pruning decision process
Input: Intra frame I and its patch-level best similar picture P
Initialize Pruning decision matrix $H = 0$
Repeat processing the block with the left-top corner at (i, j)
Calculate RD cost J_1 when it's not pruned (Case 1)
Calculate RD cost J_2 when it's pruned (Case 2)
If $J_1 < J_2$
$H(i, j) = 0$
else
$H(i, j) = 1$ and the best similar block is set to be unpruned
Until all blocks are processed
Output: H

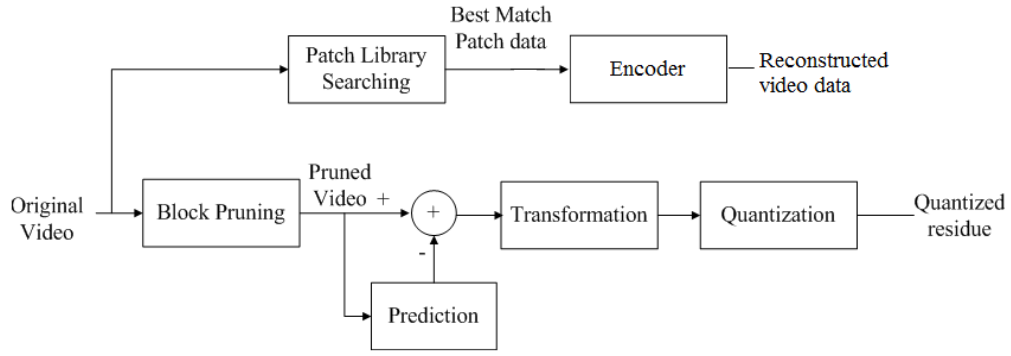
Table 4.2: *Descriptions of pruning decision algorithm*

non-overlapping blocks with size 16×16. A progressive scanning order the same as the encoding order in H.264/AVC is adopted. The decision process is repeated for each block until all the blocks are processed. In every iteration, first of all, RD cost J_1 for Case 1 is calculated. Case 1 denotes that the current block is not pruned, which means the original video data is kept intact. Secondly, RD cost J_2 for Case 2 is computed. In Case 2, the current block is pruned, which means the whole block is replaced by the average color of the original values. Thirdly, two RD costs are compared and a decision is made. If $J_2 < J_1$, prune this block and the best similar block is set to be unpruned if it has not been processed, otherwise, keep the original values.

The RD cost estimation should imitate the encoder characteristics and then the pruning strategy is able to make a correct decision so as to achieve a trade-off between distortion and bit rate. The estimation of RD cost in each block is similar to the intra-mode decision in H.264/AVC encoder. Distortion is estimated between the original and reconstructed video data. Rate is the whole bits of residue and mode selection. Residue is obtained through the prediction process. There are nine prediction directions in H.264/AVC intra mode, such as vertical, horizontal, DC, diagonal-down-left, diagonal-down-right, vertical-right, horizontal-down, vertical-left, and horizontal-up. Prediction directions are illustrated in Figure 4.8, and number $0 - 8$ stand for the corresponding directions. DC direction is not shown and its prediction is the average of neighboring pixels.



(a) *Case 1: non-pruned block*



(b) *Case 2: pruned block*

Figure 4.7: *Distortion estimation in pruning decision process for (a) Case 1 the block is not pruned (b) Case 2 the block is pruned*

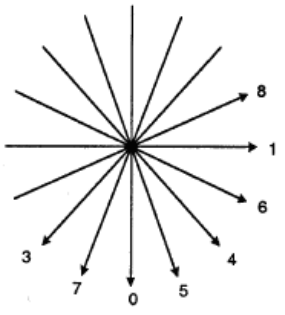


Figure 4.8: *Intra prediction direction in H.264/AVC*

Figure 4.7 shows the distortion estimation for Case 1 and Case 2. In Case 1, the input of encoder is the original video data. Residue after prediction goes through the transformation and quantization process, and then the reconstructed video data is obtained through the reverse quantization and reverse transformation. Therefore, the distortion is calculated in mean square error (MSE) between the original video and the reconstructed one. For data X and Y with size 16×16, the definition of MSE is as follows:

$$MSE = \sum_{j=1}^{16} \sum_{i=1}^{16} (X_{ij} - Y_{ij})^2 \tag{4.4}$$

While in Case 2, the input of encoder is a pruned block which has a constant value in the flat region. The best match patch of this block is available through searching in the patch library. Once Case 2 is chosen for a certain block, the best match patch will not be pruned. Similar to Case 1, the reconstructed best match patch is estimated. Therefore, distortion is calculated in MSE between the original video data and reconstructed best match patch, as shown in Equation (4.4). However, the residue of flat block after prediction still goes to transformation and quantization process. The encoded residue is one part of bitstream together with other bits on the encoded mode and metadata which consists the pruning information.

Rate is estimated in two ways. One simple way is to estimate the non-zero coefficients of quantized residue and approximated mode bits. The other way is more complex by estimating the bits in entropy coding the quantized residue. CAVLC or CABAC is used in the entropy coding process. In current implementation, the simple way is used, but experiments show that it's a good estimation of actual RD cost. It should be mentioned that Case 2 not only includes the bits of residue and mode, but also the side information about the position of best match patch. Therefore, the rate in Case 2 includes the two parts.

RD cost is calculated after distortion and rate are estimated. Similar to the λ used in the intra prediction mode of H.264/AVC, in current implementation the Lagrangian parameter λ is also related to the quantization parameter QP, as shown in Equation (4.5).

$$\lambda = 0.68 \times 2^{(QP-12)/3} \tag{4.5}$$

Figure 4.9: *A mixture of pruned and non-pruned blocks after pruning process*

4.2.3 Pruning Process

In the pruning decision process, the output is a decision matrix H in which $H(i,j) = 1$ denotes that block at (i,j) is pruned. In the pruning process, the first frame is divided into non-overlapping blocks with size 16×16. If $H(i,j) = 1$, the pixels in block at (i,j) is removed and replaced with the average value of that block. If $H(i,j) = 0$, the block is maintained. Figure 4.9 is an example of mixed picture after pruning process.

In Liu's paper [37], the removed patch is skipped in JPEG encoding. This patch-based inpainting scheme needs to modify codec structures, which violates the motivation of applying processing tools to improve coding efficiency. Moreover, it will cause the encoder in a low efficiency since spatial correlation is damaged. By contrast, in the proposed EDP scheme, the synthesized picture is a mixture of pruned and non-pruned video data and it is able to go through encoder and decoder directly. Although this pruning approach affects the spatial correlation of original picture as well, it doesn't completely damage the intra mode prediction. It achieves a trade-off between reduced bits and a bigger distortion in an overall measurement.

4.2.4 Metadata Composition

How to recover the pruning blocks is also an important part in this scheme. Since the goal is to estimate the missing data, image inpainting algorithms are just for this purpose.

Flag	Threshold	block IDs	Best match Patch IDs

Table 4.3: *Metadata composition*

If adopting a conventional inpainting method, it will usually fail when large part is pruned. More importantly, since this EDP scheme is for compression purpose, some side information is able to be transmitted to the decoder in order to assist the missing data recovery. Which blocks are pruned? Where are the best match patches of pruned blocks in the decoded frame? These information should be sent to the decoder.

Most importantly, the locations of pruned blocks are needed to tell the decoder which blocks are pruned. One simple way is to send a block ID sequence, which indicates the coordinate of each block in a frame. However, it's not efficient, because the pruned blocks are flat and contain no high-frequency components. This characteristic should be fully utilized. Therefore, a novel way of representing pruned blocks ID is proposed. After encoding, a threshold is determined to guarantee that the variance of all decoded pruned blocks is under it. Sometimes there are some non-pruned blocks whose variances are also under this threshold, which are regarded as false blocks. If the number of false blocks is smaller than the number of pruned blocks, block IDs of false blocks are encoded in the metadata; Otherwise, block IDs of pruned blocks are encoded.

Therefore, by calculating the variance of each block at the decoder, the blocks with smaller variance than threshold are chosen. If false block IDs are sent, the pruned blocks IDs can be the ones excluding the false IDs in the chosen blocks. If pruned block IDs are sent, they are directly obtained. The threshold and a flag denoting that whether false or pruned blocks are adopted are also included in the metadata.

Moreover, the best match block ID of each pruned block is also one part of metadata. The metadata composition is shown in Table (4.3).

- **Flag**

  When flag is 0, the block IDs are the index of pruned blocks. When it's 1, the block IDs are the index of false blocks.

- **Threshold**

Threshold is encoded as an unsigned number in the variable-length code.

- **Block IDs**

  The block IDs are sorted so that the numbers are placed in an increasing order. To further reduce redundancy, a differential coding scheme is used, because the difference between an ID number to its previous one is usually smaller than the number itself. For example, assuming the ID sequence is $3, 4, 5, 8, 13, 14$, the differential sequence becomes $3, 1, 1, 3, 5, 1$. This differentiation process makes the numbers smaller, and therefore results in a distribution with smaller entropy.

- **Best match patch IDs**

  The best match block ID of each pruned block is included in the metadata to notify the decoder the similarity of pruned block and the rest of image. The ID sequence is also encoded in the differential code. Then the whole bitstream will be further compressed with entropy coding, for example, Huffman coding in the current implementation.

4.3 Post-Processing Techniques After Decoder

Image inpainting is an ill-posed problem with the goal of filling in the missing data in an incomplete image, which is widely investigated in object removal, error concealment and image restoration. In the proposed EDP scheme, the recovery process at the decoder is regarded as an inpainting problem, although the pruned blocks are replaced with flat regions which are not completely unknown. In addition, metadata also includes some side information. However, similar idea as the exemplar-based image inpainting algorithm is adopted in the missing data recovery at the decoder.

The bitstream containing the synthesized picture and metadata are transmitted to the decoder. After correctly decoding the bitstream, the reconstructed picture and metadata are obtained to recover the pruned blocks. The goal of this post-processing module is to estimate the missing data using the decoded non-pruned blocks and metadata with inpainting techniques. Figure 4.10 illustrates the post-processing part at the decoder.

Diffusion-based approach [69, 70] is a general solution for image inpainting, in which the

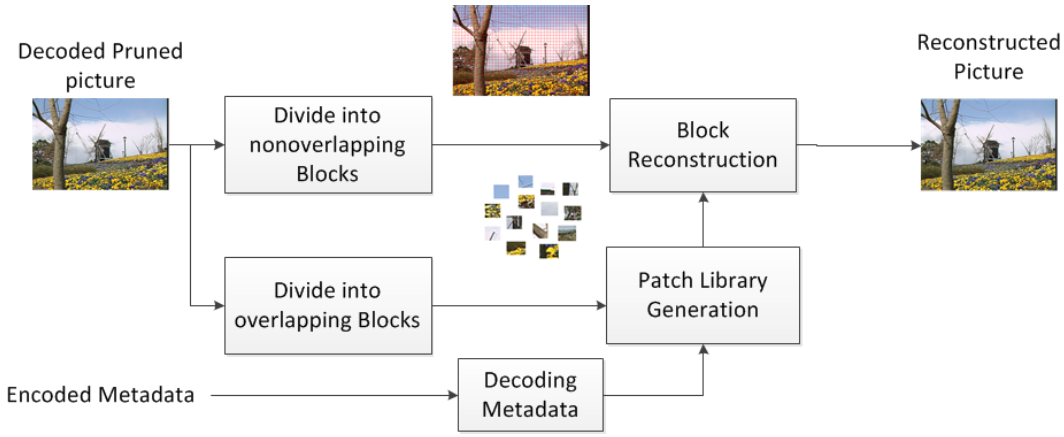


Figure 4.10: *Illustration of post-processing part at the decoder side*

missing region is filled by diffusing the image information from the known region at the pixel level. These algorithms are based on partial differential equation (*PDE*). Another statistical approach is based on the image statistics learned from the natural images in [51, 47, 71]. These algorithms achieved very good results in filling the non-texture region or relatively small holes. However, they tend to oversmooth the texture region and often fails in the large holes.

Different from the above approaches, exemplar-based inpainting approach will propagate the image information from the known region into the missing region at the patch level. The state-of-art algorithm in [72] achieved good visual quality not only in texture region but also in the complex region composed of textures and structures. Inspired by the latest exemplar-based approach, an inpainting algorithm is proposed to recover the pruned blocks in the proposed EDP scheme. The pruned blocks are located and their best match blocks are also known from the decoded metadata. By searching the similar patches of the best match block in the known decoded region, the pruned block is represented by a sparse linear combination of candidate patches. Linear coefficients are obtained by solving a constrained optimization problem. This non-parametric approach utilized the sparseness of patches in the Bayesian framework.

Algorithm: Missing data recovery
Input: Decoded image $\hat{I}$, pruned blocks $\{\Psi_1, \Psi_2, \cdots, \Psi_J\}$
best match blocks $\{B_1, B_2, \cdots, B_J\}$.
Repeat recovering pruned block Ψ_P
Step 1: Establish patch library L and cluster it using K-means algorithm
Step 2: Obtain N similar patches of B_P
Step 3: Solve the optimized problem in Equation (4.11)
Estimate the pruned block in Equation (4.6)
Update reconstructed image $\hat{I}$
Until all pruned blocks are processed
Output: reconstructed image $\hat{I}$

Table 4.4: *Descriptions of missing data recovery algorithm*

4.3.1 Missing Data Recovery

Denote pruned blocks as $\{\Psi_1, \Psi_2, \cdots, \Psi_J\}$ and the corresponding best match blocks as $\{B_1, B_2, \cdots, B_J\}$. Ψ_i and Ψ_j are discrete blocks in reconstructed image $\hat{I}$. The goal is to estimate the pruned blocks $\{\hat{\Psi}_1, \hat{\Psi}_2, \cdots, \hat{\Psi}_J\}$. The recovery of the pruned block Ψ_P consists of several steps, as shown in Table (4.4).

In the first step, a patch library L is established using the decoded non-pruned blocks. Similar to the patch library generation at the encoder, L is clustered into several groups and each group is represented by its centroid.

In the second step, the best match patch B_P will find several similar patches in L. Similar to searching the best match patch, B_P is compared with centroids to find a cluster with the minimum Euclidean distance between B_P and its centroid. Then the searching continues into that cluster and N candidate patches $\{\Psi_q\}_{q=1}^{N}$ are found. They are sorted in an increasing order according to the Euclidean distance between B_P and Ψ_q. N is set to be 25 in current implementation.

In the third step, the pruned block Ψ_P is approximated as the linear combination of $\{\Psi_q\}_{q=1}^{N}$ in Equation (4.6). Figure 4.11 is an example to represent the missing block with other candidate patches.

$$\hat{\Psi}_P = \sum_{q=1}^{N} \alpha_q \Psi_q \tag{4.6}$$

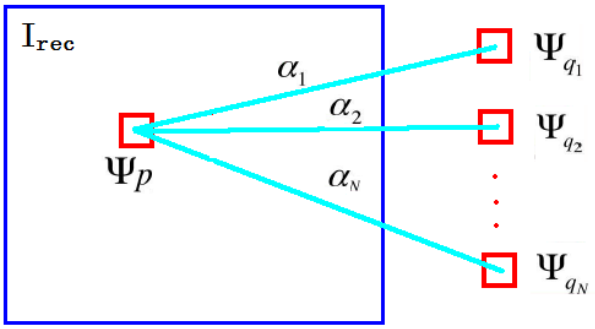


Figure 4.11: *Illustration of a pruned block recovery*

4.3.2 Constrained Optimization Problem

In above recovery process, the most important step is to solve the combination coefficients $\vec{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$. They are inferred by minimizing a constrained optimization problem in the framework of sparse representation. The objective of this constrained problem is to minimize the ℓ^0 norm of $\vec{\alpha}$, which is the number of nonzero elements in vector $\vec{\alpha}$, because the patch Ψ_P is represented by the sparsest linear combination of candidate patches and hence sparseness of combination coefficients is the objective.

There are two constraints of this optimization problem. The first constraint is on the appearance of $\hat{\Psi}_P$, which is a local patch consistency constraint. In order to make the estimated $\hat{\Psi}_P$ consistent with the neighboring patches, the estimated $\hat{\Psi}_P$ should approximate the target patch Ψ_P over the neighboring known pixels.

Assuming the size of Ψ_P is $m \times n$. A bigger patch Φ_P with size $(m+k) \times (n+k)$ includes the target patch Ψ_P at the center and the neighboring pixels. Then the problem estimating $\hat{\Psi}_P$ is converted to estimate $\hat{\Phi}_P$ as the sparse linear combination of $\{\Phi_q\}_{q=1}^N$. Similar to Equation (4.6), $\hat{\Phi}_P$ is represented by Equation (4.7).

$$\hat{\Phi}_P = \sum_{q=1}^{N} \alpha_q \Phi_q \tag{4.7}$$

A mask M with the same size $(m+k)\times(n+k)$ is defined to describe the known neighboring pixels. Then the constraint condition is represented by Equation (4.8).

$$\|M \circ \hat{\Phi}_P - M \circ \Phi_P\|^2 < \epsilon \tag{4.8}$$

where ϵ is a parameter to control the error tolerance of this approximation.

Another constraint is that the summation of the coefficients vector $\vec{\alpha}$ equals to one: $\sum_{i=1}^{N} \alpha_i = 1$. This constraint is widely used in the local linear embedding literature for invariance to transform when reconstructing the target patch from its neighboring candidate patches.

Finally, $\vec{\alpha}$ is able to be inferred by optimizing the following constrained optimization problem:

$$\begin{aligned} &\min\{||\vec{\alpha}||_0\} \\ s.t. &\|M \circ \hat{\Phi}_P - M \circ \Phi_P\|^2 < \epsilon \\ &\sum_{i=1}^{N} \alpha_i = 1 \end{aligned} \tag{4.9}$$

Then the missing block $\hat{\Psi}_P$ is obtained from the optimized $\hat{\Phi}_P$ as:

$$\hat{\Psi}_P = (1 - M) \circ \hat{\Phi}_P \tag{4.10}$$

4.3.3 Optimization Algorithm

The constrained optimization problem is formulated as an energy minimization problem in Equation (4.11). It is a derived format of Bayesian framework.

$$\vec{\alpha}^* = \arg\min_{\vec{\alpha}} \left\{ ||\vec{\alpha}||_0 + \lambda \|M \circ \hat{\Phi}_P - M \circ \Phi_P\|^2 + \eta \| \sum_{i=1}^{N} \alpha_i - 1 \|^2 \right\} \tag{4.11}$$

It is equivalent to Equation (4.10) when parameters λ and η are chosen properly. The this energy minimization problem achieves a global minimized solution.

Generally, the ℓ^0 norm regularized model is a NP problem. Matching pursuit (MP) in [73] and basis pursuit (BP) in [74] are main algorithms to retrieve the sparse representation and approximate the optimal solution in a greedy fashion, but the results are unreliable. Recently, a breakthrough for this NP problem is to convert it to a ℓ^1 norm regularization problem. The most well-known ℓ^1 norm regularization algorithm is Lasso [75] in literature. Xu in [72] proposed a greedy algorithm similar to MP algorithm. The coefficients are updated during iterations when the others are fixed, but it's not a fast algorithm.

Algorithm: Bregman iterative regularization
Input: $J(u)$, $H(u,f)$ and λ
Initialize: $k=0$ $u^0=\vec{0}$, $f^0=\vec{0}$
Repeat Bregman iteration
Step 1: Solve $u^{k+1} \rightarrow \arg\min_u \|u\|_1 + \lambda\|Au - f^k\|_2^2$ by coordinate descent
Step 2: $f^{k+1} \rightarrow f + (f^k - Au^{k+1})$
Step 3: $k \rightarrow k+1$
Until $\frac{\|Au-f\|_2}{\|f\|_2} < 1 \times 10^{-5}$
Output: optimized solution u

Table 4.5: *Bregman Iteration algorithm*

I use the state-of-art Bregman iteration algorithm to solve this ℓ^1 norm regularization problem [43, 76, 77]. Equation (4.11) is rewritten in a standard matrix formulation and the second constraint is applied when an optimized solution is solved from the following function:

$$\min_u |u|_1 + \lambda\|Au - f\|_2^2, \tag{4.12}$$

The **Bregman distance** is defined as in [76].

$$D_J^p(u,v) = J(u) - J(v) - \langle p, u-v\rangle,\ p \in \partial_u J$$

where $p \in \partial J(u)$ is an element in the subgradient of J at the point u. Assume $J(u) = |u|_1$ and $H(u,f) = \lambda\|Au-f\|_2^2$ in this optimization problem.

The algorithm is listed in Table 4.5. In Step 1, the convex function $E(u) = |u|_1 + \lambda\|Au - f^k\|_2^2$ is minimized. There is a simple closed formular for the solution of each coordinate subproblem, which makes the coordinate descent method efficient for solving it. Details are found in [43].

The solution of the one dimension same problem $E(x) = |x| + \lambda(x-f)^2$ is $x = shrink(f, \frac{1}{2\lambda})$, the shrink operator is defined as follows:

$$shrink(f,\mu) = \begin{cases} f-\mu & \text{if } f > \mu \\ 0 & \text{if } -\mu \le f \le \mu \\ f+\mu & \text{if } f < -\mu \end{cases} \tag{4.13}$$

For each coordinate subproblem, all components of u except the jth component u_j are freezed. Let a_j denote the jth column of A and a_{ij} denote the element in the ith row and

jth column. f_i is the ith component of f. Then the problem is rearranged as follows.

$$\begin{aligned}\min_{u_j} E(u) &= \lambda \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} u_j - f_i \right)^2 + \sum_{i=1}^{n} |u_i| \\ &= \lambda(\|a_j\|_2^2 u_j^2 - 2\beta_j u_j + \|f\|_2^2) + |u_j| + \sum_{i \neq j} |u_i|\end{aligned} \tag{4.14}$$

where $\beta_j = \sum_{i=1}^{m} a_{ij}(f_i - \sum_{k \neq j} a_{ik} u_k)$. The optimal value for u_j with all the other coordinated fixed is

$$\tilde{u}_j = \frac{1}{\|a_j\|_2^2} shrink(\beta_j, \frac{1}{2\lambda}) \tag{4.15}$$

We can delete the whole jth column of A since the value of u_j does not affect f at all. Therefore, $\tilde{u}_j$ decreases the energy function $E(u)$. Each $\tilde{u}_j$ will be obtained in an iteration. Repeat this iteration process until $E(u)$ converges.

The parameter λ controls the weight in the penalty term and it's set to 100 in current implementation. The Bregman iteration achieves an accuracy solution. In addition, it is more efficient than Xu's method.

4.4 Experiments

4.4.1 Comparison with H.264/AVC

In the proposed EDP scheme for intra frame, the pruning decision guarantees that no pruning is conducted on a certain block if pruning can't obtain some gain. The gain comes from the fact that several similar patches exist in the picture. In order to test this scheme, six sequences are used with visual similarity in the first frame. Table 4.6 lists the name, size and frame rate of these test sequences.

The test conditions of H.264/AVC encoder are as follows: High profile, transform 8×8 is on and only encode the first frame. The quantization parameter QP is $(20, 26, 32, 38, 44)$. The bench mark is encoding the original frame in H.264/AVC high profile. PSNR of luminance component is calculated between original and reconstructed picture. Bitrate is derived from bits of encoded picture. While in the proposed EDP scheme, encoder and decoder from

Sequence	**size**	**Frame rate(fps)**
independence_day	720×480	24
american_pie	720×480	24
opening_ceremony	720×480	30
flower_garden2	720×480	30
baseball	1280×720	60
big_mamams_house2	720×480	24

Table 4.6: *Test sequences list*

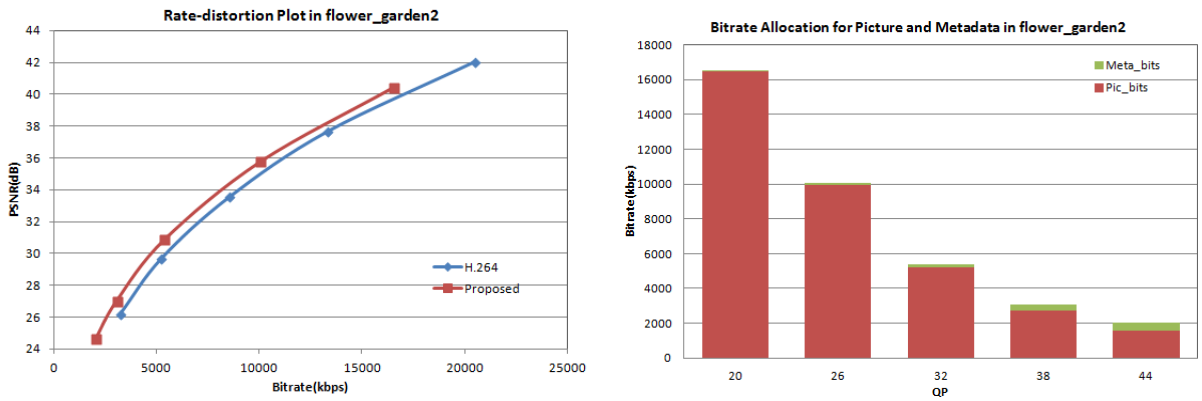
Sequence	**Bit rate saving(%)**	**PSNR gain (dB)**
big_mamams_house2	1.85%	0.144
baseball	1.92%	0.153
opening_ceremony	3.45%	0.227
american_pie	4.93%	0.331
independence_day	5.15%	0.412
flower_garden2	8.57%	0.624
Average	4.31%	0.315

Table 4.7: *Bit rate saving and PSNR gain for test sequences*

H.264/AVC standard are used in the coding process. PSNR of luminance component is still calculated between original and reconstructed picture, which is the picture after missing data recovery process at the decoder. Bit rate is derived from bits of bistream, which contains the bits of encoded picture and metadata.

The comparison between two coding schemes is usually represented by the rate-distortion (RD) plot at different QP. Figure 4.12 (a) shows the RD plot in flower_garden2. The proposed scheme is better than H.264/AVC in terms of RD cost, since its RD curve is above H.264's curve. The bits allocation for encoded picture and metadata is shown in Figure 4.12 (b). The percentage of metadata in bitstream increases with QP because more blocks are pruned at larger QP. Although the RD plot describes the coding efficiency improvement visually, it should be represented by quantity. Then the bit rate saving and PSNR gain are calculated with the method in [78] from the RD plot. The results are listed in Table 4.7.

The average bit rate saving is 4.31% and PSNR gain is $0.315dB$. In *independence_day* and *flower_garden2*, the bit rate saving is quite big, because there are plenty of texture



(a) *RD plot* **(b)** *Bits allocation*

Figure 4.12: *RD plot and bits allocation in a test sequence*

(a) *Original picture* **(b)** *Best-match picture*

Figure 4.13: *A picture composed of the best similar patch from original picture*

regions and thus a large number of similar patches are available in the patch library. This characteristic provides a good estimation of pruned blocks with the inpainting technique in the data recovery at the decoder. There are some gain in the other test sequences, since similar textures and structures also exist. Therefore, the proposed EDP video compression scheme achieves a better coding efficiency for intra frame on test sequences.

In patch library generation process, a picture composed of the best similar patch of each non-overlapping block is obtained and will assist the pruning decision module. An example from *big_mamams_house2* is illustrated in Figure 4.13.

The visual quality of reconstructed pictures of our EDP scheme is acceptable. An example from *flower_garden2* at QP 26 is shown in Figure 4.14. The pruned percentage is 30.1%.

(a) *Pruned picture at encoder side* **(b)** *reconstructed picture at decoder side*

Figure 4.14: *An example of pruned and reconstructed picture from* flower_garden2 *at QP 26*

(a) *Pruned picture at encoder side* **(b)** *reconstructed picture at decoder side*

Figure 4.15: *An example of pruned and reconstructed picture from* american_pie *at QP 32*

Another example from *american_pie* is shown in Figure 4.15. The pruned percentage is 15.2% and it is encoded at QP 32.

4.4.2 Complexity Analysis

The proposed EDP scheme achieves a video coding efficiency improvement by applying the pre- and post- processing techniques, but it will add complexity burdens. However, with the development of hardware and parallel processing, this problem will be reduced by some degrees. In the current implementation, it is using C++ and Matlab programming language. The most complex part is generating the patch library at the encoder and decoder. The fast KNN algorithm is utilized and the executable time is not high. For example, there

are about 86400 patches for an image with size 720×480. It took $2.1min$ to group them into $K = 45$ clusters. The searching of the most similar patch is also very efficient in the proposed approach. It took $8.5min$ to find the most similar patch of all the non-overlapping blocks in this image. The pruning decision process usually took $10s$ to process all the blocks when a synthesized image composed by the most similar patches is known. In the missing data recovery, the Bregman iteration will solve the ℓ^1 norm regularization problem in an efficient way. It usually took $6s$ to obtain an optimized coefficients with 25 candidate similar patches.

4.5 Summary

In this chapter, an exemplar-based data pruning video compression scheme for intra frame is proposed. The idea of applying the pre-processing and post-processing techniques to video coders is practical and experiments are convincing. This EDP scheme avoids modifying codec structures but still achieves the coding efficiency improvement. A patch library is generated for pruning decision at the encoder and for recovering pruned blocks at the decoder. The similarity of patches in natural images serves as a foundation of this EDP scheme in order to utilize the non-local information. Metadata containing the pruned information is also sent to decoder to assist the inpainting process. The location of pruned blocks as well as pixel intensities in non-pruned blocks are both encoded.

In missing data recovery, the pruned block is inferred from similar patches in the decoded picture. The sparseness of patches is served as a non-parametric image prior to solve the ℓ^1 norm regularization problem. The exemplar-based inpainting method demonstrates a remarkable reconstruction performance. In conclusion, the proposed EDP scheme is a good attempt to better deal with the texture regions when a large amount of similar patches exist in a natural image with processing techniques.

Chapter 5

Conclusions

In this dissertation, three problems in non-blind deconvolution, augmented reality and video compression are solved from Bayesian perspectives.

5.1 Non-blind Deconvolution

In the first part of dissertation, a data-driven non-blind deconvolution algorithm is proposed. Similar as other deconvolution algorithms, a MAP problem is established to estimate the latent image. In the proposed MAP model, a noise model with mixed order derivative is used to represent the spatial randomness of noises. The image prior expressed by the GSM model is estimated from the related clear images. By separating the convolution operations with other terms, this MAP problem is solved in an efficient way. Remarkable experiments demonstrate that this algorithm will recover more details and suppress ringing effects better than conventional Richardson-Lucy and state-of-art Shan's algorithm.

The proposed non-blind deconvolution algorithm is based on simple assumptions such as linear blur model and spatial invariant blur. In the future, the algorithm should consider more scenarios.

- **Big blur kernel**

  Although this algorithm works well for the small blur case, the deblurring with big blur is not good. How to better utilize the related clear images is an interesting problem in the future. The disparity of blurry image and related images should provide more

(a) *Spatially variant blur kernel [79]* **(b)** *Nonlinear blur model [20]*

Figure 5.1: *Complex blurry images in the real world*

information during the iterations of latent image reconstruction.

- **Boundary problem**

  The boundary ringing effects come from the convolution operation in Fourier domain. The Matlab command "edgetaper" is used in current implementation to smooth the boundary. How to make a good balance between ringing suppression and maintaining boundary will be an interesting topic.

- **Spatially variant blur**

  In the proposed algorithm, the blur kernel is assumed as shift-invariant for the whole image. However, sometimes it's not true in the real world. Figure 5.1 (a) is an example that blur kernel is variant in different regions. How to deal with spatially variant blur is also a challenging problem.

- **Nonlinear blur model**

  Figure 5.1 (b) shows a snow scene captured at night. If the linear blur model $B = L \otimes f + n$ is used, the deblurring result is not good, because some pixels are obtained from a nonlinear blur model due to limited dynamic range of camera sensors. How to modify the blur model will be one of my future tasks.

The proposed non-blind deconvolution algorithm is a parametric-based approach, because the image prior and likelihood are represented by certain distributions. However, the latest development of non-parametric approaches in image restoration shows that they are more

(a) *occluded objects* **(b)** *several buildings*

Figure 5.2: *Complex cases in the real world*

suitable for solving complex inverse problems. A deblurring method on face images in [80] shed some light for non-parametric approaches in deblurring, although typical face images are quite different from natural images.

5.2 Augmented Reality

In a VCI system, where, when and how to insert the content so as not to disturb the viewers are challenging issues. In the second part, a novel and generic VCI system is introduced to attack the problem of inserting an image or a movie in a video stream without knowledge of cameras. To achieve a seamless virtual to real blending, a closed-loop detection and tracking combined strategy is proposed to implement insertion in an efficient way. The inserted positions are estimated in a Bayesian framework with the standard court model or 3D geometry model as the image prior. Moreover, motion stabilization, dynamic registration and color harmonization are used to reduce the disturbance of inserted content.

The proposed VCI system in tennis video streams achieves a remarkable and robust performance. It works on four typical tennis courts such as US open, Australia open, French open and Wimbledon. However, the automatic VCI system is not a perfect solution. Its performance is largely relied on the accuracy of line detection, model fitting, camera calibration and motion stabilization.

However, inserting an image or a movie in street view video streams is more challenging, because of no standard models for building scenes. Since visual tracking on buildings which are composed of repeated patterns and structures will obtain a good estimation of camera motion, the detection and tracking combined strategy is a reasonable method.

However, the proposed VCI system only works for simple cases if the building facade is obvious and easy to detect. More scenarios should be considered when dealing with complex cases. For example, how to deal with the occluded objects such as the cars, trees and people nearby the buildings is an interesting problem, as shown in Figure 5.2 (a). If there occur several buildings as shown in Figure 5.2 (b), how to choose a building as an appropriate region for insertion needs the high level pattern recognition knowledge. In addition, what kind of content should be chosen is also an interesting open question and has the commercial values.

5.3 Video Compression

In the third part of this dissertation, an exemplar-based data pruning video compression scheme for intra frame is proposed. The idea of applying the pre- and post- processing techniques in video compression is practical. The experiments shows the convincing results in terms of bit-rate reduction. This EDP scheme avoids modifying codec structures but still achieves a coding efficiency improvement. A patch library is generated for pruning decision at the encoder and for recovering pruned blocks at the decoder. The similarity of patches in natural images serves as a foundation of this EDP scheme in order to utilize the non-local information. Metadata containing the pruned information is also sent to decoder to assist the inpainting process. The location of pruned blocks as well as pixel intensities in non-pruned blocks are both encoded.

In the missing data recovery, a ℓ^1 norm regularization problem is established to utilize the sparseness of patches. The novel Bregman iteration algorithm is adopted to solve this constrained problem in an accurate and efficient way. This non-parametric approach obtains a good reconstruction of pruned blocks. Therefore, the proposed EDP scheme is a good attempt to better deal with the texture regions when a large amount of similar patches exist

in a natural image with processing techniques.

In current implementation, the pruning decision makes a similar prediction as the one in intra mode of H.264/AVC. Although this scheme is independent from encoder and decoder, it will be an important future task to confirm the effectiveness and efficiency of this scheme in other video coders such as the new HEVC. The size of patches and the compression of metadata are also interesting problems.

In the missing data recovery, a constrained optimization model is established to solve the sparse linear combination coefficients. Two constraints are introduced to represent that the neighboring pixels of target should be known values and normalization requirement. However, there should be some other constraints about local smoothness. In order to make the estimated missing data consistent with neighboring patches, a local smoothness constraint may be needed. It will be a very challenging task to describe this constraint and solve a more difficult optimization problem.

Bibliography

[1] D.Farin, S.Krabbe, P.de With, and W.Effelsberg, "Robust Camera Calibration for Sport Videos using Court Models," in *Proc. of SPIE: Storage and Retrieval Methods and Applications for Multimedia*, 2004, pp. 80–91.

[2] J.P.Tardif, "Non-Iterative Approach for Fast and Accurate Vanishing Point Detection," in *ICCV*, 2009, pp. 1250–1257.

[3] T.Bayes and R.Price, "An Essay towards solving a Problem in the Doctrine of Chance. By the late Rev. Mr. Bayes, communicated by Mr.Price in a letter to J.Canton," *Philosophical Transactions of the Royal Society of London*, vol. 53, no. 0, pp. 370–418, 1763.

[4] T.S.Cho, A.Levin, F.Durand, and W.T.Freeman, "Motion Blur Removal with Orthogonal Parabolic Exposures," in *IEEE Intl. Conf. Computational Photography(ICCP)*, Cambridge, USA, Mar 2010, pp. 1–8.

[5] A.Levin, R.Fergus, F.Durand, and W.T.Freeman, "Image and Depth from a Conventional Camera with a Coded Aperture," *ACM Trans. Graph*, vol. 26, no. 3, pp. 70:1–70:9, 2007.

[6] N.Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, New York Wiley, 1942.

[7] R.E.Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[8] L.Lucy, "An Iterative Technique for the Rectification of Observed Distributions," *Astronomical Journal*, vol. 79, no. 6, pp. 745–754, 1974.

[9] I.Rudin, S.Osher, and E.Fatemi, "Nonlinear Total Variation Based Noise Removal Algorithms," *Physica. D*, pp. 259–268, 1992.

[10] N.Joshi, C.Zitnick, R.Szeliski, and D.J.Kriegman, "Image Deblurring and Denoising using Color Priors," in *CVPR*, Miami, USA, June 2009, pp. 1550–1557.

[11] L.Yuan, J.Sun, L.Quan, and H.Y Shum, "Progressive Inter-scale and Intra-scale Non-blind Image Deconvolution," *ACM Trans. Graph*, vol. 27, no. 3, pp. 74:1–74:10, 2008.

[12] R.Fergus, B.Singh, A.Hertzmann, S.T.Roweris, and W.T.Freeman, "Removing Camera Shake from a Single Photograph," *ACM Trans. Graph*, vol. 25, no. 3, pp. 787–794, 2006.

[13] Q.Shan, J.Jia, and A.Agarwala, "High-quality Motion Deblurring from a Single Image," *ACM Trans. Graph*, vol. 27, no. 3, pp. 73:1–73:10, 2008.

[14] A.Levin, Y.Weiss, F.Durand, and W.T.Freeman, "Understanding and Evaluating Blind Deconvolution Algorithms," in *CVPR*, Miami, USA, June 2009, pp. 1–8.

[15] T.S.Cho, N.Joshi, C.L.Zitnick, S.B.Kang, R.Szeliski, and W.T.Freeman, "A Content-aware Image Prior," in *CVPR*, San Francisco, USA, June 2010, pp. 169–176.

[16] A.Levin and Y.Weiss, "Blind Motion Deblurring Using Image Statistics," in *NIPS*, Dec 2006, pp. 1–8.

[17] S.Dai and Y.Wu, "Motion From Blur," in *CVPR*, Anchorage, USA, June 2008, pp. 1–8.

[18] S.Dai and Y.Wu, "Removing Partial Blur in A Single Image," in *CVPR*, Miami, USA, June 2009, pp. 2544–2551.

[19] S.Harmeling, S.Sra, M.Hirsch, and B.Scholkopf, "Multiframe Blind Deconvolution, Super-resolution, and Saturation Correction via Incremental EM," in *IEEE Intl. Conf. Image Processing(ICIP)*, Hongkong, China, Sep 2010, pp. 3313–3316.

[20] S.Cho, J.Wang, and S.Lee, "Handling Outliers in Non-Blind Image Deconvolution," in *ICCV*, 2011, pp. 1–8.

[21] H.Liu, S.Jiang, Q.Huang, and C.Xu, "A Generic Virtual Content Insertion System based on Visual Attention Analysis," in *ACM Multimedia*, 2008, pp. 378–388.

[22] H.Shah and S.Chaudhuri, "Automated Billboard Insertion in Video," in *ACCV*, 2007, pp. 240–250.

[23] K.Wan, X.Yan, X.Yu, and C.Xu, "Robust Goal-mouth Detection for Virtual Content Insertion," in *ACM Multimedia*, 2003, pp. 468–469.

[24] C.Chang, K.Hsieh, M.Chiang, and J.Wu, "Virtual Spotlighted Advertising for Tennis Videos," *J. of Visual Communication and Image Representation*, vol. 21, no. 7, pp. 595–612, 2010.

[25] Y.Kim, K.Lee, K.Choi, and S.cho, "Building Recognition for Augmented Reality Based Navigation System," in *IEEE Int. Conf. Computer and Information technology*, 2006, pp. 131–131.

[26] B.S.Alvarez, P.Carvalho, and M.Gattass, "Insertion of Three-dimensional Objects in Architectural Photos," *Journal of WSCG*, vol. 10, no. 1, pp. 17–24, 2002.

[27] Z.Zhang, X.Liang, A.Ganesh, and Y.Ma, "TILT:Transform Invariant Low-rank Textures," in *ACCV*, 2010, pp. 314–328.

[28] W.Zhang and J.Kosecka, "Extraction Matching and Pose Recovery based on Dominant Rectangular Structures," in *ICCV*, 2003, pp. 83–91.

[29] D.Shaw and N.Barnes, "Perspective Rectangle Detection," in *ECCV*, 2006, pp. 119–127.

[30] G.J.Sullivan and J.Ohm, "Recent Developments in Standardization of High Efficiency Video Coding(HEVC)," *SPIE Applications of Digital Image Processing XXXIII*, vol. 7798, no. 30, Aug 2010.

[31] K.McCann, B.Bross, S.Sekiguchi, and W.Han, *High Efficiency Video Coding(HEVC) Test Model 2 (HM2) Encoder Description*, JCTVC, Guangzhou, China, Oct 2010.

[32] A.Dumitras and B.G.Haskell, "A Texture Replacement Method at the Encoder for Bit Rate Reduction of Compressed Video," *IEEE Trans. Circuits and Systems for Video Tech*, vol. 13, no. 2, pp. 163–175, Feb. 2003.

[33] A.Dumitras and B.G.Haskell, "An Encoder-decoder Texture Replacement Method with Application to Content-based Movie Coding," *IEEE Trans. Circuits and Systems for Video Tech*, vol. 14, no. 6, pp. 825–840, Jun. 2004.

[34] C.Zhu, X.Sun, F.Wu, and H.Li, "Video Coding with Spatio-Temporal Texture Synthesis," in *IEEE Intl. Conf. Multimedia and Expo(ICME)*, Beijing, China, Jul 2007, pp. 112–115.

[35] C.Zhu, X.Sun, F.Wu, and H.Li, "Video Coding with Spatio-Temporal Texture Synthesis and Edge-based Inpainting," in *IEEE Intl. Conf. Multimedia and Expo(ICME)*, Jun 2008, pp. 813–816.

[36] D.Liu, X.Sun, F.Wu, S.Li, and Y.Q.Zhang, "Image Compression with Edge-based Inpainting," *IEEE Trans. Circuits and Systems for Video Tech*, vol. 17, no. 10, pp. 648–665, 2007.

[37] D.Liu, X.Sun, and F.Wu, "Inpainting with Image Patches for Compression," *J. of Visual Communication and Image Representation*, vol. 23, pp. 100–113, 2012.

[38] D.T.Vo, J.Sole, P.Yin, G.Gomila, and T.Q.Nguyen, "Data Pruning-Based Compression Using High-order Edge-directed Interpolation," in *IEEE Intl. Conf. Acoustics and Speech and Signal Processing(ICASSP)*, Apr 2009, pp. 997–1000.

[39] D.T.Vo, J.Sole, P.Yin, G.Gomila, and T.Q.Nguyen, "Slective Data Pruning-Based Compression Using High-order Edge-directed Interpolation," *IEEE Trans. Image Processing*, vol. 19, no. 2, pp. 399–409, Feb 2010.

[40] D.Zhang, S.Bhagavathy, and J.Llach, "Method and Apparatus for Data Pruning for Video Compression using Example-based Super-resolution," patent PU100014, Technicolor Research Center at Princeton NJ, 2009.

[41] D.Zhang, S.Bhagavathy, and J.Llach, "Method and Apparatus for Block-based Mixed-resolution Data Pruning for Improving Video Compression Efficiency," patent, Technicolor Research Center at Princeton NJ, 2010.

[42] W.Freeman, T.R.Jones, and E.C.Pasztor, "Example-Based Super-Resolution," *IEEE Computer Graphics and Application*, pp. 56–65, Mar/April 2002.

[43] W.Yin, S.Osher, D.Goldfarb, and J.Darbon, "Bregman Iterative Algorithm for l^1 Minimization with Applications to Compressed Sensing," *SIAM J. Imaging Sciences*, vol. 1, no. 1, pp. 143–168, 2008.

[44] Y.Huang, Q.Hao, and H.Yu, "Virtual Ads Insertion in Street Building Views for Agumented Reality," in *IEEE Intl. Conf. Image Processing(ICIP)*, Belgium, Sep 2011, pp. 1117–1120.

[45] Q.Hao, Y.Huang, and H.Yu, "Method and Apparatus for Video Abstraction," patent 13/340883, Huawei US Technologies, 2010.

[46] Q.Hao, D.Zhang, S.Bhagavathy, and J.Llach, "Method and Apparatus for Pruning Strategy in Example-based Data Pruning Compression," patent PU100197, Technicolor Research Center at Princeton NJ, 2010.

[47] S.Roth and M.J.Black, "Fields of Experts: A Framework for Learning Image Priors," in *CVPR*, 2005, pp. 860–867.

[48] M.F.Tappen, B.C.Russell, and W.T.Freeman, "Exploiting the Sparse Derivative Prior for Super-Resolution and Image Demosaicing," in *ICCV*, 2003, pp. 673–678.

[49] S.Dai, M.Han, W.Xu, Y.Wu, and Y.Gong, "Soft Edge Smoothness Prior for Alpha Channel Super Resolution," in *CVPR*, June 2007, pp. 1–8.

[50] E.P.Simoncelli, "Statistical Models for Images:Compression restoration and synthesis," in *Proc Asilomar Conference on Signals, Systems and Computeres*, 1997, pp. 673–678.

[51] A.Levin, A.Zomet, and Y.Weiss, "Leaning How to Inpaint from Global Image Statistics," in *ICCV*, 2003, pp. 305–312.

[52] N.E.Apostoloff and A.W.Fitzgibbon, "Bayesian Video Matting using Learnt Image Priors," in *CVPR*, 2004, pp. 407–414.

[53] Y.Weiss and W.T.Freeman, "What Makes A Good Model of Natural Images?," in *CVPR*, Minneapolis, USA, June 2007, pp. 1–8.

[54] A.P.Dempster, N.M.Laird, and D.B.Rubin, "Maximum Likelihood From Incomplete Data Via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[55] R.N.Bracewell, *The Fourier Transform and Its Applications*, McGraw-Hill, 1988.

[56] Z.Wang, A.C.Bovik, H.R.Sheikh, and E.P.Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[57] T.Mei and X.Hua, "Contextual Internet Multimedia Advertising," *Proc. of the IEEE*, vol. 98, no. 8, pp. 1416–1433, Aug. 2010.

[58] G.Papagiannakis, G.Singh, and N.M.Thalmann, "A Survey of Mobile and Wireless Technologies for Augmented Reality Systems," *Computer Animation and Virtual Worlds*, vol. 19, no. 1, pp. 3–22, Feb 2008.

[59] Unkown writer, "Top 5 smartphone augmented relaity apps," `http://www.phonesreview.co.uk/2012/01/19/top-5-smartphone-augmented-reality-apps/`, Jan. 2012.

[60] R.Hartley and A.Zisserman, *Multiple view Geometry in Computer Vision*, Cambridge University Press, 2003.

[61] S.Beauchemin and J.Barron, "The Computation of Optical Flow," *ACM Computing Surveys*, vol. 27, no. 3, pp. 143–168, Sep. 1995.

[62] J.Shi and C.Tomasi, "Good Features to Track," in *CVPR*, 1994, pp. 593–600.

[63] M.A.Fischler and R.C.Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. of the ACM* , vol. 24, pp. 381–395, 1981.

[64] W.H.Press, B.P.Flannery, S.A.Teukolsky, and W.T.Vetterling, *Numerical recipes in C*, Cambridge University Press, 1988.

[65] X.Li, "Video Processing Via Implicit and Mixture Motion Models," *IEEE Trans. Circuits and Systems for Video Tech*, vol. 17, no. 8, pp. 953–963, Aug 2007.

[66] T.Wiegand, G.J.Sullivan, G.Bjontegaard, and A.Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits and Systems for Video Tech*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[67] G.J.Sullivan, P.Topiwala, and A.Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," in *SPIE conference on Applications of Digital Image Processing XXVII*, 2004, pp. 454–474.

[68] D. Shakhnarovish and Indyk, *Nearest-Neighbor Methods in Learning and Vision*, MIT Press, 2005.

[69] M.Bertalmio, G.Sapiro, V.Caselles, and C.Ballester, "Image Inpainting," in *Proc. SIGGRAPH*, 2000, pp. 417–424.

[70] T.Chan and J.Shen, "Local Inpainting Models and TV Inpainting," *SIAM J.Appl.Math*, vol. 62, no. 3, pp. 1019–1043, 2001.

[71] S.Roth and M.J.Black, "Steerable Random Fields," in *CVPR*, 2007, pp. 1–8.

[72] Z.Xu and J.Sun, "Image Inpainting by Patch Propagation Using Patch Sparsity," *IEEE Trans. Image Processing*, vol. 19, no. 5, pp. 1153–1165, May. 2010.

[73] S.Mallat and Z.Zhang, "Matching Pursuit in a Time Frequency Dictionary," *IEEE Trans. Signal Process*, vol. 41, pp. 3397–3415, 1993.

[74] S.S.Chen, D.L.Donoho, and M.A.Saunders, "Atomic Decomposition by Basis Pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.

[75] R.Tibshirani, "Regression Shrinkge and Selection Via the Lasso," *J.Roy.Statist.Soc.B.*, vol. 58, no. 1, pp. 267–288, 1996.

[76] S.Osher, Y.Mao, B.Dong, and W.Yin, "Fast Linearized Bregman Iteration for Compressive Sensing and Sparse Denoising," *Commun. Math. Sci*, vol. 8, no. 1, pp. 93–111, 2010.

[77] J.Cai, S.Osher, and Z.Shen, "Fast Linearized Bregman Iteration for Compressed Sensing and Sparse Denoising," in *2008. UCLA CAM Reprots*, 2008, pp. 8–37.

[78] G.Bjontegaard, *Calculation of average PSNR differences between RD-curves, VCEG-M33*, JCTVC, Austin, USA, April 2001.

[79] A.Gupta, N.Joshi, L.Zitnick, M.Cohen, and B.Curless, "Single Image Deblurring Using Motion Density Functions," in *ECCV*, 2010.

[80] M.Nishiyama, A.Hadid, H.Takeshima, J.Shotton, T.Kozakaya, and O.Yamaguchi, "Facial Deblur Inference Using Subspace Analysis for Recognition of Blurred Faces," *PAMI*, vol. 33, no. 4, pp. 838–845, 2011.