The analysis, detection and correction of flicker and stain artifact in degraded motion pictures

Wong, Kin Keong

2008

Wong, K. K. (2008). The analysis, detection and correction of flicker and stain artifact in degraded motion pictures. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/41195>

<https://doi.org/10.32657/10356/41195>

THE ANALYSIS, DETECTION & CORRECTION OF FLICKER AND STAIN ARTIFACTS IN DEGRADED MOTION PICTURES

**Submitted in Fulfillment of the Requirement for the degree of
Doctor of Philosophy**

By

Wong Kin Keong

**School of Computer Engineering
Nanyang Technological University**

February 2008

Acknowledgements

When I have finally arrived at the last section of this thesis that I have been writing, my feeling is mixed.

Like a sailor who has set sail for a solo seafaring adventure and finally reached the destination, there is only one word that can best describe how I feel at this very moment: RELIEVED!

Over the years, metaphorically, I have traversed over immense spans of water spaces and sailed through many treacherous waters, at times with ocean waves rising as high as mountains. I have also sailed into numerous stormy weathers, with days as dark as nights and the sun was no where to be found to give me directions. On the more smooth-sailing part of my journey, I have visited bays of breathtaking scenery, seen a kaleidoscope of unnamed faunas and floras, and have also collected from different shores pebbles of extreme beauty. I am grateful to have gone on this wonderful trip, and it gives me great pleasure and satisfaction that I have the chance to take back some of these beautiful pebbles to share with you.

These pebbles are actually the collection of all the flicker correction and stain removal algorithms that have come out from the work of this thesis, which I feel is most aptly likened to a seafaring adventure, with all the elements of unknown throughout. Now is the time for me to say thank you to a number of people, who have in one way or another, helped made my trip possible.

I would like first to thank Dr. M.N. Chong, the person who took me to a boat one day many years ago, pointed his finger to the far horizon and told me to set sail into the then unknown world of “motion picture restoration”. Despite the rough voyage, I am still grateful to him for opening up a new world to me.

The next person to thank is my supervisor Dr. Amitabha Das, who has trusted me so much that he has practically given me absolute freedom in exploring and developing my sailing skill. It is always a joy talking to him, from the Indian cast system to views by Noam Chomsky, it never gets dull.

The support from my family has been crucial in seeing me through the whole journey; their understanding and tireless patience with me are something that I will always remember and feel ever grateful.

Special mention should also go to my good friend W.Y. Cheung, whose encouraging words have helped me stay firmly on course and keep my spirit high.

At this juncture, I would also like to take this opportunity to pay tribute to Zheng He, the great Chinese admiral who has led seven great expeditions to more than 30 countries between 1405 and 1433 during the Ming Dynasty. The remarkable story of his great explorations has fascinated me since I was a boy, and I believe the story must have also somehow unknowingly, but surely, planted in me the seed of the spirit of discovery. So I would attribute my discovery in the work of this thesis to this very origin which is linked to this great Chinese mariner, and I shall pay my tribute by attaching a picture of each of his “treasure ships” at the beginning of every chapter in this thesis.

Finally I would like to dedicate this thesis to my late father who had wished to (but could not) write the following Chinese couplet* to honor our family’s scholastic tradition:

一門三博士
滿堂皆學人

I hope his wish could finally be fulfilled this time...

* A literary form of Chinese writing consisting of a pair of lines of poetry that are usually rhymed.

Table of Contents

Abstract	xi
Glossary	xiii
1 Introduction	1
1.1 Background	1
1.1.1 Anatomy of a Typical Motion Restoration System	3
1.2 Scope of the project	5
1.3 Thesis outline	5
1.4 Novel contributions	7
2 Flicker and its correction: A review	9
2.1 Introduction	9
2.2 Flicker – The basics	9
2.3 Flicker classifications	10
2.3.1 Global flicker and local flicker	10
2.3.2 Intensity flicker and color flicker	11
2.4 Flicker correction – an overview	12
2.4.1 Flicker correction and its related problems	12
2.4.2 Criteria for evaluating flicker correction results	13
2.4.2.1 Smoothness of the intensity mean curve and intensity variance curve	14
2.4.2.2 PSNR (peak signal-to-noise ratio)	15
2.4.2.3 Coding efficiency	17
2.4.2.4 Subjective evaluation	18
2.4.3 Motion picture sequences used for testing flicker correction algorithms	20

2.5	Overview of existing flicker correction methods	23
2.5.1	Roosmalen's flicker parameter based algorithm and related work	24
2.5.2	Ohuchi's hierarchical flicker correction model and related work	24
2.5.3	Naranjo's non-linear flicker model and related work	25
2.6	Conclusion	26
3	Global flicker correction	28
3.1	Introduction	28
3.2	The histogram mapping method	28
3.2.1	Histogram mapping	28
3.2.2	Using histogram mapping to correct flicker	29
3.2.3	Experimental results and analysis	30
3.2.4	Discussions	36
3.3	The intensity mean averaging method	38
3.3.1	Frame averaging – the basics	38
3.3.2	Intensity mean averaging	39
3.3.3	The intensity mean averaging method – a few issues	40
3.3.3.1	Flicker frame detection	40
3.3.3.2	Problem of consecutive flicker frames	41
3.3.4	Experimental results and analysis	42
3.3.5	Discussions	46
3.4	Conclusion	47
4	Local flicker correction	49
4.1	Introduction	49
4.2	Local enhancement – the basics	49
4.3	Histogram mapping with local enhancement	50
4.3.1	Experimental results and analysis	51
4.4	Intensity mean averaging with local enhancement	57
4.4.1	Experimental results and analysis	57
4.4.2	Discussions	61
4.5	Conclusion	62

5	Motion-compensated local flicker correction	64
5.1	Introduction	64
5.2	Motion estimation and compensation – the basics	64
5.2.1	Image sequence modeling and the occlusion problem	65
5.3	Block-based motion estimation	67
5.3.1	Block Matching (BM) method	67
5.3.2	Overlapped Block Matching (OBM) method	68
5.3.3	Multi-resolution Block Matching (MRBM) method	69
5.4	Motion-compensated intensity mean averaging with local enhancement	70
5.4.1	Motion detection using the block-matching method	71
5.4.1.1	Motion detection	71
5.4.1.2	Effect of flicker on motion detection	75
5.4.2	Motion compensation	76
5.4.2.1	Iterative interpolation	77
5.4.3	Experimental results and analysis	78
5.5	Conclusion	80
6	Analysis and detection of global and local flicker artifacts using flicker parameter estimation	83
6.1	Introduction	83
6.2	The flicker parameters estimation (FPE) method	83
6.2.1	Flicker parameters model	84
6.2.2	Flicker parameters estimation	85
6.2.3	Motion block detection and flicker parameters estimation	88
6.2.4	Intensity correction	88
6.2.5	Synthetic test sequence preparation	89
6.2.5.1	Global flicker generation	89
6.2.5.2	Local flicker generation	91
6.2.6	Experimental results and analysis	92
6.2.6.1	Synthetic test sequence	92
6.2.6.1.1	Global flicker	92
6.2.6.1.2	Local flicker	95
6.2.6.2	Natural test sequence	96
6.2.6.3	Discussions	98

6.3	Least square minimization (LSM) method	100
6.3.1	Least square minimization	100
6.3.2	Motion block detection	102
6.3.3	Intensity correction	103
6.3.4	Experimental results and analysis	103
6.3.4.1	Synthetic test sequence	104
6.3.4.1.1	Global flicker	104
6.3.4.1.2	Local flicker	106
6.3.4.2	Natural test sequence	108
6.4	FPE method with stationary block re-examination incorporated	110
6.4.1	Stationary block re-examination	111
6.4.2	Experimental results and analysis	111
6.4.2.1	Synthetic test sequence	112
6.4.2.1.1	Global flicker	112
6.4.2.1.2	Local flicker	113
6.4.2.2	Natural test sequence	115
6.4.2.3	Discussions	117
6.5	Conclusion	119
7	Motion-compensated FPE algorithm	120
7.1	Introduction	120
7.2	Applying motion compensation to enhance flicker correction	120
7.2.1	The FPE method and its inherent limitations	120
7.2.2	Motion vector compensation	122
7.2.2.1	Determining motion vectors	123
7.2.2.2	Motion vector determination in a flicker environment	124
7.2.3	Experimental results and analysis	126
7.2.3.1	Synthetic test sequence	126
7.2.3.1.1	Global flicker	126
7.2.3.1.2	Local flicker	128
7.2.3.2	Natural test sequence	129
7.3	Conclusion	132

8	Simplified motion-compensated FPE algorithm	133
8.1	Introduction	133
8.2	Inherent problems of the flicker model used in the FPE method	133
8.3	A simplified flicker model	134
8.4	Simplified alpha estimation algorithm	135
8.4.1	Determining the alpha parameter	135
8.4.2	Checking value of the alpha parameter	136
8.4.3	Smoothing the alpha value	139
8.4.4	Restoring image using the alpha parameter	141
8.5	Experimental results and analysis	141
8.5.1	Synthetic test sequence	141
8.5.1.1	Global flicker	141
8.5.1.2	Local flicker	142
8.5.2	Natural test sequence	143
8.5.3	Discussions	146
8.6	Conclusion	147
9	Flicker correction in color motion pictures	149
9.1	Introduction	149
9.2	Color space – the fundamentals	149
9.2.1	The RGB color model	150
9.2.2	The YUV color model	153
9.2.3	The CMY color model	156
9.3	Flicker in color motion pictures	156
9.3.1	Intensity flicker	156
9.3.2	Color flicker	158
9.4	Correcting flicker in color motion pictures	158
9.5	Experimental results and analysis	160
9.5.1	Synthetic test sequence	160
9.5.2	Natural test sequence	165
9.5.3	Discussions	170
9.6	Conclusion	171

10	Stain removal	172
10.1	Introduction	172
10.2	Stain problem in color motion pictures	172
10.3	A method for stain removal	173
10.3.1	Stain model	174
10.3.2	Stain detection	175
10.3.3	Stain removal	177
10.4	Experimental results and analysis	178
10.4.1	Natural stain test sequence	178
10.4.1.1	Correction results based on an imperfect reference frame	179
10.4.1.1.1	Discussions	182
10.4.1.2	Correction results based on a perfect reference frame and a better threshold window	184
10.4.2	Synthetic stain test sequence	188
10.5	Conclusion	191
11	Conclusion and Future Work	192
11.1	Conclusion	192
11.2	Recommendation for future work	197
	Epilogue	200
	References	201
Appendix A	Least Square Minimization (LSM) method	212
Appendix B	Proof of expression used in Intensity Mean Averaging	218
Appendix C	An Improved Low-pass filtering Algorithm	219
Appendix D	Flicker Correction Results Comparison	223

Appendix E	Organizational Chart of Image Files	225
Appendix F	List of Publications	228

Abstract

This dissertation presents algorithms associated with the development of a robust automatic motion restoration system, in particular, in the areas of flicker correction and stain removal.

Motion pictures are records of images on the historic, artistic, and cultural developments of the human society of the 20th century. Just like a historical relic, it is a heritage to all mankind. Unfortunately, due to the aging of film material and other factors, many of these historically significant items are now in a fragile state and need to be “rescued” immediately, and restoration is the very first step in this rescue operation.

Restoration is about removing all the defects that have crept into the motion picture and restore it back to its original state. The common defects are dirt and sparkle, flicker, line scratches, and stain etc; these are all very visible to the human eyes and so their presence will bring much displeasure to one’s viewing experience. However, restoration is a very tedious and time-consuming process and one major contributing factor is the massive number of frames that have to be corrected. The situation is made worse when the frame sequence suffers from more than one type of defects, which is not uncommon and this would increase the work load by a few folds. Past experience with manual correction has shown that it is both prohibitively expensive and slow; facing such a reality, an automated restoration system becomes not just an alternative, but very much a necessity.

Motion picture restoration has been blessed in that it is particularly suitable for automation. The reason is that due to the high frame rate, there exists a large amount of temporal redundancy between successive frames in a frame sequence. The implication of this is that a current frame can be corrected so long as a preceding good frame is available and motion between the two frames can be estimated. Once the current frame is corrected,

it can then be used for the correction of the next frame, and the process will just repeat itself in this manner until the last frame of the sequence is reached. Such an iterative process is highly suited for programming implementation.

The direction of the research work in this thesis was set after extensive literature survey showed that not much work has been done in the area of flicker correction; and for stain, the work done was a complete blank.

A number of new flicker correction algorithms were produced as a result of the research work done in this thesis. Flicker correction algorithms developed before chapter 6 were original and were based on heuristic approaches. Chapter 6 and 7 have extended the work of a well-known current method and provided solutions to its associated problems. The flicker correction algorithm developed in chapter 8 was based on a new flicker model, and the work was later extended to correct color flicker, an area with no known previous work done. All the flicker correction algorithms have been thoroughly tested with both natural and synthetic flicker test sequences, and have been found to be effective but varied in their performances.

Stain is the last problem addressed in this thesis. A stain model has been proposed in chapter 10, and based on this model, a stain removal algorithm was developed. Just like for flicker, the stain-removal algorithm has been tested with both natural and synthetic stain test sequences, and is found to be very effective.

Glossary

2AFC	Two Alternatives Forced Choice
AURORA	Automatic Restoration of Original Film and Video Archives
BM	Block Matching
BRAVA	Broadcast Archives Restoration through Video Analysis
CDF	Cumulative Distribution Function
CMY	Cyan Magenta Yellow
DSCOS	Double Stimulus Continuous Quality Scale
DSIS	Double Stimulus Impairment Scale
DVD	Digital Video Disc
FPE	Flicker Parameter Estimation
HSI	Hue Saturation Intensity
HSV	Hue Saturation Value
IPTV	Internet Protocol TeleVision
ITU	International Telecommunication Union
LSM	Least Square Minimization
MAD	Mean Absolute Difference
MOS	Mean Opinion Score
MPEG	Motion Picture Experts Group
MPR	Motion Picture Restoration
MRBM	Multi-Resolution Block Matching
OBM	Overlapped Block Matching
PAL	Phase Alternating Line
PSNR	Peak Signal to Noise Ratio
RGB	Red Green Blue
SSCQE	Single Stimulus Continuous Quality Evaluation
VCD	Video Compact Disc

Chapter 1

Introduction

1.1 Background

The motion picture industry has a relatively short history; however, its emergence since slightly more than a century ago is by far one of the few major developments that has had a far-reaching social impact on the human society. With the birth of this new industry, entertainment has taken on an entirely new dimension in which continuous images that make up a story could be recorded and made into a “motion picture” for the entertainment of the general public. In a wider perspective, motion pictures are much more than just pure entertainment. They actually form testimony to the historic, artistic, and cultural developments of the human society in the last hundred over years, and can be quite aptly put together as the “collective memory” of the human society of the last century.

Over the years, throughout the world large stocks of motion pictures have been produced. They are typically stored in mediums like nitrate, acetate or celluloid based film and more recently, video tape in various formats. All these material unfortunately will degrade with time, and coupled with frequent usage and less-than-ideal storage conditions, many of these historically significant items are now in a fragile state. It is imperative that swift actions are taken to conserve and restore these materials, else such precious heritage to all mankind will be lost forever. The severity of this precarious situation has been highlighted in a survey released by James H. Billington, U.S. Librarian of Congress [1] in 1993. Some of the more important findings in that survey are:

- Less than 50% of all titles produced before 1950 survive in complete form. For productions before 1920 and 1910, the survival rate drops respectively to 20% and 10%.
- Acetate-based film, to which volatile nitrate-based films have been transferred and once considered “safe”, was found to suffer from a “vinegar syndrome” decaying problem which eventually renders the film unusable.
- Serious physical deterioration has been discovered in films produced within the last four decades.

The significance of the preservation of old motion pictures goes beyond the primary objectives of keeping intact our cultural past and important moments in our history. The commercial benefit derived from old film restoration is one aspect that cannot be underestimated either. Broadcasting stations all over the world need programs to fill up their air-time, and DVD producers are always on the lookout for new titles to put into their discs. In addition, HD-DVD, Blue-Ray DVD [60], IPTV [61], and digital cinema [73] will fuel the demand for even more content material. It is obvious that the huge collection of old motion pictures upon successful restoration will provide an economical source of program material which is much cheaper than creating new programs.

Preservation of old motion pictures can be achieved by copying them onto new digital media in a compressed format, such as the MPEG standard [42]. However it is important that old motion pictures are properly restored first before they are stored into digital media. This is because all old motion pictures typically suffer from various degrees of degradation, and they have to be restored to a reasonable standard first before they can be considered good enough for general viewing or commercial release. On the other hand, restoration generally leads to more efficient compression [62], and this will translate into cost saving in storage or broadcasting.

Restoration of old motion pictures is a highly labor-intensive and extremely costly undertaking. The first successfully restored motion picture was Disney’s 1937 masterpiece “Snow White and the Seven Dwarfs”. It was re-released in 1993 after massive restoration work was done on the original degraded film. The restoration work was carried out in Cinesite Digital Film Center, California, and it took sixty workstation operators working round the clock to churn out about two thousand frames daily [2].

The Disney restoration experience highlights two major problems in motion picture restoration: the prohibitive time and cost involved. The main contributing factor to these problems is the heavy reliance on human labor in the restoration process. To restore the large number of old film archives in a timely and cost-effective way, the obvious solution is to keep human labor involvement to a minimum, and if possible, eliminate human intervention altogether. This means, ideally it should be an automated system that can remove the artifacts in the degraded old motion pictures automatically. Attempts have been made in recent years to develop such systems. They include: AURORA [64,65,70], a hardware-based project started in 1995 under the European Union ACTS program; BRAVA [71], a project initiated in 1999 with the support of the European Commission; and REVIVAL [69], a commercial system developed in Nanyang Technological University, Singapore and subsequently bought over by the U.S. based company Da Vinci Systems. These systems have reported significant success in correcting common defects found in old motion pictures, such as dirt and sparkle noise [3-8,63,79,86], blotches [9-15,67,68,80,81,89], line scratches [16,17,66,82-85,88], and film unsteadiness [18, 90]. However, other types of defects, such as flicker [12,19-24,92-94] and color fading [58,59,91,97-103] have only had relatively little work done so far. Also there is a glaring absence of reported work done in the area of stain [76] removal. These are among the few remaining areas that need much more research effort before the eventual goal of motion picture restoration automation can be declared a success.

1.1.1 Anatomy of a Typical Motion Restoration System

The operation of a typical motion picture restoration (MPR) system in general involves a number of steps. This is illustrated in figure 1.1.

Stage 1 of the process involves digitization of the original (degraded) film into digital frames at a high resolution so that full details of the original film can be retained. This is illustrated in figures 1.1 (a) and (b). The digital frames created in stage 1 serve as the input to stage 2, which forms the heart of the MPR system. As shown in figures 1.1 (c) – (f) [grouped together within the dotted box], the restoration process passes the corrupted digital frames through a number of restoration steps in a sequential manner, where a particular type of artifact is dealt with in each step. Fully restored digital images are produced at the end of stage 2. Note that the order of the correction procedure in stage 2 may differ as the methods or algorithms employed in different MPR systems can be quite

dissimilar. The output from stage 2 is passed on to stage 3, where the fully restored digital images can be either stored away for archiving purpose, or get printed on reels of films for mass distribution. This is illustrated in figures 1.1 (g) and (h).

Fig. 1.1 Operation of a typical motion picture restoration (MPR) system

Note that unlike flicker/blotch/dirt/stain removal, scratch removal is carried out entirely on its own. The reason being that the algorithm for scratch removal is essentially very different from others. While other forms of defects generally require both spatial and temporal information for correction and hence all involve the determination of motion vectors, scratch is a defect that is often persistent at a particular place on a frame and hence spatial information alone will suffice in its correction.

1.2 Scope of the project

The research project in this thesis is on flicker correction and stain removal in motion picture restoration. Attempts will be made to achieve the following with regard to these two areas:

- Study the natural phenomenon of flicker artifacts in motion pictures (black-&-white, as well as color images)
- Study the natural phenomenon of stains in color motion pictures
- Analyze and model flicker artifacts in motion pictures
- Study and examine the performance of existing flicker correction algorithms
- Develop novel flicker correction algorithms
- Develop novel stain removal algorithms

Realizing these objectives will be significant in the sense that any success will mean we are one step closer to our final objective of creating a fully automated motion picture restoration system. The success of projects/products like AURORA and REVIVAL has made significant progress towards reaching this goal; it is the wish of the author that the work done in this thesis will take us one or more steps closer.

1.3 Thesis outline

This thesis is organized into eleven chapters.

Chapter 1 covers the background that is relevant to the research work in this thesis; it also states the objectives and scope of the whole project. Outline of the thesis is also given in this chapter.

Chapter 2 is divided into 2 sections. Section 1 covers the basics of flicker: the definition, types and causes of flicker. It is then followed by a brief discussion on the flicker correction process, a study of the criteria used for checking the quality of flicker correction, and a description of all the test sequences to be used to test out the algorithms in this thesis. Section 2 gives a general review on all the flicker correction work that has been reported in the research literature; a brief description of each of these methods is given.

Chapter 3 begins with the early part of the research work carried out by the author. Two heuristic approaches for correcting global flicker are presented in this chapter: Histogram Mapping and Intensity Mean Averaging.

Chapter 4 covers further work that has been developed based on those in chapter 3. Deficiency in the previous two methods is overcome. Two new methods that can handle local flicker are presented.

Chapter 5 touches on the motion estimation techniques that can be used in motion picture restoration. An improved flicker correction method is then developed by incorporating motion estimation into one of the flicker removal methods developed in chapter 4. Correction results of the new improved method are presented and discussed in this chapter.

Chapter 6 looks into a well-known flicker removal technique postulated by Roosmalen [25] in 1997. This method is based on a novel approach that involves correctly estimating two flicker parameters that can be used for flicker correction. A detailed study of this method [henceforth known as the FPE (Flicker Parameter Estimation) method], which includes a full mathematical derivation of the algorithm is presented. It is then followed by the presentation of a new algorithm based on Least Square Minimization (LSM), which is found to be an effective alternative to the FPE method. Both the FPE method and the LSM method are known to suffer from a common problem in not being able to detect motion blocks accurately; a solution based on stationary block re-examination is presented to address the problem.

Chapter 7 – Stationary block re-examination proposed in chapter 6 only managed to improve flicker correction marginally and a better solution is needed to address the

existing problem related to motion blocks detection. A new method through motion compensation is introduced to solve this problem and the results obtained are discussed and analyzed in detail in this chapter.

Chapter 8 proposes a new mathematical model to describe the degraded motion pictures. Based on this model, a new flicker correction method is developed and tested. This method has the advantage of being simple and computationally efficient compared to many earlier methods, while at the same time able to maintain reasonable quality of flicker correction. In addition, this new method is able to handle frames that contain scene change and occlusion, a task that none of the previous methods can handle.

Chapter 9 takes flicker correction to an uncharted territory – color motion pictures. This chapter begins with an introduction to the color space including the RGB and YUV standards, followed by a study on flicker correction problems in color motion pictures. Some ground-breaking work is carried out here and eventually a multi-channel flicker removal scheme is developed which is found to be very effective in correcting flicker in color motion pictures.

Chapter 10 – The flicker correction algorithm developed in chapter 9 is extended in this chapter to address another common artifact in color films – stain. It is found that the model developed earlier in chapter 8 works well for stain also, and hence a stain removal algorithm can be developed readily based on that model. The proposed stain removal algorithm is thoroughly tested and proved to be very effective.

Chapter 11 is the concluding chapter summarizing all the work done in this thesis, with recommendations given for possible future research work in this area.

1.4 Novel contributions

The research work described in this thesis makes several novel contributions to the field of motion picture restoration, particularly in the area of flicker correction and to a lesser extent in stain correction. The novel contributions include:

1. Global and local flicker reduction methods (Histogram Mapping method and Intensity Mean Averaging method) that employ motion compensation. However,

due to the heavy reliance on human intervention, these methods prove to be of limited use for practical flicker correction.

2. Incorporating stationary block re-examination into flicker parameter estimation (FPE) method for better accuracy in motion block detection thus improving the overall flicker correction results.
3. Using a Least Square Minimization algorithm to provide an effective alternative to the FPE method.
4. Incorporating motion-compensation to overcome the inherent deficiency in the FPE method – its inability to detect motion blocks accurately.
5. A simplified motion-compensated FPE algorithm that is able to deliver reasonable flicker correction results compared to the original FPE method.
6. Extending flicker correction to include color motion pictures, where no known previous flicker-correction work was done.
7. A stain removal/reduction algorithm for color motion pictures (stain removal/reduction is a virtually un-explored research area).

Chapter 2

Flicker and Its Correction – An Overview

2.1 Introduction

This chapter addresses the core problem of the research work in this thesis: flicker correction. It begins by an introduction to the basics of flicker, such as its definition, causes and types. It is then followed by a discussion on the fundamentals of a flicker correction process and its related problems. Criteria for evaluating flicker correction are then presented, followed by a description of the motion picture sequences to be used in testing various flicker correction algorithms developed later in this thesis. These background knowledge lays the foundation for the understanding of the later part of this chapter, which is an overview of flicker correction methods that have so far been reported in the research literature.

2.2 Flicker – The basics

Intensity flicker is defined as the unnatural temporal fluctuations of frame intensities that do not originate from the original scene [21]. Such fluctuations have the effect of making the frames look randomly brighter or darker than what they should be, and as a result bring much discomfort to human viewing. Flicker is a very common artifact in old black-and-white film sequences, and its causes are numerous, such as aging of film, unstable chemical processing, copying, shutter time variations and so on. Color films suffer from an additional flicker problem – color flicker[74], which is caused by uneven fading of the dye (cyan, magenta and yellow) commonly found in aging film materials. Typically the yellow dye has the worst fading problem because it lies on the outer-most layer of the film and so is most vulnerable to external factors such as moisture and dust.

Flicker is in general quite easily noticeable as it normally appears over large areas of the corrupted frame. Figure 2.1 shows 3 frames from a black and white flickered sequence known as the “Taxi” sequence. It can be seen that the second frame is considerably brighter than the other two frames and the extra brightness is homogeneous over the whole of frame 2. This type of flicker is known as “global flicker”; more on it will be elaborated in the next section.

Fig. 2.1 Flicker frames – an example of global flicker

2.3 Flicker classifications

Flicker can be classified in two ways, one based on the way flicker appears in the picture frames and the other on its origin.

2.3.1 Global flicker and local flicker

This classification is based on whether the fluctuation in intensity occurs in a localized manner, i.e., whether intensity fluctuation is location-dependent within a frame. As the name itself suggests, global flicker refers to flicker that affects the whole frame in a homogeneous manner, i.e., independent of location. For frames that have global flicker defects, individual frames are uniformly brighter or darker than their neighboring frames. On the other hand, local flicker refers to flicker that affects just a portion (or portions) of a whole frame, i.e., the effect is localized. Sometimes a frame may have regions of different levels of flicker over the entire frame; this type of flicker is also classified under local flicker.

Figure 2.2 shows an example of local flicker. It consists of three consecutive frames taken out from a 1911 Irish movie “Rory O’More”. All the three frames suffer from local flicker of different degree of severity, with localized flicker appearing as darker regions at different areas in these three frames. The most conspicuous local flicker in frame 2 is the

dark slanting bar at the far right-hand side of the frame. In frame 3, this flicker region is seen migrated about one-third length towards the left; in frame 1, this flicker is seen moved a little to the left, and is much fainter. Careful examination reveals more localized flicker regions in frame 2 and frame 3: darker regions can also be identified at the back of the human figure, as well as in the lower left corner of the frames.

Fig. 2.2 Flicker frames – an example of local flicker

2.3.2 Intensity flicker and color flicker

This classification is based on the origin of flicker. If the flicker effect is due to brightness variations in pixels of a frame, then this kind of flicker is intensity flicker. Flicker found in all black-and-white motion pictures is intensity flicker. Intensity flicker can also be found in color video when there are variations in the Y channel of the YUV color space.

Color flicker is due to unequal fading of dyes in the film material, which leads to temporal fluctuations in the various color channels. Shown in figure 2.3 are three frames taken from a Lady sequence known to have suffered from color flicker. Visually it can be seen that in the middle frame, there is a faint pinkish horizontal bar that runs through the lower chin portion of the lady's face. A slight pinkish smear also appears at the top left corner of the window. A more detailed coverage on the topic of flicker in color motion pictures will be given later in chapter 9 section 9.3.

Fig. 2.3 Flicker frame – a color flicker example

2.4 Flicker correction – an overview

2.4.1 Flicker correction and its related problems

Earlier in section 2.2 it was mentioned that flicker is basically the unnatural temporal intensity fluctuations in consecutive frames in motion pictures. It is thus obvious that an effective flicker correction process should be one that is able to either remove or reduce such fluctuations in a significant manner.

On the surface, flicker seems to be a rather simple problem to solve. The simplest heuristic idea is to define a particular frame intensity mean and use that as a reference to adjust all other frames to this intensity level. However a closer examination shows that this approach cannot work for a number of reasons. Firstly, to adjust all frames to have the same mean intensity will not work because it is wrong to assume that all frames in a good sequence should have a constant intensity mean. This is obvious as a scene change in any of the good frame sequence will immediately change the frame intensity mean of that particular frame. Secondly this simple approach will also not work in the presence of local flicker as adjusting the whole frame intensity mean to a fixed level will not remove any flicker that is localized.

There are other problems that make flicker correction an even more difficult task. One such problem is when there are motion regions in the frame sequence. With the presence of motion regions, motion compensation will be required in the flicker correction process. However, all the standard motion estimation algorithms would only work in the absence of any illumination variation, i.e. in a flicker-free environment. To devise a suitable algorithm that can still work in the presence of flicker is something of an unknown nature at this point in time (this will be dealt with later in chapter 5 section 5.4.1.2).

Another problem is related to occlusion, which refers to the covering/uncovering of a surface due to 3-D rotation and translation of an object which occupies only part of the field of view [42]. A 3-frame sequence in figure 2.4 will be used to illustrate this problem.

Fig. 2.4 Sequence of three consecutive frames

In a typical flicker correction process, information of the previous frame (or frames) that has been corrected earlier, is used for the correction of the current frame. This suggests that flicker correction algorithms are in general causal. In the example shown in figure 2.4, frame 2 is the frame undergoing flicker correction, while frame 1 is used as a reference frame for the correction process. Problem will arise when the motion estimation algorithm (a necessary step in the flicker correction process when there are motion regions) is applied to frame 1 and frame 2 to search for a match for the ship in frame 2. The sudden presence of the ship in frame 2 can be considered as it being “uncovered”, and since no match for this object can be found in frame 1, this region where the ship occupies will be detected as being “flickered” (due to the large intensity difference detected), which obviously is incorrect. When the process involved is non-causal, this problem can actually be solved using a bi-directional motion estimation algorithm [11], i.e., one that searches both the previous frame and the one immediately after. If an object is found in frame 3 but not in frame 1, then it has to be an uncovered object, and the situation can be dealt with accordingly.

Even if there are other means that can help us detect the ship in frame 2 as an uncovered object so that the region it occupies is not erroneously identified as being flickered, correction of that region will still be an impossible task as there is no matching object in frame 1. A more detailed discussion on the occlusion problem will be given later in chapter 5.

2.4.2 Criteria for evaluating flicker correction results

It is important to come up with criteria that can be applied easily to evaluate the effectiveness of any flicker correction process. The following have been used for such evaluation purpose:

- (i) Smoothness of the intensity mean curve and intensity variance curve
- (ii) PSNR (peak signal-to-noise ratio)
- (iii) Coding efficiency
- (iv) Subjective evaluation

2.4.2.1 Smoothness of the intensity mean curve and intensity variance curve

This is the most widely used criterion for evaluating the correction process and the theoretical basis is very simple. If the flicker correction process is effective and the image contents do not change significantly, then the corrected frame means and variances should be quite similar from frame to frame. This implies that both the intensity mean curve and the intensity variance curve of the corrected sequence will have much less fluctuation compared to those of the original flicker sequence, i.e., they will be much smoother. Therefore, the temporal smoothness of frame means and frame variances will constitute a measure of the effectiveness of a flicker correction process [24, 25]. This is illustrated in figures 2.5(a) and 2.5(b), which show respectively the intensity mean and variance curves of a 40-frame natural flicker Lady sequence before and after correction. It can be seen that for the corrected frame sequence, both the intensity mean and variance curves are indeed much smoother compared to those of the original flicker sequence.

Fig. 2.5 (a) Intensity mean curves of Lady sequence (before and after correction)

Fig. 2.5 (b) Intensity variance curves of Lady sequence (before and after correction)

When the effectiveness of two or more flicker correction processes are to be compared, it can be easily done by comparing the relative smoothness of the intensity mean and variance curves of the corrected sequences. Obviously, the same flicker sequence has to be used as the input for all individual flicker correction processes to maintain a common reference. The best correction process will be the one that produces the smoothest intensity mean and variance curves. A natural question to ask is: how do we measure the smoothness of a curve in a quantified manner? The answer to this question is simple: just measure the variance [26]. The reasoning is very straight-forward. A curve that is smooth is one that has little fluctuation, and this means it should have a small variance. The smoother the curve, the smaller the variance and vice versa.

2.4.2.2 PSNR (peak signal-to-noise ratio)

PSNR is widely used to measure the signal quality of images or video. Based on a $M \times N$ pixel frame with 8-bit gray-scale intensity range, PSNR between a reference video frame f_j and the processed video frame f_k is defined as:

$$\text{PSNR}(f_j, f_k) = 10 \log_{10} \frac{255^2}{MSE} \quad (2.1)$$

$$\text{where MSE} = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N [i_j(x, y) - i_k(x, y)]^2$$

where $i_j(x, y)$ and $i_k(x, y)$ are respectively the pixel intensity at location (x, y) of the reference frame and processed frame. A large PSNR value means small difference between the reference and processed frames and this in general implies close resemblance between the two.

PSNR provides a convenient means to compare the effectiveness of different flicker correction methods. It can be illustrated by using a simple example here whereby two different flicker correction methods A and B are used to correct a flicker sequence. Let f_o , f_f and f_c be respectively the original (good) frame, flicker frame and corrected frame. The effectiveness of flicker correction between these two methods can be easily evaluated by comparing $\text{PSNR}(f_o, f_c)_A$ and $\text{PSNR}(f_o, f_c)_B$, where the former represents PSNR obtained using method A and the latter by method B. The flicker correction method that gives a larger PSNR value is the more effective one.

It should be pointed out that there is a big constraint in this method in that the unimpaired original frame (i.e. f_o) sequence must be available. Unfortunately this is not the case in practice – in most, if not all cases, the archived film material to be corrected just has no unimpaired references that can be found.

However there are situations in which such a constraint does not pose any problem. This happens when a flicker sequence is artificially (synthetically) created for testing purposes by adding flicker to an originally unimpaired frame sequence, and the resulting flicker sequence is then used as the input for testing a flicker correction process. There are good reasons why sometimes an artificially created flicker sequence is a better input choice to test out a flicker correction process – it gives us more control on the input used for testing. For example, to find out how effective a flicker correction method is in correcting global flicker, an artificial flicker sequence that contains only global flicker can be generated and used as the input for the test. Real flicker material normally cannot be used for such specific testing as it rarely contains only global or local flicker.

Although PSNR is able to give a quantitative indication on image quality, it cannot be a reliable predictor of perceived visual quality simply because PSNR has overly simplified actual human visual perception, which is a very complex process. As a result, high PSNR values do not always translate into signals with perceptually high quality and vice versa [27, 28].

2.4.2.3 Coding efficiency

It is generally accepted that image restoration generally leads to more efficient image compression, i.e., better coding efficiency. Therefore, higher coding efficiency achieved in compression would mean a higher image quality, and vice versa. What has been just said is true when the restoration process deals with removing artifacts such as noise, blotches and flicker etc. However it does not hold for certain types of defects such as out-of-focus images – de-blurred images actually require more bits for coding than the blurred originals.

Coding efficiency in general can be measured in either *bits* or *dB* as illustrated below. Figure 2.6 shows a scheme that can be used to measure the coding efficiency between a corrupted frame sequence and its corresponding corrected sequence.

Fig. 2.6 A scheme to measure coding efficiency

For simplicity of argument, let us consider just one frame y_{bc} of a corrupted sequence and its corresponding corrected frame i_{bc} in the corrected sequence before coding. Assume y_{ac} and i_{ac} are respectively the corrupted frame and the corrected frame after coding. For the corrupted frame, the PSNR computed between the codec input and output is given by

PSNR(y_{bc} , y_{ac}). Likewise for the corrected frame, it is given by PSNR(i_{bc} , i_{ac}). For a codec set to a fixed bit rate, a measure of quality improvement ΔQ can be defined as the difference between PSNR(i_{bc} , i_{ac}) and PSNR(y_{bc} , y_{ac}), i.e.:

$$\Delta Q = \text{PSNR}(i_{bc}, i_{ac}) - \text{PSNR}(y_{bc}, y_{ac}) \quad (2.2)$$

$\Delta Q > 0$ will mean that the quality of the corrected frame is higher than that of the corresponding corrupted frame, and a negative value will mean the opposite. For a N-frame sequence, a plot of ΔQ_k against k , where k is the frame number, will give a good indication on the quality of each frame of the corrected sequence. Alternatively, the quality of the corrected sequence can also be measured by an average value given by

$$\frac{1}{N} \sum_{k=1}^N \Delta Q_k.$$

If *bits* instead of *dB* is used as a measure of quality improvement ΔQ , the setup in figure 2.6 is still applicable but the process is slightly modified. First, the corrected sequence is coded at a fixed bit rate. Next, the bit rate for coding the corrupted sequence is searched so that PSNR(i_{bc} , i_{ac}) equals PSNR(y_{bc} , y_{ac}), i.e., the same amount of compression error is introduced into the corrected and corrupted sequence. The quality of improvement ΔQ is given by the difference in bit rates, i.e., $\Delta Q = \text{Bit-Rate}(\text{corrupted}) - \text{Bit-Rate}(\text{corrected})$. $\Delta Q > 0$ means that the corrected sequence requires fewer bits for coding than the corrupted sequence, and is thus of higher image quality; negative ΔQ value will mean the opposite.

2.4.2.4 Subjective evaluation

Since the human beings are the ultimate receivers in most image-processing applications, the most reliable way to assess the quality of an image is actually by subjective evaluation.

The international Telecommunication Union (ITU) has standardized a number of methods [29] for evaluating image sequences through subjective testing, which suggest standard viewing conditions, criteria for the selection of observers and test material, assessment procedures, and data analysis methods. The three most commonly used methods are the following:

- Double Stimulus Continuous Quality Scale (DSCQS)
- Double Stimulus Impairment Scale (DSIS)
- Single Stimulus Continuous Quality Evaluation (SSCQE)

A detailed description of these three methods is beyond the scope of this thesis and the reader may refer to [104] for more details. For all these methods, a minimum of 15 observers are needed to take part in the evaluation exercise. The ratings from all observers are then averaged into a Mean Opinion Score (MOS), which represents the subjective evaluation of the given image sequence. It has been reported that the AURORA project had used DSCOS as part of their evaluation scheme.

A method simpler than the DSCQS method and at the same time more suitable for evaluating perceived image quality has been reported in [105]. Known as the “Two alternatives forced choice” (2AFC) method [30], this method is used to determine from two image sequences under test, the one that has the highest perceived quality.

As illustrated in figure 2.7, members of the test panel are shown pairs of image sequences A and B twice. Within the pair A and B, one is a corrupted sequence and the other is the restored sequence, which is which is random. Each sequence is approximately 10 seconds in duration. In between showing sequence A and sequence B, there is a gap of 2 seconds where the screen goes blank to a mid-gray value. After showing the pair A and B for the first time, the screen goes blank to a mid-gray value for 5 seconds, after that the pair A and B is screened again. If A was the corrupted sequence in the first viewing, then it is also the corrupted sequence in the second viewing. The same is true for B. After viewing the A and B pair the second time, the observers must give their assessment on which sequence gives the better visual quality.

Fig. 2.7 An illustration of the 2AFC testing procedure

The outcome of the 2AFC is determined by one of two cases. In case 1, a majority of votes is given to either A or B. This means a general consensus has been reached on whether the perceived quality of the corrected sequence is better than that of the corrupted sequence. In case 2, about 50% of the votes is given to each of the sequences; this suggests that there is no consensus reached on which sequence (corrupted or restored) is better.

So in summary, the objective approach is able to give a quantitative measure, from which unambiguous information of a corrected image sequence, such as PSNR values, smoothness of the intensity mean curve, or saving in number of bits required for compression etc., can be derived. These information are easy to understand and are particularly useful when they are used for comparing the effectiveness of different correction processes. However, hard figures obtained under quantitative measurement, e.g., an improvement in terms of dB or bits, may not necessarily translate into improvement in the visual experience of an observer to the same extent. For example, a correction process may produce a large positive ΔQ dB value, implying a substantial improvement in image quality. However to the observer, he may feel that the improvement is just marginal. The consensus is that if a corrected sequence is shown to have improved over the original corrupted sequence with supporting quantitative measurement, such as a smoother intensity mean curve, or a positive ΔQ dB value, then it is safe to say that the visual quality of the corrected sequence is at least equal, and most probably better, than that of the original corrupted sequence.

Subjective measurement, on the other hand, is less precise but nonetheless is the most dependable means for checking the visual quality of any corrected image sequence. Unfortunately, owing to the involvement of human assessors (a minimum of 15 is recommended), subjective evaluation is troublesome, time consuming and not very practical when a large amount of information is to be evaluated. Because of the limitation of the availability of such resources and the time constraint imposed on the research work in this thesis, it was finally decided that formal subjective evaluation would not be a viable choice and hence was not adopted.

2.4.3 Motion picture sequences used for testing flicker correction algorithms

Motion picture sequences are needed to serve as testing sequences for the evaluation of the effectiveness of a flicker correction algorithm. In general two types of test sequences

are needed: Synthetic test sequences and natural test sequences. Synthetic test sequences refer to film sequences that are originally good but with artificial flicker added. It has the advantage of being able to perform tests in a controlled environment where flicker can be created according to the flicker model proposed. Due to the ease of manipulation, tests can be carried out to explore the performance under extreme conditions; this is something not achievable when natural test sequences are used. Natural test sequences refer to film sequences that contain real (non-synthetic) flicker. They are used to verify the practical effectiveness of a flicker correction algorithm. In general, to demonstrate the effectiveness of a flicker correction algorithm, a few natural test sequences (preferably each with a different characteristic) will be used in the testing process.

Synthetic test sequences were created by adding the desired artificial noise to the following good film sequences. Note that except for the Wedding sequence where stain is added, the other three sequences are all added with flicker.

- (i) 10-frame Taxi (256x190) B&W sequence (for adding local flicker)
- (ii) 20-frame West (256x256) B&W sequence (for adding global flicker)
- (iii) 20-frame Fountain (720x301) color sequence (for adding color flicker)
- (iv) 20-frame Wedding (720x301) color sequence (for adding stain)

The Taxi sequence is a 10-frame (frame 0 to frame 9) black-and-white sequence with three moving cars traveling along the same road. A white car is at the centre of the frame and is turning right at a T-junction, while two other cars in black traveling in opposite directions are appearing gradually from the bottom two ends of the frame. The sequence is basically clean (free from any noise defects) and has no camera panning at all.

The West sequence is a 20-frame (frame 0 to frame 19) black-and-white sequence which shows a man in the process of passing a bag through the car window to a person sitting inside. The upper trunk of the man occupies the left half portion of the frame and almost takes up the full height of the frame. There is considerable arm movement in the man while his body movement is only minimal. The sequence is basically clean except in frames 4 and 19 where a tiny dirt mark can be spotted near the centre of the frame. There is no camera panning in the sequence.

The Fountain sequence is a 20-frame (frame 5130 to frame 5149) color sequence which shows a lady sitting on the rim of a fountain talking to a man standing beside her. These two figures occupy the centre of the frame and at the background is a street dotted with a few pedestrians walking passed. There is no camera panning and the sequence is very clean.

The Wedding sequence is a 20-frame (frame 7051 to frame 7070) color sequence of a wedding scene showing about eight to nine persons toasting to each other. The sequence is clean with no camera panning.

The natural flicker test sequences used in this thesis are:

- (i) 40-frame (720 pixel x 576 pixel) Lady sequence
- (ii) 40-frame (720 pixel x 576 pixel) Woman sequence
- (iii) 30-frame (360 pixel x 288 pixel) Tunnel sequence
- (iv) 21-frame (1556 pixel x 1828 pixel) Lantern sequence

The Lady sequence is a 40-frame (frame 0 to frame 39) sequence in both black-and-white and color. It shows a lady in a room with her upper trunk in the centre of the frame. The lady has her face facing left and gradually rotating right up to about frame 6, after which her body rotation is mainly confined to her upper trunk (between frames 6 and 22). Considerable hand movement is noticed between frames 28 to 39. Camera pan and shake are nearly absent in the sequence. Both local and global flicker are present throughout the sequence, with local flicker being the more prominent one (especially in frames 6, 21, 26 and 37). Small amount of noise mainly in the form of short white vertical scratches are noticeable in certain frames.

The Woman sequence is a 40-frame (frame 60 to frame 99) sequence in color. It shows the upper trunk of a woman occupying the centre of the frame. The woman is seen rotating her body gradually to the right up to about frame 83. Further on, she begins walking towards right across the room accompanied by substantial horizontal camera panning in the same direction. Severe color flicker appears in frames 68, 73 and 84. A small amount of noise in the form of white marks can be seen in frames 70, 71 and 74.

Tunnel sequence is a 40-frame (frame 0 to frame 39) black-and-white sequence which shows a man standing in a tunnel with another man squatting next to him outside the tunnel. The upper trunk of the man in tunnel is seen leaning forward slowly while talking to the squatting man, whose head shows some downward then upward movement. There is some slight camera unsteadiness in the sequence and the camera pans slightly and gradually to the right in the first 7 frames of the sequence. Considerable local and global flicker can be identified throughout the whole sequence.

Lantern sequence is a 21-frame (frame 0 to frame 20) sequence in both black-and-white and color. It shows three men walking past a pillar and entering a courtyard lit up by three hanging lanterns. Camera unsteadiness in the form of up-down movement runs through the whole sequence, and there is considerable camera panning towards the right from frame 11 onwards. A small amount of both local and global flicker can be identified, together with a small amount of sporadic speckle noise.

All the test sequences mentioned above can be found in a server at the following URL:
<http://kkweb.ntu.edu.sg>.

Corrected frames by methods presented in this thesis using these test sequences as inputs can also be found at the same location. Readers may refer to Appendix E for information on how these image sequences are organized and stored at the server.

2.5 Overview of existing flicker correction methods

This section presents an overview of known flicker correction methods that have been published in the research literature up to the present time.

The earliest research work on flicker correction probably can be traced back to that carried out in the restoration of an old Australian film “Story of the Kelly Gang” by P. Richardson and D. Suter in 1995 [19]. More work by others was carried out in the following year [12,20,54]. These earlier work tried to correct flicker by either equalizing the intensity histograms or equalizing the mean frame values of the frame sequence. However, the correction results obtained were rather limited and none formed general solutions to the flicker correction problem.

It was not until 1997 that more effective work in flicker correction was produced. This work will be reviewed in the following section.

2.5.1 Roosmalen's flicker parameter based algorithm [25] and related work

Postulated by P. Roosmalen in 1997, this method uses a linear noisy frame model (excluding non-flicker noise) given by:

$$Y(x,y,t) = \alpha(x,y,t)I(x,y,t) + \beta(x,y,t) \quad (2.3)$$

where x, y are discrete spatial coordinates and t indicates the frame number, $Y(x,y,t)$ and $I(x,y,t)$ respectively indicate the observed and original image intensities, $\alpha(x,y,t)$ and $\beta(x,y,t)$ are respectively the flicker gain and flicker offset parameters (these two parameters are collectively known as the “flicker parameters”).

Flicker parameters $\alpha(x,y,t)$ and $\beta(x,y,t)$ can be estimated using intensity information of both the current corrupted frame and the previous frame (which has been corrected earlier), the original image intensity $I(x,y,t)$ can then be easily estimated using equation 2.3.

Subsequent work [23, 24, 32] that is also based on the parameters estimation approach has since been reported after Roosmalen's pioneering work.

2.5.2 Ohuchi's hierarchical flicker correction model [23] and related work

Based on the same linear noisy frame model given by equation 2.3, T. Ohuchi presented in year 2000 a hierarchy of flicker correction models as given below for correcting flicker:

$$(i) \quad I(x,y,t) = \gamma(t) Y(x,y,t) + \delta(t) \quad (2.4)$$

$$(ii) \quad I(x,y,t) = \gamma(x,y,t) Y(x,y,t) + \delta(t) \quad (2.5)$$

$$(iii) \quad I(x,y,t) = \gamma(t) Y(x,y,t) + \delta(x,y,t) \quad (2.6)$$

$$(iv) \quad I(x,y,t) = \gamma(x,y,t) Y(x,y,t) + \delta(x,y,t) \quad (2.7)$$

where I and Y take on the same meaning as given in section 2.4.1, and γ and δ are respectively the correction-gain parameter and the correction-offset parameter.

Ohuchi applies a gradual strategy for robust flicker correction. For instance, it first corrects image flicker according to the lowest model given by equation 2.4. The corrected image is then used as an initial setting for flicker correction based on a higher model given by say, equation 2.5. By minimizing a robust cost function in an iterative fashion, a refined flicker corrected image will be produced. Different combinations of hierarchical models can be used depending on the flicker types in order to achieve the best correction result.

Improvement was made by A. Kokaram [55] in 2003 who proposed an alternative approach for parameter estimation with the aim to eliminate the bias related to the linear regression used in Ohuchi's work.

A method that was suitable for sequences without camera motion was proposed by Jung [57] in 2000. The main idea behind this method is to isolate the common background for the sequence and the moving objects using spatio-temporal segmentation. The background is estimated through a regularized average of the frames of the sequence, while the moving objects are motion-compensated, average and regularized to preserve spatial continuities.

2.5.3 Naranjo's non-linear flicker model [22] and related work

Proposed in year 2000, V. Naranjo's work deviates from the previous few methods in that it uses a non-linear model to describe flicker, and the correction is done through a histogram matching process [31].

The idea of the method is as follows. First the histogram $h_f(v)$ of the flicker frame is obtained, then a target histogram $h_t(u)$ is also obtained by averaging the histograms of a few neighboring frames (v and u are the respective intensity value on the x-axis in the two histograms). From the two histograms, the respective cumulative histograms $C_f(v)$ and $C_t(u)$ are then obtained. The target histogram is then used as a reference to correct the flicker frame by equating the two monotonously increasing functions $C_f(v)$ and $C_t(u)$.

Work by P. Schallauer [56] in 1999 is also of a similar nature as it is also based on histogram equalization.

Naranjo's work was extended by F. Pitie [33] in 2004 to deal with occlusion and spatial variations. It uses a flicker model $I(\mathbf{x}, t) = f[Y(\mathbf{x}, t), \mathbf{x}]$ where I and Y are respectively the flicker-free frame and observed frame, $\mathbf{x} = (x, y)$ represents the pixel location and f is the non-linear distortion function to be estimated. Flicker is locally estimated in several control points and a dense correction function is obtained using interpolated splines.

Another piece of related work was that described by T. Vlacho [34] in 2004 which is also a method based on histogram manipulation. It does not provide a mechanism for spatial adaptation but instead uses non-linear compensation motivated by principles of photographic image registration.

This model takes into account the Hurter-Driffield Density versus Log Exposure characteristic to estimate image density errors caused by exposure inconsistencies. The model is shown to be highly nonlinear and can accommodate typical grayscale manipulations performed during film-to-video transfer. The correction profile used is one that is obtained from the maximum values of histogrammed gray level differences between a reference and a test frame.

Improved versions of the work were later reported by G. Forbin in 2006 [92-95]. New components were introduced to overcome some earlier short-comings. Spatial variability of flicker was addressed by applying the baseline algorithm in a block-based fashion followed by bilinear interpolation to avoid blocking artifacts. A weighted estimation solution was developed to give a more reliable intensity error profile. Motion compensated prediction was used to prevent contamination of measurements due to motion.

2.6 Conclusion

This chapter has provided all the necessary background for the understanding of flicker, which is the main research focus in this thesis. A thorough literature survey on flicker research has also been carried out, and it is found that the bulk of the reported research work are basically linked to three separate major sources of work; respectively they are the work done by Roosmalen, Ohuchi and Naranjo. With a full understanding of flicker as

well as a general understanding of the earlier work done, we are now in the position to take the very first step into flicker research. The work in the next chapter marks the very beginning of our research work carried out in this thesis.

Chapter 3

Global Flicker Correction

3.1 Introduction

This chapter covers the flicker correction work carried out during the initial phase of the research work in this thesis. Global flicker was chosen as the subject of study as it was less complex in nature and would serve as an ideal spring-board to the study of more complex flicker correction problems later. Two heuristic methods for correcting global flicker were produced as a result of the research work done in this chapter: The Histogram Mapping method and The Intensity Mean Averaging method. The flicker correction performance of these methods was evaluated with natural flicker test sequences.

3.2 The histogram mapping method

The idea of histogram mapping has its root in histogram equalization, a technique used to modify a given image by adjusting its original histogram into a uniform histogram. Histogram equalization is a well-studied topic in the field of image processing and a detailed coverage on this can be found in [31]. For this reason, it will be left out in our discussion here and our focus will be just on the topic of histogram mapping.

3.2.1 Histogram mapping

For a given picture frame, a histogram can be constructed which is basically a plot with its gray levels on the horizontal axis and the number of pixels with the corresponding gray level on the vertical axis. In general, for an image with discrete gray levels in the range $\{0, L-1\}$ where $L = 2^m$ in which m is the number of binary bits used to represent the gray levels, a probability density function $p(r_k)$ can be defined such that

$$p(r_k) = n_k/n, \quad 0 \leq r_k \leq L-1 \text{ and } k = 0, 1, \dots, L-1$$

where $p(r_k)$ represents the probability of the occurrence of the k^{th} gray level, n_k the number of times this k^{th} gray level appears in the n -pixel image, and r_k the normalized gray level which equals to $k/(L-1)$. A plot of $p(r_k)$ versus r_k is known as the normalized histogram of the given frame.

With $p(r_k)$ defined, a transformation function S_k that can be used for histogram mapping can now be introduced. S_k is basically a cumulative distribution function (CDF) defined by equation 3.1. Two examples of such a transformation function are given in figures 3.1(a) and 3.1(b).

$$S_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p(r_j) \quad (3.1)$$

where $0 \leq r_j \leq 1$ and $k = 0, 1, \dots, L-1$.

3.2.2 Using histogram mapping to correct flicker

The histogram provides a global visual estimate of the image pixel value distribution. Although a histogram gives nothing specific about the content of an image, the shape of the histogram of a frame does give useful information about intensity and contrast. For consecutive motion picture frames, if there is no scene change in the sequence, then we would expect the histogram of each of these frames in the sequence to be almost the same. This implies that if the first frame in the sequence is perfect (either originally perfect or has been flicker-corrected earlier), then its histogram can be used as a reference to correct the second frame. When the second frame is corrected, it can then be used to correct the third frame, and this process is repeated until the last frame is done. Given below is a detailed illustration of this process.

Fig. 3.1(a) CDF of flicker frame

Fig. 3.1(b) CDF of reference frame

Assuming that the current frame is a flicker frame to be corrected and it has a cumulative distribution function $S = T(r)$ as shown in figure 3.1(a). The preceding (reference) frame has a cumulative distribution function $V = G(z)$ as depicted in figure 3.1(b). Using these two CDFs, the image correction process can be carried out in the following manner: For pixels in the flicker frame with the k^{th} gray-level (i.e. normalized gray level r_k), a corresponding transformed value z_q can be obtained by:

$$z_q = G^{-1}[T(r_k)] \quad (3.2)$$

where q denotes the q^{th} gray level in the transformed domain, i.e., $q = z_q (L-1)$. Image correction can now be carried out with all k^{th} gray-level pixels in the flicker image changed to q^{th} gray level. Correction of the current flicker frame is completed when the full range of k values are covered. When this is done, correction of the next flicker frame can be started by repeating the same process, with the previously corrected frame used as the reference frame.

The histogram mapping method presented here is quite similar to Naranjo's work [22] except in the way the histogram of the reference frame is obtained. As explained earlier, our method relies on a "perfect" first frame whose histogram is used as the reference to correct the second frame, and upon corrected, the histogram of the second frame is used as the reference to correct the third frame and so on until the last frame in the sequence is processed. In Naranjo's work, the reference histogram is obtained differently. For a frame to be corrected, the reference histogram is obtained by averaging the histograms of its neighboring frames.

3.2.3 Experimental results and analysis

The histogram mapping method was tested on a 40-frame natural flicker Lady sequence. The correction process will be illustrated by using two neighboring frames, with one of them (the good frame) used as the reference to correct the other which is a flicker frame.

The original flicker Lady sequence has a median intensity mean of 81.77, and this value will be used as a reference value to check if a frame is flickered. If a frame has an intensity mean at a certain threshold value above or below this value, then it is considered

a flicker frame. Based on this criterion and using a threshold value of 1.5, it is noticed that in the Lady sequence, frame 20 is good (intensity mean = 82.08) and frame 21 is flicker (intensity mean = 85.86), hence these two neighboring frames are suitable candidates to be used to illustrate the correction process.

Figures 3.3(a) and 3.3(b) respectively show the histograms of frame 20 [figure 3.2(a), good and used as the reference] and frame 21 [figure 3.2 (b), flicker], and figures 3.4(a) and 3.4(b) are respectively the corresponding transformation functions. It can be seen that both the differences between the two histograms and the transformation functions are rather minute; this is due to the rather small amount of flicker present in frame 21. The corrected image after using Histogram Mapping is given in figure 3.2(c), and its histogram and transformation functions are respectively given in figures 3.3(c) and 3.4(c).

Fig. 3.2 (a) frame 20 [good], (b) frame 21 [flicker] and (c) frame 21 [corrected]

(a)

(a)

(b)

(b)

(c)

(c)

Fig. 3.3 Histogram of
(a) frame 20 [good]
(b) frame 21 [flicker] and
(c) frame 22 [corrected]

Fig. 3.4 Transformation function of
(a) frame 20 [good]
(b) frame 21 [flicker] and
(c) frame 22 [corrected]

There are a number of observations that can be made with regard to the Histogram Mapping method and a discussion will be given below.

When a comparison is made between the histogram of the corrected frame [figure 3.3 (c)] with that of the reference frame [figure 3.3 (a)], it can be seen that the former has a shape that comes close to that of the latter, but nonetheless the two are not identical. This can be explained by the fact that the reference frame, which is a preceding good frame, is not an exact copy of the current frame. For this reason, their histograms will only be close but never identical.

Also vertical “spikes” are seen at certain intensity levels in the histogram of the corrected frame. This error is inherent in the Histogram Mapping method because the transformation will yield exact results only in continuous case. In our case here, discrete values are used for intensity levels and hence error is unavoidable.

The closeness in shape between the histogram of the corrected frame and that of the reference frame implies that the Histogram Mapping method is effective in flicker correction.

Visual checking also gives an impression that the corrected frame has an intensity level closer to that of the reference frame. This is confirmed when intensity measurements are made on the image frames. The reference frame (frame 20), frame under correction (frame 21) and corrected frame 21 respectively give intensity values of 82.08, 85.86 and 83.00. Clearly, there has been a slight improvement in the intensity difference, from 3.78 ($=85.86-82.08$) down to 0.92 ($=83.00-82.08$).

To evaluate the effectiveness of flicker correction of the proposed method, the full 40-frame Lady sequence was corrected, and the results are given in figures 3.5(a) and 3.5(b). Figure 3.5(a) shows the intensity mean curve of the original flicker sequence as well as that of the corrected sequence. It can be seen that the intensity mean curve of the corrected sequence is much smoother than that of the original, implying much reduced intensity fluctuations (i.e. flicker) in the corrected sequence. As explained in chapter 2 section 2.4.2.1, the smoothness of an intensity mean curve is an indicator of its intensity fluctuations, and it can be conveniently measured by the variance of the curve. In figure 3.5(a), the original intensity mean curve has a variance value of 3.11, whereas that of the

corrected sequence is 0.05, which is just 1.6% of the original. Visual checking of the corrected image sequence confirms the reduction of flicker after the correction process.

Similar comparison is done between the intensity variance curve of the original sequence and that of the corrected sequence; this is shown in figure 3.5(b). The corrected sequence again gives a smoother curve than that of the original sequence, with a variance value of 138 compared to 5173; this represents a reduction of 97%.

The Histogram Mapping method was also tested with more natural flicker sequences, and the test results of two of them, namely the Tunnel sequence and the Lantern sequence will be presented below. To keep things simple, only intensity mean curves will be presented (see figures 3.6 and 3.7). Generally speaking, a comparison of the intensity mean curves alone is a good enough check on the effectiveness of flicker correction.

Fig. 3.5 (a) Intensity mean curves of the Lady sequence

Fig. 3.5 (b) Intensity variance curves of the Lady sequence

Fig. 3.6 Intensity mean curves of the Tunnel sequence

Fig. 3.7 Intensity mean curves of the Lantern sequence

3.2.4 Discussions

Test results obtained earlier show that intensity mean curve of the corrected Lady sequence has a variance just 1.6% of the original. The corresponding figures for the Tunnel and Lantern sequences are respectively 0.5% and 5.0%. As a matter of fact, our test results on all natural flicker sequences tested typical return figures that are less than 10%. Such low figures strongly suggest that the Histogram Mapping method is indeed effective in flicker correction.

However, it must be pointed out that since the Histogram Mapping method uses global information of the image for image correction, it is only effective for correcting global flicker but not local flicker. In an image where the flicker region is localized, the flicker regions will not get removed as the Histogram Mapping method has no means of detecting these regions. More will be elaborated in the following example.

Fig. 3.8 (a) frame 5 [good], (b) frame 6 [local flicker] and (c) frame 6 [corrected]

Frame 5 [figure 3.8(a)] and frame 6 [figure 3.8(b)] in the original Lady sequence are respectively a good frame and a frame that suffers from local flicker. When these two frames are compared side-by-side, it can be seen that in frame 6, there is a local flicker region (shown within the dotted-lined box) that appears as a relatively bright horizontal bar that runs across the lower chin and neck portion of the lady's head. In the corrected frame 6 in figure 3.8(c), this local flicker region is still clearly visible.

It should be pointed out that looking at the frame intensity mean alone is not a reliable way to check the effectiveness of a flicker correction process when the frame contains local flicker. In our case here, frame 6 has an intensity mean of 87.36 before correction; after correction, its intensity mean drops to 82.80, which is much closer to that (82.07) of frame 5. These intensity values on surface seem to suggest flicker correction is working, but the foregoing results in the earlier paragraph showed that this is not true at all.

3.3 The intensity mean averaging method

The Intensity Mean Averaging method to be proposed here is closely related to frame averaging, a method that has been used for general noise reduction of image sequences [53, 37]. As flicker is the unnatural intensity fluctuation in a frame sequence and can be considered as some kind of noise, it is likely that the frame averaging method (with the necessary modification) may also work for flicker reduction. In this section, our work in adapting the frame averaging method to flicker correction will be presented, details of which are given as follows.

3.3.1 Frame averaging – the basics

Noise reduction using frame averaging, as illustrated in figure 3.9, is essentially a simple process in which N adjacent frames of an image sequence are averaged together to form a noise-reduced frame. Let f_j be the input frames and $\hat{f}_i$ be the frame obtained through averaging, and (x,y) be the location of a pixel within the frame, then the frame averaging process can be written as:

$$\hat{f}_i(x, y) = \frac{1}{N} \sum_{j=i-\frac{(N-1)}{2}}^{i+\frac{(N-1)}{2}} f_j(x, y) \quad (3.3)$$

Fig. 3.9 Frame averaging for N=3

3.3.2 Intensity mean averaging

If the expected value operator E is applied to equation 3.3 over the image frame, then the following will result:

$$\begin{aligned}
 E[\hat{f}_i(x, y)] &= E\left[\frac{1}{N} \sum_{j=i-\frac{(N-1)}{2}}^{i+\frac{(N-1)}{2}} f_j(x, y)\right] \\
 &= \frac{1}{N} \sum_{j=i-\frac{(N-1)}{2}}^{i+\frac{(N-1)}{2}} E[f_j(x, y)] \\
 \text{i.e. } \hat{m}_i &= \frac{1}{N} \sum_{j=i-\frac{(N-1)}{2}}^{i+\frac{(N-1)}{2}} m_j
 \end{aligned} \tag{3.4}$$

where $\hat{m}_i$ is the intensity mean through averaging and m_j is the respective intensity mean of the input frames.

Consider N adjacent frames of an image sequence in which f_i is the flicker frame, then the correct (i.e. original) intensity mean value of frame f_i is expected to be approximately given by $\hat{m}_i$ in equation 3.5. Note that the flicker frame has to be taken out of the averaging process (as indicated by the constraint $j \neq i$ in equation 3.5) and the number of frames has to be adjusted to $N-1$ accordingly.

$$\hat{m}_i = \frac{1}{N-1} \sum_{\substack{j=i-\frac{(N-1)}{2} \\ j \neq i}}^{i+\frac{(N-1)}{2}} m_j \tag{3.5}$$

To illustrate how intensity mean averaging can be used for flicker correction, let us consider just three frames f_1 , f_2 and f_3 in an image sequence in which f_2 is the flicker frame and f_1 , f_3 are respectively the previous and next frame (both f_1 and f_3 are good frames). The original intensity mean of frame f_2 can be approximated by the intensity mean average m_{avg} computed using equation 3.5:

$$m_{avg} = \frac{1}{2}(m_1 + m_3)$$

Let us now define flicker gain, m_{gain} , as the difference between m_2 , the intensity mean of the flicker frame f_2 , and m_{avg} , the approximated original intensity mean of frame f_2 , i.e.:

$$m_{gain} = m_2 - m_{avg} \quad (3.6)$$

Equation 3.6 can be re-written as:

$$m_{avg} = m_2 - m_{gain} \quad (3.7)$$

From equation 3.7, it can be deduced (see Appendix B) that $\hat{I}_2(x, y)$, an estimate of the original intensity at location (x, y) in frame f_2 ; $I_2(x, y)$, the intensity of the flicker frame f_2 at the same location; and m_{gain} , the flicker gain as defined in equation 3.7, can all be linked together by equation 3.8 given below:

$$\hat{I}_2(x, y) = I_2(x, y) - m_{gain} \quad (3.8)$$

Equation 3.8 is the equation that can be used for flicker correction.

3.3.3 The intensity mean averaging method – a few issues

3.3.3.1 Flicker frame detection

It has been shown earlier that in the Intensity Mean Averaging method, good neighboring frames are used to correct the flicker frame. It is obvious that before any flicker correction work can be started, the very first task that needs to be carried out is to identify the good frames and flicker frames in a given frame sequence. As have been discussed in section 3.2.3, a parameter that is suitable for this job is the median intensity mean, which is defined as the median of the average intensity of each frame in the image sequence. Using the median intensity mean as a reference value, any frame that has an average intensity value above or below this reference value by a certain threshold value will be taken as a flicker frame.

3.3.3.2 Problem of consecutive flicker frames

So far in our analysis of the Intensity Mean Averaging method, it has been assumed that in an image sequence, the flicker frame is always sandwiched between two or more consecutive good frames. Obviously, this is a highly idealized situation which seldom happens in practice. The more likely scenario is one that has a random mix of good and flicker frames.

Shown in figure 3.10 is an example where none of the flicker frame is sandwiched between consecutive good frames. This occurs when there are two or more consecutive flicker frames in the image sequence.

$f_0 \quad f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5 \quad f_6 \quad f_7$

Fig. 3.10 An example of consecutive flicker frames (G – good frame, F – flicker frame)

Under this situation, the Intensity Mean Averaging method developed above can still be applied to correct the flicker frames. Using this particular example for illustration, the steps to be taken are as follows:

- (i) correct flicker frame f_1 using good frames f_0 and f_4
- (ii) correct flicker frame f_2 using f_1 [treated as a good frame as it has been corrected in (i)] and frame f_4
- (iii) correct flicker frame f_3 using f_2 [treated as a good frame as it has been corrected in (ii)] and frame f_4
- (iv) frames f_5 and f_6 can be corrected using the same approach, starting with using the two good frames f_4 and f_7

The above method will not work if the first frame and last frame of the sequence being processed are not good frames. One way to get around the problem is to manually correct these two frames first; once this is done, the above method can be applied as before.

Another situation that will cause problems is when there is a long stretch of consecutive flicker frames. A possible solution to this problem is to manually correct a few frames within this stretch first to break it down to more manageable short stretches of consecutive flicker frames before applying the above method.

3.3.4 Experimental results and analysis

The same three test sequences used earlier for testing the Histogram Mapping method will be used again here for testing the proposed method. As before, the 40-frame Lady test sequence will be covered in detail for illustration purpose.

As pointed out in section 3.3.3.1, in the Intensity Mean Averaging method, the very first step in the correction process is to identify the flicker frames. The 40-frame Lady sequence has a frame average intensity distribution as given in Table 3.1, where the median intensity mean is 81.77. Using the criterion that any frame that has an average intensity value 1.5 above or below this median value is taken as a flicker frame, there were altogether seven flicker frames identified, as highlighted by an asterisk next to the frame number in the table.

Table 3.1 Frame intensity mean distribution in the Lady sequence

Frame #	0*	1	2	3	4	5	6*	7	8	9
Average intensity	79.82	80.33	81.27	80.77	81.60	82.07	87.36	82.19	81.84	81.47
Frame #	10*	11	12	13	14	15	16	17	18	19
Average intensity	83.32	82.50	81.79	81.79	82.17	81.23	81.09	82.22	82.12	81.58
Frame #	20	21*	22*	23	24	25	26*	27	28	29
Average intensity	82.08	85.86	83.33	82.18	81.51	82.43	85.17	81.36	81.20	81.80
Frame #	30	31	32	33	34	35	36	37*	38	39
Average intensity	80.81	80.45	80.27	81.74	81.14	81.39	81.48	88.52	82.22	81.73

The correction process will now be illustrated by studying the correction of a particular frame, followed by a discussion on the correction results of the whole 40-frame Lady sequence.

Frame 10, which is a frame with global flicker sand-witched between two good frames, will be used for illustration. It has an average intensity level of 83.32, which is 1.55 above the median intensity mean ($=81.77$) of the sequence and so qualified as a flicker frame. Its intensity mean is also considerably higher than those of its neighbors, whose average intensities are respectively at 81.47 and 82.50. According to equations 3.5 and 3.6, this represents a flicker gain of 1.34, and the intensity at each pixel in frame 10 can now be corrected using equation 3.8. Shown in figures 3.11 (a), (b) and (c) are respectively frame 9, 10 and 11 in the original sequence, and figure 3.11 (d) is the corrected frame 10. Note that the corrected frame 10 has an average intensity of 81.33, which is considerably lower than its original value of 83.32 and is closer to those of its neighbors frame 9 and frame 10 (respectively at 81.47 and 82.50).

Figure 3.11 (a) original frame 9 [good], (b) original frame 10 [flicker], (c) original frame 11 [good] and (d) frame 10 [corrected]

The remaining six flicker frames (frame 0, 10, 21, 22, 26, and 37) were corrected in the same manner and their average intensities were successfully brought closer to those of their immediate neighbors. Figure 3.12(a) shows the intensity means curves before and after correction. The original intensity mean curve has a variance value of 3.11, compared to 0.37 for that of the corrected sequence, representing a reduction of 88.1%. This means the proposed method has been effective in reducing fluctuations in frame intensity mean in the original sequence, as evident by the removal of all the intensity “spikes” in the intensity mean curve of the corrected sequence. On the other hand, visual checking on the corrected frames also confirms the finding above.

Fig. 3.12 (a) Intensity mean curves of the Lady sequence

Fig. 3.13 Intensity mean curves of the Tunnel sequence

Fig. 3.14 Intensity mean curves of the Lantern sequence

As in the Histogram Mapping method, the proposed method was also tested with more natural flicker sequences. Again only two of them, the Tunnel sequence and Lantern sequence will be selected here for illustration. They are respectively given in figure 3.13 and figure 3.14.

3.3.5 Discussions

Test results obtained earlier show that intensity mean curve of the corrected Lady sequence has a variance just 11.9% of the original. The corresponding figures for the Tunnel and Lantern sequences are respectively 26.1% and 28.3%. This shows that the Intensity Mean Averaging method tends to perform better when the sequence under correction contains “spiky peaks” (i.e. large intensity fluctuations), such as in the Lady sequence. When this is not the case, then the reduction in intensity fluctuation will be much reduced, as in the case of the correction of the other two sequences.

Similar to the Histogram Mapping method, the Intensity Mean Averaging method shares the same limitation in that it is only effective in dealing with global flicker. When the method is applied to frames that suffer from local flicker, it simply will not work. Frame 6, which is known to contain local flicker, is used again for illustration, and the results are given below in figure 3.15.

Figure 3.15 (a) original frame 5 [good], (b) original frame 6 [local flicker], (c) original frame 7 [good] and (d) corrected frame 6

As expected, the local flicker region (shown within the dotted-lined box) that appears as a relatively bright horizontal bar that runs across the lower chin and neck portion of the lady's head) in frame 6 before correction [figure 3.15(b)] is still clearly visible in the corrected frame [figure 3.15 (d)].

3.4 Conclusion

It has been shown in this chapter that both the Histogram Mapping method and the Intensity Mean Averaging method are effective in correcting global flicker but not local flicker. With regard to the relative flicker correction performance of the two methods based on test results of the Lady, Tunnel and Lantern sequences, the Histogram Mapping method was clearly more effective in flicker correction. This is supported by the fact that the intensity mean curves of the corrected sequences by the Histogram Mapping method were found to be much smoother than those by the Intensity Mean Averaging method.

Quantitatively, as can be seen in Table 3.2, for the Intensity Mean Averaging method, the intensity mean curve of the corrected Lady, Tunnel and Lantern sequences have respectively variance values of 11.9%, 26.1% and 28.3% relative to that of the original. These figures are considerably larger than those for the Histogram Mapping method, which are just respectively at 1.6%, 0.5% and 5.0%. This strongly suggests that between the two methods, the Histogram Mapping method is more capable in the reduction of intensity fluctuations, hence more effective in global flicker removal.

Table 3.2 Variances of intensity mean curves

Flicker correction method	Variance of intensity mean curve		
	Lady sequence	Tunnel sequence	Lantern sequence
No Correction	3.11	3.79	0.60
Histogram Mapping method	0.05 (1.6%) [†]	0.02 (0.5%)	0.03 (5.0%)
Intensity Mean Averaging method	0.37 (11.9%)	0.99 (26.1%)	0.17 (28.3%)

[†]Note: Figure inside bracket () is the percentage value relative to that with no correction. This notation is used for all other variance tables in the rest of this thesis.

When comparison is made in terms of algorithm complexity and computational overhead, the Intensity Mean Averaging method is however a clear winner. This is due to the fact that in this method only the flicker frames are required to be processed, compared to all frames that are required in the other method.

Real flicker film sequences seldom contain only global flicker. An obvious question to ask at this juncture is: are there ways to modify the two methods developed in this chapter so that they can also handle local flicker? This question will be addressed in detail in the next chapter.

Chapter 4

Local Flicker Correction

4.1 Introduction

The work developed in chapter 3 is modified in this chapter with the aim to overcome the shortcoming in the two methods developed there, i.e., their inability to handle local flicker. The proposed method, known as the “Local Enhancement Technique”, is incorporated into the earlier work and produces two new improved methods respectively known as the “Histogram Mapping with Local Enhancement” method and “Intensity Mean Averaging with Local Enhancement” method. Test results with natural flicker sequences show varying degree of improvement in these two methods with the incorporation of local enhancement.

4.2 Local enhancement – the basics

The two methods developed in chapter 3 have been shown to be effective for correcting global flicker only. Essentially in these methods, image pixels are modified by using a transformation function or intensity mean values based on the gray-level distribution over an entire image. The underlying assumption made is that any flicker, if present, runs uniformly over the whole frame. In practice, flicker in real films seldom occurs in this manner. More often than not, flicker appears in different areas of a frame, and typically varies in degree of severity. Flicker of this nature is known as local flicker, and new ways have to be found to deal with this problem effectively.

A local enhancement approach is proposed here to solve this problem. The idea is actually quite simple; instead of dealing with the whole frame all in one go, correction is done on

each small block formed by sub-dividing the frame into a suitable number of equal-sized rectangular blocks. For a frame with local flicker, it is known that flicker will appear in different strengths over different areas of the frame, and the spatial spread of flicker is very gradual. Hence, for a suitably small block in the flicker frame, it can be quite justifiably assumed that within that small block flicker is almost constant. As a result, methods developed in chapter 3 for global flicker, which assume uniform flicker across the entire frame, can now be applied to each small block where flicker can be approximately taken as uniform. In the coming sections, more details will be given on how the local enhancement technique can be incorporated to address the local flicker problem.

4.3 Histogram mapping with local enhancement

The method is best illustrated using figure 4.1, which shows a flicker image sequence of k frames.

Fig. 4.1 A k -frame image sequence with each frame divided into 64 blocks

The basic idea behind the new method is almost identical to that in the Histogram Mapping method described in chapter 3 section 3.2. As before, a reference frame (either originally perfect or has been flicker-corrected earlier) is first chosen to be used for the correction of the flicker frame. However, instead of working on an image frame basis, the new method works on an image block basis. Using figure 4.1 for illustration, let us assume frame f_0 is the reference frame and frame f_1 is the flicker frame to be corrected.

The two frames are first divided into certain number (64 in the example of figure 4.1) of equal-sized image blocks. Once this is done, flicker correction can be started by applying the Histogram Mapping method on image blocks of the two frames, starting from block $f_0(0,0)$ of frame f_0 and block $f_1(0,0)$ of frame f_1 (the two integers within the bracket denote respectively the row and column positions of the block).

Briefly, the block-based correction process can be described as follows:

- (i) Starting from the first block pair $[f_0(0,0)$ and $f_1(0,0)]$, get the cumulative distribution function $[V = G(z)]$ for the block in the reference frame and that $[S = T(r)]$ for the block in the flicker frame
- (ii) Using equation 3.3, correct image block in the flicker frame based on the corresponding image block in the reference frame
- (iii) Repeat steps (i) and (ii) for the next corresponding block-pair until all blocks are covered

After frame f_1 has been corrected, it will be used as a reference frame for the correction of the next frame f_2 . The process just repeats itself until all frames are corrected.

4.3.1 Experimental results and analysis

To test out the effectiveness of the Histogram Mapping with Local Enhancement method in correcting local flicker, frame 6 in the 40-frame Lady sequence will be an ideal candidate as it is known to contain a prominent local flicker region (as indicated by the brighter region within the dotted-box in figure 4.2(b)). As before, frame 5 which is a good frame will be used as the reference frame for correction. Using block size of 90×72 pixels, each of these two 720×576 -pixel frames is divided into a 64-block frame, and correction is carried out on a block-by-block basis as illustrated earlier in section 4.3.

The correction result is unfortunately rather unsatisfactory in the sense that while it is able to reduce local flicker, it nevertheless also at the same time introduces some undesirable side-effects that are quite unacceptable. A detail discussion will now be given below.

Given in figures 4.2 (a), (b) and (c) are respectively frame 5 (good and used as reference), frame 6 (flicker) and frame 6 (corrected) of the flicker Lady sequence. Frame 5 has an intensity mean of 82.07 whereas the flicker frame 6 has a mean of 87.36, which is considerable higher. After correction, the intensity mean of frame 6 has dropped to 83.75, which is much closer to that of frame 5, implying a reduction in flicker of the whole frame.

(a)

(b)

(c)

Fig. 4.2 (a) frame 5 [reference], (b) frame 6 [local flicker] and (c) frame 6 [corrected]

The area affected by local flicker [region within the dotted box in figure 4.2 (b)], when checked visually, is seen to have become darker in the corrected frame in figure 4.2 (c) and is thus closer in intensity to the same region in the reference frame. In short, this means local flicker has been successfully reduced. This observation is supported by intensity mean measurements in the local flicker region in frame 6 before and after correction, which give values of 111.10 (i.e. brighter) and 101.90 (i.e. darker) respectively. The intensity mean in the corresponding region in frame 5 is 98.69. This means that for this local flicker region, the intensity mean (=101.90) in the corrected frame 6 becomes much closer to that (=98.69) of the reference frame 5. In fact the intensity mean difference is now only 3.21 (=101.90-98.69), compared to 12.41 (=111.10-98.69) before correction. Since the intensity mean difference in the affected area has been so much reduced after correction, it can be concluded that local flicker has been effectively removed.

Despite the fact that local flicker is considerably reduced, the correction result obtained is far from being acceptable due to two glaring short-comings: blocky effect and distortion in motion regions.

The blocky effect is an easier problem to solve as there are established ways to deal with it. One obvious way is to choose smaller size blocks and then apply a low-pass filtering process (if necessary) to smoothen out boundaries of the blocks.

The distortion problem on the other hand needs more investigation before a solution can be found. A detailed analysis of this problem will now be given below.

Distortion is seen appearing in two forms in the corrected frame, and it turns out that both are in fact related to motion. As can be seen in figure 4.2, two types of distortion can be identified in the corrected frame; the first appears as “grayish speckles” in the corrected block, whereas the other one appears in the form of “image smudge”

Let us look into the distortion problem by examining the grayish speckles first. The image block in which this type of distortion shows up most prominently is the block with row and column location of (5, 5) in the corrected frame, for this reason, it will be named as block $\Omega_{5,5}$. A detailed explanation of the origin of grayish speckles will now be given below.

Fig. 4.3 (a) frame 5 (reference), (b) $\Omega_{5,5}$ (magnified)

Fig. 4.4 (a) frame 6 (flicker), (b) $\Omega_{5,5}$ (magnified)

Fig. 4.5 (a) frame 6 (corrected), (b) $\Omega_{5,5}$ (magnified)

It is observed that going from frame 5 to frame 6, the lady in the film sequence actually undergoes a slight body rotation movement. As a result, a larger portion of the area in $\Omega_{5,5}$ gets occupied by the lady's body when going from frame 5 to frame 6. This is clearly demonstrated by the larger dark area found in $\Omega_{5,5}$ of frame 6 [figure 4.4(b)] and the smaller dark area found in $\Omega_{5,5}$ of frame 5 [figure 4.3(b)].

When the block $\Omega_{5,5}$ in frame 6 undergoes correction, $\Omega_{5,5}$ in frame 5 will be used as a reference. Given in figure 4.6 (a) and (b) are respectively the histograms of $\Omega_{5,5}$ in frame 6 and frame 5 whereas those in 4.7(a) and (b) are their corresponding cumulative distribution functions (CDF). Note that for a better view of the graphs, the intensity range used in the x-axis has been truncated from 0-255 to 0-120; this is acceptable as there are no pixels in $\Omega_{5,5}$ with intensity level beyond 120.

The histograms of block $\Omega_{5,5}$ in frames 5 and 6 share a common characteristic in that there are two distinct peaks in the histograms, with one at a lower intensity level and the other at a higher intensity level. This is consistent with the image in $\Omega_{5,5}$ because the block essentially contains two regions, one being darker and the other brighter – the darker region corresponds to the peak at low intensity level whereas the brighter region corresponds to the higher intensity level.

It has been said earlier that $\Omega_{5,5}$ in frame 6 has a larger darker region than that in frame 5. This is clearly reflected in the histograms in figure 4.6. As can be seen, the low-intensity peak in the histogram of $\Omega_{5,5}$ in frame 6 has a larger area underneath than that of the corresponding peak of $\Omega_{5,5}$ in frame 5.

An explanation as to why there are distortions appearing in $\Omega_{5,5}$ of the corrected frame 6 can now be given. Distortions in the form of grayish speckles, appear mainly on the top right-hand corner of $\Omega_{5,5}$ in the corrected frame 6 as shown in figure 4.5(b). These gray speckles are actually some pixels that are originally in the dark region of $\Omega_{5,5}$ in frame 6 that get mapped into brighter pixels after the histogram mapping process.

As can be seen in figure 4.7(a), a pixel at intensity level 50 (which is in the dark region) will get mapped into a pixel at intensity level 85 (which is much brighter) after mapping. As a matter of fact, for pixels that are at intensity level close to 50, they will all get

mapped into pixels at intensity level close to 85; i.e., these pixels will become much brighter after mapping.

Fig. 4.6 Histogram of $\Omega_{5,5}$ in (a) frame 6 (flicker), (b) frame 5 (reference)

Fig. 4.7 CDF of $\Omega_{5,5}$ in (a) frame 6 (flicker), (b) frame 5 (reference)

It can be said that whenever there is such a scenario, i.e., a block with predominantly two distinct dark and bright regions, and at the same time there is a significant change in the proportions of these two regions (e.g. due to motion) in the block concerned between the reference frame and the frame under correction, then this type of distortion will always be introduced in any block-based histogram mapping correction method.

The second distortion problem, i.e., image smudge, is seen occurring around the hands of the lady figure, where large hand movement is known to exist. With no motion compensation taken into consideration at the moment, what actually has happened is incorrect blocks in the reference frame have been used in the correction of all the blocks affected by motion in the flicker frame. It is not surprising that these blocks corrected are all smudged.

The above-mentioned problems with regard to the Histogram Mapping with Local Enhancement method also occurred when the method was applied to other test sequences such as the Tunnel and Lantern sequences. With no apparent solution to the “grayish speckles” type of distortion problem in sight, it was decided that the proposed method was not worth pursuing any further, and attention should be shifted to applying local enhancement to the Intensity Mean Averaging method and see if it could work better.

4.4 Intensity mean averaging with local enhancement

This method is basically the same as the Intensity Mean Averaging method covered in chapter 3 section 3.3. As in section 4.3, flicker correction is now carried out on an image block basis instead of an image frame basis. As before, the image frame will have to be divided into equal-sized image blocks first as illustrated earlier in figure 4.1.

The rest of the steps are very straight-forward. Starting from the top left corner block, each block of the flicker frame is corrected by applying the Intensity Mean Averaging method until all blocks are covered. When this is done, the same process is performed on the next flicker frame until all flicker frames are corrected eventually.

4.4.1 Experimental results and analysis

To test out the effectiveness of the Intensity Mean Averaging with Local Enhancement method in correcting local flicker, correction of flicker frame 6 (using two neighboring good frames, frame 5 and frame 7 for correction) in the 40-frame Lady sequence will be used again for illustration purpose. To avoid introducing blocky effects as had happened earlier in the Histogram Mapping with Local Enhancement method, a much smaller block-size of 8x8 pixels (compared to 90x72 pixels used earlier) will now be used. As a result, each frame will now be divided into 6480 8x8-pixel blocks (compared to 64

90x72-pixel blocks earlier); this means the new block is now about 100 times smaller than the old block used earlier.

Initial evaluation of the correction result seems to suggest that the method is substantially better than that of the Histogram Mapping with Local Enhancement method. Visual checking shows that the local flicker region is no longer visible in the corrected frame; and unlike the previous method, there is no sign of any “grayish speckles” type of distortion in the corrected frame. A more thorough analysis into this method will now be given below.

Figure 4.8 (a) frame 5 [good], (b) frame 6 [local flicker], (c) frame 7 [good] and (d) frame 6 [corrected]

In the original sequence, frames 5, 6 and 7 have intensity means of 82.07, 87.36 and 82.19 respectively. The intensity mean of frame 6 is much higher than those of its two

neighbors, giving a flicker gain of 5.23. After correction, the corrected frame 6 has an intensity mean of 81.63, which is substantially lower than its original value of 87.36 and comes close to those of its neighbors frame 5 and frame 7 (respectively at 82.07 and 82.19). Obviously, flicker in the whole frame of frame 6 has been substantially reduced. Shown in figures 4.8 (a), (b) and (c) are respectively frame 5, 6 and 7 in the original sequence, and figure 4.8 (d) is the corrected frame 6.

Visual checking shows that the local flicker region within the dotted box in figure 4.8 (b) is successfully removed in the corrected frame. Quantitative evidence is now needed to see if this observation is true. This can be done by measuring intensity mean in the area affected by local flicker in frame 6 before and after correction, and then compare these values with those of its two immediate neighbors. The results obtained are as follows:

Before correction, the local flicker region in frame 5, 6 and 7 have respective intensity mean values of 98.69, 111.10 and 99.29. This represents an average intensity difference of 12.11 $[=111.1-0.5 \times (98.69+99.29)]$ between frame 6 and its two neighbors. After correction, the local flicker region in frame 6 has an intensity mean value of 98.41, suggesting that the average intensity difference is now only 0.58 $[=98.41-0.5 \times (98.69+99.29)]$, i.e., only 4.8 % of the original intensity difference. Obviously, these figures lend great support to the argument that flicker in the local flicker region in frame 6 has been successfully removed.

The effectiveness of the Intensity Mean Averaging with Local Enhancement method has just been demonstrated through the work of correcting frame 6 of the Lady sequence. Let us proceed now to correct the rest of the remaining seven flicker frames (frame 0, 10, 21, 22, 26, and 37) and evaluate the overall flicker correction performance on the whole sequence. The correction results are presented graphically in figure 4.9, which shows the intensity means curves before and after correction.

The original intensity mean curve has a variance value of 3.11, whereas that after correction is only 0.37, which is just 11.9% of the original value. These figures confirm the effectiveness of the proposed method in flicker reduction, and its effectiveness is evident by the removal of all the intensity “spikes” in the intensity mean curve of the corrected sequence. As before, visual checking on the corrected frames also confirms the finding above.

The proposed method was further tested with more natural flicker sequences, and the results for two of them, the Tunnel and Lantern sequences are given respectively in figures 4.10 and 4.11 below.

Fig. 4.9 Intensity mean curves of the Lady sequence

Fig. 4.10 Intensity variance curves of the Tunnel sequence

Fig. 4.11 Intensity mean curves of the Lantern sequence

4.4.2 Discussions

Test results obtained earlier show that intensity mean curve of the corrected Lady sequence has a variance just 11.9% of the original. The corresponding figures for the Tunnel and Lantern sequences are respectively 21.4% and 41.7%. These figures are very similar to those for the Intensity Mean Averaging method obtained in chapter 3, indicating that the proposed method tends to perform better when the sequence under correction contains large variation of intensity fluctuations, such as in the Lady sequence.

Despite all the encouraging results, this method however, just like the Histogram Mapping with Local Enhancement method, fails to handle motion regions. Although no “grayish speckles” type of distortion is noticed anywhere, there is still “image smudge” type of distortion found in areas near the lady’s hands which are known to be moving. This is illustrated in figure 4.12.

As highlighted in the circled region showing the lady’s hands where large movement is known to exist, distortion in the form of “smudged” checker blocks is clearly visible. Distortion is also visible, though less prominently, in another motion region near the

lady's right shoulder. The distortion there appears as a narrow strip of white checker blocks stretching from the top of the lady's collar region down to her upper right arm.

Fig. 4.12 Distortion in motion regions in corrected frame 37 of the Lady sequence

4.5 Conclusion

In this chapter, local enhancement is incorporated into the two methods developed in Chapter 3 by dividing frames into small blocks and correcting each block individually. The two resulting new methods are respectively known as Histogram Mapping with Local Enhancement method and Intensity Mean Averaging with Local Enhancement method.

The Histogram Mapping with Local Enhancement method is found to be able to remove local flicker. However, the correction result is rather unsatisfactory because distortions in the form of "grayish speckles" and "image smudge" are found to be present, and blocky effect is also introduced.

The Intensity Mean Averaging with Local Enhancement method is also found to be able to remove local flicker and unlike the earlier method, there is no "grayish speckles" type of distortion. Blocky effect is successfully avoided by choosing a much smaller 8x8-pixel

block size. Intensity mean curves of the Lady, Tunnel and Lantern sequences are found to be much smoother after correction, indicating the effectiveness of the proposed method.

What is the effect in the intensity mean curve of the corrected sequence after local enhancement is incorporated into the Intensity Mean Averaging method? Not much indeed. As can be seen in row 4 of Table 4.1, the variances of intensity mean curves after correction have not changed much after the incorporation of local enhancement, indicating that the same flicker correction quality is being maintained in the new method.

Table 4.1 Variance of intensity mean curves

Flicker correction method	Variance of intensity mean curve		
	Lady sequence	Tunnel sequence	Lantern sequence
No Correction	3.11	3.79	0.60
Intensity Mean Averaging method	0.37 (11.9%)	0.99 (26.1%)	0.17 (28.3%)
Intensity Mean Averaging with Local Enhancement method	0.37 (11.9%)	0.81 (21.4%)	0.25 (41.7%)

In areas of algorithm complexity, computational overhead and number of frames needed to be processed, the Intensity Mean Averaging with Local Enhancement method is, as expected, better than the Histogram Mapping with Local Enhancement method for the same reasons as given in section 3.4 of chapter 3.

So in conclusion, it can be said that between the two methods, the Intensity Mean Averaging with Local Enhancement method is an overall better performer. However, there is still a shortcoming in this method that needs to be addressed: its inability to handle motion regions. This problem will be looked into in the next chapter to see if a solution can be found.

Chapter 5

Motion-compensated Local flicker Correction

5.1 Introduction

The two flicker correction methods developed in chapter 4 are found to be unable to handle motion regions and as a result flicker correction in those regions is less than satisfactory. In this chapter, a comprehensive study on the topic of motion estimation and compensation is presented. Motion compensation is then incorporated into the Intensity Mean Averaging with Local Enhancement method to test for its effectiveness in correcting flicker in motion regions.

5.2 Motion estimation and compensation – the basics

An image sequence, such as a motion picture, consists of a set of images of a scene recorded at regular intervals in time [41]. For example, TV broadcast in PAL [75] format runs at 25 frames per second, which means that the time interval between consecutive frames is $1/25$ second. Due to the high frame rate, the recording of each image in a motion picture will occur more rapidly than the change of information in the scene; this is in general true except for a frame that contains scene change. The implication of this is that there exists a large amount of similar image information in consecutive frames of a sequence. This high temporal correlation between frames, as it turns out, is a property that can be exploited for motion picture restoration so long as ways can be found to determine the motion vectors between corresponding points in consecutive frames. It should be pointed out that this spatio-temporal technique actually forms the basis of most of the existing algorithms in motion picture restoration.

Motion estimation, which may refer to image-plane motion (2-D motion) or object motion (3-D motion) estimation, is one of the fundamental problems in digital video processing. It has been the subject of much research effort and continues to draw considerable research interests in recent years. Due to the substantial amount of work done in this area [42-46], a detailed coverage on this topic will probably fill many volumes and this is not our intention here. Instead, what shall be attempted here is to give a general coverage wide enough to cover the more important methods while the focus will be on those that will be used later in our work.

5.2.1 Image sequence modeling and the occlusion problem

Currently the prevailing thinking with respect to image sequences revolves around a translational model [41] which can be described by equation 5.1 below:

$$I_n(\mathbf{x}) = I_{n-1}(\mathbf{x} - \mathbf{d}_{n,n-1}) \quad (5.1)$$

where $I(\mathbf{x})$ is the intensity of the pixel at the location given by the position vector $\mathbf{x}$, $n-1$ and n respectively denote the previous and current frame, and $\mathbf{d}_{n,n-1}$ is the displacement vector between the pixel at location $\mathbf{x}$ (in frame n) and its original location (in frame $n-1$). Using this model, the intensity at each pixel in the current frame can be estimated from that of the corresponding pixel in the previous frame. All that is needed is to find the motion vector $\mathbf{d}$.

Fig. 5.1 Typical motion types across successive frames

Figure 5.1 shows the three different forms of motion typically encountered in an image sequence. If the motion is small enough in-between frames, then all three types of motion can be treated as translational, and as a result all of them can be described by equation 5.1.

The model described by equation 5.1 forms the basis of most of the current techniques for image sequence processing; however, there are situations under which problems can occur due to difficulties in determining motion vectors.

Motion vectors, as a matter of fact, can be determined only if the image area concerned can be found in both frame n and frame $n-1$. However, this may not always be possible; one well-known example is the occlusion problem. As discussed previously in chapter 2 section 2.4.1, occlusion refers to the covering/uncovering of surfaces due to the movement of objects in a frame [42]. The concept of covered and uncovered background due to occlusion can be illustrated in figure 5.1 where the object represented by the solid dark block undergoes three different types of motion. In both translation and rotation, when going from frame $n-1$ to frame n , the area labeled 'U' is uncovered in frame n , while the area labeled 'O' is occluded (covered) in frame n . For these areas there can be no motion vector since no match can be found between frame $n-1$ and frame n . This is also the case in zoom, where the occluded area in frame n has no match in frame $n-1$. Motion estimation problem caused by occlusion is an issue that will be addressed later in our work in chapter 8 section 8.4.2.

When the image area concerned can be found in both frame n and frame $n-1$, then finding the motion vectors is quite a manageable task. Many different methods have been proposed and in essence they are all about finding ways to solve equation 5.1 for the unknown $\mathbf{d}$. Block-based motion estimation is among the most popular approaches [42]. A substantial amount of work using a linearization of equation 5.1 to solve for the motion vector directly has also been reported [43]. Other works include methods based on Gibbs distributions [47] and Bayesian methodology [48]. In the coming sections, a review of some of the more commonly used block-based motion estimation methods will be presented; in particular, attention will be given to the Block Matching method as it is the method that will be used in our flicker correction work in later chapters.

5.3 Block-based motion estimation

Among all known methods, block-based motion estimation is undoubtedly the most popular one. It has been adopted in the international standards for digital video compression, such as H.261 [42] and MPEG 1-2. In addition, it is also adopted in the coding of CD-ROM applications, HDTV [77] standards, and a large number of other digital video applications.

The following sections will describe three common block-based motion estimation methods, namely Block Matching (BM), Overlapped Block Matching (OBM) and Multi-resolution Block Matching (MRBM).

5.3.1 Block Matching (BM) method

The most popular and probably also the most robust technique to date for motion estimation is the BM method [42,49,50].

In this method, image in frame n is in general divided into many equal-sized small blocks and an assumption is made that pixels in each of these blocks undergo the same translational motion. Each small block is processed in turn to find the motion vector through certain matching process. The motion vector is determined by matching the block in frame n with a set of blocks within a pre-defined search space in the neighborhood of the block but in frame $n-1$.

The basic idea of block matching is best illustrated by figure 5.2. The displacement (or motion vector $\mathbf{d}$) of the block Ω of size $N_1 \times N_2$ in frame n (the current frame) centered at location $\mathbf{x}$ is determined by searching for the best matching same-size block within a search space P in frame $n-1$ (i.e. previous frame). The search space is usually a rectangular area centered at location $\mathbf{x}$ and of size $(N_1+2S_1) \times (N_2+2S_2)$, where S_1 and S_2 are the maximum displacement estimated in the x and y directions. There are many methods that can be used for determining the best matching block; the two most commonly used are the Mean Squared Error (MSE) method and the Mean Absolute Difference (MAD) method.

Determining the best matching block hinges on minimizing an error function $E(\mathbf{x}, \mathbf{v})$ defined in equation 5.2 below:

$$E(\mathbf{x}, \mathbf{v}) = I_n(\mathbf{x}) - I_{n-1}(\mathbf{x} - \mathbf{v}) \quad (5.2)$$

where $\mathbf{x}$ represents the position vector within the image block Ω , $n-1$ and n denote respectively the previous and current frame, and $\mathbf{v}$ is some displacement vector. The best matching block will be given by the vector $\mathbf{v}$ that results in a minimization of $E(\mathbf{x}, \mathbf{v})$, and the displacement vector $\mathbf{v}$ thus found is known as the motion vector $\mathbf{d}$.

Fig. 5.2 Block matching

In the MAD method, the error function is defined as [42]:

$$MAD(V_H, V_V) = \frac{1}{N_1 N_2} \sum_{(x,y) \in \Omega} |I_n(x,y) - I_{n-1}(x - V_H, y - V_V)| \quad (5.3)$$

where Ω is a $N_1 \times N_2$ block centered at location (x,y) , and (V_H, V_V) is a set of candidate motion vectors. The estimate of the motion vector is taken to be the value of (V_H, V_V) that minimizes the MAD.

The full motion search in MAD is computationally intensive, which requires $2S_1 \times 2S_2$ number of matches. However, due to its robustness and ease of implementation, it is widely used in VLSI [78] implementations and many digital image processing algorithms.

5.3.2 Overlapped Block Matching (OBM) method

The OBM method [51] can be considered as a refined version of the BM method; the following will help explain why this is so.

In order to get a reliable and accurate motion estimate, it is crucial that the size of blocks for block-matching is chosen correctly. Since the image sequence to be processed is always degraded, the estimate will be unreliable and more affected by noise if small blocks are used. On the other hand, if large blocks are used, then the estimate will not be accurate because within a large area, the assumption of a single motion vector may not hold as different areas may each have its own unique movement. The OBM method is able to circumvent these problems as will be illustrated below.

Fig. 5.3 OBM method illustration

As shown in figure 5.3, the original $N_1 \times N_2$ search block is replaced by an enlarged $E_1 \times E_2$ block where $E_1 > N_1$ and $E_2 > N_2$, and both centered at (n_1, n_2) . Similarly, the search space is also enlarged to $(N_1 + 2P_1) \times (N_2 + 2P_2)$ where $P_1 > S_1$ and $P_2 > S_2$. Block-matching will now be carried out using the $E_1 \times E_2$ block within the enlarged search space. The matching criterion used is MAD, the same as that used in the BM method. When the motion vector for the $E_1 \times E_2$ block is determined, it will be assigned to the original $N_1 \times N_2$ block.

The OBM method is able to determine motion vectors more accurately in a noisy environment when compared to the BM method. For this reason, it will be a better choice when dealing with films that are badly degraded. However, one should realize that this method is also more computationally intensive as complexity is increased due to the introduction of enlarged blocks.

5.3.3 Multi-resolution Block Matching (MRBM) method

In our discussion in section 5.3.2, it was mentioned that the BM method was not very reliable in dealing with image sequences that are very noisy, and the OBM method would be a preferred choice instead. Similarly, there is another situation under which the BM

method performs rather poorly: when large displacements are present in the image sequences. This is because in the BM method, to effectively detect a large displacement, a large search space has to be used for searching, and this will substantially increase the amount of computations involved.

The MRBM method [52] is widely accepted as the most practical way to deal with this problem effectively. In the MRBM method, frames in a hierarchical levels of resolution are first created for each frame through either low-pass filtering or sub-sampling. A pyramid representation of a single frame is depicted in figure 5.4. In this 3-level example, the full (highest) resolution image (i.e. the original image) is shown at the bottom as level 1, and the lowest resolution image is at level-3, the top level of the pyramid.

Fig. 5.4 Multi-resolution image representation

The basic idea of the MRBM method is to perform motion estimation at each level successively, starting from the lowest resolution. The lower resolution levels serve to determine a rough estimate of the motion vector using relatively larger blocks and search spaces. Once estimated, the motion vector at a lower resolution level is passed on to the next higher resolution level as an initial estimate, whereby smaller blocks and search spaces can be used for block-matching. The process continues until the highest resolution level is reached, using progressively smaller blocks and search spaces in each level along the way.

5.4 Motion-compensated intensity mean averaging with local enhancement

It has been shown in chapter 4 that the two methods developed there are not able to handle regions that are in motion. Such deficiency can be overcome if motion compensation is incorporated, as will be elaborated below.

From the work in chapter 4, it can be seen that the Intensity Mean Averaging with Local Enhancement method is computationally much less demanding and is much easier to implement than the Histogram Mapping with Local Enhancement method. At the same time, it can be envisaged that if motion compensation is to be incorporated, the former method would allow a much easier implementation. For this reason, it is decided that only the former method will be used to study the effect when motion compensation is incorporated.

A brief outline of the new method to be proposed here, i.e., the Motion-compensated Intensity Mean Averaging with Local Enhancement method, will now be given. Being a block-based method, the new method is similar to its predecessor in that it also requires the image frames to be divided into equal-sized rectangular blocks first. The major difference in the new method is that an extra step is needed which involves using a motion detection process to identify motion blocks. Once motion and non-motion blocks are identified, flicker correction can be started. For stationary blocks, the flicker correction process will be the same as that described in chapter 3 section 3.3.2; for motion blocks, they will be dealt with differently as will be shown later in section 5.4.2.1.

5.4.1 Motion detection using the block-matching method

The block-matching method has been chosen here for the detection of motion blocks. It is the more appropriate choice over the other two methods due to its simplicity and proven robustness.

5.4.1.1 Motion detection

Following the same argument as in chapter 3 section 3.3.2, a sequence of three frames as depicted in figure 5.5 will be used here to illustrate the motion detection process.

In figure 5.5, f_k is a flicker frame to be corrected using two neighboring good frames f_{k-1} (previous frame) and f_{k+1} (next frame). As before, all the three frames have been divided into equal-sized rectangular blocks; the task now is to identify motion blocks in the flicker frame.

Fig. 5.5 A flicker frame f_k sand-witched between two good frames f_{k-1} and f_{k+1}

A standard block-matching method using the minimum Mean Absolute Difference (MAD) criterion will be applied here for motion detection. Consider a block $\Omega_{m,n}$ in the flicker frame f_k for which a best match is to be found in the preceding good frame f_{k-1} . Let $MAD_{(k, k-1)}$ be the MAD between a block in the flicker frame f_k and a searched block in the preceding good frame f_{k-1} . If V_H and V_V are respectively the horizontal and vertical motion vector in a pre-defined search area, then the best matching block can be determined by finding the pair of V_H and V_V values such that the following $MAD(V_H, V_V)_{(k, k-1)}$ expression is minimized:

$$MAD(V_H, V_V)_{(k, k-1)} = \frac{1}{N_1 N_2} \sum_{(x, y) \in \Omega_{m, n}} |I_k(x, y) - I_{k-1}(x - V_H, y - V_V)| \quad (5.4)$$

where $I(x, y)$ is the intensity at location (x, y) , subscript k and $k-1$ respectively denote the current and preceding frame, and $\Omega_{m, n}$ represents the partitioned image block of size $N_1 \times N_2$ at the m^{th} row and n^{th} column.

The MAD associated with this best displaced match is called $MAD_{\min}$. Having defined $MAD_{\min}$, another term MAD_o , which is the MAD between a block in the flicker frame f_k and its non-displaced-location block in the preceding frame, will now be defined:

$$MAD_o = \frac{1}{N_1 N_2} \sum_{(x, y) \in \Omega_{m, n}} |I_k(x, y) - I_{k-1}(x, y)| \quad (5.5)$$

With both MAD_{min} and MAD_o defined, the algorithm that is to be used for motion detection will now be presented. The details are as given below.

Let us consider two consecutive frames containing a moving object with a south-east direction movement as shown in figure 5.6. The dashed lines within the frame are introduced to form a grid for easy position referencing.

Fig. 5.6 Two consecutive frames containing a moving object

In general, each frame is divided into a fixed number of blocks for processing. In the n^{th} frame shown above, three types of blocks can be identified: stationary blocks (in the white area), motion blocks (in the textured area) and uncovered blocks (in the pink area). An analysis on the values of MAD_o and MAD_{min} associated with each of these three types of block will now be given.

It has been observed that for stationary blocks, both MAD_o and MAD_{min} are small and close in values, hence the ratio MAD_o / MAD_{min} is approximately 1.

For motion blocks, MAD_o is very high and MAD_{min} is low compared to MAD_o . Hence the ratio of MAD_o to MAD_{min} is high.

For uncovered blocks, both MAD_o and MAD_{min} are large, and in general MAD_{min} is slightly smaller. This gives rise to a MAD_o / MAD_{min} ratio of slightly greater than 1.

In the event that the frames are not clean and contain noise, the above observations in general still hold true. These observations form the basis of the work done by J.M. Boyce

[37], who proposed a way to identify motion and stationary blocks in a given frame by checking respectively the values of MAD_0 and the ratio MAD_0 / MAD_{min} against two threshold values θ and λ as illustrated in figure 5.7 below. In our work here, one extra criterion is introduced to identify the uncovered blocks; i.e., if $MAD_0 > \theta$ and $\lambda > MAD_0 / MAD_{min} > \gamma$, then the block is an uncovered block, where γ is a threshold value greater than 1 but less than λ . This is illustrated in figure 5.8. Note that the three threshold values θ , λ and γ in general have to be chosen carefully in order to produce the desirable results. We will be investigating the optimization of these parameters in Chapter 7.

Fig. 5.7 Motion blocks and stationary blocks detection rule by J.M. Boyce

Fig. 5.8 Proposed rule for the detection of motion, stationary and uncovered blocks

The algorithm for detecting motion, stationary and uncovered blocks is illustrated in the flow-chart given in figure 5.9:

Fig. 5.9 Flow-chart for the detection of motion, stationary and uncovered blocks

5.4.1.2 Effect of flicker on motion detection

When the motion detection algorithm developed above was tested, it was found to be not very reliable in that it produced many ‘false’ detections. After some careful analyses, the problem was found to be caused by flicker. It is a known fact that in order for the block-matching method to work, the condition of constant luminance constraint must be satisfied [21]. However, the presence of flicker upsets this condition, and as a result the algorithm can no longer work properly.

A modification to equations 5.4 and 5.5 is needed to overcome this problem. The modification is done in this manner: the intensity mean of the block for each of the frames is first computed, and this mean is later subtracted from each pixel of the respective block. The rationale behind this is by removing any intensity variation effect caused by factors such as flicker, the constant luminance condition needed can be re-established. The modified equations are as given in the following page:

$$MAD(V_H, V_V)_{(k,k-1)} = \frac{1}{N_1 N_2} \sum_{(x,y) \in \Omega_{m,n}} \{ [I_k(x,y) - M_k] - [I_{k-1}(x - V_H, y - V_V) - M_{k-1}] \} \quad (5.6)$$

$$MAD_o = \frac{1}{N_1 N_2} \sum_{(x,y) \in \Omega_{m,n}} [I_k(x,y) - M_k] - [I_{k-1}(x,y) - M_{k-1}] \quad (5.7)$$

where M_k and M_{k-1} are respectively the intensity mean of the $N_1 \times N_2$ -sized block in the current and preceding frames.

Test results show that the modified block-matching algorithm is able to detect motion regions much more accurately in the presence of flicker, thus eliminating the earlier problem of false detection. We will use this algorithm again in Chapter 7 (see chapter 7 section 7.2.2.1).

5.4.2 Motion compensation

The discussions in sections 5.4.1.1 and 5.4.1.2 have been on detecting motion, stationary and uncovered blocks in frame f_k by comparing frames f_k and f_{k-1} . Since frame f_k is to be corrected using frames f_{k-1} and f_{k+1} in the proposed method, the same motion detection process must also be carried out by comparing frame f_k and frame f_{k+1} . If M_1 represents the set of motion blocks detected between frames f_k and f_{k-1} , and M_2 the set for those between frames f_k and f_{k+1} , then the motion blocks in frame f_k will be given by the set M which is the union of M_1 and M_2 , i.e., $M = M_1 \cup M_2$, where $\cup$ is the mathematical operator “union”. The same is true for the uncovered blocks. If the letter U is used to represent uncovered blocks and the subscripts 1 and 2 take on the same meaning as for the motion blocks, then $U = U_1 \cup U_2$. At the end of the motion detection process, all the image blocks in the flicker frame f_k will be separated into three groups: stationary blocks, motion blocks and uncovered blocks. For stationary blocks, the flicker gain m_{gain} can be computed as described earlier in chapter 3 section 3.3.2. However, the same computation cannot be applied to a motion block or an uncovered block simply because information from the corresponding block at the same location in the two neighboring frames is no longer relevant. The flicker gain has to be computed differently using an iterative interpolation process to be described in the next section.

5.4.2.1 Iterative interpolation

Shown in figure 5.10(a) is a flicker frame where the shaded blocks represent stationary blocks with flicker gains that have been computed earlier. The white blocks correspond to motion blocks or uncovered blocks with unknown flicker gain yet to be determined.

Fig. 5.10 Flicker gain computation using an iterative interpolation process

The iterative interpolation process is carried out as described below. For a motion/uncovered block (marked 'x' in figure 5.11) that has eight neighboring blocks, its flicker gain is computed by averaging the flicker gain of its neighbors provided there are at least five neighbors whose flicker gains are known. If this condition is satisfied, then the motion block will be assigned the computed flicker gain value. Otherwise this motion block will be given a miss in this round of interpolation and another motion/uncovered block will be taken to go through the same operation until all motion/uncovered blocks in the frame are covered. By repeating this dilation process an estimate for flicker gain can be assigned to all motion/uncovered blocks eventually, this is illustrated in figure 5.10 (b) –(d).

Fig. 5.11 A motion/uncovered block with eight neighbors

Once the flicker gain for all (stationary, motion and uncovered) blocks are obtained, flicker correction can be carried out using equation 3.8 as discussed earlier in chapter 3 section 3.3.2.

5.4.3 Experimental results and analysis

The 40-frame Lady sequence will again be used to verify the effectiveness of the proposed method. In particular, the key concern here is to check if motion compensation is able to overcome the short-coming uncovered in the Intensity Mean Averaging with Local Enhancement method developed in chapter 4 – its inability to handle motion regions.

Earlier in figure 4.12 of chapter 4, the lady in corrected frame 37 was shown to exhibit “image smudge” type of distortion after correction by the Intensity Mean Averaging with Local Enhancement method. Distortion occurs mainly around the hands of the lady, which are known to be in motion. Using the motion detection approach outlined in section 5.4.1.1, motion or uncovered blocks in frame 37 were successfully identified, as shown by the white color blocks in figure 5.12 below. As expected, areas around the lady’s hands are all in white.

Fig. 5.12 Motion blocks (in white) in frame 37

Figure 5.13 shows the corrected frame 37 when motion compensation is incorporated into the correction process. “image smudge” distortion in areas around the lady’s hands are now all gone, which clearly demonstrates the effectiveness of the new method.

Fig. 5.13 Frame 37 corrected by the proposed method

Let us now examine the general effectiveness in flicker correction of the proposed method by taking a look at the intensity mean curves shown in figure 5.14 below:

Fig. 5.14 Intensity mean curves of the Lady sequence

Just like in the Intensity Mean Averaging with Local Enhancement method, the spikes in the intensity mean curve after correction by the proposed method are all removed, leaving behind an intensity mean curve that is rather smooth. This suggests that flicker has been successfully removed.

As before, the proposed method was further tested with more natural flicker sequences, and the results for two of them, the Tunnel and Lantern sequences are given respectively in figures 5.15 and 5.16.

Fig. 5.15 Intensity mean curves of the Tunnel sequence

5.5 Conclusion

Comparing the variances of the intensity mean curves of the corrected sequences by the three Intensity Mean Averaging related methods in Table 5.1, it can be seen that for the Lady sequence, there is no difference among them -- the percentage variance value stays unchanged at 11.9%. For the Tunnel sequence, there is only a slight change (from 26.1% down to 21.4%) after local enhancement is incorporated (with or without motion compensation). This means that the quality of flicker correction (in terms of producing a smoother intensity mean curve) is roughly maintained when the three Intensity Mean Averaging related methods are applied to the Lady and Tunnel sequences. The only exception is the Lantern sequence, where the percentage variance value is seen increased from 28.3% to 41.7% when local enhancement is incorporated. Obviously, among the three, the proposed method emerges as the best method because all the shortcomings of the other two methods have now been successfully eliminated.

Fig. 5.16 Intensity mean curves of the Lantern sequence

Table 5.1 Variance of intensity mean curves

Flicker correction method	Variance of intensity mean curve		
	Lady sequence	Tunnel sequence	Lantern sequence
No Correction	3.11	3.79	0.60
Intensity Mean Averaging method	0.37 (11.9%)	0.99(26.1%)	0.17(28.3%)
Intensity Mean Averaging with Local Enhancement method	0.37(11.9%)	0.81(21.4%)	0.25(41.7%)
Proposed method	0.37(11.9%)	0.81 (21.4%)	0.25(41.7%)

Even though the proposed method has managed to overcome all the earlier problems, there are still areas of deficiency in it that render it not very useful as a serious tool for practical flicker correction.

The first drawback of the proposed method is that it falls short of being a fully automatic process, which is a major criterion for an acceptable practical flicker correction solution. Currently in the proposed method, human intervention is needed to identify the flicker frames within a given corrupted sequence before correction can be started. One may argue that this procedure can be automated; while this may be true but then if the corrupted sequence contains long stretch of consecutive flicker frames as discussed in chapter 3 section 3.3.3.2, then human intervention will be unavoidable.

The second drawback is that the proposed method while being able to reduce flicker, the amount of flicker reduced varies greatly depending on a number of factors, such as the size of the intensity threshold used to identify flicker frames, the distribution of flicker frames within the given corrupted sequence, the size of the intensity peaks and troughs in the intensity mean curve of the given corrupted sequence, and also on how frequently these peaks and troughs occur in the intensity mean curve. Test results on the three test sequences show that the percentage variance values for the intensity mean curves after correction vary between 11.9% (for the Lady sequence) to 41.7% (for the Lantern sequence). Other than being spread over a wide range, these values are also rather high compared to those obtained under other flicker correction methods. For example, our experience with the Histogram Mapping method shows that the values are consistently under 10%, with the best result at 0.5% (for the Tunnel sequence).

Such weaknesses of the proposed method are in fact inherent to all Intensity Mean Averaging related flicker correction methods and it appears that nothing much can be done about it. It is probably timely now to explore some new avenues to address the flicker correction problem.

Chapter 6

Flicker Correction – based on Flicker Parameters Estimation

6.1 Introduction

This chapter begins by looking into the Flicker Parameter Estimation (FPE) method which is based on an idea first postulated by Roosmalen [25] in 1997. In this block-based correction method, two flicker parameters are estimated for each image block after which they are then applied to correct the flicker image. Based on the same flicker model, an entirely different algorithm using least square minimization is developed here to serve as an alternative to the FPE method. Subsequently work is carried out to address a known problem in the FPE method, i.e., its inability in detecting motion regions properly which leads to unreliable flicker correction in these regions. A solution is successfully developed by incorporating into the FPE method a re-examination process [35] to detect the motion regions more accurately.

6.2 The flicker parameters estimation (FPE) method

In our earlier work on flicker correction, the two methods developed in chapter 3, namely the Histogram Mapping method and Intensity Mean Averaging method, are essentially effective for correcting global flicker. The work in chapter 4 managed to extend the Intensity Mean Averaging method to cover local flicker, but it still failed to handle motion regions. Further work in chapter 5 successfully overcame the motion region problem; however some remaining problems inherent in the Intensity Mean Averaging related methods, e.g., the need for human intervention, make it rather unsuitable for practical flicker correction. Because the problems are inherent in the method itself, it is very unlikely that any further work in this method can get around these problems.

A departure from all the previous methods is suggested in the search of a better solution to the flicker correction problem. The FPE method based on Roosmalen's idea in 1997 seems to be an avenue that warrants some exploration. The FPE method has been proven to be effective in handling both global and local flicker but it appears that there are still rooms for improvement. Literature survey shows that since 1997 after the publication of this method, only very limited related work has been produced, and many known problems in this method still remain unresolved, most notable of which is its inability to detect motion blocks accurately. Other problems such as scene change, occlusion, motion smear and non-translational motion etc., are also yet to be addressed. This suggests that much further work could possibly be done in this area. The main work in this thesis is in one way or another related to the FPE method. Our work has not only produced an alternative way to estimate flicker parameters, but also solutions to address deficiencies in the FPE method. In addition, new flicker correction methods that are either simpler, or are able to do more (handle color flicker), have also been found. Our work is extended to address color stain, which is yet another intensity related problem in motion pictures but somehow has remained untouched by the research community. All these will be covered in detail in subsequent chapters.

Let us begin to present our work by first looking into the FPE method originally proposed by Roosmalen.

6.2.1 Flicker parameters model

Flicker correction based on flicker parameters estimation uses a noisy frame model given by:

$$Y(x,y,t) = \alpha(x,y,t)I(x,y,t) + \beta(x,y,t) + \eta(x,y,t) \quad (6.1)$$

where x, y are discrete spatial coordinates and t indicates the frame number. $Y(x,y,t)$ and $I(x,y,t)$ indicate the observed and original image intensities, $\alpha(x,y,t)$ and $\beta(x,y,t)$ the flicker gain and flicker offset parameters (collectively known as "flicker parameters"), and $\eta(x,y,t)$ the flicker-independent noise, which includes collectively all non-flicker related noise like grain noise, additive Gaussian noise etc.

Assuming the noisy frames have gone through a pre-processing step first to remove most of the non-flicker noise, equation (6.1) can be simplified to:

$$Y(x,y,t) = \alpha(x,y,t)I(x,y,t) + \beta(x,y,t) \quad (6.2)$$

It is obvious from equation 6.2 that a flicker-free image is obtained when $\alpha(x,y,t)=1$ and $\beta(x,y,t)=0$.

If $\alpha(x,y,t)$ and $\beta(x,y,t)$ are spatially smooth, then these two parameters can be assumed to be locally constant, i.e.,

$$\begin{cases} \alpha(x,y,t) = \alpha_{m,n}(t) \\ \beta(x,y,t) = \beta_{m,n}(t) \end{cases} \quad \forall (x,y) \in \Omega_{m,n}$$

where $\Omega_{m,n}$ indicates a small region. Typically $\Omega_{m,n}$ is created by partitioning the image frame into equal-sized rectangular blocks where m,n indicate respectively the row and column locations of the image block. Equation 6.2 can now be re-written as:

$$Y(x,y,t) = \alpha_{m,n}(t)I(x,y,t) + \beta_{m,n}(t) \quad (6.3)$$

Equation 6.3 can be used to form the basis of flicker correction on which the FPE method was developed. Since $Y(x,y,t)$ is the observed intensity, the original intensity $I(x,y,t)$ can be obtained (implying the image can be corrected) if ways can be found to correctly estimate the flicker parameters $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$.

6.2.2 Flicker parameters estimation

Expressions for $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ can be derived by determining the expected value and variance of $Y(x,y,t)$ given in equation 6.3. This is illustrated below:

The expected value of $Y(x,y,t)$ in equation 6.3 for $x,y \in \Omega_{m,n}$ is given by:

$$\begin{aligned} E[Y(x,y,t)] &= E[\alpha_{m,n}(t)I(x,y,t) + \beta_{m,n}(t)] \\ &= \alpha_{m,n}(t)E[I(x,y,t)] + \beta_{m,n}(t) \end{aligned} \quad (6.4)$$

The variance of $Y(x,y,t)$ can be deduced as follows:

$$\begin{aligned}
 \sigma^2[Y(x,y,t)] &= E[Y^2(x,y,t)] - \{E[Y(x,y,t)]\}^2 \\
 &= E[\alpha_{m,n}^2(t)I^2(x,y,t) + 2\alpha_{m,n}(t)\beta_{m,n}(t)I(x,y,t) + \beta_{m,n}^2(t)] - \{\alpha_{m,n}(t)E[I(x,y,t)] + \beta_{m,n}(t)\}^2 \\
 &= \alpha_{m,n}^2(t)E[I^2(x,y,t)] + 2\alpha_{m,n}(t)\beta_{m,n}(t)E[I(x,y,t)] + \beta_{m,n}^2(t) \\
 &\quad - \alpha_{m,n}^2(t)\{E[I(x,y,t)]\}^2 - 2\alpha_{m,n}(t)\beta_{m,n}(t)E[I(x,y,t)] - \beta_{m,n}^2(t) \\
 &= \alpha_{m,n}^2(t)E[I^2(x,y,t)] - \alpha_{m,n}^2(t)\{E[I(x,y,t)]\}^2 \\
 &= \alpha_{m,n}^2(t)\sigma^2[I(x,y,t)]
 \end{aligned} \tag{6.5}$$

$\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ can now be readily obtained from equations 6.4 and 6.5:

$$\alpha_{m,n}(t) = \frac{\sigma[Y(x,y,t)]}{\sigma[I(x,y,t)]} \tag{6.6}$$

$$\beta_{m,n}(t) = E[Y(x,y,t)] - \alpha_{m,n}(t)E[I(x,y,t)] \tag{6.7}$$

Equations 6.6 and 6.7 cannot be used for flicker correction because they are both in terms of $I(x,y,t)$, the unknown intensity to be determined. However, if a stationary scene is assumed, a solution to this problem can be found by considering from a temporal sense, the expected value and variance of frame t :

$$\begin{aligned}
 E[I(x,y,t)]_t &= E_T\{E[I(x,y,t)]\} = E_T\left\{\frac{E[Y(x,y,t)] - \beta_{m,n}(t)}{\alpha_{m,n}(t)}\right\} \\
 &\approx \frac{1}{N-1} \sum_{p=t-N}^{t-1} \left\{\frac{E[Y(x,y,p)] - \beta_{m,n}(p)}{\alpha_{m,n}(p)}\right\}
 \end{aligned} \tag{6.8}$$

$$\begin{aligned}
 \sigma^2[I(x,y,t)]_t &= E_T\{\sigma^2[I(x,y,t)]\} = E_T\left\{\frac{\sigma^2[Y(x,y,t)]}{\alpha_{m,n}^2(t)}\right\} \\
 &\approx \frac{1}{N-1} \sum_{p=t-N}^{t-1} \left\{\frac{\sigma^2[Y(x,y,p)]}{\alpha_{m,n}^2(p)}\right\}
 \end{aligned} \tag{6.9}$$

where N in both equations 6.8 and 6.9 indicates the number of frames used in the averaging process.

Hence $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ can now be put in terms of parameters that can all be determined:

$$\alpha_{m,n}(t) = \frac{\sigma[Y(x, y, t)]}{\sqrt{\frac{1}{N-1} \sum_{p=t-N}^{t-1} \left\{ \frac{\sigma^2[Y(x, y, p)]}{\alpha_{m,n}^2(p)} \right\}}} \quad (6.10)$$

$$\beta_{m,n}(t) = E[Y(x, y, t)] - \frac{\sigma[Y(x, y, t)]}{\sqrt{\frac{1}{N-1} \sum_{p=t-N}^{t-1} \left\{ \frac{\sigma^2[Y(x, y, p)]}{\alpha_{m,n}^2(p)} \right\}}} \left\{ \frac{1}{N-1} \sum_{p=t-N}^{t-1} \frac{E[Y(x, y, p)] - \beta_{m,n}(p)}{\alpha_{m,n}(p)} \right\} \quad (6.11)$$

The expressions obtained above for $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ are rather complex and computationally intensive. The complexity can be substantially reduced by using the preceding frame (which had been corrected earlier) as a reference so that the following approximations are valid for $x, y \in \Omega_{m,n}$:

$$E[I(x, y, t)] \approx E[\hat{I}(x, y, t-1)] \quad (6.12)$$

$$\sigma[I(x, y, t)] \approx \sigma[\hat{I}(x, y, t-1)] \quad (6.13)$$

where $\hat{I}(x, y, t-1)$ indicates the intensity of the image at location (x, y) in the preceding corrected frame.

Hence from equations 6.6 and 6.7, $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ are now given by:

$$\alpha_{m,n}(t) \approx \frac{\sigma[Y(x, y, t)]}{\sigma[\hat{I}(x, y, t-1)]} \quad (6.14)$$

$$\beta_{m,n}(t) \approx E[Y(x, y, t)] - \frac{\sigma[Y(x, y, t)]E[\hat{I}(x, y, t-1)]}{\sigma[\hat{I}(x, y, t-1)]} \quad (6.15)$$

6.2.3 Motion block detection and flicker parameters estimation

Expressions for $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ derived above were based on the assumption of stationary scene, i.e., they are not valid for non-stationary (motion) blocks. Because of this, ways must be found to identify motion blocks in the frame being corrected, and subsequently flicker parameters of these motion blocks have to be found before any flicker correction can take place.

Since flicker parameters $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ are respectively the flicker gain and flicker offset of the image block $\Omega_{m,n}$, they form ideal candidates that can be used for motion block detection. It is a known fact that motion leads to significant change in intensity mean and variance in temporal statistics, and consequently also results in significant change in the values of flicker parameters, i.e., large deviation from 1.0 for $\alpha_{m,n}(t)$ and 0 for $\beta_{m,n}(t)$ [25]. Hence threshold values of the form $1 \pm T_\alpha$ and $\pm T_\beta$ can be used to check respectively on $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ for motion block detection. When the threshold values are surpassed, then the block is flagged as a motion block. Typical values for T_α and T_β are around 0.3 and 20 (suggested in [25]) and could be adjusted for optimal results.

Once all the motion blocks are identified, their flicker parameters can be obtained via iterative interpolation. The process is identical to that discussed in chapter 5 section 5.4.2.1 except that the parameters involved now are $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$.

6.2.4 Intensity Correction

With flicker parameters for all blocks (both stationary and motion) determined, the next step is to apply them for flicker correction. However, the flicker parameters $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ that have been obtained so far are block-based parameters. They have to be converted into pixel-based values through a pixel-by-pixel low-pass filtering process first before they can be applied for flicker correction, otherwise blocky artifacts will be introduced. Note that a computationally more efficient low-pass filtering algorithm has been developed in this thesis to cut down the computation time; the details are given in Appendix C.

Once flicker parameters $\alpha(x,y,t)$ and $\beta(x,y,t)$ for all pixels are determined, they can be used to correct the flicker frame based on equation 6.2. The equation is re-written in the

form of equation 6.16 below, in which $\hat{I}(x, y, t)$ indicates an estimate of the intensity of the flicker-free image at location (x, y) .

$$\hat{I}(x, y, t) = \frac{Y(x, y, t) - \beta(x, y, t)}{\alpha(x, y, t)} \quad (6.16)$$

6.2.5 Synthetic test sequence preparation

6.2.5.1 Global flicker generation

Since the FPE method is based on a flicker model given by equation 6.2, synthetic flicker can be easily generated based on this flicker model. This allows tests to be conducted in a controlled environment not achievable using natural test sequences. For example, based on the flicker model, control over various aspects of the flicker generated will now be feasible, such as the types (whether local or global flicker), the level of severity, the frequency (how often a flicker frame appears in the film sequence) etc. In addition, PSNR calculation can also be carried out; this is not possible using natural test sequences due to the absence of original good frames.

Generation of global flicker can be carried out by substituting a constant value pair of $\alpha(x, y, t)$ and $\beta(x, y, t)$ in equation 6.2 for each frame in the original good sequence. By adjusting the values of $\alpha(x, y, t)$ and $\beta(x, y, t)$ for each frame, frames that are homogeneously darker or brighter, i.e., frames with global flicker, can be generated.

Three types of flicker sequences, namely high-frequency, medium-frequency and low-frequency flicker sequences can be generated by controlling how often flicker frames appear in the frame sequence. This allows us to evaluate the performance of flicker algorithms with respect to the occurrence frequency of flicker frames.

For the work in this thesis, global flicker is added to a 20-frame sequence known as the West sequence. Shown in figures 6.1, 6.2 and 6.3 are respectively the intensity mean curves of the global flicker sequences created with high-frequency, medium-frequency and low-frequency flickers added.

Fig. 6.1 Intensity mean curves of the West sequence with high-frequency flicker added

Fig. 6.2 Intensity mean curves of the West sequence with medium-frequency flicker added

Fig. 6.3 Intensity mean curves of the West sequence with low-frequency flicker added

6.2.5.2 Local flicker generation

Generation of local flicker can be carried out using equation 6.2 where in each frame, the value of $\alpha(x,y,t)$ is generated from a normal distribution $N[1, \sigma_\alpha]$, and $\beta(x,y,t)$ from a normal distribution $N[0, \sigma_\beta]$ ($N[\mu, \sigma]$ represents a normal distribution with mean μ and standard deviation σ). By choosing different values of σ_α and σ_β , local flicker of different magnitude can be added to each frame.

For the work in this thesis, local flicker is added to a 10-frame sequence known as the Taxi sequence. The values for σ_α and σ_β have been chosen as 0.5 and 20 respectively. Figure 6.4 shows 3 consecutive frames from the Taxi sequence before and after local flicker is added. Significant amount of local flicker is clearly visible in the synthetic frames.

Addition of local flicker has not only altered the intensity distribution within a frame but also the average intensity of each frame in the sequence. Figure 6.5 shows the average intensity mean variations in the Taxi sequence before and after local flicker is added.

Fig. 6.4 Top row: original frames 5, 6 and 7 of the Taxi sequence

Bottom row: Synthetic frames 5, 6 and 7 with local flicker added

Fig. 6.5 Intensity mean curves of the Taxi sequence (original and with local flicker added)

6.2.6 Experimental results and analysis

It was mentioned in section 6.2.3 that threshold values of the form $1 \pm T_\alpha$ and $\pm T_\beta$ are used to check respectively on $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ for motion block detection. In our experimental work here for testing the FPE method, T_α and T_β have been chosen as 0.3 and 20 respectively.

6.2.6.1 Synthetic test sequence

6.2.6.1.1 Global flicker

The FPE method is first tested with the West sequence with high-frequency global flicker added (hence forth known as the high-frequency Taxi sequence) and the results are shown in figure 6.6. Note that after correction, the variance of the intensity mean curve of the Taxi sequence is reduced to only 0.19, which is much smaller than that ($= 27.82$) before correction, and is comparable to that ($= 0.71$) of the original sequence.

Similar results are obtained when the FPE method is tested with the medium and low frequency Taxi sequences; these are given in figures 6.7 and 6.8.

Fig. 6.6 Intensity mean curves of the West sequence with high-frequency flicker added

Fig. 6.7 Intensity mean curves of the West sequence with medium-frequency flicker added

Fig. 6.8 Intensity mean curves of the West sequence with low-frequency flicker added

Table 6.1: PSNR of the West sequence with high, medium and low frequency flicker added

Frame number	High-freq. flicker added PSNR value (dB)		Med.-freq. flicker added PSNR value (dB)		Low-freq. flicker added PSNR value (dB)	
	Before correction	After correction	Before correction	After correction	Before correction	After correction
0	∞	∞	∞	∞	∞	∞
1	28.13	40.23	∞	39.35	∞	39.35
2	∞	38.28	28.13	37.54	28.13	37.81
3	28.13	35.81	28.13	34.95	28.13	34.26
4	∞	36.97	∞	35.78	28.13	32.85
5	28.13	35.02	∞	36.24	28.13	32.27
6	∞	36.28	28.13	35.40	28.13	32.41
7	28.13	36.08	28.13	35.06	∞	34.23
8	∞	36.36	∞	35.74	28.13	33.34
9	28.13	34.76	∞	35.95	28.13	32.08
10	∞	35.71	28.13	33.41	28.13	30.85
11	28.13	35.51	28.13	34.47	28.13	31.16
12	∞	35.34	∞	35.03	28.13	31.16
13	28.13	35.33	∞	36.27	∞	34.40
14	∞	34.76	28.13	33.92	28.13	32.67
15	28.13	33.82	28.13	33.68	28.13	32.10
16	∞	33.95	∞	34.16	28.13	31.02
17	28.13	34.75	∞	37.14	28.13	32.09
18	∞	35.94	28.13	34.02	28.13	32.96
19	28.13	33.21	28.13	32.73	∞	34.62
Average PSNR (dB)	28.13	35.69	28.13	35.31	28.13	33.24

When the corrected sequence is evaluated in terms of PSNR values, it can be seen from Table 6.1 that the average PSNR values after correction range between 33.24 (for low-frequency flicker) to 35.69 (for high-frequency flicker); this represents an improvement of 5.11 dB to 7.56 dB after correction. Visual checking of the corrected frames shows effective removal of the global flicker, with no blocky effect or new artifacts introduced.

Note that averaging has been carried out by leaving out the perfect original frames which have PSNR equal to ∞ . This applies to average PSNR calculations in all other tables in this thesis.

6.2.6.1.2 Local flicker

When the FPE method is applied to the test sequence with local flicker added as described in section 6.2.5.2, the following results were obtained. With regard to intensity fluctuation, the corrected sequence has an intensity mean curve that is much smoother than that of the flicker sequence. It gives a variance of 0.39, which is comparable to that ($=0.13$) of the original good Taxi sequence, and much better than that ($=77.34$) of the flicker sequence. This is shown in figure 6.9.

On PSNR comparison, it can be seen from figure 6.10 that the PSNR curve for the corrected sequence is consistently higher than that of the flicker sequence, suggesting improvement over all frames. The corrected sequence has an average PSNR value of 36.23, compared to 28.57 before correction; this represents an improvement of 7.66 dB. Just like in the case for global flicker correction done earlier, visual checking of the corrected frames shows effective removal of local flicker, with no blocky effect or new artifacts introduced.

Fig. 6.9 Intensity mean curves of the Taxi sequence

Fig. 6.10 PSNR curves of the Taxi sequence

6.2.6.2 Natural test sequence

The FPE method was next tested with a number of natural flicker sequences, and the result for the Lady, Tunnel and Lantern sequences are respectively given in figures 6.11, 6.12 and 6.13. As expected, the intensity mean curves of all corrected sequences exhibit much reduced fluctuations. A quantitative measure of the intensity reduction is shown in Table 6.2, which gives a comparison of the variances of the intensity mean curves before and after correction.

Fig. 6.11 Intensity mean curves of the Lady sequence

Fig. 6.12 Intensity mean curves of the Tunnel sequence

Fig. 6.13 Intensity mean curves of the Lantern sequence

Table 6.2 Variance of intensity mean curves for natural flicker sequences

Flicker correction method	Variance of intensity mean curve		
	Lady sequence	Tunnel sequence	Lantern sequence
No Correction	3.11	3.79	0.60
FPE method	0.36 (11.6%)	0.021(0.6%)	0.017(2.8%)

6.2.6.3 Discussions

Test results with synthetic flicker sequences summarized in Table 6.3 below shows that flicker-correction performance of the FPE method is insensitive to whether the test sequence is with global or local flicker. Intensity mean curves of the corrected sequences all return very small variances -- under 2% relative to those before correction.

The same kind of insensitivity also prevails in PSNR improvement. As have been shown earlier, the FPE method produced PSNR improvement between 5.11 dB to 7.56 dB for global flicker sequences, and 7.66 dB for the local flicker sequence. Clearly, these improvements are of the same order of magnitude.

Table 6.3 Variance of intensity mean curves for global and local flicker sequences

Flicker correction method	Variance of intensity mean curve			
	Global flicker West sequence			Local flicker Taxi sequence
	High-freq.	Medium- freq.	Low-freq.	
No correction	27.82	25.59	18.40	77.34
FPE method	0.19 (0.68%)	0.35 (1.37%)	0.35 (1.9%)	0.39 (0.5%)

As shown in table 6.2, test results with natural test sequences show that only the Tunnel and Lantern sequences produced intensity mean curves that give percentage variances close to those of the synthetic test sequences. The percentage variance corresponding to Lady sequence is somewhat larger at 11.6%. Despite this, based on all the test results on both the synthetic and natural test sequences, it can still be concluded that the FPE method is an effective method for flicker correction.

It should be pointed out that the frame chosen as the reference frame for correction has great bearing on the correction results. In the work on correcting natural flicker sequences, the first frame of each sequence has been chosen as the reference frame for correction. In practice, it may not always be the case. A reference frame can be decided on a number of considerations; for example, the desired average frame intensity of the corrected sequence, the distribution of flicker and non-flicker frames in the original sequence etc. If it is desired that the corrected sequence is to have a higher average frame intensity, then a reference frame that is brighter should be chosen.

In the correction of the Lady sequence done earlier, frame 0 (average frame intensity = 79.8) has been used as the reference frame, and the corrected sequence has an average frame intensity of 80.9. If it is desired that the corrected sequence is to have a higher average frame intensity, then a brighter reference frame can be chosen instead. A possible candidate is frame 4, which has an average frame intensity of 81.3.

Figure 6.14 shows the intensity mean curves of the Lady sequence before and after correction using frame 4 as the reference. Correction is carried out in two directions: first going forward from frame 4 to frame 39, and then going backward from frame 4 to frame 0, each time using frame 4 as the reference. The corrected sequence gives an average frame intensity of 81.8, which is higher than that (=80.9) when frame 0 was used as the reference. Also it is noted that the variance of the intensity mean curve of the corrected sequence is now 0.14, which is substantially smaller than that (=0.36) when frame 0 is used as the reference. In fact with variance at 0.14, the percentage variance now stands at only 4.5% relative to that without correction. This highlights the dependence of the quality of correction on the reference frame chosen. Due to this reason, it is important that in the comparison of different correction methods using the same image sequence, the same reference frame must always be used.

Fig. 6.14 Intensity mean curves of the Lady sequence with different reference frames

6.3 Least square minimization (LSM) method

Having successfully implemented the FPE method, we next explore the possibility of developing new flicker-parameter-based algorithms for flicker correction. A new algorithm based on Least Square Minimization [42] was developed as a result of our research. This method took a different approach from the FPE method in deriving expressions for a pair of “flicker-related parameters” that can be used for flicker correction. The mathematical derivation for these two flicker-related parameters as well as how they can be applied for flicker correction will be presented. An analysis of the experimental results and comparisons with the FPE method will also be covered in this section.

6.3.1 Least square minimization

The Least Square Minimization method (hence-forth known as the “LSM” method) proposed here is based on the same flicker frame model given by equation 6.2 used in the FPE method. This equation is repeated below for easy reference:

$$Y(x,y,t) = \alpha(x,y,t)I(x,y,t) + \beta(x,y,t)$$

Just like in the FPE method, if only a small rectangular block $\Omega_{m,n}$ is considered, this equation can be simplified as:

$$Y(x,y,t) = \alpha_{m,n}(t)I(x,y,t) + \beta_{m,n}(t)$$

An estimate of the original intensity $\hat{I}(x,y,t)$ can thus be represented as:

$$\hat{I}(x,y,t) = \frac{[Y(x,y,t) - \beta_{m,n}(t)]}{\alpha_{m,n}(t)} = a_{m,n}(t)Y(x,y,t) + b_{m,n}(t) \quad (6.19)$$

where the gain $a_{m,n}(t)$ and off-set $b_{m,n}(t)$ are collectively known as the “flicker-related parameters”. Note that $a_{m,n}(t)$ and $b_{m,n}(t)$ are linked to the flicker parameters $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ used in the FPE method by $a_{m,n}(t) = 1/\alpha_{m,n}(t)$ and $b_{m,n}(t) = -\beta_{m,n}(t)/\alpha_{m,n}(t)$.

The estimated intensity $\hat{I}(x,y,t)$ differs from the original intensity $I(x,y,t)$ by an error $\varepsilon(x,y,t)$ equal to $I(x,y,t) - \hat{I}(x,y,t)$. It can then be argued that the original intensity $\hat{I}(x,y,t)$ can be approximated by minimizing a function $f(a,b)$, defined as the sum of squared error components in a small region $\Omega_{m,n}$ containing N pixels, where $f(a,b)$ is given by:

$$f(a,b) = \sum_{(x,y) \in \Omega_{m,n}} \varepsilon^2(x,y,t) = \sum_{(x,y) \in \Omega_{m,n}} \{I(x,y,t) - [a_{m,n}(t)Y(x,y,t) + b_{m,n}(t)]\}^2 \quad (6.20)$$

Values of the two flicker-related parameters $a_{m,n}(t)$ and $b_{m,n}(t)$ can be determined by minimizing $f(a,b)$, i.e., by setting both $\frac{\partial f(a,b)}{\partial a}$ and $\frac{\partial f(a,b)}{\partial b}$ to zero. These two values are referred to as the least square estimate and are denoted as $\hat{a}_{m,n}(t)$ and $\hat{b}_{m,n}(t)$. They can be shown to be given by [see Appendix A]:

$$\hat{a}_{m,n}(t) = \frac{1}{N\sigma^2[Y(x,y,t)]} \sum_{(x,y) \in \Omega_{m,n}} \{I(x,y,t) - E[I(x,y,t)]\} \{Y(x,y,t) - E[Y(x,y,t)]\} \quad (6.21)$$

$$\hat{b}_{m,n}(t) = E[I(x,y,t)] - \hat{a}_{m,n}(t)E[Y(x,y,t)] \quad (6.22)$$

$\hat{a}_{m,n}(t)$ and $\hat{b}_{m,n}(t)$ can be determined if all the terms in equations 6.21 and 6.22 are known. As it is only the pixel number N , the variance and also the expected value of $Y(x,y,t)$ are known quantities. The expected value of $I(x,y,t)$ is, unfortunately an unknown. However, if the previous frame (which has been flicker corrected earlier) is taken as a reference and only stationary regions are considered, then for $x,y \in \Omega_{m,n}$, the following approximation holds:

$$I(x,y,t) \approx I(x,y, t-1)$$

$$E[I(x,y,t)] \approx E[I(x,y,t-1)]$$

With this approximation, $\hat{a}_{m,n}(t)$ and $\hat{b}_{m,n}(t)$ can now be fully determined and they are now given by:

$$\hat{a}_{m,n}(t) \approx \frac{1}{N\sigma^2[Y(x,y,t)]} \sum_{(x,y) \in \Omega_{m,n}} \{I(x,y,t-1) - E[I(x,y,t-1)]\} \{Y(x,y,t) - E[Y(x,y,t)]\} \quad (6.23)$$

$$\begin{aligned} \hat{b}_{m,n}(t) &\approx E[I(x,y,t-1)] \\ &- \frac{E[Y(x,y,t)]}{N\sigma^2[Y(x,y,t)]} \sum_{(x,y) \in \Omega_{m,n}} \{I(x,y,t-1) - E[I(x,y,t-1)]\} \{Y(x,y,t) - E[Y(x,y,t)]\} \end{aligned} \quad (6.24)$$

6.3.2 Motion block detection

Expressions for $\hat{a}_{m,n}(t)$ and $\hat{b}_{m,n}(t)$ derived above are based on the assumption of stationary blocks. Just like in the FPE method, motion blocks in the frame being corrected have to be identified, and subsequently flicker-related parameters of these motion blocks have to be found before any flicker correction can take place.

In fact the way to detect motion blocks here is also very similar to that in the FPE method. As mentioned in section 6.2.3, motion leads to significant change in the values of flicker parameters $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$, the same also applies to the values of flicker-related parameters $a_{m,n}(t)$ and $b_{m,n}(t)$. Therefore, threshold values of the form $1 \pm T_a$ and $\pm T_b$ can

also be used to check respectively on $a_{m,n}(t)$ and $b_{m,n}(t)$ for motion block detection. When either $a_{m,n}(t)$ and $b_{m,n}(t)$ surpasses its threshold value, the image block will be labeled as a motion block.

6.3.3 Intensity Correction

With flicker-related parameters $a_{m,n}(t)$ and $b_{m,n}(t)$ for all blocks (both stationary and motion) determined, these block-based parameters are converted into pixel-based values through a pixel-by-pixel low-pass filtering process first before they are applied for flicker correction; otherwise blocky artifacts may result.

Once pixel-based flicker-related parameters $a(x,y,t)$ and $b(x,y,t)$ for all pixels are determined, they can then be used to correct the flicker frame based on equation 6.25 below. The equation is based on equation 6.19 with the restriction on $x,y \in \Omega_{m,n}$ removed.

$$\hat{I}(x, y, t) = a(x, y, t)Y(x, y, t) + b(x, y, t) \quad (6.25)$$

There are situations in which the estimated values of the flicker-related parameters are not so reliable, such as when the image contains large motion regions resulting in excessive rounds of interpolation being used in getting the flicker-related parameters. To address this issue, a bias factor k can be introduced into equation 6.25 in a manner described by the new equation 6.26:

$$\hat{I}(x, y, t) = k[a(x, y, t)Y(x, y, t) + b(x, y, t)] + (1 - k)Y(x, y, t) \quad (6.26)$$

Factor k is basically a measure of the importance of the current estimation. In situations where the reliability of the estimated flicker-related parameters is low, a small value of k should be chosen to reduce the contribution of the current estimation. Note that when doing this, the contribution from the original image content will be increased accordingly due to the $(1-k)$ factor.

6.3.4 Experimental results and analysis

Similar to the FPE method, in the LSM method, threshold values of the form $1 \pm T_a$ and $\pm T_b$ are used to check respectively on $a_{m,n}(t)$ and $b_{m,n}(t)$ for motion block detection. In our

experimental work here for testing the LSM method, T_a and T_b have been chosen as 3 and 60 respectively.

6.3.4.1 Synthetic test sequence

In testing the FPE method earlier using global flicker, sequence of high, medium and low flicker frequencies were used. However, such elaborate tests over different frequencies will not be employed in our test of other correction methods as such tests contribute little in gaining any insight in the evaluation of a particular correction method. In fact, it is sufficient just to use test sequence of one particular frequency to evaluate the effectiveness of flicker correction. Based on this common denominator, comparison on the relative effectiveness of flicker correction between different correction methods can be easily made. The low-frequency global flicker West sequence has been arbitrarily chosen here for all future tests involving global flicker sequences.

6.3.4.1.1 Global flicker

From figure 6.15, it can be seen that the intensity mean curve of the corrected sequence by the LSM method is as smooth as that obtained by the FPE method, and its variance value of 0.88 is also of the same order of magnitude. The only comment is that the intensity mean curve of the corrected sequence by the LSM method has a slightly higher intensity mean value, implying that the corrected sequence is slightly brighter. However, this difference is hardly noticeable when checked visually due to its small value.

Fig. 6.15 Intensity mean curves of the West sequence with low-frequency flicker added

PSNR comparison in Table 6.4 shows that there is little difference in PSNR value between the LSM-corrected frame and the corresponding FPE-corrected frame, varying between 0.03 to 1.65 dB in absolute terms. Overall the difference in average PSNR values between the two methods is only a small 0.03 dB. Also the difference in PSNR values between corresponding corrected frames is seen to alternate between '+' and '-' in an apparently random fashion, with about half being '+' and the other half being '-'.

Table 6.4: PSNR of the West sequence with low-frequency flicker added

Frame no.	PSNR value (dB)			
	Before correction	Corrected by FPE method	Corrected by LSM method	Difference between LSM method and FPE method
0	∞	∞	∞	0
1	∞	39.35	41.0	1.65
2	28.13	37.81	40.71	2.9
3	28.13	34.26	35.2	0.94
4	28.13	32.85	32.03	-0.82
5	28.13	32.27	31.78	-0.40
6	28.13	32.41	32.16	-0.25
7	∞	34.23	34.95	0.72
8	28.13	33.34	34.31	0.97
9	28.13	32.08	32.05	-0.03
10	28.13	30.85	31.4	0.55
11	28.13	31.16	32.25	1.09
12	28.13	31.16	31.21	0.05
13	∞	34.40	32.65	-1.75
14	28.13	32.67	31.74	-0.93
15	28.13	32.10	31.41	-0.69
16	28.13	31.02	30.43	0.59
17	28.13	32.09	31.27	0.82
18	28.13	32.96	31.72	-1.24
19	∞	34.62	32.7	-1.92
Average PSNR	28.13	33.24	33.21	-0.03

6.3.4.1.2 Local flicker

Similar to the test results for global flicker in section 6.3.4.1.1, there is little difference between the FPE method and LSM method when tested with the local flicker Taxi sequence. Figure 6.16 shows that the intensity mean curves of the FPE-corrected sequence and the LSM-corrected sequence almost overlap with each other; except that the LSM-corrected sequence has a slightly larger variance.

Results of PSNR comparison are very similar to those obtained earlier for global flicker. The two PSNR curves in figure 6.17 share a similar trend and are very close to each other, implying that there is very little difference between the two methods. The difference in average PSNR value is a small 0.02 dB, which is very close to the 0.03 dB obtained earlier for global flicker. The PSNR curves also reveal that about half of the LSM-corrected frames have a higher PSNR value than the FPE-corrected frames, and the distribution is seen to be in a random manner; this is also consistent with our results obtained earlier for global flicker.

Fig. 6.16 Intensity mean curves of the Taxi sequence

Fig. 6.17 PSNR curves of the Taxi sequence

Fig. 6.18 Top row: original frames 5, 6 and 7 of the Taxi sequence
 Middle row: Synthetic frames 5, 6 and 7 with local flicker added
 Bottom row: frames 5, 6 and 7 corrected by LSM method

Figure 6.18 shows three arbitrarily chosen consecutive frames from the Taxi sequence. Starting from the top row to the bottom row, they are respectively the original good frames, frames added with local flicker and frames corrected by the LSM method. It can be seen that the corrected sequence is uniform in intensity, with all the visible uneven brightness fluctuations in the flicker sequence clearly removed. Subjective evaluation was also carried out by viewing the Taxi sequence in real-time playback. It was found that the visual quality of the corrected image sequence was substantially better than that of the flicker sequence.

6.3.4.2 Natural test sequence

Test results with the same three natural flicker sequences (Lady, Lantern and Tunnel) are given below in figures 6.19 to 6.21. A comparison of the variances of the intensity mean curves between the LSM-corrected sequence and the FPE-corrected sequence is given in Table 6.5.

Fig. 6.19 Intensity mean curves of the Lady sequence

Judging by the variance comparison in Table 6.5, it can be said that except for the Lantern sequence where the performance of the LSM method is on par with the FPE method, the LSM method does not perform as well as the FPE method in both the Lady and Tunnel

sequences. For these two sequences, the intensity mean curves after correction by the LSM method are not as smooth compared to those by the FPE method, as indicated by the slightly larger variance values in the corrected curves. However, considering that both the FPE-corrected sequence and the LSM-corrected sequence have intensity mean curves that are much flatter than that before correction, the slight difference in the variance value between the two is rather unlikely to cause much difference in terms of the viewing experience of the corrected sequences. This was readily verified when the corrected sequences were visually checked, and so on this note it is acceptable to conclude that the LSM method works to a large extent as well as the FPE method.

The fact that there are no reliable ways to select threshold values in the LSM method for motion block detection is very similar to the same problem encountered in the FPE method. This problem will be looked into again later in section 6.4, where a solution is proposed to solve this problem.

Fig. 6.20 Intensity mean curves of the Tunnel sequence

Fig. 6.21 Intensity mean curves of Lantern sequence

Table 6.5 Variance of intensity mean curves

Flicker correction method	Variance of intensity mean curve		
	Lady sequence	Tunnel sequence	Lantern sequence
No Correction	3.11	3.79	0.60
FPE method	0.36 (11.6%)	0.021 (0.6%)	0.017(2.8%)
LSM method	0.74 (23.8%)	0.039 (1.0%)	0.017 (2.8%)

6.4 FPE method with stationary block re-examination incorporated

Earlier work in section 6.2 showed that the quality of flicker correction in the FPE method depends very much on the accuracy of identifying stationary and motion blocks. In the FPE method, threshold values of the form $1 \pm T_\alpha$ and $\pm T_\beta$ are used to check respectively on $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ for motion blocks. Unfortunately there are no reliable ways to select these threshold values properly, and hence the accuracy of motion block detection cannot be guaranteed. As a result the quality of flicker correction is adversely

affected. To address this apparent weak link in the FPE method, a re-examination process on stationary blocks is proposed here, details of which are illustrated below.

6.4.1 Stationary block re-examination

A two-tier checking process [35] is proposed here to overcome such a deficiency in the FPE method. As before, motion blocks are first detected using some pre-selected threshold values in the form of $1 \pm T_\alpha$ and $\pm T_\beta$. Stationary blocks that have been identified can then be corrected based on equation 6.16. However some of these stationary blocks identified at this stage could well be motion blocks as it is known that their detection was not very reliable. These preliminary stationary blocks will now be subjected to a re-examination process to double-check its validity. This is done by first working out the mean square error (MSE) between the current corrected stationary block and the corresponding block at the same location in the previous corrected frame using equation 6.17 given below:

$$MSE = \frac{1}{N_1 \times N_2} \sum_{(x,y) \in \Omega_{m,n}} [\hat{I}(x,y,t) - \hat{I}(x,y,t-1)]^2 \quad (6.17)$$

where $N_1 \times N_2$ represents the size of the small image block $\Omega_{m,n}$ at the m^{th} row and n^{th} column; t and $t-1$ respectively indicate the current and previous frames.

If the MSE exceeds certain threshold value, then this block that has been identified as “stationary” earlier will now be re-labeled as a motion block. This re-examination process will be carried out on all blocks that were previously identified as stationary and at the end of this process motion blocks that were wrongly identified as stationary blocks will be identified. These newly discovered motion blocks can be corrected by having their flicker parameter values obtained via iterative interpolation, followed by flicker correction as described in section 6.2.4.

6.4.2 Experimental results and analysis

In order to be consistent, the same set of threshold values $T_\alpha (=0.3)$ and $T_\beta (=20)$ used in section 6.2.5 for the FPE method are used here to check respectively on $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ for motion block detection.

6.4.2.1 Synthetic test sequence

6.4.2.1.1 Global flicker

Figure 6.22 shows that when re-examination is incorporated into the FPE method, the intensity mean curve of the corrected sequence has a variance of 0.22, which is 37% smaller than that without re-examination. This indicates that re-examination has significantly improved the correction results.

Fig. 6.22 Intensity mean curves of the West sequence with low-frequency flicker added

PSNR comparison in Table 6.6 shows with stationary block re-examination incorporated into the FPE method, the corrected frames consistently return a higher PSNR value than those obtained without re-examination. The largest improvement occurs in frame 12, which is 2.84 dB. Overall the average improvement is 1.63 dB. This improvement in dB value may appear small, however when translated into MSE (mean square error), it is in fact quite substantial. Using equation 2.1 in chapter 2, this PSNR difference can be

translated into a ratio in MSE, where $MSE = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N [i_j(x, y) - i_k(x, y)]^2$ and $i_j(x, y)$

and $i_k(x, y)$ are respectively the pixel intensity at location (x, y) of the good frame and corrected frame. It can be easily shown that $MSE_{(FPE)} / MSE_{(FPE \text{ with re-examination incorporated})} = 42.65$, which shows that the incorporation of re-examination into the FPE method has quite substantially improved the correction results.

Table 6.6: PSNR of the West sequence with low-frequency flicker added

Frame no.	PSNR value (dB)			
	Before correction	Corrected by FPE method	Corrected by FPE method incorporated with re-examination	Difference between the two methods
0	∞	∞	∞	0
1	∞	39.35	39.59	0.24
2	28.13	37.81	39.95	2.14
3	28.13	34.26	36.13	1.87
4	28.13	32.85	34.68	1.83
5	28.13	32.27	34.16	1.89
6	28.13	32.41	34.07	1.93
7	∞	34.23	35.15	0.92
8	28.13	33.34	34.37	1.03
9	28.13	32.08	33.92	1.84
10	28.13	30.85	33.06	2.21
11	28.13	31.16	33.87	2.71
12	28.13	31.16	34.00	2.84
13	∞	34.40	35.88	1.48
14	28.13	32.67	34.03	1.36
15	28.13	32.10	33.27	1.17
16	28.13	31.02	32.35	1.33
17	28.13	32.09	34.03	1.94
18	28.13	32.96	34.19	1.23
19	∞	34.62	35.57	0.95
Average PSNR (dB)	28.13	33.24	34.87	1.63

6.4.2.1.2 Local flicker

Figure 6.22 shows that when re-examination is incorporated into the FPE method, the intensity mean curve of the corrected sequence gives a variance of 0.22, which is 44%

smaller than that without re-examination. This result is consistent with that obtained under global test sequence earlier in section 6.4.2.1.1.

Fig. 6.23 Intensity mean curves of Taxi sequence

Similar to the results obtained for global flicker sequence, with re-examination incorporated, the corrected frames consistently return a higher PSNR value. This is evident from figure 6.24, which shows that the PSNR curve for the FPE method incorporated with re-examination is always higher than that without re-examination. The former has an average PSNR of 36.61 dB, compared to 36.23 dB of the latter; this amounts to an average improvement of 0.38 dB.

Fig. 6.24 PSNR curves for Taxi sequence

Test results showed that PSNR improvement has been observed in both global and local flicker test sequences. Such improvement can be explained by the two-tier motion block examination process when re-examination is incorporated into the FPE method. Figure 6.25(a) and figure 6.25(b) respectively show the stationary blocks (black regions) and motion blocks (white regions) detected by the FPE method, and the FPE method with re-examination incorporated, in frame 12 of the low-frequency flicker West sequence. It can be seen that a significant number of black blocks in figure 6.25(a) are re-colored as white in figure 6.25(b). This is due to the fact that these stationary blocks identified in the FPE method are actually motion blocks after the re-examination process. Improvement in the accuracy of motion blocks detection thus leads to a higher PSNR value.

Fig. 6.25 Stationary blocks (black regions) and motion blocks (white regions) in frame 12 of the low-frequency flicker West sequence by (a) FPE method (b) FPE method with stationary block re-examination incorporated

6.4.2.2 Natural test sequence

Test results with real natural flicker sequences are given in figures 6.26 to 6.28. Intensity mean curves of the corrected sequences look very similar to those corrected by the FPE method. Marginal improvement over the FPE method can be seen in the data presented in Table 6.7, which shows a marginally reduced variance value for all the intensity mean curves of the corrected sequences when re-examination is incorporated. Quantitatively speaking the variance figures indicate that flicker correction has been improved with the incorporation of re-examination, however, the slightly improved figures do not produce much, if any, visual improvement when viewing the corrected sequences. This is quite

expected because human eyes in general are not sensitive enough to detect such minute reduction in intensity fluctuation.

Fig. 6.26 Intensity mean curves of the Lady sequence

Fig. 6.27 Intensity mean curves of the Tunnel sequence

Fig. 6.28 Intensity mean curves of the Lantern sequence

Table 6.7 Variance of intensity mean curves

Flicker correction method	Variance of intensity mean curve		
	Lady sequence	Tunnel sequence	Lantern sequence
No Correction	3.11	3.79	0.60
FPE method	0.36 (11.6%)	0.021 (0.6%)	0.017(2.8%)
FPE method with stationary block re-examination incorporated	0.33 (10.6%)	0.019 (0.5%)	0.016 (2.6%)

6.4.2.3 Discussions

How much improvement in flicker correction can be achieved is closely related to how many more motion blocks can be identified when re-examination is incorporated into the FPE method. If in the FPE method the choice of threshold values T_α and T_β has already quite accurately identified the motion blocks, then the incorporation of re-examination may not find more motion blocks, and thus will not help much to improve flicker correction. This will be illustrated by the examples given in the following page.

Shown in figure 6.29 (a) and (b) are respectively the motion blocks detected in frame 37 of the Lady sequence by the FPE method and FPE method incorporated with stationary block re-examination. It can be seen that quite a few extra motion blocks (in white) are detected when re-examination is incorporated, and as a result better flicker correction is achieved.

Fig. 6.29 Motion blocks (shown as white) detected in frame 37 of the Lady sequence
(a) FPE method (b) FPE method incorporated with stationary block re-examination

When tested with the Lantern sequence, the incorporation of re-examination was found to have detected very few extra motion blocks. This is shown in figure 6.30. For this reason, the improvement in flicker correction was only marginal. This is consistent with the results in Table 6.8, which shows the variance of the intensity mean curve is reduced marginally from 0.017 to 0.016 after the incorporation of re-examination.

Fig. 6.30 Motion blocks (shown as white) detected in frame 2 of the Lantern sequence
(a) FPE method (b) FPE method incorporated with stationary block re-examination

6.5 Conclusion

In this chapter, three flicker correction methods based on flicker parameter estimation were studied in detail and fully tested with both synthetic and natural test sequences. A comparison of their performances with regard to natural test sequences is given in Table 6.8 below:

Table 6.8 Variance of intensity mean curves

Flicker correction method	Variance of intensity mean curve		
	Lady sequence	Tunnel sequence	Lantern sequence
No Correction	3.11	3.79	0.60
FPE method	0.36 (11.6%)	0.021 (0.6%)	0.017(2.8%)
LSM method	0.74 (23.8%)	0.039 (1.0%)	0.017 (2.8%)
FPE method with stationary block re-examination incorporated	0.33 (10.6%)	0.019 (0.5%)	0.016 (2.6%)

All three methods are found to be effective in flicker correction. The LSM method presents an entirely new algorithm for flicker correction and managed to achieve comparable results to those of the FPE method. Stationary block re-examination is shown to be a viable approach to overcome the short-coming in the FPE method, i.e., its inability to detect motion blocks accurately. More research will be carried out in coming chapters to explore if better ways can be found to address the flicker correction problem.

Chapter 7

Motion-compensated FPE Algorithm

7.1 Introduction

In chapter 6, a stationary block re-examination method was presented to overcome the shortcoming of the FPE method, i.e., its inability to detect motion blocks accurately. However, the method was shown to be able to improve the correction results only marginally. This chapter presents a new method based on motion compensation to address this deficiency in the FPE method. The new method is found to be very effective and delivers much better result than that achieved by incorporating re-examination into the FPE method.

7.2 Applying motion compensation to enhance flicker correction

7.2.1 The FPE method and its inherent limitations

In chapter 6, the FPE method postulated by Roosmalen was presented in detail and the drawback associated with this method was discussed at length. The essentials of the FPE method will be briefly revisited below to set the scene for bringing in on stage a new method that is able to provide a much improved solution to the known problem in the FPE method.

Given below, in a very brief manner, are what have been established earlier in chapter 6 and they are repeated here for readers' easy reference:

In the FPE method, a noisy frame with all non-flicker noise removed is modeled by:

$$Y(x,y,t) = \alpha(x,y,t)I(x,y,t) + \beta(x,y,t) \quad (6.2)$$

where $\alpha(x,y,t)$ and $\beta(x,y,t)$ are known as the flicker parameters.

If $\alpha(x,y,t)$ and $\beta(x,y,t)$ are spatially smooth, then they can be considered as constant over a small region $\Omega_{m,n}$, i.e.,

$$\begin{cases} \alpha(x,y,t) = \alpha_{m,n}(t) \\ \beta(x,y,t) = \beta_{m,n}(t) \end{cases} \quad \forall (x,y) \in \Omega_{m,n}$$

The flicker parameters for this small region are shown to be respectively equal to:

$$\alpha_{m,n}(t) = \frac{\sigma[Y(x,y,t)]}{\sigma[I(x,y,t)]} \quad (6.6)$$

$$\beta_{m,n}(t) = E[Y(x,y,t)] - \alpha_{m,n}(t)E[I(x,y,t)] \quad (6.7)$$

where E and σ respectively denote the expected value and standard deviation.

To a good approximation, the following are true for stationary regions:

$$E[I(x,y,t)] \approx E[\hat{I}(x,y,t-1)] \quad (6.12)$$

$$\sigma[I(x,y,t)] \approx \sigma[\hat{I}(x,y,t-1)] \quad (6.13)$$

Hence flicker parameters $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ are now given by:

$$\alpha_{m,n}(t) \approx \frac{\sigma[Y(x,y,t)]}{\sigma[\hat{I}(x,y,t-1)]} \quad (6.14)$$

$$\beta_{m,n}(t) \approx E[Y(x,y,t)] - \frac{\sigma[Y(x,y,t)]E[\hat{I}(x,y,t-1)]}{\sigma[\hat{I}(x,y,t-1)]} \quad (6.15)$$

Motion blocks are detected if the two flicker parameters $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ exceed certain threshold values. Flicker parameters for motion blocks are later obtained via an iterative interpolation process. With flicker parameters for both stationary and motion blocks

determined, they are converted into pixel-based values through a low-pass filtering process. These pixel-based values can then be applied for flicker correction according to the equation below:

$$\hat{I}(x, y, t) = \frac{Y(x, y, t) - \beta(x, y, t)}{\alpha(x, y, t)} \quad (6.16)$$

The main problem of the FPE method lies in the difficulty in selecting the appropriate threshold values for $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ for motion block detection. As a result, motion blocks detected are not very reliable and hence will adversely affect the quality of flicker correction.

7.2.2 Motion vector compensation

A novel motion vector compensation method is proposed here to overcome such short-coming in the FPE method.

In our proposed method, motion vector compensation is to be introduced to both equations 6.12 and 6.13 to account for motion. Assuming that location (x, y) in the current frame under correction was originally located at location (x', y') in the preceding frame, in going from the preceding frame to the current frame, location (x, y) has actually undergone some movement in both the horizontal and vertical directions. If V_H and V_V are respectively the distance moved in the horizontal and vertical directions, then $V_H = x - x'$ and $V_V = y - y'$, and V_H and V_V are in general known as the “motion vectors”. Equations 6.12 and 6.13 can now be re-written as:

$$E[I(x, y, t)] \approx E[\hat{I}(x', y', t-1)] = E[\hat{I}(x - V_H, y - V_V, t-1)] \quad (7.1)$$

$$\sigma[I(x, y, t)] \approx \sigma[\hat{I}(x', y', t-1)] = \sigma[\hat{I}(x - V_H, y - V_V, t-1)] \quad (7.2)$$

in which V_H and V_V are respectively the horizontal and vertical motion vectors for all locations within $\Omega_{m,n}$. As a result, flicker parameters $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ in equations 6.14 and 6.15 can now be re-written as:

$$\alpha_{m,n}(t) \approx \frac{\sigma[Y(x, y, t)]}{\sigma[\hat{I}(x - V_H, y - V_V, t-1)]} \quad (7.3)$$

$$\beta_{m,n}(t) = E[Y(x, y, t)] - \frac{\sigma[Y(x, y, t)]E\{\hat{I}[x - V_H, y - V_V, t - 1]\}}{\sigma[\hat{I}(x - V_H, y - V_V, t - 1)]} \quad (7.4)$$

It can be seen that once the pair of motion vectors V_H and V_V are determined, flicker parameters $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ for the small region $\Omega_{m,n}$ will be readily obtained through equations 7.3 and 7.4. The remaining steps follow exactly those of the FPE method, i.e., $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ are next converted into pixel-based values through a low-pass filtering process, followed by the correction step using equation 6.16.

7.2.2.1 Determining motion vectors

A standard block-matching method using the minimum Mean Absolute Difference criterion was employed to determine motion vectors. The process, as described by the flow chart in figure 7.1 below, is almost identical to the motion detection process described in chapter 5 section 5.4.1.2.

The slight difference between the two lies in the action taken after the nature (i.e. being stationary, motion or uncovered) of the block is determined. An additional step is taken here which is to record down the motion vector pair V_H and V_V .

The determination of motion vector pairs depends very much on the nature of the block being dealt with; this is illustrated in the flow-chart in figure 7.1. It is obvious that for a stationary block, the motion vectors in both the horizontal and vertical directions should be equal to zero. For a motion block, the motion vectors are the pair of V_H and V_V obtained for MAD_{min} . For uncovered blocks, since there is no match in the previous frame, the pair of motion vectors obtained will be unreliable and so should be rejected. As such equations 7.3 and 7.4 will not be applicable, and the flicker parameters $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ can instead be obtained using an iterative interpolation approach as described in chapter 5 section 5.4.2.1. That is, $\alpha_{m,n}(t)$ for all uncovered blocks are to be interpolated from $\alpha_{m,n}(t)$ of the stationary and motion blocks computed from equation 7.3, whereas the $\beta_{m,n}(t)$ for all uncovered blocks are to be interpolated from $\beta_{m,n}(t)$ of the stationary and motion blocks computed from equation 7.4.

Fig. 7.1 Flow-chart for determining motion vectors

The success of our proposed motion vector compensation method depends very much on whether reliable motion vectors can still be obtained in a flicker environment in which the assumption of constant luminance is no longer valid. In chapter 5 section 5.4.1.2, it has already been shown that our motion detection algorithm was able to work in the presence of flicker. The motion vector determination algorithm, of which a large part is based on the motion detection algorithm developed in chapter 5, is expected to work fine as well. A special test case for testing the effectiveness of the motion vector determination algorithm was created and it turned out that our motion vector determination algorithm was extremely competent in a flicker environment. Details of the test are as given below.

7.2.2.2 Motion vector determination in a flicker environment

A centrally-positioned 256x256 pixels cosine ball described by equation 7.5 was added with artificial flicker to the lower right hand quadrant as shown in figure 7.2(a). It was then made to move 10 pixels towards both right and downward as shown in figure 7.2(b).

$$I(r) = 100 \exp(-0.01r) \cos \left[\frac{2\pi r}{(1 - \frac{r}{80}) \times 10 + 10} \right] + 128 \quad (7.5)$$

Fig. 7.2(a) Cosine ball with a flickered quadrant

Fig. 7.2(b) The same cosine ball moved 10 pixels both right and downward

Figures 7.3(a) and 7.3(b) show respectively the motion vectors detected using the normal block-matching method and the modified block-matching method. It can be clearly seen that motion vectors detected by the former method in the flicker quadrant do not tally with the movement of the cosine ball, whereas those detected by the latter method shows almost perfect results.

Fig. 7.3(a) Motion vectors detected using the normal block-matching method

Fig. 7.3(b) Motion vectors detected using the modified block-matching method

7.2.3 Experimental results and analysis

7.2.3.1 Synthetic test sequence

7.2.3.1.1 Global flicker

Figure 7.4 shows that when motion vector compensation is incorporated into the FPE method, the intensity mean curve of the corrected sequence has a variance of 0.17, which is 51% smaller than that without motion vector compensation. This indicates that motion vector compensation has significantly improved the correction results. This result is also better than that obtained earlier in Chapter 6 section 6.4.2.1.1 using FPE method incorporated with stationary block re-examination. The reduction then was at 37%.

Fig. 7.4 Intensity mean curves of the West sequence with low-frequency flicker added

PSNR comparison in Table 7.1 shows with motion vector compensation incorporated into the FPE method, the corrected frames consistently return a higher PSNR value than those obtained without compensation. Overall the average improvement is 2.19 dB.

Table 7.1: PSNR for the West sequence with low-frequency flicker added

Frame no.	PSNR value (dB)			
	Before correction	Corrected by FPE method	Corrected by motion-compensated FPE method	Difference between the two methods
0	∞	∞	∞	0
1	∞	39.35	39.78	0.43
2	28.13	37.81	40.23	2.42
3	28.13	34.26	37.19	2.93
4	28.13	32.85	34.98	2.13
5	28.13	32.27	34.86	2.59
6	28.13	32.41	34.77	2.36
7	∞	34.23	35.35	1.12
8	28.13	33.34	34.88	1.54
9	28.13	32.08	34.26	2.18
10	28.13	30.85	33.87	3.02
11	28.13	31.16	34.57	3.41
12	28.13	31.16	34.69	3.53
13	∞	34.40	36.48	2.08
14	28.13	32.67	34.83	2.16
15	28.13	32.10	33.87	1.77
16	28.13	31.02	32.95	1.93
17	28.13	32.09	34.93	2.84
18	28.13	32.96	34.79	1.83
19	∞	34.62	35.85	1.23
Average PSNR (dB)	28.13	33.24	35.43	2.19

7.2.3.1.2 Local flicker

Figure 7.5 shows that when motion vector compensation is incorporated into the FPE method, the intensity mean curve of the corrected sequence gives a variance of 0.11, which is only 0.14% of that of the sequence with local flicker added. This result is also better than those achieved under the pure FPE method and the FPE method incorporated with re-examination (see chapter 6 figure 6.23), whose corresponding variance values are at 0.39 and 0.22 respectively.

Fig 7.5 Intensity mean curves of the Taxi sequence

Fig. 7.6 PSNR curves for the Taxi sequence

Figure 7.6 shows that the corrected frames by the proposed method consistently return a higher PSNR value when compared to the frames with local flicker added. The former has an average PSNR of 37.13 dB, compared to 28.57 dB of the latter; this amounts to an average improvement of 8.56 dB. In fact the proposed method returns the highest PSNR value when compared with the values obtained using other correction methods.

7.2.3.2 Natural test sequence

As before, the proposed method was tested with a number of naturally degraded flicker sequences, and the test results of three of them, namely the Lady, Tunnel and Lantern sequences are presented here.

Fig. 7.7 Intensity mean curves of the Lady sequence

Let us begin with the 40-frame Lady sequence. Figure 7.7 shows the intensity mean curve of the corrected frame sequence and that of the original natural flicker frame sequence. As can be seen, the corrected sequence has an extremely flat and smooth intensity mean curve, with a variance value of only 0.02. This amounts to only 0.64% of that of the original flicker sequence. Shown in Table 7.2 is a comparison between the proposed method and other FPE-based methods. At 0.02, the variance value corresponding to the proposed method is substantially smaller than that of the FPE method, which stands at 0.36. This amounts to approximately a 94% reduction. When compared to the FPE with stationary block re-examination method, the percentage reduction is also about 94%. Hence it can be said that the improvement over the earlier two methods is indeed very substantial.

Test results with the Tunnel sequence are equally impressive. As shown in figure 7.8, the corrected sequence has an intensity mean curve that is extremely flat and smooth, with a variance value of only 0.003. This amounts to only 0.08% of that of the original flicker sequence. Comparison given in Table 7.2 again shows the proposed method out performs the other two FPE-based methods by a large margin – a reduction of about 84% in variance value is noted in both cases.

Fig. 7.8 Intensity mean curves of the Tunnel sequence

For the Lantern sequence, the proposed method performs marginally better than the FPE method. The intensity mean curve of the corrected sequence in figure 7.9 gives a variance of 0.016, which is a fraction better than that (0.017) of the FPE method. Table 7.2 shows that the corresponding variance value for the FPE with stationary block re-examination method is 0.016, the same as that by the proposed method.

Fig. 7.9 Intensity mean curves of the Lantern sequence

Table 7.2 Variance of intensity mean curves

Flicker correction method	Variance of intensity mean curve		
	Lady sequence	Tunnel sequence	Lantern sequence
No Correction	3.11	3.79	0.60
FPE method	0.36 (11.6%)	0.021 (0.6%)	0.017(2.8%)
FPE with stationary block re-examination method	0.33 (10.6%)	0.019 (0.5%)	0.016 (2.6%)
Motion-compensated FPE method	0.02 (0.64%)	0.003 (0.08%)	0.016 (2.6%)

Visual inspection shows that flicker is significantly (if not completely) removed from the corrected Lady, Tunnel and Lantern sequences, while at the same time without introducing any visible artifacts.

7.3 Conclusion

Based on the results in Table 7.2, it can be said that the proposed method consistently outperforms the stationary block re-examination method, except for the Lantern sequence where the performance is the same. Although both the stationary block re-examination method and the proposed method set out to address the inherent limitation problem in the FPE method, i.e., its inability to detect motion blocks accurately, the approaches taken by them are quite different. The stationary block re-examination method approaches the problem by applying a two-tier checking process to identify motion blocks in a frame, and subsequently determine the flicker parameters of these motion blocks via iterative interpolation before using them for flicker correction. In the proposed method, motion blocks are not only identified but their motion vectors are also determined, and their flicker parameters are subsequently computed through equations 7.3 and 7.4.

Because flicker parameters of motion blocks are obtained through computation rather than interpolation, which is by nature a very rough estimation process, the proposed method is able to more accurately determine the flicker parameters. Hence when these more accurately determined flicker parameters are used to correct flicker, it is not surprising that better flicker correction results than those of the stationary block re-examination method are obtained.

Chapter 8

Simplified Motion-compensated FPE Algorithm

8.1 Introduction

All the flicker correction methods developed thus far in this thesis have in one way or another produced improvement over earlier methods; however solutions to long existing problems such as scene change, occlusion, motion smear and non-translational motion etc., still remain very much elusive. In this chapter, a new mathematical model is proposed to describe the degraded motion pictures, and a corresponding flicker correction method based on this model will be developed. Test results show that this new method is both competent and effective in flicker removal. In addition, it has the added advantage of being more computational efficient and most important of all, it is able to provide a solution to all the above-mentioned problems that none of the existing methods can deliver.

8.2 Inherent problems of the flicker model used in the FPE method

All the flicker correction methods described in chapter 6 are FPE-based methods. They are based on a noisy frame model given by equation 6.1 (repeated here for easy reference):

$$Y(x,y,t) = \alpha(x,y,t)I(x,y,t) + \beta(x,y,t) + \eta(x,y,t) \quad (6.1)$$

where $\alpha(x,y,t)$ is the multiplicative flicker gain and $\beta(x,y,t)$ the offset; collectively the pair is known as the “flicker parameters”. Applying a pre-processing step to remove all the non-flicker noise first will simplify equation 6.1 to equation 6.2:

$$Y(x,y,t) = \alpha(x,y,t)I(x,y,t) + \beta(x,y,t) \quad (6.2)$$

Such a noisy frame model has an inherent problem in that it is not easy to estimate both $\alpha(x,y,t)$ and $\beta(x,y,t)$ accurately. In all the methods developed in chapter 6, the expressions for $\alpha(x,y,t)$ and $\beta(x,y,t)$ are inter-dependent; typically $\alpha(x,y,t)$ is first determined followed by $\beta(x,y,t)$. Hence any error in determining $\alpha(x,y,t)$ will lead to error in the determination of $\beta(x,y,t)$. There are situations whereby the determination of $\alpha(x,y,t)$ and $\beta(x,y,t)$ is extremely unreliable; a well-known case is one that involves a large area of uniform intensity. This is because for any original image intensity in an uniform region, there exist an infinite number of combinations of $\alpha(x,y,t)$ and $\beta(x,y,t)$ that lead to the observed intensity [21,42]. Blocky artifacts [42] may be formed as a result of erroneous determination of $\alpha(x,y,t)$ and $\beta(x,y,t)$, and the problem is aggravated because these blocky artifacts are particularly visible in uniform intensity regions in real film sequences, such as the wall in a room. On the other hand, methods developed in chapter 6 all use the preceding corrected frame as a reference for correcting the current frame, and later use the corrected current frame as the reference to correct the next frame. It is obvious that any error in the determination of $\alpha(x,y,t)$ and $\beta(x,y,t)$ will have an accumulative effect as more frames are corrected, resulting in a continuous deterioration of correction quality.

Even if these problems are put aside, there is still a long list of old problems for which currently developed flicker correction methods offer no answer. These well-known problems are related to scene change, occlusion, motion smear (blurring) and non-translational motion (e.g., deformable objects, zooming). It is felt that some fresh thinking is probably needed to break new grounds in our research. It is against this backdrop that the idea of a new single flicker parameter model started to take shape.

8.3 A simplified flicker model

In our newly proposed simplified flicker model, the additive flicker gain $\beta(x,y,t)$ is discarded from equation 6.2, thus giving us a much simplified flicker model shown below:

$$Y(x,y,t) = \alpha(x,y,t)I(x,y,t) \quad (8.1)$$

This means that the observed intensity $Y(x,y,t)$ differs from its original intensity $I(x,y,t)$ by just a factor $\alpha(x,y,t)$, which is the flicker gain. If $\alpha(x,y,t)$ can be accurately estimated, then the original intensity $I(x,y,t)$ can be easily restored using equation 8.1.

As mentioned in section 8.3, it is in general difficult to accurately determine the offset parameter $\beta(x,y,t)$, especially in regions with uniform intensity. Eliminating $\beta(x,y,t)$ altogether will not only solve this problem, but it will also simplify computations of the whole flicker correction process. In fact, as it turns out, this simplified model can do even more – many long-existing flicker correction problems can be easily dealt with using a method developed based on this model, as will be seen later.

8.4 Simplified alpha estimation algorithm

8.4.1 Determining the alpha parameter

The derivation of the expression for $\alpha(x,y,t)$ in the proposed method is very similar to that of the FPE method covered in section 6.2.2, except that it is much simpler. Details of the derivation are now given below.

Real flicker found in films are known to be varying in a non-abrupt manner (spatially speaking) and thus the flicker gain $\alpha(x,y,t)$ can be assumed to be spatially smooth. For this reason, $\alpha(x,y,t)$ can be considered as constant locally, i.e., $\alpha(x,y,t) = \alpha_{m,n}(t)$, $\forall (x,y) \in \Omega_{m,n}$, where $\Omega_{m,n}$ is a small block typically created by partitioning the image frame into equal-sized rectangular blocks, m and n are respectively the row and column positions of the image block.

Hence for the small block $\Omega_{m,n}$, equation 8.1 can be written as:

$$Y(x,y,t) = \alpha_{m,n}(t)I(x,y,t) \quad (8.2)$$

Expressions for $\alpha_{m,n}(t)$ can be derived by determining the expected value of $Y(x,y,t)$ given in equation 8.2. The expected value of $Y(x,y,t)$ in equation 8.2 for $x,y \in \Omega_{m,n}$ is given by:

$$E[Y(x,y,t)] = E[\alpha_{m,n}(t)I(x,y,t)]$$

$$= \alpha_{m,n}(t)E[I(x, y, t)]$$

$$\text{Hence } \alpha_{m,n}(t) = \frac{E[Y(x, y, t)]}{E[I(x, y, t)]} \quad (8.3)$$

Using the preceding frame (which had been corrected earlier) as a reference, the following approximation holds true for $x, y \in \Omega_{m,n}$ if only stationary scene is considered:

$$E[I(x, y, t)] \approx E[\hat{I}(x, y, t-1)] \quad (8.4)$$

where $\hat{I}(x, y, t-1)$ indicates the intensity of the image at location (x, y) in the preceding corrected frame. Hence equation 8.3 can now be re-written as:

$$\alpha_{m,n}(t) \approx \frac{E[Y(x, y, t)]}{E[\hat{I}(x, y, t-1)]} \quad (8.5)$$

Equation 8.5 has a limitation in that it only holds true for stationary scene. This constraint can be removed and an expression that is valid for both stationary and motion regions can be obtained if the motion vectors V_H and V_V for the small block $\Omega_{m,n}$ under consideration can be found. As a matter of fact, it turns out that exactly the same motion vector determination method developed earlier in chapter 7 section 7.2.2 is applicable over here. With motion vectors determined, equation 8.5 can now be re-written as:

$$\alpha_{m,n}(t) \approx \frac{E[Y(x, y, t)]}{E[\hat{I}(x - V_H, y - V_V, t-1)]} \quad (8.6)$$

8.4.2 Checking value of the alpha parameter

There are situations under which the determination of motion vectors is not reliable. These include: occlusion, motion smear, non-translational motion (caused by deformable objects, zooming etc.), and background of uniform intensity. The example given below in figure 8.1 that involves occlusion will be used to illustrate this problem.

Frame 1

Frame 2

Frame 3

Figure 8.1 An example illustrating the occlusion problem

Figure 8.1 shows 3 consecutive frames from a movie sequence known as the “Sword Fighting” sequence. It can be seen that the human face at the background was progressively uncovered frame by frame. For these uncovered areas, it is not possible to carry out any reliable motion vector determination due to the lack of information in the preceding frame. There is also another occlusion problem in this film sequence. It can be seen that in frame 3, an object (a sword) abruptly appears at the top left corner of the frame. Again under this kind of condition, no proper matching can be done for the occluded area and any motion vector determined will be highly unreliable.

Frame sequence that contains motion smear and non-translational motion will also face the same kind of problem where there will be regions for which no match can be found in the preceding frame for proper motion vector determination. In all these cases the motion vectors determined are deemed to be highly unreliable.

There is another scenario that will also give rise to unreliable motion vector determination: when there are many blocks within the search region that are more or less the same as the searched block, thus resulting in many probable candidates for the matching block. Let us use the example of a frame where the search area is a region of uniform intensity, such as the wall in a room, for illustration. It is obvious under this situation, the computed MAD for all blocks within the search area will be very close in value. Due to the presence of noise, the block that returns MAD_{min} is not necessarily the actual matching block; other blocks with MAD value very close to MAD_{min} are also probable candidates. For this reason, the pair of motion vectors determined via MAD_{min} cannot be fully trusted.

A simple rule is proposed here to identify this type of unreliable motion vectors: Within the search region, if there are 10% of blocks with MAD value that falls within $1.2MAD_{min}$, then it will be concluded that there are too many probable matching blocks. Under this situation, the matching block found that gives MAD_{min} cannot be trusted, and the corresponding pair of motion vectors will be considered unreliable.

For all the motion vectors that are deemed unreliable, the same can be said for the corresponding values of $\alpha_{m,n}(t)$ determined (via equation 8.6). It is important to identify these unreliable values of $\alpha_{m,n}(t)$ as they will result in erroneous flicker correction; a direct consequence of this is artifacts will be introduced in the restored frame.

A simple threshold-based method is proposed here to address this problem. If the value of $\alpha_{m,n}(t)$ falls outside a range defined by $(\alpha_{min}, \alpha_{max})$ where α_{min} and α_{max} are two properly chosen lower and upper bound values for $\alpha_{m,n}(t)$, then it is considered as invalid and the corresponding block $\Omega_{m,n}$ will be labeled accordingly. For all such blocks detected, their flicker gain values $\alpha_{m,n}(t)$ can be obtained using recursive interpolation, a method that had been described in detailed in chapter 5 section 5.4.2.1.

8.4.3 Smoothing the alpha value

With values of $\alpha_{m,n}(t)$ for all blocks determined, a two-step smoothing process is applied before the flicker gain values can be applied for flicker correction. The first step is on “spike” removal, followed by a second step that converts block-based flicker gain values into pixel-based values.

Knowing that real flicker is a non-spiky artifact that occurs rather spatially smoothly over the affected region of a frame, any “spike”, i.e., value of $\alpha_{m,n}(t)$ that differs significantly from those of its neighbors is most likely wrong and should be weeded out. If these spikes are not dealt with properly, small artifacts will appear in the corrected frame.

Median-filtering has long been known for its effectiveness in impulsive noise suppression [31,38-40] and is thus a suitable candidate to be considered for spikes removal here. A brief description of how a median-filter [42] works is given below.

To perform median filtering on a pixel, a suitable size window is first chosen. The mask is then put around the pixel with the pixel at the center. All the pixels within the mask are then sorted by value in either ascending or descending order, and the median is determined. The median is then assigned to that pixel under consideration. For example, if a 3x3 mask is used and the pixels are sorted in ascending order, then the median is the 5th largest value. In our work here, a 3x3 mask has been chosen for median filtering. When median-filtering is applied, it turns out that it is extremely competent in removing spikes among the flicker gain values. This is illustrated in figures 8.2(a) and 8.2(b).

The flicker gain values after spike removal are still block-based values; if they are used directly for flicker correction, visible sharp transition in intensity at boundaries of blocks will appear in the restored image and this is undesirable. Therefore, these block-based flicker gain $\alpha_{m,n}(t)$ values have to be converted to pixel-based $\alpha(x,y,t)$ values first before they can be used for flicker correction. As before, this can be easily achieved by using a simple low-pass filtering process over the whole frame.

Fig. 8.2 (a) Map of raw $\alpha_{m,n}(t)$

Fig. 8.2(b) Map of $\alpha_{m,n}(t)$ with spikes removed using median-filtering

8.4.4 Restoring image using the alpha parameter

With $\alpha(x,y,t)$ for all pixels determined, flicker correction can now be carried out based on equation 8.7, which is simply equation 8.1 re-written in a more convenient form.

$$I(x, y, t) = \frac{Y(x, y, t)}{\alpha(x, y, t)} \quad (8.7)$$

When the frame is restored, it will be used as a reference frame for the correction of the next frame, and the same process just repeats itself until the last frame in the sequence is done. Note that the order of correction is flexible in that it can be carried out in either a forward or a backward direction, the decision is entirely depending on which way is more suitable. For example, if degradation appears mainly over the front portion of a frame sequence, then it will make more sense to choose a good frame at the back of the sequence as a reference and restore backward. In cases of severe flicker where none of the frame in the sequence is good enough to be used as a reference frame, then a single frame must be manually corrected (using any of the standard image editing tools) first and it is then used as a reference for correcting others. Typically the least corrupted frame or the one that is easiest to be corrected manually will be chosen as the starting frame.

8.5 Experimental results and analysis

The same set of synthetic and natural test sequences used before are used here for testing the proposed method.

8.5.1 Synthetic test sequence

8.5.1.1 Global flicker

It can be seen in figure 8.3 that the intensity mean curve of the corrected sequence is substantially smoother than that before correction, with the variance dropping from 18.40 to 1.82, i.e., a 90.1% reduction. However, this smoother curve is really not that smooth when compared to the intensity mean curve of the original flicker-free sequence, which has a variance of only 0.71. It is noticed that the intensity mean curve of the corrected sequence has a tendency to “follow” the shape of that of the flicker sequence, as evident by the three gentle “humps” within the curve.

Fig. 8.3 Intensity mean curves of the low-frequency West sequence

8.5.1.2 Local flicker

Similar to the results for global flicker, the intensity mean curve of the corrected sequence is much flatter than that before correction, with the variance dropping from 77.34 to 7.12. This amounts to a drop of 90.8%, a figure very close to the 90.1% obtained for global flicker test earlier. Again, it is noticed that the intensity mean curve of the corrected sequence follows roughly the shape of that of the flicker sequence. In fact, the curve looks much like a “squashed-down” version of that of the flicker sequence.

Fig. 8.4 Intensity mean curves of the Taxi sequence

8.5.2 Natural test sequence

To get a better idea of the effectiveness in flicker correction of the proposed method, the correction results will be presented together with those of the FPE method (obtained in chapter 6) which are to serve as a reference here. Figure 8.5 shows respectively the intensity mean curve of the original flicker sequence, corrected sequence by FPE method, and corrected sequence by the proposed method.

It can be seen in figure 8.5 that the intensity mean curve of the corrected sequence by the proposed method is in general much smoother than that of the original flicker sequence. Visual checking on the corrected image sequence also confirms the reduction of flicker after correction.

A quantitative comparison of the flicker correction performance by the proposed method and the FPE method is now given. Figure 8.5 shows that the corrected sequence by the proposed method produces a smooth intensity mean curve comparable to that by the FPE method, with most part of the curve almost flat except at frames 6, 21 and 37 where there is a slight intensity jump in the form of a small spike. These frames correspond to those in the original flicker sequence where sharp rises of intensity are present. Overall the intensity mean curve corresponding to the proposed method gives a variance of 0.41, compared to 0.36 by the FPE method; this represents a 14% increase in variance. Judging by this small percentage increase in the variance value, it can be argued that under the simplified parameter model where the additive flicker offset term β is dropped, the performance of flicker correction has indeed gone down slightly but is by no means “excessive”. Visual checking of the corrected sequence also gives satisfactory results except that frames 6, 21 and 37 are noticed to be slightly brighter than the rest.

Fig. 8.5 Intensity mean curves of the Lady sequence

Similar results were obtained when tests were carried out on the Tunnel and Lantern sequences. A comparison of the respective variances of the intensity mean curves before and after correction is given in Table 8.1. Two main observations can be made from the table. The first observation is the variances corresponding to the corrected sequences by the proposed method are all much smaller than those before correction, ranging between 5.0% to 16.7%. Such figures confirm the effectiveness of the proposed method in correcting flicker. The second observation is the variances corresponding to the corrected sequences by the proposed method are consistently larger than those of the FPE method, suggesting that the proposed method is out performed by FPE method in all the three test sequences. However it should be pointed out that the performance disparity is very much sequence dependent as can be seen from the different variance values in Table 8.1.

Fig. 8.6 Intensity mean curves of Tunnel sequence

Fig. 8.7 Intensity mean curves of Lantern sequence

Table 8.1 Variance of intensity mean curves

Flicker correction method	Variance of intensity mean curve		
	Lady sequence	Tunnel sequence	Lantern sequence
No Correction	3.11	3.79	0.60
FPE method	0.36 (11.6%)	0.021 (0.6%)	0.017 (2.8%)
Simplified motion-compensated FPE method	0.41 (13.2%)	0.19 (5.0%)	0.10 (16.7%)

8.5.3 Discussions

Earlier in our examination of the correction result of the Lady sequence in section 8.5, it was mentioned that in the intensity mean curve of the corrected Lady sequence by the proposed method, small intensity spikes were noticed in frames 6, 21, 26 and 37. Although these spikes have already been much reduced from the original, the affected frames however still show up as slightly brighter frames when screening.

The fact that these spikes are few and in-between in the corrected sequence suggests that they can be easily removed using an earlier method developed in section 3.3 of Chapter 3, i.e., the Intensity Mean Averaging method. Frames 6, 21, 26 and 37 of the corrected sequence by the proposed method have respectively intensity mean of 82.00, 81.58, 80.84 and 83.2, which are considerably higher than 80.24, the median intensity mean of the corrected sequence. The corrected sequence obtained earlier by the proposed method can now be taken as a brand-new flicker sequence, with frames 6, 21, 26 and 37 as the flicker frames, and the Intensity Mean Averaging method can then be applied to correct these flicker frames.

By incorporating Intensity Mean Averaging into the propose method, spikes in the intensity mean curve of the corrected Lady sequence are successfully removed, and the result is a much improved corrected Lady sequence. This is illustrated in figure 8.8. With Intensity Mean Averaging incorporated, the intensity mean curve of the corrected Lady sequence gives a variance value of only 0.07, which is substantially better than 0.41 achieved before the incorporation. Note also that it is now performing better than the FPE method, which gives a variance of 0.36.

Fig. 8.8 Intensity mean curves of the Lady sequence

8.6 Conclusion

It has been shown in this chapter that the Simplified Motion-compensated FPE method, which uses only the flicker gain term to model flicker, to a large extent, is still able to effectively correct flicker despite such simplification. When compared to the FPE method which uses a full flicker model, the proposed method no doubt suffers a drop in the performance of flicker correction but it is noticed that the drop is within an acceptable range. The drop in performance is noticed to be dependent on the sequence under correction. Of the three natural flicker sequences under test, the Lady sequence suffers the least drop in performance whereas the Lantern sequence suffers the biggest. One possible explanation is that the Lady sequence may contain flicker that is quite adequately described by the simplified model, and so the effect of dropping the flicker offset term β did not cause very significant change in the correction result. In the case of the Lantern sequence, the simplified flicker model may not have described the flicker contained in the sequence very appropriately, and hence the correction result differs rather significantly from that of the FPE method, which uses a full model. This explanation borders on just a heuristic guess; certainly more work needs to be done to establish a full explanation.

It has been shown that the Intensity Mean Averaging method developed in chapter 3 can be incorporated into the propose method to help improve the correction results under certain conditions, such as when the corrected sequence produces an intensity mean curve that is largely flat but sprinkled with a few intensity spikes. Improvement from such a two-tier correction process can be quite substantial. As illustrated by the case of the Lady sequence, the proposed method managed to out-perform the FPE method by a large margin (evident by the small variance value of 0.07 in Table 8.2) after intensity mean averaging was incorporated.

Table 8.2 Variance of intensity mean curves

Flicker correction method	Variance of intensity mean curve (Lady sequence)
No Correction	3.11
FPE method	0.36 (11.6%)
Proposed method	0.41 (13.2%)
Proposed method with intensity mean averaging incorporated	0.07 (2.3%)

Chapter 9

Flicker Correction in Color Motion Pictures

9.1 Introduction

All flicker correction work that has been reported thus far in the research literature is confined to only black-and-white motion pictures. Because color films make up a large percentage of old film archives, it is imperative that ways can be found to address the flicker correction problem in color motion pictures. The novel work carried out in this chapter has successfully produced an answer to this challenge. A color flicker correction scheme based on the earlier work done in chapter 8 was successfully developed and it turns out that the method is very effective in correcting flicker in color motion pictures.

9.2 Color space – the fundamentals

There has been a reasonable amount of research work done in the area of flicker reduction for motion pictures, but to date to the best of the author's knowledge, none of these reported works involves color motion pictures. If flicker correction can be done in the color space [31], the implication will be far reaching. Once cleaned, these large volumes of archived old color films with flicker damage will have a chance for a new lease of life, and this will have great cultural and commercial significance. Let us begin our exploration by first taking a look at the fundamentals of the color space.

A color space, also known as "color model", is basically a system for describing color numerically. There are several color models that have come into existence for a variety of reasons, the three models that are most commonly used in practice are:

- RGB model – mainly used in scanners and color monitors
- YUV model – mainly used in TV broadcast
- CMY model – mainly used in color printing

Note that none of these color models is directly related to the way in which human beings perceive color. For this reason, other models that are more intuitive to the human eyes have been developed, such as the HSI (hue, saturation, intensity) model and the HSV (hue, saturation, value) model [31].

It should be pointed out that all the color models are convertible from one another. Such convenience is desirable because quite often it may be more convenient to perform certain operation in one color space and then another operation in a different color space. For example, in practice video images captured by studio video cameras or converted from films by telecines [72] are in the RGB format. However for subsequent processing, the RGB signals are in general converted to the more manageable YUV format. The YUV format is used in common video compression algorithms such as MPEG-1, MPEG-2 and MPEG-4 to produce digital content for storage in medium like VCDs and DVDs, or for broadcast in digital television [77].

9.2.1 The RGB color model

In the RGB (red, green, blue) model, three primary color components red, green and blue are added together to form a unique color. The model is based on a Cartesian coordinate system as shown in figure 9.1, in which RGB values (each normalized to 1) are at three corners, and the cube thus formed defines a color space within which all colors are defined. Color black is at the origin; and at the corner diagonally opposite the origin furthest away from it is the color white. The remaining three corners respectively represent color cyan, magenta, and yellow.

R Fig. 9.1 The RGB color model

Images in the RGB color model consist of three independent image planes, one for each primary color. When images from these three planes are fed into a RGB monitor, they combine together to form a composite color image. The RGB format is widely used in image buffering as most video cameras these days capture images in this format.

Figure 9.2 shows a sample color frame (of 720x576 resolution); figures 9.3 (a), (b) and (c) are its respective frame in the R, G and B channel.

Fig. 9.2 A frame from the color Lady sequence

Fig. 9.3(a) Channel R of the corresponding frame

Fig. 9.3 (b) Channel G of the corresponding frame

Fig. 9.3 (c) Channel B of the corresponding frame

9.2.2 The YUV color model

The YUV model is the color space used in the PAL system of color TV broadcasting, which is the standard used in most parts of Europe. It is particularly suited for this job for two reasons: transmission efficiency and backward compatibility with old monochrome TV standards. In this model, the intensity (Y) and color information (U and V) are decoupled. Hence a monochrome TV set will only use the intensity (Y) component of the received YUV signal for image creation; the U and V components are completely ignored altogether. A color TV set on the other hand will make use of all the components in the YUV signal received.

A sample of each of the Y, U and V frames for the color frame in figure 9.2 are shown respectively in figures 9.4 (a), (b) and (c).

Conversions between the RGB and YUV models are given by the matrix equations in the following page:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.148 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (9.1)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.0 & 1.140 \\ 1 & -0.395 & -0.581 \\ 1 & 2.032 & 0.0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \quad (9.2)$$

Fig. 9.4(a) Channel Y of the corresponding frame

Fig. 9.4(b) Channel U of the corresponding frame

Fig. 9.4(c) Channel V of the corresponding frame

9.2.3 The CMY color model

CMY (cyan, magenta, yellow) is the color space used for commercial printing and most color computer printers. Effectively it is a “subtractive” model in that each component is obtained by removing one of the three primary RGB components from white light. For example, when a surface coated with the cyan color pigment is illuminated with white light, no red light is reflected from the surface. In other words, red “subtracted” from the white light gives the color cyan. Likewise magenta is a result of subtracting green, and yellow comes from subtracting blue. Assuming that all color values have been normalized to the range $[0, 1]$, the equation for converting RGB to CMY is simply given by:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (9.3)$$

9.3 Flicker in color motion pictures

9.3.1 Intensity flicker

As discussed in chapter 2 section 2.3.2, intensity flicker is caused by brightness variations in pixels of a frame, and is commonly found in all black-and-white motion pictures. Intensity flicker can also be found in color video; this occurs when there are variations in the Y channel, as illustrated in figure 9.5.

It can be seen that for the given color film sequence, the intensity (Y) channel is the one that contains the largest number of spikes and the spikes are also among the biggest. For the color channels, i.e., the U and V components, only relatively mild fluctuations are noticeable. All these suggest that the flicker type in this particular color sequence is intensity flicker.

Fig. 9.5 Intensity flicker in a color sequence

Fig. 9.6 Color flicker in an old footage

9.3.2 Color flicker

Color flicker in color motion pictures refers to flicker caused by unequal fading of dyes in the film material. This has the effect of creating temporal fluctuations in the color of the video. An old footage that has developed some severe flicker in all the red, blue and green channels is used here for illustration. When this film sequence is analyzed in the YUV color space as shown in figure 9.6, it can be seen that sizable variations appear not only in the mean value curve of the intensity (Y) channel, but also in that of the color (U and V) channels.

9.4 Correcting flicker in color motion pictures

The algorithm developed in chapter 8 has been developed for correcting flicker in black and white sequences, i.e., correcting intensity flicker. Obviously, if this algorithm is to be used to correct intensity flicker in color motion pictures, all that is needed is to apply this algorithm to the intensity (Y) component of the color sequence and it should work. A natural question to ask next is: can this algorithm be used to correct color flicker in color motion pictures?

Let us go back to the source of color flicker and see if an answer can be found for this question. As explained in chapter 2 section 2.2, color films suffer from an additional flicker problem that is unique to color films: aging color film materials over the years will gradually develop color flicker, which is caused by uneven fading of the dye (cyan, magenta and yellow) in the film material. This means in the YUV color space, flicker will appear in not just the intensity component Y, but also in the color components U and V (as seen in figure 9.6). Likewise in the RGB color space, flicker will be present in all three color channels. It is thus obvious that just restoring the intensity component alone in a color sequence only offers a partial solution as color flicker is left untouched.

To explore how color flicker could be corrected in color motion pictures, let us begin our analysis in the YUV color space. Earlier it was learned in section 9.2.2 that in the YUV model, the Y component contains the intensity information whereas the U and V components contain the color information. Due to the presence of color flicker, it is obvious that the flicker correction process would not be complete unless correction is done in all three components.

When the algorithm developed in chapter 8 was put to test in the intensity component Y, the results obtained showed that the algorithm was working fine. This is an expected result because after all the algorithm was originally developed for intensity correction. As far as the algorithm is concerned, when it reads in intensity, it sees no difference whether the input is from black and white or color films.

The algorithm however, did not work well for the U and V components. This is mainly due to the lack of variance in the U and V values as clearly demonstrated in figures 9.4 (b) and (c). For this reason, it is near impossible to carry out any kind of motion estimation in the U and V channels. This problem is further aggravated in the presence of flicker.

Switching over to the RGB color space provides a solution to this problem. Each of the RGB channels actually contains almost as much information as the original frame; this is clearly demonstrated in figures 9.3 (a), (b) and (c). This means each of them can be treated as a monochrome frame, which can be restored separately using the algorithm developed in chapter 8. When this is done, the three restored R, G and B frames can be converted back to the YUV color space using equation 9.1, and as a result, a flicker-free YUV frame can be successfully restored.

Restoration of each of the R, G and B channels requires determination of motion vectors. However, since the R, G and B channels are all from the same frame, the motion vectors in each of the three channels are actually identical at the same location. This means that motion vector determination is only needed to be performed in a chosen channel and the same motion vector can then be applied in other channels. In color motion pictures, it is often the case that the channels are unequally corrupted; obviously, for accurate motion vector determination, the cleanest channel should be chosen for the job.

An alternative to this approach is to use the intensity (Y) channel in the YUV color space for motion vector determination and later apply it in each of the R, G and B channels for restoration. This approach is simpler in that there is no need to check first which of the R, G and B channels is the cleanest before motion vector determination can be initiated. For this reason, this approach has been chosen in our restoration work here.

Figure 9.7 is a flow-chart that describes the steps involved in our proposed method of color flicker correction:

Fig. 9.7 Steps in flicker correction in color motion pictures

9.5 Experimental results and Analysis

9.5.1 Synthetic flicker test sequence

Instead of the usual coverage of both global and local flicker test sequences, it is decided that only the synthetic local flicker sequence will be covered here. This is because most of the color flickers encountered in real life are caused by unequal fading of dyes in the film material, which is typically local in nature.

Local flicker of varying severity were added to each of the R, G and B channels of the frames in the 20-frame Fountain sequence. When the R, G and B channels are combined together, severe fluctuation in color is clearly visible during play-back of the sequence. Shown in figure 9.8 are a few consecutive frames from the flicker Fountain sequence.

Figure 9.8 Four consecutive frames from the Fountain sequence with local flicker added

When the proposed algorithm was applied to the flicker Fountain sequence, color flicker that was previously very visible was all gone, and there were no artifacts introduced in the correction process. This is clearly demonstrated by the corresponding restored frames shown in figure 9.9.

Figure 9.9 The corresponding four frames from the corrected Fountain sequence

The graphs in figure 9.10 (a), (b) and (c) respectively show the intensity mean curves of the original, flicker and corrected sequences in each of the R, G and B channels, whereas those for the Y channel in the YUV color space are shown in figure 9.11.

Fig. 9.10(a) Intensity mean curves – channel R of the Fountain sequence

Fig. 9.10(b) Intensity mean curves – channel G of the Fountain sequence

It can be seen that the intensity mean curves of the corrected Fountain sequence in the R, G, B and Y channels are much smoother than those before correction. Table 9.1 gives a quantitative comparison in terms of the variance of the intensity mean curves, which shows that the variance value of the corrected Y channel is only about 5.3% of that before correction. For individual R, G and B channels, the corresponding figures are in the range between 7% to 11%.

Fig. 9.10(c) Intensity mean curves – channel B of the Fountain sequence

Figure 9.11 Intensity mean curves for the Y channel

Table 9.1 Variance of intensity mean curve of flickered Fountain sequence

		variance
Flickered Fountain sequence	R channel	27.5
	G channel	28.44
	B channel	23.60
	Y channel	12.45
Corrected Fountain sequence	R channel	2.08
	G channel	2.35
	B channel	2.72
	Y channel	0.66

On PSNR comparison, it can be seen from figure 9.12 that the PSNR curve for the corrected sequence is consistently higher than that of the flicker sequence, suggesting improvement over all frames. The corrected sequence has an average PSNR value of 49.41, compared to 37.34 before correction; this represents an improvement of 12.07 dB.

Figure 9.12 PSNR curves of Fountain sequence

9.5.2 Natural test sequence

The proposed method has been tested with a few flicker-damaged color frame sequences and found to be effective in all. Only the correction results of a naturally degraded 40-frame color Lady sequence will be presented in detailed here for illustration purpose.

Fig. 9.13 Intensity mean curve of R, G & B channels of color Lady sequence

Shown in figure 9.13 are the respective intensity mean curves of the R, G and B channels of this sequence. Figures 9.14(a), (b) and (c) show respectively the intensity mean curves of the R, G and B channels in the original flicker sequence, and those of the corrected sequence by the proposed method.

It can be seen that for the R channel, the intensity mean curve after correction is much smoother than that before correction. For the corrected sequence, there are a few small spikes in the intensity mean curve; this is a phenomenon known to the Simplified Motion-compensated FPE algorithm and has been discussed in section 8.5 of Chapter 8. Similar results are also obtained when the same set of intensity mean curves are plotted for the G and B channels, suggesting effective flicker correction in these two channels. With flicker corrected for all the three RGB channels, they are converted back to the YUV color space where similar comparison is made in the intensity (Y) channel for the original and corrected frames. Again, similar results (see figure 9.15) are obtained thus confirming the effectiveness of the proposed method in color flicker correction.

Fig. 9.14(a) Intensity mean curves – Channel R of the color Lady sequence

To quantitatively evaluate the performance of flicker correction by the proposed method, an examination on the variance of the intensity mean curve before and after correction is needed; these information is shown in Table 9.2.

Evaluation of the correction performance is carried out by first examining the individual RGB channels, followed by Y channel of the YUV color space. For individual RGB

channels, let us choose one of the channels, say the R channel, for illustration. It can be seen that for the R channel, the intensity mean curve after correction gives a variance of 0.81, which is only 14.3% of that ($= 5.68$) before correction. Improvements of similar order are also observed in the G and B channels. The overall effectiveness of the proposed method is confirmed when checking is done on the Y channel, where the intensity mean curve after correction gives a variance of 0.35, which is only 11.3% of that before correction. Finally, visual checking is made on both the original and corrected image sequences. As expected, the much reduced flicker in the corrected sequence gives a much more pleasant viewing experience than the original flicker sequence.

Table 9.2 Variance of intensity mean curve of color Lady sequence

		variance
original flicker sequence	R channel	5.68
	G channel	2.22
	B channel	2.54
	Y channel	3.11
corrected sequence	R channel	0.81
	G channel	0.38
	B channel	0.30
	Y channel	0.35

Fig. 9.14(b) Intensity mean curves – Channel G of the color Lady sequence

Fig. 9.14(c) Intensity mean curves – Channel B of the color Lady sequence

Fig. 9.15 Intensity mean curves – Channel Y of the color Lady sequence

Similar tests on the color Woman sequence and the color Lantern sequence were carried out and the results are given in figures 9.16, 9.17 and Table 9.3.

Fig. 9.16 Intensity mean curves -- Channel Y of the color Woman sequence

Fig. 9.17 Intensity mean curves -- Channel Y of the color Lantern sequence

Table 9.3 Variance of intensity mean curves

Flicker correction method	Variance of intensity mean curve		
	Lady sequence	Woman sequence	Lantern sequence
No Correction	3.11	3.16	0.60
Proposed method	0.35 (11.3%)	0.39 (12.3%)	0.07 (11.6%)

9.5.3 Discussions

Earlier in section 9.5.2 in our evaluation of the correction results of the color Lady sequence, it was mentioned that intensity spikes were noticed in the respective intensity mean curve of the R, G and B channels. Obviously, these spikes can be removed by incorporating Intensity Mean Averaging into the proposed method in the same way as in Chapter 8, except that it is to be carried out separately in each channel.

Fig. 9.18 Intensity mean curves – Channel Y of color Lady sequence

When respective result of the R, G and B channels are put together and converted back to the YUV color space, as expected, a much improved intensity mean curve is obtained. This is illustrated in figure 9.18. With Intensity Mean Averaging incorporated, the intensity mean curve of the corrected color Lady sequence gives a variance value of only 0.06, which is substantially better than 0.35 achieved before the incorporation.

Similar improvement can also be made to the color Woman sequence when Intensity Mean Averaging is incorporated. The results are shown in Table 9.4.

Table 9.4 Variance of intensity mean curves

Flicker correction method	Variance of intensity mean curve	
	Lady sequence	Woman sequence
No Correction	3.11	3.16
Proposed method	0.35 (11.3%)	0.39 (12.3%)
Proposed method with intensity mean averaging incorporated	0.06 (1.9%)	0.26 (8.2%)

9.6 Conclusion

In this chapter, the Simplified Motion-compensated FPE method developed in chapter 8 was employed to correct flicker in color motion pictures. It was found to be effective and test results with two natural color sequences “Lady” and “Woman” showed that the respective variance of the intensity mean curves was substantially reduced to only 11.3% and 12.3% of the original after correction. Correction was substantially improved when Intensity Mean Averaging was incorporated, which further reduced the respective percentage variance value to only 1.9% and 8.2%.

Chapter 10

Stain Removal

10.1 Introduction

Color motion pictures are known to suffer from another problem – stain [76]. This defect is actually an intensity-related problem and can be suitably represented by the model developed in chapter 8. It turns out that with some slight modification, the algorithm developed in chapter 8 can be successfully applied to stain removal in color film sequences. Test results on a few stain-damaged film sequences show that the proposed method is very effective for stain removal.

10.2 Stain problem in color motion pictures

Other than flicker, another common artifact found in old color motion pictures is stain, which is typically caused by some kind of severe damage on the surface of the film. Such damage results in either a reduction of the original signal in the affected area, or, in the worst case, removes the signal altogether. This has the adverse effect of creating unsightly patches when the film is projected onto a screen, or when it is scanned into another medium. An example is given in figure 10.1, which shows the second frame (frame 2) from a color sequence (known as the “Sword-Fighting” sequence) that has been damaged by stain. Most of the stain which appears as a greenish smear is clearly visible in the top right-hand quadrant of the frame; smaller stained regions can also be identified near the centre as well as at the lower right-hand corner.

Watching a stain-damaged motion picture is irritating because the stain is normally very visible to the human eyes. To make matter worse, a film sequence with stain problems will in general have a high percentage of frames that are stain-corrupted, thus making the whole viewing process very unpleasant or even painful.

10.3 A method for stain removal

To explore further into the stain problem, let us split the stain-corrupted frame 2 shown in figure 10.1 into the red (R), green (G) and blue (B) channels. The resulting frames in the R, G and B channels are shown respectively in figures 10.2, 10.3 and 10.4.

Fig. 10.1 Frame 2 (stained-corrupted) from the Sword-Fighting sequence

Fig. 10.2 Frame 2 -- channel R

Fig. 10.3 Frame 2 -- channel G

Fig. 10.4 Frame 2 -- channel B

A close examination of the R, G and B frames reveals an interesting observation in that the extent of stain damage in them are very different. It can be seen that stain appears predominantly in the G channel; the B channel is only slightly stained, and the R channel is almost free from any stain damage. An examination of other frames of the Sword-Fighting sequence also gives similar results.

What has just been discovered seems to be a common phenomenon, i.e., stain is primarily in one of the channels and at least one of the RGB channels is clean. A study on a few more stain-corrupted color motion pictures seems to support our observation. For this reason, a stain model exploiting this special phenomenon can be readily developed to address the stain removal problem; this is to be illustrated in the next section.

10.3.1 Stain model

As can be seen in figures 10.2, 10.3 and 10.4, the presence of stain has the effect of causing a change in the intensity level in the affected areas. This is a scenario very similar to the case of flicker, for which a simplified working model has already been successfully established in chapter 8. For this reason, a similar model will be proposed here for the stain model in each of the RGB channels, i.e.:

$$Y_R(x,y,t) = \alpha_R(x,y,t) I_R(x,y,t) \quad (10.1a)$$

$$Y_G(x,y,t) = \alpha_G(x,y,t) I_G(x,y,t) \quad (10.1b)$$

$$Y_B(x,y,t) = \alpha_B(x,y,t) I_B(x,y,t) \quad (10.1c)$$

In the above equations, the Y term is the intensity of the pixel at location (x,y) and t is the frame number, the I term refers to the original intensity (i.e. with no stain damage), and the α term is known as the “stain factor” which is a measure of the amount of stain. For a stain-free pixel, the value of α is 1.0. The subscripts R, G and B are used to respectively indicate the red, green and blue channels.

10.3.2 Stain detection

Before steps can be taken to remove stain, ways must be found to detect stain in a frame. Our proposed approach here is to do stain detection separately in the R, G and B channels. For simplicity, all the expressions presented henceforth will not carry any subscript R, G, or B to indicate the specific channel. It is understood that all the expressions are applicable to the R, G or B channel, unless otherwise stated.

To set the stage for discussion, let us assume that in the stain-corrupted color sequence, there exists a good frame that can be used as a reference. In general, either the preceding frame or the succeeding frame is chosen as the reference. The reference frame is assumed to be stain-free, as it would have gone through restoration earlier.

Real stains found in films are known to be either uniform or spatially varying very gradually. For this reason, $\alpha(x,y,t)$ can be considered as constant locally, i.e., $\alpha(x,y,t) = \alpha_{m,n}(t)$, $\forall (x,y) \in \Omega_{m,n}$, where $\Omega_{m,n}$ is a small block typically created by partitioning the image frame into equal-sized rectangular blocks, m and n are respectively the row and column locations of the image block.

Using similar argument developed for flicker correction in chapter 8, the following stain expression can be deduced:

$$\alpha_{[RGB]m,n}(t) \approx \frac{E[Y(x,y,t)]}{E[\hat{I}(x-V_H, y-V_V, t-1)]} \quad (10.2)$$

where $Y(x,y,t)$ is the intensity of the current frame at location (x,y) , $\hat{I}(x-V_H, y-V_V, t-1)$ is the intensity of the previous (reference) frame in which V_H and V_V are respectively the motion vectors in the horizontal and vertical directions, m and n

are respectively the row and column locations of the image block, E is the expected value, and $[RGB]$ indicates the red, green or blue channel.

Note that the same method as illustrated in chapter 7 section 7.2.2.1 is employed here for motion vector determination. Since the motion vector pair V_H and V_V is common to all the RGB channels, it needs to be determined only in one and not all channels. Obviously the cleanest channel should be chosen to achieve the best accuracy in determining motion vector pairs.

For argument sake, let us assume that i^{th} frame (f_i) in the stain-corrupted channel G is to be cleaned up. For a stain-damaged block in f_i , the intensity will be considerably higher or lower than its original (stain-free) value; this implies that its stain factor $\alpha_{m,n}(t)$ will deviate significantly from the value 1.0. By comparing the value of the stain factor with a properly chosen threshold range ($\alpha_{\min}$, $\alpha_{\max}$), the stain-damaged blocks can be readily detected.

However, as pointed out in chapter 8 section 8.4.2, there are situations under which the determination of motion vectors is not reliable and consequently lead to the value of $\alpha_{m,n}(t)$ deviating significantly from 1.0. These include: occlusion, motion smear, non-translational motion, and background of uniform intensity. This means that by checking the value of $\alpha_{m,n}(t)$ against ($\alpha_{\min}$, $\alpha_{\max}$), the stain-damaged blocks captured are only the “apparent” ones (hence-forth known as the “apparent” stain-damaged blocks). They actually contain not only the actual stain-damaged blocks, but also those “pseudo” stain-damaged blocks, which are actually blocks with un-reliable motion-vector pairs due to the afore-mentioned reasons.

It is obvious that in order to find the real stain-damaged blocks, all that is needed is to first identify the “pseudo” stain-damaged blocks in f_i . Once done, subtracting these blocks from the “apparent” stain-damaged blocks will give us the real stain-damaged blocks. The question now is: how can these “pseudo” stain-damaged blocks in f_i be identified?

The answer to this question lies in the cleanest channel, which in practice can be identified by visually checking through the R, G and B channels. Assuming that among the three channels, channel R is the cleanest. It is obvious that if f_i contains occlusion, and motion smear etc., they will appear in all the channels in exactly the same manner. When

the same threshold window ($\alpha_{\min}$, $\alpha_{\max}$) used for channel G is used in channel R of f_i , the set of “pseudo” stain-damaged blocks detected there will be identical to that in channel G of f_i , and hence it can be used as a substitute for that in channel G. Note that because channel R is stain-free, checking the value of $\alpha_{m,n}(t)$ against the threshold window ($\alpha_{\min}$, $\alpha_{\max}$) in channel R of f_i will yield directly the “pseudo” stain-damaged blocks. This is not the case in the stain-damaged G channel, where both real and “pseudo” stain-damaged blocks in f_i are detected together at the same time.

With the set of “pseudo” stain-damaged blocks in channel R of f_i found and used as a substitute for that in channel G of f_i , the real stain-damaged blocks in channel G of f_i are readily found as described earlier. That is, for every detected block in channel G of f_i , check it against the corresponding detected block in channel R of f_i . If this block is also detected in channel R, then it is not a real stain-damaged block; otherwise it is. Repeating this process until all the detected blocks in channel G of f_i are covered will yield all the real stain-damaged blocks in channel G of f_i .

The real stain-damaged blocks in frames of the other stain-corrupted channel (channel B) can be detected in exactly the same manner.

10.3.3 Stain removal

With the stain-damaged blocks in the stain-corrupted channels successfully detected, the next step is to restore pixels within these blocks back to normal, i.e., restore back to their original intensities. This is to be carried out using equations 10.3a -10.3c, which are simply equations 10.1a -10.1c re-written in a more convenient form:

$$I_R(x,y,t) = Y_R(x,y,t)/\alpha_R(x,y,t) \quad (10.3a)$$

$$I_G(x,y,t) = Y_G(x,y,t)/\alpha_G(x,y,t) \quad (10.3b)$$

$$I_B(x,y,t) = Y_B(x,y,t)/\alpha_B(x,y,t) \quad (10.3c)$$

For all the stain-damaged blocks, the stain factor values to be used for stain correction are those obtained earlier using equation 10.2, whereas for all the non-stain-damaged blocks, a stain factor value of 1.0 should be used.

Note that the stain factor values obtained from equation 10.2 are block-based values and so they have to be converted back to pixel-based values first before they can be used in equations 10.3a -10.3c for stain correction. This can be easily achieved by going through a standard low-pass filtering process over the whole frame before carrying out the stain correction process.

10.4 Experimental results and Analysis

10.4.1 Natural stain test sequence

The proposed method has been tested with a few natural stain-damaged sequences, and the results for two of the test sequences, namely the Sword-Fighting sequence and the Shock sequence will be given.

A brief description of these two test sequences will now be given.

The 10-frame Sword-Fighting sequence shows continuous shots of a Kung-Fu fighting scene with the two main figures occupying the left and right sides of the frame. There is a third man near the middle at some distance away from the two men. The man on the right is seen charging towards the man on the left, and in the process, the man in the centre is progressively covered by the charging man, until completely covered on the last frame. At the background of the sword-fighting scene are a few pine trees, gently rolling hills and a large stretch of open sky. The whole sequence is stain-corrupted except for frame 1, which is relatively stain-free. Stains of different shapes and sizes appear greenish in color, and are mainly on the right-half portion of each frame.

The Shock sequence is a 21-frame sequence which shows a series of close-up shots of a man at the centre of the frame. The camera-facing man is seen staring hard at something and his facial expression undergoing a series of changes, from being captivated, mildly awed to eventually totally shocked by what he sees. There are some tree branches in the foreground on the right-hand side of the frame, and at the background is a clear blue sky. Stains of varying degree of severity appear randomly in the sequence, with most of them appearing in the form of greenish spider-web across each frame.

10.4.1.1 Correction results using an imperfect reference frame

A detailed study of the stain correction process will be presented using the correction of frame 2 in the Sword-Fighting sequence for illustration. The Sword-Fighting sequence is known to have a clean (stain-free) R channel, and two moderately stain-corrupted G and B channel. Let us begin with channel G and choose a threshold window ($\alpha_{\min}$, $\alpha_{\max}$) of (0.95, 1.05) [section 10.4.1.1.1 explains how this threshold window is obtained] to identify the “apparent” stain-damaged blocks. They are as shown below in white blocks in figure 10.5.

Fig. 10.5 “Apparent” stain-damaged blocks in channel G of frame 2

As explained in section 10.3.2, the “apparent” stain-damaged blocks actually contain both the real stain-damaged blocks and “pseudo” stain-damaged blocks and the latter are actually blocks linked to factors like occlusion and motion smear etc., which give rise to unreliable motion vector detection.

These “pseudo” stain-damaged blocks in channel G are identical to those in the clean channel R, and they can be easily detected by applying the same threshold window (0.95, 1.05) in channel R. The white blocks in figure 10.6 show the “pseudo” stain-damaged blocks identified in channel R.

Fig. 10.6 Pseudo stain-damaged blocks in channel R of frame 2

It can be seen that the detected blocks are fewer, and they form a subset of those detected in figure 10.5. These “pseudo” stain-damaged blocks, as discussed earlier, are regions of unreliable motion vector pairs. An analysis of figure 10.6 lends support to this argument. Large areas of “pseudo” stain-damaged blocks are seen detected in regions near the head and shoulder of the person on the right. These regions are characterized by a uniform color and texture, and it is a known fact that under such conditions, detection of motion vector pairs will be highly unreliable.

Knowing that these “pseudo” stain-damaged blocks in channel R are identical to those in channel G, the real stain-damaged blocks in channel G can now be easily obtained by subtracting these “pseudo” stain-damaged blocks in channel R from the “apparent” stain-damaged blocks in channel G. The white regions in figure 10.7 below show all the real stain-damaged blocks identified in channel G. Visual checking shows that the real stain-damaged blocks shown in figure 10.7 tally well with the stain regions that are clearly visible in figure 10.3.

Fig. 10.7 Real stain-damaged blocks in channel G of frame 2

The real stain-damaged blocks in channel B of frame 2 can also be identified in exactly the same manner. Shown in figures 10.8 and 10.9 are respectively the “apparent” stain-damaged blocks and real stain-damaged blocks in channel B. It can be seen there are very few stain-damaged blocks detected in channel B. This is an expected result as it is known that there are indeed very little stain in channel B of frame 2 as can be seen in figure 10.4.

Fig. 10.8 “Apparent” stain-damaged blocks in channel B of frame 2

Fig. 10.9 Real stain-damaged blocks in channel B of frame 2

With real stain-damaged blocks in both channel G and B detected, the stain removal step can now be carried out as described in section 10.3.3. A stain-removed G-channel frame 2 and also a stain-removed B-channel frame 2 will be produced. These two stain-removed frames can subsequently be combined with the original clean R-channel frame 2 to create the composite color frame 2, which is shown in figure 10.10 below.

Fig. 10.10 Stain-removed color frame 2

10.4.1.1.1 Discussions

It can be seen in figure 10.10 that most of the stain in the original color frame has been successfully removed. However there are remnants of stain that are still visible in small pockets of regions in the frame. The most visible un-removed stain can be found in some small areas in the top right-hand quadrant of the frame, as well as near the head of the person on the right.

There are two main reasons for the less-than-ideal stain removal results obtained here. The first is related to the detection of the stain-damaged blocks in the stain-corrupted frame, whereas the second is linked to the reference frame used in the stain removal process. An elaboration will now be given below.

The first main reason for the presence of remnants of stain in the restored frame is due to the failure in fully capturing the stain-damaged blocks in the corrupted frame. The performance of our proposed stain-detection algorithm depends very much on the chosen threshold value for $(\alpha_{\min}, \alpha_{\max})$. As there is no hard-and-fast rule for choosing the optimal threshold value that will most accurately detect the stain regions. The results presented here have been based on a chosen stain factor threshold range of (1.05, 0.95). This chosen threshold range was the result after experimenting with about five sets of different values, and within this limited number of experimentations, this particular set has produced the best stain removal results.

A poor choice of threshold range will result in poor stain-damaged block detection, and consequently produce poor stain removal results. Let us use a poorly chosen stain factor

threshold of (1.1, 0.9) for illustration. The white blocks shown in figure 10.9 are the real stain blocks detected in the G channel using this particular threshold. A comparison with the actual stain-damaged G channel frame 2 in figure 10.3 clearly shows that the detected stain-damaged regions are smaller than what they should be, and it is most obvious in the two encircled regions highlighted in figure 10.11. Similar observation also applies to the B channel, although it is to a lesser extent. Shown in figure 10.12 is the restored stain-removed color frame. Overall the correction result is poorer than that shown in figure 10.10, which was based on a better-chosen threshold of (1.05, 0.95). There are more stain-damaged areas that failed to get removed, most notably in the areas that correspond to the two encircled regions in figure 10.11. This is an expected result because with fewer stain-damaged blocks detected, there will be fewer stain-damaged blocks that get restored.

Fig. 10.11 Real stain-damaged blocks in channel G of frame 2 [threshold = (1.1, 0.9)]

Fig. 10.12 Stain-removed color frame 2 [threshold = (1.1, 0.9)]

The second main reason for the presence of remnants of stain in the restored frame is due to the imperfect reference frame used in our correction here. The whole basis of our correction algorithm is built upon the assumption that a neighboring “perfect” (i.e. completely stain-free) frame is available as a reference to correct the current stain-damaged frame. Unfortunately the preceding frame (frame 1) that was used as a reference here is not entirely stain-free (see figure 10.13), and as a result the stain factor values determined from equations 10.1a -10.1c will certainly contain some errors. As these stain factor values are used in both detecting stain-damaged blocks and also later in correcting the stain-damaged blocks (using equations 10.3a -10.3c), errors will invariably creep into these two processes. In the former, error will appear in the form of wrong detection of stain-damaged blocks, whereas in the latter, error will lead to wrong intensity correction (i.e. corrected to the wrong intensity level) in certain detected stain-damaged blocks. The only way to get around this problem is to manually correct the reference frame first so that a “perfect” reference frame is available for correcting the frame sequence. This can be done through some special image editing software tools, such as Adobe Photoshop.

Fig. 10.13 An imperfect reference (preceding) frame used for correcting frame 2

10.4.1.2 Correction results based on a perfect reference frame and a better threshold window

It was argued in section 10.4.1 that two main factors have contributed to the less-than-ideal stain removal results: poor choice of threshold window ($\alpha_{\min}$, $\alpha_{\max}$) and imperfect reference frame used in correction. Further work guided by these findings will now be explored to see if better stain removal results can be achieved.

A much better reference frame was first created through manual correction with the help of image editing software mentioned above. The same reference frame in figure 10.13 was manually corrected and the corrected frame is shown in figure 10.14. Visual checking shows that faint traces of stain that are most visible on the top right-hand corner of the original frame are no longer there. Further visual examination on the whole frame gives us the impression that most stains, if not all, have been successfully removed. As such, the frame in figure 10.14 can now be used as a “perfect” reference frame to correct frame 2; it is envisaged that it will do a better job than the original reference frame shown in figure 10.13.

Fig. 10.14 A perfect reference (preceding) frame used for correcting frame 2

With a “perfect” reference frame now available, the next refinement step is on choosing a better threshold window ($\alpha_{\min}$, $\alpha_{\max}$). After much experimentation through visual checking, it was found that a threshold window of (1.035, 0.965) was able to produce the best correction result. The corrected frame based on this threshold window is given in figure 10.15. It is appropriate now to compare this improved corrected frame with that obtained earlier in section 10.4.1, i.e., figure 10.10. Figure 10.10 is reproduced in figure 10.15 for easy comparison.

Remnants of stain that were present in figure 10.10 in areas such as the sky on the top right-hand corner, among the tree leaves in the middle, and on the ground on the bottom right corner are now all gone in figure 10.15, suggesting a much improved, if not perfect, stain removal effect. Further test on subsequent frames of the same sequence delivered equally good stain removal results, as demonstrated by a comparison between another two before-and-after-correction frames in the Sword-Fighting sequence shown in figure 10.16 and figure 10.17.

Fig. 10.15 Stain-removed frame 2 [“perfect” reference frame, threshold = (1.035, 0.965)]

Fig. 10.10 Stain-removed frame [imperfect reference frame, threshold = (1.05, 0.95)]

Fig. 10.16 Another stain-damaged frame from the Sword-fighting sequence

Fig. 10.17 The corresponding stain-removed frame

Given in figure 10.18 and figure 10.19 are two before-and-after-correction frames taken from another sequence, the Shock sequence. Again, very good stain removal results are obtained.

Fig. 10.18 A stain-damaged frame from the Shock sequence

Fig. 10.19 The corresponding stain-removed frame

Judging by the much improved stain-removal results given in section 10.4.1.2, it can be said that the proposed stain model and the whole stain-removal method developed based on this model provide a highly effective solution to the problem of stain-removal in color motion pictures. The only drawback of the proposed method is there is no hard-and-fast rule in determining an optimal threshold window to guarantee the best correction result; much trial-and-error is needed which could be rather tedious and time-consuming.

10.4.2 Synthetic stain test sequence

In order to measure in a quantifiable manner the effectiveness of the stain removal algorithm, as well as to establish a benchmark for comparison, quantitative analysis is a necessary part of the whole testing process. Typically, this is carried out by first adding artificial stains to a clean image sequence to create a synthetic stain test sequence; the synthetic sequence is then stain-corrected and finally a comparison is made between the restored and original sequences from which quantitative measurement can be made.

Frame 1 (clean)

Frame 2 (artificial stain added)

Fig. 10.20 Two consecutive frames from the Wedding sequence

To illustrate, two consecutive frames taken from the Wedding sequence are shown in figure 10.20; the first frame is clean whereas the second frame has been added with artificial stains in the form of two square blocks.

To mimic the real-world situation whereby stains in color motion pictures typically occur predominantly in just one single channel, stain blocks in the Wedding sequence have been purposely added only in the R channel. When the stain detection algorithm is applied, all the stain blocks added in the synthetic sequence are successfully identified. Let us use frame 2 in figure 10.20 for illustration. The stain blocks there are detected and are shown as two white blocks in the stain detection map of figure 10.21.

While the size of the stain blocks detected in figure 10.21 may appear visually to be identical to that in frame 2 of figure 10.20, the fact is they are not. When examined closely, the white blocks are found to be slightly larger by about 2 to 3 percent. The cause of this seemingly strange result is simple: the stain detection algorithm is a block-based algorithm and hence for a stain block that is only partially stained, i.e., not all pixels contained within that block are stained, the entire block will still be detected as a stained block. Obviously, the overall detected stain areas will eventually be slightly larger than what it should be. Such partially-stained blocks typically can be found near the fringes of a stained area.

Fig. 10.21 Detected stain blocks of frame 2

When the proposed stain removal algorithm as described in section 10.3 is applied to the stained frame 2, the stain blocks are successfully removed. The restored image is given in figure 10.22. As can be seen, frame 2 is perfectly restored without introducing any

undesirable artifacts. In terms of PSNR improvement, the PSNR of frame 2 is found to have improved from 32.02dB to 36.80dB after correction, i.e., a total of 4.78 dB.

Fig. 10.22 Frame 2 with stain blocks removed

Figure 10.23 shows the PSNR curves of the Wedding sequence before and after stain removal. It can be seen that the PSNR values for the restored Wedding sequence are consistently larger than those of the stained Wedding sequence. Overall, an average improvement of 2.12 dB is obtained.

Fig. 10.23 PSNR curves of the Wedding sequence

10.5 Conclusion

In this chapter, a novel model was successfully developed in the representation of stain in color motion pictures. A correction algorithm based on this model was then developed and tested on both natural and synthetic test sequences. Visual checking showed that the stains in the two natural stain sequences “Sword-fighting” and “Shock” were practically all removed after the correction process. Despite being effective, the proposed stain correction method depends heavily on two factors in its correction results: the availability of a good-quality stain-free (or stain-corrected) reference frame, and the selection of a good threshold window for correction. The latter is currently dealt with through much human interaction in the form of trial and error, and this is certainly one aspect that needs to be improved in future. Also a proper framework for the evaluation of stain correction is currently lacking. While it is possible to have a quantitative evaluation on synthetic test sequences as illustrated in section 10.4.2, the same cannot be said for natural test sequences. The current method of visual evaluation on natural test sequences is limited in the sense that it is very subjective in nature; a more quantitative measure is certainly needed to serve as an objective evaluation.

Chapter 11

Conclusion and Future Work

11.1 Conclusion

This thesis has presented original work on the correction/removal of two types of artifacts commonly found in degraded motion pictures: flicker and stain. Flicker correction is a research topic that is not so well studied, and to date only limited research work has been reported. On the other hand, stain removal appears to be an even less-studied topic, with apparently no reported work done whatsoever. The work in this thesis sets out to explore these two not so well-studied areas in image processing; it is to our satisfaction that we are able to reap a decent harvest at the concluding phase of our work. The work in this thesis has produced a number of new methods and algorithms for flicker correction, and possibly more importantly, it has not only pioneered the research work in stain removal, but also successfully delivered a working solution.

The contributions of this thesis are summarized below.

Our earliest work began with a thorough literature survey under the general topic of restoration of degraded motion pictures, and soon it became apparent that there are two areas where relatively little research work has been done. The first is flicker, which has substantially less work done on it compared to other artifacts and also all the reported work did not touch on color flicker at all. The next is stain; despite being quite a common defect in old color motion pictures, there is a glaring absence of any research work reported. It became quite clear to us that the direction of our research effort should be geared towards these two areas.

Our first attempt on flicker correction was done in chapter 3 on global flicker. Two heuristic methods, namely Histogram Mapping method and Intensity Mean Averaging method, were introduced and found to be effective in correcting global flicker. However, the same cannot be said when these methods were tested on local flicker.

To address this problem, Local Enhancement was incorporated into these two methods in chapter 4 by carrying out flicker correction on a block-based basis instead of the original frame-based basis. For the Histogram Mapping method, the incorporation of Local Enhancement was found to be able to correct local flicker; however, it also at the same time introduced two types of undesirable artifacts: blocky effect and “grayish speckles”. While the first artifact can be easily removed using smaller-sized blocks, the latter turned out to be a problem that cannot be solved. In contrast, the incorporation of Local Enhancement into the Intensity Mean Averaging method was a far smoother operation. Both global and local flicker were successfully removed and at the same time no artifacts were introduced. However, one problem did crop up -- distortion in the form of “smudged” checker box was found to appear in regions that contain motion.

As no solution to the problem of “grayish speckles” artifact could be found, it was decided that Histogram Mapping was not worth pursuing any further. Since the “smudged” checker box distortion had its root cause in motion regions, incorporation of motion compensation should provide a solution to this problem. For motion compensation to work properly, it is paramount that motion blocks are detected accurately. To this end, a block-matching-based motion-block detection algorithm capable of functioning even in a flicker environment was specially developed to do the job. This special algorithm is one of the highlights of the work in chapter 5 and was used widely in our later work in this thesis. With a working motion-block detection program, motion compensation was successfully incorporated, and all “smudged” checker boxes were eliminated. The motion-compensated Intensity Mean Averaging with Local Enhancement method thus emerged as a very effective flicker correction method – one that works, able to handle motion regions, and does not introduce any artifacts. While this is true, unfortunately there are two deficiencies that render this method unsuitable as a practical flicker correction solution. The first is the substantial human intervention that is needed to identify flicker frames before correction, which goes against our primary objective of restoration automation. The second is the need to define an intensity threshold for the

identification of flicker frames; currently this window is selected through a trial-and-error process which again requires heavy human intervention.

Since restoration automation is our primary objective, any flicker correction algorithm that deviates from this will be deemed as unacceptable. For this reason, all Intensity Mean Averaging based methods have to be abandoned unless ways can be found to automate the process. It was decided at this point that a new direction was needed in our next phase of work. After some detailed studies, we discovered that the earlier flicker-correction work by P.M.B. Roosmalen, who had proposed a highly automated algorithm (known as the FPE method), still had many problems remained unresolved. We felt that we might have answers to these problems and for this reason we decided to explore along this direction.

Our initial work in chapter 6 was on developing a new flicker correction algorithm (known as LSM method) by minimizing two “flicker-related parameters” using least square minimization. The LSM method was found to be effective in flicker correction; although it is about the same as the FPE method in terms of computational complexity, the LSM method nonetheless does offer itself as an effective alternative algorithm to the FPE method. The well-known short-coming of the FPE method was tackled next, which was its inability to identify motion blocks accurately. Our solution was to incorporate stationary block re-examination into the FPE method to detect motion blocks more accurately, and this had brought marginal improvement to the correction results.

The quest for a better flicker correction algorithm continued in chapter 7, where a far better solution was found to address the short-coming of the FPE method. The new algorithm solved the problem by incorporating motion compensation, whereby the set of motion vectors of each motion block is determined and incorporated into the equations used for flicker correction. The improvement in flicker correction was substantial. In fact of all the flicker correction algorithms developed in this thesis, the motion-compensated FPE method produced the best results in terms of both the intensity mean curve (the smoothest) and quality of restored frames.

The motion-compensated FPE method developed in chapter 7 was found to be able to deliver good correction results provided the set of motion vectors for each motion block can be determined very accurately. This cannot always be guaranteed as the motion

vector detection algorithm depends on three user-set thresholds θ , λ and γ (chapter 7 section 7.2.2.1), and choosing an appropriate value for each of these three thresholds is no easy task. Typically much trial-and-error is needed on the part of the user before an optimal set of threshold values can be obtained. The user may need to twiddle or even re-select the threshold values when moving onto a new section of a given frame sequence or a new sequence. The motion-compensated FPE method also suffers from another problem which is the difficulty in accurately determining the pair of flicker parameters $\alpha(x,y,t)$ and $\beta(x,y,t)$ in an area of uniform intensity (chapter 8 section 8.2), and such errors will lead to poor correction results. Further more, there are other problems such as scene change, occlusion, motion smear and non-translational motion for which the current method offers no answers. Surely, all these problems when put together are big enough to warrant some work to be done on them; the work in chapter 8 is our attempt to provide a solution to these problems.

A simplified motion-compensated FPE algorithm was proposed in chapter 8, whereby only the flicker gain term $\alpha(x,y,t)$ was used in the flicker model. The removal of the flicker offset term $\beta(x,y,t)$ has resulted in a very simple expression for the block-based flicker gain term $\alpha_{m,n}(t)$ (chapter 8 section 8.4.1 equation 8.6) that can be used later for flicker correction. This expression of $\alpha_{m,n}(t)$ also facilitates easy detection of the occurrence of scene change, occlusion, motion smear and non-translational motion as such events will lead to the value of $\alpha_{m,n}(t)$ of the affected block deviating significantly from 1. By checking $\alpha_{m,n}(t)$ against an appropriately chosen window (α_{min} , α_{max}), the occurrence of such events can be readily captured. The $\alpha_{m,n}(t)$ values of the affected blocks are deemed unreliable and their correct values are to be obtained through an interpolation process as described in chapter 5 section 5.4.2.1. Finally, $\alpha_{m,n}(t)$ of all blocks will undergo a median-filtering process to remove any spike before converting back to a pixel-based value used for flicker correction. The simplified motion-compensated FPE algorithm is found to be effective in flicker correction, but its performance is slightly below that of the FPE method, which uses a full flicker model that contains both $\alpha(x,y,t)$, the gain term, and $\beta(x,y,t)$, the off-set term. The drop in performance can be seen as the price paid for using a simpler model, but it has gained in terms of reduced computational complexity as well as being able to handle problems like scene change, motion smear and non-translational motion etc., for which there were no answers before. One possible way to improve the correction result is to apply Intensity

Mean Averaging developed in chapter 3 on the corrected sequence. However, it was found to work only under certain conditions (chapter 8 section 8.5.3).

Right at the beginning of our research work, it was noted that there was no reported flicker correction work on color motion pictures. Having successfully developed a number of new flicker correction algorithms in the earlier chapters, we felt that the time is now right to extend our work into the color space. Based on the flicker model used in chapter 8, a color flicker correction algorithm was successfully developed in chapter 9. Essentially, the corrupted color sequence is first split into its R, G and B channels, each channel is then flicker corrected using the algorithm developed in chapter 8, and finally the individually-corrected R, G and B channels are combined together to produce the flicker-corrected color sequence. Note that motion vectors used for flicker correction need to be determined from only one channel as they are the same for either the R, G or B channel. The success of our color flicker correction algorithm is another major achievement of the work done in this thesis; it is believed to be the very first reported work in color flicker correction.

Stain correction in color motion pictures was another area found totally neglected in the literature survey carried out at the beginning of our research work. Because stain is another intensity-related defect, it turns out that the simplified flicker model used in chapter 8 can also be used to describe stain adequately; i.e., $Y(x,y,t) = \alpha(x,y,t) I(x,y,t)$, where Y is the observed intensity, I the original (uncorrupted) intensity, α the stain factor, (x,y) the pixel location, and t the frame number. The equation is applicable for either the R, G or B channel. An expression for the block-based stain factor can be derived which is given by (meaning of symbols and subscripts are given in chapter 10 section 10.3.2):

$$\alpha_{[RGB]_{m,n}}(t) \approx \frac{E[Y(x,y,t)]}{E[\hat{I}(x - V_H, y - V_V, t - 1)]}$$

The proposed stain correction method essentially works in this manner. The stain-corrupted sequence is first split into its R, G and B channels. Using the equation above, the block-based stain factor of all blocks in each channel can be determined. A stain-damaged block is known to have a stain factor value deviating significantly from 1, and so by checking the stain factor value against an appropriately chosen window ($\alpha_{\min}$, $\alpha_{\max}$), we can easily determine if a block is stain damaged. With all stain-damaged blocks in

each channel detected, they can be readily corrected using equation 10.3 in section 10.3.3 of chapter 10. The final stain-corrected color frame can be obtained by simply combining the corresponding frame in the stain-corrected R, G and B channels.

Our stain correction algorithm is found to be effective and is another major contribution from our research work. The work is particularly significant in that it is the first known piece of work on stain correction, which is an important algorithm in any serious restoration system.

A comparison of all the flicker correction algorithms developed in this thesis in terms of the variance of the intensity mean curves of the corrected sequence are available in Appendix D.

All the motion picture sequences reported in this thesis, which include the natural flicker, synthetic flicker (original plus flicker-added) and flicker-corrected sequences, as well as the natural stain, synthetic stain (original plus stain-added) and stain-corrected sequences are available for viewing at the following URL: <http://kkweb.ntu.edu.sg>

For information on how these image files are organized, please refer to Appendix E.

11.2 Recommendation for future work

As we look back now over the work done in this thesis, we can see that there are many problems that we have solved, but there remain also many problems that are yet to be solved, or for which better solutions are to be found. It is believed that further investigation can be made in the following:

The problem of “grayish speckles” artifacts introduced during the incorporation of local enhancement into the Histogram Mapping method in chapter 4. A solution to this problem will give a new lease of life to the Histogram Mapping method, which is known to be very competent in correcting global flicker.

For the Intensity Mean Averaging related methods in chapters 4 and 5, the selection of the intensity threshold to decide if a frame is flickered and also the detection of flicker frames in a given flicker-corrupted sequence are currently all manually done. Ways should be studied to automate these two processes.

The FPE method in chapter 6 is hampered by its inability to detect motion regions properly; although incorporating stationary block re-examination could improve the results, the improvement was only marginal. Some fresh thinking may be needed for a better solution.

The motion-compensated FPE method in chapter 7 has produced the best flicker correction results of all methods presented in this thesis, but it is heavily dependent on the accurate detection of motion regions, which in turn is dependent on proper setting of three thresholds θ , λ and γ . Ways should be found to replace the current tedious trial-and-error manual process in setting these thresholds by an automatic process. An entirely new motion region detection algorithm is another possible alternative.

More in-depth study should be made into the simplified motion-compensated FPE method proposed in chapter 8 so as to have a better understanding of the strengths and limitations of the simplified flicker model. Currently the method was found to be working, although less competently, when compared to the FPE method which uses a full flicker model. Although regions affected by scene change, occlusion, image smear and non-translational motion can be detected by checking the block-based flicker gain term $\alpha_{m,n}(t)$ against a window $(\alpha_{\min}, \alpha_{\max})$, no means are available to differentiate these events. It is felt that if each of these can be differentiated, then a more refined and reliable correction process can probably be formulated. For example, if scene change can be distinctly detected, then the user can be alerted accordingly. The frame with the new scene should not be corrected based on the previous corrected frame containing a different scene as this violates the very principle on which film restoration is built. Instead ways must be found to correct this frame with the new scene, and later this frame is used as a reference frame for correcting the subsequent frame and so on until another scene change occurs. Also the selection of window $(\alpha_{\min}, \alpha_{\max})$ is currently a manual process which requires much trial and error on the part of the user; obviously automation of this process should be looked into seriously. The comments made here on chapter 8 are also applicable to chapter 9 as the basic algorithm used is the same in these two chapters.

For the stain correction algorithm in chapter 10, the selection of the window $(\alpha_{\min}, \alpha_{\max})$ is also a tedious manual process, and again study should be made to automate this process.

More work can also be carried out in the area of testing. Objective tests carried out in this thesis have been focused on checking the smoothness of intensity mean curves and PSNR evaluation, more thorough test can be carried out by incorporating coding-efficiency tests. Also, subjective evaluation for testing perceived quality of corrected image sequences should be performed to make the whole testing process more complete.

What have just been covered above are the more obvious and immediate areas in which further work can be carried out. However, it is felt that work in the following areas should also be considered.

Restoration of heavily flickered sequence still remains a formidable challenge with many problems unresolved. For any flicker correction algorithm that involves motion detection to work in such an extreme condition, effective motion detection algorithms that can operate in the presence of heavy flicker are needed. Current motion detection algorithms are known to be less than adequate in this aspect.

Flicker correction algorithms and stain removal algorithms (in fact all motion picture restoration algorithms) are by nature computationally intensive because of the huge size of image data that have to be processed. A distributed computing environment employing parallel processing would help share the heavy number crunching application among the networked clients and thus increase the overall throughput. There is much room to be explored in areas such as parallel algorithm design, system architecture for distributed processing, tasks scheduling and management etc.

We believe that the goal of creating an ultimate fully automated restoration system is one that is realizable. Ideally, this ultimate system will simply take in the corrupted sequence and without any specification from the user, produce the corrected sequence. We do not foresee this will happen soon, but with the ongoing effort put in by researchers working in this area, we should be getting closer, and eventually reach there one day.

Epilogue

A short poem that captures my mood at this concluding stage will be fitting as an epilogue for this thesis. It is originally written in Chinese, and its English translation is given below.

少 有 凌 云 志

登 峰 在 此 时

极 目 楚 天 远

鹰 击 长 空 阔

*"I had a dream to reach for the sky since young,
And now I am at the peak of this high mountain;
Standing here, my eyes could reach the far horizon of the northern sky;
Flapping my wings, like an eagle I am ready to soar high..."*

References

- [1] J.H. Billington, "Film Preservation 1993: A Study of the Current State of American Film Preservation, 1993.
- [2] B. Fisher, "Digital Restoration of Snow White: 120,000 Famous Frames are Back", Advanced Imaging, pp. 32-36, September 1993.
- [3] E. Dubios and S. Sabri, " Noise Reduction in Image Sequences using Motion Compensated Temporal Filtering", IEEE Trans. on Communications, No. 32, pp. 826-831, 1984.
- [4] G.R. Arce, "Multistage Order Statistic Filters for Image Sequence Processing", IEEE Trans. on Signal Processing, Vol. 39, No. 5, pp. 1146-1163, 1991.
- [5] M.K. Ozkan, A.T. Erdem, M.I. Sezan, and A.M. Tekalp, "Efficient Multiframe Wiener Restoration of Blurred and Noisy Image Sequences", IEEE Trans. on Image Processing, Vol1, No. 4, pp. 453-476, 1992.
- [6] M.K. Ozkan, A.T. Erdem, M.I. Sezan, and A.M. Tekalp, "Adaptive Motion Compensated Filtering of Noisy Image Sequences", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 3, No. 4, 1993.
- [7] J.C. Brailean, R.P. Kleihorst, S.N. Efstratiadis, A.K. Katsaggelos, and R.L. Lagendijk, "Noise Reduction Filters for Dynamic Image Sequences: A Review", Proc. of the IEEE, Vol. 83, No. 9, pp. 1272-1292, 1995.

- [8] E. Abreu, M. Lightstone, S.K. Mitra, and K. Arakawa, "A New Efficient Approach for the Removal of Impulse Noise from Highly Corrupted Images", IEEE Trans. on Image Processing, Vol. 5, No. 6, pp. 1012-1025, 1996.
- [9] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, and P.J.W. Rayner, "Detection of Missing Data in Image Sequences", IEEE Trans. on Image Processing, pp. 1496-1508, Vol. 4, No. 11, 1995.
- [10] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, and P.J.W. Rayner, "Interpolation of Missing Data in Image Sequences", IEEE Trans. on Image Processing, pp. 1509-1519, Vol. 4, No. 11, 1995.
- [11] W.B. Goh, M.N. Chong, S. Kalra and D. Krishnan, "Bi-Directional 3D Auto-Regressive Model Approach to Motion Picture Restoration", Proc. of ICASSP 96, pp. 2277-2280, 1996.
- [12] H. Muller-Seelich, W. Plaschzug, P.Schallauer, S.Potzman, and W. Hass, "Digital Restoration of 35 mm Film", Proc. of ECMAST 96, Vol. I, pp. 255-265, Louvain-la-Neuve, Belgium, 1996.
- [13] S. Kalra, M.N. Chong, and D. Krishnan, "A New Auto-Regressive (AR) model-based Algorithm for Motion Picture Restoration", Proc. of ICASSP 97, Vol. 4, pp. 2557-2560, Munich, Germany, 1997.
- [14] M.J. Nadenau and S.K. Mitra, "Blotch and Scratch Detection in Image Sequences based on Rank Ordered Differences", in Time-Varying Image Processing and Moving Object Recognition, V. Cappelini ed., Elsevier, 1997.
- [15] P.M.B. van Roosmalen, R.L. Lagendijk, and J.Biemon, "Improved Blotch Detection by Postprocessing", Proc. of IEEE Signal Processing Symposium SPS'98, pp. 223-226, Leuven, Belgium, 1998.
- [16] R.D. Morris, W.J. Fitzgerald, and A.C. Kokaram, "A Sampling Based Approach to Line Scratch Removal from Motion Picture Frames", Proc. of ICIP-96, Vol. I, pp. 801-804, Lausanne, Switzerland, IEEE, 1996.

- [17] A.C. Kokaram, "Motion Picture Restoration", Springer Verlag, 1998.
- [18] T. Vlachos and G. Thomas, "Motion Estimation for the Correction of Twin-Lens Telecine Flicker", Proc. of ICIP-96, Vol. I, pp. 109-112, Lausanne, Switzerland, IEEE, 1996.
- [19] P. Richardson and D. Suter, "Restoration of Historic Film for Digital Compression: A Case Study", Proc. of ICIP-95, Vol. II, pp. 49-52, Washington D.C. USA, IEEE, 1995.
- [20] D. Ferrandiere, "Motion Picture Restoration using Morphological Tools", International Symposium on Mathematical Morphology (ISMM), pp. 361-368, Kluwer Academic Press, 1996.
- [21] P.M.B. van Roosmalen, R.L. Lagendijk, and J.Biemon, "Correction of Intensity Flicker in Old Film Sequences", IEEE Trans. on Circuits and Systems, 1999.
- [22] V. Naranjo and A. Alboil, "Flicker Reduction in Old Films", Proc. of the 2000 International Conf. on Image Processing (ICIP-2000), volume 2, pp. 657-659, Vancouver BC, Canada, 2000.
- [23] T. Ohuchi, T.Seto, T.Komatsu, and T. Saito, "A Robust Method of Image Flicker Correction for Heavily-Corrupted Old Film Sequences", Proc. of the 2000 International Conference on Image Processing (ICIP-2000), 2000.
- [24] X. Yang, M.N. Chong, and K.K. Wong, "A Stable Flicker Removal Technique for Film Archives", Proc. of Image Conference 2000, Leicester, U.K. 2000.
- [25] P.M.B. van Roosmalen, R.L. Lagendijk and J.Biemon, "Flicker Reduction in Old Film Sequences", in Time-Varying Image Processing and Moving Object Recognition, 4, V.Cappellini, ed, Elsevier Science, pp. 11-18, 1997.
- [26] K.K. Wong, A. Das and M.N. Chong, "Improved Flicker Removal through Motion Vectors Compensation", 3rd International Conference on Image and Graphics (ICIG 2004), Hong Kong, 2004.

- [27] R. Krishnamurthy, P. Moulin, and J.W. Woods, "Multiscale motion for scalable video coding", in Proc. IEEE Int. Conf. Image Processing, Lausanne, Switzerland, Sept. 1996, pp. 965-968.
- [28] C. Cafforio, F. Rocca, and S. Tubaro, "Motion Compensated Image Interpolation", IEEE Trans. Commun., vol.32, pp. 215-222, Feb. 1990.
- [29] ITU-R Rec. BT.500-11, "Methodology for the Subjective Assessment of Quality of Television Pictures", ITU, 2002.
- [30] J. Allnat, "Transmitted-picture Assessment", John Wiley & Sons, 1983.
- [31] R.C. Gonzalez and R.E. Woods, "Digital Image Processing", Addison Wesley, 1993.
- [32] A.C. Kokaram, R. Dahyoy, F. Pitie and H. Denman, "Simultaneous Luminance and Position Stabilization for Film and Video", Visual Communications and Image Processing, San Jose, California USA, January 2003.
- [33] F. Pitie, R. Dahyot, F. Kelly and A. Kokaram, "A New Robust Technique for Stabilizing Brightness Fluctuations in Image Sequences" "ECCV Workshop SMVP (Statistical Methods in Video Processing), pp. 153-164, Prague, Czech Republic, 2004.
- [34] T. Vlachos, "Flicker Correction for Archived Film Sequences Using a Nonlinear Model", IEEE Transactions on Circuits and Systems for Video Technology, Vol.14, No. 4, April 2004
- [35] K.K. Wong, A. Das and M.N. Chong, "An Improved Flicker Removal Method using Stationary Block Re-examination", WSEA, Athens, Dec. 2003.
- [36] K.K. Wong, A. Das and M.N. Chong, "Effective Flicker Removal through Repetitive Corrections", WSEA, Athens, Dec. 2003.
- [37] J.M. Boyce, "Noise Reduction of Image sequences Using Adaptive Motion Compensated Frame Averaging", IEEE ICASSP, vol. 3, pp. 461- 464, 1992.

- [38] A.K. Jain, "Fundamentals of Digital Image Processing", Prentice Hall, 1989.
- [39] Jae S. Lim, "Two-Dimensional Signal and Image Processing", Prentice Hall, 1990.
- [40] I. Pitas and A.N. Venetsanopoulos, "Nonlinear Digital Filters Principles and Applications", Kulwer Academic Publishers, 1990.
- [41] A.C. Kokaram, "Motion Picture Restoration", Phd.D Thesis, department of Engineering, University of Cambridge, Churchill College, U.K., May 1993.
- [42] A.M. Tekalp, "Digital Video Processing", Prentice Hall, 1995.
- [43] J. Robbins and A. Netravali, "Image sequence processing and dynamic scene analysis, chapter recursive motion compensation: a review", pp. 76-103, Springer-Verlag, 1983.
- [44] H. Nagel, "Spatio-temporal modeling based on image sequences", Int. symposium on image processing and its applications, pp. 222-252, 1984.
- [45] D.M. Martinez, "Model-based motion estimation and its application to restoration and interpolation of motion pictures", PhD thesis, Massachusetts Institute of Technology, 1986.
- [46] J.Riveros and K. Jabbour, "Review of motion analysis techniques", IEE Proceedings, 136:397-404, December, 1989.
- [47] I. Abdelqader and S. Rjala, "Energy minimization approach to motion estimation", Signal Processing, 28:291-309, 1992.
- [48] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images", IEEE PAMI, 6:721-741, 1984.
- [49] M. Ghanbari, "The cross-search algorithm for motion estimation", IEEE Transactions on Communications, 38:950-953, July 1990.

- [50] A. Zaccarin and B. Liu, "Fast algorithms for block motion estimation", IEEE ICASSP, volume 3, pp. 449-452, 1992.
- [51] W.B. Goh, M.N. Chong, S. Kalra and D. Krishnan, "A bi-directional 3D AR model approach to motion picture restoration", IEEE Int. conf. on acoustics, speech & signal processing, pp. 2277-2280, May 1996.
- [52] M. Bierling, "Displacement estimation by hierarchical block-matching", Proceedings of visual communication and image processing, SPIE vol. 1001, pp. 942-951, 1988.
- [53] J.S. Lim, "Two Dimensional Signal Processing and Image Processing", Prentice Hall, 1990.
- [54] Y. Wu and D. Suter, "Historical film processing", Proc. SPIE, volume 2564, pp. 289-300, San Diego, USA, 1995.
- [55] A.C. Kokaram, R.Dahyot, F. Pitie, and H. Denman, "Simultaneous luminance and position stabilization for film and video", Proc. SPIE, Visual Communications and Image Processing, San Jose, USA, 2003.
- [56] P. Schallauer, A. Pinz, and W. Hass, "Automatic restoration algorithms for 35mm film", Videre, 1(3):60-85, 1999.
- [57] J. Jung, M. Antonini, and M. Barlaud, "Automatic restoration of old movies with an objected oriented approach", Proc. RFIA (Conference on Pattern Recognition and Artificial Intelligence), Paris, France, 2000.
- [58] A. Rizzi, C. Gatta, and D. Marini, "Color correction between gray world and white patch", Electronic Imaging 2002, 20-25/01/02, San Jose, California, USA.
- [59] M. Chambah, B.Besserer, and Courtellemont, "Latest results in digital color film restoration", Machine Graphics & Vision International Journal, vol. 11, pp. 363-395, 2002.
- [60] Invention: Triple standard DVD, New Scientist, 18 Sept. 2006.

- [61] P. Burrows, "Next: TV meets IP", Business Week, 23 Nov. 2004.
- [62] T. Hall, A. Darling and M. Fiddy, "Image compression and restoration incorporating prior knowledge", Optics Letters, Volume 7, Issue 10, October 1982, pp.467-468.
- [63] S. Skoneczny, "Defects removing in old movies", Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis, June 2001, pp. 38-42.
- [64] J. Chenot, J. Drewery and D. Lyon, "Restoration of archived television programs for digital broadcasting", International Broadcasting Convention 1998, Amsterdam, The Netherlands, IBC'98 Proceedings, 11-15 Sept 1998.
- [65] J. Chenot, "Restoration tools for the future: the Aurora project", FIAT-IFTA Annual Conference 1996, Paris, France, FIAT-IFTA Annual Conference 96 proceedings Sept. 1996.
- [66] A. Kokaram, "Detection and removal of line scratches in degraded motion picture sequences", IEEE International SPIE European Conference on Signal Processing VIII (EUSIPCO)Trieste Italy, EUSIPCO Vol. 1, pp. 5-8, September 1996.
- [67] P.M.B. Roosmalen, "Improved Blotch Detection by post-processing" Proceedings of IEEE Benelux Signal Processing Chapter, Leuven, Belgium, March 1998.
- [68] A. Kokaram, "Advances in the Detection and Reconstruction of Blotches in Archived Film and Video", Proceedings of the IEE seminar "Digital Restoration of Film and Video Archives", Savoy Place London, 16 January 2001, UK ISSN 0963-3308-REFERENCE NO: 2000/110.
- [69] P. Chamberlain, "Digital film restoration – re-mastering workflow", Da Vinci publications, 2007.
- [70] S. Sommerville, "Archangel: automated real time archive restoration", IEE Seminar on Digital restoration of film and video archives, January 2001.

[71] BRAVA Web site, URL:

www.ina.fr/recherche/projets/finis/brava/brava_summary.en.html

[72] Kallenberger, Richard H., Cvjetnicinanin, and George D. (1994). "Film into Video: A Guide to Merging the Technologie",. Focal Press, 1994.

[73] "Digital Cinema: A slow revolution", IEE Review, UK, November 2003.

[74] T. Takeuchi, K.K De Valois and J.L. Hardy, "The influence of color flicker on the detection of luminance patterns", Vol. 1, Number 3, Journal of Vision, Dec 2001.

[75] "Videos: NTSC vs PAL format", SPIE (The International Society for Optical Engineering) web site, URL: <http://www.spie.org/app/education>.

[76] H. Wilhelm, "Monitoring the fading and staining of color photographic prints", Journal of the American Institute of Conservation, Vol. 21, number 1, article 3, pp. 49-64.

[77] Mpeg.org website, URL: <http://www.mpeg.org/MPEG/starting-points.html>

[78] C. Mead and L. Conway, "Introduction to VLSI systems", Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1979.

[79] R.D. Morris and W.J. Fitzgerald, "Detection and correction of speckle degradation in image sequences using a 3D Markov random field," Proceedings International Conference on Image Processing : Theory and Applications (IPTA '93), Elsevier, 1993.

[80] L. Tenze, G. Ramponi, and S. Carrato, "Blotches correction and contrast enhancement for old film pictures", Proceedings of International Conference on Image Processing, ICIP, Canada, pp. 660-663, 2000.

[81] T. Komatsu, T. Ohuchi, and T. Saitio, "Detection and restoration of film blotches using global motion Segmentation", Proceedings of the 1999 International Conference on Image Processing (ICIP '99), Kobe, Japan, October 24-28, 1999.

- [82] L. Joyeux, S. Boukir, and B. Besserer, "Film Line Scratch Removal Using Kalman Filtering and Bayesian Restoration," 5th IEEE Workshop on Applications of Computer Vision, Dec. 4-6, 2000, pp. 8-13.
- [83] N.D. Kim, S. Udpa, "Nonlinear Operators for Edge Detection and Line Scratch removal", Proceedings of IEEE International Conference on SMC, pp. 4401-4404, 1998.
- [84] G. Laccetti, L. Maddalena, and A. Petrosino, "P-LSR: A parallel algorithm for line scratch restoration", Seventh International Workshop on Computer Architectures for Machine Perception (CAMP 2005), 4-6 July 2005, Palermo, Italy.
- [85] D. Tegolo and F. Isgro, "A genetic algorithm for scratch removal in static images", 11th International Conference on Image Analysis and Processing (ICIAP 2001), 26-28 September 2001, Palermo, Italy.
- [86] P.J. Brightwell, R.E. May, and G.A. Thomas, "Automated correction of film unsteadiness, dirt and grain", International Broadcasting Convention – IBC'94, Amsterdam, The Netherlands, pp. 667-672, 16-20 Sept. 1994.
- [87] P.M.B. Roosmalen, J. Biemond, and R.L. Lagendijk, "Restoration and storage of film and video archive material", Journal of Signal Processing and Multimedia, 1(3), 167-191 (1999).
- [88] A.C. Kokaram, "Detection and removal of line scratches in degraded motion picture sequences", IEEE International SPIE European Conference on Signal Processing VIII (EUSIPCO), Trieste Italy, EUSIPCO Vol. 1, pp. 5-8, September 1996.
- [89] J. Biemond, P.M.B. Roosmalen, and R.L. Lagendijk "Improved blotch detection by post-processing", Proc. of IEEE Benelux Signal Processing Chapter, Leuven, Belgium, March 1998.
- [90] T. Vlachos, "Improving the efficiency of MPEG-2 coding by means of film unsteadiness correction", SPIE, Digital Compression Technologies and Systems for Video Communications SPIE Proceedings, Vol. 2952, pp. 534-542, Oct. 1996.

- [91] C. Gatta, A. Rizzi, D. Marini, "ACE: an automatic color equalization algorithm", CGIV02, The first European Conference on Color in Graphics Image and Vision, 2-5/4/2002, Poitiers, France.
- [92] G. Forbin, T. Vlachos and S. Tredwell, "Spatially-adaptive non-linear correction of flicker using grey level tracing", Proceedings of the 7th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking & Broadcasting (PGNET 2006), Liverpool, June 2006.
- [93] G. Forbin, T. Vlachos and S. Tredwell, "Flicker Compensation for Archived Film Using A Spatially-Adaptive Nonlinear Model", International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006), Toulouse, May 2006.
- [94] G. Forbin, T. Vlachos and S. Tredwell, "Spatially-Adaptive Flicker Compensation for Archived Film Sequences using a Nonlinear Model", 2nd IEEE European Conference on Visual Media Production (CVMP 2005), London, November 2005.
- [95] G. Forbin, B. Besserer, J. Boldy, and D. Tschumper, "Temporal Extension to Exemplar-based Inpainting Applied to Scratch Correction in Damaged Image Sequences", Visualization, Imaging, and Image Processing (VIIP 2005), Benidorm, September 2005.
- [96] A. Rizzi, M. Chambah, D. Lenza, B. Besserer, and D. Marini, "Tuning of perceptual technique for digital movie color restoration", Visual Communications and Image Processing 2004, Proceedings of the SPIE, Volume 5308, pp. 1286-1294 (2004).
- [97] Y. Ando, A. Hansuebsai, and K. Khantong, "Digital restoration of faded color images by subjective method", The Journal of imaging science and technology.
- [98] M. Chambah, A. Rizzi, C. Gatta, B. Besserer, and D. Marini, "Perceptual approach for unsupervised digital restoration of cinematographic archives".
- [99] M. Chambah, and B. Besserer, "Digital color restoration of faded motion pictures", Computer graphics and image processing conference CGIP2000, Saint-Etienne, France, 2000, pp. 338-342.

- [100] M. Chambah, and B. Besserer, "Digital restoration of faded color movies: a four-step method", 8th Color imaging conference CIC8, IS&T/SID, Scottsdale, AZ, USA, 2000, pp. 161-166.
- [101] M. Chambah, B. Besserer, and P Courtellemont, "Recent progress in automatic digital restoration of color motion pictures", SPIE Electronics Imaging 002, San Jose, CA, USA, Janvier 2002, vol. 4663, pp. 98-109.
- [102] M. Chambah, B. Besserer, and P Courtellemont, "Approach to automatic digital restoration of faded color films", IS&T Conference on Graphics, Imaging, and Vision CGIV 2002, Poitiers, France, avril 2002, pp. 613-618.
- [103] M. Chambah, B. Besserer, and P Courtellemont, "Latest results in digital color film restoration", Machine Graphics and Vision (MG&V) Journal, 2002.
- [104] S.Winkler, "Digital Video Quality, vision models and metrics", Genista Corporation, Montreux, Switzerland, 2005.
- [105] P.M.B. van Roosmalen, "Restoration of Archived Film and Video", PhD Thesis, 1999.

Appendix A

Least Square Minimization (LSM) method

A.1 Derivation of the $a_{m,n}(t)$ and $b_{m,n}(t)$ expressions

It was shown in chapter 6 section 6.1.2 that the estimated value of the original intensity $\hat{I}(x,y,t)$ can be expressed as:

$$\hat{I}(x,y,t) = \frac{[Y(x,y,t) - \beta_{m,n}(t)]}{\alpha_{m,n}(t)} = a_{m,n}(t)Y(x,y,t) + b_{m,n}(t) \quad (A.1)$$

where the parameters $a_{m,n}(t)$ and $b_{m,n}(t)$ respectively given by $1/\alpha_{m,n}(t)$ and $-\beta_{m,n}(t)/\alpha_{m,n}(t)$ are named here as the ‘inverse flicker parameters’.

The estimated intensity $\hat{I}(x,y,t)$ differs from the original intensity $I(x,y,t)$ by an error $\varepsilon(x,y,t)$ given by:

$$\varepsilon(x,y,t) = I(x,y,t) - \hat{I}(x,y,t) \quad (A.2)$$

It can be argued that the original intensity can be approximated by minimizing a function $f(a,b)$, defined as the sum of squared error components in a small region $\Omega_{m,n}$ containing N pixels, where $f(a,b)$ is given by:

$$f(a,b) = \sum_{(x,y) \in \Omega_{m,n}} \varepsilon^2(x,y,t) = \sum_{(x,y) \in \Omega_{m,n}} \{I(x,y,t) - [a_{m,n}(t)Y(x,y,t) + b_{m,n}(t)]\}^2 \quad (A.3)$$

Setting both $\frac{\partial f(a,b)}{\partial a}$ and $\frac{\partial f(a,b)}{\partial b}$ to zero for minimizing $f(a,b)$ will gives us two equations to solve for values of $a_{m,n}(t)$ and $b_{m,n}(t)$ that can be used for intensity correction, this is as illustrated below:

$$\begin{aligned}\frac{\partial f(a,b)}{\partial a} &= \frac{\partial}{\partial a} \left[\sum_{(x,y) \in \Omega_{m,n}} \{I(x,y,t) - [a_{m,n}(t)Y(x,y,t) + b_{m,n}(t)]\}^2 \right] \\ &= \sum_{(x,y) \in \Omega_{m,n}} 2\{I(x,y,t) - [a_{m,n}(t)Y(x,y,t) + b_{m,n}(t)]\}[-Y(x,y,t)]\end{aligned}\quad (A.4)$$

$$\begin{aligned}\frac{\partial f(a,b)}{\partial b} &= \frac{\partial}{\partial b} \left[\sum_{(x,y) \in \Omega_{m,n}} \{I(x,y,t) - [a_{m,n}(t)Y(x,y,t) + b_{m,n}(t)]\}^2 \right] \\ &= \sum_{(x,y) \in \Omega_{m,n}} 2\{I(x,y,t) - [a_{m,n}(t)Y(x,y,t) + b_{m,n}(t)]\}(-1)\end{aligned}\quad (A.5)$$

Now setting $\frac{\partial f(a,b)}{\partial a}$ to 0 gives:

$$\begin{aligned}\sum_{(x,y) \in \Omega_{m,n}} 2\{I(x,y,t) - [a_{m,n}(t)Y(x,y,t) + b_{m,n}(t)]\}[-Y(x,y,t)] &= 0 \\ \sum_{(x,y) \in \Omega_{m,n}} \{-I(x,y,t)Y(x,y,t) + [a_{m,n}(t)Y^2(x,y,t) + b_{m,n}(t)Y(x,y,t)]\} &= 0 \\ \sum_{(x,y) \in \Omega} I(x,y)Y(x,y) &= [\sum_{(x,y) \in \Omega} Y^2(x,y)]a(t) + [\sum Y(x,y)]b(t)\end{aligned}\quad (A.6)$$

Setting $\frac{\partial f(a,b)}{\partial b}$ to 0 gives:

$$\begin{aligned}\sum_{(x,y) \in \Omega_{m,n}} 2\{I(x,y,t) - [a_{m,n}(t)Y(x,y,t) + b_{m,n}(t)]\}(-1) &= 0 \\ \sum_{(x,y) \in \Omega_{m,n}} \{I(x,y,t) - [a_{m,n}(t)Y(x,y,t) + b_{m,n}(t)]\} &= 0 \\ \sum_{(x,y) \in \Omega_{m,n}} I(x,y,t) &= \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t)a_{m,n}(t) + Nb_{m,n}(t)\end{aligned}\quad (A.7)$$

Equations A.6 and A.7 can now be used together to solve for $a_{m,n}(t)$ and $b_{m,n}(t)$. From equation A.7:

$$b_{m,n}(t) = \frac{1}{N} \left\{ \sum_{(x,y) \in \Omega_{m,n}} I(x,y,t) - \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \right\} a_{m,n}(t) \quad (\text{A.8})$$

Substituting equation A.8 into equation A.6:

$$\sum_{(x,y) \in \Omega_{m,n}} I(x,y,t) Y(x,y,t) = a_{m,n}(t) \sum_{(x,y) \in \Omega_{m,n}} Y^2(x,y,t) + \frac{1}{N} \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \left\{ \sum_{(x,y) \in \Omega_{m,n}} I(x,y,t) - \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \right\} a_{m,n}(t)$$

Hence:

$$\begin{aligned} & N \sum_{(x,y) \in \Omega_{m,n}} I(x,y,t) Y(x,y,t) \\ &= N a_{m,n}(t) \sum_{(x,y) \in \Omega_{m,n}} Y^2(x,y,t) + \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \left\{ \sum_{(x,y) \in \Omega_{m,n}} I(x,y,t) - \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \right\} a_{m,n}(t) \\ &= N a_{m,n}(t) \sum_{(x,y) \in \Omega_{m,n}} Y^2(x,y,t) - a_{m,n}(t) \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) + \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \sum_{(x,y) \in \Omega_{m,n}} I(x,y,t) \\ &= a_{m,n}(t) \left[N \sum_{(x,y) \in \Omega_{m,n}} Y^2(x,y,t) - \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \right] + \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \sum_{(x,y) \in \Omega_{m,n}} I(x,y,t) \end{aligned}$$

An expression for $a_{m,n}(t)$ can now be obtained:

$$a_{m,n}(t) = \frac{N \sum_{(x,y) \in \Omega_{m,n}} I(x,y,t) Y(x,y,t) - \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \sum_{(x,y) \in \Omega_{m,n}} I(x,y,t)}{N \sum_{(x,y) \in \Omega_{m,n}} Y^2(x,y,t) - \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t)}$$

Multiplying top and bottom by $1/N$:

$$a_{m,n}(t) = \frac{\sum_{(x,y) \in \Omega_{m,n}} I(x,y,t)Y(x,y,t) - \frac{1}{N} \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \sum_{(x,y) \in \Omega_{m,n}} I(x,y,t)}{\sum_{(x,y) \in \Omega_{m,n}} Y^2(x,y,t) - \frac{1}{N} \left[\sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \right]^2} \quad (\text{A.9})$$

By definition of variance:

$$\begin{aligned} \sigma^2 Y(x,y,t) &= EY^2(x,y,t) - [EY(x,y,t)]^2 \\ &= EY^2(x,y,t) - \left[\frac{\sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t)}{N} \right]^2 \\ &= \frac{1}{N} \sum_{(x,y) \in \Omega_{m,n}} Y^2(x,y,t) - \frac{1}{N^2} \left[\sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \right]^2 \end{aligned}$$

Hence:

$$N\sigma^2 Y(x,y,t) = \sum_{(x,y) \in \Omega_{m,n}} Y^2(x,y,t) - \frac{1}{N} \left[\sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \right]^2 \quad (\text{A.10})$$

Substituting equation A.10 into equation A.9:

$$\begin{aligned} a_{m,n}(t) &= \frac{1}{N\sigma^2 Y(x,y,t)} \left[\sum_{(x,y) \in \Omega_{m,n}} I(x,y,t)Y(x,y,t) - \frac{1}{N} \sum_{(x,y) \in \Omega_{m,n}} Y(x,y,t) \sum_{(x,y) \in \Omega_{m,n}} I(x,y,t) \right] \\ &= \frac{1}{N\sigma^2 Y(x,y,t)} \left[\sum_{(x,y) \in \Omega_{m,n}} I(x,y,t)Y(x,y,t) - EY(x,y,t) \sum_{(x,y) \in \Omega_{m,n}} I(x,y,t) \right] \quad (\text{A.11}) \end{aligned}$$

By introducing a term $\sum_{(x,y) \in \Omega_{m,n}} EI(x,y,t)\{Y(x,y,t) - EY(x,y,t)\}$ which equals to 0 (see proof later) into A.11:

$$\begin{aligned}
 a_{m,n}(t) &= \\
 & \frac{1}{N\sigma^2 Y(x, y, t)} \left[\sum_{(x,y) \in \Omega_{m,n}} I(x, y, t) Y(x, y, t) - EY(x, y, t) \sum_{(x,y) \in \Omega_{m,n}} I(x, y, t) - \sum_{(x,y) \in \Omega_{m,n}} EI(x, y, t) \{Y(x, y, t) - EY(x, y, t)\} \right] \\
 \therefore \hat{a}_{m,n}(t) &= \frac{1}{N\sigma^2 Y(x, y, t)} \left[\sum_{(x,y) \in \Omega_{m,n}} \{I(x, y, t) - EI(x, y, t)\} \{Y(x, y, t) - EY(x, y, t)\} \right] \quad (A.12)
 \end{aligned}$$

From equation A.8:

$$\begin{aligned}
 b_{m,n}(t) &= \frac{1}{N} \left\{ \sum_{(x,y) \in \Omega_{m,n}} I(x, y, t) - \left[\sum_{(x,y) \in \Omega_{m,n}} Y(x, y, t) \right] a_{m,n}(t) \right\} \\
 &= EI(x, y, t) - \hat{a}_{m,n}(t) EY(x, y, t) \quad (A.13)
 \end{aligned}$$

Substituting equation A.12 into equation A.13:

$$\therefore \hat{b}_{m,n}(t) = EI(x, y, t) - \frac{EY(x, y, t)}{N\sigma^2 Y(x, y, t)} \left[\sum_{(x,y) \in \Omega_{m,n}} \{I(x, y, t) - EI(x, y, t)\} \{Y(x, y, t) - EY(x, y, t)\} \right] \quad (A.14)$$

A. 2 Proving $\sum_{(x,y) \in \Omega_{m,n}} EI(x, y, t) \{Y(x, y, t) - EY(x, y, t)\}$ equals to 0

$$\begin{aligned}
& \sum_{(x,y) \in \Omega_{m,n}} EI(x, y, t) \{Y(x, y, t) - EY(x, y, t)\} \\
&= \sum_{(x,y) \in \Omega_{m,n}} EI(x, y, t) Y(x, y, t) - \sum_{(x,y) \in \Omega_{m,n}} EI(x, y, t) EY(x, y, t) \\
&= EI(x, y, t) \sum_{(x,y) \in \Omega_{m,n}} Y(x, y, t) - EI(x, y, t) \sum_{(x,y) \in \Omega_{m,n}} EY(x, y, t) \\
&= EI(x, y, t) \sum_{(x,y) \in \Omega_{m,n}} Y(x, y, t) - EI(x, y, t) N[EY(x, y, t)] \\
&= EI(x, y, t) \sum_{(x,y) \in \Omega_{m,n}} Y(x, y, t) - EI(x, y, t) N\left[\frac{1}{N} \sum_{(x,y) \in \Omega_{m,n}} Y(x, y, t)\right] \\
&= EI(x, y, t) \sum_{(x,y) \in \Omega_{m,n}} Y(x, y, t) - EI(x, y, t) \sum_{(x,y) \in \Omega_{m,n}} Y(x, y, t) \\
&= 0
\end{aligned}$$

Appendix B

Proof of expressions used in Intensity Mean Averaging

To prove $\hat{I}_2(x, y) = I_2(x, y) - m_{gain}$

Proof:

Let us assume that there are n pixels in each frame.

If $\hat{I}_2(x, y) = I_2(x, y) - m_{gain}$ is true. Then:

$$\sum_{(x,y) \in f_2} \hat{I}_2(x, y) = \sum_{(x,y) \in f_2} I_2(x, y) - nm_{gain}$$

$$\text{i.e.} \quad \frac{1}{n} \sum_{(x,y) \in f_2} \hat{I}_2(x, y) = \frac{1}{n} \sum_{(x,y) \in f_2} I_2(x, y) - m_{gain}$$

$$\text{i.e.} \quad m_{avg} = m_2 - m_{gain}$$

Since $m_{avg} = m_2 - m_{gain}$ is true, it follows that $\hat{I}_2(x, y) = I_2(x, y) - m_{gain}$ must be true.

Appendix C

An Improved Low-pass filtering algorithm

A low-pass filtering algorithm that is computationally simpler and faster has been developed in this thesis. It is essentially a standard mean (average) filter but implemented in a special way to achieve a substantial speedup.

A square filter of 5x5 spatial mask will be used as a start in our illustration here. Figure 1(a) shows a 5x5 mask with weighting of one throughout and figure 1(b) is a 5x5 pixel block with pixel intensities given by $\{I_1, I_2, \dots, I_{25}\}$.

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Fig.1(a) A 5x5 filtering mask
with equal weightings of 1

I_1	I_2	I_3	I_4	I_5
I_6	I_7	I_8	I_9	I_{10}
I_{11}	I_{12}	I_{13}	I_{14}	I_{15}
I_{16}	I_{17}	I_{18}	I_{19}	I_{20}
I_{21}	I_{22}	I_{23}	I_{24}	I_{25}

Fig.1(b) A 5x5 pixels image block

The filtering process is carried out by moving the center of the mask to the pixel location for which the mean is to be computed. For example, when the mask in figure 1(a) is applied to the image block in figure 1(b), the mean pixel intensity I_m at I_{13} is given by:

$$I_m = (I_1 \times 1 + I_2 \times 1 + \dots + I_{25} \times 1) / 25 \quad (1)$$

Equation (1) shows that for a 5×5 filtering mask, the mean computation requires $5^2 - 1$ additions and 1 division. In general, for a $n \times n$ mask, it will require $n^2 - 1$ additions and 1 division, or a total of n^2 arithmetic operations. This suggests that the number of arithmetic operations grows rather rapidly as n increases; a computationally less intensive averaging method is thus very much desired.

A fast averaging method is proposed here which can significantly speed up the low-pass filtering process. The gain in speed is brought about by making use of computational results that were obtained earlier for other blocks. This is illustrated in the example given in figure 2.

Figure 2 shows a pixel P_{22} undergoing a 5×5 low-pass filtering process, which is essentially to compute the average value based on the pixels within the 5×5 filtering mask centered at pixel P_{22} . In a typical low-pass filtering process that works on all pixels in a given image frame, the direction of movement of the moving filter is from left to right, starting from the top left hand corner and ending at the bottom right hand corner, with movement limited to 1 pixel each time in either the horizontal or vertical direction. From the movement of the filtering mask, it can be seen that all the pixels under the current filtering mask have actually all been involved in the averaging of some earlier blocks before, with only one exception – the bottom right pixel under the current filtering mask.

Pixel P_{22} being processed by the 5×5 filtering mask centered at P_{22} .

Fig.2 An illustration of the proposed low-pass filtering algorithm

$$\begin{aligned} \text{Mean} = & \text{NBm} + \text{WBm} - \text{NWBm} \\ & + (\text{NWp} + \text{Sep} - \text{Nep} - \text{SWp})/n^2 \end{aligned} \quad (2)$$

where

NBm = North Block mean (mean of the block 1 pixel higher)

WBm = West Block mean (mean of the block 1 pixel to the left)

NWBm = North West Block mean (mean of the block 1 pixel to the left and 1 pixel higher)

NWp = North West pixel (intensity of the top-left corner pixel)

SEp = South East pixel (intensity of the bottom-right corner pixel)

NEp = North East pixel (intensity of the top-right corner pixel)

SWp = South West pixel (intensity of the bottom-left corner pixel)

n^2 = size of the low-pass filter

Based on this observation, a new algorithm for intensity mean calculation is proposed here by linking it to the means of other blocks that have been obtained earlier; this is illustrated in equation 2.

Substantial gain in computational speed in the mean calculation can be achieved because there are now only 7 arithmetic operations (6 additions/subtractions and 1 division) in the computation, and it is independent of n . In contrast, in the standard approach as illustrated by equation (1), there are n^2 arithmetic operations (n^2-1 additions and 1 division). This suggests that the computational load will rise very rapidly with the size of the filter. Table A below gives a quantitative comparison between the standard and the proposed methods.

Table A Computational speed comparison between the standard and proposed method

n^2 (filter size)	Number of arithmetic operations in the standard approach	Number of arithmetic operations in the proposed algorithm	Gain in speedup
9	9	7	1.29
25	25	7	3.57
49	49	7	7.00
81	81	7	11.57
121	121	7	17.29

It can be seen that the gain in speedup is given by the ratio $n^2/7$ where n^2 is the size of the filter. For a large filter, the gain is quite significant, e.g., for a 11×11 low-pass filter, the gain is a substantial 17 times.

The proposed algorithm is independent of the size of the filter and involves only a fixed low number of arithmetic operations. This is superior to the standard approach in which the number of arithmetic operations grows rapidly with the size of the filter. The substantial gain in computational speed makes the proposed algorithm an ideal candidate for usage in all flicker parameters based flicker removal methods proposed in this thesis.

Appendix D

Flicker Correction Results Comparison

Flicker correction method	Variance of intensity mean curve		
	Lady sequence	Tunnel sequence	Lantern sequence
No Correction	3.11	3.79	0.60
Histogram Mapping method	0.05 (1.6%)	0.02(0.5%)	0.03(5.0%)
Intensity Mean Averaging method	0.37 (11.9%)	0.99 (26.1%)	0.17 (28.3%)
Intensity Mean Averaging with Local Enhancement method	0.37 (11.9%)	0.81 (21.4%)	0.25 (41.7%)
Motion-compensated Intensity Mean Averaging with Local Enhancement method	0.37(11.9%)	0.81 (21.4%)	0.25 (41.7%)
FPE method	0.36 (11.6%)	0.021 (0.6%)	0.017(2.8%)
LSM method	0.74 (23.8%)	0.039 (1.0%)	0.017 (2.8%)
FPE with stationary block re-examination method	0.33 (10.6%)	0.019 (0.5%)	0.016 (2.6%)
Motion-compensated FPE Method	0.02 (0.64%)	0.003 (0.08%)	0.016(2.6%)
Simplified Motion-compensated FPE Method	0.41(13.2%)	0.19 (5.0%)	0.10 (16.7%)
Simplified Motion-compensated FPE Method with intensity mean averaging	0.07 (2.2%)	Not applicable	Not applicable

	Lady sequence	Woman sequence	Lantern sequence
Simplified Motion- compensated FPE Method for color sequence	0.35 (11.3%)	0.39 (12.3%)	0.07 (11.6%)
Simplified Motion- compensated FPE Method for color sequence with intensity mean averaging	0.06 (1.9%)	0.29 (9.2%)	Not applicable

[figure inside () is the percentage value relative to that with no correction]

Appendix E

Organizational Chart of Image Files

Chapter 2

2.1 Sequences for flicker

2.1.1 Good (flicker-free) sequences

- West*
- Taxi*
- Fountain[†] (color)

2.1.2 Synthetic flicker sequences

2.1.2.1 Global flicker added

- West-global

2.1.2.2 Local flicker added

- Taxi-local

2.1.3 Natural flicker sequences

- Lady*
- Tunnel*
- Lantern*
- Lady* (color)
- Woman* (color)

2.2 Sequences for stain

2.2.1 Good (stain-free) sequences

- Wedding[†] (color)

2.2.2 Natural stain sequences

- Sword-Fighting[#] (color)
- Shock[#] (color)

Chapter 3

3.1 Histogram Mapping method

- Corrected Lady
- Corrected Tunnel

-- Corrected Lantern

3.2 Intensity Mean Averaging method

-- Corrected Lady

-- Corrected Tunnel

-- Corrected Lantern

Chapter 4

4.1 Histogram Mapping with Local Enhancement

-- Corrected Lady

4.2 Intensity Mean Averaging with Local Enhancement

-- Corrected Lady

-- Corrected Tunnel

-- Corrected Lantern

Chapter 5

5.1 Motion-compensated Intensity Mean Averaging with Local Enhancement

-- Corrected Lady

-- Corrected Tunnel

-- Corrected Lantern

Chapter 6

6.1 FPE method

-- Corrected West

-- Corrected Taxi

-- Corrected Lady

-- Corrected Tunnel

-- Corrected Lantern

6.2 LSM method

-- Corrected West

-- Corrected Taxi

-- Corrected Lady

-- Corrected Tunnel

-- Corrected Lantern

6.3 FPE method with Stationary Block Re-examination Incorporated

-- Corrected West

-- Corrected Taxi

-- Corrected Lady

-- Corrected Tunnel

-- Corrected Lantern

Chapter 7

7.1 Motion-compensated FPE algorithm

-- Corrected West
-- Corrected Taxi
-- Corrected Lady
-- Corrected Tunnel
-- Corrected Lantern

Chapter 8

8.1 Simplified motion-compensated FPE method

-- Corrected West
-- Corrected Taxi
-- Corrected Lady
-- Corrected Tunnel
-- Corrected Lantern

Chapter 9

9.1 Flicker correction in color motion pictures

-- Corrected Fountain
-- Corrected Lady
-- Corrected Woman
-- Corrected Lantern

Chapter 10

10.1 Stain removal

-- Corrected Wedding
-- Corrected Sword-fighting
-- Corrected Shock

* Courtesy of Parallel Processing Laboratory, Nanyang Technological University, Singapore

† Standard Evaluation Material (StEM), courtesy of DCI (Digital Cinema Initiatives) LLC

Courtesy of Celestial Films Hong Kong Ltd

Appendix F

List of Publications

Conference publications:

1. Xue Yang, Man Nang Chong, and Kin Keong Wong, "A Stable Flicker Removal Technique for Film Archives", Third Image Processing: Mathematical Methods, Algorithms and Applications, Leicester, U.K. Aug 2000.
2. K.K. Wong, A. Das & M.N. Chong, "Flicker Removal Through Histogram Mapping", 11th International Workshop on Systems, Signals and Image Processing (IWSSIP04), Poland, Sept 2004.
3. K.K. Wong, A. Das & M.N. Chong, "Improved Flicker Removal Through Motion Vectors Compensation", Third International Conference on Image and Graphics (ICIG 2004), Hong Kong, Dec 2004.
4. K.K. Wong, A. Das & M.N. Chong, "Applying Motion Vectors to Enhance Flicker Correction", International Conference on Signal Processing (ICSP 2004), Dec 2004, Istanbul.
5. K.K. Wong, A. Das & M.N. Chong, "Flicker Correction Using A Simplified Flicker Parameter Model", The 8th IASTED International Conference on Computer Graphics and Imaging, CGIM 2005, Aug. 2005, Honolulu, Hawaii, USA.

6. K.K. Wong, A. Das & M.N. Chong, "Flicker Correction in Color Motion Pictures", ICGST International Conference on Graphics, Vision and Image Processing (GVIP-05), Dec 2005, Cairo.
7. K.K. Wong, A. Das & M.N. Chong, "Correcting Flicker in Color Motion Pictures", Digital Image Computing: Techniques and Applications (DICTA), Dec 2005, Cairns, Australia.
8. K.K. Wong, A. Das & M.N. Chong, "Stain Removal in Color Motion Pictures", Asia-Pacific Workshop on Visual Information Processing (VIP2005), Dec 2005, Hong Kong.
9. K.K. Wong, A. Das & M.N. Chong, "Flicker Correction using Intensity Mean Averaging", International Conference on Signals and Electronic Systems (ICSES'06), 17-20 September 2006, Lodz, Poland.

Journal publications:

1. K.K. Wong, A. Das & M.N. Chong, "An Improved Flicker Removal Method using Stationary Block Re-examination", pp 430-434, WSEAS Transaction on Computers, April 2004.
2. K.K. Wong, A. Das & M.N. Chong, "Effective Flicker Removal through Repetitive Corrections", pp 277-282, WSEAS Transactions on Circuits and Systems, April 2004.
3. K.K. Wong, A. Das & M.N. Chong, "Applying Motion Vectors to Enhance Flicker Correction", International Journal of Signal Processing (IJSP), Jan 2005.
4. K.K. Wong, A. Das & M.N. Chong, "Stain Removal in Degraded Color Motion Pictures", IEEE Transactions on Circuits and Systems for Video Technology (submitted)
5. K.K. Wong, A. Das & M.N. Chong, "A Method for Correcting Flicker in Color Motion Pictures", IEEE Signal Processing Letters (submitted)