Flicker Removal and Color Correction for High Speed Videos

Ali Kanj

▶ To cite this version:

Ali Kanj. Flicker Removal and Color Correction for High Speed Videos. Other [cs.OH]. Université Paris-Est, 2017. English. NNT : 2017PESC1115 . tel-01744714

HAL Id: tel-01744714
https://pastel.hal.science/tel-01744714

Submitted on 27 Mar 2018

UNIVERSITÉ
PARIS-EST
MARNE-LA-VALLÉE

École Doctorale Mathématiques et Sciences et Technologies de l'Information et de la Communication

Thèse présentée pour l'obtention du titre de

Docteur en Traitement du Signal et des Images

Flicker Removal and Color Correction for High Speed Videos

par

Ali KANJ

à soutenir le devant le jury composé de

Rapporteur	Fréderic	DUFAUX
Rapporteur	Fabrice	MERIAUDEAU
Examinatrice	Julie	DELON
Examinateur	Etienne	DÉCENCIÈRE
Examinateur	Raoul	RODRIGUEZ LUPARELLO
Directeur de Thèse	Hugues	TALBOT
Co-Directeur de Thèse	Jean-Christophe	PESQUET

Le 7 Juillet 2017

Abstract

Deflickering consists of removing rapid, artifactual changes of luminosity and colorimetry from image sequences and improving luminosity consistency between video frames. It is a necessary and fundamental task in multiple applications, for instance in archived film sequences, compressed videos and time-lapse videos. In recent years, there has been a renewal of interest for improving luminosity consistency acquisition technology in the flicker removal problem, in particular for periodic flickering. In this context, flicker corresponds to undesirable intensity and chroma variations due to the interaction between the acquisition frequencies on the one hand, and the alternating current powering the light sources on the other hand. The present thesis formulates the periodic deflickering problem in high speed videos, studies the physical properties of flicker and suggests both theoretical and experimental solutions for its removal from image sequences. Finally, a new flicker removal approach is proposed performing jointly motion tracking and color correction.

Résumé

Le deflickering consiste à supprimer le scintillement présent dans les séquences d'images afin de réduire les variations lumineuses entre chacune des images de la vidéo. Il s'agit d'une tâche essentielle, nécessaire dans plusieurs applications, en particulier dans les séquences de films archivés, les vidéos comprimées et les vidéos time-lapse. Au cours de ces dernières années, avec le développement des technologies d'acquisition à haute vitesse, il y a eu un regain d'intérêt pour le problème de suppression de flicker, en particulier le flicker périodique. Dans ce contexte, le flicker correspond à des variations indésirables de luminosité et des couleurs dues à l'interaction entre la vitesse d'acquisition d'une part, et d'autre part le courant alternatif alimentant les sources lumineuses. La présente thèse formule le problème du déflickering périodique dans les vidéos à haute vitesse, étudie les propriétés physiques du flicker et propose à la fois des solutions théoriques et expérimentales pour sa suppression des séquences d'images. Enfin, une nouvelle approche est proposée permettant d'effectuer simultanément le suivi de mouvement et la correction des couleurs.

Acknowledgements

I would like to thank the members of the jury, for they have accepted to assess the present thesis: Fréderic Dufaux, Fabrice Meriaudau, Julie Delon, Etienne Décencière.

Hugues Talbot and Jean-Christophe Pesquet were my phd advisors during the last three years. They accepted me first as a research trainee and then as a phd candidate, they encouraged me to think freely. I am grateful for their availability and expertise. I am thankful for their patience, positivity and open-mindedness.

I would like to thank SUBLAB Production, represented by Raoul Rodriguez Luparello, for funding this phd studies.

The LIGM lab members accepted me amongst them. I thank them for their welcome and for their cheerful mood and for the good memories we share.

Last, for obvious reasons, I thank my wife Ghina and my family members.

Contents

List of Figures

Chapter 1

Introduction

1 Context

Image acquisition technology has improved significantly over the last few years, both at the scientific and industrial levels. These improvements had considerable impact on the consumer market.

First, over the last two decades, the entire pipeline of video production from acquisition to playback via editing has moved from analog to digital tools, allowing perfect replay and archiving.

Secondly, both spatial and temporal resolutions have increased tremendously from the old PAL/NTSC standards of the 20th century, where VGA resolution and 25 frames per second (fps) were the norm.

As a consequence, it is possible now to acquire videos at 4k resolution (3840×2160) and 1000 fps.

While increased spatial resolution induces few drawbacks besides necessitating significant computing power to process: increased temporal resolution may produce artifact, particularly when using artificial lighting, and when the camera frame rate approaches or exceeds the power frequency of the AC current. In this case, the illumination artifacts and chrominance changes may become visible over the whole frame. These variations are commonly termed “periodic flicker”. Indeed, when the motion in the sequence is complex and the scene is illuminated from several light sources with potentially different properties (incandescent, fluorescent, etc..), or itself contains lighting appliances, avoiding or correcting these artefacts can be a difficult problem.

This problem is likely to become more prevalent with the recent advent of consumer-level high speed video acquisition devices, for instance in newer smartphone generations or with sports/action cameras providing high speed acquisition options.

The aim of the present manuscript, is to contribute to a better understanding of the flickering problem and properties in general, and to suggest new methods for the periodic and aperiodic flicker removal in high speed videos in particular.

The first strategy for flicker removal considers a global, frame-based correction, especially if the lighting conditions and motion are not very complex, i.e. it consists of processing all the pixels of an image in a uniform manner. We propose a method based on histogram matching for color correction, and another global method based on the image registration using key points detection. These methods are promising in terms of real time constraints. This strategy will be examined in chapter 4.

The second strategy, is local, able to deal with multiple light sources illuminating the scene and in the presence of complex motions. We propose a first local approach for color correction, based on a block matching algorithm that is able to include the constraints of illumination and chroma variations. A second approach is suggested, which is based on superpixels tracking for motion estimation and color correction. This local strategy will be studied in chapters 5 and 6.

2 Collaboration

The presented work in this manuscript was carried out at Laboratoire d'Informatique Gaspard Monge of the University of Paris-Est Marne-la-Vallée, in collaboration with Sublab production, a service provider company based in France and Spain, that offers support for the most complex technical aspects of audiovisual projects. Sublab provides high speed imaging equipment, a packshot studio, and solutions for lighting, underwater shooting, post-production workflow, moving shots, etc.

Thanks to the technical team in Sublab, this collaboration led me to understand the techniques of high-speed video acquisition and to create our database of image sequences on which our different algorithms have been tested.

3 Organization of the manuscript

Chapter 2: Generalities on color acquisition devices, and introduction to the flicker removal problem

Chapter 2 will be divided into four main sections. First, a quick reminder about the color acquisition devices for computer vision applications will be presented in order to familiarize the reader with some useful basics such as color perception, optical sensors. The second section will describe the history of high speed acquisition and its applications. The third section will describe and interprete flicker problems for multiple applications, and then specifically in high speed videos. In the last part of this chapter, we will propose a general strategy for video color correction and flicker removal for periodic and pseudo-periodic flicker in high speed videos.

Chapter 3: State of the art - Preliminaries

Chapter 3 will present an overview of existing motion estimation and color correction techniques for images in different contexts.

In this chapter, we will first briefly describe the classical set of methods for motion estimation. We will turn to methods adapted to the presence of illumination variations. We will then describe some color correction approaches, in particular the model-based and non-parametric based techniques.

Chapter 4: Global methods for color correction

In chapter 4, periodic flicker in high speed videos will be considered. A reasonable approach considering a global frame-based correction will be proposed, which is applicable when the lighting conditions and motions are not very complex.

Two global flicker removal methods will be described. The first method is based on histogram matching for color correction. The second method is based on the image registration using keypoint detection.

Chapter 5: Block matching-based colorimetric correction

In chapter 5, we will propose a flexible local color correction technique to remove flickering artifact, qualified as periodic/non-periodic, suitable for correcting high-speed color video taken under artificial lighting, with more complex motions and lighting conditions.

We will describe a causal (with single reference frame) and non-causal (with multiple reference frames) tracking methods involving per block color correction matrix estimation, and followed by a per-pixel post-processing / block artifact removal approach.

Finally, we will introduce a pyramidal strategy based on the causal approach, to estimate tracking and correction parameters at reduced scales. This principle will be useful to reduce the method computational time both for real time implementations and for offline applications.

Chapter 6: Local method based on superpixels and spatial interpolation

In chapter 6, we will propose new avenues for video color correction, and typically for flicker removal applications. We will take the viewpoint that it is not always necessary to track and model the color correction for all image regions in a video for local color correction.

We will suggest another local method based on superpixels segmentation, that is able to track objects with undefined and complex shapes, and find the corresponding local color correction. We will also describe one dimensional and two dimensional spatial interpolation methods in order to use one of them in the post-processing step.

Chapter 7: A comparative chapter

In chapter 7, a comparison between all proposed methods will be made in order to present the advantages and disadvantages of each method. We will compare our three proposed approaches. The global image registration based method (in chapter 4), the local block-based colorimetric correction method (in chapter 5) and the local superpixel tracking based flicker removal approach (in chapter 6). We will test these methods on four real, studio-lit videos affected with periodic flicker and featuring multiple light sources and global/local complex motions and also on three synthetic periodic flicker sequences produced from a flicker-free, naturally lit video, in order to quantitatively compare the processing results in terms of signal to noise ratio, and to prove the efficiency of each method against different colorimetric, illumination, motion and acquisition conditions.

Chapter 8: Conclusion and openings

In chapter 8, a general conclusion will be drawn and the potential future enhancements will be listed.

4 Publications

Published conference papers:

- A. Kanj, H. Talbot, and R. Rodriguez Luparello. **Global image registration and video color correction in presence of illumination variations.** *IEEE Fifth International Conference on Digital Information and Communication Technology and its Applications*, pages 92-95, Beirut, Lebanon, 29 April - 1 May 2015.

- A. Kanj, H. Talbot, J-C. Pesquet, and R. Rodriguez Luparello. **Color deflickering for high-speed video in the presence of artificial lighting.** *IEEE International Conference on Image Processing*, pages 976-980, Quebec city, Canada, 27-30 September 2015.

- A. Kanj, H. Talbot, J-C. Pesquet, and R. Rodriguez Luparello. **Correction des variations colorimétriques pseudo-périodiques en vidéo haute vitesse.** *XXVème colloque GRETSI*, Ecole Normale Supérieure de Lyon, Lyon, 8-11 September 2015.

Accepted conference papers:

- A. Kanj, H. Talbot and R. Rodriguez Luparello. **Flicker removal and superpixel-based motion tracking for high speed videos.** *IEEE In-*

ternational Conference on Image Processing, Beijing, China, 17-20 September 2017.

Patent:

- A. Kanj, H. Talbot, J-C. Pesquet, and R. Rodriguez Luparello. **Method for correcting flicker in an image sequence.** *United Kingdom Intellectual Property Office*, 1615349.6, 9 Sep 2016, United Kingdom.

Journal paper:

- A. Kanj, H. Talbot, J-C. Pesquet, and R. Rodriguez Luparello. **A variational method for flicker removal in high speed video sequences,** *to be submitted to IEEE Transactions on image processing.*

Chapter 2

Generalities on color acquisition devices, and introduction to the flicker removal problem

1 Color cameras

The principal element of a camera is the image sensor. It is constituted of a set of photoreceptors which convert the received light rays into electrical output information to provide signals to the digitization system. These photoreceptors are arranged on a plane (matrix camera). Thus, the obtained image is constituted of a set of points called pixels that correspond to the photoreceptors.

In this work, we only consider color sequences, which have become standard for a long time.

Color is a very complex concept because it involves on the one hand human physiology and psychology, and physical phenomena on the other hand. Various theories and studies have been proposed in order to model this rich and complex information. They provided many color representation systems, in which any color can be represented by its digital coordinates.

1.1 Color representation systems

Most humans (at least 90% of the population) have three types of color sensors in the eye, each with a different spectral response, suggesting a representation of three stimuli. Thus, one can indeed reproduce most colors seen in nature from one to three monochromatic sources with different wavelengths.

The first color representation systems (psychological systems) were inspired by the study of human psycho properties to develop color classifications such as CIE (Commission Internationale de l'Eclairage), NCS (Natural Color System), OSA (Optical Society of America) and DIN (Deutsches Institut für Normung). These representations are rarely used for the analysis of color images.

In the acquired image, the color information is defined numerically. Thus, in the image analysis process, colorimetric systems are used in order to distinguish between different colors.

These colorimetric systems could be grouped into four categories: primary systems, luminance-chrominance systems, perceptual system and independent axis systems. We describe below some of the most used color systems.

1.1.1 Standard CIE RGB

In 1931, the International Commission on Illumination (CIE) conducted color matching experiments with 3 monochromatic sources: red ($645.2nm$), green ($526.31nm$) and blue ($444.4nm$) to yield the CIE RGB system. With this system, it is possible to reproduce most of natural colors. However reproducing some natural colors, such as produced by some monochromatic wavelength sources, requires negative weights.

1.1.2 Standard CIE XYZ

To avoid the negative weights, the CIE has developed a tri-stimulus based system, called XYZ, which is derived from RGB system, independent of any physical source, in which all weights are nonnegative. This system is a superset of all visible colours:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2,769 & 1,7518 & 1,13 \\ 1 & 4,5907 & 0,0601 \\ 0 & 0,0565 & 5,5943 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.1}$$

This improvement of RGB system was the first step to a color description system consistent with the human vision. Independently to color, XYZ system introduces the concept of luminance (brightness intensity), yielding directly the Y component. It uses two other positive variables X and Z, to describe visible colors. This opened the way for CIE the xzY system, which purposefully separates the luminance Y and chrominance xz notions.

1.1.3 Standard CIE L*a*b*

In 1976, the CIE showed that in the color space XYZ, the visual difference is more or less important depending on the examined hues. Our eye has greater sensitivity in the blue and can distinguish slight hue variations. Conversely our eye has a low sensitivity in the green and yellow. Therefore, CIE proposed a widely used uniform color space, which is the $L^*a^*b^*$ space defined from XYZ system:

$$L^* = 116\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16 \tag{2.2}$$

$$a^* = 500\left[\left(\frac{X}{X_n}\right)^{\frac{1}{3}} - \left(\frac{Y}{Y_n}\right)^{\frac{1}{3}}\right] \tag{2.3}$$

$$b^* = 200\left[\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - \left(\frac{Z}{Z_n}\right)^{\frac{1}{3}}\right] \tag{2.4}$$

where (X_n, Y_n, Z_n) are the coordinates of a white reference point.

1.1.4 System CIE Lu*v*

CIE $L^*u^*v^*$ is a color space based on the CIE color space $U'V'W'$ (1976), itself based on the CIE XYZ color space. It belongs to the family of uniform color systems: it is derived from a non-linear transformation providing a more uniform color distribution with respect to human perception.

To compute L, u^* and v^* we must first go through $u'v'$ space, which are derived from XYZ:

$$(u', v') = \left(\frac{4X}{X + 15Y + 3Z}, \frac{9Y}{X + 15Y + 3Z}\right) \tag{2.5}$$

The non-linear relations for $L*$, $u*$, and $v*$ are given below:

$$L^* = \begin{cases} \left(\frac{29}{3}\right)^3 \frac{Y}{Y_n}, & \text{if } \frac{Y}{Y_n} \le \left(\frac{6}{29}\right)^3 \\ 116\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16, & \text{if } \frac{Y}{Y_n} > \left(\frac{6}{29}\right)^3 \end{cases} \tag{2.6}$$

$$u^* = 13L^*(u' - u'_n) \tag{2.7}$$

$$v^* = 13L^*(v' - v'_n) \tag{2.8}$$

1.1.5 System YUV

YUV system was invented for the transition from black-and-white television to color. This system avoids the color/illumination representation limits in the RGB space. It allows to send the same analog video signal for black-and-white and color televisions.

Y is derived from XYZ system and represents the luminance intensity, and can be directly displayed on a black and white station. U and V represent the chrominance values.

The YUVsystem is computed from RGB system as follows:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,14713 & -0,28886 & 0,436 \\ 0,615 & -0,51499 & -0,10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.9}$$

The YUVsystem is usually used for the encoding of a composite video signal.

1.2 Acquisition systems

The main element of a camera is the image sensor. It is a photosensitive electronic component for converting electromagnetic radiation (UV, visible or IR) into an analog electrical signal. This signal is then amplified and digitized by an analog-digital converter and then processed to obtain a digital image. The most known sensors are described below.

1.2.1 The FOVEON sensor

Using the capacity of light absorption of silicon, this sensor consists of three photodiodes layers, each at a given depth and corresponding to one of RGB channels.
The Foveon sensor captures the colors vertically, registering the hue, value and chroma for each pixel in the final image. Unlike other conventional sensors that are equipped with RGB demosaicing filters and capture color information in the horizontal plane.
In the Foveon sensor, image data for each pixel are complete and do not require any interpolation.

1.2.2 CCD sensor

The first purely digital technology for image sensors is the CCD (Charge Coupled Device) whose photoreceptors produce an electrical potential proportional to the received light intensity. The color information is sampled using three filters that are sensitive to red, green and blue wavelengths analogously to the human perception system.
CCD sensors have been used very early in the areas of high performance image quality, such as astronomy, photography, scientific and industrial applications. Nowadays, it is very common in cameras, scanners...

There are two main types of CCD color camera:

- **Mono-CCD camera:** They are equipped with a single CCD sensor overlaid with a mosaic of color filters. Thus, photoreceptors that are located at different sites are associated with red, green and blue filters, which are typically arranged in a regular structure. The color information obtained by these several photoreceptors is incomplete, and requires an interpolation for restoring missing color information.

- **Tri-CCD camera:** It is equipped with three registered CCD sensors that are mounted on an optical system of prisms. Each of the three sensors receives respectively the red, green and blue components. Pixel color is given by the response of three photoreceptors, which potentially can offer better resolution and image quality compared with the Mono-CCD sensor cameras. However as resolution increases, it becomes increasingly diffcult to register the 3 sensors to benefit from the extra hardware.

1.2.3 CMOS sensor

CMOS sensor (Complementary Metal Oxide Semiconductors) are micro circuits on a silicon basis. The manufacturing of CMOS sensors is much less expensive than producing CCD sensors. In addition, CMOS consumes much less power than the CCD sencors and can be more reactive. The principle of CMOS is based on the active pixel concept. It combines in each pixel, a photoreceptor, a reading diode and an amplifier circuit. A switching matrix distributed over the entire chip provides access to each pixel independently which is not the case with the CCD sensors. A major difference between CCD and CMOS sensor is in the way images are acquired. CCD cameras typically work with a full frame shutter, meaning that a frame is exposed at once in its entirety. In low lights or high movement, this can result in blur artifacts. Conversely, a CMOS sensor is exposed with a so-called "rolling shutter", meaning that a small and changing area of the image is read while the remainder is continuously exposed, improving light sensitivity. This can result in image distortion rather than blur.

Next, we will describe the color filter array which is placed over the pixel sensors, it is needed because the typical photoreceptors in cameras only detect light intensity, and therefore cannot separate color information.

1.3 Color filter array

The photosites of camera sensors do feature a response spectrum, which is not necessarily flat, but is identical from photosite to photosite. Indeed this response can be thought as a brightness response. To reproduce colors in single-sensor systems, a filter system is used (CFA filter: Color Filter Array) on the sensor surface.

There exists a large variety of CFA setups, but the most commonly used filter is the Bayer from 1976. The Bayer matrix consists of 50% green, 25% red and 25% blue filters. It is often justified by some arguments of resemblance to the human eye physiology, but its main virtues are its simplicity and the fact that it is no longer protected by patents. It is not very efficient however, since it throws away 2/3rd of the incoming light power.

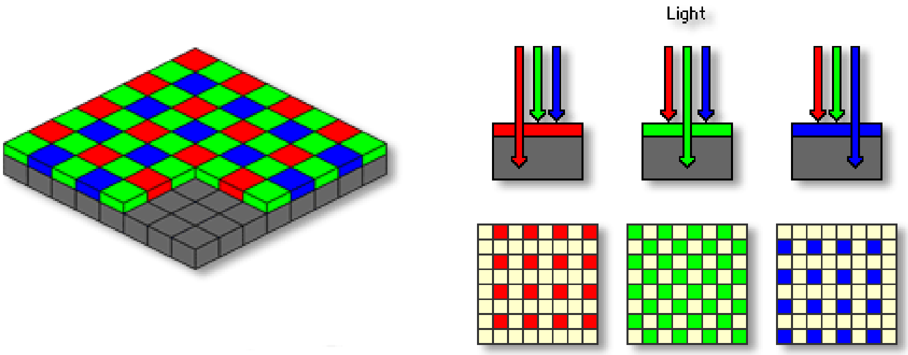


Figure 2.1. Bayer filter

2 High speed acquisition

High-speed acquisition is used to capture fast motion in order to play back recordings in slow motion. It can record phenomena that are too fast to be perceived with the naked eye. This technique is used in several areas: defense, industrial product development, manufacturing, automotive, scientific research, bio-medical applications and entertainment.

2.1 Brief history

Chronophotography is the first high-speed acquisition technique; invented by Eadweard Muybridge in 1878; it consists of taking a rapid succession of photographs to chronologically decompose successive phases of a movement or a physical phenomenon, that is too short to be observed by the naked eye. Today chronophotography is still used, both in science and in advertising or art photography.

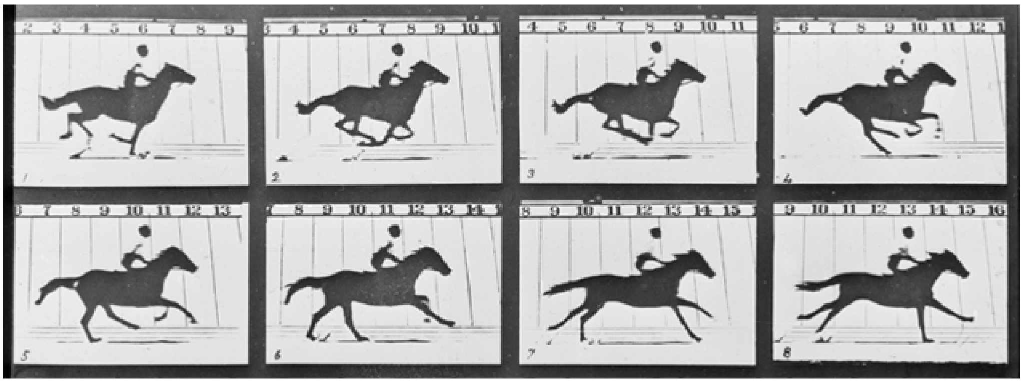


Figure 2.2. Chronophotography: First high speed sequence

The first high-speed cameras used recording film. This recording technique is now considered obsolete, all recent cameras use CCD and CMOS sensors, able to record at high frame rates up to and exceeding 2500 fps, and record them to a digital memory.

Previously, the sampling frequencies of materials, limited by the analog/digital conversion speed, physically restricted the volume of data acquired. In recent years, hardware manufacturers have increased the speed of data collection and allowed engineers and scientists to cross new boundaries in terms of speed and resolution.

2.2 Applications

2.2.1 Scientific research

A large number of scientific studies have been conducted using high-speed imaging techniques. These facilitate understanding and allow a better analysis of many key problems in several fields. It is often used in medical imaging applications, for example to study human and animal blood flow, for retinal imaging, for tracking cells in human body or for studying the living being anatomy.
In addition, high speed imaging is of great interest for military and defense research, because it is able to record explosions, bullets trajectories, guns action, etc. It is also known in underwater researches, for example, to study drops and bubbles, or to track underwater projectiles.

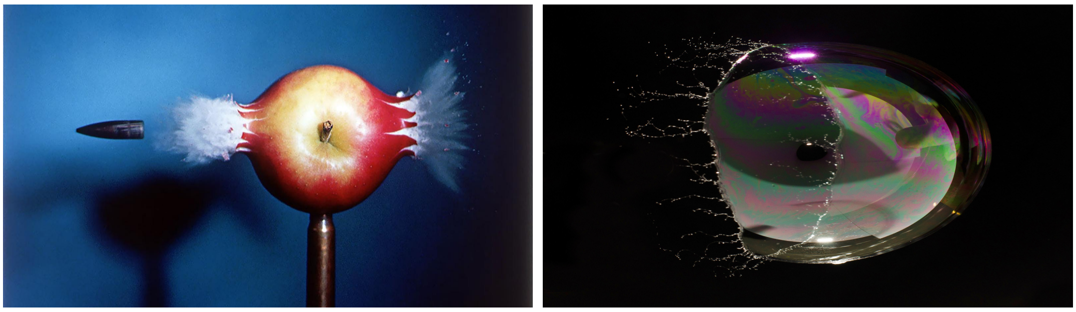

(a) Bullet trajectory (b) Bubble motion

Figure 2.3. High speed imaging examples

High speed imaging is increasingly frequently used in a large variety of research, because analysing and tracking motion can be an interesting pathway to resolve or better understand the problem under study.

2.2.2 Sport events

High speed acquisition has been involved for a long time in all sports broadcastings, it allows to play back interesting actions in slow motion, and sometimes helps the referee to make some decisions (in Tennis, Rugby, ...), so it has become essential to most television watchers.

Figure 2.4. Rugby action in slow motion.

2.2.3 Advertising and media

Advertising is a form of mass communication, which aims to fix the attention of a target audience (consumer, user, etc.) for encouraging people to adopt a desired behavior such as buying a product. To help in this regard, the use of slow motion may be able to ensure acceptance and the attraction to the product.

Figure 2.5. The use of high speed imaging in advertising photography

Next, we shall describe the flicker problem for multiple applications in the state of the art, and specifically in high speed videos.

3 The flicker problem

Flicker in video processing is generally defined as a measurable and undesirable brightness fluctuation in a video sequence. It can happen at any acquisition frequency and can be due to a wide variety of phenomena such as sensor artifacts, sudden illumination changes, video transmission problems, and more. It can be transient or periodic, it may affect the whole frame or just a small portion of it. The general problem of flicker removal is highly challenging since it is linked to an objective brightness change in a sequence, which is not sufficient to identify it. Illumination can change in a sequence for desirable reasons.

3.1 State of the art

Methods for video luminosity stabilization have been studied in the literature, for instance in underwater image sensing, surveillance systems, camcorder videos, archived videos, image-video compression and time lapse videos. However, modes for the flicker effect differ from one application to another.

3.1.1 Underwater image sensing

Sunflicker effect, which is a challenge in underwater image sensing, is non-periodic, it is created from refracted sunlight casting fast moving patterns on the seafloor. Shihavuddin et al. (2012) proposed a sunflicker removal method which considers sunflicker effect as a dynamic texture. They proposed to warp the previous illumination field to the current frame, then to predict the current illumination field, by finding the homography between previous and current frames, and finally removing sunflicker patterns from images. It can be considered as a local flicker effect in images.

3.1.2 Video surveillance

Flicker is also annoying in the indoor smart surveillance cameras in presence of fluorescent lamps, which makes tracking persons more complicated, Ozer and Wolf (2014) proposed to eliminate background which contains reflective surfaces and shadows, and track the regions of interest only.

3.1.3 Camcorded videos

Camcorded videos are flicker affected when scrolling stripes are observed with the video. This problem is due the fact that the higher frequency backlight of the screen is sampled at a lower rate by the camcorder. In this case, flicker is periodic and it is defined as a rectangular signal. Baudry et al. (2015) used the temporal discrete Fourier transform (DFT) to estimate flicker parameters.

3.1.4 Archived films

Fuh and Maragos (1991) developed a model allowing for affine shape and intensity transformations. They proposed a method for motion estimation in image sequences, which is robust to affine transformations at motion or light intensity levels. They implemented a block matching algorithm taking into account brightness variations between two images. This method can predict motion and illumination variation parameters between two grayscale images, i.e. the rotation angle, two translation components and a brightness amplitude factor. A pixel-recursive model was developed by Hampson and Pesquet (2000) to estimate simultaneously motion and illumination variation parameters.

We can classify existing methods for flicker removal in archived videos into two categories depending on whether they use a linear or a non-linear model. Decencière (1997) proposed a linear model linking the observed image to the original image. Van Roosmalen et al. (1999) used also an affine model taking into account spatial dependency. Yang and Chong (2000) and Van Roosmalen et al. (1999) estimated flicker parameters by interpolation, and Rosmalen and Yang suggested to resolve block mismatching problems caused by occlusions or blotches, but their methods may fail to detect outliers. Ohuchi et al. (2000) used the same affine model together with an M-estimator to find flicker parameters. They considered objects in motion as outliers, which may lead to failures in the presence of large objects. Kokaram et al. (2003) proposed a model to reduce the accumulating error taking the previous restored frame as reference. They used also a linear model with a robust regression together with registration. Zhang et al. (2011) offer a method to generate reference images followed by a proposal to correct flicker using a linear model involving an M-estimator. Non-linear models have also been studied. Among them, Naranjo and Albiol (2000) and Delon (2006) find flicker parameters by optimizing a non-linear model using histogram matching. Pitié et al. (2004) used Naranjo's model, taking spatial variations into account. Pitié et al. (2006) expanded these works by developing a method for finding parameters on a pixel-wise basis, allowing for non-linear flicker distortions. Separately, Vlachos (2004), Forbin et al. (2006) and Forbin and Vlachos (2008) computed differences of grey-scale histograms between degraded and reference images.

3.1.5 Video coding

In image sequence coding, flicker may appear when similar regions between two images are incoherently encoded. Ren et al. (2013) proposed a block-matching, adaptive multiscale motion method. This method may encounter problems in presence of thin objects or outliers and tends to smooth image details. Jimenez Moreno et al. (2014) used a low pass filtering approach to compensate flicker. Unfortunately, most of these works consider grey-level sequences and none specifically deals with periodic flicker.

3.1.6 High speed videos

The flicker problem is relatively recent in high speed acquisition. In this specific context relatively little research has been conducted, except for a few patents published in the last few years. For instance, Sorkine Hornung and Schelker (2015) provide techniques for stabilizing coloration in high speed videos based on reference points. The coloration adjustment component applies a per frame color mapping based on sparse color samples tracked through all frames of the video. It identifies reference frame, and computes color histograms for the reference and subsequent frames. From the reference histogram, they select a number of colors and consider them as reference colors for the remaining frames in the video. This performs a kind of image registration by tracking locations of the selected reference points in the reference frame through the sequence of frames. In their work, the moving foreground object is ignored supposing that the camera motion is very small in high speed videos and outlier reference points are out of consideration.

In another patent application by Asano and Matsusaka (2007), moving averages of accumulative histograms are calculated for each frame on image data. Then, gamma tables for correcting the image data of a frame in the plurality of frames are built so that the accumulative histograms after corrected with the gamma tables match with the moving averages of the accumulative histograms. Then, the image data of the frame in the plurality of frames is corrected with the gamma tables. In some cases, each frame is divided into areas, and the flicker correction is performed for each area if necessary. In summary, this patent ignores colorimetric variations, does not take local motions into account and performs intra-channel histogram matching providing some brightness artifacts.

3.2 Flicker in high speed acquisition under artificial lighting

In all video shootings, sufficient lighting is essential to achieve a successful recording, to illuminate the scene first, and then to adjust brightness and contrast in images.

3.2.1 Natural light

Sunlight is neither constant nor uniform. The height of the sun, clouds, pollution etc. are all factors that affect this natural light source and cause endless variations. Moreover, the sun color temperature also varies depending on the time of day. It is therefore possible to achieve different visual appearances with respect to the shooting time. These variations are clearly perceptible in time lapse videos, that are created from a large amount of images taken for the same place and at a specific time interval. However, the natural light can be considered uniform for small periods.

3.2.2 Artificial light

We use artificial light sources when the light level is no longer sufficient. This occurs while shooting indoor or at night, and sometimes when a more controlled lighting is needed. The common types of artificial light sources existing today are: incandescent, fluorescent, LED, and studio strobe.

Most lighting devices do not emit a regular light flux. The induced lighting variation is generally called periodic flicker. If a lamp is turned on and off at a very fast rate, there comes a threshold where we feel that the light stays on without interruption. This phenomenon is due to the persistence of the image on the retina. For most people, the illusion starts when the frequency of the lighting intensity variation is above 60 Hz.

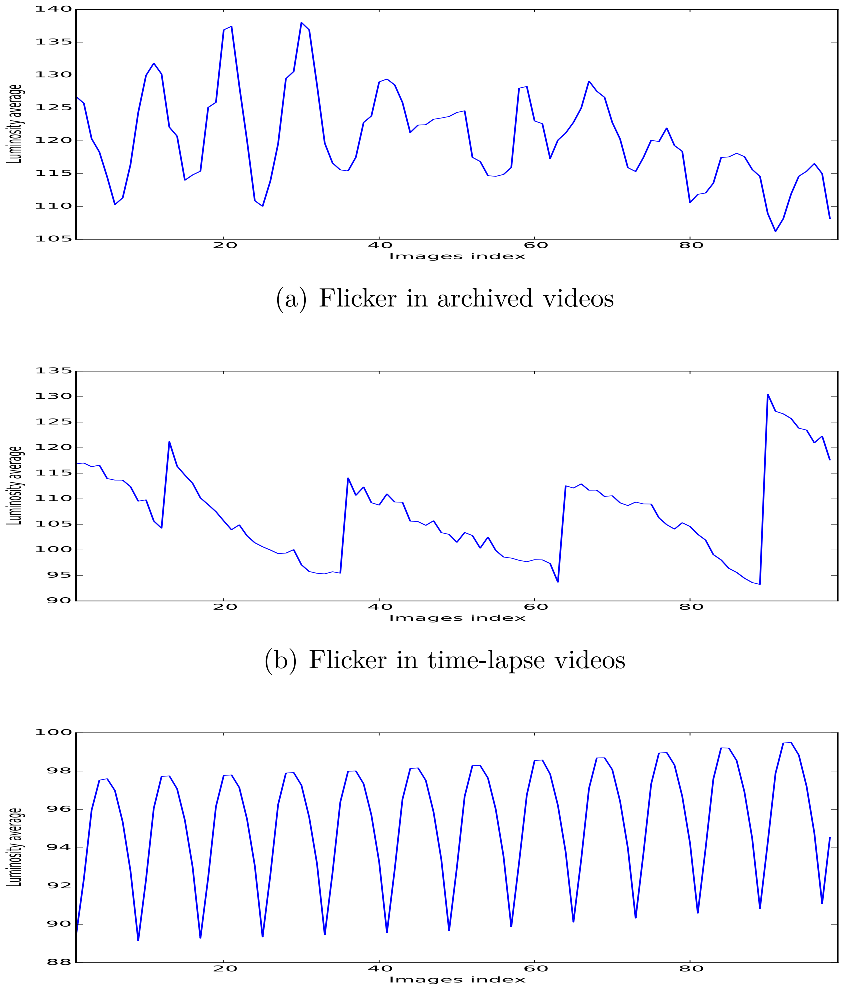
The electrical current supplied by a standard European socket (230V / 50Hz) is called "alternating current AC" because the flow of electrical charge inside periodically reverses direction, whereas in direct current (DC), the flow of electrical charge is constant in one direction. If an incandescent lamp is operated on a low-frequency current, the filament cools on each half-cycle of the alternating current, leading to perceptible change in brightness and flicker of the lamps; the effect is more pronounced with arc lamps, and fluorescent lamps.

In high speed video acquisition, camera can record with very high frame rates, at any speed from 64 frames-per-second (fps), which allows to capture all lower frequencies, especially those which may not be noticeable to the human eye (around 25 fps).

3.3 Flicker analysis

Flicker analysis differs from one application to the next. With respect to archived movies, flicker is defined as unnatural temporal fluctuations in perceived image intensity that do not originate from the original scene, and it is due to variations in shutter time. This artifact presents global, sudden and *random* variations of luminance and contrast between consecutive frames of a sequence (See Figure 2.6(a)).

Time-lapse acquisitions are very slow and sometime irregular sequence acquisitions, that are used in animations, advertisement or documentary films. They



(a) Flicker in archived videos

(b) Flicker in time-lapse videos

(c) Flicker in high speed imaging videos

Figure 2.6. Comparison between several types of flicker

are typically used to accelerate rates of changes, for instance to show how a plant grows. Flicker may occurs when luminosity is varying depending on the time of day under natural lighting, or because of artificial lighting if the recording takes place at night and also according to the weather. This kind of flicker is defined as a random luminosity variation (see Figure 2.6(b)), and usually it is compensated in a causal manner refering to a past frame in each sequence portion which induces some luminosity distortions in the corrected sequence.

3.3.1 Brightness variations

In high speed video acquisition, flicker corresponds to an undesirable *periodic* intensity and chroma variations due to the interaction between the lighting and acquisition frequencies (See Figure 2.6(c)). Assuming negligible motion and a simple camera model, at the acquisition speed, each pixel value on a frame corresponds to the integration of a lighting function. Standard light sources such as incandescent or fluorescent lamps powered by an alternating current are subject to power variations at a frequency ν of about 100 or 120Hz. At usual video rates (25 frames per second), these variations are integrated over a long enough period. Assuming approximately sinusoidal variations, we have at a given pixel $\mathbf{s}$ and time t:

$$\begin{aligned} f(\mathbf{s},t) &= \frac{f_0(\mathbf{s},t)}{T}\int_t^{t+T}\big(1+\Delta(\mathbf{s})\cos(2\pi\nu\tau+\varphi)\big)d\tau \\ &= f_0(\mathbf{s},t)\Big(1+\Delta(\mathbf{s})\frac{\sin(2\pi\nu(t+T)+\varphi)-\sin(2\pi\nu t+\varphi)}{2\pi\nu T}\Big) \\ &\sim f_0(\mathbf{s},t) \end{aligned} \tag{2.10}$$

where $T \gg 1/\nu$ is the exposure time, $\varphi \in [0, 2\pi[$ is a phase shift, $f_0(\mathbf{s},t)$ is the field intensity in the absence of illumination variations, and $\Delta(\mathbf{s})$ is the magnitude of these variations. Therefore, the intensity / chroma variations are usually not perceptible.

In contrast, at high acquisition frame rates, the integration of the lighting function is performed over a very short interval $T \ll 1/\nu$. Then, we get

$$f(\mathbf{s},t) \sim f_0\big(\mathbf{s},t\big)\big(1+\Delta(\mathbf{s})\big(\cos(2\pi\nu t+\varphi)-\pi\nu T\sin(2\pi\nu t+\varphi)\big)\big) \tag{2.11}$$

and the intensity variations are no longer negligible.

3.3.2 Chromatic variations

In the lighting field, the color temperature provides information on the general hue of the emitted light: from the "warm" hues - where the red is dominating - like sunrise or sunset colors, to the "cold" hues - where the blue color is dominating -

like in the intense midday sun colors, or over a snow field. The color temperature is given usually in Kelvin degrees ($^\circ K$), as the color temperature increases, so does the proportion of blue in light. These temperatures are those of the equivalent black body radiation that would be producing the perceived color.

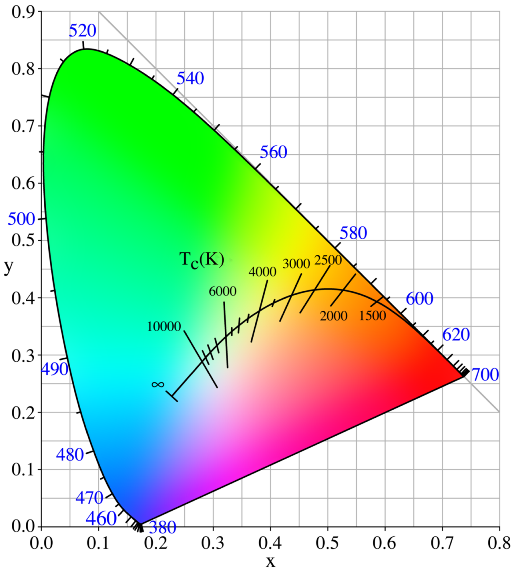


Figure 2.7. Chroma alterations with light temperature variations

With most light sources, power variations are accompanied with changes in lighting temperature, inducing visible chroma alterations. For example, while using the classical incandescent lamps (see Figure 2.8), the lamp filament temperature is proportionately varying to the power alterations, so with each AC period, the light color becomes first red and gradually transforms into blue.

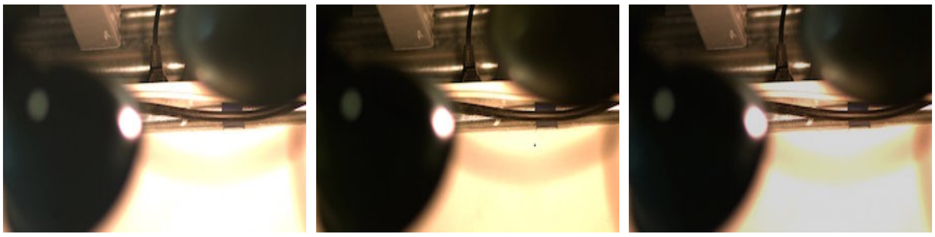


Figure 2.8. Rapid illumination / chromatic changes in a high-speed video sequence.

In contrast, fluorescent and LED light sources do not use a filament to produce lights, so the excitation of the chemical substances containing phosphorus, allows the perception of the emitted visible light (see an example is Figure 3.7).

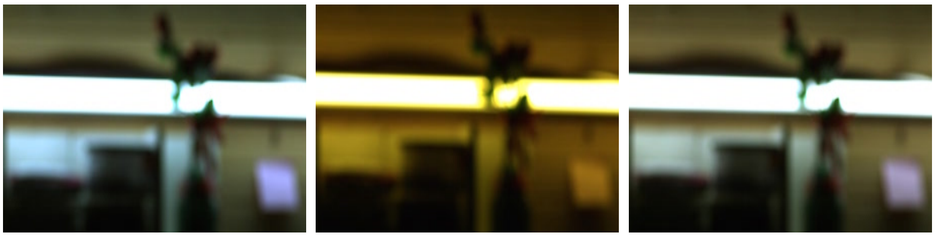


Figure 2.9. Illumination / chromatic changes in a high-speed video sequence under fluorescent lighting.

4 General strategy for video correction

To address this periodic illumination and color variations problem, we assume that we know the acquisition frame rate and the local mains AC frequency. We therefore can estimate the period of the flickering effect. We also assume that general intensity variation integrated over few flickering periods is slow. Given this, it is easy to find the peak illumination intensity in the local period. In Figure 2.10, the acquisition frame rate is 1000 images/second and the AC mains power frequency is 100Hz. As expected, the period of the flicker is 10 frames long.

4.1 Selecting reference frames

In theory, any frame in a flicker period could serve as a reference, but we expect the reference frame to be the richest in information. However, it is easier to detect peak illumination in a period, and use the last maximum intensity frame in a period, at time t_{ref}, as a reference image. This frame may be considered as a good choice in terms of signal-to-acquisition noise ratio.

In this goal, we computed the average of pixel intensities in each image along the sequence, so a selected reference frame in such a flicker period should be the

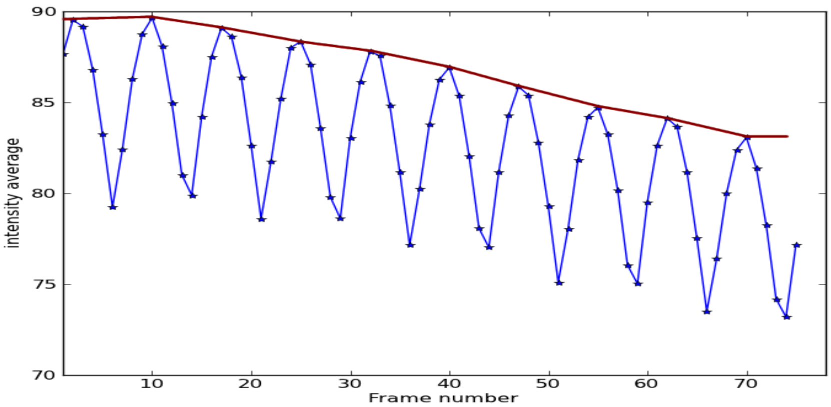


Figure 2.10. The blue curve shows the luminosity variation in an affected image sequence. In red we show the intensity variation in the absence of periodic flicker.

one having the highest average in this period.

As seen in Figure 2.10, changes in global illumination not related to flicker may induce low frequency variations in the luminosity of the scene.

4.2 Cross-channel correlation study

In order to check our assumption regarding the influence of periodic flicker on illumination, we acquired a sequence affected with periodic flicker due to a single source and no motion. In Figure 2.11, we plotted the cross-channel correlations between a pixel in a manually selected reference image and a subsequent frame in the same period. We observe that the vast majority of pixel colors transform linearly from the flicker-affected frame to the reference image, except at the extremal range values, due to sensor saturation effects. In the remainder, we thus assume a linear color transformation model between a reference and flicker-affected frame.

4.3 Flicker correction model

We model the color illumination transform by a 3×3 matrix $\boldsymbol{M}_{\mathbf{s},t}$ between two pixels $\mathbf{s}$ and $\mathbf{s}'$ referring to the same physical area in the scene at times t and t_{ref}

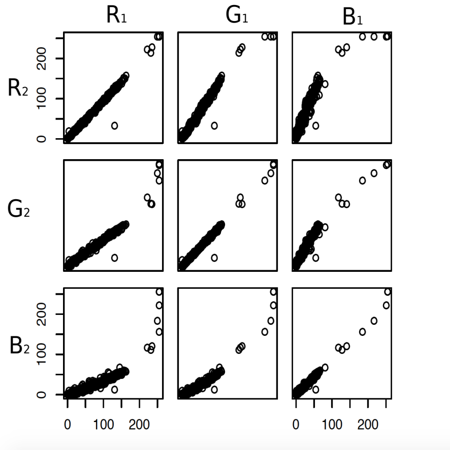


Figure 2.11. Cross-channel correlation between a flicker-affected frame and a reference image.

respectively:

$$\boldsymbol{M}_{\mathbf{s},t} = \begin{bmatrix} r_1(\mathbf{s},t) & g_1(\mathbf{s},t) & b_1(\mathbf{s},t) \\ r_2(\mathbf{s},t) & g_2(\mathbf{s},t) & b_2(\mathbf{s},t) \\ r_3(\mathbf{s},t) & g_3(\mathbf{s},t) & b_3(\mathbf{s},t) \end{bmatrix}. \tag{2.12}$$

Using this matrix, we have

$$\mathbf{f}(\mathbf{s}', t_{\text{ref}}) = \boldsymbol{M}_{\mathbf{s},t}\,\mathbf{f}(\mathbf{s},t) \tag{2.13}$$

where $\mathbf{f}(\mathbf{s},t) \in \mathbb{R}^3$ is the vector of chrominance values of pixel $\mathbf{s}$ at time t. This vector model will play a prominent role in the approaches developed in the remainder of the thesis.

Chapter 3

A short overview of state-of-the-art techniques

In this manuscript, we are studying the problem of video color correction in the presence of complex motions. In this chapter, we first briefly describe the classical set of methods for motion estimation. We then turn to methods for dealing with illumination variations.

1 Motion estimation

The motion estimation in temporal sequences of two-dimensional images is a fundamental problem in image processing. Application areas are numerous and include image compression using motion information, robotics, meteorology with tracking cloud masses, motion tracking in medical imaging (e.g. of the heart or lungs), tracking in surveillance video systems, etc. The images are typically the projection of 3D real scenes. For this reason, we can identify three types of motion: the real motion, the apparent motion and the estimated motion. The apparent motion is often very different from the real motion and generally represents the projection of the real movement in the image plane. For example the Barber's pole illusion, the apparent and real motion fields of a rotational motion are different.

To obtain the estimated motion, several types of methods exist. We have classified them in this chapter into four categories: feature based methods, frequency based methods, differential methods and block matching based methods.

1.1 Feature based methods

Feature based methods for object tracking is performed in two steps: the detection of features in the acquired image sequence, the matching of the detected features. The matching step should be accurate and invariant to several parameters such as the illumination variation or the occlusion of the object in order to detect efficiently the object motion.

Many approaches considering object tracking and motion detection were studied in the literature. A classification of these methods is presented by Yilmaz et al. (2006). The authors distinguished 3 categories: keypoints based tracking Serby et al. (2004), template based tracking Veenman et al. (2001); Birchfield (1998) and shape based tracking Yilmaz et al. (2004).

Points of interest (called keypoints) in an image correspond to discontinuities of the intensity function. These can be caused, by discontinuities of the reflectance function or by depth discontinuities. They can be for example: the corners, the junctions in T or the points of strong variations of texture. A major advantage of the tracking methods based on keypoints detection is that these keypoints are in general invariant to multiple factors. Several examples of tracking based on feature points detection were proposed by Moravec (1977), Harris and Stephens (1988) and Schmid et al. (2000).

An obvious approach for tracking an object is based on the use of a template. Indeed, if the object to be followed is of known shape (such as a car, face, etc), it is relatively simple to find the part of the image most similar to the template considered. To do this, a search is carried out exhaustively on all or part of the image. The information used can simply be intensity or color. The major disadvantages of this method are the slowness of exhaustive search and its sensitivity to brightness variations.

The representation of an object by a simple form such as a rectangle or an ellipse may be unsuitable if the target object is of a very complex shape (e.g. hand, human body, animal, tree, etc.). The representation of such an object by a corresponding shape allows to precisely take into account the shape of the object. The goal of shape based tracking methods is to estimate the shape of the objects of interest for each image of the video. This approach is related to image segmentation.

1.2 Frequency based methods

1.2.1 Fourier shift based method

This method uses the Fourier transform properties to compute the global motion between two images. It does not allow to measure the image displacement directly, but it searches in the image Fourier transform the traces of motion in order to compute the corresponding parameters. The used Fourier transform to model the problem is the continuous Fourier transform in two dimensions.

The Fourier transform preserves the rotations, and transforms a translation into a frequency shift. To calculate the translation between two images, it is sufficient to observe the phase shift between their two respective transforms.

Let $\mathcal{F}$ the Fourier operator, $f(x,y)$ is an image and $F(u,v)$ is its Fourier transformation:

$$F(u,v) = \mathcal{F}[f(x,y)]. \tag{3.1}$$

Let g the translation of f with a translation vector $\vec{t} = (\delta x, \delta y)$, its Fourier transform is given by

$$\begin{aligned} G(u,v) =& \mathcal{F}[g(x,y)] \\ =& \mathcal{F}[f(x+\delta x, y+\delta y)] \\ =& e^{-2i\pi(u\delta x+v\delta y)} F(u,v). \end{aligned} \tag{3.2}$$

The ratio G/F is directly related to the translation vector $\vec{t}$.

A more appropriate method can be used, the phase correlation approach, as explained in the next section.

1.2.2 Phase correlation method

The phase correlation method is based on the Fourier Shift theorem (Yang et al. (2004)) and was originally proposed for the registration of translated images. It is used for estimating the difference between two similar images or other relative translational datasets. It is commonly used in image registration and relies on a representation of data in the frequency domain, it is generally implemented by using the fast Fourier transform. The term is applied to a particular subset of cross-correlation techniques that separate the phase information from the Fourier representation space of the cross-correlogram defined below.

Definition 1. *The cross-correlogram, called also cross-power spectrum is the resulting image of cross-correlation statistics:*

$$\frac{\mathcal{F}(f)\mathcal{F}(g)^*}{\mid \mathcal{F}(f) \mid\mid \mathcal{F}(g) \mid} = e^{2i\pi(u\delta x + v\delta y)},$$

where $*$ indicates the complex conjugate.

In practice, we compute the two dimensional discrete Fourier transform (DFT) for both images and we can apply a Hamming window on the DFT result of both images to reduce boundary effects. A one dimensional Hamming function is being applied on all rows and columns separately.

Definition 2. *The Hamming windowing, also called weighting window, is a classical technique used to focus on a section of measured signal, in order to minimize the distortions that cause a spectral leakage of the Fast Fourier Transform. The associated one-dimensional window is given by*

$$h(t) = \begin{cases} 0.54 - 0.46 \cos 2\pi \frac{t}{T}, & \text{\textit{if} } t \in [0, T] \\ 0, & \text{\textit{otherwise.}} \end{cases}$$

The normalized cross correlation between the two transforms is then calculated. The best phase shift is the one that minimizes the cross correlation function.

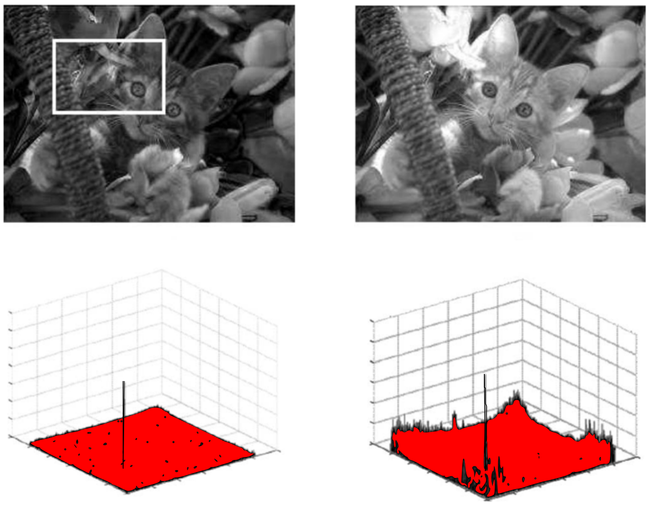


Figure 3.1. Phase correlation example Zitova and Flusser (2003): the graph on the left depicts red-red channel matching and the right one demonstrates red-blue channel matching. The peaks correpond to the matching positions.

De Castro and Morandi (1987) extended the phase correlation method to include the rotation transformation. An affine image registration is performed using phase correlation and log-polar mapping based on a log-polar coordinates system (where a point is identified by two numbers, one for the logarithm of the distance to a certain point, and one for an angle) by Wolberg and Zokai (2000). Foroosh et al. (2002) have introduced a phase correlation extension for sub-pixel registration by means of the analytical expression of phase correlation on downsampled images.

The phase correlation technique is relatively robust because all frequencies contribute to the calculation of correlation coefficients, and it is relatively fast thanks to the fast Fourier transform, but it is practically limited to global, translation-only transforms on the whole image.

1.2.3 Analytical Fourier-Mellin invariant (AFMI)

The use of the Analytical Fourier-Mellin transform (AFMT) allows us to calculate the similarity parameters (rotation and homothety) as a translation vector, in an analogous manner to the Fourier basic method already described in (1.2.1), through a polar representation (r, θ) of the frequency space $(u,\ v)$.

Let g and f two gray-level images related by an homothetic 4-parameter geometric transformation that maps each point (x_g, y_g) in g to a corresponding point (x_f, y_f) in f according to the matrix equation

$$\begin{bmatrix} x_f \\ y_f \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha\cos\beta & \alpha\sin\beta & -\delta x \\ -\alpha\sin\beta & \alpha\cos\beta & -\delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_g \\ y_g \\ 1 \end{bmatrix}, \tag{3.3}$$

where δx and δy are the translation vector coordinates, α is the uniform scale factor, and β is the rotation angle. Equivalently, for any pixel (x, y) we have:

$$g(x, y) = f(\alpha(x\cos\beta + y\sin\beta) - \delta x, \alpha(-x\sin\beta + y\cos\beta) - \delta y). \tag{3.4}$$

By assuming that the translation is null and reparamterizing the spatial coordinates with (r, θ), the relationship between f and g in form of the polar coordinate system can be written as

$$g(r, \theta) = f(\alpha r, \theta + \beta). \tag{3.5}$$

The AFMT (Ghorbel (1994)) of f and g can be written in this way:

$$\mathcal{M}_f(q, s) = \int_{r=0}^{\infty} \int_{\theta=0}^{2\pi} f(r, \theta) r^{\sigma - is} e^{-iq\theta}\, \mathrm{d}\theta\, \frac{\mathrm{d}r}{r} \tag{3.6}$$

$$\mathcal{M}_g(q, s) = \int_{r=0}^{\infty} \int_{\theta=0}^{2\pi} f(\alpha r, \theta + \beta) r^{\sigma - is} e^{-iq\theta}\, \mathrm{d}\theta\, \frac{\mathrm{d}r}{r} \tag{3.7}$$

where $(q, s) \in \mathbb{Z} \times \mathbb{R}$ and $\sigma > 0$ is a parameter necessary for the existence of the AFMT in the case of finite domain images. Performing a simple change of the variables r and θ in (3.6), (3.7) yields the following relation:

$$\mathcal{M}_g(q,s) = \alpha^{-\sigma+is} e^{iq\beta} \mathcal{M}_f(q,s). \tag{3.8}$$

Equation (3.8) makes the AFMT appropriate for extracting features that are invariant to scale and rotation changes. In particular by considering the norm of $\mathcal{M}_g(q,s)$.

When applying the AFMT on images, we lose the translation parameters and the phase information. The complementary method SPOMF (Symmetric Phase Only-Matched Filtering) provides the rotation and homothety information. This can be followed by a phase correlation between the original images (Chen et al. (1994); Reddy and Chatterji (1996)).

Note that similarly to the phase correlation method, this method is used in practice to estimate global transformations because it uses the contribution of the whole frequency spectrum (or at least a large part of it) for estimating the transformation parameters.

1.3 Differential methods

The differential methods for motion estimation are based on the spatial and temporal gradients of light intensity of the pixels. The principle of these methods is based on the assumption of conservation of the pixel intensity along the motion path. This conservation hypothesis can be written as:

$$\frac{df(\mathbf{s}(t),t)}{dt} = 0 \tag{3.9}$$

where $\mathbf{s}(t)$ is the pixel position.

The pattern of apparent motion of objects is also called Optical Flow; it is used for motion detection and shares some similarity with the measurement of stereoscopic disparity. It relies on the texture movement of objects and their edges. The optical flow allows us to approximately calculate motion vectors using the intensity variation of the image over time. Up to the aperture problem (which is linked to partial knowledge of objects in motion Wu (2001)), optical flow is equivalent to the image motion if the following conditions are respected:

- It is assumed that the light intensity of an image is preserved from an image to another.
- The scene may not include reflective surfaces.
- There is only one light source.

In the case of real images, these conditions may not be strictly respected. However, they are assumed locally verified.

Several methods have been proposed in this context. Fleet and Weiss (2006) presented a comparison between different optical flow methods including differential methods, which are based on local Taylor series approximations to the image signal.

1.3.1 Gradient based Methods

The assumption of luminance conservation means that variations of temporal luminance are due to the motion. According to this hypothesis, we can conclude that the intensity of each pixel is shifted from an image to another.

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{3.10}$$

This equation can be written in vector form:

$$I(\vec{x}, t) = I(\vec{x} + \vec{d}, t + \Delta t) \tag{3.11}$$

where I is the pixel intensity which is a function of position $\vec{x} = (x, y)$, time t and the displacement vector $\vec{d} = (\Delta x, \Delta y)$.

Assuming the movement to be small, the image constraint at $I(x, y, t)$ can be simplified through a Taylor expansion yielding the following approximation:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t \tag{3.12}$$

Since the luminance intensity is kept, the gradient constraint equation at $I(x, y, t)$ can be given by:

$$\nabla I(\vec{x}, t).\vec{u} + I_t(\vec{x}, t) = 0 \tag{3.13}$$

where $\vec{u} = (u_1, u_2) = (\frac{\Delta x}{\Delta t}, \frac{\Delta y}{\Delta t})^T$ is the optical flow (or velocity vector), $\nabla I = (I_x, I_y)$ and I_t represent the spatial and temporal derivatives of the image I.

Equation (3.13) has two unknowns (the components u_1 and u_2 of $\vec{u}$) and cannot be solved as such. This is known as the aperture problem of the optical flow algorithms.
To solve it, we can minimize the error

$$E(\vec{u}) = \sum_{\vec{x}} [\vec{u}.\nabla I(\vec{x}, t) + I_t(\vec{x}, t)]^2, \tag{3.14}$$

where $\vec{u}$ that minimizes $E(\vec{u})$ is the estimated optical flow.

To find this optical flow (here assumed to be constant over the image), we must set the derivatives of $E(\vec{u})$ to zero with respect to u_1 and u_2:

$$\frac{dE(u_1, u_2)}{du_1} = 0 \tag{3.15}$$

$$\frac{dE(u_1, u_2)}{du_2} = 0 \tag{3.16}$$

and we can re-write these two equations in matrix form:

$$M.\vec{u} = \vec{b} \tag{3.17}$$

where $M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$ and $\vec{b} = -\begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix}$.

Using over-determined Least Squares, the estimated optical flow can thus be written as follows:

$$\hat{u} = (M^\top M)^\dagger M^\top \vec{b}, \tag{3.18}$$

where † is the pseudo-inverse operator.

1.3.2 Iterative Methods

It is desirable to improve the gradient method for more accurate results. Iterative methods minimize the difference between two successive images; they repeat the following steps:

- estimate the motion vector between two images,
- register one image toward the other,

i.e. they use the current estimate to undo the motion, and then reapply the estimator to the warped images to find the residual motion. The method converges when the residual flow no longer changes. Let

$$\vec{u} = \vec{u}^0 + \delta\vec{u}, \tag{3.19}$$

where $\vec{u}^0$ is the first estimated optical flow. We can then create a warped image sequence I^0 from $\vec{u}^0$ and the motion constraint can be reexpressed as

$$I(\vec{x}, t) = I^0(\vec{x} + \delta\vec{u}, t + \mathit{\Delta} t) \tag{3.20}$$

where we will assume that $\mathit{\Delta} t$ is normalized to 1. The increment $\overline{\delta} u$ is calculated by using

$$\delta\hat{u} = M^{-1}\vec{b} \tag{3.21}$$

where M and b are calculated using the spatial and temporal derivatives of I^0. We find then the re-estimated optical flow $\vec{u}^1$

$$\vec{u}^1 = \vec{u}^0 + \delta\hat{u} \tag{3.22}$$

This new estimated optical flow will be used to warp the original image sequence such as in (3.20), and another residual flow can be estimated.

Iteratively, this new residual flow estimate is used to rewarp the original sequence, and another residual flow can be estimated. At iteration j, given the estimate u^j and the warped sequence I^j, we obtain a new objective error function

$$\begin{aligned} E(\delta\vec{u}) &= \sum_{\vec{x}} w(\vec{x})[I(\vec{x},t) - I(\vec{x}+\vec{u}^j+\delta\vec{u},t+1)]^2 \\ &= \sum_{\vec{x}} w(\vec{x})[I(\vec{x},t) - I^j(\vec{x}+\delta\vec{u},t+1)]^2 \\ &\approx \sum_{\vec{x}} w(\vec{x})[\nabla I^j(\vec{x},t+1).\delta\vec{u} + I^j(\vec{x},t+1) - I(\vec{x},t)]^2 \equiv \tilde{E}(\delta\vec{u}), \end{aligned} \tag{3.23}$$

where w is some nonnegative spatial weighting function.

We repeat these operations to obtain a minimal value for the residual flow, and we expect the optical flow to converge to an optimal value.

1.3.3 Lucas & Kanade

We present here the differential method developed by Lucas and Kanade (1981). It assumes that the flow is essentially constant in a local neighborhood $N_{\vec{x}}$ of a pixel under consideration at the position $\vec{x}$ and resolves the optical flow equation for all the pixels in that neighborhood using a weighted Least Squares method:

$$E(\vec{u}) = \sum_{\vec{y}\in N_{\vec{x}}} w(\vec{y}-\vec{x},t)[\vec{u}.\nabla I(\vec{y},t) + I_t(\vec{y},t)]^2, \tag{3.24}$$

and $w(\cdot,t)$ is commonly a Gaussian weighting function in order to weight constraints in the center of the neighborhood more highly, giving them more influence.

Minimizing (3.24) can be performed by a standard weighted least squares approach, which leads to expressions similar to (3.17).

1.3.4 Horn & Shunck

This method was developed by Horn and Schunck (1981). It represents the flow as a global energy functional. It attempts to solve the aperture problem by minimizing the functional below, which is based on the optical flow equation:

$$E = \iint [(I_x u_1 + I_y u_2 + I_t)^2 + \alpha^2(||\nabla u_1||^2 + ||\nabla u_2||^2)]\mathrm{d}x\mathrm{d}y \tag{3.25}$$

where I_x, I_y and I_t the derivatives with respect to x, y and t, u_1 and u_2 are the components of optical flow vector, and α is a positive regularization constant which controls the flow smoothness.

Although this method yields a dense vector field, it is more sensitive to noise than local methods and small movements may not be properly identified.

In the next section, we elaborate another dense optical flow estimation which was introduced by Farnebäck (2003), and which can be more accurate.

1.3.5 Farnebäck

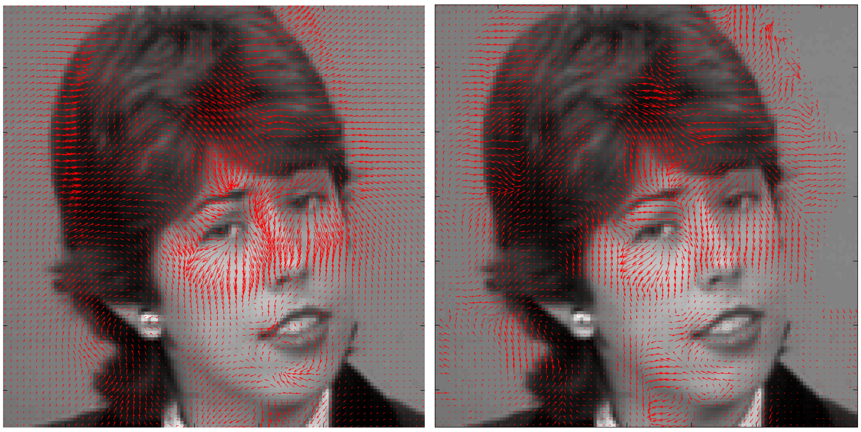
Gunnar Farnebäck proposed a dense optical flow method which finds motion vectors for all pixels in a target image, with respect to a reference image. It is based on a quadratic polynomial expansion of an image, e.g. approximating some neighborhood of each image pixel with a polynomial. He considers a multi-scale displacement estimation approach. It follows an iterative strategy to use estimated displacements in an iteration as a priori displacements in the next iteration.

Large displacements are taken into account while using multiple scales. It starts at a coarse scale to get a rough but reasonable displacement estimate and propagates this through finer scales to obtain increasingly more accurate estimates.

The next section shows some tests on optical flow estimation using the three detailed methods.

1.3.6 Experimental tests

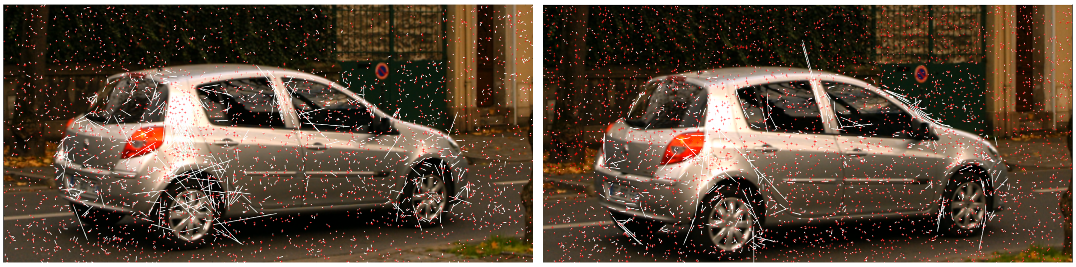
The already detailed optical flow methods are tested on two different real sequences, one containing small movements in Figure 3.2, and another with larger motions in Figure 3.3. Figure 3.2 shows that classical methods by Horn&Shunck and Lucas&Kanade are successful when illumination is stable between two images, and Horn&Shunck method presents a slight advantage concerning the provided dense vector field.



(a) Horn & Shunck tracking (b) Lucas & Kanade tracking

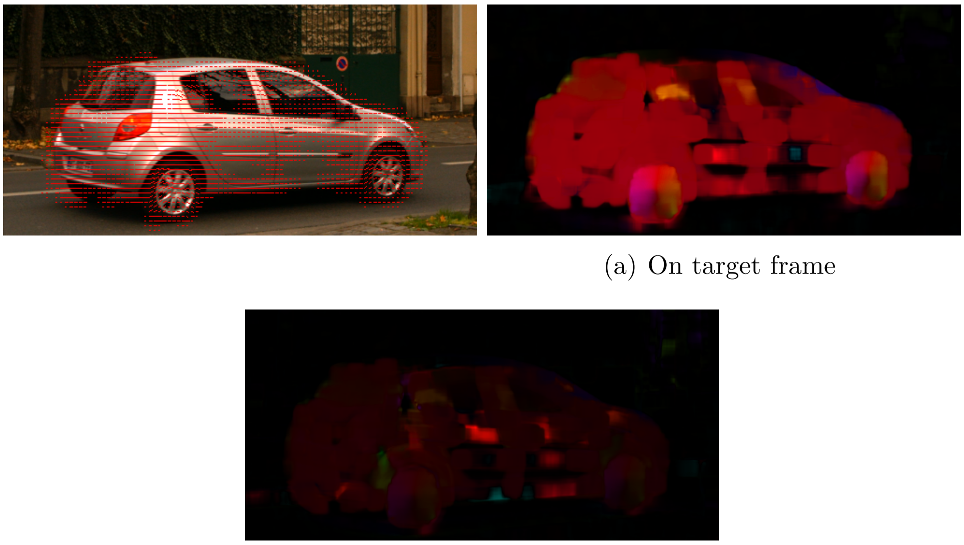
Figure 3.2. Classical variational methods for motion estimation between two images with illumination constancy.

Figure 3.3 shows that the assumption of luminance conservation has limited the efficiency of motion tracking when brightness is varying, and the presence of saturation artifacts makes the optical flow estimation more complicated in both Horn&Shunck and Lucas&Kanade methods. Finally, Figure 3.4 proves that Farnebäck's method significantly provides a better motion estimation, and with higher accuracy with respect to the previous methods. The second row images present the motion field in HSV space, and show that the estimation accuracy of magnitude and direction of motion is decreasing when flicker is affecting the target image, and also in the presence of saturation artifacts.



(a) Horn & Shunck tracking (b) Lucas & Kanade tracking

Figure 3.3. Classical variational methods for motion estimation between two images acquired from a naturally lit scene with a slight additionnal arficial flicker



(a) On target frame

(b) On target frame + synthetic flicker

Figure 3.4. Farnebäck optical flow method: the first row corresponds to the plots of some samples of motion vectors. Images in second row show the motion vectors in HSV space, and present the direction (corresponding the hue value) and magnitude (corresponding to the lightness value) on motion vectors.

1.4 Block matching based methods

Block matching methods are among the most used methods for motion estimation. They are often employed in video compression standards such as H.261 (Seferidis and Ghanbari (1994)), MPEG-1,2 or 4 (Noguchi et al. (1999)). The basic principle of block matching methods is to regularly divide the reference image into blocks of pixels, also called regions of interest (ROI). For each of these blocks, a search area is defined in the target image, the most ressembling ROI in the search area is retained as best candidate, depending on a given criterion.

Several criteria of similarity are used to find the best candidate block in a search area. These criteria can be expressed either in the image plane (directly using the pixels intensity) or in a frequency domain representation (in the Fourier transform).

For each block $\mathcal{B}_k$ (k designates the block label in a regular grid) of size $w \times h$ and centered on the pixel of coordinates $\mathbf{s} = (x, y)$ in the current image, the displacement shift between this block and the candidate block in the reference frame is noted $\mathbf{d_k} = (d_x, d_y)$, and the best candidate yields the estimated motion vector $\hat{\mathbf{d}}_\mathbf{k} = (\hat{d}_x, \hat{d}_y)$.

Matching in the image plane

Disparity measurements For a given block B_k in frame $f(\cdot, t)$, the best block candidate corresponds to the optimal geometrical transform parameters $\mathbf{d}_{k,t}$ that minimize the following energy functions:

- Sum of absolute differences:

$$\text{SAD}(\mathbf{d}_{k,t}) = \sum_{\mathbf{s}\in\mathcal{B}_k} |f(\mathbf{s}, t) - f(\mathbf{s} - \mathbf{d}_{k,t}, t_{ref})| \tag{3.26}$$

$$\hat{\mathbf{d}}_{k,t} = \underset{\mathbf{d}_{k,t}}{\text{argmin}} \; (\text{SAD}(\mathbf{d}_{k,t}))$$

- Sum of squared differences:

$$\text{SSD}(\mathbf{d}_{k,t}) = \sum_{\mathbf{s}\in\mathcal{B}_k} (f(\mathbf{s}, t) - f(\mathbf{s} - \mathbf{d}_{k,t}, t_{ref}))^2 \tag{3.27}$$

$$\hat{\mathbf{d}}_{k,t} = \underset{\mathbf{d}_{k,t}}{\text{argmin}} \; (\text{SSD}(\mathbf{d}_{k,t})).$$

These two displacement measures do not take into account the luminosity and colorimetric variations between images. Thus, the widely used SAD and SSD are modified by Giachetti (2000) to consider the effect of global gray-level variations, setting the average gray level difference equal to 0 (ZSSD, ZSAD):

$$\text{ZSAD}(\mathbf{d}_{k,t}) = \sum_{\mathbf{s}\in\mathcal{B}_k} |f(\mathbf{s}, t) - \overline{\mathcal{B}_k}(t) - f(\mathbf{s} - \mathbf{d}_{k,t}, t_{ref}) - \overline{\mathcal{B}'_k}(t_{ref})| \tag{3.28}$$

$$\hat{\mathbf{d}}_{k,t} = \underset{\mathbf{d}_{k,t}}{\text{argmin}} \; (\text{ZSAD}(\mathbf{d}_{k,t}))$$

$$\text{ZSSD}(\mathbf{d}_{k,t}) = \sum_{\mathbf{s}\in\mathcal{B}_k} (f(\mathbf{s}, t) - \overline{\mathcal{B}_k}(t) - f(\mathbf{s} - \mathbf{d}_{k,t}, t_{ref}) - \overline{\mathcal{B}'_k}(t_{ref}))^2 \tag{3.29}$$

$$\hat{\mathbf{d}}_{k,t} = \underset{\mathbf{d}_{k,t}}{\text{argmin}} \ (\text{ZSSD}(\mathbf{d}_{k,t}))$$

where $\overline{\mathcal{B}_k}(t)$ and $\overline{\mathcal{B}'_k}(t_{ref})$ are the pixel average values of the block $\mathcal{B}_k(t)$ in the current image and the candidate block $\mathcal{B}'_k(t_{ref})$ in the reference image respectively.

Similarity measurements

- Cross-Correlation criteria (CC):

  In signal processing, the cross-correlation, is a dependency measure between two signals. This tool can also be employed for images. A cross-correlation approach looks for $f(\mathbf{s} - \mathbf{d}_{k,t}, t_{ref})$ maximizing the correlation with $f(\mathbf{s}, t)$. This technique is often used under the principle of block matching. This method has been adapted for some special cases, citing for example the discontinuous movements, loss of information in images, etc, but it was widely applied on images in grayscale, and with normal brightness conditions:

$$CC(\mathbf{d}_{k,t}) = \sum_{\mathbf{s}\in\mathcal{B}_k} [f(\mathbf{s},t) - \overline{\mathcal{B}_k}(t)][f(\mathbf{s} - \mathbf{d}_{k,t}, t_{ref}) - \overline{\mathcal{B}'_k}(t_{ref})] \qquad (3.30)$$

$$\hat{\mathbf{d}}_{k,t} = \underset{\mathbf{d}_{k,t}}{\text{argmin}} \ (CC(\mathbf{d}_{k,t}))$$

- Normalized Cross-Correlation criteria (NCC) that ranges between −1 (no dependency) and 1 (full dependency) can be used alternatively:

$$NCC(\mathbf{d}_{k,t}) = 1 - \frac{\sum_{\mathbf{s}\in\mathcal{B}_k} [f(\mathbf{s},t) - \overline{\mathcal{B}_k}(t)][f(\mathbf{s} - \mathbf{d}_{k,t}, t_{ref}) - \overline{\mathcal{B}'_k}(t_{ref})]}{\sqrt{\sum_{\mathbf{s}\in\mathcal{B}_k} [f(\mathbf{s},t) - \overline{\mathcal{B}_k}(t)]^2 \sum_{\mathbf{s}\in\mathcal{B}_k} [f(\mathbf{s} - \mathbf{d}_{k,t}, t_{ref}) - \overline{\mathcal{B}'_k}(t_{ref})]^2}} \qquad (3.31)$$

$$\hat{\mathbf{d}}_{k,t} = \underset{\mathbf{d}_{k,t}}{\text{argmin}} \ (NCC(\mathbf{d}_{k,t})).$$

Statistical measurements

Mutual information is a measure of statistical dependence between two random variables, it is often measured in bits. The mutual information is zero if the variables are independent, and increases with dependency.

The mutual information $I(X,Y)$ measures the average amount of information provided by an occurence of X on the occurence of Y.

Considering that a probability distribution represents our knowledge on a random phenomenon, we measure the lack of information by the entropy of this distribution.

The mutual information is expressed via:

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \tag{3.32}$$

where $H(X)$ et $H(Y)$ are the respective entropies of X and Y, and $H(X,Y)$ is the joint entropy between X and Y.

$H(X)$ is defined as

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i) \tag{3.33}$$

where $x_1, \ldots, x_n$ are the pixel intensities in each bloc, and $P(x_i)$ are their probabilities.

The joint entropy is defined as follows:

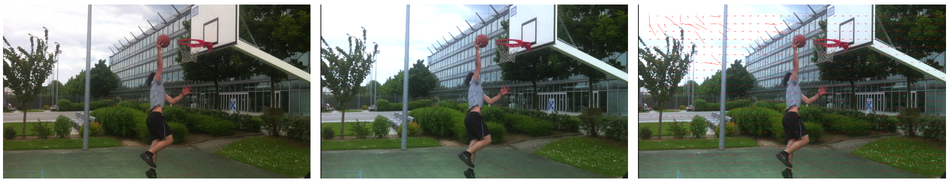
$$H(X,Y) = -\sum_{i,j} P(X = x_i, Y = y_j) \log_2 P(X = x_i, Y = y_j) \tag{3.34}$$

Based on the block matching algorithm, we look for similar blocks by maximizing the mutual information indicating a maximum of dependence between them. Since the mutual information depends on the probability of occurence of pixel intensities between images, we act on grayscale images because it involves simpler computation. According to the mutual information expression, X and Y can correspond to the block intensities in the current frame and those of the candidate block in the reference frame respectively.

The entropies $H(X)$ and $H(Y)$ can be found by computing the histogram that tabulates the total number of pixels observed for every possible intensity found in the block. This histogram is normalized by the total number of pixels to get the probability distribution function.

To find the joint entropy $H(X,Y)$, we need to calculate the joint two dimensional histogram between the two images. The joint histogram is essentially the same as a normal 1D histogram but the first dimension logs intensities for the first image and the second dimension logs intensities for the second image. At location (i, j) in the joint histogram, it provides how many intensity values we have encountered that have intensity x_i in the first image and intensity x_j in the second image.

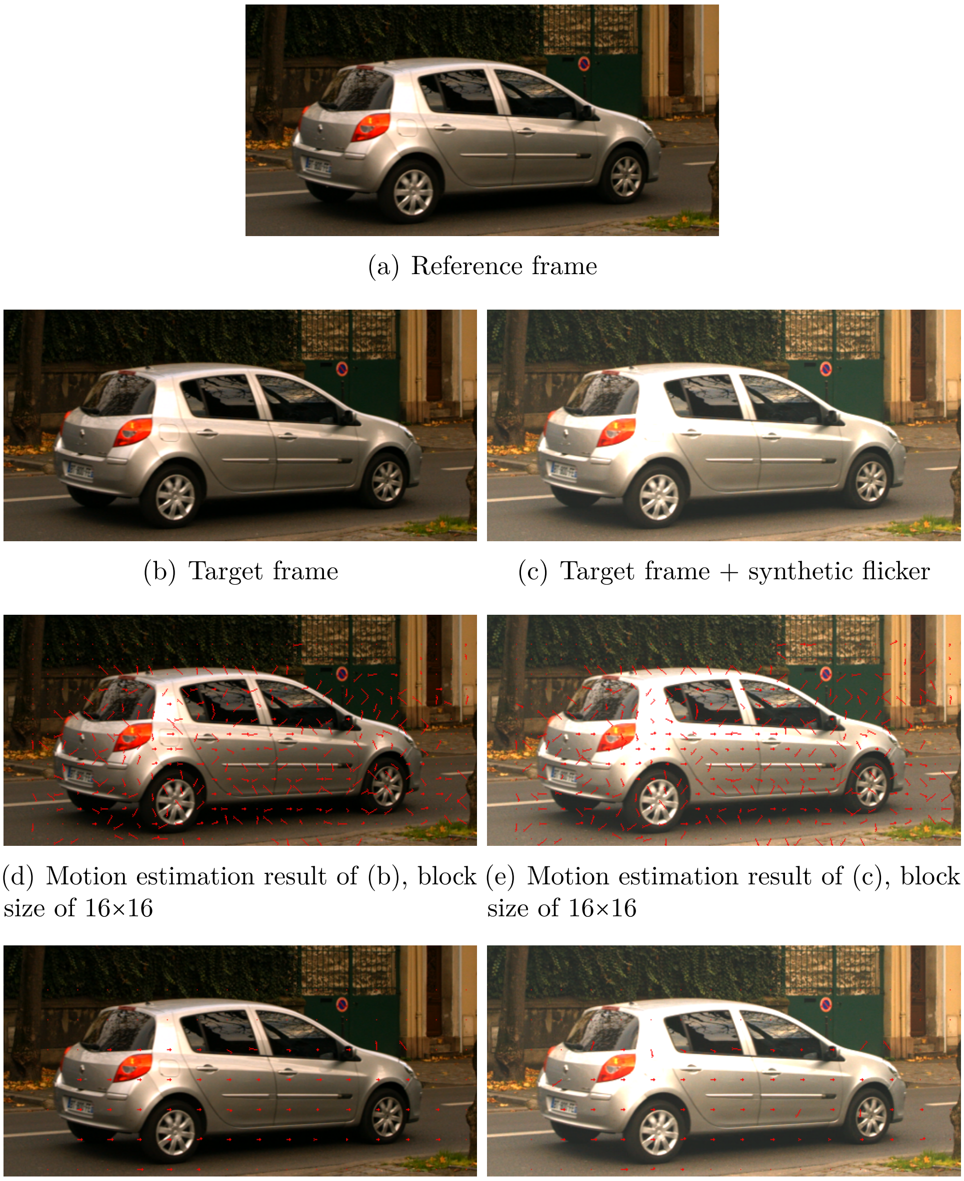
We tested this method on two naturally lit real sequences affected with synthetic flicker. The first sequence in Figure 3.5 shows that mutual information criteria is almost effective for motion estimation but it fails with saturation artifacts and some redundant similar regions, for example, the building windows.



(a) Reference frame (b) Target frame: shifted reference frame with artificial flicker (c) Optical flow result

Figure 3.5. Block matching using the mutual information criterion, (b) is the reference frame manually shifted for few pixels, (c) shows motion vectors of blocks from the reference frame to the target frame.

In addition, this similarity measure is globaly effective even if illumination is not conserved between the two images except for homogenous, saturation areas and some over-textured areas, and as Figure 3.6 shows, mutual information requires a large size of blocks, in order to obtain more information on intensity distribution, and to be able to match blocks more accurately between two images. We can see that motion estimation results using 32 × 32 blocks (3.6-d) are significantly more accurate than those of 16 × 16 blocks (3.6-f). Moreover, adding artificial flicker on the target image also affects the precision of motion estimation (3.6-g).



(a) Reference frame

(b) Target frame

(c) Target frame + synthetic flicker

(d) Motion estimation result of (b), block size of 16×16

(e) Motion estimation result of (c), block size of 16×16

(f) Motion estimation result of (b) with larger block size 32×32

(g) Motion estimation result of (c) with larger block size 32×32

Figure 3.6. Test on the car sequence: the horizontal motion of the car is well estimated, but it fails on some homogenous regions. It is recommended to increase the block size in order to include more texture information.

2 Color correction approaches

The main target of color correction algorithms is to transform the color distribution in a source image to a target image. In the context of high speed videos under artificial lightings, the source image corresponds to the selected reference image having the highest luminosity average in a flicker period, and the target image corresponds to another period frame whose color is to be corrected.

With the growing demand of high definition images and videos, the field of color correction has become of increasing importance in the computer vision and video processing communities.

Color correction approaches are widely studied in the context of multi-view imaging, which consists of correcting the color differences between neighboring views due to different exposure levels and view angles. Xu and Mulligan provided an interesting comparative study for the performance evaluation of color correction approaches in the context of automatic multi-view image and video stitching. Color correction approaches are classified into two categories: parametric (model-based) and non-parametric.

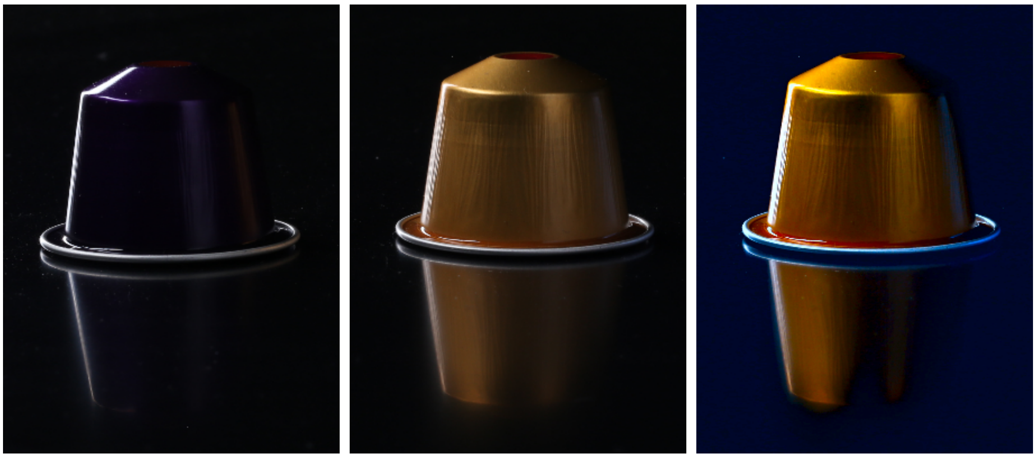
2.1 Model-based techniques

2.1.1 Global color transfer

Model based techniques for color correction includes global and local approaches. Global color transfer methods assume a global relation between the whole source and target images, and use a 3×3 color correction matrix which is almost diagonal, but can be an affine transformation matrix or arbitrary matrix. Funt and Lewis (2000) compared between affine and diagonal-based matrices for color correction. Tian et al. (2002) provided a colour histogram based colour correction, the process consists of matching the histograms of colours in the overlapping region of panoramic images. The simplest global approach was provided by Reinhard et al. (2001), it transforms color space between the source and target image with a unique color transfer matrix, authors demonstrated that by utilizing the $L^*a^*b^*$ color space and the mean and standard deviation of each L^*, a^*, and b^* channel, respectively, that the color can be transferred between two images.

Ruderman et al. (1998) worked on a decorrelated color space $l\alpha\beta$ instead of the RGB color space, where l is the luminance channel and $\alpha\beta$ are the chromatic channels. In order to transfer colors to grayscale images, Welsh et al. (2002) used this approach to colorize grayscale images, matched the luminance and texture information between a source and target image, and retained the original luminance values while transfering only the chromatic information to the target image.

However, most global color transfer methods do not take into account local color distributions in different regions of image.



(a) Source image (b) Target image (c) Destination image

Figure 3.7. Global color transfer example: Basing on the paper of Reinhard et al. (2001), this method requires only the mean and standard deviation of pixel intensities for each channel in the $L^*a^*b^*$ color space. The transfer fails to resynthesise the colour scheme of the target image.

2.1.2 Local color transfer

When processing an image with different color regions, global color transfer cannot distinguish the color statistics of the regions. Thus, the color transfer techniques with complex contents is still a problem that needs further study. Some local methods are proposed to provide a better mapping between the source and the target images. Unlike global-based methods, Tai et al. (2005) used probabilistic segmentation and region mapping for color transfer, and takes local color information into consideration. Reinhard's method is used to perform the color transfer on the matched local regions. Xiang et al. (2009) adapted the latter method to multiple source images.

2.2 Non-parametric based techniques

Unlike the parametric color transfer techniques, the non-parametric methods do not use a particular parametric mapping function for color transfer, and most of these methods use a look-up table to estimate color mapping between images. This look-up table is usually computed from 2D joint histogram of image feature correspondences or pixel pairs in the overlapped area of two images. Most existing non-parametric approaches perform a matching step between the source image and the target image, because data are often sensitive to outliers, noise and saturation effects, due to the different view angles, climate change, different lighting conditions and reflection properties of scene objects. Thus, matching is required to estimate mapping function from histogram.

Yamamoto and Oi (2008) use scale invariant feature transform (SIFT) to match two neighboring views in a multi-view camera network, use a diagonal weight matrix to treat RGB channels independently, calculate lookup tables with a non-linear energy-minimization approach and use these tables to transfer colors between images.

Jia and Tang (2003) globally and locally adjust luminance of two overlapping images. Authors proposed a two-step approach to maintain monotonicity. First, a denoising and filling the data gaps are performed using 2D tensor voting, which produces an initial estimate of the mapping function. In the second stage, an iterative local alignment scheme is proposed to adjust the initial estimate. The estimation of the replacement mapping functions using the tensor voting provides a fast convergence of the iterative scheme.

Fecker et al. (2008) suggested the use of cumulative color histogram to compensate the differences of luminance and chrominance between different camera views. A histogram matching between the distorted and the reference sequences is performed to estimate the mapping function for color correction. To avoid some visual artifacts on the luminance component, the authors also proposed a post-processing step consisting on the correction of the first and the last histogram bins.

Pitié et al. (2005, 2007) proposed an original non-linear, iterative approach for the color distribution transfer. It presents the distribution of colors of an image in a high dimensional space, and authors project this distribution into random 1D marginal distributions. They repeatedly map these new distributions of the target image to those of the source image in order to transfer the color distribution of the source image into the target image. They also proposed a post-processing method for grading colors between various images.

3 An outline

This chapter presented an overview of existing motion estimation and color correction techniques for the image sequences in several different contexts. However, classical motion estimation is based on the assumption of preservation of brightness over time, especially in the conventional methods of optical flow, which implies that the use of these methods in the presence of light and chromatic variations is not useful. Another statistical approach basing on the mutual information criterion was studied but its lack of robustness to saturation artifacts and redundant information is proved. In addition, the studied block matching based methods do not take into account colorimetric variations between images, and deal only with illumination variations. In the next chapters, we are inspired by this wide state of the art in order to find a solution for the flicker removal problem in the context of high speed imaging.

Chapter 4

Global methods for color correction

In video processing, it is necessary to take into account the constraints of processing time, especially as most video applications deliver a large amount of data, which needs to be processed in a timely manner. Some applications may require close to real-time processing. This functionality requires to use efficient algorithms.

For video color correction, a reasonable approach is to consider a global, frame-based correction, especially if the lighting conditions and motion are not very complex.

The purpose of this chapter is to present two different approaches for global video color correction. The first is based on histogram computations. The second is based on feature tracking and image registration.

1 Histogram matching based method

Most videos contain background and object motion, but in this chapter, we assume that motion is (a) global between frames in high speed videos (e.g. due to the camera motion) and (b) small compared with the frame size, and so the contents of two contiguous frames does not vary significantly. In this way, matching pixels may not be necessary and estimating the motion between frames may not be needed.

Color features are among the most important features used in image database retrieval. Techniques for comparing color distributions are fundamental in pattern recognition, image retrieval and image restoration. This is due to a compact representation and low complexity in measuring color similarity of images.

We propose to compare the global distribution of colors between a target image and a previous reference frame in order to compute the corresponding color transformation matrix for all image pixels. Matching regions are required to estimate the most appropriate color transform between images.

For this purpose, we use an histogram matching technique, a well known method in image processing to find similarities between digital images, acquired at the same global illumination. Our approach involves applying a global brightness transformation to all pixels of the target image, and thus obtaining a new image from the same operation on each pixel. This transformation is built from the cumulative histograms of the target and reference images. The transformation takes into account the intensity variations of color channels separately, so colorimetric variations are handled correctly by this process.

In the next paragraph, we briefly recall some specifications about histograms in image processing.

1.1 Histogram specifications

In image processing, the histogram is a fundamental tool that represents the distribution of intensities (or colors) of the image. Histograms are often used to adjust intensities (e.g. contrast, brightness etc) in images. The histogram is defined as a discrete function that associates to each intensity value the number of pixels that have this value. The determination of the histogram is carried out in a single pass on the image by counting the number of pixels for each image intensity. Optionally, the histogram may be normalized, by dividing the values of each class by the total number of pixels of the image. Then, the value of a class ranges between 0 and 1, and can be interpreted as an estimate of its probability of occurrence in the image. The normalized histogram is defined by:

$$H(x_k) = p_x(x_k) = \frac{n_k}{n}, \quad 0 \leq k < L \tag{4.1}$$

where n is the total number of pixels, n_k is the occurence number of the intensity x_k and L is the number of intensity levels.

We also recall that the normalized cumulative histogram can be defined by:

$$\widehat{H}(x_k) = \sum_{j=0}^{k} p_x(x_j) \tag{4.2}$$

The following section describes the methodology of flicker enhancement using different approaches of histogram matching.

1.2 Methodology

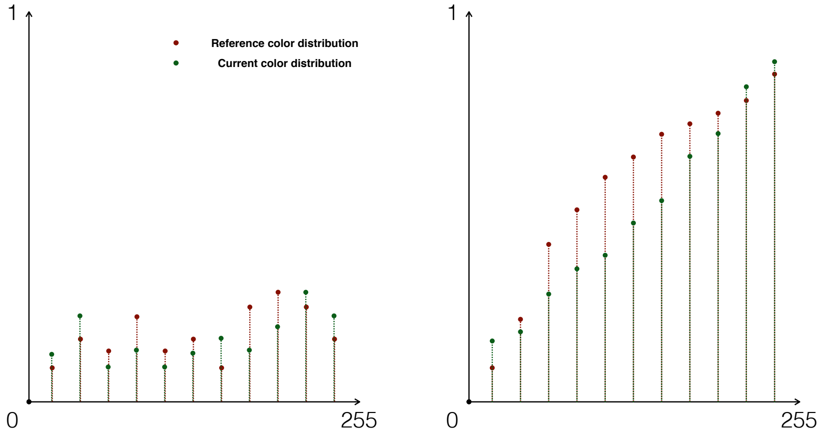
1.2.1 1st approach: Illumination is enhanced separately

In this section we assume that each color channel can be processed separately, i.e. that the color correction matrix $\boldsymbol{M}$ is diagonal. This is essential to explain the histogram matching technique in a simpler case.

Each sub-sequent frame (in a flicker period sequence) can be processed independently with respect to its previous reference frame. First, we compute the normalised cumulative histograms $\widehat{H}^R_{\text{ref}}$, $\widehat{H}^G_{\text{ref}}$ and $\widehat{H}^B_{\text{ref}}$ of the RGB channels of the reference frame $f(\cdot, t_{\text{ref}})$ respectively. Similarly, for a frame $f(\cdot, t)$, we compute its corresponding normalised cumulative histograms, $\widehat{H}^R_t$, $\widehat{H}^G_t$ and $\widehat{H}^B_t$. Figure 4.2-a illustrates why discrete cumulative histograms can hardly be used to match different distributions. Indeed, while all discrete values in the input domain are associated with an output value, the reverse is not true. We will see later that we need to make this reverse association.

To correct for this problem, we use a nearest-neighbor interpolated version of the discrete histogram, which results in a linearly interpolated version of the cumulated histogram, as seen in Figure 4.2-b.

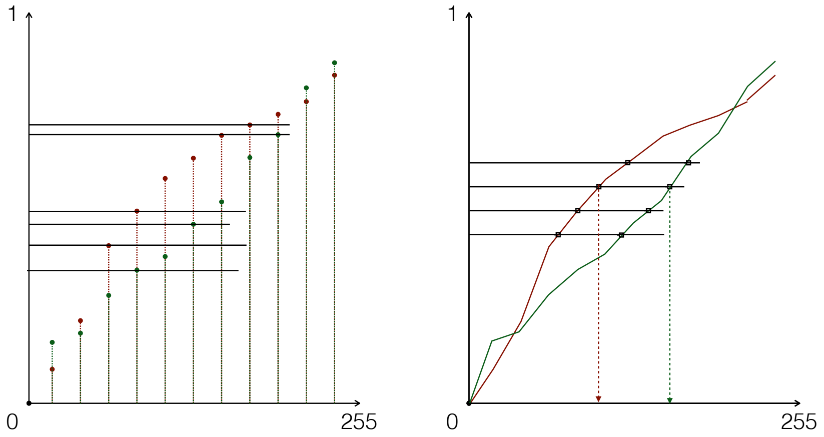
Because the interpolated histogram contains only positive values, the cumulated



(a) Discrete histogram

(b) Discrete cumulative histogram

Figure 4.1. An example of the computation of the discrete cumulative histograms to present the color distributions in two images.



(a) Discrete cumulative histogram matching

(b) Interpolated cumulative histogram matching

Figure 4.2. An example of the matching between different data sets in discrete and interpolated cumulative histograms

histogram is a monotonically increasing piecewise affine function. As a result, we can redefine the cumulative histogram function as an integral rather than a sum. We now show how using the cumulated histogram can be used for color matching.

So far we have assumed a linear relationship between color channels, i.e. for all pixels in the color channel c, the intensity l^c of the same point in the reference image $f(\cdot, t_{\text{ref}})$ and the target image $f(\cdot, t)$ is

$$l^c_{\text{ref}} = \alpha_c \, l^c_t \tag{4.3}$$

where α_c is the gain parameter between the two matched data sets of such color channel in the target and reference frames. Then, for each color channel c, and for each intensity level l^c_t, we can write:

$$H^c_{\text{ref}}(\alpha_{\text{c}} l^{\text{c}}_t) = H^c_t(l^{\text{c}}_t). \tag{4.4}$$

The normalized cumulative histogram H^{ct}_{cum} on l^{c}_t is then provided by:

$$\begin{aligned}\widehat{H}^c_t(l^{\text{c}}_t) &= \frac{\int_0^{l^{\text{c}}_t} H^c_t(u)\,\mathrm{d}u}{\int_0^{\max l^c_t} H^c_t(u)\,\mathrm{d}u} \\ &= \frac{\int_0^{l^{\text{c}}_t} H^c_{\text{ref}}(\alpha_{\text{c}} u)\,\mathrm{d}u}{\int_0^{\max l^c_t} H^c_t(u)\,\mathrm{d}u},\end{aligned} \tag{4.5}$$

where the denominator in (4.5) is the normalization factor and $\max l^c_t$ designates the maximum of grayscale level acquired in the image $f(\cdot, t)$. It can be assumed such that

$$\int_0^{\max l^c_t} H^c_t(u)\,\mathrm{d}u = \int_0^{\max l^c_t} H^c_{\text{ref}}(\alpha_c u)\,\mathrm{d}u, \tag{4.6}$$

which by performing the change of variable $v = \alpha_c u \Rightarrow \mathrm{d}v = \alpha_c \,\mathrm{d}u$, yields:

$$\int_0^{\max l^c_t} H^c_t(u)\,\mathrm{d}u = \frac{1}{\alpha_c} \int_0^{\max l^c_{\text{ref}}} H^c_{\text{ref}}(v)\,\mathrm{d}v. \tag{4.7}$$

Similarly, the same change of variable is used on the numerator term of (4.5), and this equation becomes:

$$\begin{aligned}\widehat{H}_t^c(l_t^c) &= \frac{\frac{1}{\alpha_c}\int_0^{\alpha_c l_t^c} H_{\text{ref}}^c(v)\,\mathrm{d}v}{\frac{1}{\alpha_c}\int_0^{\max l_{\text{ref}}^c} H_{\text{ref}}^c(v)\,\mathrm{d}v} \\ &= \widehat{H}_{\text{ref}}^c(\alpha_c l_t^c). \end{aligned} \tag{4.8}$$

Then, for all levels l^c in both target and reference, we have

$$\widehat{H}_t^c(l_t^c) = \widehat{H}_{\text{ref}}^c(l_{\text{ref}}^c) \tag{4.9}$$

where from any value in the output domain of the normalized cumulated histograms, we can associate two values l_t^c and l_{ref}^c, for which the relationship $l_{\text{ref}}^c = \alpha_c l_t^c$ should hold for all of them.

We can consider a number of such pair of points and estimate α_c by linear regression, as illustrated in Figure 4.3. We recall that c designates the color channels R, G and B. Then we can independently estimate the three enhancement

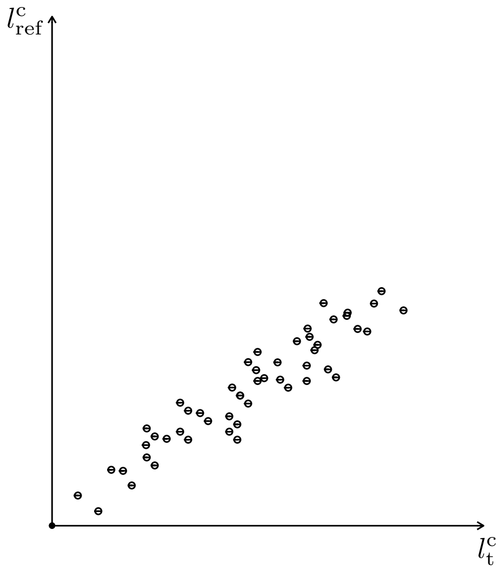


Figure 4.3. l_{ref}^c vs. l_t^c.

parameters for each color channel by minimizing the following criterion:

$$\underset{c\in(R,G,B)}{J_H(\widetilde{\alpha}_c)} = \sum_{i=0}^{S} \Phi(l^c_{\mathrm{ref}}(i) - \widetilde{\alpha}_c\, l^c_t(i)) \tag{4.10}$$

where i ranges between 0 and S the number of sampled values in the cumulated histogram and Φ is some cost function. In practice, the cumulated histograms may not be strictly increasing so an interval of image values may match a given output value. We specifically avoid these flat regions in the cumulated histograms. Also values near the extremities of the histograms may be affected by saturation, so we only use the central values (Figure 4.4).

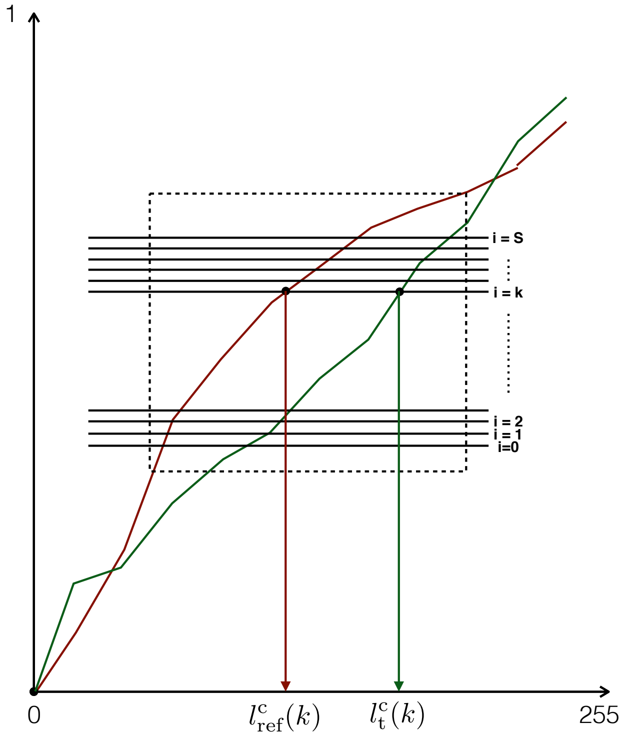


Figure 4.4. The choice of data sets samples from the central part of histograms.

If Φ is the quadratic functional, then minimizing J_H is very easy, it is simply the average of all the individual ratios:

$$\widehat{\alpha}_c = \frac{\sum_{i=0}^{S} l_{\text{ref}}^c(i) l_t^c(i)}{\sum_{i=0}^{S} l_t^c(i)^2} \tag{4.11}$$

The described method in the present section is limited, because it considers the three color channels separately. The next section provides a second approach that deals with inter-channel variations and enhances colors as well as illumination flicker in image sequences.

1.2.2 2nd approach: Inter-channel correlation

In order to better take into account chromatic variations in our correction approach, the cumulated histogram for the three color channels is used again to generate two 3D vectors (corresponding to the RGB values), L_{ref} and L_t, of level intensities that verify an extension of the linear relationship in (4.9):

$$L_{\text{ref}} = \widetilde{\boldsymbol{M}} L_t, \tag{4.12}$$

where $L_t = \begin{bmatrix} l_t^R \\ l_t^G \\ l_t^B \end{bmatrix}$, $L_{\text{ref}} = \begin{bmatrix} l_{\text{ref}}^R \\ l_{\text{ref}}^G \\ l_{\text{ref}}^B \end{bmatrix}$ and $\widetilde{\boldsymbol{M}}$ is the flicker correction matrix, containing three illumination parameters (diagonal elements) and six parameters for chroma correction:

$$\widetilde{\boldsymbol{M}} = \begin{bmatrix} \tilde{\alpha}_{r11} & \tilde{\alpha}_{r12} & \tilde{\alpha}_{r13} \\ \tilde{\alpha}_{g11} & \tilde{\alpha}_{g12} & \tilde{\alpha}_{g13} \\ \tilde{\alpha}_{b11} & \tilde{\alpha}_{b12} & \tilde{\alpha}_{b13} \end{bmatrix}. \tag{4.13}$$

Similarly, we can estimate the flicker enhancement parameters for the three color channels by minimizing the following criterion:

$$J_H(\widetilde{\boldsymbol{M}}) = \sum_{i=0}^{S} \Phi(L_{\text{ref}}(i) - \widetilde{\boldsymbol{M}} L_t(i)), \tag{4.14}$$

where Φ is now a function defined on $\mathbb{R}^3$.

For minimizing this new energy function J_H, a common choice of Φ is the quadratic functional where $\Phi = \|.\|^2$, and the estimated correction matrix is

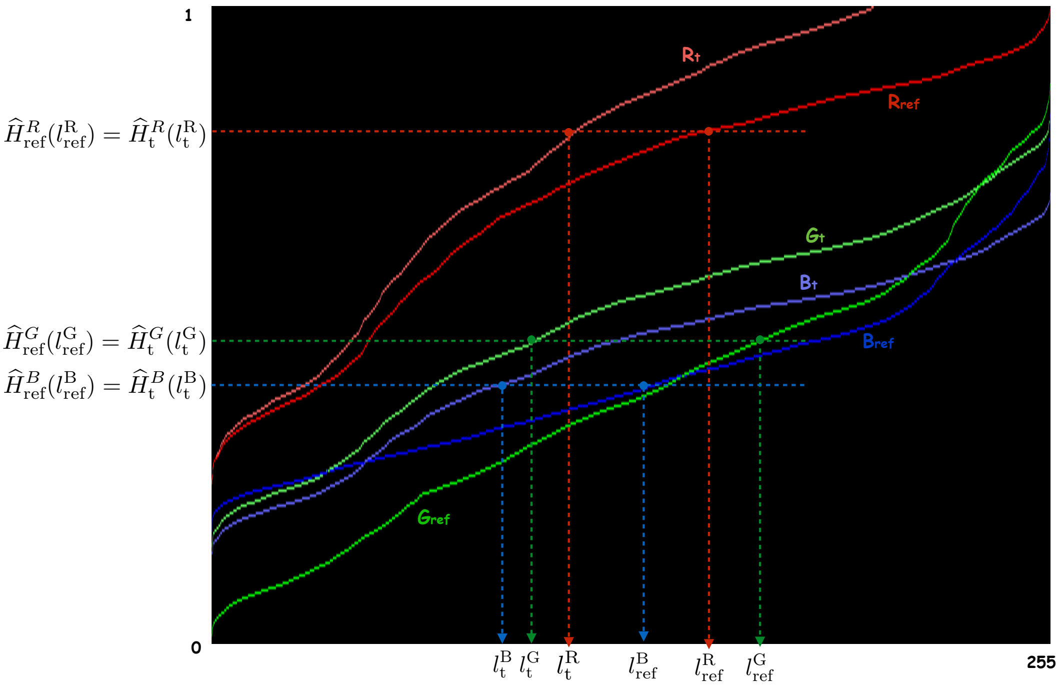


Figure 4.5. The normalized cumulative histograms representing the color distribution of three color channels in the reference and target frames $f(\cdot, t_{\mathrm{ref}})$ and $f(\cdot, t)$.

provided by the least square estimate given by

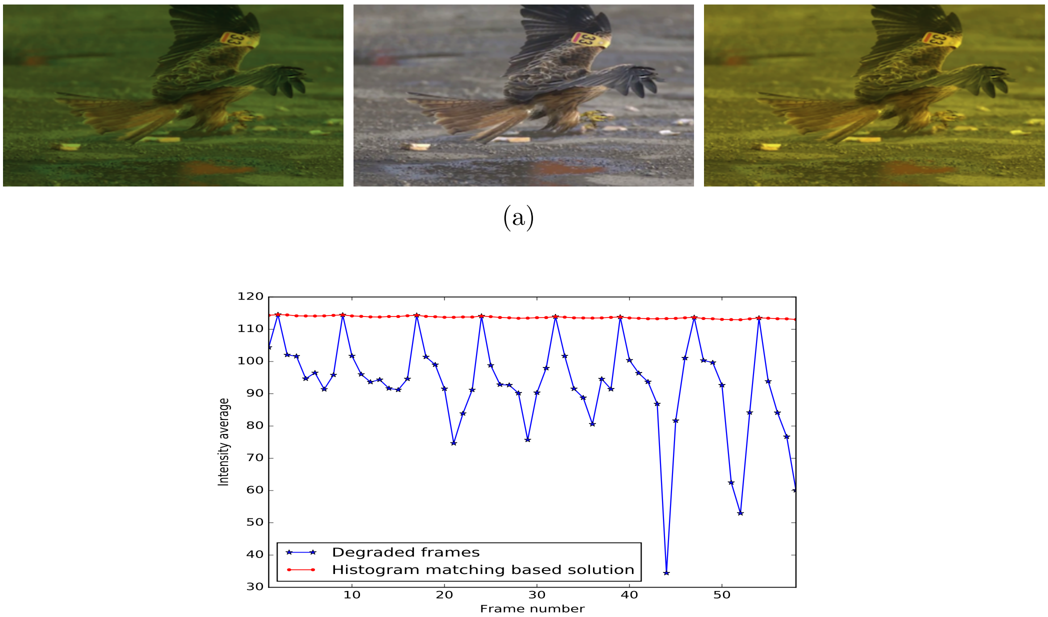
$$\widehat{M}_\alpha = \sum L_{\mathrm{ref}} L_t^\top \left(\sum L_t L_t^\top\right)^\dagger. \tag{4.15}$$

Now we are going to present some results on the histogram based approach for color correction.

1.3 Results and discussion

In this section, we present some results of flicker correction applied to various image sequences with artificial and real flicker. It is not possible to precisely evaluate the performance of the method by just visualizing the images before and after correction, because temporal variations are the main issue. Thus, in the section b of the figures below, we plot the intensity averages of images before and after correction, this shows that global artificial flicker is almost perfectly corrected (Figure 4.6-b,), as expected, even in the presence of large chromatic variations.

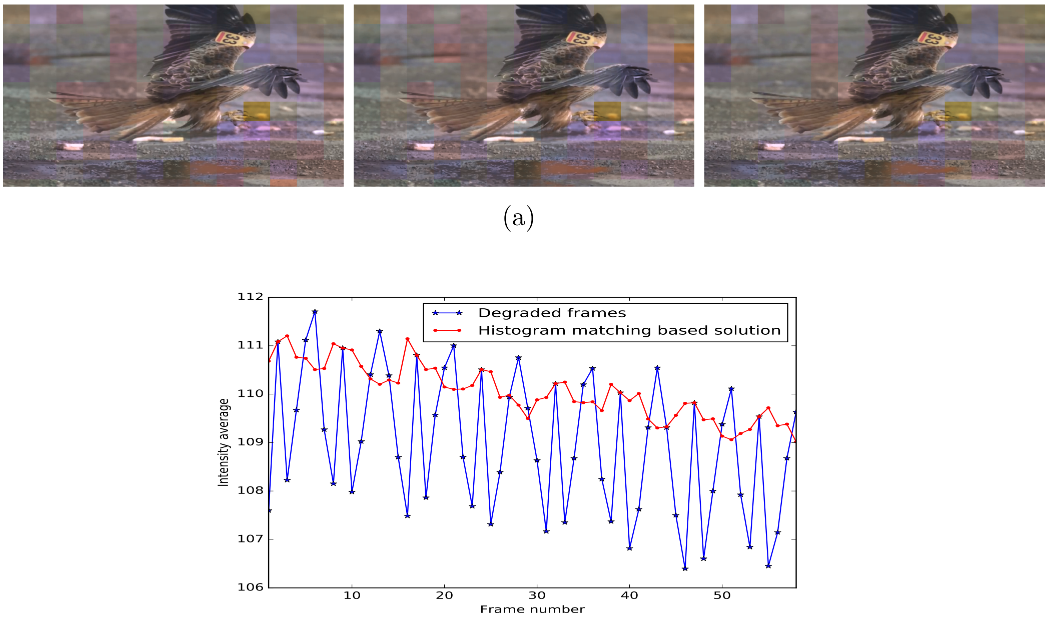
However local artificial flicker is not well corrected in some cases, especially for the real, studio-lit videos affected with periodic flicker, when multiple light sources are illuminating the scene (Figures 4.8-b, 4.9-b, 4.10-b). This is due to the important sensitivity of this method to illumination variations and outliers, and the presence of varying flicker characteristics depending on the image location.



(a)

(b)

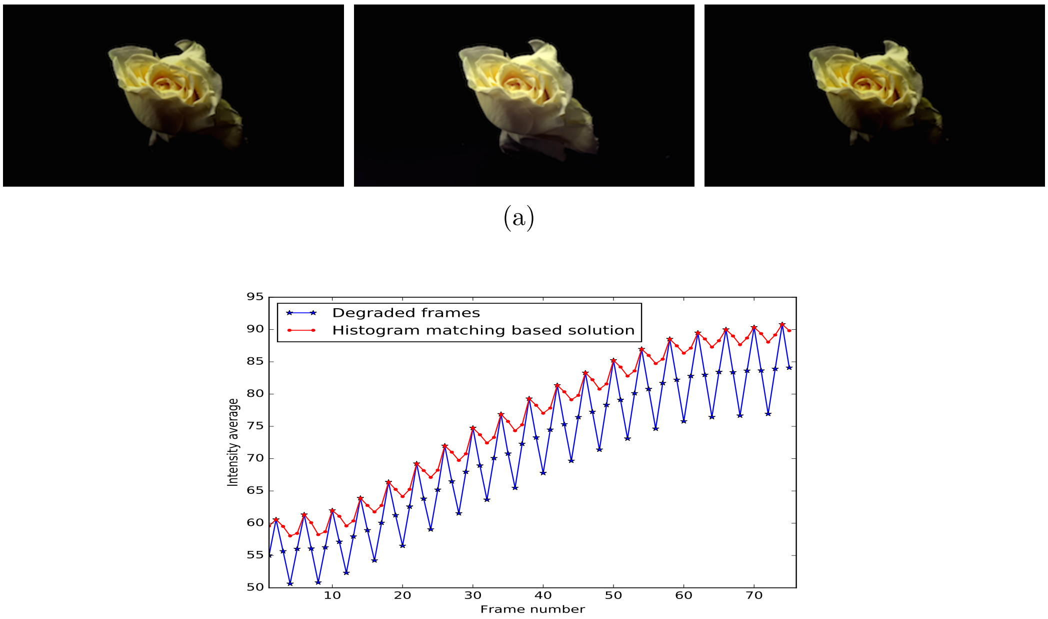
Figure 4.6. Falcon sequence with global artificial flicker



(a)

(b)

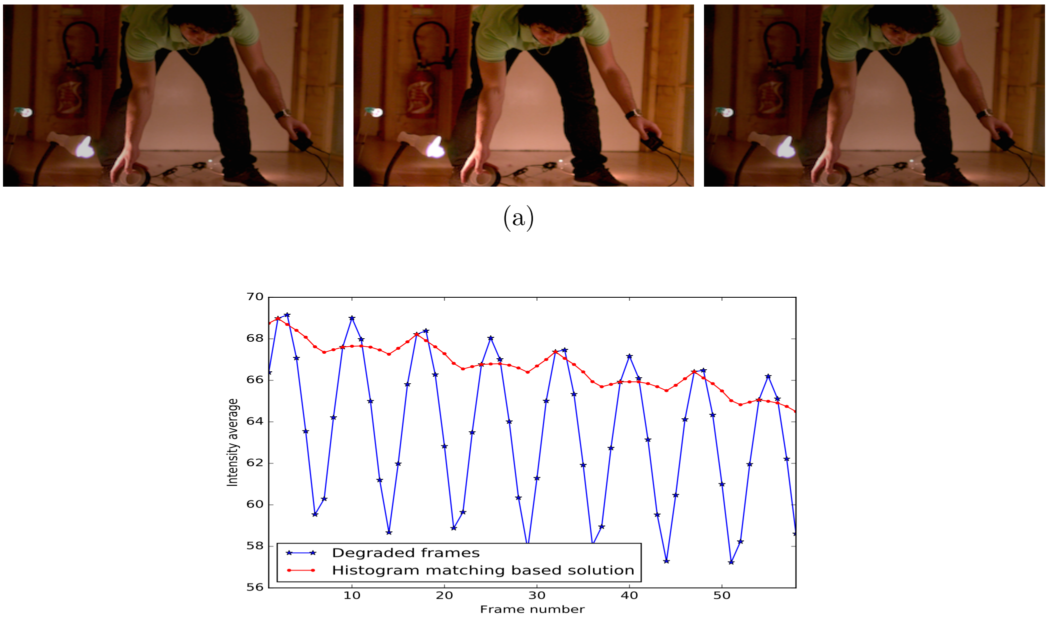
Figure 4.7. Falcon sequence with local artificial flicker



(a)

(b)

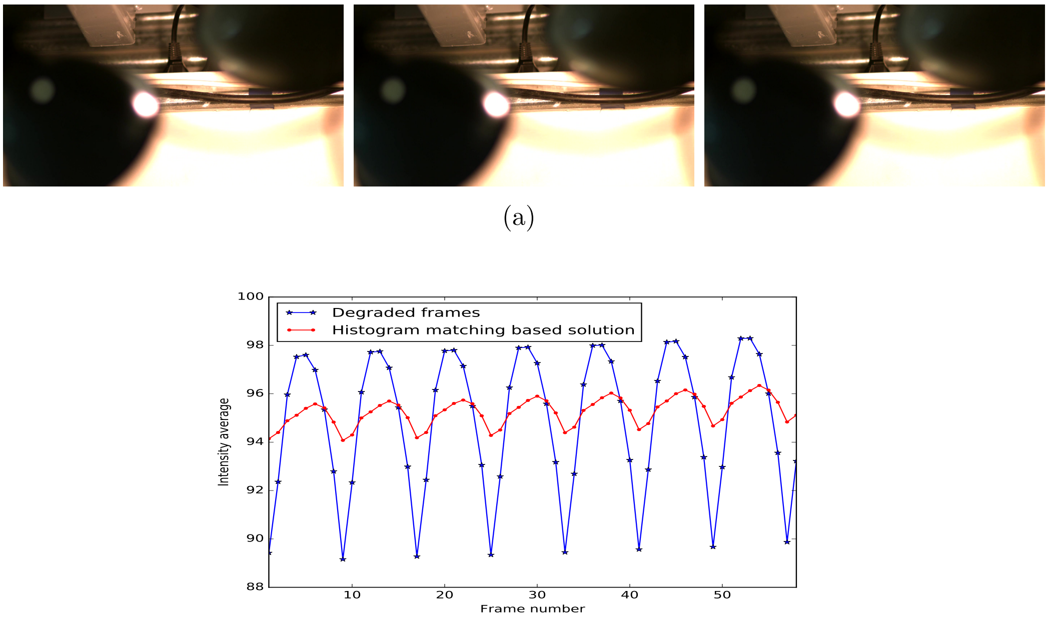
Figure 4.8. Flower sequence



(a)

(b)

Figure 4.9. Real flickering sequence 1



(a)

(b)

Figure 4.10. Real flickering sequence 2

1.4 Conclusion

In this section we have presented a global, histogram-based correction method, which is simple and fast, flicker removal is satisfactory when a single light source is illuminating the scene, and when motion is well approximated with a global transform.

The advantages of this method are its efficiency and simplicity compared to other techniques, but this method remains limited with respect to complex motions in videos, the presence of multiple light sources and the presence of noise. We provided two parametric approaches for flicker enhancement: the first takes into account the intensity variations of color channels separately, so colorimetric variations are not dealt with in this specific process, and the second approach takes into account the interchannel correlation in order to provide a joint illumination/chroma correction for image sequences. Unfortunately, this technique is very sensitive to saturation in the high and low color ranges. This method can nonetheless be used as a preprocessing to reduce the flicker in videos.

In the next section, we present another global approach using image registration techniques based on keypoints matching.

2 Image registration based method

2.1 Generalities

Image registration consists of mapping images to compare their information. This technique aims to find the geometric transformation that best physically aligns the similar pixels between two images (the reference and sensed images) acquired in different imaging conditions (at different times, from different sensors or viewpoints).

Image registration has many applications: in medical imaging it is used to merge multiple imaging modalities for example combining computer tomography (CT) and NMR data to obtain complementary information about the patient, automated cell tracking, monitoring tumor growth, treatment verification, comparison of the patient's data with anatomical atlases. In video processing, multiple applications require the use of registration technique, typically for motion tracking and automated detection and tracking of forms and objects. It is also an essential stage in image stitching consisting in combining multiple photographic images with overlapping fields of view to produce a segmented panorama or high-resolution image. In remote sensing, image registration is used in mosaicing of images of the surveyed area. There are multiple problems that pose some particular challenges while performing image registration, for instance:

- sensors noise that is sometimes due to the temperature or humidity
- occlusions, objects movements, deformations and scale changes
- lighting and atmospheric changes inducing brightness and chroma variations in images
- the target images to register may come from multiple sources.

2.2 About image registration theory

The alignment of pixels between a reference and moving image in simple terms is a change in the coordinate system so that the coordinate system of the moving image adopts that of the reference image. The types of transformations of an image can go through translation, rotation, a similarity transformation including translation, rotation and scaling of the image to be processed, and affine transforms. More complex deformations can also be sought for but will not be considered in this work. The mapping between two images f_1 and f_2 can be expressed as:

$$f_2(x,y) = g(f_1(x',y')) \tag{4.16}$$

where g is a $1D$ intensity or radiometric transformation and (x',y') is the $2D$ spatial coordinate transformation result of (x,y)

$$(x',y') = T(x,y). \tag{4.17}$$

The choice of spatial transformation type is a fundamental step for image registration. In addition, the best transform should distinguish between the spatial distortions due to different acquisitions and those due to differences in scene characteristics. The transformation that is most often used for image registration is the affine transformation, which is able to match two images acquired from the same viewing angle but from different positions. It is a global transformation which is often performed for rigid registration, i.e. the geometric relationships between pixels in the whole image do not change. This transformation is composed of four parameters, t_x, t_y, s and θ representing the horizontal and vertical translations, the scale and the rotation angle respectively, these parameters map a point (x, y) of the image f_1 to the point (x', y') of the image f_2 as follows:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + s \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{4.18}$$

Equation (4.18) can be generalized to yield the following $2D$ homothety transformation:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a_{13} \\ a_{23} \end{pmatrix} \tag{4.19}$$

or in so-called homogeneous coordinates (Jia (Iowa State University, 2013)):

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{4.20}$$

Registration methods are classified into three categories (Brown (1992); Zitova and Flusser (2003); Maintz and Viergever (1998)): geometric methods (also called feature based methods) based on matching the geometric primitives (corners, edges, ...) extracted from images, intensity-based methods that do not require any pre-segmentation and are directly based on pixel intensities of the two images via the use of a similarity measure without taking into account the geometrical information, and finally hybrid methods combining the previous two approaches.

2.3 Feature-based approach

Usually, registering images that are acquired by the same sensor type and with ideal acquisition conditions, can be performed with simple algorithms such as block matching and using some defined similarity criteria. Nevertheless, in multi-modality methods and in our case, while using the same camera for all images acquisitions and in the presence of brightness and chroma variations in images, the similarity search criteria quickly becomes problematic. Therefore geometric registration seems more suitable.

However, in order to apply this technique, it is essential to estimate the geometric transformation between images, hence feature matching algorithms are well suited for this. Good quality keypoints can be provided by the SIFT (Lowe (1999, 2004)) and SURF (Bay et al. (2006)) algorithms. These use local descriptors that are insensitive to affine transform (mostly rotation and change of scale). There is theoretically little need to pre-process images. These methods propose keypoints together with their descriptors from images and measure a number of geometric characteristics depending on a scale factor. First, we used the keypoints detection and matching by the SIFT method to perform a rigid registration, and a global color correction is performed in the second stage.

2.3.1 An overview on SIFT

SIFT method (scale-invariant feature transform) was developed by David Lowe, to transform an image into a set of features vectors that are invariant to the usual geometric transformations (scaling, rotation) and to affine transformations and to some degree on illumination. The strength of this method is that it is able to match distant points with important camera movements. The algorithm of SIFT largely fills the limitions of methods of features extraction already developed earlier by Harris and Stephens (1988), and later by Schmid and Mohr (1997). Indeed, it has contributed to an improvement of information extraction techniques in an image.

The computation of SIFT keypoints requires two main steps. First, it is necessary to extract the characteristics of an object in the image and compute an associated descriptor, i.e. detect the features that are most representing/discriminating this object from others. Second, it must establish a matching procedure that is the ultimate goal of the method.

2.3.2 Scale-space construction for keypoints detection

Scale-space is an important theory in artificial vision. This theory is based on the fact that in the real world, objects have properties that are associated with notion of scale, i.e. a geometric magnification. Hence the need for tools that allow us to describe this object at varying scales. The basic idea is to decompose the original signal into a family of signals depending on a gradually smoothed parameter, in which details at a very small scale are successively removed. SIFT represents grayscale signal data using a Gaussian pyramid by combining sub-sampling operations with a smoothing step.

The scale-space is a discrete space wherein is assigned to each pixel, in addition to its cartesian coordinates (x, y), a third component σ, which represents the scale factor. For this, a conventional convolution is performed between the original image f, and a Gaussian function G that takes x, y and σ as arguments:

$$L(x, y, \sigma) = G(x, y, \sigma) * f(x, y) \tag{4.21}$$

where G is a gaussian filter and "$*$" refers to the spatial convolution product between the Gaussian filter and the target image f, and L is the smoothed image. G can be expressed as follows:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma} \exp^{-(x^2+y^2)/(2\sigma^2)} \tag{4.22}$$

where σ is a smoothness factor.

Then the difference of Gaussians (DoG) between two consecutive images of the same octave in the Gaussian pyramid is computed to obtain the DoG pyramid (Figure 4.12).

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * f(x, y) \tag{4.23}$$

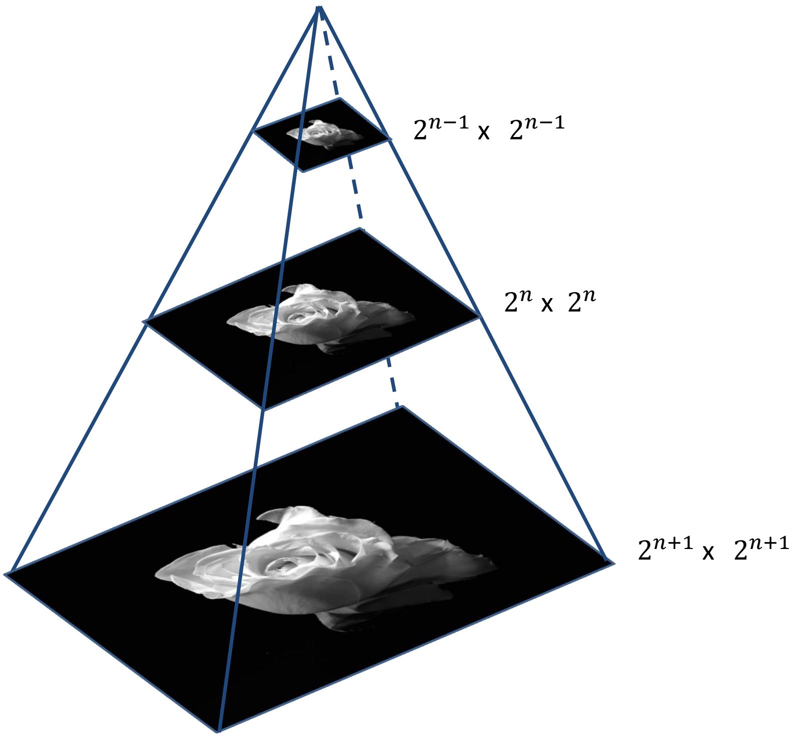


Figure 4.11. Scale space construction: Gaussian pyramid

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma) \tag{4.24}$$

The difference between two consecutive smoothed images by a Gaussian filter is the result of a high pass filter, useful for the edge detection.

Via this step, it is possible to detect keypoints at various resolutions, and so this detection method is approximately scale-invariant. The keypoints are the local extrema of DoG images across scales. Each pixel of the DoG images is compared with its eight neighbors in the same scale level and to its nine neighbors respectively in each of nearest two scale levels.

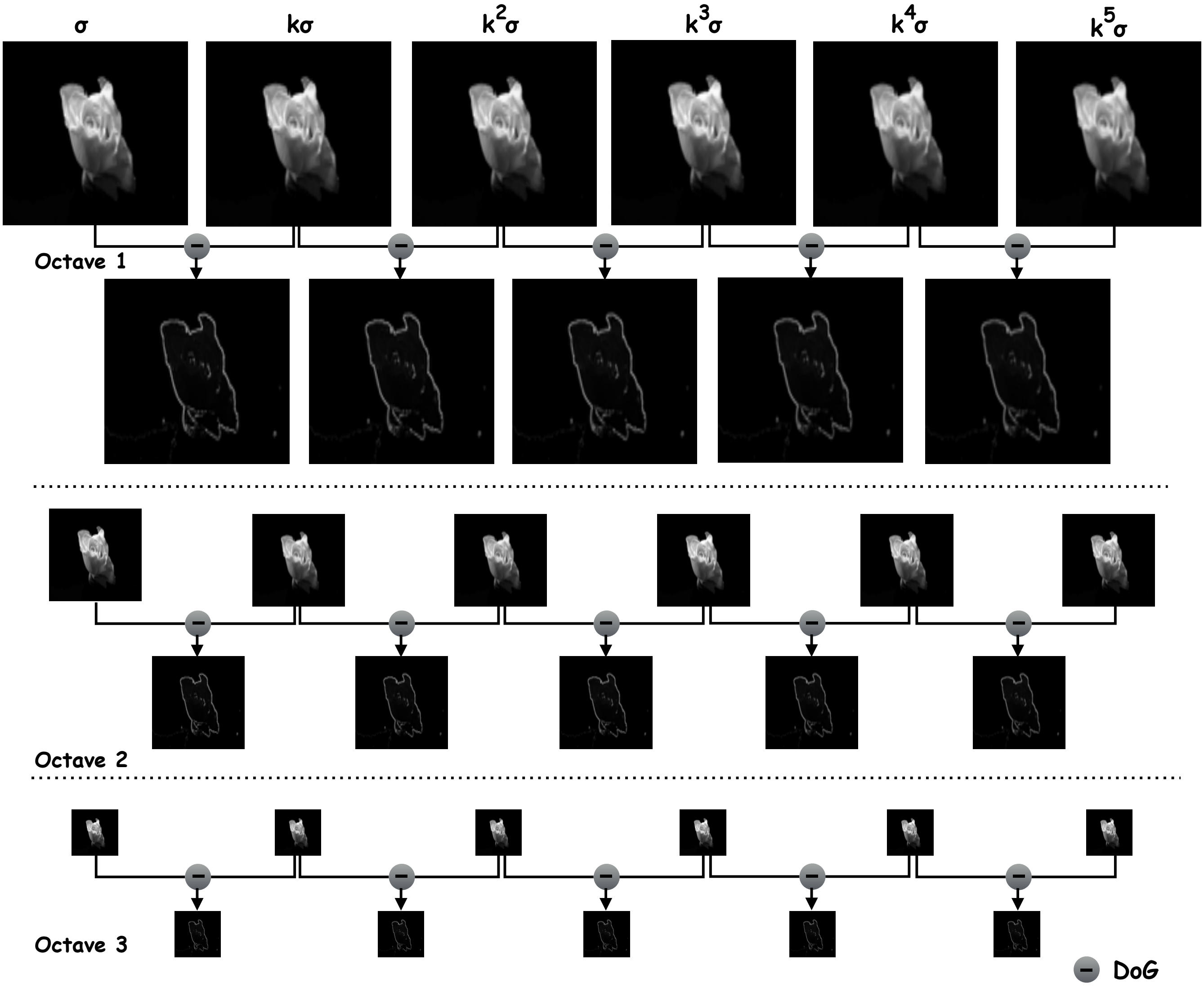


Figure 4.12. DoG pyramid

2.3.3 Descriptors calculation

Only the useful keypoints are restored, that are robust to noise, due to the extensive use of Gaussian blurring. In addition, an interpolation of the coordinates is performed by a second-order Taylor expansion of the DoG function $D(\mathbf{x})$ where $\mathbf{x} = (x, y, \sigma)$ is a selected candidate point in the previous step.

One or more orientations (descriptors) locally determined in the image are assigned to each keypoint, using the direction of the gradients in a surrounding neighborhood. This step is essential to ensure the rotation invariance. Following this step, a keypoint is defined by four parameters (x, y, σ, θ), where θ is the rotation parameter. On the same image, keypoint may differ by only one of these four parameters (for example the scale factor or the orientation factor, this is particularly true for zones of textures).

2.4 Methodology

We use the SIFT keypoints descriptors to register each subsequent frame to its reference image in the same flicker period. The SIFT algorithm generates a number of keypoints, which are invariant to image translation, rotation and scale, and so yields matches between two images even in the presence of illumination variations. The management of these matches allows us to estimate better geometrical transformations.

2.4.1 Image matching

We extract SIFT keypoints from each subsequent reference frame, which are stored in a database. Similarly, we extract the keypoints for each image in the sequence. We find matching keys between each subsequent frame and its reference. For each keypoint, we have a location vector, which provides the keypoint position (row, column, scale, orientation). We compute translations that each keypoint undergoes, by subtracting the corresponding location vectors in the reference frame and the target frame respectively. In our case, we can neglect the rotations which are very small.

Figure 4.13 shows vertical plotted against horizontal translations in order to check whether or not there are bad matches estimated. We can see that most points are centered in a narrow range, but some keypoints seem to undergo a larger translations than the others. In addition, as we study videos with global motion, we expect all translations to be similar, so we can consider that there was a mismatch in the previous step.

To compute the exact geometric transformations, first, we propose to eliminate

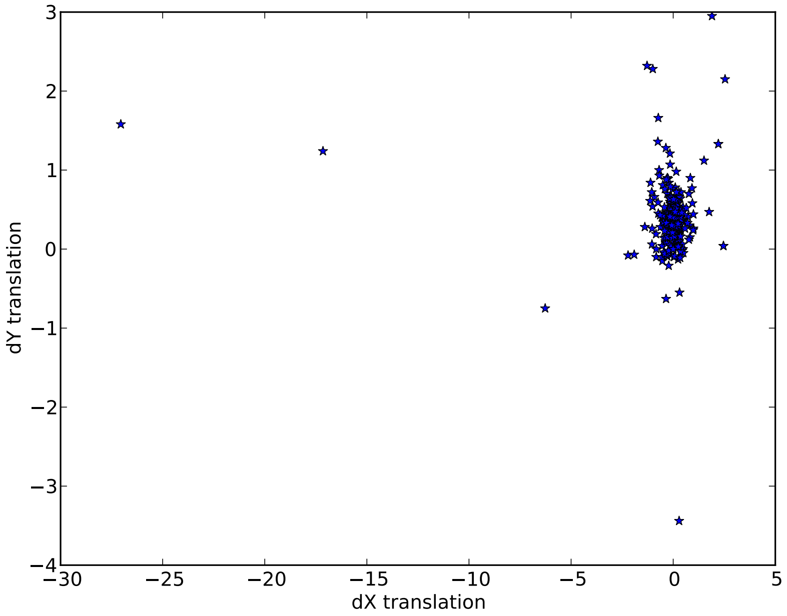


Figure 4.13. Vertical vs. horizontal translations of matched keypoints

outlier translations, and then calculate the average of all translations. We add a condition which examines whether or not a translation value will be taken into account. In our case, we propose that the Euclidean distance between its translation parameters and those of its neighbors should be less than a given threshold.

This Euclidean distance between two different points a and b in the plot in Figure 4.13 is defined as follows:

$$D_{ab} = \sqrt{(dX(a) - dX(b))^2 + (dY(a) - dY(b))^2} \tag{4.25}$$

where (dX, dY) are the horizontal and vertical translations.

We plot again dY with respect to dX after applying the Euclidean distance constraint: Figure 4.14 shows that remaining points have very close translation averages, and all points with large translations are eliminated. Then we calculate the average of the translations as the most accurate estimate of the true

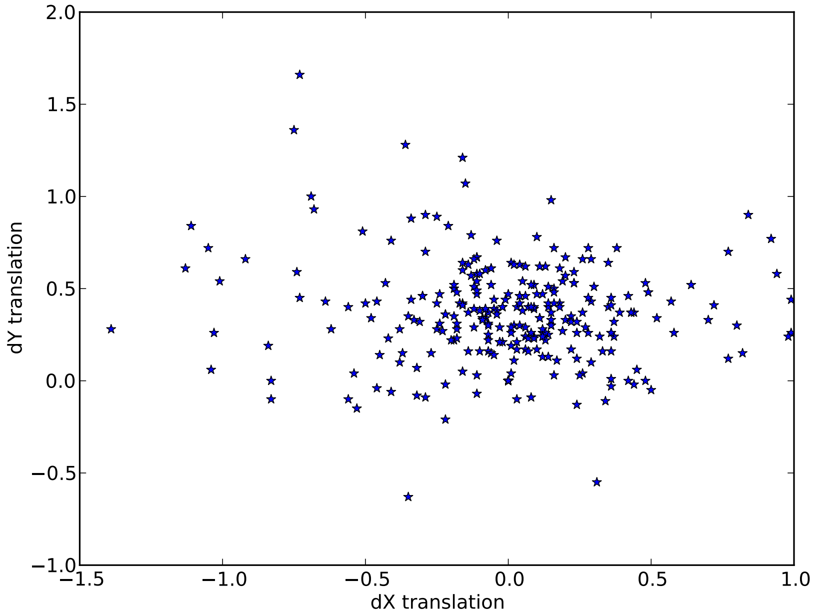


Figure 4.14. Vertical vs. horizontal translations of matched keypoints with Euclidean distance

translation.

2.4.2 Application of the transformation matrix

Now we have enough information to form the geometric transformation matrix between each image its corresponding reference, and it takes the following form:

$$\mathbf{T} = \begin{bmatrix} \cos\hat{\theta} & -\sin\hat{\theta} & T_x \\ \sin\hat{\theta} & \cos\hat{\theta} & T_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.26}$$

where $\hat{\theta}$ is the rotation mean of all keypoints which is almost negligible (in our case), T_x and T_y are the translation means with respect to x and y, i.e. those are the mean of vectors dX and dY respectively.

The inversion of this matrix shifts geometrically the image $f(\cdot, t)$ to the reference image $f(\cdot, t_{\text{ref}})$. Once the transformation matrix between the reference and each current frame has been found, it is applied to each.

2.4.3 Color correction step

Sampling data

There is no need to consider all the points in the image. We sample a few thousand points randomly placed on the image, as in Figure 4.15. At each sampled point we associate a vector $\boldsymbol{V}_{\text{ref}} = (R_{\text{ref}}, G_{\text{ref}}, B_{\text{ref}})$ in the reference image to a vector $\boldsymbol{V}_{\text{reg}} = (R_{\text{reg}}, G_{\text{reg}}, B_{\text{reg}})$ in the registered image. Because they are registered, we expect both sets of vector to correlate well.

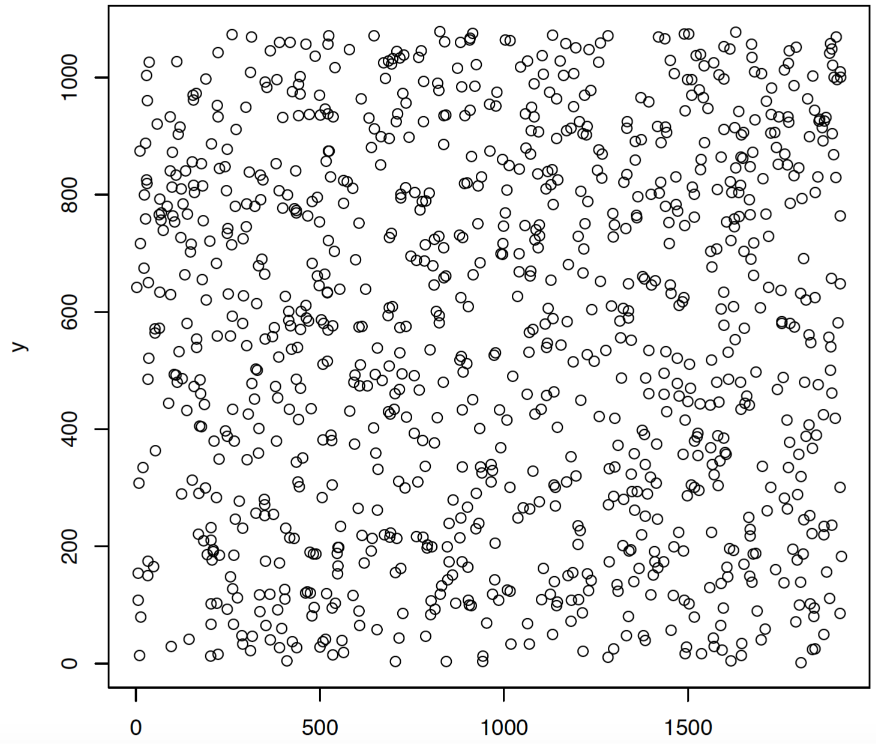


Figure 4.15. Random sampling on the image

Multiple least-square regression

We build a multiple least-square regression model fitting the reference vs. the registered image.

Given a vector $\boldsymbol{V}_{\text{reg}}$ corresponding to a point $\mathbf{s}$ in the registered image, we try to estimate the color correction matrix $\boldsymbol{M}$ such that

$$\boldsymbol{V}_{\text{ref}} = \boldsymbol{M}\boldsymbol{V}_{\text{reg}} + \boldsymbol{O} \tag{4.27}$$

we recall that $\boldsymbol{M}$ is always a 3×3 matrix representing the luminance from the diagonal elements and chroma parameters from the others, and $\boldsymbol{O}$ is an offset for modeling additive noise.

Color correction model

The linear model is very easy to apply using equation (4.27). In accordance with our experimental model, for a given frame $f(\cdot, t)$, the optimal transform matrix $\widehat{\boldsymbol{M}}_t$ is that minimizes the following energy function:

$$J(\widetilde{\boldsymbol{M}}_t) = \sum_{\mathbf{s}} \|\widetilde{\boldsymbol{M}}_t \mathbf{V}_{reg}(\mathbf{s}) - \mathbf{V}_{ref}(\mathbf{s})\|^2, \tag{4.28}$$

For each frame, 9 scalar parameters for the color correction have to be estimated. For minimizing the energy function J, we use the standard least square method:

$$\widehat{\boldsymbol{M}}_t = \sum_{\mathbf{s}} (\mathbf{V}_{ref}(s)\mathbf{V}^{\top}_{reg}(s))\,(\sum_{\mathbf{s}} \mathbf{V}_{reg}(s)\mathbf{V}^{\top}_{reg}(s))^{\dagger} \tag{4.29}$$

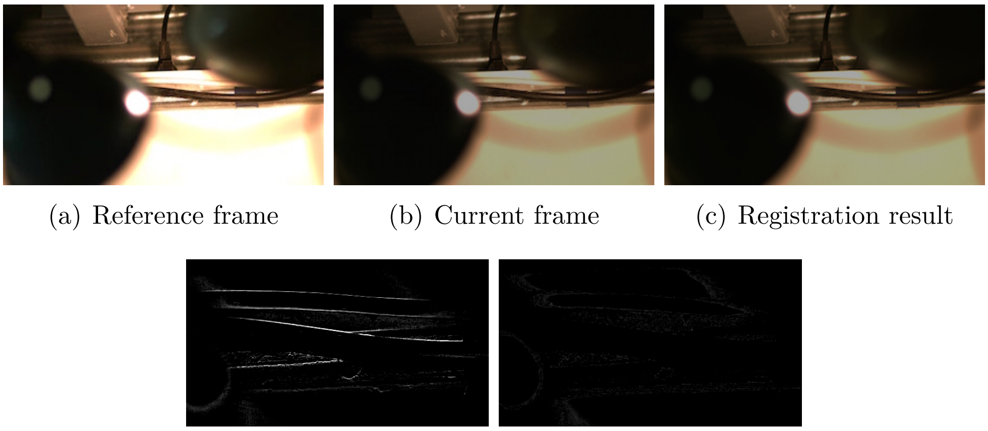
Once the optimal color transform between the reference and the current frame warped to the reference has been found, it is applied onto the original one. This yields a flicker-compensated frame. Reproducing the same procedure for each frame yields a flicker-compensated sequence.

However some data will exceed the range of the image. We clip data above 255 to that value, and also negative values to zero. We find very few negative values, but corrected values exceeding 255 are expected in areas of high intensity.

Although this is not obvious on the printed page, the correction is quite effective when we compare the reference and corrected images pixel per pixel.

2.4.4 Results and performances

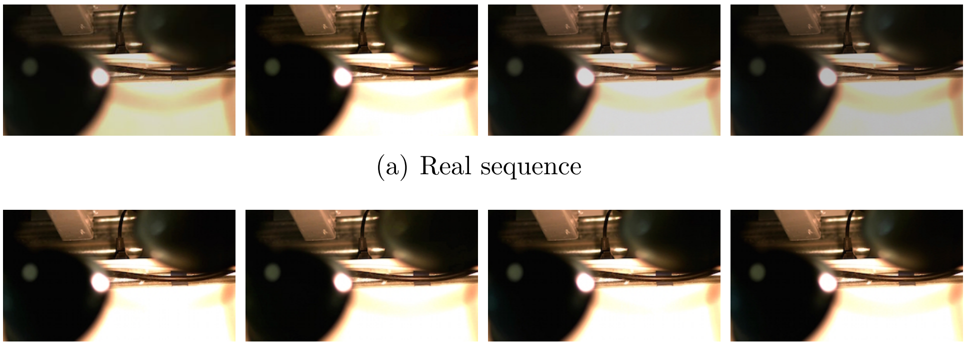
In this section, we present some experimental results to show the performance of this global method. The registration method was tested on a real sequence, the disparity images before and after registration show that the global motion is well estimated (see Figure 4.16(d, e)). Our color correction model was tested on a synthetic flicker sequence produced from a flicker-free, naturally lit video and also on a real, studio-lit video affected with global flicker and motion. Figure 4.17 shows the processing result for the real degraded image sequence.



(a) Reference frame (b) Current frame (c) Registration result

(d) Disparity before registration (e) Disparity after registration

Figure 4.16. Image sequence with periodic flicker from studio lighting.



(a) Real sequence

(b) Processing result

Figure 4.17. Global color correction using image registration based method

The luminosity variation graph (Figure 4.18) shows that brightness levels are enhanced in the processed frames, but there are still few variations between luminosity averages.

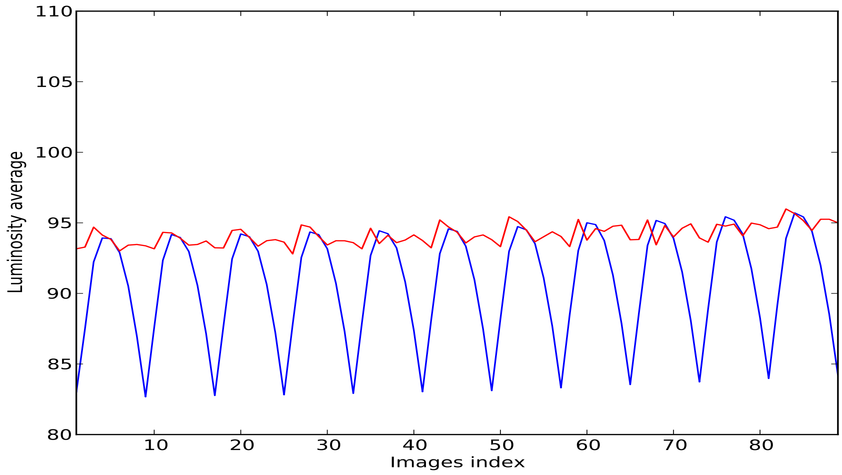
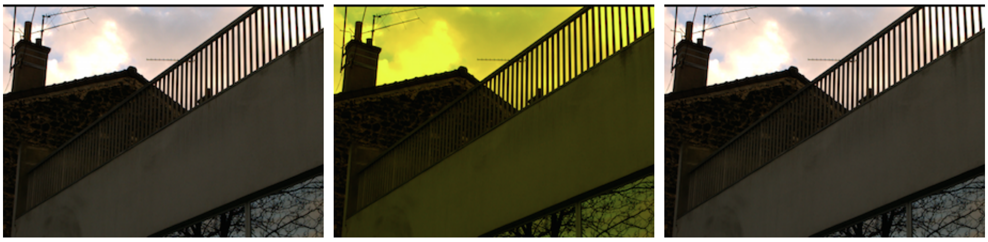


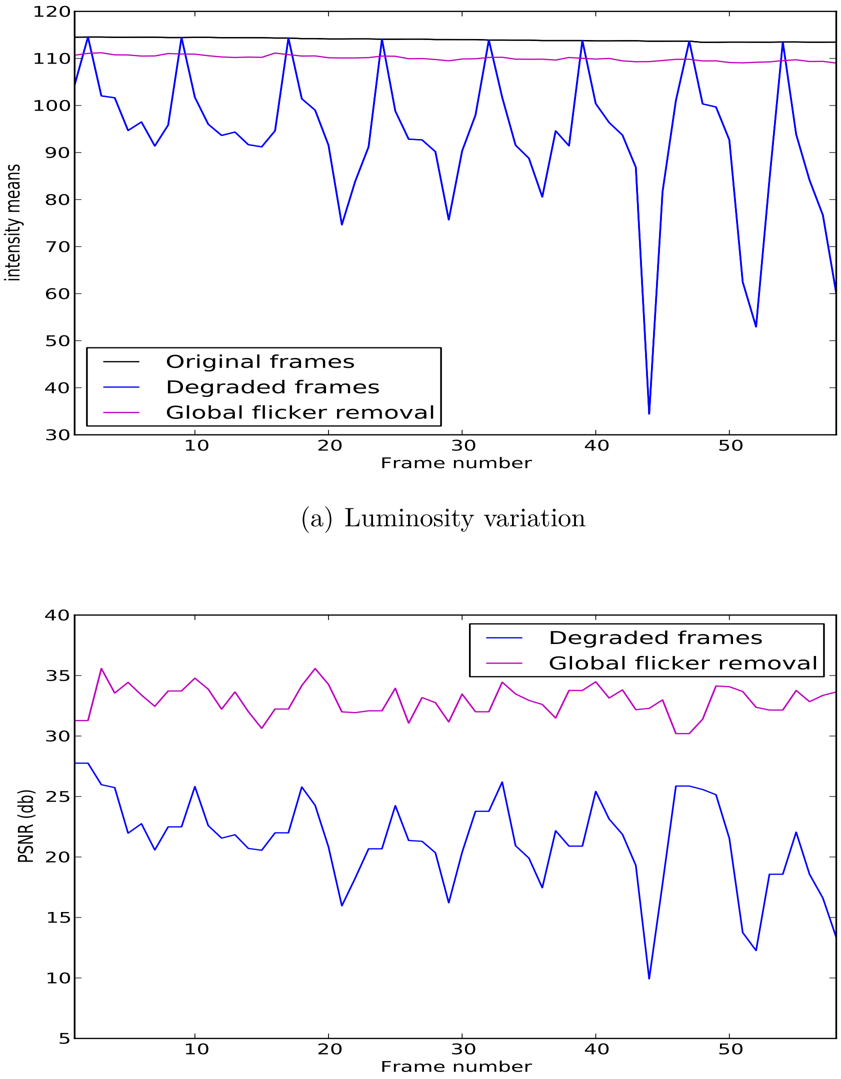
Figure 4.18. A comparison between luminosity average before and after flicker correction using image registration based method

As we do not have access to a flicker-free reference sequence, we cannot precisely estimate similarity measures only from the luminosity variation graph. So we performed our global processing approach on a sequence with synthetic flicker. We use the structural similarity index (SSIM), and signal-to-noise ratio (SNR) estimators to measure the similarity between the original and degraded images on the one hand, and then between the original and restored sequences on the other hand. Figure 4.19 represents a restoration example on a sequence with artificial flicker. This example features global motion between frames. The PSNR average increases from 21 to 32 dB after color correction (see Figure 4.20(b)), so the gain is significant. Similarly, the SSIM index is very close to 1 after restoration (see Figure 4.20(c)).



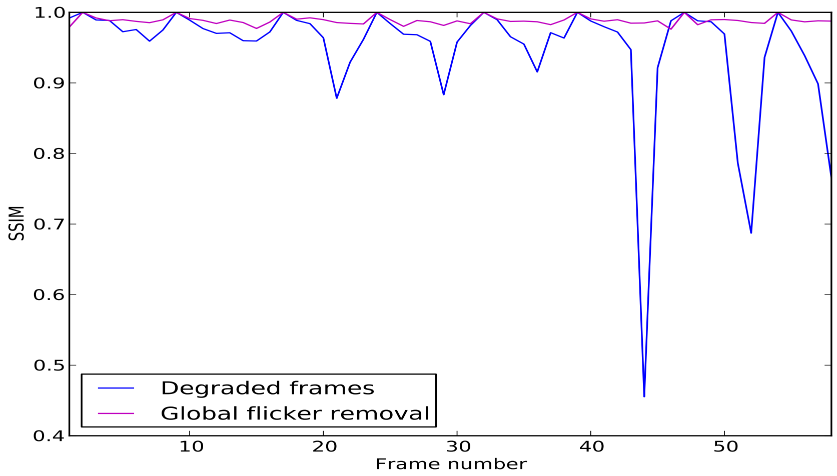
(a) Original image (b) Degraded image (c) Color correction result

Figure 4.19. Sequence with global motion and artificial flicker.



(a) Luminosity variation

(b) SNR comparison



(c) SSIM comparison

Figure 4.20. Artificial flicker: similarity measures

3 Summary

In this chapter, first we used histogram matching for flicker correction. This method has some limitations with respect to the saturation effects and outliers, but we showed that it can be used to efficiently reduce flicker in videos. We also proposed a different global approach using keypoints matching for image registration followed by a least-squares linear color correction. The restoration method works well in the presence of single light source, global flicker and simple motion.

Both methods are efficient in terms of computation time and are useful for reducing flicker in most acquisition conditions. The registration-based method has an advantage with respect to the histogram-based method. It is more robust to outliers, and yields a better estimate of the color correction parameters. This is due to the fact that a registration match, even inaccurate, provides more information than a histogram match.

In the next chapters, we develop more sophisticated methods that deal with videos affected with flicker featuring multiple light sources and complex motion.

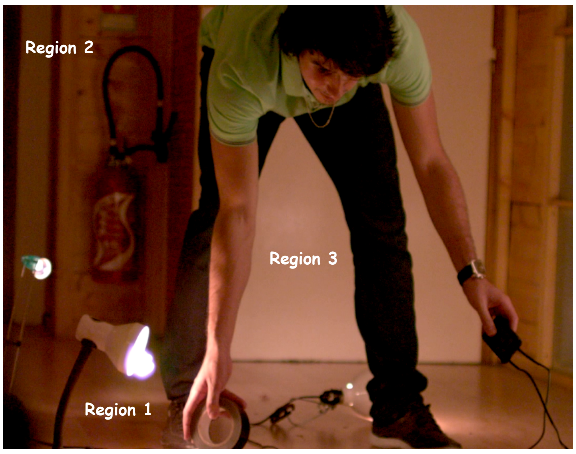
Chapter 5

Block matching-based colorimetric correction

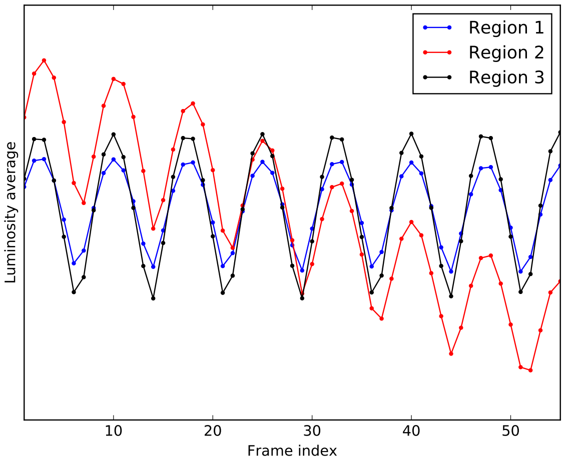
1 Introduction

In the case of multiple light sources illuminating the scene, we observe a composite flicker effect in image sequences depending on the location of each light source. This means that we cannot model the illumination variation by a single correction matrix for all pixels in the frame. We demonstrate this in Figure 5.1 which shows the layout of flicker in different regions in the scene. Flicker can be modeled in a different manner from one region to another, depending on several factors, for instance the position of the region with respect to the light source in the scene, the random noise distribution over images for various reasons (high/low luminosity, compression artifacts, etc) or due to the different light absorption characteristics between the multiple scene regions and materials. For example, a black material absorbs all wavelengths, whereas a white material reflects them. To handle situations where the flicker varies across the image, we propose a joint tracking/color correction scheme using a block matching technique paired with color variation estimation.

As previously presented in the litterature review, local colorimetric and luminosity variations constitute a significant challenge for motion tracking on the one hand, and for color restoration on the other hand. Most videos contain background and object motion, and so matching pixels between frames needs to be performed, although most tracking methods assume consistency of color between frames.



(a)



(b)

Figure 5.1. An illustration of flicker effects in different regions of a single sequence. Luminosity variations vary significantly between regions, and so cannot be corrected in the same way everywhere.

In this chapter, we propose local approaches for color correction, based on a block matching algorithm that is able to include the constraints of illumination and chroma variations. A post-processing step is also proposed in order to deal with blocking artifacts. Firstly, a classical causal strategy is proposed for correcting an image with respect to its previous reference frame. Secondly, a non-causal strategy is implemented to take into account multiple reference frames in order to handle pseudo/non-periodic illimination and chroma variations. Finally, a multi resolution strategy is adopted to accelerate processing.

2 Causal method for periodic flicker removal

2.1 Causal definition

In signal processing, a system is called causal if it has, at time t, an output signal which depends only on the previous values and optionally on current values of the input signal. Future signal values cannot be used. This means in our case that a current frame in the video sequence should be processed with respect only to previous reference frames. This classical approach is widely used in video processing. This approach is suitable for periodic flicker because in this case the past and present overlap to some degree. However it is less suitable to spurious flicker and does not cope well with low frequency illumination variations.

2.2 Tracking step

Classical optical flow methods assume constancy of illumination as we have seen in Chapter 3. They are sensitive to the presence of these illumination variations, whereas keypoint-based methods are more robust to illumination changes and so are more robust for global motion estimation. In order to generalize our procedure, we propose a localized color transformation estimation method. Since tracking single pixels is impractical, we use a block-matching method, assuming that each spatial block $\mathcal{B}_k$ indexed by k can be assigned its own model parameters, i.e. $(\forall \mathbf{s} \in \mathcal{B}_k)\ \boldsymbol{M}_{\mathbf{s},t} = \widetilde{\boldsymbol{M}}_{k,t}$, which means that all pixels indexed by $\mathbf{s}$ of the block $\mathcal{B}_k$ are processed using the same color transformation matrix $\widetilde{\boldsymbol{M}}_{k,t}$. More precisely, we propose to divide all frames into disjoint blocks of identical sizes, typically 64 × 64, and to perform a joint block-matching / color transform estimation between flicker-affected frames and the previous reference frame.

In accordance with our experimental model, for a given block $\mathcal{B}_k$ in frame t, the optimal transform matrix $\widehat{\boldsymbol{M}}_{k,t}$ and displacement vector $\widehat{\mathbf{d}}_{k,t} \in \mathbb{R}^2$ are those that minimize the following energy function:

$$J(\widetilde{\boldsymbol{M}}_{k,t}, \mathbf{d}_{k,t}) = \sum_{\mathbf{s} \in \mathcal{B}_k} \Phi\big(\widetilde{\boldsymbol{M}}_{k,t}\, \mathbf{f}(\mathbf{s}, t) - \mathbf{f}(\mathbf{s} - \mathbf{d}_{k,t}, t_{\mathrm{ref}})\big), \tag{5.1}$$

where $\mathbf{d}_{k,t}$ is a candidate displacement vector between two corresponding blocks and $\Phi \colon \mathbb{R}^3 \to [0, +\infty[$ is a cost function. Usually, a typical choice for Φ is the squared Euclidean norm where $\Phi = \| \cdot \|^2/2$. J can thus be expressed as follows:

$$J(\widetilde{\boldsymbol{M}}_{k,t}, \mathbf{d}_{k,t}) = \frac{1}{2} \sum_{\mathbf{s} \in \mathcal{B}_k} \left\| \widetilde{\boldsymbol{M}}_{k,t}\, \mathbf{f}(\mathbf{s}, t) - \mathbf{f}(\mathbf{s} - \mathbf{d}_{k,t}, t_{\mathrm{ref}}) \right\|^2, \tag{5.2}$$

The minimization of (5.2) can be performed by the standard least squares strategy, i.e. setting to zero its derivative with respect to the sought parameter $\widetilde{\boldsymbol{M}}_{k,t}$,

$$\frac{\partial J(\widetilde{\boldsymbol{M}}_{k,t}, \mathbf{d}_{k,t})}{\partial \widetilde{\boldsymbol{M}}_{k,t}} = \sum_{\mathbf{s} \in \mathcal{B}_k} \big(\widetilde{\boldsymbol{M}}_{k,t}\, \mathbf{f}(\mathbf{s}, t) - \mathbf{f}(\mathbf{s} - \mathbf{d}_{k,t}, t_{\mathrm{ref}})\big) \mathbf{f}(\mathbf{s}, t)^\top = 0. \tag{5.3}$$

Hence, the solution to the quadratic minimization problem is provided by

$$\widehat{\boldsymbol{M}}_{k,t} = \Big(\sum_{\mathbf{s} \in \mathcal{B}_k} \mathbf{f}(\mathbf{s} - \mathbf{d}_{k,t}, t_{\mathrm{ref}}) \mathbf{f}(\mathbf{s}, t)^\top \Big) \cdot \Big(\sum_{\mathbf{s} \in \mathcal{B}_k} \mathbf{f}(\mathbf{s}, t)\, \mathbf{f}(\mathbf{s}, t)^\top \Big)^{\dagger}. \tag{5.4}$$

Otherwise, with more complex choices of the cost function Φ, the previous convex optimization problem defined in (5.1) can be solved by an iterative strategy (typically, a Majorization-Minimization strategy). This choice of optimization strategy is briefly explained below.

2.3 An iterative optimization strategy

Generally, the optimization of a function $g(\mathbf{u})$ with respect to a given parameter $\mathbf{u}$, consists of looking for an estimation of the unknown parameter $\hat{\mathbf{u}}$ that minimizes or maximizes the cost function. In our context, our cost function Φ represents the error of matching blocks between the target image and the reference image, hence we are interested in finding the minimization strategy:

$$\hat{\mathbf{u}} = \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^N} g(\mathbf{u}), \tag{5.5}$$

where g could include multiple terms, the first term to represent the relation between the target and reference data, and other regularization terms to improve the conditioning of the problem, and for finding a better solution for the sought variables. It should be noted that most optimization algorithms operate within a general iterative strategy, consisting, at each iteration to get closer to the minimum by solving a sub-problem minimization. Obviously, this strategy makes sense only if these sub-problems are easier to solve than the original problem. There are exceptions for some very specific cases, for instance fully quadratic problems, as seen above.

2.3.1 Majorization-Minimization approaches

The general MM algorithm was first proposed by Ortega and Rheinboldt (1970). It consists of finding a solution to the optimization problem (5.5), where Φ is assumed to be differentiable, in an iterative manner. It relies upon the following concept:

Definition 3. *Let $g\colon \mathbb{R}^N \to]-\infty,+\infty]$. Let $v\in\mathbb{R}^N$. $q(\cdot\mid v)\colon \mathbb{R}^N \to]-\infty,+\infty]$ is a tangent majorant function of g at v if $(\forall u\in\mathbb{R}^N)\quad f(u)\le q(u\mid v),\quad$ and $f(v) = q(v\mid v)$.*

The MM algorithm minimizes a tangent majorant function q of g at each iteration $k\in\mathbb{N}$. It can be summarized by the following algorithm and Figure 5.2.

A typical choice of majorant function is a stricly convex quadratic function. It ensures a unique solution to sub-problems of Algorithm 1.

Algorithm 1: Majorization-Minimization algorithm

```
Initialization: Set $u_0 \in \mathbb{R}^N$
for  $k$= 0,1, .. do
    $u_{k+1} \in \underset{u \in \mathbb{R}^N}{\operatorname{argmin}}\; q(u \mid u_k)$
end
```

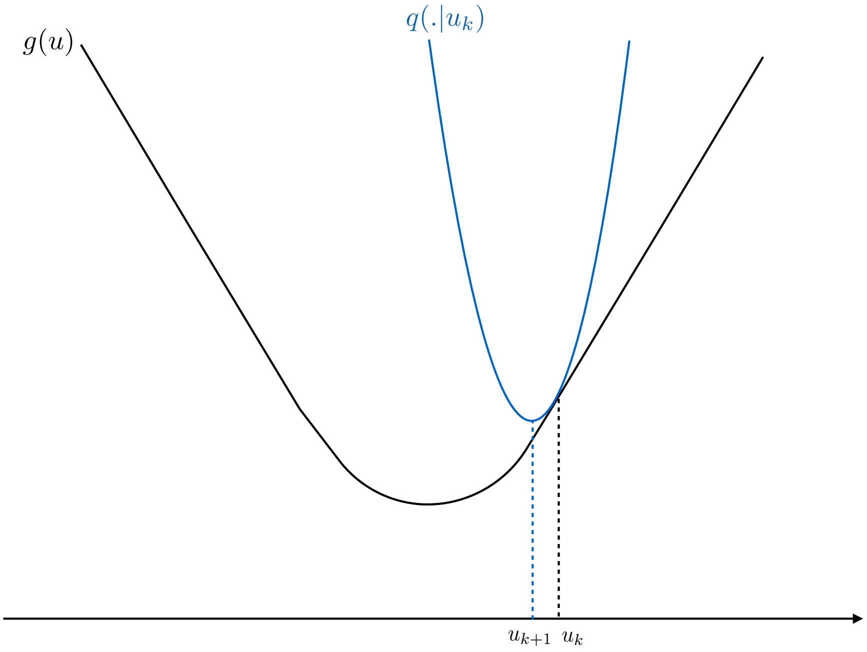


Figure 5.2. Illustration of MM algorithm for the minimization of a function $g : \mathbb{R}^N \to]-\infty, +\infty]$. At iteration $k \in \mathbb{N}$, we use a majorant function $q(\cdot \mid u_k)$ of g at a point u_k, and then we define u_{k+1} as a minimizer of $q(\cdot \mid u_k)$.

Property If g is β-Lipschitz differentiable on $\mathbb{R}^N$, according to the so-called Descent Lemma, a tangent majorant function of g at v is defined by

$$q(u \mid v) = g(v) + \langle u - v, \nabla g(v) \rangle + \frac{\mu}{2} \|u - v\|^2 \tag{5.6}$$

where $\mu \in [\beta, +\infty[$.

Consequently, this property holds in our context, provided that Φ is a differentiable cost function with a Lipschitzian gradient.

Our aim is to construct a majorant function Q of J such that:

$$\begin{cases} J(\boldsymbol{M}_{k,t}^{(n+1)}, \mathbf{d}_{k,t}^{(n+1)}) \leq Q_{\boldsymbol{M}_{k,t}^{(n)}, \mathbf{d}_{k,t}^{(n)}}(\boldsymbol{M}_{k,t}^{(n+1)}, \mathbf{d}_{k,t}^{(n+1)}) \\ J(\boldsymbol{M}_{k,t}^{(n)}, \mathbf{d}_{k,t}^{(n)}) = Q_{\boldsymbol{M}_{k,t}^{(n)}, \mathbf{d}_{k,t}^{(n)}}(\boldsymbol{M}_{k,t}^{(n)}, \mathbf{d}_{k,t}^{(n)}), \end{cases} \tag{5.7}$$

for every $(\boldsymbol{M}_{k,t}^{(n)}, \mathbf{d}_{k,t}^{(n)}, \boldsymbol{M}_{k,t}^{(n+1)}, \mathbf{d}_{k,t}^{(n+1)})$.

The proposed algorithm then reads:

```
Algorithm 2: Majorization-Minimization algorithm
for  $n$= 0,1, .. do
    $\widehat{\mathbf{d}}_{k,t}^{(n+1)} = \operatorname*{argmin}_{\mathbf{d}\in\mathcal{W}} J(\widetilde{\boldsymbol{M}}_{k,t}^{(n+1)}(\mathbf{d}), \mathbf{d})$,
    where $\widetilde{\boldsymbol{M}}_{k,t}^{(n+1)}(\mathbf{d}) = \operatorname*{argmin}_{\boldsymbol{M}} Q_{\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \widehat{\mathbf{d}}_{k,t}^{(n)}}(\boldsymbol{M}, \mathbf{d})$
    and $\widehat{\boldsymbol{M}}_{k,t}^{(n)} = \widetilde{\boldsymbol{M}}_{k,t}^{(n)}(\widehat{\mathbf{d}}_{k,t}^{(n)})$.
end
```

Note that this algorithm constitutes an original contribution of this thesis since it mixes a discrete optimization step with respect to $\mathbf{d}$ with a continuous optimization with respect to $\boldsymbol{M}$. The first step is performed in an exact manner, while the second one employs the surrogate function Q. The majorization-minimization property is instrumental for proving the convergence of our cost function J as shown below:

$$\begin{aligned} J(\widehat{\boldsymbol{M}}_{k,t}^{(n+1)}, \widehat{\mathbf{d}}_{k,t}^{(n+1)}) &= J(\widetilde{\boldsymbol{M}}_{k,t}^{(n+1)}(\widehat{\mathbf{d}}_{k,t}^{(n+1)}), \widehat{\mathbf{d}}_{k,t}^{(n+1)}) \\ &\leq J(\widetilde{\boldsymbol{M}}_{k,t}^{(n+1)}(\widehat{\mathbf{d}}_{k,t}^{(n)}), \widehat{\mathbf{d}}_{k,t}^{(n)}) \\ &\leq Q_{\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \widehat{\mathbf{d}}_{k,t}^{(n)}}(\widetilde{\boldsymbol{M}}_{k,t}^{(n+1)}(\widehat{\mathbf{d}}_{k,t}^{(n)}), \widehat{\mathbf{d}}_{k,t}^{(n)}) \\ &\leq Q_{\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \widehat{\mathbf{d}}_{k,t}^{(n)}}(\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \widehat{\mathbf{d}}_{k,t}^{(n)}) = J(\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \widehat{\mathbf{d}}_{k,t}^{(n)}). \end{aligned} \tag{5.8}$$

These inequalities show that J is a decreasing sequence, hence converging because it is lower bounded by zero.

In our particular case, we optimize (5.1). The property (5.6) holds since Φ is a differentiable cost function with a Lipschitzian gradient. Without loss of generality, the Lipschitz constant is normalized to 1. Applying the Descent Lemma to Φ yields:

$$\Phi(\varphi) \le \Phi(\varphi*) + \nabla\Phi(\varphi*)^{\top}(\varphi - \varphi*) + \frac{1}{2}\|\varphi - \varphi*\|^2 \tag{5.9}$$

where

$$\begin{cases} \varphi = \boldsymbol{M}_{k,t}^{(n+1)}\,\mathbf{f}(\mathbf{s},t) - \mathbf{f}(\mathbf{s}-\mathbf{d}_{k,t}^{(n+1)}, t_{\mathrm{ref}}), \\ \varphi* = \widehat{\boldsymbol{M}}_{k,t}^{(n)}\,\mathbf{f}(\mathbf{s},t) - \mathbf{f}(\mathbf{s}-\widehat{\mathbf{d}}_{k,t}^{(n)}, t_{\mathrm{ref}}) = \mathbf{e}^{(n)}(\mathbf{s},t). \end{cases} \tag{5.10}$$

This yields the following inequality:

$$\begin{aligned} J(\boldsymbol{M}_{k,t}^{(n+1)}, \mathbf{d}_{k,t}^{(n+1)}) \le & \sum_{\mathbf{s}\in\mathcal{B}_k} \Phi(\mathbf{e}^{(n)}(\mathbf{s},t)) \\ & \underbrace{\begin{aligned} &+ \sum_{\mathbf{s}\in\mathcal{B}_k} \nabla\Phi(\mathbf{e}^{(n)}(\mathbf{s},t))^{\top}(\boldsymbol{M}_{k,t}^{(n+1)}\,\mathbf{f}(\mathbf{s},t) - \mathbf{f}(\mathbf{s}-\mathbf{d}_{k,t}^{(n+1)}, t_{\mathrm{ref}}) - \mathbf{e}^{(n)}(\mathbf{s},t)) \\ &+ \frac{1}{2}\sum_{\mathbf{s}\in\mathcal{B}_k} \|\boldsymbol{M}_{k,t}^{(n+1)}\,\mathbf{f}(\mathbf{s},t) - \mathbf{f}(\mathbf{s}-\mathbf{d}_{k,t}^{(n+1)}, t_{\mathrm{ref}}) - \mathbf{e}^{(n)}(\mathbf{s},t)\|^2 \end{aligned}}_{Q_{\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \widehat{\mathbf{d}}_{k,t}^{(n)}}(\boldsymbol{M}_{k,t}^{(n+1)}, \mathbf{d}_{k,t}^{(n+1)})}, \end{aligned} \tag{5.11}$$

where $Q_{\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \widehat{\mathbf{d}}_{k,t}^{(n)}}(\boldsymbol{M}_{k,t}^{(n+1)}, \mathbf{d}_{k,t}^{(n+1)})$ is thus the majorant function of J at iteration $n+1$, and it can be simplified as follows:

$$\begin{aligned} Q_{\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \widehat{\mathbf{d}}_{k,t}^{(n)}}(\boldsymbol{M}_{k,t}^{(n+1)}, \mathbf{d}_{k,t}^{(n+1)}) = C + & \sum_{\mathbf{s}\in\mathcal{B}_k} \nabla\Phi(\mathbf{e}^{(n)}(\mathbf{s},t))^{\top}\boldsymbol{M}_{k,t}^{(n+1)}\,\mathbf{f}(\mathbf{s},t) \\ + \frac{1}{2}\sum_{\mathbf{s}\in\mathcal{B}_k} \Big(& \|\boldsymbol{M}_{k,t}^{(n+1)}\,\mathbf{f}(\mathbf{s},t)\|^2 - 2(\mathbf{f}(\mathbf{s}-\mathbf{d}_{k,t}^{(n+1)}, t_{\mathrm{ref}}) - \mathbf{e}^{(n)}(\mathbf{s},t))^{\top}\boldsymbol{M}_{k,t}^{(n+1)}\,\mathbf{f}(\mathbf{s},t)\Big) \end{aligned} \tag{5.12}$$

where C is a constant independent from $\boldsymbol{M}_{k,t}^{(n+1)}$.

This can be rewritten as

$$
\begin{aligned}
&Q_{\widehat{\boldsymbol{M}}_{k,t}^{(n)},\widehat{\mathbf{d}}_{k,t}^{(n)}}(\boldsymbol{M}_{k,t}^{(n+1)},\mathbf{d}_{k,t}^{(n+1)})\\
&\quad = C + \sum_{\mathbf{s}\in\mathcal{B}_k} \operatorname{tr}\Big[\boldsymbol{M}_{k,t}^{(n+1)}\,\mathbf{f}(\mathbf{s},t)\big(\nabla\Phi(\mathbf{e}^{(n)}(\mathbf{s},t) - \mathbf{f}(\mathbf{s}-\mathbf{d}_{k,t}^{(n+1)},t_{\mathrm{ref}}) - \mathbf{e}^{(n)}(\mathbf{s},t)\big)^\top\Big]\\
&\qquad\qquad + \frac{1}{2}\sum_{\mathbf{s}\in\mathcal{B}_k}\operatorname{tr}\big(\boldsymbol{M}_{k,t}^{(n+1)}\,\mathbf{f}(\mathbf{s},t)\,\mathbf{f}(\mathbf{s},t)^\top\,\boldsymbol{M}_{k,t}^{(n+1)^\top}\big)\\
&\quad = C + \big\langle\boldsymbol{M}_{k,t}^{(n+1)},\sum_{\mathbf{s}\in\mathcal{B}_k}\big(\nabla\Phi(\mathbf{e}^{(n)}(\mathbf{s},t) - \mathbf{f}(\mathbf{s}-\mathbf{d}_{k,t}^{(n+1)},t_{\mathrm{ref}}) - \mathbf{e}^{(n)}(\mathbf{s},t)\big)\mathbf{f}(\mathbf{s},t)^\top\big\rangle_{\mathrm{F}}\\
&\qquad\qquad + \frac{1}{2}\big\langle\boldsymbol{M}_{k,t}^{(n+1)},\boldsymbol{M}_{k,t}^{(n+1)}\sum_{\mathbf{s}\in\mathcal{B}_k}\mathbf{f}(\mathbf{s},t)\,\mathbf{f}(\mathbf{s},t)^\top\big\rangle_{\mathrm{F}}
\end{aligned}
$$

where $\langle\cdot,\cdot\rangle_{\mathrm{F}}$ is the Frobenius scalar product, defined by $\langle A,B\rangle_{\mathrm{F}} = \operatorname{tr}(A\,B^\top)$.

$Q_{\widehat{\boldsymbol{M}}_{k,t}^{(n)},\widehat{\mathbf{d}}_{k,t}^{(n)}}(\cdot,\mathbf{d}_{k,t}^{(n+1)})$ is a convex function whose gradient with respect to $\boldsymbol{M}_{k,t}^{(n+1)}$ is given by

$$
\begin{aligned}
\nabla_{\boldsymbol{M}_{k,t}^{(n+1)}} Q_{\widehat{\boldsymbol{M}}_{k,t}^{(n)},\widehat{\mathbf{d}}_{k,t}^{(n)}}(\boldsymbol{M}_{k,t}^{(n+1)},\mathbf{d}_{k,t}^{(n+1)}) &= \sum_{\mathbf{s}\in\mathcal{B}_k}\big(\nabla\Phi(\mathbf{e}^{(n)}(\mathbf{s},t)) - \mathbf{f}(\mathbf{s}-\mathbf{d}_{k,t}^{(n+1)},t_{\mathrm{ref}}) - \mathbf{e}^{(n)}(\mathbf{s},t))^\top\big)\mathbf{f}(\mathbf{s},t)^\top\\
&\quad + \boldsymbol{M}_{k,t}^{(n+1)}\sum_{\mathbf{s}\in\mathcal{B}_k}\mathbf{f}(\mathbf{s},t)\,\mathbf{f}(\mathbf{s},t)^\top, \qquad (5.13)
\end{aligned}
$$

and for every candidate vector $\mathbf{d}_{k,t}\in\mathcal{W}$, it admits as minimizer:

$$
\begin{aligned}
\widetilde{\boldsymbol{M}}_{k,t}^{(n+1)}(\mathbf{d}_{k,t}) = &\Big(\sum_{\mathbf{s}\in\mathcal{B}_k}\big(\mathbf{f}(\mathbf{s}-\mathbf{d}_{k,t},t_{\mathrm{ref}}) + \mathbf{e}^{(n)}(\mathbf{s},t) - \nabla\Phi(\mathbf{e}^{(n)}(\mathbf{s},t))\big)\mathbf{f}(\mathbf{s},t)^\top\Big)\\
&\times\Big(\sum_{\mathbf{s}\in\mathcal{B}_k}\mathbf{f}(\mathbf{s},t)\mathbf{f}(\mathbf{s},t)^\top\Big)^\dagger. \qquad (5.14)
\end{aligned}
$$

For each block, 11 scalar parameters need to be estimated: 9 for the color correction and 2 translation parameters. If estimates $(\widehat{\boldsymbol{M}}_{k,t}^{(n)},\widehat{\mathbf{d}}_{k,t}^{(n)})$ are available at iteration n, updates of the sought parameters are obtained by Algorithm 2. For any translation vector, this algorithm was shown to generate a convergent decaying sequence of energy values as a result of Property (5.8). We then select the translation vector with lowest energy obtained at convergence.

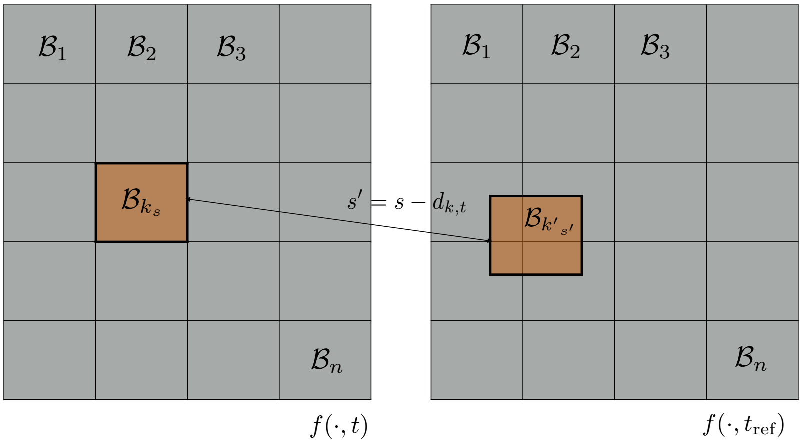


Figure 5.3. Block matching scheme

The result of this process is the transformation matrix and displacement vectors that best match the correspondence between blocks.

2.4 Choice for the cost function Φ

Typical choices for Φ are the squared Euclidean norm, yielding a standard least squares approach, the separable Huber function, which constitutes a smoothed version of the ℓ_1 norm and is useful for performing robust regression as well as smooth approximations of the ℓ_0 pseudo-norm.

2.4.1 Quadratic norm

When $\Phi = \| \cdot \|^2/2$, we obtain $\nabla\Phi(\mathbf{e}^{(n)}(\mathbf{s}, t)) = \mathbf{e}^{(n)}(\mathbf{s}, t)$ and the standard least squares solution is recovered, and so the convergence is ensured in one iteration.

2.4.2 Huber function

The Huber function is useful in robust regression. As a differentiable approximation of the ℓ_1 norm, it is well known to be less sensitive to outliers than the quadratic norm, which makes the estimation of displacement and color transformation parameters more precise, and thus block artifacts are reduced in the corrected sequence.

We can write $\mathbf{e}(\mathbf{s},t)$ in matrix form as follows:

$$\mathbf{e}(\mathbf{s},t) = \begin{bmatrix} e_R(\mathbf{s},t) \\ e_G(\mathbf{s},t) \\ e_B(\mathbf{s},t) \end{bmatrix}. \tag{5.15}$$

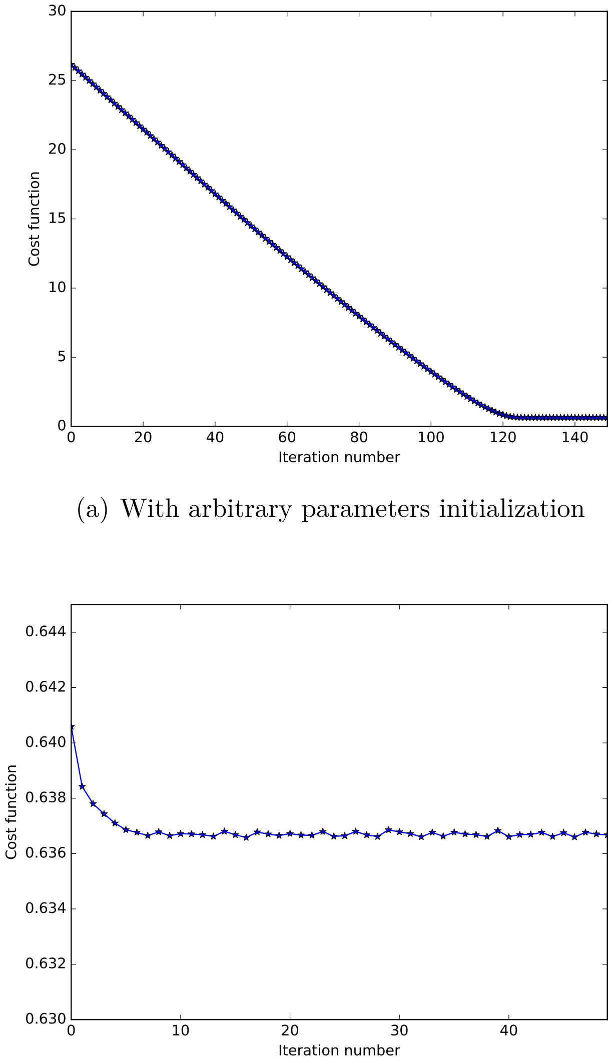
Using Peter J. Huber definition (Huber et al. (1964)):

$$\Phi_h(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \varepsilon, \\ \varepsilon(|a| - \frac{1}{2}\varepsilon), & \text{otherwise,} \end{cases} \tag{5.16}$$

the gradient of our cost function is defined by:

$$\nabla\Phi(\mathbf{e}(\mathbf{s},t)) = \begin{bmatrix} \Phi_h'(e_R(\mathbf{s},t)) \\ \\ \Phi_h'(e_G(\mathbf{s},t)) \\ \\ \Phi_h'(e_B(\mathbf{s},t)) \end{bmatrix} = \begin{bmatrix} \begin{cases} \varepsilon, & \text{if } \; e_R(\mathbf{s},t) > \varepsilon \\ -\varepsilon, & \text{if } \; e_R(\mathbf{s},t) < -\varepsilon \\ e_R(\mathbf{s},t), & \text{if } \; |e_R(\mathbf{s},t)| \leq \varepsilon \end{cases} \\ \\ \begin{cases} \varepsilon, & \text{if } \; e_G(\mathbf{s},t) > \varepsilon \\ -\varepsilon, & \text{if } \; e_G(\mathbf{s},t) < -\varepsilon \\ e_G(\mathbf{s},t), & \text{if } \; |e_G(\mathbf{s},t)| \leq \varepsilon \end{cases} \\ \\ \begin{cases} \varepsilon, & \text{if } \; e_B(\mathbf{s},t) > \varepsilon \\ -\varepsilon, & \text{if } \; e_B(\mathbf{s},t) < -\varepsilon \\ e_B(\mathbf{s},t), & \text{if } \; |e_B(\mathbf{s},t)| \leq \varepsilon \end{cases} \end{bmatrix}. \tag{5.17}$$

Using this cost function, if the initialization of parameters is performed arbitrarily, the convergence can be somewhat slow (more than 100 iterations), therefore the initialization step is very important. We proposed to estimate initial parameters, in particular the translation vectors, using the quadratic approach, and reuse them as initial values for the subsequent iterations. This procedure reduces the number of required iterations by a factor of more than 20.



(a) With arbitrary parameters initialization

(b) Initialization with the quadratic solution

Figure 5.4. The convergence of our iterative algorithm depends of the parameters initialization. This test was performed on an image for the Bird sequence.

In qualitative terms, the smaller ε, the more effective the algorithm is at reducing the outliers and the algorithm converges more slowly to an optimal best corrected solution. As shown in Figure 5.4, the optimal choice for ε appears to be around 0.2.

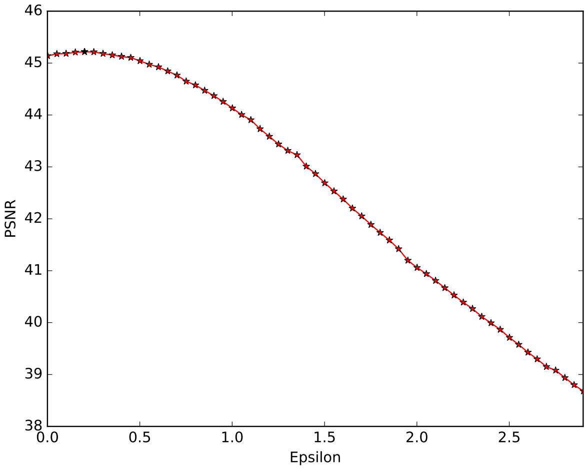


Figure 5.5. Huber norm: The PSNR of the processed image for the Bird sequence while increasing ε.

2.4.3 Other error functions

We tested our iterative optimization strategy with other more complex criteria. We will consider some non-convex functions for the majorant function, typically the Welsch and Geman-McClure approximations of the ℓ_0 pseudo-norm.

Welsch function

Using Welsch definition (Holland and Welsch (1977)), we can write:

$$\Phi(\mathbf{e}(\mathbf{s},t)) = \sum_{c\in\{R,G,B\}} \Phi_w(e_c(\mathbf{s},t)), \tag{5.18}$$

where

$$\Phi_w(e_c(\mathbf{s},t)) = \frac{\lambda_w}{2}[1 - \exp(-\frac{e_c^2(\mathbf{s},t)}{C_w^2})], \tag{5.19}$$

C_w and λ_w are constant variables. Its derivative, called influence function, is expressed as follows:

$$\Phi'_w(e_c(\mathbf{s},t)) = \frac{\lambda_w}{C_w^2} e_c(\mathbf{s},t) \exp\left(-\frac{e_c^2(\mathbf{s},t)}{C_w^2}\right), \tag{5.20}$$

and its second-order derivative is given by

$$\Phi''_w(e_c(\mathbf{s},t)) = \frac{\lambda_w}{C_w^2}\left(1 - \frac{2e_c^2(\mathbf{s},t)}{C_w^2}\right) \exp\left(-\frac{e_c^2(\mathbf{s},t)}{C_w^2}\right). \tag{5.21}$$

λ_w is fixed while ensuring that the Lipschitz constant L_w of the derivative (Hessian maximum value) is normalized to 1:

$$\begin{aligned} L_w = \Phi''_w(0) = \frac{\lambda_w}{C_w^2} = 1 \\ \Rightarrow \lambda_w = C_w^2. \end{aligned} \tag{5.22}$$

According to our experiments, the best choice for the parameter C_w is then given by $C_w = 0.15$.

Geman-McClure function

We test also the Geman-McClure M-estimator proposed by Geman and McClure (1987). It is defined as:

$$\Phi(\mathbf{e}(\mathbf{s},t) = \sum_{c\in\{R,G,B\}} \Phi_g(e_c(\mathbf{s},t)), \tag{5.23}$$

where

$$\Phi_g(e_c(\mathbf{s},t)) = \lambda_g \frac{e_c^2(\mathbf{s},t)}{2{C_g}^2 + e_c^2(\mathbf{s},t)}, \tag{5.24}$$

$$\Phi'_g(e_c(\mathbf{s},t)) = \lambda_g \frac{4e_c(\mathbf{s},t){C_g}^2}{(2{C_g}^2 + e_c^2(\mathbf{s},t))^2}, \tag{5.25}$$

$$\Phi''_g(e_c(\mathbf{s},t)) = 4\lambda_g C_g^2 \frac{2C_g^2 - 3e_c^2(\mathbf{s},t)}{(2C_g^2 + e_c^2(\mathbf{s},t))^3}. \tag{5.26}$$

λ_g must also be chosen so as to ensure that the Lipschitz constant L_g is normalized to 1:

$$\begin{aligned} L_g = \Phi_g''(0) = \frac{\lambda_g}{C_g^2} = 1 \\ \Rightarrow \lambda_g = C_g^2 \end{aligned} \tag{5.27}$$

The choice of the parameter which appears to provide the best results when running the MM algorithm is $C_g = 0.11$.

2.4.4 Comparison between different cost functions

In order to test the performance of the color correction strategy using our iterative algorithm with different cost functions, a PSNR similarity measure is performed on the processing results of a synthetic flicker sequences produced from a flicker-free, naturally lit video. We constructed the majorant function Q using the different approximations of norms/pseudo-norms with convexity (quadratic and Huber norms) and non-convexity properties (Welsch and Geman-McClure norms).

It is noted that Huber function provides the best PSNR ratio comparing to the other norms. It allows us to gain approximately 4 dB along the processed sequence (Figure 5.6). This is due to its robustness to limiting the influence of outliers in the observed data, and its hybridity of squared error for relatively small errors and absolute error for relative large ones.

2.5 Flicker compensation

Once the optimal color transform between the reference and the current frames has been found for each block, it is applied on each. This yields a flicker-compensated frame. Reproducing the same procedure for each frame yields a flicker-compensated sequence.

This procedure is pretty simple, however there is no expectation for the optimization procedure that we have just described to yield similar color-correction

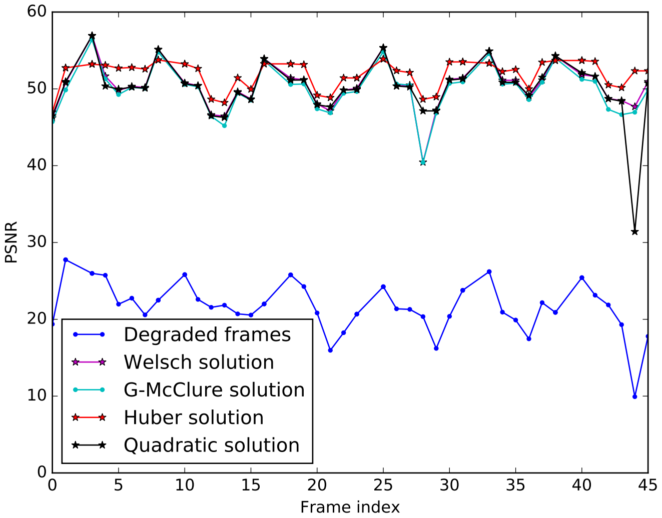


Figure 5.6. Similarity comparison: PSNR average of the processed sequence using different cost functions. The sequence is acquired with large local motion and a periodic artificial flicker has been added.

matrices for neighboring blocks. As a result, the boundary between blocks may become perceptible in the flicker-compensated results. This may happen for instance when the translation parameters are not estimated precisely enough due to the arbitrary process of image division into blocks, which implies that some blocks may include strong intensity edges or outliers. A variable-size block matching could be implemented to alleviate this issue, but at the expense of an increased complexity. For this purpose, a post processing step is required.

2.6 Block artifact removal

Following the flicker compensation step, we propose to re-process each frame of the affected video to remove block artifacts in an efficient manner. We base our approach on the color transformation matrices and translation parameters that we have already found. A simple idea could be to interpolate the color matrices near the boundaries of the blocks, however, we have found that this method does not yield accurate enough results in practice.

Instead, we re-estimate a new color correction matrix for each pixel in a frame, taking into account its neighborhood and its position in the block relative to neighboring blocks. For each pixel $\mathbf{s}$, we reuse the pair $(\widehat{\boldsymbol{M}}_{k_{\mathbf{s}},t},\ \widehat{\mathbf{d}}_{k_{\mathbf{s}},t})$ already estimated for its corresponding block. In the optimization process, we now include a regularization term weighted by a constant $\lambda \in]0,+\infty[$. A new energy function is then defined as

$$K(\boldsymbol{M}_{\mathbf{s},t}) = \sum_{\mathbf{s}'\in\mathcal{B}_{k_{\mathbf{s}}}} \Phi\big(\boldsymbol{M}_{\mathbf{s},t}\,\mathbf{f}(\mathbf{s}',t) - \mathbf{f}(\mathbf{s}' - \widehat{\mathbf{d}}_{k_{\mathbf{s}},t}, t_{\text{ref}})\big) + \underbrace{\frac{\lambda}{2}\sum_{\mathbf{s}'\in\mathcal{V}_{\mathbf{s}}} \|\boldsymbol{M}_{\mathbf{s},t} - \widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t}\|_{\mathrm{F}}^2}_{\mathbf{R}_{\mathbf{g}}(\boldsymbol{M}_{s,t})}, \tag{5.28}$$

where $\mathbf{s}$ is the position of the pixel to be processed, $\mathcal{V}_{\mathbf{s}}$ is a neighbourhood of it, and $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. The previous optimization problem can also be solved by the iterative Majorize-Minimize strategy, since the Descent Lemma 5.6 holds again. We construct a new majorant function called $\mathbf{P}$ defined as follows:

$$\mathbf{P} = \mathbf{Q} + \mathbf{R}_{\mathbf{g}}. \tag{5.29}$$

At an iteration $n+1$, the minimizer of (5.29) is given by setting to zero the derivative of $\mathbf{P}$ with respect to $\boldsymbol{M}_{\mathbf{s},t}$. The derivative function can be expressed by

$$\nabla_{\boldsymbol{M}^{(n+1)}_{\mathbf{s},t}} \mathbf{P}_{\widehat{\boldsymbol{M}}^{(n)},\widehat{d}^{(n)}}(\boldsymbol{M}^{(n+1)}_{\mathbf{s},t}) = \nabla_{\boldsymbol{M}^{(n+1)}_{\mathbf{s},t}} \mathbf{Q}_{\widehat{\boldsymbol{M}}^{(n)},\widehat{d}^{(n)}}(\boldsymbol{M}^{(n+1)}_{\mathbf{s},t}) + \nabla_{\boldsymbol{M}^{(n+1)}_{\mathbf{s},t}} \mathbf{R}_{\mathbf{g}}(\boldsymbol{M}^{(n+1)}_{k,t}). \tag{5.30}$$

$\mathbf{R}_{\mathbf{g}}$ is expanded as follows,

$$\mathbf{R}_{\mathbf{g}}(\boldsymbol{M}_{\mathbf{s},t}) = \frac{\lambda}{2}\sum_{\mathbf{s}'\in\mathcal{V}_{\mathbf{s}}} \big(\|\boldsymbol{M}_{\mathbf{s},t}\|_{\mathrm{F}}^2 - 2 < \widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t}, \boldsymbol{M}_{\mathbf{s},t} >_{\mathrm{F}} + \|\widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t}\|_{\mathrm{F}}^2\big) \tag{5.31}$$

This step facilitates the computation of the derivative of regularization term at an iteration $n+1$,

$$\nabla_{\boldsymbol{M}^{(n+1)}_{\mathbf{s},t}} \mathbf{R}_{\mathbf{g}}(\boldsymbol{M}^{(n+1)}_{\mathbf{s},t}) = \lambda \sum_{\mathbf{s}'\in\mathcal{V}_{\mathbf{s}}} \boldsymbol{I}_3\,\boldsymbol{M}^{(n+1)}_{\mathbf{s},t} - \lambda \sum_{\mathbf{s}'\in\mathcal{V}_{\mathbf{s}}} \widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t} \tag{5.32}$$

Refering to (5.13) and (5.32), (5.30) becomes:

$$
\begin{aligned}
&\nabla_{\boldsymbol{M}^{(n+1)}_{\mathbf{s},t}} \mathbf{P}_{\widehat{\boldsymbol{M}}^{(n)},\widehat{\boldsymbol{d}}^{(n)}}(\boldsymbol{M}^{(n+1)}_{\mathbf{s},t}) = \\
&\sum_{\mathbf{s}\in\mathcal{B}_k} \Big(\nabla\Phi(\mathbf{e}^{(n)}(\mathbf{s},t)) - \mathbf{f}(\mathbf{s}-\mathbf{d}^{(n+1)}_{\mathbf{s},t}, t_{\text{ref}}) - \mathbf{e}^{(n)}(\mathbf{s},t))^{\top}\Big)\mathbf{f}(\mathbf{s},t)^{\top} + \boldsymbol{M}^{(n+1)}_{\mathbf{s},t} \sum_{\mathbf{s}\in\mathcal{B}_k} \mathbf{f}(\mathbf{s},t)\,\mathbf{f}(\mathbf{s},t)^{\top} \\
&\qquad + \lambda \sum_{\mathbf{s}'\in\mathcal{V}_{\mathbf{s}}} \boldsymbol{I}_3\, \boldsymbol{M}^{(n+1)}_{\mathbf{s},t} - \lambda \sum_{\mathbf{s}'\in\mathcal{V}_{\mathbf{s}}} \widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t} \qquad (5.33)
\end{aligned}
$$

with $\mathbf{d}^{(n)}_{s,t} = \mathbf{d}^{(n+1)}_{\mathbf{s},t} = \widehat{\mathbf{d}}_{k_s,t}$. Equation (5.33) thus admits as minimizer:

$$
\begin{aligned}
\check{\boldsymbol{M}}^{(n+1)}_{\mathbf{s},t} &= \Big(\sum_{\mathbf{s}'\in\mathcal{B}_{k_{\mathbf{s}}}} \big(\mathbf{f}(\mathbf{s}' - \widehat{\mathbf{d}}_{k_{\mathbf{s}},t}, t_{\text{ref}}) + \mathbf{e}^{(n)}(\mathbf{s}',t) \quad - \nabla\Phi(\mathbf{e}^{(n)}(\mathbf{s}',t))\big)\mathbf{f}(\mathbf{s}',t)^{\top} + \\
&\lambda \sum_{\mathbf{s}'\in\mathcal{V}_{\mathbf{s}}} \widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t}\Big)\Big(\sum_{\mathbf{s}'\in\mathcal{B}_{k_{\mathbf{s}}}} \mathbf{f}(\mathbf{s}',t)\mathbf{f}(\mathbf{s}',t)^{\top} + \lambda|\mathcal{V}_{\mathbf{s}}|\boldsymbol{I}_3\Big)^{-1}, \qquad (5.34)
\end{aligned}
$$

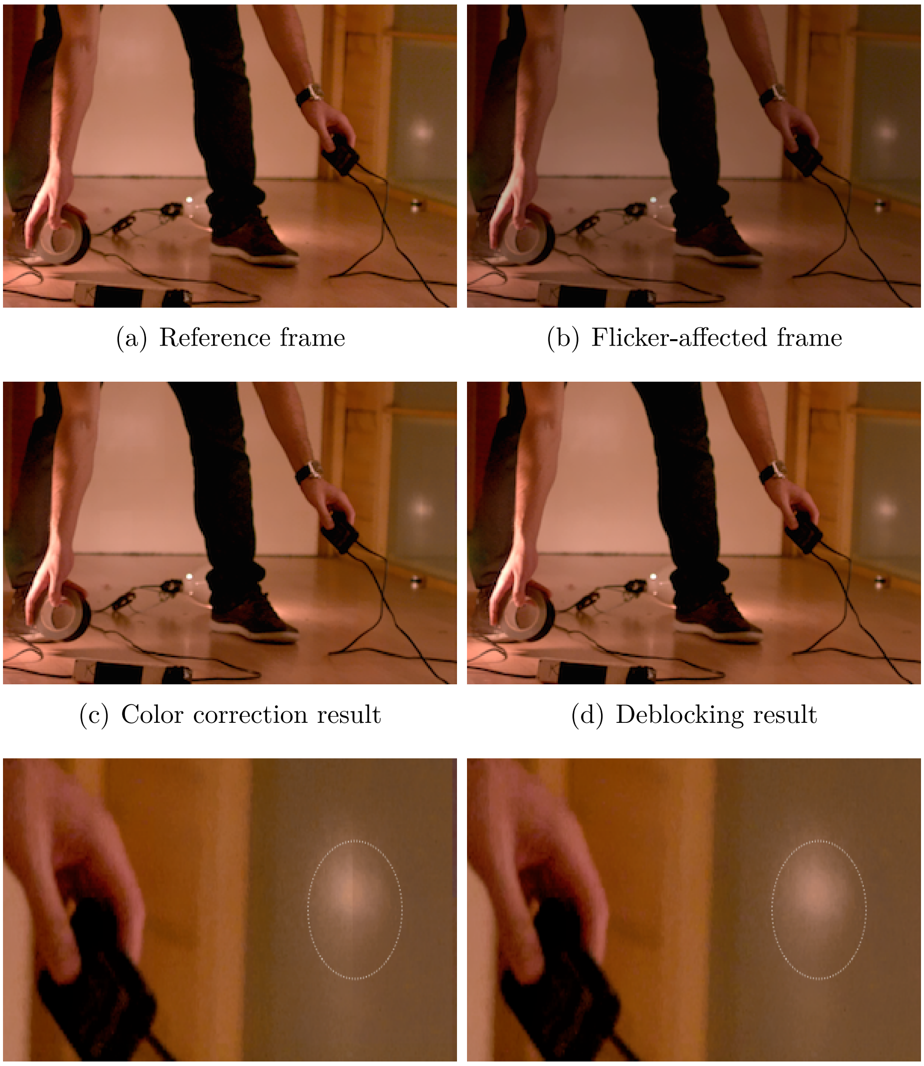
where

$$
\mathbf{e}^{(n)}(\mathbf{s}',t) = \check{\boldsymbol{M}}^{(n)}_{\mathbf{s},t}\, \mathbf{f}(\mathbf{s}',t) - \mathbf{f}(\mathbf{s}' - \widehat{\mathbf{d}}_{k_{\mathbf{s}},t}, t_{\text{ref}}). \qquad (5.35)
$$

Note that this formulation ensures that, if we set λ to zero, we recover the previous solution computed on block $\mathcal{B}_{k_{\mathbf{s}}}$.

2.7 An outline

This causal method compensates well the periodic flicker in an image with respect to a previous reference frame. When flicker is not uniform in some image regions, for some reasons (different light source, higher noise level, etc..), the flicker layout of Region 2 in Figure 5.1 varies between areas, the reference frames along the sequence have different illumination averages. This creates some local luminosity distortions between flicker periods in the corrected sequence. In the next section, we propose a new method which is suitable for aperiodic flicker removal.



(a) Reference frame

(b) Flicker-affected frame

(c) Color correction result

(d) Deblocking result

(e) Zoom before deblocking

(f) Zoom after deblocking

Figure 5.7. Image sequence with periodic flicker from studio lighting.

3 Non-causal method using two reference sources

As seen in chapter 1, most state of the art methods use a single reference to process the current frame. Some consider the previous corrected frame as a reference to maintain constancy in luminosity variation, and especially when processing non-periodic flicker, this approach may accumulate the restoration errors from one image to another.

In this context, in order to avoid luminosity distortions between flicker periods in the corrected sequences, we propose a new strategy consisting of processing each frame with respect to a preceeding and a following reference frames, depending on its position between these two, this approach is called "non-causal" because it uses both past and future frames. Obviously it can only be used offline, so the moniker "non-causal" is only descriptive. As already explained in Section 4.1, previous and next reference frames are respectively the last and next maximum intensity frames in the current period, to be more reliable which are better in terms of signal-to-acquisition noise ratio and in terms of matching similar contents.

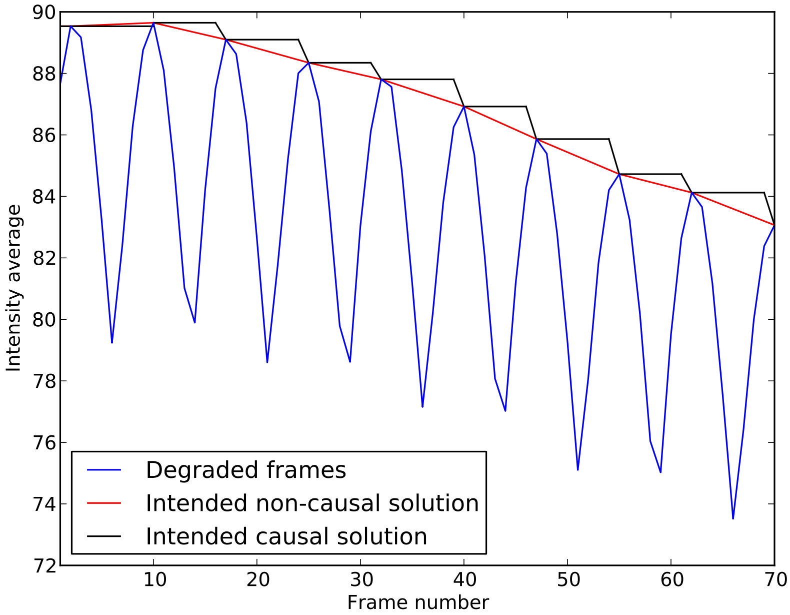


Figure 5.8. Blue curve shows the brightness variation of images in a flicker affected sequence. In black, we display the intensity averages of the corrected frames using a single reference image by flicker period (causal approach). The red curve shows the desired solution that better aligns brightness levels using two reference frames through a non-causal approach.

3.1 Tracking step

We continue to use the same joint block-matching / color transform estimation principle. More precisely, we find for each spatial block $\mathcal{B}_k$ indexed by k its two similar blocks in the previous and next reference frames respectively, and we assign to this block its own model parameters:

$$\boldsymbol{M}_{k,t} = \alpha_t \boldsymbol{M}_{k,t/t_{\mathrm{ref1}}} + (1 - \alpha_t)\boldsymbol{M}_{k,t/t_{\mathrm{ref2}}}, \tag{5.36}$$

where $\alpha_t = \frac{t_{\mathrm{ref2}}-t}{t_{\mathrm{ref2}}-t_{\mathrm{ref1}}}$ is a linear interpolation factor, $\boldsymbol{M}_{k,t/t_{\mathrm{ref1}}}$ and $\boldsymbol{M}_{k,t/t_{ref2}}$ are the same correction matrices for all pixels of the block $\mathcal{B}_k$, which are computed with respect to their similar blocks in the two reference sources at instants t_{ref1} et t_{ref2}.

Following the same methodology as described above, for a block $\mathcal{B}_k$ in a frame at an instant t, we minimise the following energy function:

$$J(\boldsymbol{M}_{k,t/t_{\mathrm{ref}i}}, \mathbf{d}_{k,t/t_{\mathrm{ref}i}}) = \sum_{\mathbf{s}' \in \mathcal{B}_k} \Phi\big(\boldsymbol{M}_{k,t/t_{\mathrm{ref}i}} \mathbf{f}(\mathbf{s}', t) - \mathbf{f}(\mathbf{s}' - \mathbf{d}_{k,t/t_{\mathrm{ref}i}}, t_{\mathrm{ref}i})\big), \tag{5.37}$$

where $i \in \{1, 2\}$, $\mathbf{d}_{k,t/t_{\mathrm{ref}i}}$ is a translation vector between two spatial blocs corresponding to instants t and $t_{\mathrm{ref}i}$ and $\Phi \colon \mathbb{R}^3 \to [0, +\infty[$ is a differentiable cost function with a Lipschitzian gradient.

Although various choices for Φ are possible, the simplest one is $\Phi = \| \cdot \|^2/2$ leading to a least squares approach. Similarly to the previous section, the Huber error function is also a suitable choice.

For each block in the current image, two color correction matrices (each comprising 9 parameters) and two translation vectors are estimated with respect to the two associated reference frames, the final color correction parameters are estimated by a linear interpolation using the factor α_t.

In order to solve the previous optimization problem, we note that, for a fixed

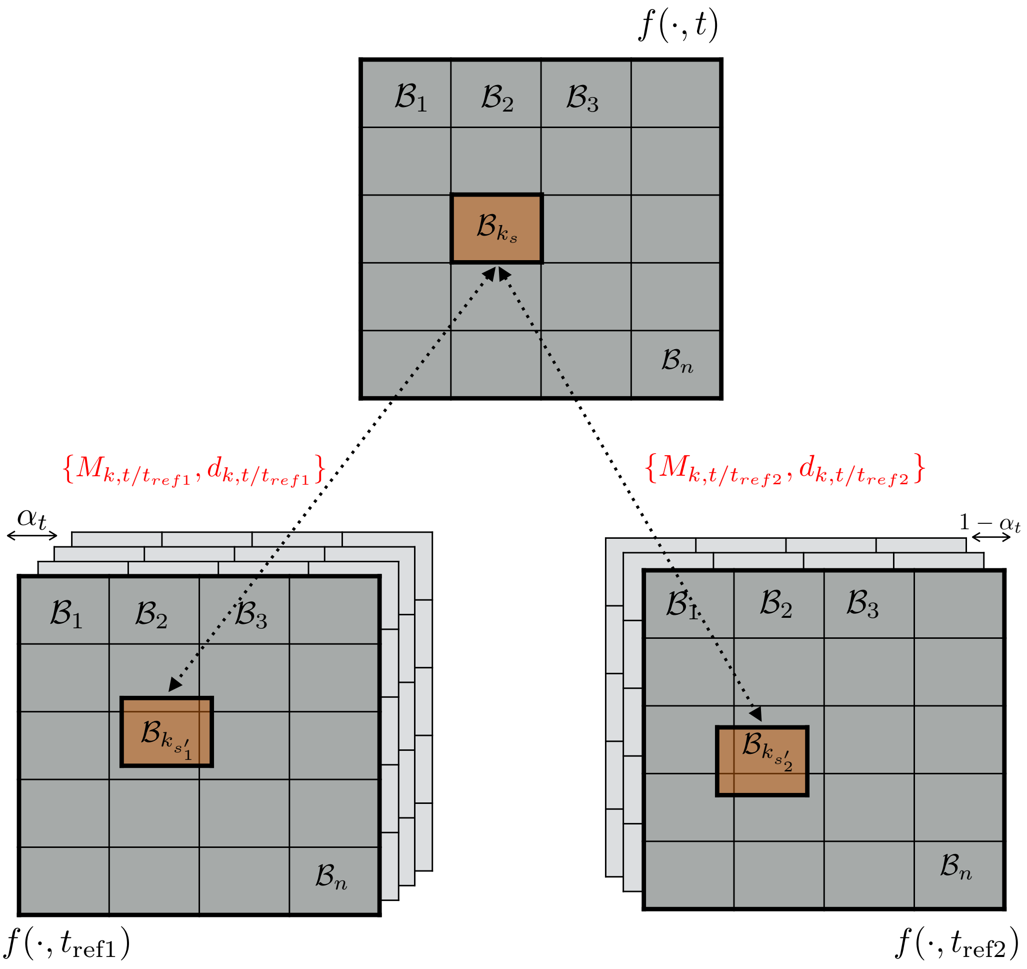


Figure 5.9. Two reference sources tracking method

$\mathbf{d}_{k,t/t_{\text{ref}i}}$ the least squares solution is

$$\widetilde{\boldsymbol{M}}_{k,t/t_{\text{ref}i}}(\mathbf{d}_{k,t/t_{\text{ref}i}}) = \Big(\sum_{\mathbf{s}'\in\mathcal{B}_k} \mathbf{f}(\mathbf{s}'-\mathbf{d}_{k,t/t_{\text{ref}i}}, t_{\text{ref}i})\mathbf{f}(\mathbf{s}',t)^\top\Big)\boldsymbol{R}^\dagger_{k,t} \tag{5.38}$$

where $\boldsymbol{R}_{k,t} = \sum_{\mathbf{s}'\in\mathcal{B}_k} \mathbf{f}(\mathbf{s}',t)\mathbf{f}(\mathbf{s}',t)^\top$ and $(\cdot)^\dagger$ is the pseudo-inverse operator.

It follows that the optimal pair $\big(\widehat{\boldsymbol{M}}_{k,t/t_{\text{ref1}}}, \widehat{\boldsymbol{M}}_{k,t/t_{\text{ref2}}}\big)$ is equal to $\big(\widetilde{\boldsymbol{M}}_{k,t/t_{\text{ref1}}}(\widehat{\mathbf{d}}_{k,t/t_{\text{ref1}}}),$ $\widetilde{\boldsymbol{M}}_{k,t/t_{\text{ref2}}}(\widehat{\mathbf{d}}_{k,t/t_{\text{ref2}}})\big)$ where $\widehat{\mathbf{d}}_{k,t/t_{\text{ref}i}}$, $i\in\{1,2\}$, is the minimizer, in a finite search window, of

$$\mathbf{d}_{k,t/t_{\mathrm{ref}i}} \mapsto J(\widetilde{\boldsymbol{M}}_{k,t/t_{\mathrm{ref}i}}(\mathbf{d}_{k,t/t_{\mathrm{ref}i}}), \mathbf{d}_{k,t/t_{\mathrm{ref}i}}). \tag{5.39}$$

3.2 Block artifact removal step

For each pixel $\mathbf{s}$ in the block $\mathcal{B}_{k_{\mathbf{s}}}$, we re-use the two pairs $\big(\widehat{\boldsymbol{M}}_{\mathbf{s},t/t_{\mathrm{ref1}}}, \widehat{\mathbf{d}}_{k_{\mathbf{s}},t/t_{\mathrm{ref1}}}\big)$ and $\big(\widehat{\boldsymbol{M}}_{\mathbf{s},t/t_{\mathrm{ref2}}}, \widehat{\mathbf{d}}_{k_{\mathbf{s}},t/t_{\mathrm{ref2}}}\big)$ already found. As in the artifact removal step for the causal method, we include to the optimization process a regularization term weighted by a multiplicative constant $\lambda \in]0,+\infty[$.

The new energy function is defined by

$$\begin{aligned} K(\boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref1}}}, \boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref2}}}) = & \sum_{\mathbf{s}' \in \mathcal{B}_{k_{\mathbf{s}}}} \Phi\big(\boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref1}}} \mathbf{f}(\mathbf{s}',t) - \mathbf{f}(\mathbf{s}' - \mathbf{d}_{k_{\mathbf{s}},t/t_{\mathrm{ref1}}}, t_{\mathrm{ref1}})\big) \\ & + \sum_{\mathbf{s}' \in \mathcal{B}_{k_{\mathbf{s}}}} \Phi\big(\boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref2}}} \mathbf{f}(\mathbf{s}',t) - \mathbf{f}(\mathbf{s}' - \mathbf{d}_{k_{\mathbf{s}},t/t_{\mathrm{ref2}}}, t_{\mathrm{ref2}})\big) \\ & + \frac{\lambda}{2} \sum_{\mathbf{s}' \in \mathcal{V}_{\mathbf{s}}} \|\alpha_t \boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref1}}} + (1-\alpha_t)\boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref2}}} - \widehat{\boldsymbol{M}}_{\mathbf{s}',t}\|_{\mathrm{F}}^2, \end{aligned} \tag{5.40}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. $\widehat{\boldsymbol{M}}_{\mathbf{s}',t}$ is the estimate of color correction matrix for the block containing the pixel $\mathbf{s}'$.

When $\Phi = \|\cdot\|^2/2$, this quadratic optimization problem is solved by setting to zero the partial derivatives with respect to the two correction matrices $\boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref1}}}$ and $\boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref2}}}$ which we can estimate again. This solution is given by

$$\begin{aligned} \begin{bmatrix} \boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref1}}} \\ \boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref2}}} \end{bmatrix} = & \begin{bmatrix} \sum_{\mathbf{s}' \in \mathcal{B}_{k_{\mathbf{s}}}} \mathbf{f}(\mathbf{s}' - \mathbf{d}_{k_{\mathbf{s}},t/t_{\mathrm{ref1}}}, t_{\mathrm{ref1}}) \mathbf{f}(\mathbf{s}',t)^{\top} + \lambda\alpha_t \sum_{\mathbf{s}' \in \mathcal{V}_{\mathbf{s}}} \widehat{\boldsymbol{M}}_{\mathbf{s}',t} \\ \sum_{\mathbf{s}' \in \mathcal{B}_{k_{\mathbf{s}}}} \mathbf{f}(\mathbf{s}' - \mathbf{d}_{k_{\mathbf{s}},t/t_{\mathrm{ref2}}}, t_{\mathrm{ref2}}) \mathbf{f}(\mathbf{s}',t)^{\top} + \lambda(1-\alpha_t) \sum_{\mathbf{s}' \in \mathcal{V}_{\mathbf{s}}} \widehat{\boldsymbol{M}}_{\mathbf{s}',t} \end{bmatrix} \\ & \times \begin{bmatrix} \boldsymbol{R}_{k_{\mathbf{s}},t} + \lambda\alpha_t^2 |\mathcal{V}_s| \boldsymbol{I}_3 & \lambda\alpha_t(1-\alpha_t)|\mathcal{V}_s|\boldsymbol{I}_3 \\ \lambda\alpha_t(1-\alpha_t)|\mathcal{V}_s|\boldsymbol{I}_3 & \boldsymbol{R}_{k_{\mathbf{s}},t} + \lambda(1-\alpha_t)^2|\mathcal{V}_s|\boldsymbol{I}_3 \end{bmatrix}^{-1}. \end{aligned} \tag{5.41}$$

It is noted that this formulation ensures that if we set λ to zero, we recover the previous solution estimated for the block $\mathcal{B}_{k_\mathbf{s}}$.

The comparison results with causal approach are shown and discussed in the final section of this chapter. Due to the important amount of processing time required by block matching algorithm, we propose in the next section a multiresolution approach for flicker correction.

4 Pyramidal approach to accelerate processing

Processing time is particularly important in video processing, and thus for high speed imaging, it is a challenge to meet real time constraints during the processing of long image sequences. Block matching and the pixel-wise block artifact removal algorithms are more of a time-wise challenge, and especially when we are processing HD images (1920 × 1080 pixels). So, we propose a multi-resolution algorithm to accelerate the tracking step.

4.1 The adopted strategy

In image processing, pyramidal approaches are multiresolution representations of images. These strategies are often used for image segmentation.

4.2 Algorithm

In order to process a current image with respect to a previous reference, (without loss of generality, we consider here only the causal approach), for each block $\mathcal{B}_k$, we estimate first the translation parameters represented by $\widehat{\mathbf{d}}_k^{S_N} = (\widehat{d}_{k_x}^{S_N}, \widehat{d}_{k_y}^{S_N})$ between the two images at the most reduced scale ($S_N = \frac{1}{2^N}$), using a small search grid for the displacement vector, say for instance $((-2,-2), ..., (+2,+2))$. At the next higher resolution $S_{N-1} = \frac{1}{2^{N-1}}$, these estimated parameters become the new centers of the search grid $((2\widehat{d}_{k_x}^{S_N} - 2, 2\widehat{d}_{k_y}^{S_N} - 2), ..., (2\widehat{d}_{k_x}^{S_N} + 2, 2\widehat{d}_{k_y}^{S_N} + 2))$, which are used to improve the estimate of the displacement vector. We repeat this process at all the scale levels up to scale $S_0 = \frac{1}{2^0} = 1$, to find the final estimation of displacement vector and color transformation parameters $(\widehat{\mathbf{d}}_k^{S_0}, \widehat{\boldsymbol{M}}_k^{S_0})$ in the current original frame. Figure 5.10 illustrates the proposed approach.

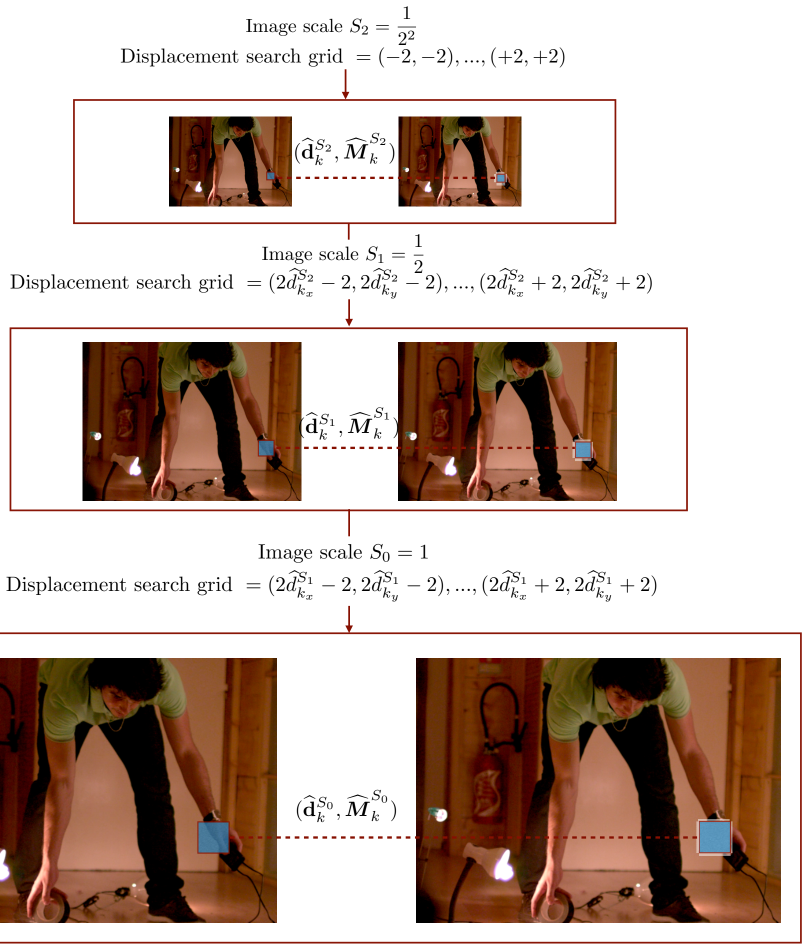


Image scale $S_2 = \frac{1}{2^2}$
Displacement search grid $= (-2, -2), ..., (+2, +2)$
$(\widehat{\mathbf{d}}_k^{S_2}, \widehat{\boldsymbol{M}}_k^{S_2})$
Image scale $S_1 = \frac{1}{2}$
Displacement search grid $= (2\widehat{d}_{k_x}^{S_2} - 2, 2\widehat{d}_{k_y}^{S_2} - 2), ..., (2\widehat{d}_{k_x}^{S_2} + 2, 2\widehat{d}_{k_y}^{S_2} + 2)$
$(\widehat{\mathbf{d}}_k^{S_1}, \widehat{\boldsymbol{M}}_k^{S_1})$
Image scale $S_0 = 1$
Displacement search grid $= (2\widehat{d}_{k_x}^{S_1} - 2, 2\widehat{d}_{k_y}^{S_1} - 2), ..., (2\widehat{d}_{k_x}^{S_1} + 2, 2\widehat{d}_{k_y}^{S_1} + 2)$
$(\widehat{\mathbf{d}}_k^{S_0}, \widehat{\boldsymbol{M}}_k^{S_0})$

Figure 5.10. Multi resolution scheme with the most reduced scale is $S_N = \frac{1}{4}$.

The complexity of the proposed methods can be represented with respect to multiple criteria. To perform the block matching-based method, the complexity of the used method depends of the search grid area around the target block and the size of this block. In order to deduce the computation time gain, we can compare the complexity of the pyramidal approach with respect to our standard causal block matching approach.

Considering a block $\mathcal{B}_k$ in the current original frame (at the full resolution), the complexity of applying the causal method on this block can be written as:

$$C_o \propto n \cdot m \cdot w \cdot h \tag{5.42}$$

where n, m are the horizontal and vertical search pixels interval respectively, and w, h are the width and height sizes (in pixel) of the processed block (Figure 5.11).

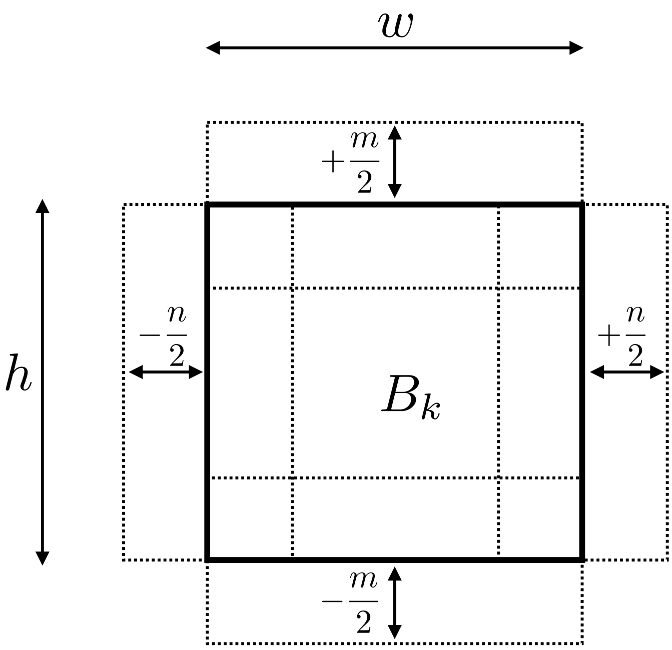


Figure 5.11. An illustration of a block B_k in the current image.

Using the pyramidal approach, the complexity of processing such a block depends on the complexities of processing this block at the different image resolutions. We note that the horizontal and vertical search intervals are constant at all resolution levels. Compared with the full-resolution search interval, the search interval can be reduced by a factor 2^c, where c is an arbitrary small positive integer (Figure 5.12).

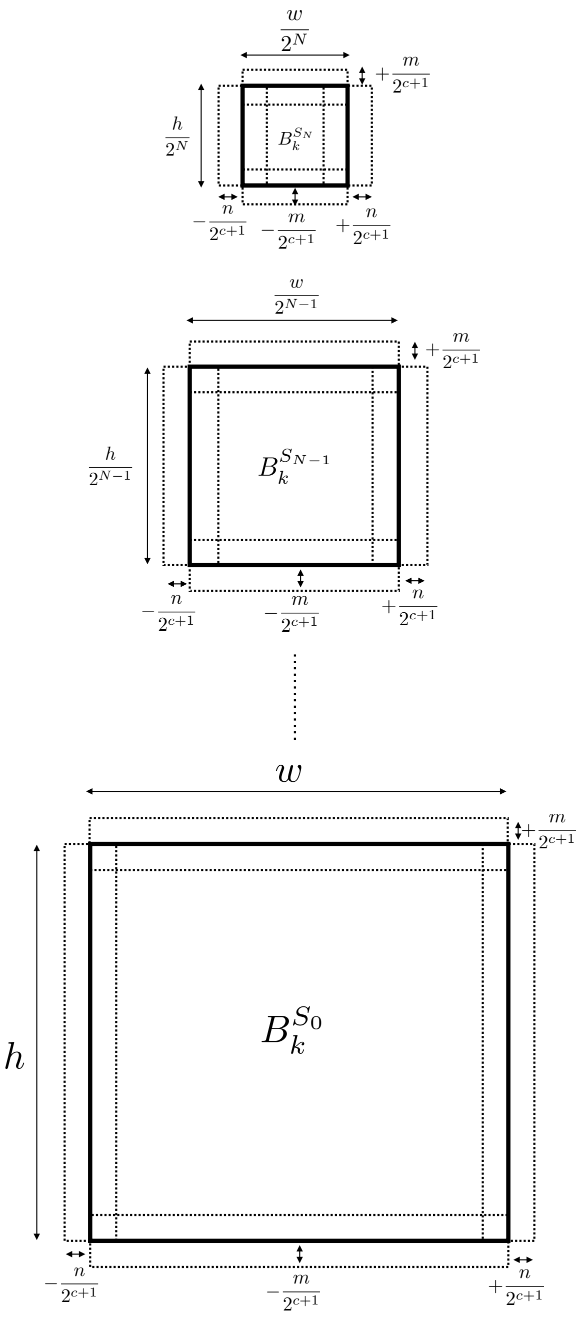


Figure 5.12. The choice of displacement search grid in pyramidal approach.

The complexities at all reduced scales, are presented below:

$$\begin{aligned}
C_0 &\propto \frac{n}{2^c} \cdot \frac{m}{2^c} \cdot w \cdot h \\
C_1 &\propto \frac{n}{2^c} \cdot \frac{m}{2^c} \cdot \frac{w}{2} \cdot \frac{h}{2} \\
C_2 &\propto \frac{n}{2^c} \cdot \frac{m}{2^c} \cdot \frac{w}{2^2} \cdot \frac{h}{2^2} \\
&\vdots \\
C_N &\propto \frac{n}{2^c} \cdot \frac{m}{2^c} \cdot \frac{w}{2^N} \cdot \frac{h}{2^N}
\end{aligned} \tag{5.43}$$

The total complexity of the pyramidal approach is:

$$\begin{aligned}
C_p &= C_0 + C_1 + \cdots + C_N \\
\Rightarrow C_p &\propto \frac{nm}{2^{2c}} \cdot w \cdot h \sum_{i=0}^{N} \frac{1}{2^{2i}}.
\end{aligned} \tag{5.44}$$

Then the improvement of the processing time of pyramidal approach with respect to the standard block matching based algorithm can be given by

$$\begin{aligned}
g_r = \frac{C_o}{C_p} &= \frac{n \cdot m \cdot w \cdot h}{\frac{nm}{2^{2c}} \cdot w \cdot h \sum_{i=0}^{N} \frac{1}{2^{2i}}} \\
&= \frac{2^{2c}}{\sum_{i=0}^{N} \frac{1}{2^{2i}}}.
\end{aligned} \tag{5.45}$$

The denominator's term in (5.45) is a geometric series, and can be expressed as follows:

$$\sum_{i=0}^{N} \frac{1}{2^{2i}} = \sum_{i=0}^{N} \frac{1}{4^i} = \frac{1 - \frac{1}{4^{N+1}}}{1 - \frac{1}{4}} = \frac{4}{3}\left(1 - \frac{1}{4^{N+1}}\right), \tag{5.46}$$

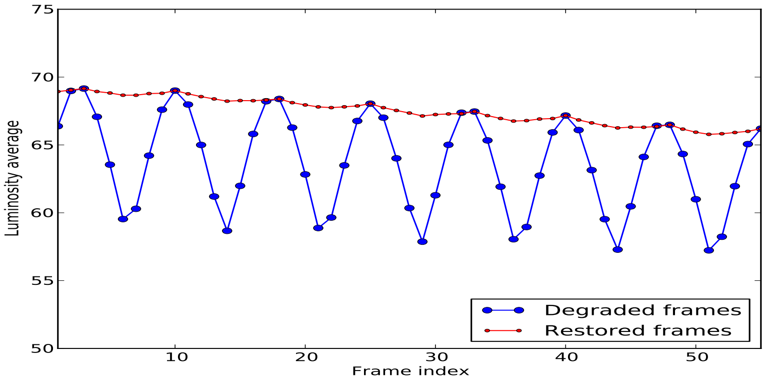
and thus Equation (5.45) becomes:

$$g_r = \frac{3 \cdot 4^{c-1}}{1 - \frac{1}{4^{N+1}}}. \tag{5.47}$$

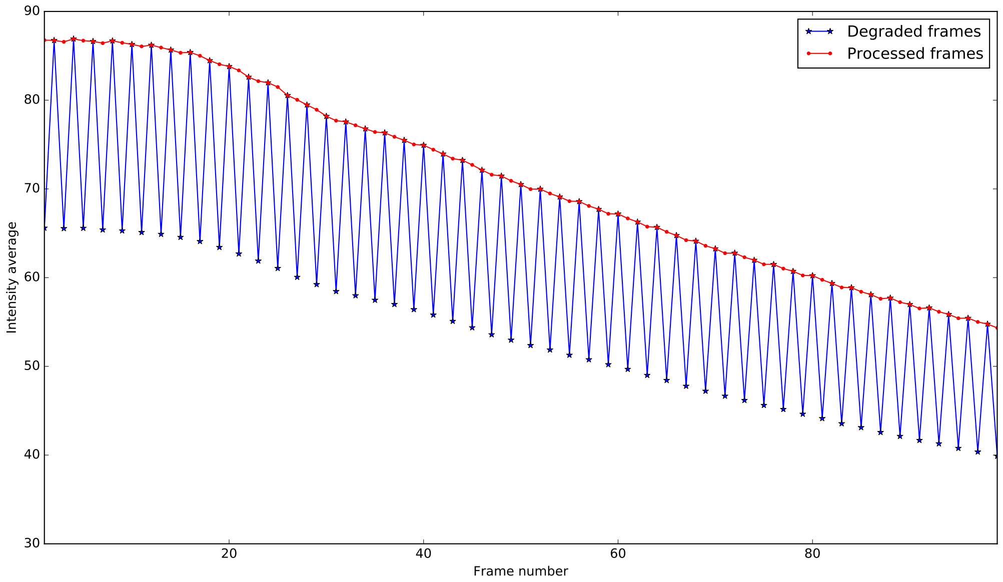
In our context, we often process HD images, we find it is sufficient to reduce the resolution twice ($N = 2$), i.e. up to $S_N = \frac{1}{2^2}$. For simplicity we choose to reduce the search interval by the same factor by choosing $c = 2$, which is consistent with the case already studied in Figure 5.10 where $n = m = 16$ and the constant search interval is a range of four pixels (i.e. ±2). Note that this is not mandatory. This yields a more than 12 fold improvement in speed (256/21). The larger the images, the more we can increase c, and so the larger the gain g_r. g_r does not depend much on N, however, we may have to increase N nonetheless if displacements are large.

5 Results and discussion

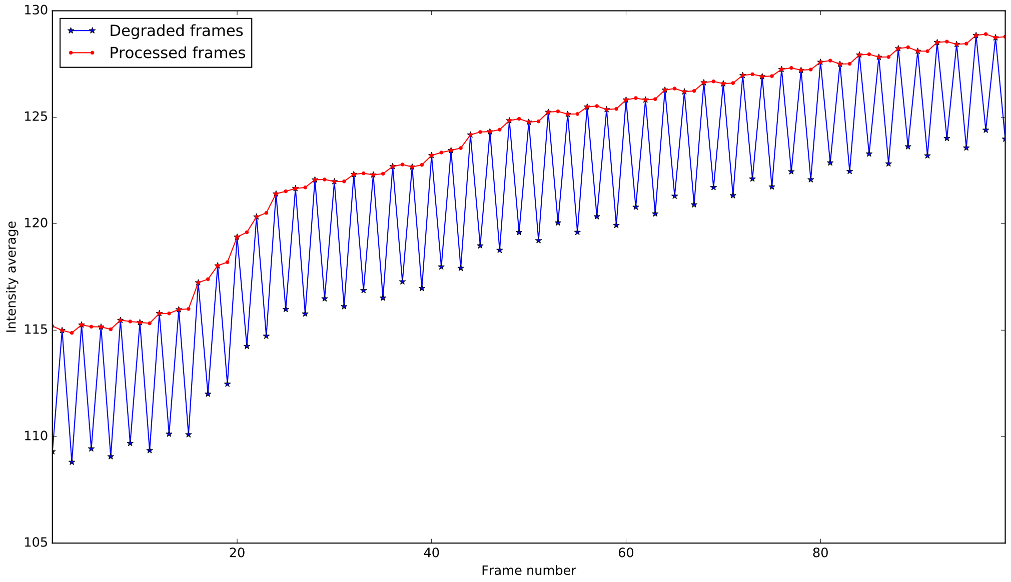
In this section, we present some experimental results to demonstrate the efficacy of the proposed approaches. First, our color correction and deblocking models were tested on a synthetic periodic/non-periodic flicker sequences produced from a flicker-free, naturally lit video and also on a real, studio-lit video affected with periodic flicker and featuring multiple light sources and complex motion. The artificial non-periodic flicker was created in such a way that large differences in brightness between the different periods were exhibited. Plotting the global luminosity vs. the image sequence index allows us to evaluate the performance of the methods. Figure 5.13 shows the result of processing the real degraded image sequences using the causal method. The luminosity variation graphs in Figures 5.14, 5.15, 5.16 and 5.17 show that brightness levels are well restored in the color-corrected videos, and even at different acquisition and lighting conditions, different geometrical motions and also at varying frame rates.



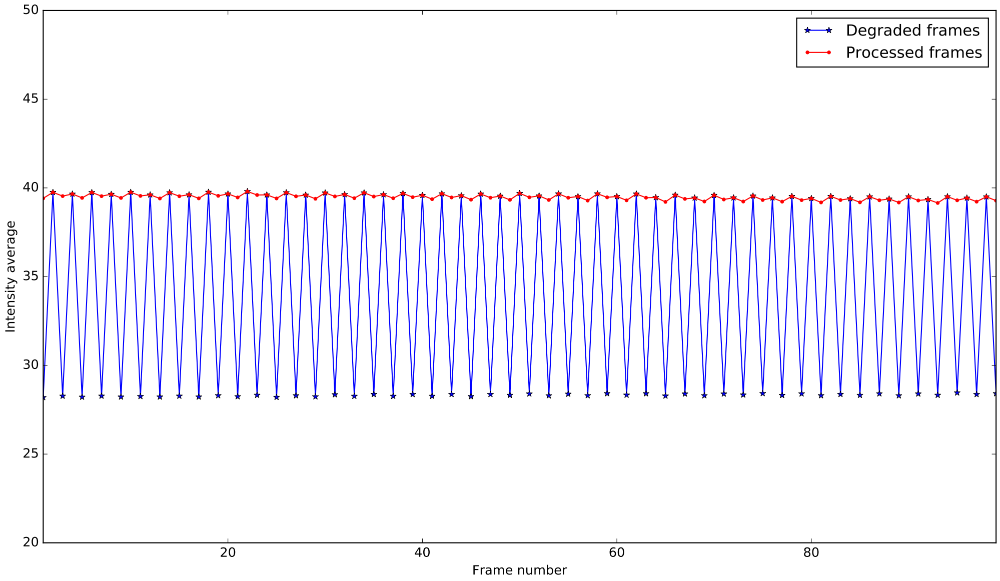
(a) Video 1: including complex motions at 1000 fps



(b) Video 2: including translations, scale changes at 240 fps

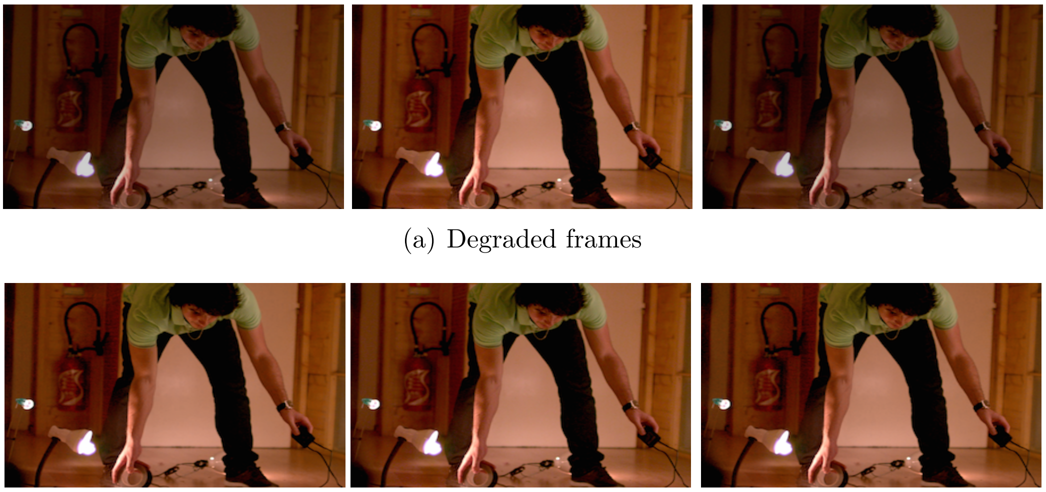


(c) Video 3: including rotation, translations, scale changes, focusing and defocusing effects



(d) Video 4: including fast rotations at 240 fps

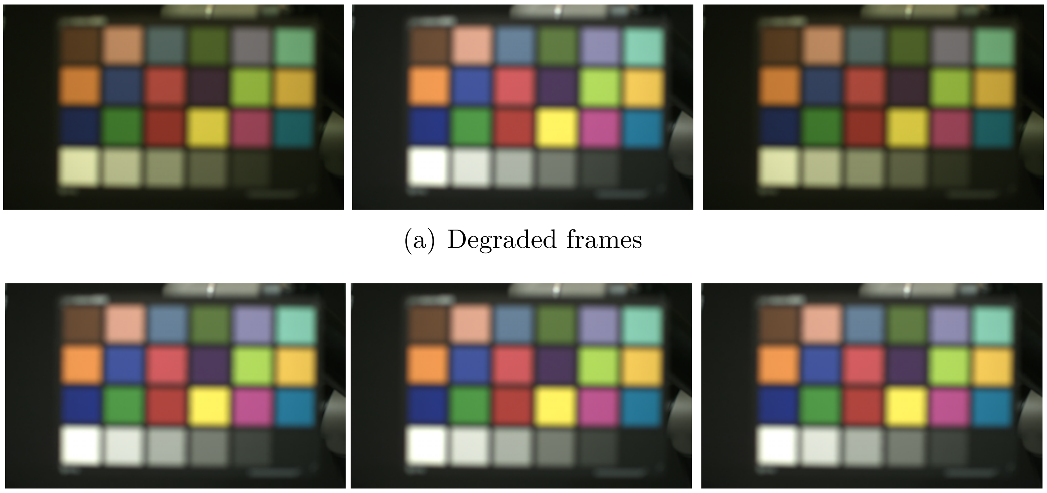
Figure 5.13. Luminosity fluctuation and restoration in some real sequences with different acquisition properties: complex motions, different lighting conditions and acquisition frame rates.



(a) Degraded frames

(b) Processed frames

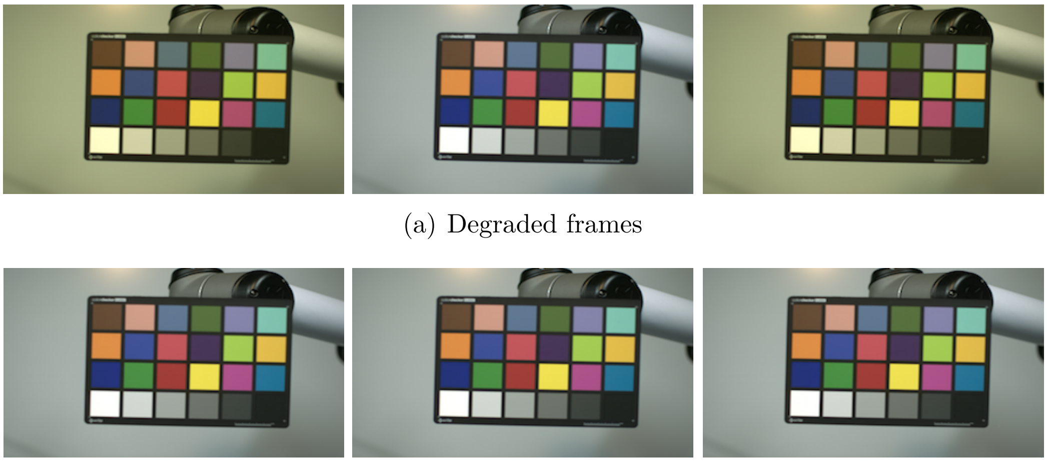
Figure 5.14. Video 1: Acquired at 1000 fps, illuminated by three light sources, it includes complex motions, saturation effects, noise at the background and some outliers.



(a) Degraded frames

(b) Processed frames

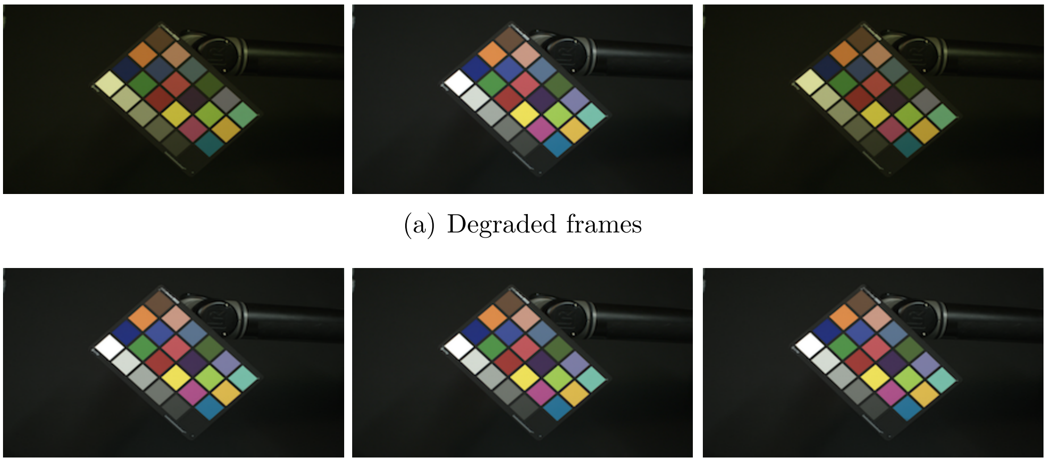
Figure 5.15. Video 2: Acquired at 240 fps, and so we have rapid illumination/ chromatic changes. It includes focusing/defocusing effects accompanied by translation motions and scale changes.



(a) Degraded frames

(b) Processed frames

Figure 5.16. Video 3: Acquired at 240 fps, it presents high contrast variations between image regions. Different light sources are illuminating the scene with different flicker properties.



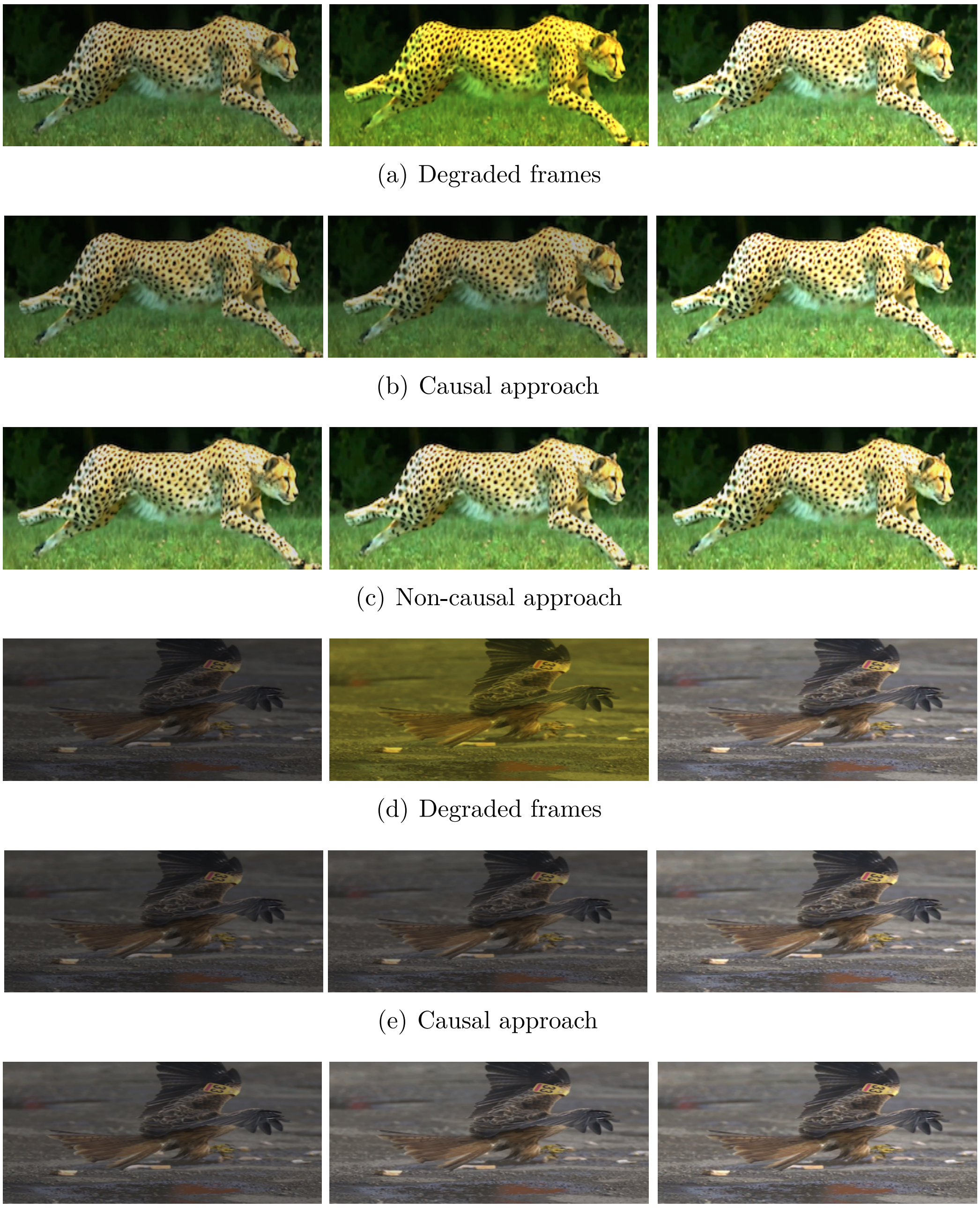
(a) Degraded frames

(b) Processed frames

Figure 5.17. Video 4: Acquired at 240 fps under multiple light sources, it includes very fast rotations.

As we do not have a flicker-free reference sequence in this case, we cannot precisely estimate similarity measures. We then utilize the structural similarity index (SSIM), and signal-to-noise ratio (SNR) estimators to measure the similarity between the original and degraded images on the one hand, and then between the original and restored sequences on the other hand. Figure 5.18 represents a restoration example on two sequences with artificial flicker. These examples feature large motion between frames, occlusions in the background and compound motion: for example, in the bird sequence, the camera is moving in the opposite of the bird's direction.

Deblocking is used after flicker removal to alleviate blocking artifacts caused by occlusions and complex displacements. The PSNR average increases from 21 to 48 dB after color correction. A further 1 dB is gained on average after the deblocking step (see Figure 5.20(a)). We compared our method to another deflickering procedure adapted from Naranjo and Albiol (2000), which was initially tailored for old grayscale movies, and that we extended to color sequences. This procedure compensates flicker globally and provides a single matrix for the whole image at each timestep. It only improves the PSNR to 32 dB on average, so our gain is significant. Similarly, our SSIM index is very close to 1 after restoration (see Figure 5.20(b)), and its improvement after removing blocking artifacts and comparing to the global deflickering method is also significant.



(a) Degraded frames

(b) Causal approach

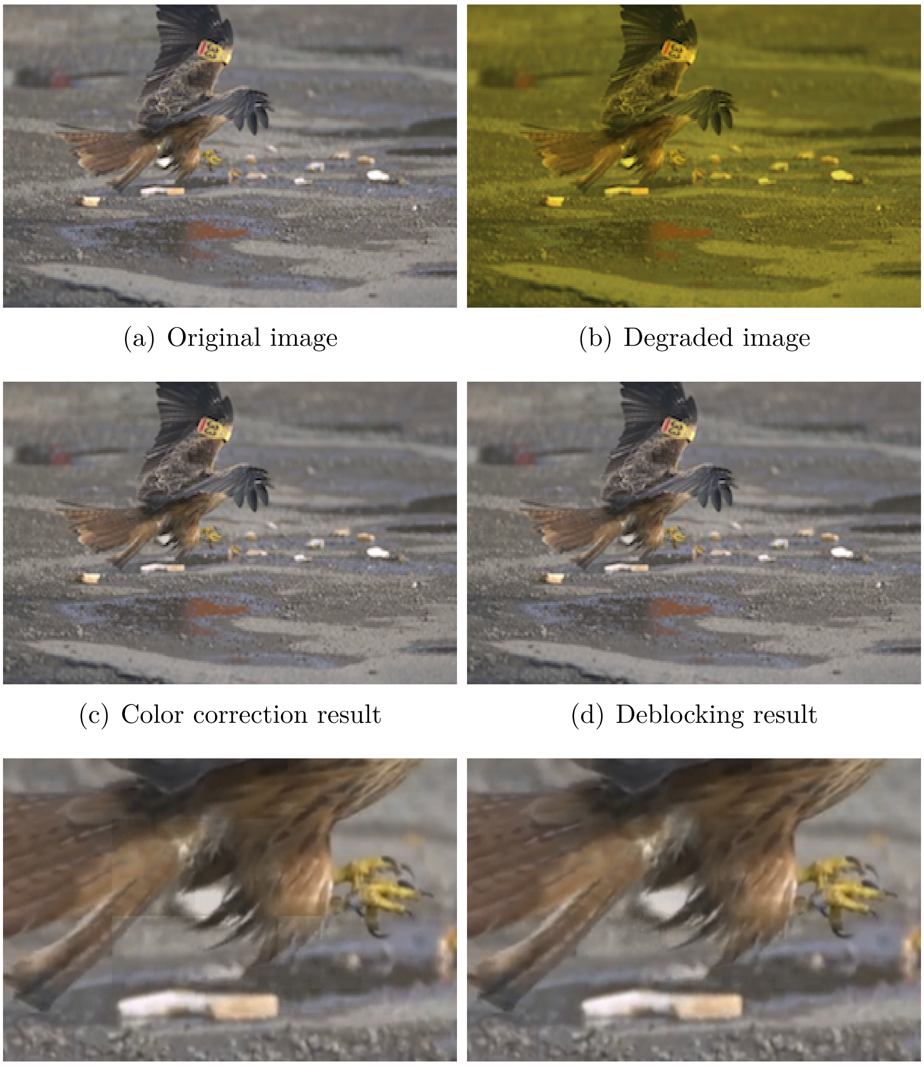
(c) Non-causal approach

(d) Degraded frames

(e) Causal approach

(f) Non-causal approach

Figure 5.18. Videos 5-6: Comparison between causal and non-causal approaches for color correction on synthetic flicker sequences produced from a flicker-free, naturally lit video.



(a) Original image

(b) Degraded image

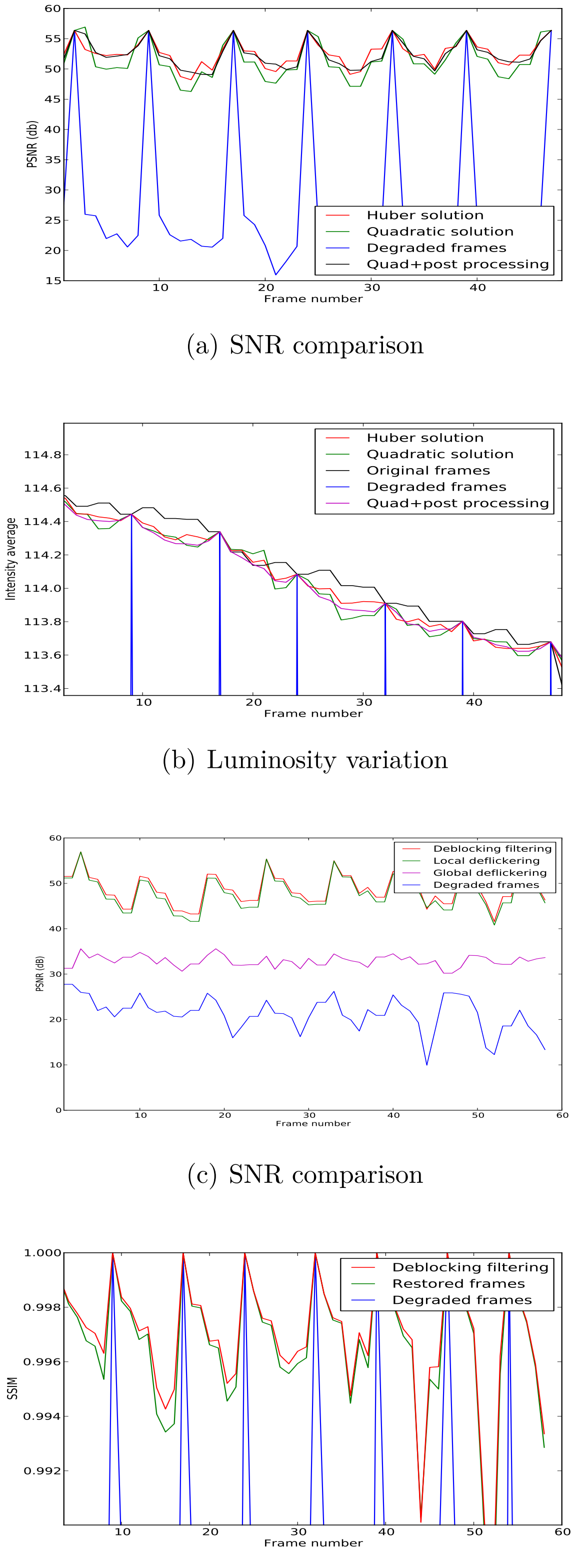
(c) Color correction result

(d) Deblocking result

(e) Zoom before deblocking

(f) Zoom after deblocking

Figure 5.19. Sequence with large local motion and periodic artificial flicker.



(a) SNR comparison

(b) Luminosity variation

(c) SNR comparison

(d) SSIM comparison

Figure 5.20. Artificial flicker: similarity measures

Finally, in order to compare between causal and non-causal approach, the similarity measures were computed on a non-periodic flicker affected sequence, see Figure 5.21.

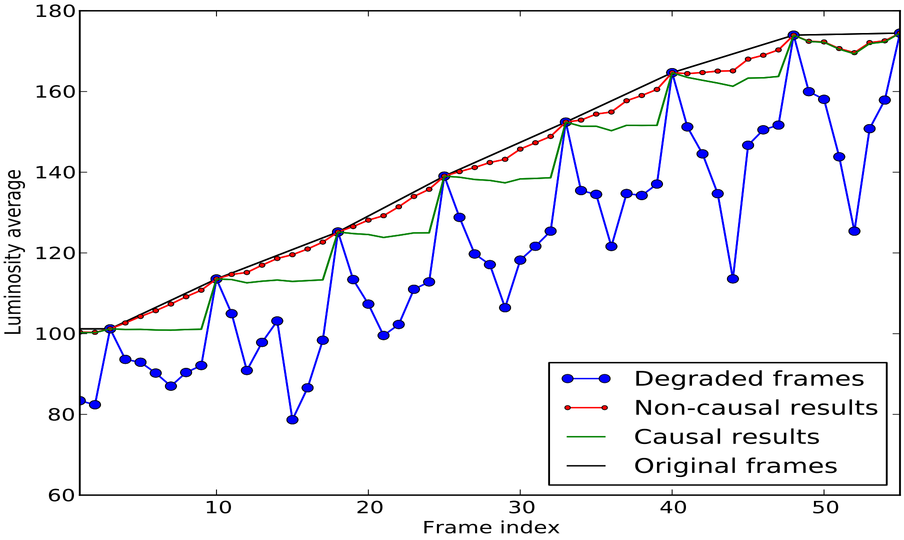


Figure 5.21. Luminosity average vs. images sequence for the causal and non-causal approaches, on a non-periodic flicker affected video.

It is clear in the figure above, that the non-causal approach provides a reasonable illumination correction result, the sudden brightness changes are removed between different flicker periods. The non-causal approach seems to be more efficient for the non-periodic variations compared to the causal approach, and can be used for several applications as well as high speed imaging, for instance, to stabilize brightness variations in time lapse videos.

Finally, the pyramidal approach was tested on the causal method using Huber solution to verify whether or not the processing results keep the same quality while reducing the resolution of images and decreasing the computation time. Figure 5.22 shows that PSNR average almost keeps the same value with and without applying the pyramidal approach on the block-based causal method. This approach has allowed us to accelerate the processing time of causal method five times.

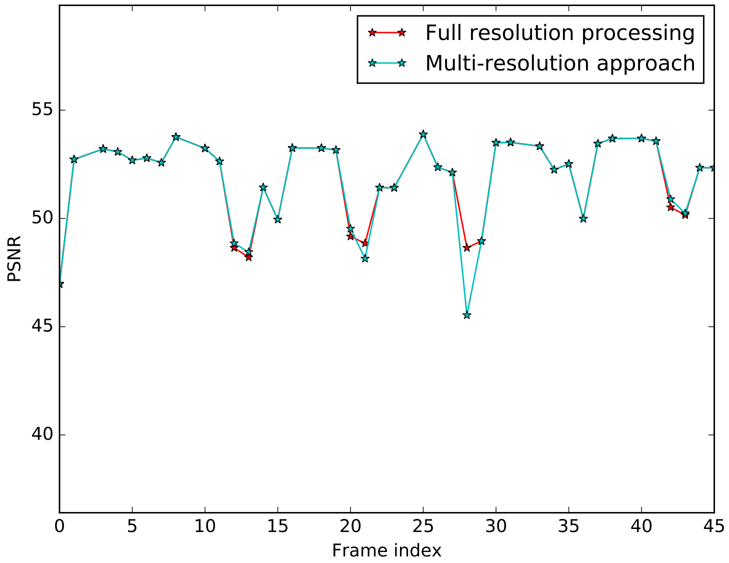


Figure 5.22. PSNR measure for the pyramidal approach and the causal approach using Huber solution on the bird sequence.

6 Conclusion

In this chapter, we have proposed a new flexible local color correction technique to remove flickering artifact, qualified as periodic/non-periodic, suitable for correcting high speed color video taken under artificial lighting and which can be useful for some other video applications.

We have described a causal and non-causal tracking methods involving per block color correction matrix estimation, followed by a per-pixel post-processing / block artifact removal step requiring a color correction matrix estimation and regularization.

In order to reduce the processing time, we introduced a pyramidal strategy based on the causal approach, to estimate tracking and correction parameters at reduced scales, while maintaining approximately the same quality of results. This principle is very useful to improve the method computational time both for real time implementations and for the offline applications.

Chapter 6

Local method based on superpixels and spatial interpolation

1 Introduction

In computer vision, the detection of features of interest in images is a fundamental task, since it enables tracking, stereovision, 3D reconstruction, and many other tasks. Features can have any shape but are most often associated with point locations in images. As described previously, some problems make the tracking of these areas or objects difficult, depending on various factors related to the acquired data or to the application, for instance the presence of noise in images, illumination and chroma variations, complex motions, partial and total occlusions, and other problems.

The representation of objects is a very important part of the tracking process. An object can be represented by a point (center of the object for example) or a set of points of interest (keypoints). An object may also be represented by a simple shape such as a rectangle (as seen above in chapter 5), ellipse, etc. This representation is robust to follow rigid objects (i.e. vehicles) but is also suitable for non-rigid objects. An object can also be represented by its edges. This representation is adapted for tracking non-rigid objects with a complex or undefined shape.

In this chapter, we adopt the viewpoint that it is not always necessary to track all image regions in a video, but that it is beneficial to ignore noisy areas, featureless regions or poorly exposed areas. We propose thus to track the regions of interest in a video to find their corresponding color correction models, i.e. to find a sufficient number of estimates on which we can rely to predict the remaining local color corrections for all pixels of the entire image using spatial interpolation.

The objective of this chapter is to propose new avenues for video color correction, and typically for flicker removal applications. We suggest another local method based on superpixels segmentation, that is able to track objects with undefined and complex shapes. In other words, regions of interest will guide the object of interest over time.

2 Preliminaries

The interpolation is a classical problem of estimating a function $F(p)$, where $p = (x, y)$ is a point in the image plane, from N known values of F from the surrounding points p_i

$$F(p) = \sum_{i=1}^{N} w_i F(p_i). \tag{6.1}$$

The main issue is the determination of the weighting w_i, for each of the surrounding points. There are several ways for choosing these weights.

For example, to make a weather map, there is not necessarily a sufficient number of evenly distributed weather stations to cover the whole region. Interpolation can estimate the temperature, humidity, atmospheric pressure, etc in different places without recorded data by using the known measurements in close weather stations. This technique is also often used for other types of data, such as population density, snow accumulation, etc.

There exist many interpolation methods. These methods can be classified into two categories: one dimensional and two dimensional methods. Among the most common one dimensional methods are the quadratic interpolation, cubic interpolation and cubic B-spline Interpolation. Higher dimensional methods are often derived from the one dimensional methods. In our context, we are interested in the two dimensional methods and we will briefly explain some of the most commonly used methods in the present chapter.

2.1 Spatial interpolation

Spatial interpolation is the interpolation on functions of more than one variable. The function to be interpolated is known at given points $(x_i, y_i, z_i, \dots)$ and the interpolation problem consists of yielding values at arbitrary points $(x, y, z, \dots)$.

Multivariate interpolation is particularly important in geostatistics, meteorology and in image processing applications. We explain below some of the most commonly used methods. We mention the nearest neighbor interpolation, bilinear interpolation, bicubic interpolation, inverse distance weighting interpolation and kriging interpolation method.

2.1.1 Nearest neighbor interpolation

The nearest neighbor algorithm is a multivariate interpolation method that can be useful for interpolating irregular grids of estimated points. It selects the value of the nearest known point and does not consider the values of neighboring points at all, yielding a piecewise-constant interpolant related to the Voronoï diagram of the sampled points. The algorithm is very simple to implement and is commonly used (usually along with multi-resolution applications) in real-time 3D rendering to select color values for a textured surface, but it yields a very heteregenous result in terms of color and requires a large number of known estimates to provide a reasonable results, thus its use is very much limited in our context.

2.1.2 Bilinear interpolation

The bilinear interpolation allows to calculate the intensity value of a two dimensional function or an image f at any point, using its four nearest neighbors $C_{11}(x_1, y_1)$, $C_{12}(x_1, y_2)$, $C_{21}(x_2, y_1)$, $C_{22}(x_2, y_2)$. It performs a linear interpolation first in one direction, and then again in the other direction. It allows to obtain better results than the already mentionned interpolation method by nearest neighbor, and also remains of reasonable complexity.

Contrary to what its name suggests, the interpolation function is not linear but is of quadratic form because of the term xy:

$$f(x,y) = ax + by + cxy + d \tag{6.2}$$

where $f(x,y)$ is the interpolation value of f at the position (x,y), and a, b, c and d are four unknown variables that we can estimate by linearly interpolating in the x direction and then interpolate in the y direction.

The linear interpolation in the x direction yields:

$$f(x,y_1) = \frac{x_2 - x}{x_2 - x_1} f(C_{11}) + \frac{x - x_1}{x_2 - x_1} f(C_{21}) \tag{6.3}$$

$$f(x,y_2) = \frac{x_2 - x}{x_2 - x_1} f(C_{12}) + \frac{x - x_1}{x_2 - x_1} f(C_{22}) \tag{6.4}$$

Then, the linear interpolation in the y direction is performed as follows and presents the solution of the bilinear interpolation:

$$\begin{aligned} f(x,y) &= \frac{y_2 - y}{y_2 - y_1} f(x,y_1) + \frac{y - y_1}{y_2 - y_1} f(x,y_2) \\ &= \frac{1}{(x_2 - x_1)(y_2 - y_1)} \begin{bmatrix} x_2 - x & x - x_1 \end{bmatrix} \begin{bmatrix} f(C_{11}) & f(C_{12}) \\ f(C_{21}) & f(C_{22}) \end{bmatrix} \begin{bmatrix} y_2 - y \\ y - y_1 \end{bmatrix} \end{aligned} \tag{6.5}$$

This interpolation is continuous, but at the connection between pixels, derivatives are not continuous, which can introduce discontinuities between interpolated intensity levels. The method described here is valid for a regular grid but can be readily extended to an irregular grid.

2.1.3 Bicubic interpolation

Bicubic interpolation is an extension of the cubic interpolation to interpolate a set of distributed points on a regular two-dimensional grid.

It is often preferred to bilinear interpolation or the nearest neighbor technique for image resampling even if the processing time is not critical. The bicubic interpolation considers a neighborhood of 16 pixels (4 × 4) which ensures a smoother interpolation result with fewer discontinuity artefacts.

This two dimensional interpolation is expressed as follows:

$$f(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}x^i y^j \tag{6.6}$$

and its partial derivatives are thus expressed as

$$f_x(x,y) = \sum_{i=1}^{3}\sum_{j=0}^{3} i a_{ij}x^{i-1} y^j \tag{6.7}$$

$$f_y(x,y) = \sum_{i=0}^{3}\sum_{j=1}^{3} j a_{ij}x^{i} y^{j-1} \tag{6.8}$$

$$f_{xy}(x,y) = \sum_{i=1}^{3}\sum_{j=1}^{3} ij a_{ij}x^{i-1} y^{j-1} \tag{6.9}$$

where 16 parameters have to be estimated.

Suppose that the function values f and the derivatives f_x, f_y and f_{xy} are known at the four corners $C(0,0)$, $C(0,1)$, $C(1,0)$, $C(1,1)$ of the unit square. The solution to the unknown parameters a_{ij} is introduced by Thévenaz et al. (2000) as follows:

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) & f_y(0,0) & f_y(0,1) \\ f(1,0) & f(1,1) & f_y(1,0) & f_y(1,1) \\ f_x(0,0) & f_x(0,1) & f_{xy}(0,0) & f_{xy}(0,1) \\ f_x(1,0) & f_x(1,1) & f_{xy}(1,0) & f_{xy}(1,1) \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \tag{6.10}$$

Similarly to the bilinear interpolation, this method works on a regular grid of known points, and thus it will not be useful in the context of superpixels. These methods can be useful for the post processing step of our block-based local method.

Beside the nearest-neighbor interpolation method, there exist other state-of-the-art methods that are defined for scattered data on an irregular grid. For instance, we can mention methods based on triangulated irregular network-based natural neighbor, radial basis function, thin plate spline, natural neighbour interpolation, inverse distance weighting (IDW), kriging interpolation and others.

2.1.4 Inverse distance weighting IDW

IDW (Shepard (1968)) is a distance-based method, it considers that the weight of any known point is set inversely proportional to its distance from the estimated point.

There is a common way to find an interpolated value u from a given point x, that uses the following interpolation function:

$$u(x) = \begin{cases} \frac{\sum_{k=0}^{N} w_k(x) u_k}{\sum_{k=0}^{N} w_k(x)} & \text{if } d(x, x_k) \neq 0 \\ u_k & \text{if } d(x, x_k) = 0 \end{cases} \tag{6.11}$$

where w_k is a simple weighting function:

$$w_k(x) = \frac{1}{d(x, x_k)^p}, \tag{6.12}$$

x is an interpolated point, x_k is a known interpolated point (estimate), d is a given distance between x and x_k, N is the total number of known points used in the interpolation and p is a real positive number, the power parameter. The weight of the neighboring points decreases as the distance increases. The choice of p depends on the desired smoothing level for the interpolation. Larger values of p give more influence for the nearest values to the interpolated point.

IDW method relies on minimizing a function related to the measurement of the deviations between the tuples of interpolating points (x, u) and the k pairs of interpolated points (x_k, u_k), defined as:

$$\phi(x, u) = \left(\sum_{k=0}^{N} \frac{(u - u_k)^2}{d(x, x_k)^p} \right)^{\frac{1}{p}}. \tag{6.13}$$

The disadvantage of this method is that the quality of the interpolation result may decrease if the distribution of sampled data points is irregular. Several modified versions of this interpolation algorithm have been developed in (Liszka (1984); Łukaszyk (2004)).

2.1.5 Kriging Interpolation

Kriging, also named Gaussian process regression, was proposed in the master's thesis of Krige (1951), for mining applications. The theoretical basis for kriging was developed by Matheron (1971).

In the previously detailed weighting methods. Only the distances between the samples and the point to be estimated are taken into account. Instead, kriging is an interpolation method that estimates the weights from the degree of similarity between the values of F, e.g. from the covariance between the points with respect to the distance separing them. There are three main types of kriging: Simple kriging, Universal kriging and Ordinary kriging. Simple kriging is equivalent to a linear interpolation by minimizing the mean square error of a stationary field (Lapierre and Fortet (1953)). Ordinary kriging is the most frequently used and will be detailed and used below.

Ordinary kriging assumes second-order stationarity, which means that the mean and the variance of the function F are stationary, e.g. they do not depend on the position of the points, and the correlation between two points depends only on the distance between them. Kriging is mainly based on an estimator of the covariance called the variogram.

Definition 4. *Let Z be a random variable of a space variable x, which is assumed to be stationary, in particular the mean and the variance of $Z(x)$ are independent of x. The variogram between points x and y is given by*

$$\gamma(x,y) = \mathrm{Var}[Z(x) - Z(y)] = \mathrm{E}[\mid Z(x) - Z(y) \mid^2].$$

In the following, we assume that the variogram can be modelled by an isotropic function, meaning that it only depends on the distance $h_{x,y}$ between x and y. Since it is isotropic, only the positive half of the variogram (called the semi-variogram) is necessary to determine the weights in (6.1).

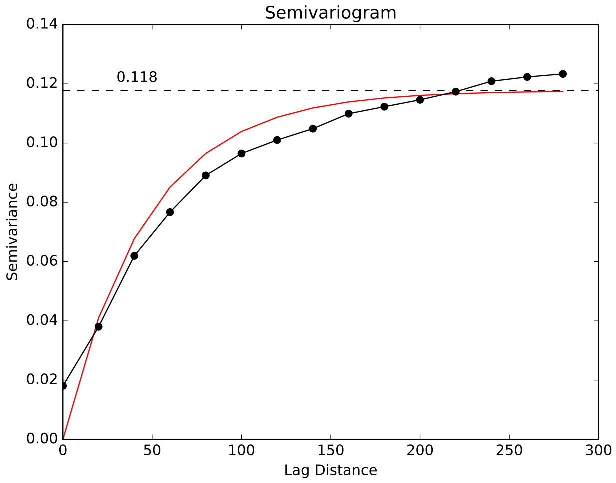


Figure 6.1. The computation of the semi-variogram and its best fit exponential model

Kriging then consists of calculating the weights w_i using the γ values corresponding to N selected points. The sum of weights must to be one to ensure the unbiasedness of ordinary kriging system.

Definition 5. *The estimation error of the random variable Z in x_p is expressed by:*

$$\epsilon(x_p) = \hat{Z}(x_p) - Z(x_p) = \sum_i^N w_i(x_p) Z(x_i) - Z(x_p),$$

and since Z is stationary in ordinary kriging, $E(Z(x_i)) = E(Z(x_p)) = m$, with m the mean of Z. The estimation is unbiased when $E(\epsilon(x_p)) = 0$, we can deduce:

$$\sum_i^N w_i(x_p) E(Z(x_i)) - E(Z(x_p)) = 0 \Leftrightarrow m \sum_i^N w_i(x_p) - m = 0 \Leftrightarrow \sum_i^N w_i(x_p) = 1.$$

For a point x_p, ordinary kriging consists of finding the weights w_i which ensures that corresponding variances fit with the estimated semi-variogram.

Ordinary kriging is expressed as follows:

$$\begin{cases} \Gamma \cdot W & = \Gamma_0 \\ \sum_i^N w_i & = 1 \end{cases} \tag{6.14}$$

where $\Gamma = \begin{pmatrix} \gamma(h_{1,1}) & \cdots & \gamma(h_{1,n}) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \gamma(h_{n,1}) & \cdots & \gamma(h_{n,n}) & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}$, $\Gamma_0 = \begin{pmatrix} \gamma(h_{1,p}) \\ \vdots \\ \gamma(h_{N,p}) \\ 1 \end{pmatrix}$ and $W = \begin{pmatrix} w_0 \\ \vdots \\ w_N \\ \varrho \end{pmatrix}$,

where ϱ is a Lagrange factor used in the minimization of the kriging error σ_p^2 to honor the unbiasedness condition and h_{x_i,y_j} represents the distance between two points x_i and y_j. This variation error is computed as follows:

$$\sigma_p^2 = W^\top \cdot \Gamma_0. \tag{6.15}$$

The minimal variance and the unbiased estimation condition shows the advantage of using ordinary kriging for spatial interpolation instead of other methods.

3 Superpixel based method

3.1 Superpixel definition

A superpixel is a grouping of pixels sharing similar features like color or texture in an atomic way, which can replace the rigid structure of the classic regions of interest. Superpixel algorithms capture redundancy in an image, provide a practical primitive for further analysis such as detection, segmentation, classification and tracking of objects. The associated algorithms also significantly reduce the complexity of image processing tasks, and simplify the memory usage.

Various algorithms have been used to generate superpixels, that are based on graphs (Felzenszwalb and Huttenlocher (2004); Shi and Malik (2000); Veksler et al. (2010)), k-means (Achanta et al. (2012)) and geometrical flows (Machairas et al. (2014); Levinshtein et al. (2009)).

In our work, we employ the Simple Linear Iterative Clustering (SLIC) algorithm which is easy to use and offers a flexibility in terms of compactness and number of generated superpixels. This algorithm will be briefly detailed later in the next section.

3.2 General strategy

This section describes a local approach based on superpixels tracking for motion estimation and color correction. The suggested method is implemented as a two main step procedure. At the first stage, regions of interest are tracked from the reference image to the target images in order to estimate a color correction matrix. Secondly, an interpolation between the maintained color estimates is performed to process the unknown estimated points.

3.3 Motion tracking

Construction of superpixels in images allows us to work locally on the extracted information. Our method begins by calculating a superpixels segmentation of the reference image $f(\cdot, t_{\text{ref}})$ and the target images $f(\cdot, t)$. For this, we initialize our algorithm with the Simple Linear Iterative Clustering algorithm (Achanta et al. (2012)), which is briefly described below.

3.3.1 SLIC segmentation

The Simple Linear Iterative Clustering algorithm clusters pixels in the combined five-dimensional color and image plane space to efficiently generate compact, nearly uniform superpixels. SLIC algorithm uses color images in the $CIELAB$ color space.

It allows to generate superpixels regularly on the image surface. First, the centers of superpixels $C_k = [l_k \quad a_k \quad b_k \quad x_k \quad y_k]^\top$ are initialized on a regular grid and spaced by S pixels, with $S = \sqrt{\frac{K}{N}}$ where N is the total number of desired superpixels and K is the approximative number of superpixels. l, a and b designate the color components, and x and y correspond to the geometrical position. These centers can be optionally moved to avoid being on image boundaries.

SLIC is an iterative method consisting of two steps:

- assigning pixels to a center C_k basing on a similarity criteria,
- updating the centers.

This approach seeks to minimize, in the first step, the similarity criteria corresponding to a distance between C_k and the current pixel $p = [l \quad a \quad b \quad x \quad y]^\intercal$, defined by:

$$D^2_{SLIC}(C_k, p) = d^2_{lab}(C_k, p) + \frac{d^2_{xy}(C_k, p)}{S^2} m^2 \tag{6.16}$$

where m is the compactness, d_{lab} is a colorimetric distance and d^2_{xy} is the geometric distance between C_k and p in the current image,

$$\begin{aligned} d^2_{lab}(C_k, p) &= (l_k - l)^2 + (a_k - a)^2 + (b_k - b)^2 \qquad (6.17) \\ d^2_{xy}(C_k, p) &= (x_k - x)^2 + (y_k - y)^2 \qquad (6.18) \end{aligned}$$

When m is large, spatial proximity is more important and the resulting superpixels are more compact (i.e. they have a lower area to perimeter ratio). When m is small, the resulting superpixels adhere more tightly to image boundaries, but have less regular size and shape. When using the $CIELAB$ color space, m can be in the range $[1, 40]$.

For each center, a similarity search area is defined, of $2S \times 2S$ size and centered on C_k. Only pixels of this zone are examined.

Thereafter, we show how we can track the motion using this segmentation process.

3.3.2 Tracking of labels

For a specific flicker period, the reference frame is segmented using the SLIC algorithm using a regular initialization of superpixels centers (See Figure 6.2). Updated centers are associated to all superpixels at the end of the segmentation process. Each superpixel is identified by its own label number and its center coordinates. SLIC parameters for the segmentation of next subsequent frames are initialized with the last centers grid obtained from the segmentation of the previous frame. This improves the likelihood that a tracked superpixel maintains the same label number during the whole flicker period. A validation test is performed in order to verify that the same label number in two consecutive

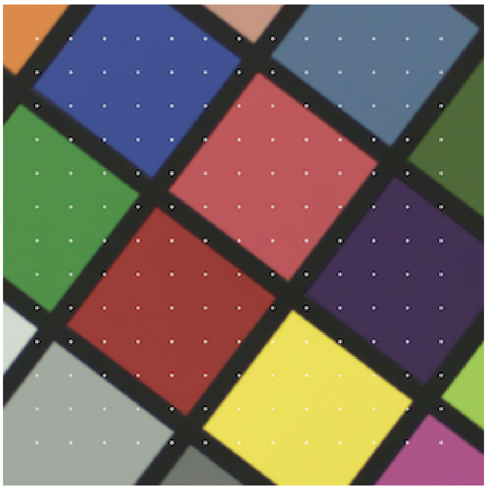


Figure 6.2. Centers initialization for the reference frame segmentation.

images does indeed correspond to the same region of interest.

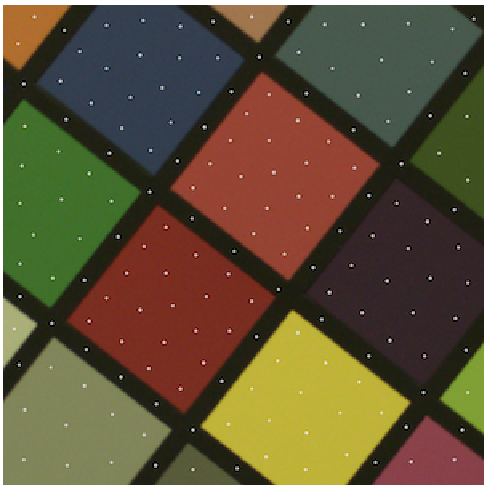


Figure 6.3. Centers initialization for a subsequent frame segmentation: It depends on the centers grid already estimated for the previous segmented frame.

In our context, displacements are considered over a small range. As a first check, a validation test depends on the displacement value between the corresponding matched superpixels centers, i.e. if this value is higher than a given threshold, tracking of the corresponding superpixel is considered to have failed. It will therefore be ignored for the color correction estimation. We also assume that a given superpixel retains the same neighbors between two successive frames to validate its matching procedure.

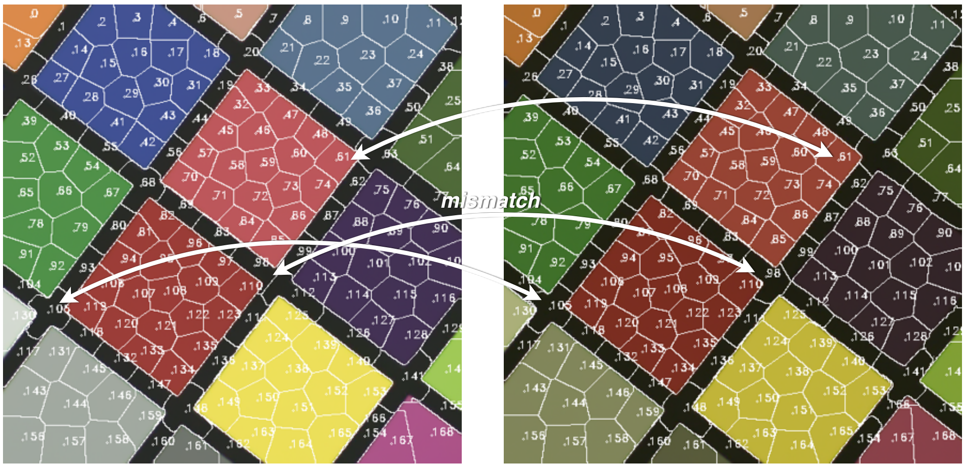


Figure 6.4. Example of some good and bad matches between segmented reference and target frames.

Once a superpixel is successfully tracked and maintains its label number between the reference frame and the current frame, a color correction estimation is subsequently performed.

3.4 Color correction step

Due to the presence of motion and illumination variations between the segmented frames, the shape and area of a superpixel S_k at label k can vary significantly. Thus, a color distribution comparison between superpixels with the same label would not yield good results.

In addition, the SLIC algorithm classifies the regions of interest in images based on a color similarity criteria. As a result, superpixels tend to be homogeneous in color. This implies that attempting to correlate the color content of superpixels between frames would be a poorly conditioned match. We observed that classical histogram-based approaches fail in the context of superpixel matching.
We propose a ROI based color comparison in order to estimate the colorimetric transformation for each valid tracked superpixel.

3.5 ROI based color matching

We consider a superpixel $S_k(t)$ in a current frame $f(\cdot,t)$ and centered by $\mathbf{C}_k(t)$, that is successfully tracked from its initial position centered on $\mathbf{C}_k(t_{ref})$ in the reference frame $f(\cdot,t_{ref})$:

$$\mathbf{C}_k(t_{ref}) = \mathbf{C}_k(t) + \widehat{\mathbf{d}}_S \tag{6.19}$$

where $\widehat{\mathbf{d}}_S$ is the displacement vector between the initial k label position in the reference frame and its position in the current fame. We then compare two ROI regions $B_k(t)$ and $B_k(t_{ref})$ surrounding $\mathbf{C}_k(t)$ and $\mathbf{C}_k(t_{ref})$ respectively. The flicker matrix is estimated by minimizing the following energy function

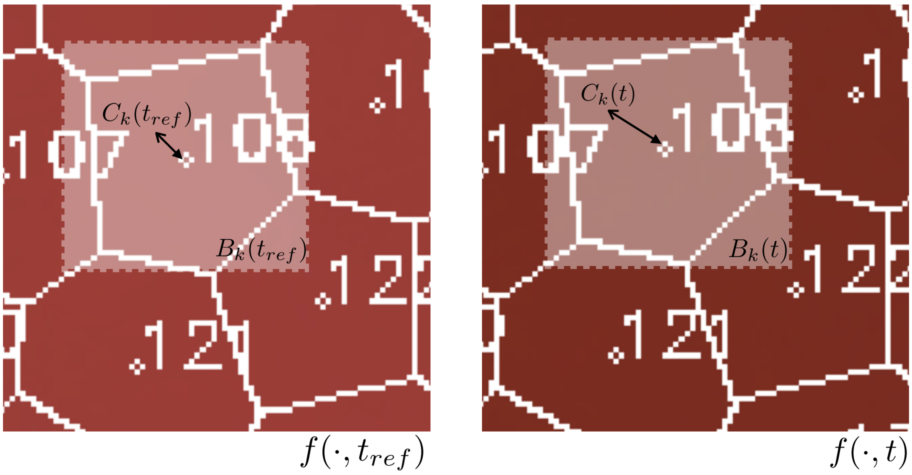


Figure 6.5. Pixel wise matching for similar superpixels.

$$J(\widetilde{\boldsymbol{M}}_{k,t},\mathbf{d}_S) = \sum_{\mathbf{s}\in\mathcal{B}_k} \Phi\big(\widetilde{\boldsymbol{M}}_{k,t}\,\mathbf{f}(\mathbf{s},t) - \mathbf{f}(\mathbf{s}-\mathbf{d}_S,t_{\text{ref}})\big), \tag{6.20}$$

where $\Phi:\mathbb{R}^3 \to [0,+\infty[$ is a cost function and $\mathcal{B}_k$ is the pair of the blocks $B_k(t)$ and $B_k(t_{\text{ref}})$ that are matched by the labelling of superpixels. The simplest and fastest choice for Φ is the quadratic norm where $\Phi = \|\cdot\|^2/2$. The quadratic solution is given by

$$\widehat{\boldsymbol{M}}_{k,t} = \Big(\sum_{\mathbf{s} \in \mathcal{B}_k} \mathbf{f}(\mathbf{s} - \mathbf{d}_S, t_{\text{ref}}) \mathbf{f}(\mathbf{s}, t)^\top \Big) \cdot \Big(\sum_{\mathbf{s} \in \mathcal{B}_k} \mathbf{f}(\mathbf{s}, t) \, \mathbf{f}(\mathbf{s}, t)^\top \Big)^\dagger . \tag{6.21}$$

Once a color correction matrix $\widehat{\boldsymbol{M}}_{k,t}$ is computed for a successfully tracked superpixel, we apply it uniformly on all superpixel pixels:

$$\forall \mathbf{x} \in S_k(t) \, , \; \mathbf{f}_p(\mathbf{x}, t) = \widehat{\boldsymbol{M}}_{k,t} \cdot \mathbf{f}(\mathbf{x}, t), \tag{6.22}$$

where $\mathbf{f}_p(\cdot, t)$ is the processed frame at time t.

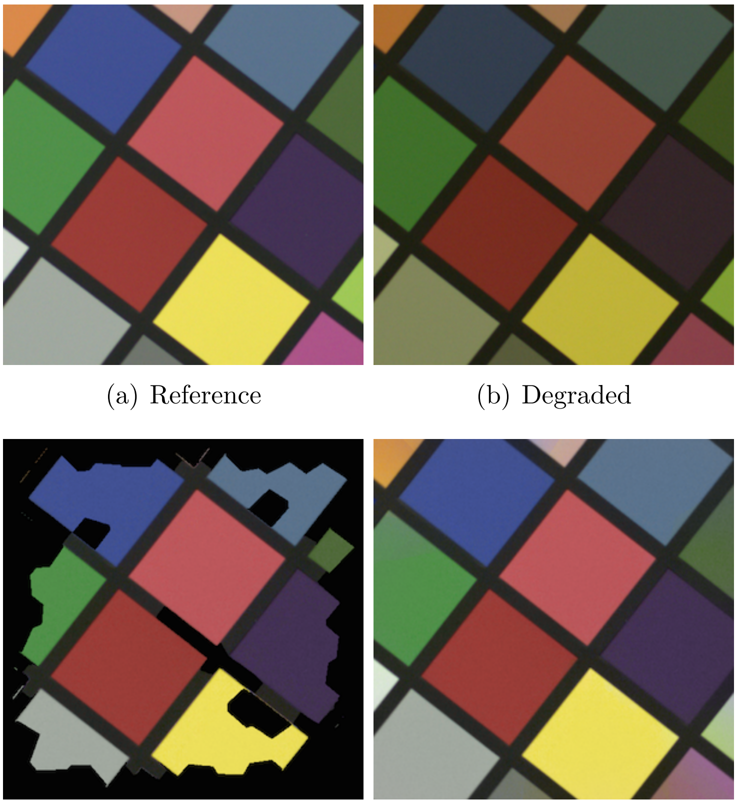
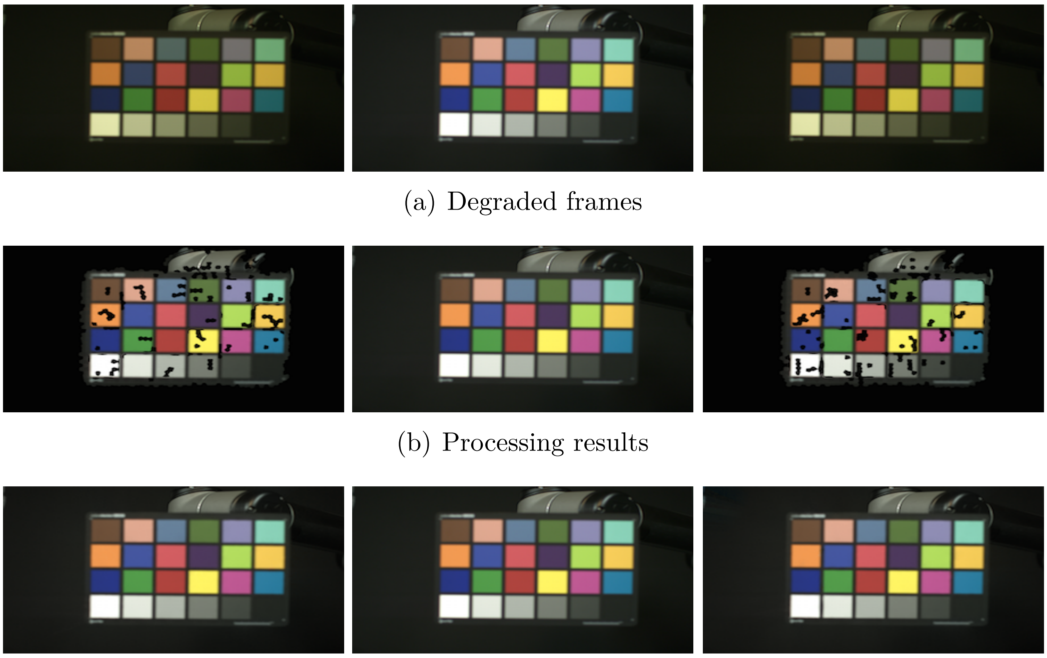

(a) Reference
(b) Degraded
(c) Correction
(d) Interpolation

Figure 6.6. Superpixel based method for color correction: In (c) only superpixels with succesfull tracking are processed, other superpixels are ignored. In (d) interpolation of the correction matrix provides a solution for the whole image.

3.6 Post processing step

Post-processing consists of finding the unknown transformation estimates for the remaining superpixels on the one hand, and to remove borders artifact on the other. For this we interpolate the transformation matrix over the whole image using a kriging model already explained in Section 2.1.5.

Figure 6.6 shows that post processing is an essential step in order to estimate the unknown color correction parameters for the ignored superpixels at the first stage, and thus to perform the color correction on the whole image, and also to eliminate some superpixel borders artifacts if they exist.



(a) Degraded frames

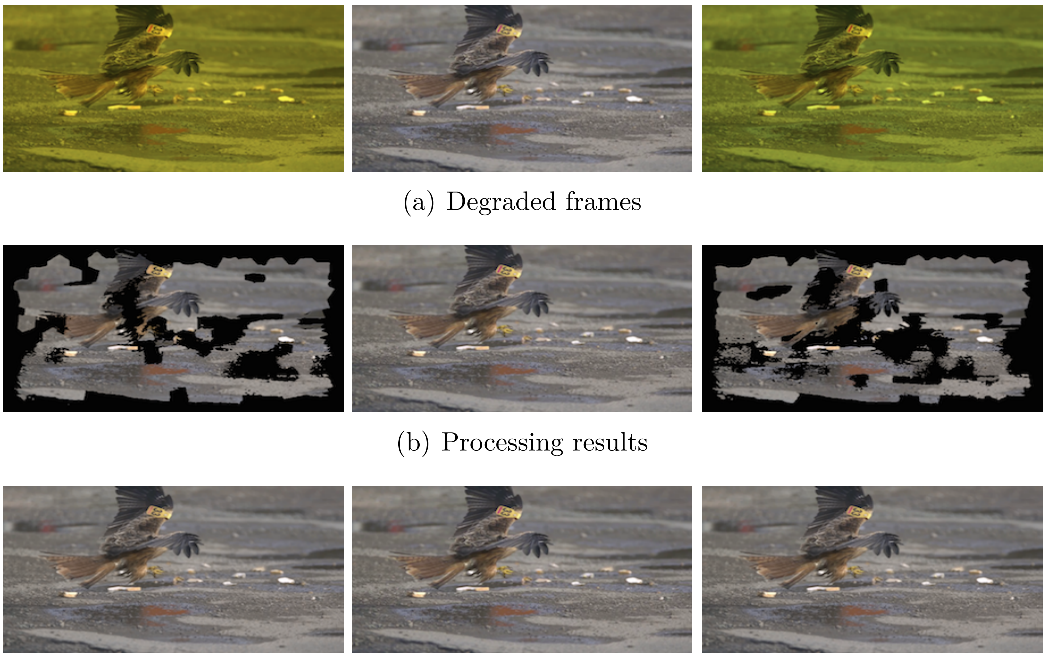
(b) Processing results

(c) Interpolation results

Figure 6.7. Video 2: Superpixel-based method for flicker removal.

3.7 Results and discussion

The proposed approach was tested on real studio-lit videos affected with local flicker and various motion including rotations (see Figure 6.7). We also validate our method on a synthetic flicker sequence in order to be able to compute similarity measures between original and processed sequences (see Figure 6.8). In Figures 6.7 and 6.8, (b) presents the first processing step on the successfully tracked superpixels, the black regions are the ignored superpixels. (c) shows the post-processing step based on the spatial interpolation of color correction parameters, it shows that the color correction of all image regions are succesfully predicted.



(a) Degraded frames

(b) Processing results

(c) Interpolation results

Figure 6.8. Video 6: Superpixel-based method for flicker removal on synthetic flicker sequence.

Figure 6.9 plots the luminosity variation of processed images of Video 1 vs. frame index, it shows that flickering is much reduced along the sequence, even if some very small, imperceptible fluctuations remain. Figure 6.10 indicates a significant image quality gain after our processing. The PSNR average increases from 20 dB to around 40 dB. These show that our approach is promising for simultaneous motion tracking and color correction for high speed imaging.

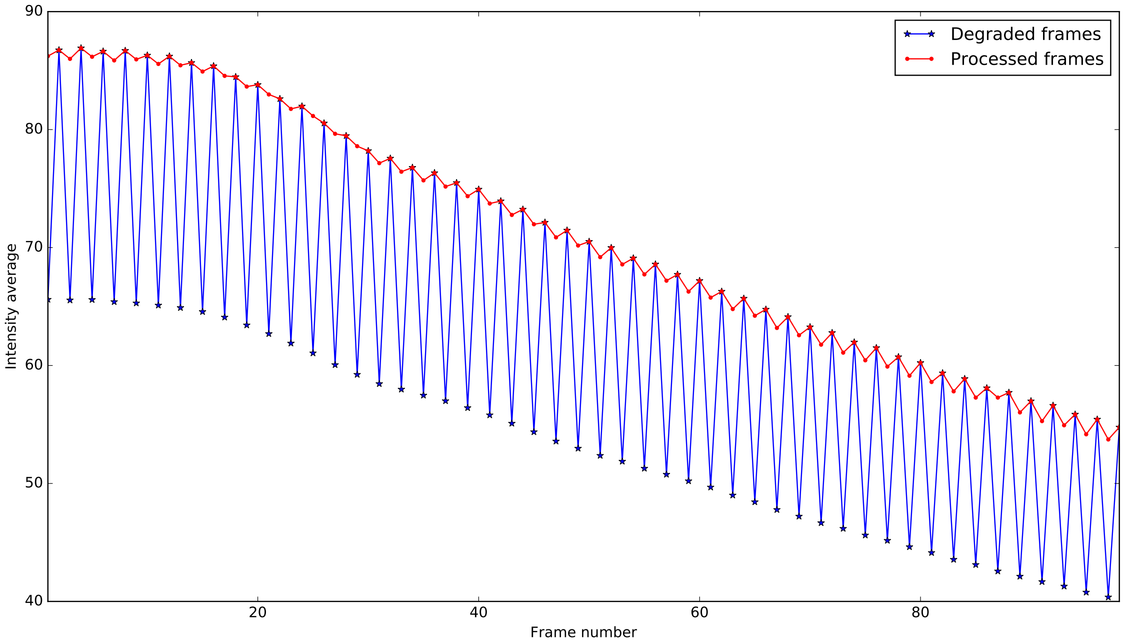


Figure 6.9. Luminosity variation of Video 2: Test of performance of superpixel based method

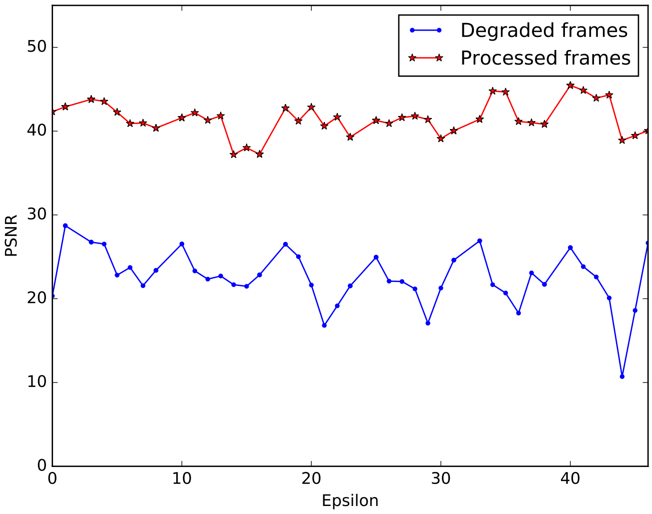


Figure 6.10. PSNR similarity measure on Video 6: Test of performance of superpixel based method on synthetic flicker sequence.

3.8 Conclusion

In this chapter, we were inspired by the superpixel-based segmentation algorithms for motion tracking in image sequences in the presence of brightness and chromatic changes. Based on this tracking, a pixel-wise matching approach was established to compare the color distributions between tracked regions of interest from a reference image to a target one in order to locally estimate flicker parameters in an image sequence.

The results show that our superpixel-based method yields good results, especially in the color distribution matching step. Using multiple constraints based on the neighboring superpixels allows for a reliable tracking method. However, this tracking method only yields a motion vector per pair of matched superpixels. For some applications, this may be too inaccurate. Augmenting the density of superpixels may be an appropriate solution, however this is more demanding in terms of computation time in very high definition videos (8k or more).

Chapter 7

A comparative chapter

In this chapter we compare our three proposed approaches. The global image registration based method (in chapter 4), the local block-based colorimetric correction method (in chapter 5) and the local superpixel tracking based flicker removal approach (in chapter 6).

1 Simulation scenarios

We tested these methods on four real, studio-lit videos affected with periodic flicker and featuring multiple light sources and complex motion. Video 1 is acquired at 1000 fps using multiple light sources, it includes complex large motions and high noise level. The videos 2, 3 and 4 are acquired at 240 fps. Video 2 includes focusing/defocusing effects accompanied by translation motions and scale changes. Video 3 presents high contrast variations and different light sources are illuminating the scene with different flicker properties. Video 4 includes fast rotations, which helps to test the robustness of our approaches to different motion caracteristics.
These methods are also tested on three synthetic periodic flicker sequences (the videos 5, 6 and 7) produced from a flicker-free, naturally lit videos, in order to quantitatively compare the processing results in terms of signal to noise ratio, and to prove the efficiency of each method against different colorimetric, illumination, motion and acquisition conditions. Video 5 is acquired at 300 fps and includes global motion. The videos 6 and 7 are also acquired at 300 fps and feature large local motion between frames.

This chapter allows us to further analyse each method result and thus to improve the quality of tracking and flicker removal results in the future works.

2 Quantitative results

We plot the global luminosity vs. the image sequence index in order to qualitatively evaluate the performance of the proposed methods.

Figure 7.1 shows that illumination levels in Video 1 are better aligned with local approaches than global approach basing on features tracking. There is a slight advantage for block based method compared to the superpixel-based one.

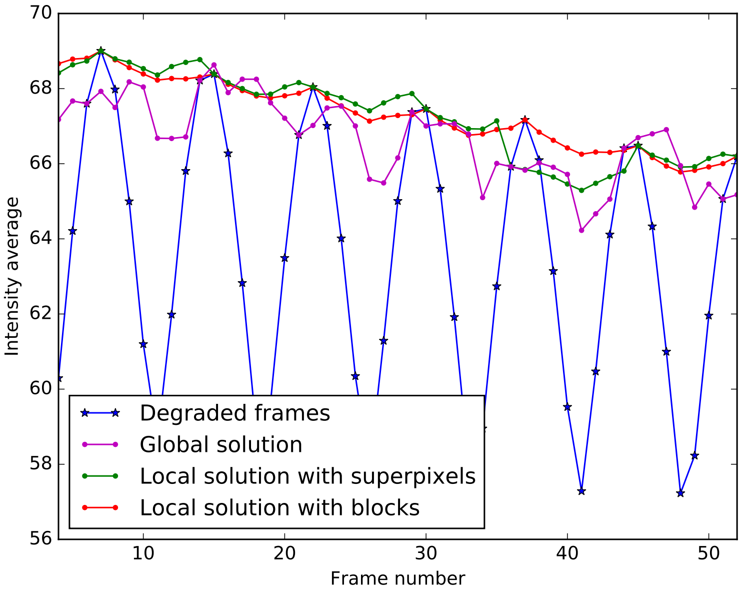


Figure 7.1. Video 1: including complex motions at 1000 fps

Figures 7.2, 7.3, 7.4 show that the illumination degradation are well adjusted with block-based method (red curve) in the videos 2, 3 and 4 respectively. The superpixel method (green curve) is also suitable for the present acquisition conditions. These two local approaches provide a continuous illumination level after processing, but the global method cannot deal with local colorimetric variations and only reduces the flicker artifacts (magenta curve).

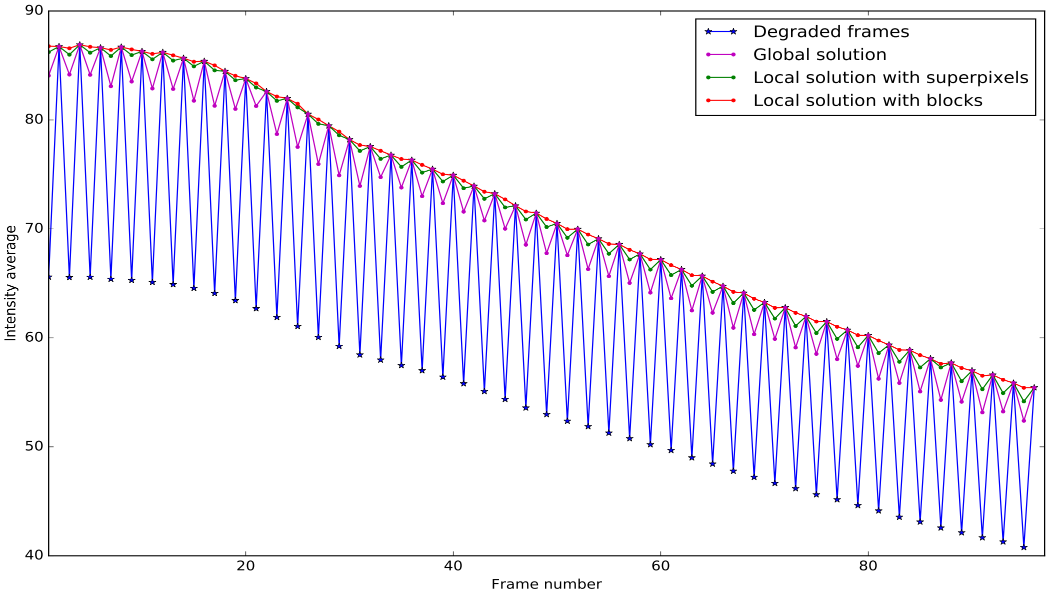


Figure 7.2. Video 2: Acquired at 240 fps, and so we have rapid illumination/ chromatic changes.

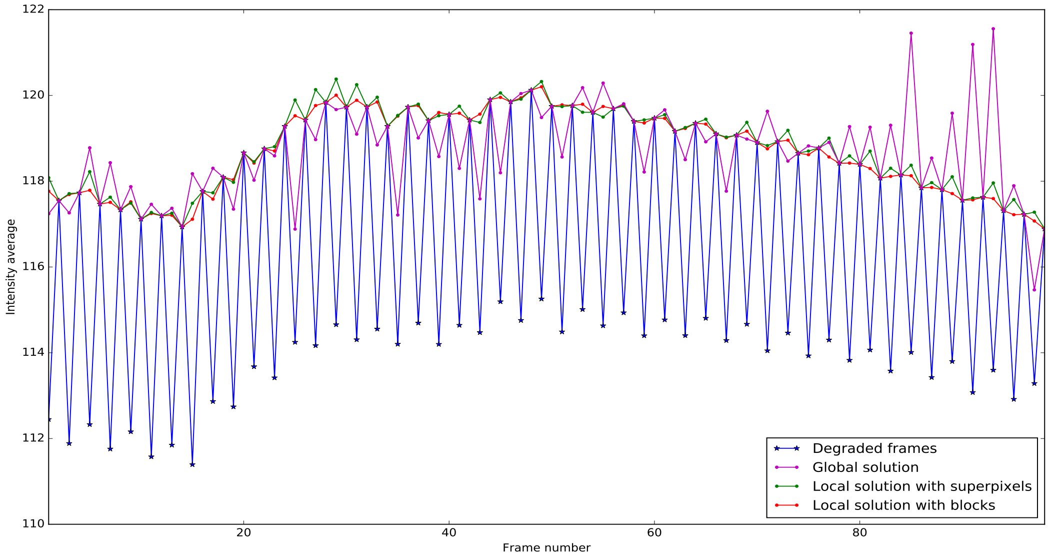


Figure 7.3. Video 3: Acquired at 240 fps, it presents high contrast variations.

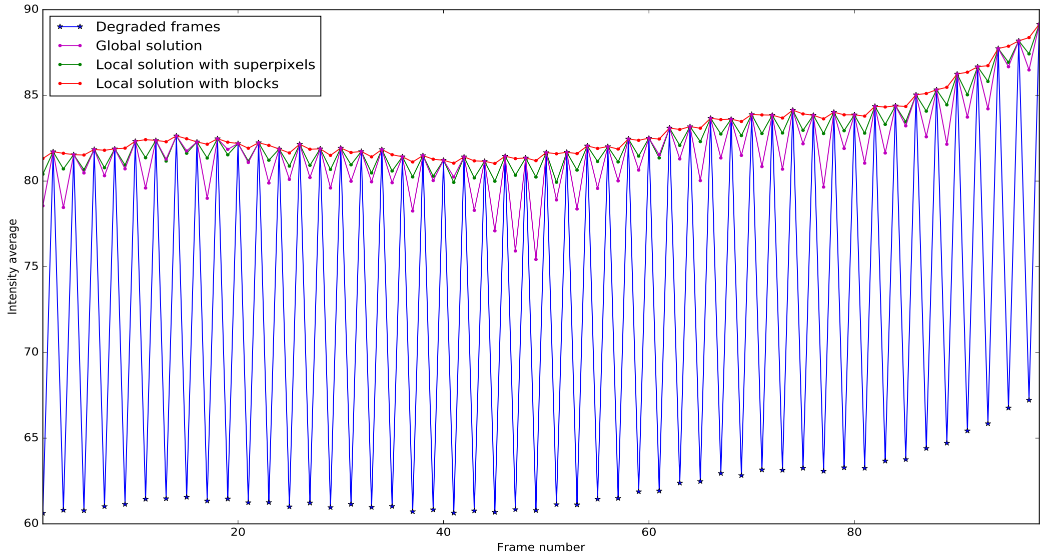
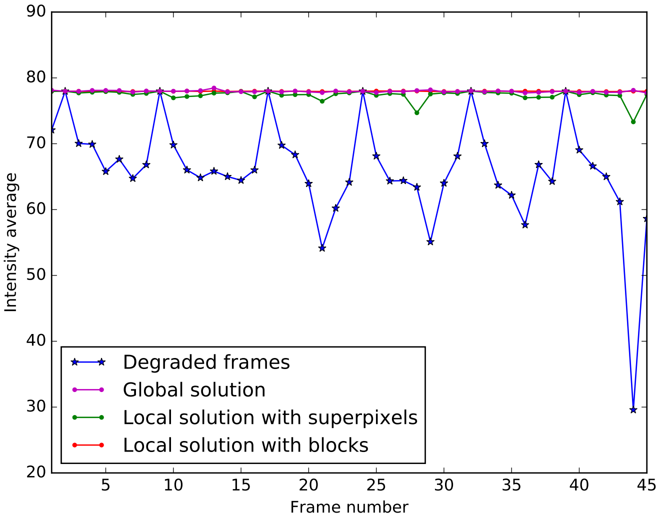


Figure 7.4. Video 4: Acquired at 240 fps with multiple sources, including fast rotations.

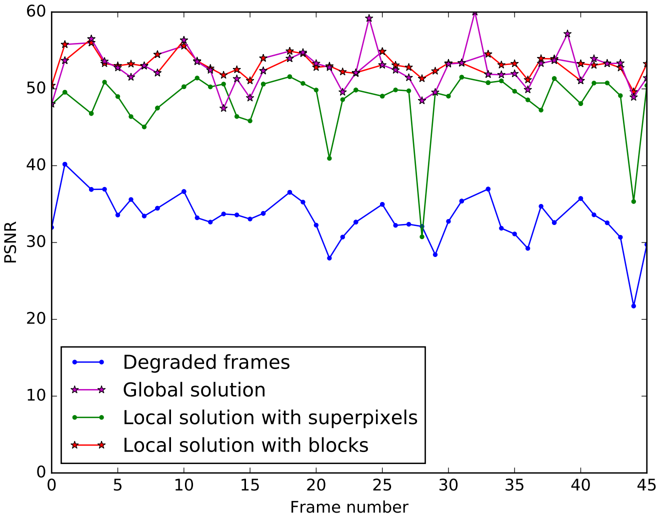
Plotting the global luminosity vs. the image sequence index allows us to qualitatively evaluate the performance of the methods. As we do not have a flicker-free reference sequence in the previous videos, we cannot quantitatively estimate similarity measures. In the next section, we use the signal-to-noise ratio (SNR) estimators to measure the similarity between the original and artificially degraded images on the one hand, and then between the original and restored sequences on the other hand.

We test our methods on three synthetic flicker sequences (Videos 5, 6 and 7) produced from a flicker-free, naturally lit video.

Figure 7.5 compares our proposed approaches on the building video, which features global motion between frames, and with global artificial flicker. We can see that the block-matching based method yields the best similarity ratio between processed images and original images, with a small advantage with respect to the global solution, due to the presence of outliers and saturation regions. Superpixel tracking is less accurate in this context.



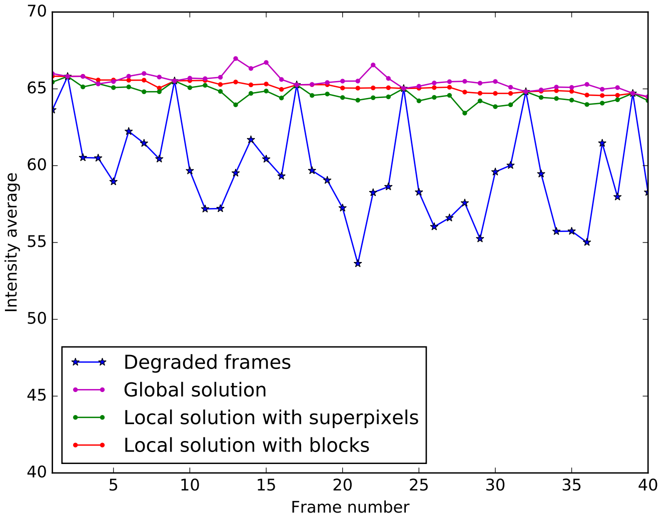
(a) Global luminosity vs. image index



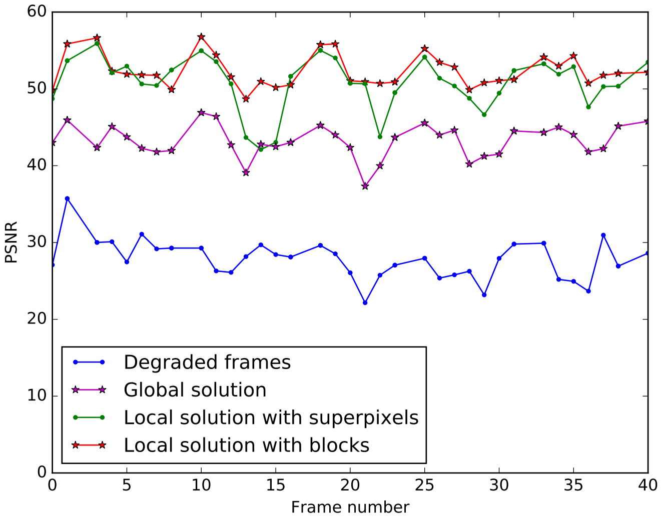
(b) PSNR

Figure 7.5. Video 5: The building sequence, acquired at 300 fps, including global motion

Figure 7.6 measures the similarity on the Car sequence, which represents a large local motion, the camera is fixed, and only the car is moving in the video.



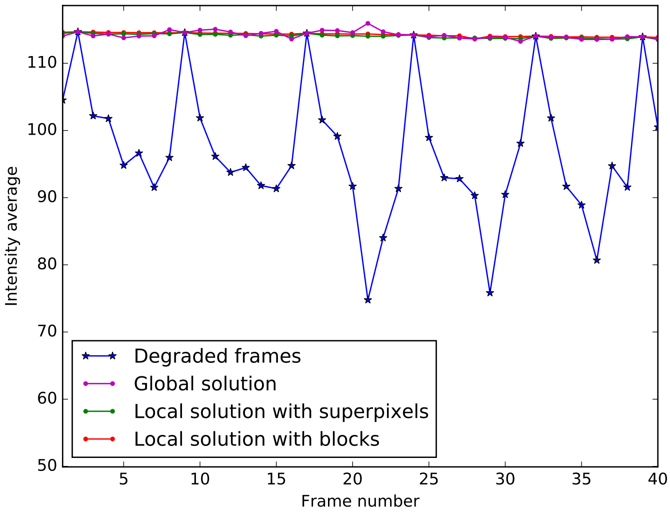
(a) Global luminosity Vs. image index



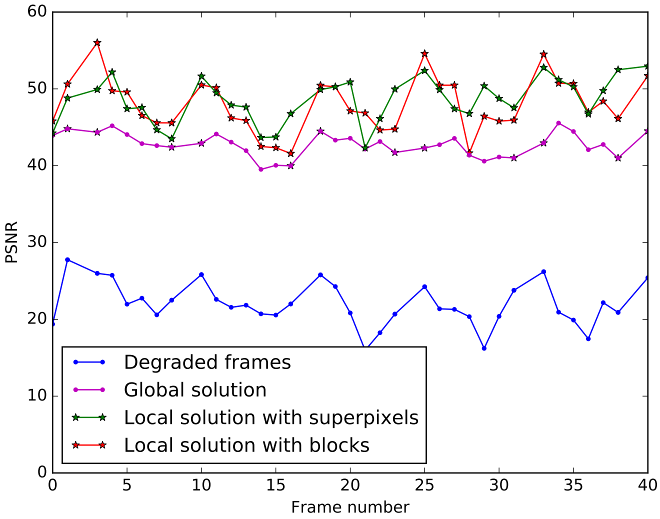
(b) PSNR

Figure 7.6. Video 6: The Car sequence, acquired at 300 fps

Figure 7.7 example (Bird sequence) also features large local motion between frames, occlusions in the background but with compound motion: the camera is moving in the opposite of the bird's direction.



(a) Global luminosity vs. image index



(b) PSNR

Figure 7.7. Video 7: Bird video

In the Car and Bird sequences, we can see that the local approaches are more suitable, and allows to gain in terms of signal to noise ratio comparing to the global approach. In Table 7.1, we show the PSNR average values corresponding to the processing solutions of our suggested methods. These are compared to the

PSNR average between degraded and original frames.

	Global solution	Block-based solution	Superpixel-based solution	Degraded frames
Building seq.	52.04	53.07	48.22	33.04
Car seq.	43.08	52.05	50.32	27.51
Bird seq.	42.58	47.61	47.46	21.69

Table 7.1. PSNR comparison table

We notice that the block-based approach yields the best results in the sequence with global motion as well as in the sequences with local motion and flickering. This method results in a gain of around 23 dB. The superpixel-based method is also promising for local color correction estimation, but it does not yield a highly accurate motion estimation. This is the reason why its corresponding PSNR is lower than the block-based approach. The global solution can be used only with some specific conditions, it requires a global motion with single flicker frequency on the images. It yields an ideal PSNR ratio for the building sequence (52.04 dB), but it fails when motion and flicker artifacts become more complex (in the car and bird sequences).

Chapter 8

Conclusion and openings

1 Summary of our contributions

In this manuscript, we have studied the problem of luminosity and chroma variations in videos in general, and more particularly in the context of high speed video acquisition under artificial lighting. These variations are termed "flicker". This problem is likely to become more prevalent with the recent advent at consumer level of high speed video acquisition devices, for instance newer smartphone generations or sports/action cameras. The state of the art has shown a lack of similar works in this context, but has presented some approaches that were useful, for example the works on flicker removal for archived videos, the color transfer methods as well as a variety of motion tracking methods.

This manuscript recalled some fundamental and essential basics on color perception, imaging sensors and video acquisition techniques, motion tracking methods as well as color correction techniques in different contexts. Then, a global strategy was introduced for the reduction and removal of periodic and aperiodic flicker in videos.

Several approaches were proposed to stabilize chroma and luminosity variations in videos. First, a global approach was introduced consisting of uniformly processing a subsequent frame in a video with respect to a reference image. The first proposed method is global, it estimates a unique color correction matrix for all image pixels using the cumulated histogram matching. Another suggested global method computes the desired parameters based on a rigid registration between the target image and the reference image in order to match similar areas. These two methods are interesting in terms of computation time because only one correction process is performed on the whole image. Flicker is well reduced on videos in the presence of simplified lighting and motion conditions. On the other hand, these methods are too limited in the case of complex sequences, for instance in the presence of large motions, or in the presence of multiple light sources that illuminate the scene. Multiple illumination typically result in complex flicker patterns in different regions of the image, and must be separately processed in order to better stabilize the brightness and color levels. For this, a local approach is essential to improve the quality of the obtained results.

For a local color correction strategy, motion tracking is an essential step in order to properly match the regions of a target image to a reference image. Optical flow methods that assume a luminosity consistency were not adequate in the studied context, neither are methods that are based on the extraction of regions of interest because they cannot always cover all regions of the image. Our first proposed local method uses the block matching technique taking into account the color change factor to match similar regions, in order to simultaneously estimate a better color correction and displacement parameters. This used estimation process is very promising because it simultaneously optimizes a continuous variable (color correction matrix) and a discrete variable (displacement vector). A post-processing step is performed to remove edge effects around the processed blocks. A second local method was proposed, based on a superpixel segmentation of images, in order to take into account irregular shapes and complex motions such as rotations. This method uses the SLIC segmentation algorithm whose initialization was modified to track the generated superpixels. The correction matrix is computed by pixe-wisel matching of neighborhood areas around the centers of similar superpixels between the target image and the reference image. A kriging interpolation technique was applied in order to predict the color correction parameters on some ignored areas and to eliminate the edge effects that appear at the first processing level. Despite the fact that this method has advantages in terms of detecting rotations and irregular shapes compared to the first proposed local method, it has a lower signal-to-noise ratio due to inaccurate motion tracking. This influences the pixel-wise matching for the computation of the color correction parameters.

The two global and local approaches were tested on video sequences with an artificial and real flicker, and all proposed methods show remarkable restoration performance compared to other color transfer and some global correction methods that, although designed in different contexts, were adapted to the problem of interest. The block-matching based method presents the best results overall, because motion tracking and brightness/chroma variations are simultaneously studied. This guarantees more accurate match between images.

2 Some perspectives

Some future work, in the continuation of what has been presented in this manuscript, can be considered.

The use of 3D histogram and optimal transport methods

Several improvements can be useful in the future, especially in the histogram matching based method in Chapter 4, particularly if we no longer require pixel-wise matching but if the color transfer function can be deduced from the color histogram. The use of 3D histograms can add precision to the estimation of colorimetric variations. These 3D histograms can also be useful for matching superpixels in the local processing approach already explained in Chapter 6, using optimal transport methods between similar superpixels for predicting flicker parameters (Ferradans et al. (2014)), which could accelerate the procedure and improve the accuracy of flicker parameter estimation.

The advantage of these 3D histograms is that inter-channel correlations can better be taken into account, unlike methods considering separate 1D histograms for representing color channels.

Improvements in the block matching based method

The block-matching based method, described in Chapter 5, can also take into account the rotations that were ignored in our high-speed video context. In addition, a new criterion can be defined in order to estimate the color correction parameters while dealing with the block artifacts simultaneously. These improvements will pose significant challenges in meeting computational time constraints.

Openings for multiple applications

If translations and rotations can be estimated with the block-matching based method, this can open a new avenue for other applications. For instance, the image registration problem.

In addition, since flicker problem exists also in video compression, which uses block matching algorithm to estimate motion between video frames. This can be used to discover temporal redundancy in the video sequence, increasing the effectiveness of inter-frame video compression by defining the contents of a macroblock by reference to the contents of a known macroblock which is minimally different. Classical video compression techniques do not take into account luminosity variations between frames, which causes block artifacts in the compressed videos. Our block-matching based flicker removal method may be useful in this context, especially the block artifact removal algorithm which ensures the per-block motion estimation and a per-pixel processing step.

Our global approaches that have the advantage in terms of computation time could be an interesting solution for real time application, for example in newer smartphone generations or sports/action cameras providing high speed acquisition options. These devices require high performance in order to deal with real-time acquisitions. Another possibility would be to develop an offline application or plugin for video flicker removal.

Implementation

In order to accelerate the computation time, it will be interesting to implement our methods on many-core processor cards, for instance on GPGPU, FPGA or XEON PHI. These offer a massively parallel architecture. Thus, we could significantly accelerate a wide range of the computation steps required by our methods.

Bibliography

R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

M. Asano and K. Matsusaka. Flicker correction for moving picture, May 15 2007. US Patent 7,218,777.

S. Baudry, B. Chupeau, M. De Vito, and G. Doerr. Modeling the flicker effect in camcorded videos to improve watermark robustness. In *IEEE National Conference on Parallel Computing Technologies*, pages 42–47, Bengaluru, India, 2015.

H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, Graz, Austria, 2006. Springer.

S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 232–237, Santa Barbara, CA, USA, 1998.

L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys (CSUR)*, 24(4):325–376, 1992.

Q. Chen, M. Defrise, and F. Deconinck. Symmetric phase-only matched filtering of fourier-mellin transforms for image registration and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1156–1168, 1994.

E. De Castro and C. Morandi. Registration of translated and rotated images using finite fourier transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):700, 1987.

É. Decencière. *Restauration automatique de films anciens*. PhD thesis, École Nationale Supérieure des Mines de Paris, France, 1997.

J. Delon. Movie and video scale-time equalization application to flicker reduction. *IEEE Transactions on Image Processing*, 15(1):241–248, 2006.

G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian Conference on Image Analysis*, pages 363–370, Halmstad, Sweden, 2003. Springer.

U. Fecker, M. Barkowsky, and A. Kaup. Histogram-based prefiltering for luminance and chrominance compensation of multiview video. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(9):1258–1267, 2008.

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

S. Ferradans, N. Papadakis, G. Peyré, and J-F. Aujol. Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, 7(3):1853–1882, 2014.

D. Fleet and Y. Weiss. Optical flow estimation. *Handbook of Mathematical Models in Computer Vision*, pages 237–257, 2006.

G. Forbin and T. Vlachos. Nonlinear flicker compensation for archived film sequences using motion-compensated graylevel tracing. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(6):803–816, 2008.

G. Forbin, T. Vlachos, and S. Tredwell. Flicker compensation for archived film using a spatially-adaptive nonlinear model. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages II–II, Toulouse, France, 2006.

H. Foroosh, J. Zerubia, and M. Berthod. Extension of phase correlation to subpixel registration. *IEEE Transactions on Image Processing*, 11(3):188–200, 2002.

C-S. Fuh and P. Maragos. Motion displacement estimation using an affine model for image matching. *Optical Engineering*, 30(7):881–887, 1991.

B. V. Funt and B. C. Lewis. Diagonal versus affine transformations for color correction. *JOSA A*, 17(11):2108–2112, 2000.

S. Geman and D. E. McClure. Statistical methods for tomographic image reconstruction. Technical report, Division of Applied Mathematics, Center for Intelligent Control Systems, 1987.

F. Ghorbel. A complete invariant description for gray-level images by the harmonic analysis approach. *Pattern Recognition Letters*, 15(10):1043–1051, 1994.

A. Giachetti. Matching techniques to compute image motion. *Image and Vision Computing*, 18(3):247–260, 2000.

F-J. Hampson and J-C. Pesquet. Motion estimation in the presence of illumination variations. *Signal Processing: Image Communication*, 16(4):373–381, 2000.

C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50. Citeseer, 1988.

P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977.

B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185–203, 1981.

P. J. Huber et al. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

J. Jia and C-K. Tang. Image registration with global and local luminance alignment. In *Proceedings. Ninth IEEE International Conference on Computer Vision*, pages 156–163, Nice, France, 2003.

Y-B. Jia. Lecture note: Transformations in homogeneous coordinates. http://web.cs.iastate.edu/~cs577/handouts/homogeneous-transform.pdf, Iowa State University, 2013.

A. Jimenez Moreno, E. Martínez-Enríquez, V. Kumar, and F. Diaz-de Maria. Standard-compliant low-pass temporal filter to reduce the perceived flicker artifact. *IEEE Transactions on Multimedia*, 16(7):1863–1873, 2014.

A. C. Kokaram, R. Dahyot, F. Pitié, and H. Denman. Simultaneous luminance and position stabilization for film and video. In *Electronic Imaging*, pages 688–699, San Jose, California, USA, 2003. International Society for Optics and Photonics.

D. G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139, 1951.

A.B. Lapierre and R. Fortet. *Théorie des fonctions aléatoires: applications à divers phénomènes de fluctuation*. Collection d'Ouvrages de Mathématiques à l'Usage de Physiciens. Masson et Cie., 1953.

A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2290–2297, 2009.

T. Liszka. An interpolation method for an irregular net of nodes. *International Journal for Numerical Methods in Engineering*, 20(9):1599–1612, 1984.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

D.G. Lowe. Object recognition from local scale-invariant features. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, volume 2, pages 1150–1157, Kerkyra, Greece, 1999.

B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, volume 81, pages 674–679, Vancouver, BC, Canada, 1981.

S. Łukaszyk. A new concept of probability metric and its applications in approximation of scattered data sets. *Computational Mechanics*, 33(4):299–304, 2004.

V. Machairas, E. Decencière, and T. Walter. Waterpixels: Superpixels based on the watershed transformation. In *IEEE International Conference on Image Processing*, pages 4343–4347, Paris, France, 2014.

J-B. A. Maintz and M. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

G. Matheron. *The theory of regionalized variables and its applications*, volume 5. École Nationale Supérieure des Mines, 1971.

H. P. Moravec. Towards automatic visual bbstacle avoidance. In *International Conference on Artificial Intelligence (Massachusetts Institute of Technology)*, page 584, Cambridge, USA, 1977.

V. Naranjo and A. Albiol. Flicker reduction in old films. In *IEEE International Conference on Image Processing*, volume 2, pages 657–659, Vancouver, Canada, 2000.

Y. Noguchi, J. Furukawa, and H. Kiya. A fast full search block matching algorithm for mpeg-4 video. In *IEEE International Conference on Image Processing*, volume 1, pages 61–65, Kobe, Japan, 1999.

T. Ohuchi, T. Seto, T. Komatsu, and T. Saito. A robust method of image flicker correction for heavily-corrupted old film sequences. In *IEEE International Conference on Image Processing*, volume 2, pages 672–675, Vancouver, Canada, 2000.

J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30. Society for Industrial and Applied Mathematics, Classics in Applied Mathematics, 1970.

B. Ozer and M. Wolf. A train station surveillance system: Challenges and solutions. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 652–657, Columbus, Ohio, USA, 2014.

F. Pitié, R. Dahyot, F. Kelly, and A. C. Kokaram. A new robust technique for stabilizing brightness fluctuations in image sequences. In *Statistical Methods in Video Processing*, pages 153–164. Springer Berlin Heidelberg, 2004.

F. Pitié, A. C. Kokaram, and R. Dahyot. N-dimensional probability density function transfer and its application to color transfer. In *Tenth IEEE International Conference on Computer Vision*, volume 2, pages 1434–1439, Beijing, China, 2005.

F. Pitié, B. Kent, B. Collis, and A. C. Kokaram. Localised deflicker of moving images. In *European Conference on Visual Media Production*, pages 134–143, London, United Kingdom, 2006.

F. Pitié, AC. Kokaram, and R. Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107(1): 123–137, 2007.

B. S. Reddy and B. N. Chatterji. An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271, 1996.

E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.

J. Ren, J. Liu, M. Li, and Z. Guo. Postprocessing of block-coded videos for deflicker and deblocking. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1631–1635, Vancouver, BC, Canada, 2013.

D. Ruderman, T. Cronin, and C. Chiao. Statistics of cone responses to natural images: Implications for visual coding. *JOSA A*, 15(8):2036–2045, 1998.

C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997.

C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

V. Seferidis and M. Ghanbari. Generalised block-matching motion estimation using quad-tree structured spatial decomposition. *IET Proceedings-Vision, Image and Signal Processing*, 141(6):446–452, 1994.

D. Serby, E. Meier, and L. Van Gool. Probabilistic object tracking using multiple features. In *IEEE Proceedings of the 17th International Conference on Pattern Recognition*, volume 2, pages 184–187, Cambridge, UK, 2004.

D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, pages 517–524, New York, NY, USA, 1968.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

A. S. M. Shihavuddin, Nuno Gracias, and R. García. Online sunflicker removal using dynamic texture prediction. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 161–167, Rome, Italy, 2012.

A. Sorkine Hornung and M. Schelker. Flicker removal for high speed video, May 21 2015. US Patent 20,150,138,223.

Y-W. Tai, J. Jia, and C-K. Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 747–754, San Diego, USA, 2005.

P. Thévenaz, T. Blu, and M. Unser. Interpolation revisited [medical images application]. *IEEE Transactions on Medical Imaging*, 19(7):739–758, 2000.

G. Tian, D. Gledhill, D. Taylor, and D. Clarke. Colour correction for panoramic imaging. In *IEEE Sixth International Conference on Information Visualisation.*, pages 483–488, London, England, 2002.

P. M. B. Van Roosmalen, R. L. Lagendijk, and J. Biemond. Correction of intensity flicker in old film sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):1013–1019, 1999.

C. J. Veenman, M. J. T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):54–72, 2001.

O. Veksler, Y. Boykov, and P. Mehrani. Superpixels and supervoxels in an energy optimization framework. In *European Conference on Computer Vision*, pages 211–224, Hersonissos, Heraklion, Crete, Greece, 2010. Springer.

T. Vlachos. Flicker correction for archived film sequences using a nonlinear model. *IEEE Transactions on Circuits and Systems for Video Technology*, 14 (4):508–516, 2004.

T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 277–280, New York, USA, 2002.

G. Wolberg and S. Zokai. Robust image registration using log-polar transform. In *IEEE Proceedings. International Conference on Image Processing*, volume 1, pages 493–496, Vancouver, Canada, 2000.

Y. Wu. Optical flow and motion analysis. *ECE510-Computer Vision Course Notes, Department of Electrical and Computer Engineering, Northwestern University, Winter*, 2001.

Y. Xiang, B. Zou, and H. Li. Selective color transfer with multi-source images. *Pattern Recognition Letters*, 30(7):682–689, 2009.

W. Xu and J. Mulligan. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 263–270, San Francisco, USA.

K. Yamamoto and R. Oi. Color correction for multi-view video using energy minimization of view networks. *International Journal of Automation and Computing*, 5(3):234–245, 2008.

L. Yang, B. Zhang, and X. Ye. Fast Fourier transform and its applications. *Opto-Electronic Engineering*, 31:1–7, 2004.

X. Yang and M. N. Chong. Enhanced approach to film flicker removal. In *International Symposium on Optical Science and Technology*, pages 39–47, San Diego, USA, 2000. International Society for Optics and Photonics.

A. Yilmaz, X. Li, and M. Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1531–1536, 2004.

A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys (CSUR)*, 38(4):13, 2006.

X. Zhang, A. B. E. Masahide, and M. Kawamata. Flicker parameters estimation in old film sequences containing moving objects. *Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 94(12):2836–2844, 2011.

B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.