Spatio-Temporal Colour Correction of Strongly Degraded Films

A.B.M Tariqul Islam

Masteroppgave
Master i Teknologi - Medieteknikk
30 ECTS
Avdeling for informatikk og medieteknikk
Høgskolen i Gjøvik, 2010

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Spatio-Temporal Colour Correction of Strongly Degraded Films

A.B.M Tariqul Islam

2010/07/01

Master Erasmus Mundus in
Color in Informatics and Media Technology (CIMET)

[SPATIO-TEMPORAL COLOUR CORRECTION OF STRONGLY DEGRADED FILMS]

Master Thesis Report

Presented by

[A.B.M Tariqul Islam]

and defended at the

Gjøvik University College, Norway

[1st July] 2010

Jury Committee:

[Professor, Alain Trémeau]
[Professor, Jon Yngve Hardeberg]
[Professor, Jussi Parkkinen]
[Associate Professor, Javier Hernández Andrés]

Supervisor(s):

[Associate Professor, Ivar Farup]

Abstract

The archives of motion pictures represent an important part of precious cultural heritage. But unfortunately, these collections of cinematography are vulnerable to different types of distortions; specially, the effect caused by chemical support on which they are recorded becomes unstable with time, unless they are stored at low temperature. Some defects on colour movies, such as colour fading, are irreversible and hence it is beyond the capability of photochemical restoration process. Though the spatial colour algorithms, like ACE and Retinex, provide helpful solution for restoration of degraded films, there are some challenges associated with these algorithms. Firstly, they work well if the colour degradation takes place mainly along the axes of the colour space. If, instead, the degradation has resulted in an increased correlation between the colour channels, they do not perform so well. Secondly, spatial colour algorithms tend to emphasize all details in the images, including defects such as dust and scratches. This may also influence the way the algorithm restores the colours in the regions surrounding the defect. And, finally, the algorithms are inherently computationally expensive. We propose here an automatic colour correction method which eventually automates the colour fading restoration process. The proposed method uses the STRESS model - an automatic image enhancement technique that deals with correcting colour images according to the Human Visual System. We also propose an efficient preprocessing technique which will be applied to the degraded images in prior to apply STRESS algorithm. The preprocessing technique, which includes Principle Component Analysis (PCA) and some sort of saturation enhancement, will ultimately make the resulting image more appealing and acceptable to Human Visual System. We also presented a comparison scenario in case of processing time for frames between our method with STRESS and the ACE alogorithm. It is worth to mention that our method outperforms the method with ACE in case of processing time of the frames. Besides this, we depicted the comparison in case of maintainig the shape of histogram while covering the whole dynamic range between our suggested method and ACE. Since the spatial colour algorithms do not preserve some basic properties like the mean saturation and lightness value of the preprocessed image, we offered a very effective postprocessing method for preserving them. This method helps to produce pleasant final image by maintaining the expected saturation and lightness value. Apart from colour degradation, scratch line is another source of degradation in case of old movies. We have put an effort to analyze this problem and tested existing algorithm on our test movies to remove scratch line from the frames of the movies. Since, our main focus is on correcting the colour of the frames, we discussed this scratch line problem as an option and showed the effect of applying this scratch line removal algorithm on colour correction performance. We went beyond the spatial domain of the frames and extended our algorithm and implemented it on the temporal domain as well. We were able to achieve 80 percent reduction of the computational time for processing each frame in the temporal domain comparing to the processing time in the spatial domain for single frame. The implementation and achieved performance of our algorithm in case of both the spatial and temporal domain of the frames of old motion pictures were a great success in case of achieveing reduction of

processing time comparing with existing algorithm, ACE and the colour correction result is also at a good acceptable stage from the Human Visual System.

Keywords: Digital film restoration, automatic colour correction, colour constancy, image enhancement, spatial colour algorithms.

Preface

I would like to thank first the almighty creator, Allah, for giving me enough courage, knowledge and skill to accomplish this thesis work. Without his blessing it would really be difficult to complete the required task within the specific time span.

I am grateful to my parents for their love and intelligence. I thank them because they have always encouraged me to learn something new, to broaden my mind and to explore different aspects of life. This encouragement from my family helped me to build up my mind to dig deeper into different areas of research and learn new technologies.

I would like to thank my lovely wife *Sharmin Chowdhury* for her continuous support and belief in me. She always believed that I can carry on digging deeper into the subject matter in which I am interested in. I am grateful to her for being so much supportive and understanding in my difficult days.

I specially would like to show my gratitude to my supervisor, *Dr. Ivar Farup*. I thank him from the bottom of my heart for believing in my abilities and supporting me with his vast knowledge in Image processing and specially, mathematics related to Image processing. He was not only my academic supervisor, but he taught me different things about culture, history, statistics, psychology, philisophy as well. I was never worried whenever I got stuck at some points while working, because I knew that there is my supervisor with great intellect, expertise and a friendly smile on his face. Without his moral and academic support, it would be really difficult to accomplish this task.

I would like to show my gratitude to the European Union for financing my study in the Erasmus Mundus master program, CIMET. I like to thank all the coordinating university stuffs from University of Granada, University of Western Finland, University Jean Monnet and Gjøvik University College.

I thank all the *CIMET* coordinators, professors, colleagues and administrative stuffs. I also thank to all the professors and students of Gjøvik University College. I would like to thank specially *Professor Alain Tremeau*, *Professor Jon Y. Hardeberg*, *Professor Jussi Parkkinen*, *Professor Javier Hernández Andrés*, *Helene Goodsir* and *Hilde Bakke* for their kind support in academic and administrative issues. I would like to thank *Professor Jon Y. Hardeberg* for his very informative advanced colour imaging course which helped to undersatnd the colour imaging science and technology. I would like to thank *Dr. Faouzi Alaya Cheikh* for his nice and informative advanced video processing course which helped me a lot to understand diffent aspects of temporal domain image processing. I also would like to thank *Dr. Sule Yildirim* for her nice content based image retrieval course.

I would like to thank all the Norwegian colourlab members for their continuous support and understanding. I show my gratitude to all the friends and participants: *Ali Shariq Imran*, *Adnan Abdallah*, *Anna Aristova*, *Elham Rajabian*, *Fahada Fazal Elahi Guraya*, *Hameed Iqbal*, *Mekides Abebe*, *Sukalpa Chanda*, *Victoria Rudakova* and *Yaowapa Pornsamutsin*, of my psychophysical experiment.

I thank all my friends who spent time with me, who supported me with kind heart and who helped me at the time I needed help. I thank all the friends that I met in the

university, in front of the coffee machine or in the cafeteria. I would like to thank *Arne Magnus Bakke* for sharing his knowledge and expertise in colour science.

I want to thank all the smart and intelligent people who built the *STRESS* algorithm. I also would like to thank who collaborated directly or indirectly with me during my thesis work. I want to thank all the people who taught me something or a lot of things, who discussed with me about different aspects about image processing, who patiently listened to my ideas and who shared their thoughts and experience.

Finally, my gratitude to the incredibly intelligent think tank of the Retinex algorithm, *Edwin H. Land* and very well known key figure in colour correction area and inventor of ACE algorithm, *Alesssando Rizzi*.

A.B.M Tariqul Islam, 2010/07/01

Contents

List of Figures

List of Tables

1 Introduction

People, in general, are interested to know about the history of their ancestors. They are curious about the tradition, the culture, the historical moments of the past time. Many historical speech, events, sports, war footage etc was recorded on the celluloid film. Researchers are always curious to analyze the past events and disclose the reasons behind some historical movements. The famous adage, *A picture is worth a thousand words* is truly inline with the curious mind of human. Analysts and researchers can have a better understanding by viewing one scene rather than a written document. They can come to reasonable conclusion about some events if the authenticate video footage is avaialable.

For the last few decades, the cinematographic archives have been considered as the principal medium for the documentation of research and collections. But more and more old photographs become collector's items themselves. Their unique importance for research relating to the history of civilization is due to the fact that they are the only authentic sources for certain subjects or periods of time. However, in most case the storage conditions of the pictures are unsuitable, and a lot of them are in a poor state, because photographic film and paper are unstable media.

Colour photographs are most vulnerable to the degradation process. The dyes of common chromogenic processes are chemically rather unstable. Photographic films contain not only dyes but also a few other components (sensitizers, colour couplers, stabilizers etc.) which can alter with time. Since the 1950s, colour film became the standard on which millions of cinematographic works were recorded. A couple of decades later, it turned out that this process was chemically unstable, causing the fading of whole film stocks with time. Usually, a bleached colour release print is the only available record of a film. Since the bleaching phenomenon is irreversible, photochemical restoration of faded prints is not possible, hence the incontestability of digital colour restoration.

1.1 Basic movie restoration process

Movie restoration process can be pursued in photochemical way and in digital restoration process. Since each individual film reel is different from each other, the first step in film restoration involves the analysis of the film condition. Some reels might have one kind of degradation, for example only faded film problem, while others might have another, for example scratch lines and bleaching effects. So, it is very important that each film reel should be evaluated and analyzed individually depending on their condition. Depending on the physical condition of the reel, decision could be made on which restoration process could be chosen. Usually, while analyzing the film reel for physical condidition, the recording medium i.e. film reel is carefully checked for the possible bad splices, brittleness, shrinking damage, sticky area, tearness, lacquering damage, scratch lines, cinch mark damage, dust and dirt, patches, separation in emulsion, edge damage, nitrate decomposition etc.

1.1.1 Photochemical process

In photochemical process, many of these degradation factors, like lacquering damage, emulsion separation, can be taken care of by means of chemical process or by applying chemical substrate. With this process, the residues and lacquer on the film as well as repairing all film tears are performed with a clear polyester tape or splicing cement. Perforation are repaired by using special perforation repair tapes which are commercially available. Fire damage, water damage etc are also possible to recover by photochemical process.

But, many of these damages, like the fading phenomenon, cannot be removed by photochemical process only. They need further processing which is not photochemical, but it is digital. For further processing, the digital restoration process is applied.

1.1.2 Digital movie restoration process

In digital movie restoration system, the first step is the scanning of the film negatives [1]. For digital capture, the storage medium i.e. the original film negative needs to be scanned by a high quality film scanner. Different reels of the original camera negative have different exposures, so the light level of the film scanner should be adjusted properly. By means of this, the full extent of the image can be extracted. After the scanning is complete, the film can be considered as a digital representation of the film. Now, this digital representation goes through a number of automatic processes to take care of the degradation factors like, dirt/dust removal, scratch line removal, colour correction etc on the basis of image frequency analysis, motion analysis of pictures, and so on. And finally, in order to play the film in movie format, all restored images are recorded back on movie films by using film recorder [2]. The total process of digital movie restoration is depicted in Figure 1.

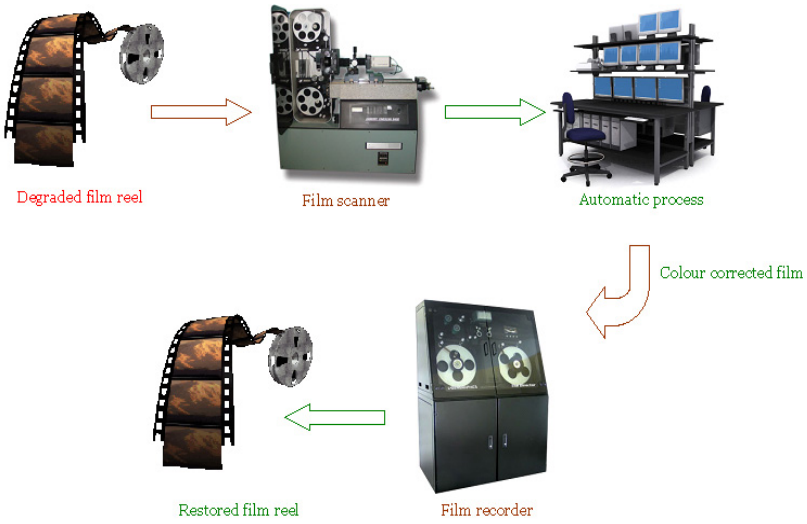

Figure 1: Digital movie restoration process.

We have paid our attention specially on the portion when the digital representation goes through a number of automatic processes. These processes are the main steps in the digital movie restoration. We have particularly focused on the colour correction techniques of digital movie restoration. Besides this, we have implemented and analyzed the

effect of scratch line removal on colour correction as well.

1.2 Spatial colour algorithms for movie restoration

In case of colour correction process, several spatial colour correction algorithms can be considered. Among them, Retinex, ACE, STRESS etc are used by many researchers.

Retinex, developed by Land and McCann [3], is one of the earliest model that works with locality of perception. In this model, locality is achieved by long paths scanning across images in case of the implementation of Retinex. A number of implementations and analysis have been carried out based on this first model. Further developments [4] [5] [6] [7] of this model mainly differed in the technique of attaining locality. All the sampling implementations use a high number of samples in order to lower the amount of noise, but they never sample the whole image in order to keep a local effect. Random Spray Retinex (RSR) [8] is a recent implementation which investigates the effect of different spatial samplings by replacing paths with random sprays i.e. two dimensional point distribution accross the image. The random sprays replaced the paths with some advantages, but still the required number of sampled points is very high. A high number of points means long computational time, which has always been the weak point in this family of algorithms [9].

ACE (Automatic colour equalization) [10], developed by Rizzi et al., is another technique to keep the local and global effect of digital images, accounting for chromatic/spatial adjustment and maximizing the image dynamic. ACE is a new algorithm for automatic enhancement of digital images, considering concurrent global and local effects. This algorithm has been implemented by following two stages: first stage deals with colour constancy and contrast tuning by accounting for chromatic/spatial adjustment, while the second stage maximizes the image dynamic by configuring the output range to implement an accurate tone mapping.

In the working process of ACE, an output image *R* is produced by the chromatic/spatial adjustment procedure. In *R*, every pixel is recomputed according to the image content. Each pixel *p* of the output image *R* is computed separately for each channel *c* by the following Eq.1.1 [10].

$$R_c(p) = \underbrace{\sum_{j \in Subset, j \neq p}}_{} \frac{r(I_c(p) - I_c(j))}{d(p,j)} \tag{1.1}$$

Here, in Eq.1.1, $I_c(p) - I_c(j)$ identifies the lateral inhibition mechanism, $d(\cdot)$ is a distance function which weights the amount of local or global contribution, $r(\cdot)$ is the function that accounts for the relative lightness appearance of the pixel. The modified version with a normalization coefficient is given in Eq.1.2.

$$R_c(p) = \frac{\sum_{j \in Subset, j \neq p} \frac{r(I_c(p) - I_c(j))}{d(p,j)}}{\sum_{j \in Subset, j \neq p} \frac{r_{max}}{d(p,j)}} \tag{1.2}$$

Here, r_{max} is the maximum value of $r(\cdot)$. The lateral inhibition mechanism is simulated by computing the difference between each pixel value and all other pixels of the selected image subset. This difference is tuned by the function $r(\cdot)$.

ACE showed potential results to mimic several characteristics of the HVS, like colour and lightness constancy, controlling the contrast etc.

Another recent development in the field of image enhancement applications is STRESS (Spatio-Temporal Retinex-like Envelope with Stochastic Sampling) developed by Kolås et al [11]. STRESS, whose properties are in line with Spatial colour Algorithms [9], is implemented with an extremely small number of sample points, using two envelopes for characterizing the local visual context [11]. This algorithm shows promising result in mimicking properties like, local contrast stretching, automatic colour correction, high dynamic range image rendering etc, of the HVS. STRESS has been successfully implemented for spatial gamut mapping and colour to gray scale conversion as well.

1.3 Our agenda

Digital movie restoration is comparatively a new area of research in the area of colour image processing. Some works have been done so far and researchers have used image enhancement model like Retinex, ACE etc in the restoration process. But, there has always been the problem of long computational time with the models like Retinex and ACE, because of large number of sample point usage; besides this, these techniques are not totally unsupervised. Some sort of user involvement took place in the implementation steps of these algoritms in colour restoration process. Sometimes, expertise of users have been taken under consideration to decide the best colour for some specific zones like sky, ground etc and then the rest of the algorithm have been processed on the basis of that zone colour selection assumption [12].

Hence, after discussing the existing methods presented above, we figured out that, there are some key places which needs to be taken care of while considering the implementation of movie restoration process.

The very important factor, which can play an important role in colour correction of old movies, is enhancing the colour information in a balanced way in all the channels. This enhancment should be applied before the degraded image is processed by colour enhance models, STRESS or ACE etc. Would this enhancement of colour help to remove the strong colour cast of the old films? If it is possible to remove the colour cast, then how much it can remove? These issues carry a big weight in the process of movie colour restoration.

Another important factor is, the preservation of the mean chromatic, saturation and lightness value of the channels of the images after they are processed by the colour image enhancement models, STRESS or ACE etc. When images are processed by these models, the processed images loose the mean saturation, lightness etc value, which are important to preserve for natural outlook from the HVS perspective. So, some postprocessing mechanism must be applied to preserve these basic properties of the image. Hence, what kind of postprocessing technique could be applied is an open research issue.

Apart from this, the well-known degrading factor in movie restoration is the scratch line. Of course, the removal of scratch line improves the perceived quality of the image. But, how much this removal of scratch lines can affect the colour correction methods. Would there be a signifcant improvement in colour restoration by applying this scratch line removal or not? This is also an open issue in old movie restoration, which needs to be investigated.

Moreover, the issue of computational time is another big factor in movie restoration. A movie consists of large number of frames. So, the restoration method must have to be computationally efficient. Which technique could be applied to reduce the computatio-

nal time at a significant level, is a primary concern as well for the researchers in movie restoration area. Would that be possible to reduce the computational time for consecutive frames in movie is a major concern. Besides this, would it be possible to develop a system which is free of user intervention, is also an open question of research in movie restoration area.

Here, in our work, we have put an effort to answer these questions and offered some solutions by which we would hopefully be able to solve these particular existing problems in old movie restoration. Our work follows the no-reference strategy, since the colour films are already degraded and there is no good quality copy of the degraded film to compare the result.

In order to automate the total process of colour restoration, we have proposed the following idea which contains the following major steps.

Firstly, we have proposed a preprocessing technique for enhancing the colour information in all the channles in a balanced way. The output image from this preprocessing technique is then processed by the STRESS algorithm so the computational time is reduced at a certain level. Since, STRESS [11] uses an extremely small number of sample points and implemented using two envelopes to characterize the local visual context, the computational time is expected to be reduced at a significant level. The properties of STRESS are in line with other Spatial colour Algorithms (SCA) [9].

Secondly, we suggested and implemented a postprocessing mechanism [13] for preserving the lightness and saturation of the preprocessed image which would be further processed by the STRESS model.

Moreover, we have tested existing algorithm to remove the scratch line from the degraded films. Scratch line removal is a well known problem in image processing area and already different algorithms exist to remove them. We pick one of the algorithms and applied to the degraded films before the preprocessing step. The reason, we applied the scracth removal process before the preprocessing step, is that, while enhancing the saturation, the scratch lines will also be enhanced and they will evntually degrade the total quality of the image. We have also analyzed the effect of this scratch removal on colour restoration. We applied this scrach removal technique as an option because this is not the main focus of the thesis.

Besides this, we have suggested and implemented an effiecient temporal domain method, by which we, hopefully, will be able to reduce the computational time for the next sequence of frames comparing with the first frame in a single cut of a movie. It is worth to mention here that, our preprocessing, spatial and temporal STRESS methods are free of user intervention. There are few parameters in the postprocessing step which needs to be tuned depending on the image.

These steps are depicted in Figure 2. In the following chapters, we have discussed each of these steps more details.

We propose here a combination of techniques for the colour digital restoration of faded movies that is based on a perceptual approach, inspired by some adaptation mechanisms of the Human Visual System (HVS). Particularly, we will be dealing with a combination of preprocessing technique for distorted images which will be further processed by the image enhancement model called STRESS and finally, the processed image from STRESS will be processed by postprocessing technique. The final processed image should have more appealing result from the Human Visual System's point of view.

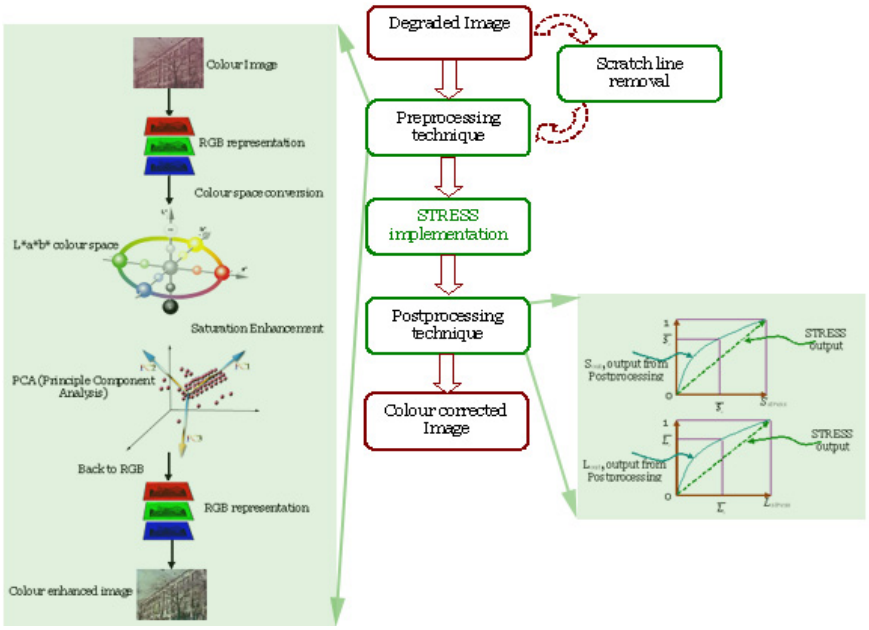

Figure 2: Basic workflow of our idea.

1.4 Thesis organization

This thesis contains seven chapters in which the state of the art, our proposed methods, different results etc are discussed. The contents of this thesis are briefly stated below.

Chaper 1, i.e. this chapter, introduces the basic idea of movie restoration, it's necessaity and existing spatial colour correction models. It describes briefly the digital movie restoration cycle and states some existing problems in digital movie restoration. This chapter also introduces our agenda for this thesis work.

In Chapter 2, we have discussed the state of the art of movie restoration. We have discussed various digital movie restoration methods proposed by different researchers. We have pointed out the advatages and disadvantages of those methods as well. Finally, we have briefly described the STRESS model for spatial colour correction.

Chapter 3 describes the preprocessing method by means of which the colour information is balanced in the different channels of the degraded image/frame.

The postprocessing method is described with resulting images in Chapter 4. various comparison results are also depicted in this chapter.

Chapter 5 contains the process of scratch line removal. The existing method which is used here is depicted step by step in this cahapter. The effect of scratch line removal of colour restoration is also discussed with resulting images and graphs.

The temporal domain method for movie restoration is presented in Chapter 6. The computational time reduction technique and the image quality issue is discussed in this chapter.

Finally, in Chapter 7, we have drawn our conclusion and some future perspectives.

Besides this, various data, information, definitions, pseudocodes are presented in the appendices.

2 State of the Art

Movie or film is considered as the oldest motion picture system with the highest image quality among existing motion picture systems. If we go through the archieve of movies, we can find out that there are many valuable cultural and scientific resources recorded as movies. Many historical moments have been captured and stored on the celluloid film. Movie and photograph are based on technology to record images by chemical transformation of materials on a film. In this technology, physical degradation of films directly degrades recorded information quality. Since the birth of motion pictures, about a hundred years have been passed. A huge amount of movie art properties have been lost already due to the degradation of film. According to the survey of the Library of Congress of the United States, about eighty percents of silent movies produced in the Unites States have already been lost or in unrestorable states. Another report states that about ninety percent of silent movies produced before 1930 and about fifty percent before 1950 have been lost.

Hence, recording in digital format is considerably the best way to inherit the current state of a film as it is. Moreover, we can bring back the original images in the films from the past by digitizing a movie by means of applying different digital restoration technology. We will discuss about different degradation factors and solutions provided by different researchers and about one particular algorithm, STRESS, which we haved used in our work. So, we will split this discussion in two sections. In the first section, we will discuss about general concepts of different existing methods for solving different degradation factor in movie restoration area and in the second section, we will shortly present the key concept of STRESS algorithm. Hence, the sections are:

1. Digital movie restoration techniques in general and
2. A brief overview of STRESS algorithm.

2.1 Digital movie restoration techniques in general

While going through the existing works done, related to movie restoration of old motion pictures, so far, we found out that this task have been accomplished in different manners. Some tasks have been pursued in the chemical phase i.e. they offer solution which deals with degradation due to lacquering damage, emulsion separation etc of frames by means of chemical substances; while some discuss about colour correction in the digital phase. In case of digital phase, they first scan the old films with a good scanner and then they apply different techniques to restore the colours of the films.

We found out that not much works have been done in this automatic processing phase of digital restoration of old movies. The existing works done so far are not so long ago. Arnaldo et al. proposed the approach [14] for old movie restoration in which, they proposed the solution by using an opening by temporal surface which is followed by a spatial geodesic reconstruction by dilation using structuring element, in order to find out the local defect detection problem and defects in consecutive images with any intersection. Their work showed that the opening by surface, for binary images, extracts

the connected components which area is greater than a specific threshold, and for gray level images, it evaluates each connected component produced by successive thresholds of the images, through binary operations. Their proposal of the movie restoration process using opening by surface consists of two principal steps; firstly, the process of opening by temporal surface applied to the image sequences that extracts connected components or domes with area greater than or equal to a specific threshold and secondly, the spatial reconstruction by geodesic dilation using structuring element applied to a single image. Guimaraes et al. also proposed a similar approach [15] of old movie restoration through using Opening by Surface. They propsed a method for restoring old movie using opening by surface which eliminates image information by area attribute, independent of the shape of its components.

Chambah et al. proposed a two stage colour correction algorithm [12] in which firstly the degaded images are non-uniformly enhanced in saturation level and then they applied the well-known Gray World [16] and Retinex White Patch method [17] for the colour balance. Their approach consists of using a different method depending on the tone of the zone to correct. In their technique, shadows and mid-tones are corrected with the gray world method and white patch method was used for correction of highlights. But, their procedure seems to be complex while implementing the graduated combination of gray world and white patch method.

Another work [18] by Chambah et al. pursued the colour correction of faded films by means of applying a two stage method. Firstly, they enhanced the saturation level and then they applied ACE algorithm [10] for further colour correction. Their approch involves some sort of user supervision and also the major drawback is the large computational time. They offered a local LUT based ACE algorithm, but the resulting image were not as reliable as the original ACE.

The concept of using rational function filters have been performed in old movie restoration by Khriji et al. [19], in which they proposed a spatial rational interpolator scheme for reconstructing the missing data, after the stationary and random defects have been localized. They focused on the blockiness and jaggedness degradation. Their work is based on the line, edge and smooth texture continuation strategy. Though the random defect removal performs at a satisfactory level, but stationary defect removal method suffers from high computational complexity.

Another comparatively new approach [20] for digital movie restoration is suggested by Rizzi et al., in which they splitted the movie into different shots and then implemented the ACE algorithm [10] for colour correction. Initially, colour correction parameters are set for each key frames for each shots and then the rest of the frames are processed with the same parameter within that shot. Their result seems satisfactory to a certain level, but again it suffers from long computational time.

Recently, a fusion based approach [21] have been proposed by Maddalena et al. for digital movie restoration. They propsed a new digital scratch restoration algorithm which achieves accuracy results higher than that of already existing algorithms and naturally adapts for implementation into high-performance computing environments. Their intention was to adopt several relatively well-settled algorithms for the problem at hand and combine obtained results through suitable image fusion techniques, with the aim of taking advantage of the adopted algorithms' capabilities and, at the same time, limiting their deficiencies. They performed extensive experiments on real image sequences which

deeply investigate both accuracy results of the presented scratch restoration approach and performance of its parallel implementation, which allows for real-time restoration.

Line scratch removal is another area of interest for the researchers in the domain of digital movie restoration. Quite a number of research have been pursued in order to remove the scratch lines from the old films. Scartch line removal process involves two steps of operation; firstly, the scratch lines or the cracks should be detected and then the scartch area should be restored by some other process.

Kyung-tai Kim and Eun Yi Kim suggested a fully automated system [22] which is capable of detecting every types of scratch within old movie clips. They were able to gain a low computational cost. This is achieved by defining the texture and shape properties from spatial domain, then using these for scratch detection. Their method involves two procedures: firstly, the input image is divided into scratches and non-scratches using a neural network (NN)-based texture classifier and secondly, some false alarms are removed by shape filtering using a morphological filter with new structuring elements defined based on the shape characteristics of scratches.

Another study [23] shows the usage of wavelet decomposition for removal of vertical scratches. This technique is based on the discrete wavelet decomposition. This transformation splits an image into approximation and detail coefficients, where the latter separate into horizontal, vertical, and diagonal representations. The algorithm reconstructs the missing data in the region of the scratch and finally the synthesis of the wavelet coefficients generates a restored version of the scratched image frame.

The technique [24] proposed by A. U. Silva and L. Corte-Real deals with both the line scratches and blotches. they designed and implemented a digital restoration chain for removing these artifacts. They defined the basic elements of the chain as the digital encoder, the artifact generator, the artifact detectors, the interpolators and the quality measurer. The role of the artifact detector is to find the degraded regions in the image and marks them. The interpolator replaces the marked degraded pixels with others that are the result of an interpolation algorithm. At the final step, the quality of the restored sequence is rated by the quality measurer.

Laccetti et al. proposed a technique [25] named as P-LSR, in which a parallel version, focusing on strategies based on both task and data partitioning, is applied to achieve good load balancing . Another work [26] related to detection and removal of vertical scratch line is proposed by Joyeux et al. They proposed a suitable detection-reconstruction approach for removing impulsive distortion and other types of deterioration from degraded image sequences. They have divided the detection procedure in two steps. First, a morphological filter provides impulsive distortions and line scratch candidates. Unlike impulsive distortions, which appear randomly in an image, line artifacts persist in nearby or the same location across several frames. Besides this, the detection process is complicated by the fact that lines occur as natural part in interesting scenes. So, they included a validation step for separating possible line defects from false detections. They used a Kalman filter for tracking the potential line artifacts over the frames. They also applied an interpolation technique, which deals with both low and high frequencies around the detected deteriorations, to achieve a nearly invisible reconstruction of damaged areas.

Bornard et al. proposed an algorithm [27] , which is inspired by texture synthesis techniques. This method is also suited to any complex natural scene and not restricted to stationary patterns. It has the property to be adapted to both still images and image

sequences and to incorporate temporal information when available while preserving the simplicity of the algorithm. Their method gives very good results in various situations without user intervention. Besides this, the computational cost is relatively low and corrections are usually produced within very short time. Their metod is used for the missing data correction in still images and in image sequences as well.

Dirt detection and removal is another area of movie restoration. Several works have been pursued in this area. Among those, a novel spatio-temporal method [28] is proposed by Jinchang Ren and Theodore Vlachos. They implemented the method for film dirt detection and recovery. Their concept consists of several steps of work. Initially, a more reliable confidence measurement of dirt is extracted for colour films. False alarms caused by motion are filtered using consistency checks among several measurements. After that, candidate dirt is detected by filtering and thresholding this confidence measurement. Lastly, bi-directional local motion compensation and ML3Dex filtering are taken for the recovery of dirt pixels. Besides this, some other works [29] [29] [30] have also been done in order to efficiently detect the dirt in the old films.

Image Inpainting is the technique of modifying an image in an undetectable form, is as ancient as art itself. It has been applied to a number of applications so far, from the restoration of damaged paintings and photographs to the removal/replacement of selected objects. Bertalmio et al. introduced a novel algorithm [31] for digital inpainting of still images that attempts to replicate the basic techniques used by professional restorators. Their method needs the user intervention for selecting the regions to be restored. After that, the algorithm automatically fills-in these regions with information surrounding them. The fill-in is done in such a way that isophote lines arriving at the regions' boundaries are completed inside. The developers of this method did not impose any limitations on the topology of the region to be inpainted.

The above discussion covers almost all the different degradation factors of old movie films. Different degradation factors and their possible solution by different researchers have been discussed briefly. In our study, we mainly focused on the colour reconstruction factor. We suggested a method which automates the process of colour restoration in old movie films. We have used the STRESS algorithm as an image enhancement model. In the next section, we have discussed briefly about STRESS algorithm.

2.2 A brief overview of STRESS algorithm

We have used the STRESS model for the image quality enhancement purpose. The reson behind choosing the STRESS model is that, it is comparatively faster and efficient and it preserves the basic shape of the color histograms in each channels. STRESS outperforms some of the existing algorithms like ACE in case of computational time. So, the choice of STRESS as image quality enhancement model was solely for the speed of computation and nice quality image.

STRESS is an image quality enhancement technique which is implemented with an extremely small number of sample points, using two envelopes to characterize the local visual context. The central part of the STRESS algorithm is to calculate, for each pixel, the local reference lighter and darker points in each chromatic channel. This is done through calculating two envelope functions, the maximum and minimum envelopes, containing the image signal. The envelopes are slowly varying functions, such that the image signal is always in between the envelopes or equal to one of them [11].

For each pixel, p_0, the maximum and minimum envelopes, E_{max} and E_{min} are computed in an iterative manner using N_i iterations. In every iteration, N_s pixels p_i, $i \epsilon \{1, ..., N_s\}$ are sampled at random with a probability of $1/d$, d being the Euclidean distance in the image from the sampled pixel to the pixel in question. An illustration, taken from [11], of the envelopes for one scanline of an image is shown in Figure 3.

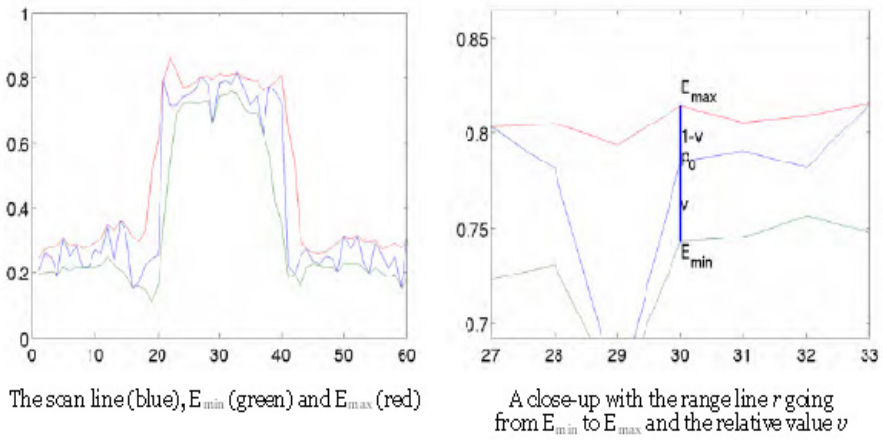

Figure 3: Illustration of the envelopes of one scanline of an image.

The range r of the samples, the relative value v of the centre pixel, $\bar{v}$, $\bar{r}$, E_{min} and E_{max} are calculated from the following equations taken from [11].

$$S_{max} = \underbrace{max}_{i \epsilon \{0,...,N_s\}} p_i \tag{2.1}$$

$$S_{min} = \underbrace{min}_{i \epsilon \{0,...,N_s\}} p_i \tag{2.2}$$

$$r = S_{max} - S_{min} \tag{2.3}$$

$$v = \begin{cases} 1/2, & \text{if} \quad r = 0 \\ (p_0 - S_{min})/r, & otherwise \end{cases} \tag{2.4}$$

$$\bar{r} = \frac{1}{N_i} \sum_{N_i} r \tag{2.5}$$

$$\bar{v} = \frac{1}{N_i} \sum_{N_i} v. \tag{2.6}$$

The envelopes are computed from the $\bar{r}$, $\bar{v}$ and the pixel (p_0) value according to the following equations:

$$E_{min} = p_0 - \bar{v}\bar{r} \tag{2.7}$$

$$E_{max} = p_0 + (1 - \bar{v})\bar{r} = E_{min} + \bar{r} \tag{2.8}$$

Here, v is always $v \epsilon [0,1]$, so, $\bar{v}$ is $\bar{v} \epsilon [0,1]$ as well. So, after getting all the necessary parameters, p_{stress}, the final STRESS value of a pixel, can be calculated by Eq. 2.9. The equation, taken from [11], for calculating p_{stress} is:

$$p_{stress} = \frac{p_0 - E_{min}}{E_{max} - E_{min}}. \tag{2.9}$$

If Eq. 2.7 and Eq. 2.8 are substituted in Eq. 2.9, the following formula, Eq. 2.10, is generated:

$$p_{stress} = \bar{v}. \tag{2.10}$$

Hense, for final calculation of p_{stress}, Eq. 2.10 is used. This technique is applied for all the three channels in the colour image and finally, we get the desired enhancement in the colour of the image. We applied this technique for restoration of colour information for degraded films. From [11], we can also get a general idea about temporal domain STRESS implementation. In [11], they have suggested to use the running average, $\bar{r}$ and $\bar{v}$, in such a way that local reference balck and white will not only depend on the current frame of the movie, but also on the previous frames as well. Since, $\bar{r}$ and $\bar{v}$ are computed in an iterative approach, performing the iteration over the consecutive frame sequences would provide a better and faster solution for motion pictures. The equations they suggested for this iterative approach are:

$$\bar{r} = \alpha r + (1 - \alpha)\overline{r_p} \tag{2.11}$$

$$\bar{v} = \alpha v + (1 - \alpha)\overline{v_p} \tag{2.12}$$

Here, $\overline{r_p}$ and $\overline{v_p}$ are the values of $\bar{r}$ and $\bar{v}$ in the previous iteration respectively. The choice of the parameter, α and the number of iteration on each frame will eventually affect how quickly the local reference black and white will change in the image.

In our study, we have also implemented temporal domain processing, but in a different manner. We will discuss about that in chapter 6.

As we will move forward to dig more depeer into our study and discuss different aspects of our implemented methods, we will find out that the processes we offered here are very efficient and faster than the existing techniques. Moreover, our method provides very good result from the HVS point of view and from the statistical point of view as well. This substantially proves the concrete base and validation of our work.

3 Preprocessing: Colour Balancing

We have described this process of balancing of colour as the preprocessing step in our suggested method. Colour balancing process involves enhancing the colour information in the channels which have lower colour information comparing to the other channels. We know that, in case of degraded images, almost all the colour information reside in a single channel. Other channels possess very few colour information. Because of this imbalance in distributing colour information among the three channels, colour cast is found in the degraded images. We can observe the illustration of this scenario in Figure 4.

We can observe that the colour cloud is mostly concentrated into a single channel and other channels get very low colour information. Hence, in order to enhance the colour information in the other channels, we need to apply some techniques. These techniques are described in the following sections.

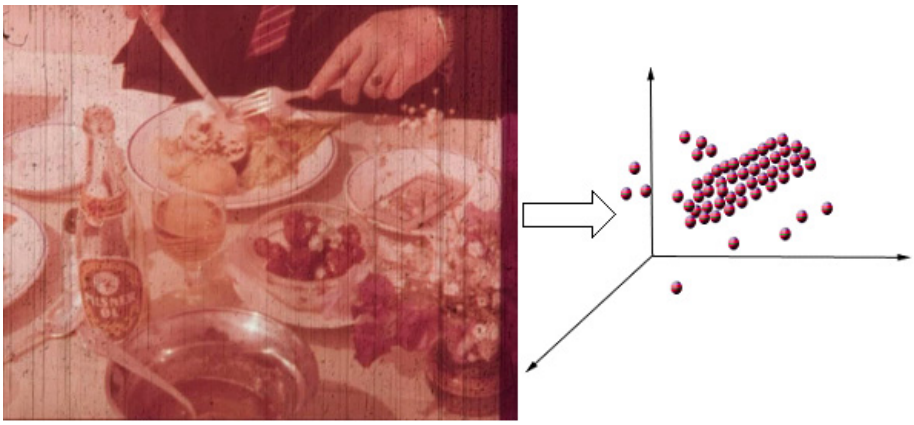

Figure 4: Illustration of an image with colour distortion.

3.1 Method

This technique involves two main steps:

1. Transformation of colour space into a more uniform one and
2. Enhance the colour information of the channels that have less information.

3.1.1 Transformation of colour space

For the transformation of colour space, we choose CIELAB space, since it is more uniform and we can control the lightness and chromaticity parameters individually, independently and efficiently. Since, our degraded image is in RGB space, we used the following formula to transform the image to the CIELAB space. It is worth to mention that there are no simple formulas to convert RGB values to L*a*b* values. At first, the conversion to CIEXYZ is required and then the resulting XYZ values can be converted to CIELAB space i.e. to L*a*b* values.

RGB to XYZ conversion:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [M] \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{3.1}$$

$$where, \quad r = R^{\gamma}, \quad g = G^{\gamma}, \quad b = B^{\gamma}$$

XYZ to L*a*b* conversion:

$$\begin{aligned} L^* &= 116f\left(\frac{Y}{Y_n}\right) - 16 \\ a^* &= 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \\ b^* &= 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right] \end{aligned} \tag{3.2}$$

$$where \quad f(t) = \begin{cases} t^{\frac{1}{3}} & t > (\frac{6}{29})^3 \\ \frac{1}{3}(\frac{29}{6})^2 t + \frac{4}{29} & otherwise \end{cases}$$

Here X_n, Y_n and Z_n are the CIE XYZ tristimulus values of the reference white point, the subscript n refers to *normalized*.

After implementing the conversion, we get the corresponding L*a*b* values from the original degraded RGB image. These L*a*b* values are used for the next step to enhance the colour information in the degraded image. The graphical illustration of this conversion is depicted in the following Figure 5.

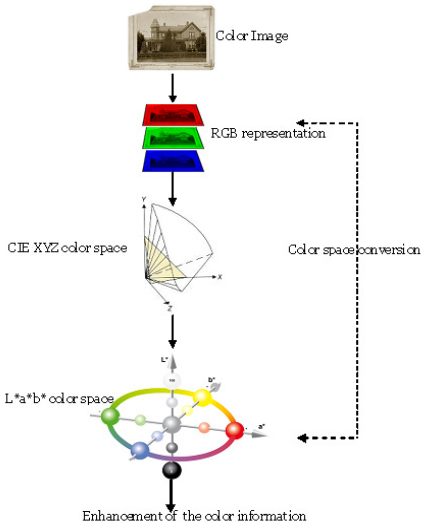

Figure 5: Colour Space transformation:RGB to L*a*b*.

3.1.2 Enhancing the colour information

This step concerns about the enhancement of the colour information in the channels which have comparatively less colour information. We used Principal Component Analysis (PCA) to decorrelate the data among different channels. So, if we apply PCA to the image which is in CIELAB colour space, then, the colour information will be splited into three different Principle components (PCs). We can have the illustration of the graphical view of this split in Figure 6.

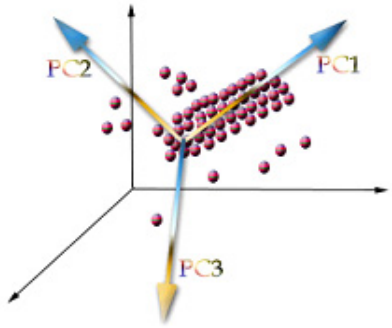

Figure 6: Illustration of PCA representation of the available colours in the image.

It is obvious from the Figure 6 that, the first PC is along to the direction which have more colour information i.e. along to the most stretched colour cloud; which eventually represents the luminance information. The second and third PCs represent chrominance information [32].

The principal question is how we are going to enhance the colour information from this kind of degraded films. For that, we have designed the working flow as the following block diagram in Figure 7.

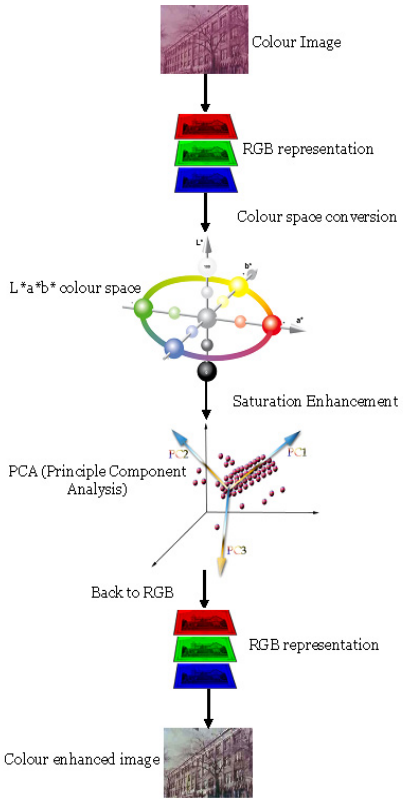

Figure 7: Block diagram of preprocessing.

From the block diagram in Figure 7, it is clear that at the very first stage, we work with the RGB image which is a strongly degraded frame from a movie. Then, we convert

the colour space to L*a*b*. The reason behind this conversion is to get a more uniform colour space so that we can expand the colour information in all the channels more uniformly. The later step is the balancing of colour i.e. enhancing the saturation.

For saturation enhancement, we have used Principle Component Analysis (PCA). PCA de-correlates the colour information among the channels and reduces the dimensionality to a certain level. We did not change the lightness information i.e the L* channel information of the image in the CIELAB colour space and worked with the other two dimensions, a* and b*. We implemented the following steps in PCA to enhance the colour information.

Let us consider, I as the image in the CIELAB space, I_m is the mean of image I, *cov* is the covariance matrix, *V* is the eigen vector matrix, *D* is the eigen value matrix. Let I_d be the independent colour channel axis.

So, the PCA implementation was in the following way:

$$\begin{aligned} I_d &= (I - I_m) \\ cov &= (I_d^T \times I_d) \end{aligned} \tag{3.3}$$

Then, we calculated the eigen Value D, Eq. 3.5 and eigen vector V, Eq. 3.4 from the covariance matrix, cov. We *diagonalized* the eigen values, Eq. 3.6 and then sorted them in the descending order, Eq. 3.7. Here, we kept the L* axis unchanged and worked with a* and b* axis, so there will be four eigen vectors and two eigen values.

$$V = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} \tag{3.4}$$

$$D = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \tag{3.5}$$

$$D = diag(D) \tag{3.6}$$

$$D = sort(D) \tag{3.7}$$

Then we projected those eigen vectors on the independent components, Eq. 3.8.

$$I_d = I_d \times V \tag{3.8}$$

Then, we calculated a parameter, called m, Eq. 3.9, which is the ratio of the squared root of the eigen values, d_1 and d_2.

$$m = \sqrt{d_1}/\sqrt{d_2}. \tag{3.9}$$

Then, we calculated the first, Eq. 3.10 and the second, Eq. 3.11 independent axis as follows:

$$I_{d1} = I_{d1} \times 2x \tag{3.10}$$

$$I_{d2} = I_{d2} \times m^x. \tag{3.11}$$

In this way, the channel which have the less colour information gets more enhanced than the channel which have comparatively more colour information. In order to enhance

the saturation level we have multiplied the second independent axis with m^x and first independent axis with $2x$.

Since, we intend to enhance the second independent axis more than the first independent axis, we have multiplied the first indpendent axis with a factor of 2x and we have multiplied the second independent axis with m^x. It is worth to mention that, m^x is bigger than 2x, since m is never smaller than 2.5 and our highest estimation of x is 0.65. In this way, we can enhance the colour information in a balanced way along both the axes. Value of x is gained from a trial and error basis.

It is worth to mention that, after enhancing the colour information in the CIELAB space, the colour enhanced image is brought back again to the RGB space for further processing.

We defined 5 different saturation parameters and picked the one which the observers liked most. The different parameters, x and their coefficient $2x$ for multiplying with 2nd PC are given in Table 1.

Table 1: Different values of parameters x and 2x.

Value of x	Value of coeffecient 2*x
0.35	0.7
0.45	0.9
0.5	1.0
0.60	1.2
0.65	1.3

The effect of implementing these parameters are depicted in Figure 8, Figure 9 and Figure 10. We have shown here, in Figure 8, Figure 9 and Figure 10, the colour enhanced image with the parameter which the observers liked most i.e. with $x = 0.5$. Once we have decided the value of $x = 0.5$, it is used as standard for colour balancing for any degraded images for preprocessing. Later, in this section, we have shown the colour balancing of different type of degraded images with this value of x.

In Figure 8, Figure 9 and Figure 10, we can observe that the intensity histogram of the colour enhanced image covers much more dynamic range than the intensity histogram of the degraded image. Besides this, the colour histogram of colour enhanced image shows that the individual colour channels (R, G, B) also cover more dynamic range than the individual colour channels (R, G, B) of the degraded image.

Moreover, from Figure 8, Figure 9 and Figure 10, we can have a clear view of the amount of enhancement in the CIELAB space by observing the a^* versus b^* scatter plot in the CIELAB space. In Figure 8, Figure 9 and Figure 10, in the scatter plot, the red area indicates the amount of colour information in the degraded image, whereas the green area indicates the amount of colour information in the colour enhanced image. It is clear from the scatter plot that, the colour information has been enhanced in case of the colour enhanced image.

The enhancements of colour information for the images in Figure 8, Figure 9 and Figure 10, are presented respectively in Figure 11, Figure 12 and Figure 13 in the CIELAB colour space.

Figure 14 shows all the five different colour enhanced images with five different values of x with the a^* versus b^* scatter plot.

For picking up the most chosen value of x, we arranged a psychophysical experiment.

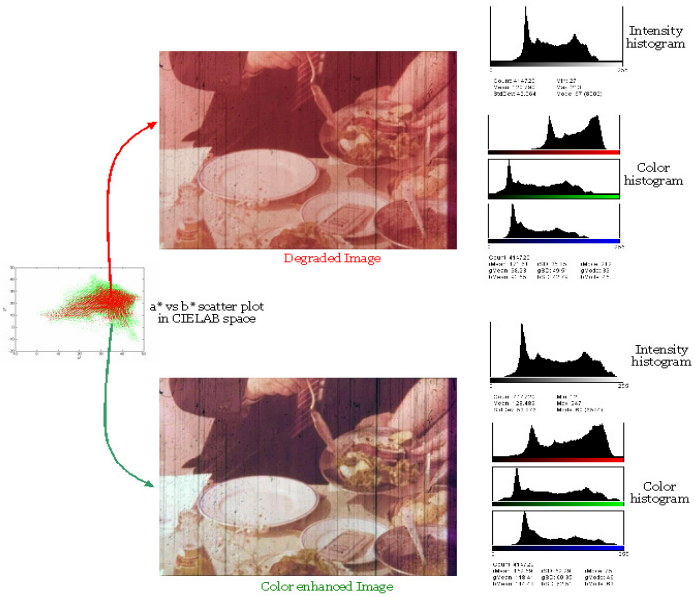

Figure 8: Enhancement of colour information in the preprocessed Image-1.

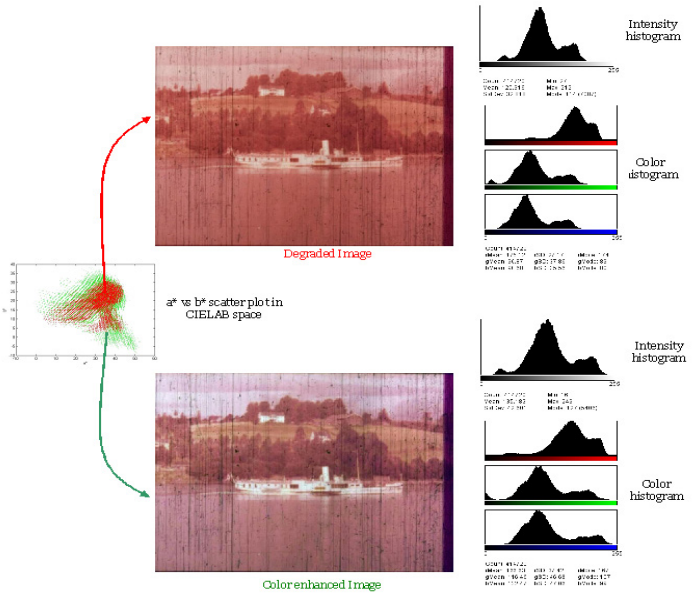

Figure 9: Enhancement of colour information in the preprocessed Image-2.

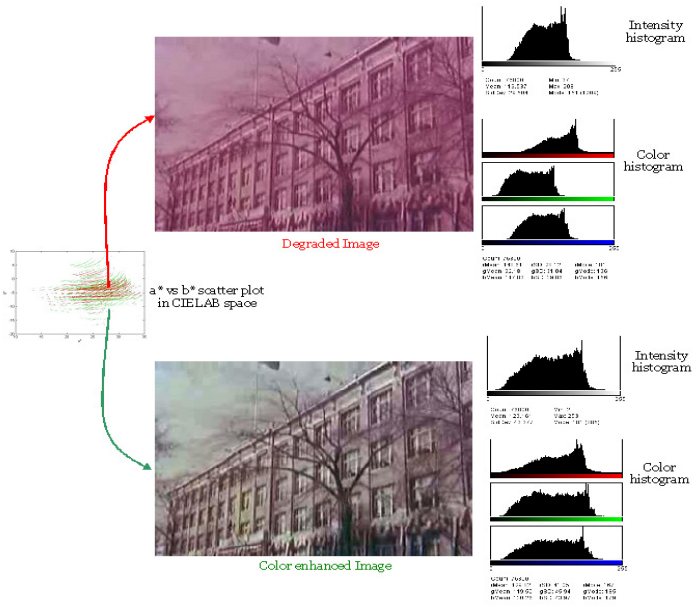

Figure 10: Enhancement of colour information in the preprocessed Image-3.

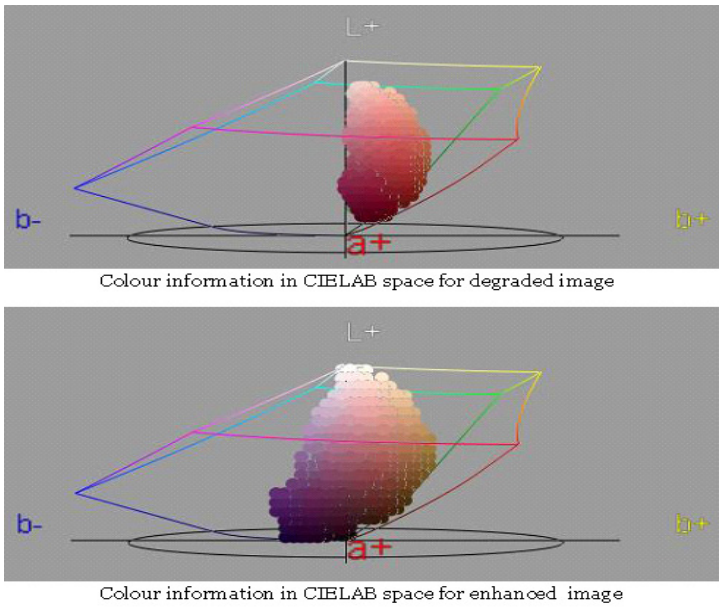

Figure 11: CIELAB representation of colour balance of the preprocessed Image of Figure 8.

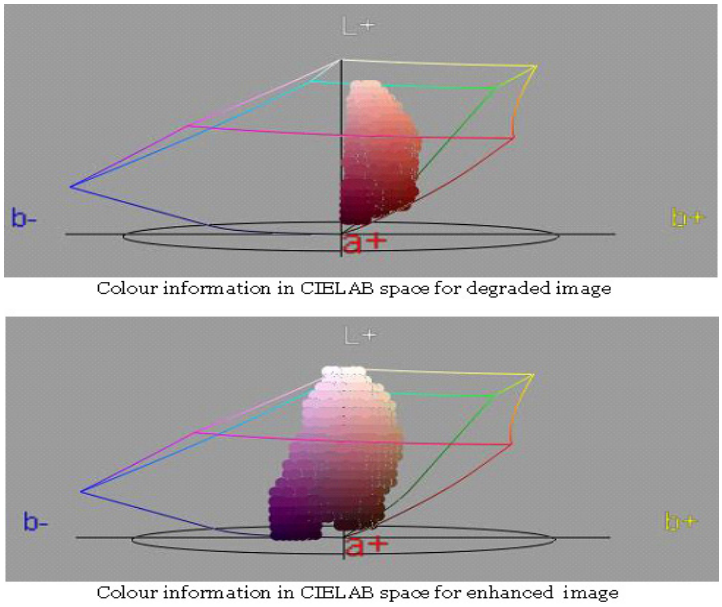

Figure 12: CIELAB representation of colour balance of the preprocessed Image of Figure 9.

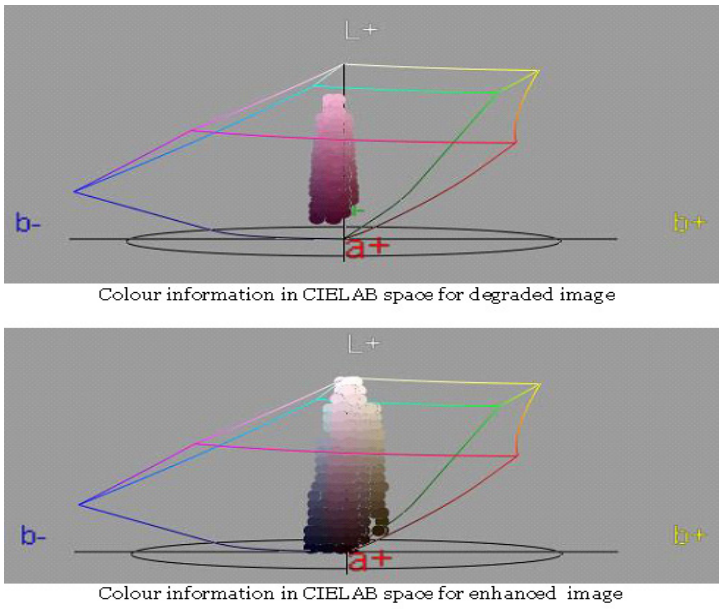

Figure 13: CIELAB representation of colour balance of the preprocessed Image of Figure 10.

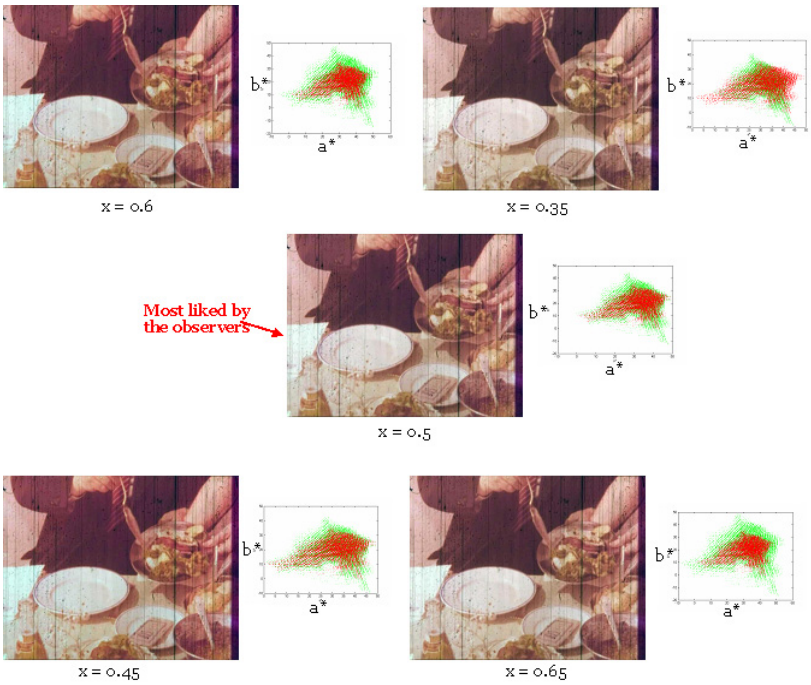

Figure 14: Colour information enhancement in the preprocessed images.

In that experiment, we asked the observers to select one image from 5 variations of saturation of the same image. On the monitor, total six images were showed at a time: the real degraded image and 5 different saturation enhanced version of that degraded image. There were total 21 different images. The total number of observers were 10. So, there were total 210 observations for 21 images with 10 observers. A sample snapshot of the experiment window is depicted in Figure 15.

We can see in the experiment window in Figure 15 that there are total 6 images in the window. The top-middle one is the original degraded image and the rest five are the five different variation in saturation of that image. Observers were asked to select anyone of the five which they prefer most from the natural colour perspective. After selecting one, the next window with another image were presented. In this way, total 21 windows with 21 different images were displayed and the observers were asked to pick the most preferable one for him/her self. After the experiment was finished, we collected the results from the observers and analyzed them statistically. The result of this analysis is discussed in the following section. It is worth to mention here that observers selected the image with saturation parameter in the third row of the Table 1, i.e. $x = 0.5$.

3.2 Result

From Figure 8, Figure 9, Figure 10 and Figure 14, we can clearly see the difference between the original degraded image and the image with colour enhancement. In the enhanced image, the colour in different channels are well balanced than the original degraded version. The food colour, skin colour, colour of the flower, bottle label, colour of the soft drink etc in Figure 8 and the sky colour, grass colour, colour of the ship in Figure 9 etc are more prominent and natural in the colour enhanced version of the

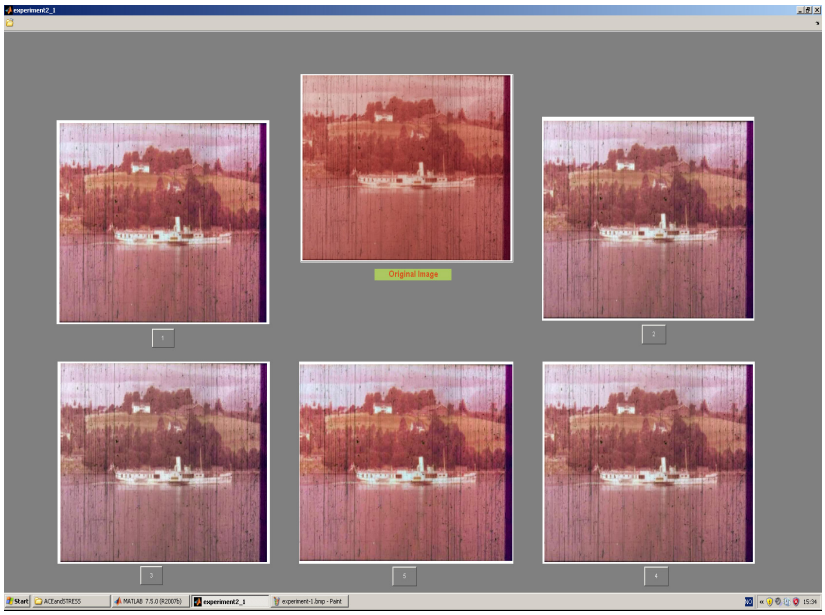

Figure 15: Snapshot of the window of the psychophysical experiment.

image.

Considering the depicted resulting images, we can come to the conclusion that the preprocessing method efficiently balance the color information in all the channels of the image. The resulting image have very less colour cast and the colour of different objects are much more prominent than the degraded image.

3.2.1 Statistical analysis of the results

We used the method of statistical analysis to analyze our results from the psychophysical experiment. In order to prove any hypothesis in statistics, anyone have to define a `NULL` hypothesis and a signifinace level first. Then, the process of the analysis should be pursued in such a way that the `NULL` hypothesis could be rejected and eventually, the opposite of the `NULL` hypothesis i.e. the `alternative` hypothesis is accepted.

So, we define the NULL Hypothesis and the significance level for the saturation level 3, Figure 16, of our experiment like below:

`NULL Hypothesis`= The observer finds all the five variations of images similar and they can select any one of the five images with equal probability.
and
Significance Level= 0.05 i.e. 5%.

These defined parameters eventually mean that if the significance level i.e. tolerance level goes beyond 5% or becomes equal to 5%, we will not reject the `NULL Hypothesis` otherwise we will reject the `NULL Hypothesis` and eventually prove that such an incidence can occur by chance.

In our experiment, we got 21 P values for 21 Images and 10 observers for each images. So, we have 210 observations. We found that for first Image, P value = 0.0064. Since, P is below 0.05 for the first image, we can reject the `NULL Hypothesis`. It is to be noted here that, this is a double sided calculation of *binomial distribution*. Since the statistical value might be negative in some cases in the double sided distribution with `E`

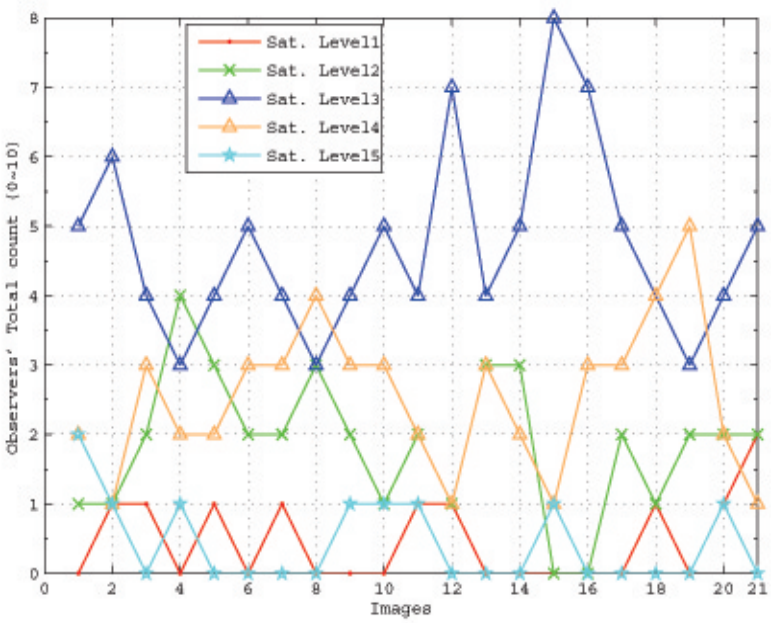

Figure 16: Statistical analysis of the psychophysical experiment for the images.

value of *2*, we used this formula, Eq. 3.12:

$$P = 1 - CDF(x, n, p). \tag{3.12}$$

Here, p=0.2, n=10 and x= outcome in each trial. CDF represents the Cumulative Distribution Function of *binomial distribution*.

Similarly, for the second image, we get P = 8.6436e-004. Here also, P is below 0.05; so, we can reject the `NULL Hypothesis`.

The following Table 2 shows the results from the experiment with different images and their corresponding P values.

So, from Table 2, we can see that 3 images have P values greater than the significance level 0.05 and 18 images have P values less than the significance level.

If we calculate the P value for all the images and for all the observers for Saturation Level 3, we get the P value = 0 which is less that the significance level 0.05. Here, x=99, n=210 and p=0.2.

So, we can reject the `NULL Hypothesis` and can conclude that the outcome we have observed is improbable due to chance.

3.3 Discussion

In order to reduce the colour cast of the degraded image, we used this preprocessing technique for colour information enhancement. After the preprocessing is completed, we can observer more balanced colour information in all the three channels. It can be easily distinguised, from Figure 8, Figure 9, Figure 10 and Figure 14, the difference between the degraded and enhanced image with the presence of strong colour cast in the degraded image, while the enhanced image doesn't have such strong cast.

Table 2: Experiment results for different images and their corresponding P values.

Image no.	P value
Image 1	0.0064
Image 2	8.6436e-004
Image 3	0.0328
Image 4	0.1209
Image 5	0.0328
Image 6	0.0064
Image 7	0.0328
Image 8	0.1209
Image 9	0.0328
Image 10	0.0064
Image 11	0.0328
Image 12	7.7926e-005
Image 13	0.0328
Image 14	0.0064
Image 15	4.1984e-006
Image 16	7.7926e-005
Image 17	0.0064
Image 18	0.0328
Image 19	0.1209
Image 20	0.0328
Image 21	0.0064

The intensity histograms, in Figure 8, Figure 9, Figure 10 and Figure 14, shows that the enhanced images covers more dynamic range than the degraded image. Besdies this, the overall shape of the histogram is also maintained. Moreover, the colour histogram also shows the enhancement information. Besides this, the scatterplot also proves the enhancement of the colour information in the enhanced image.

We have tested our preprocessing method with several kinds of images and it proves it's efficiency in case of the faded images. Besides this, our method is totally unsupervised, no user intervention is necessary for this process. Our method performs quite well in comparing with the method in [12] which involved user supervision for selecting the zone colour.

4 Postprocessing Method

The original degraded images contain strong colour cast, which is also known as the fading effect. In the degraded image, most of the colour information is concentrated in a single channel. In order to decorrelate the colour information and enhance the colour information in the other channels, we applied the preprocessing technique. The resulting images from the preprocessing technique have very less colour cast comparing with the original degraded image. Now, the colour information of these resulting images should be restored, since this is the focus of our work. In order to restore the colour information, we applied STRESS algorithm [11]. We also applied our postprocessing technique[1] [13] to the resulting images from STRESS algorithm. This postprocessing is applied mainly to preserve the lightness and saturation level of the preprocessed images in the final ouput images from STRESS algorithm. The basic workflow of this chapter is depicted in Figure 17.

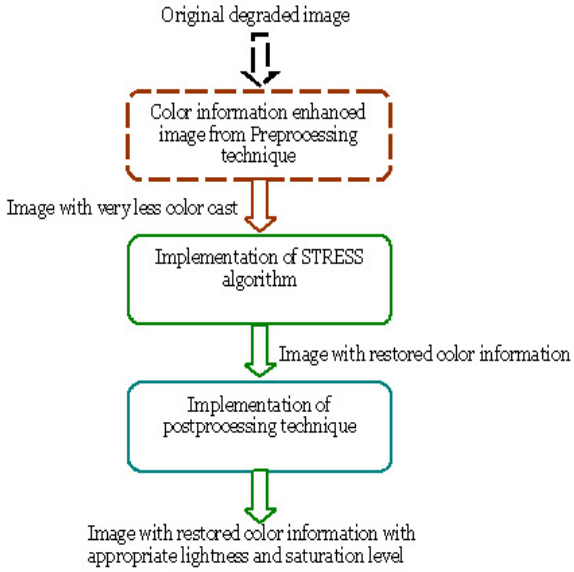

Figure 17: Work flow of STRESS and postprocessing technique.

4.1 Method

The working process of colour restoration consists of three main steps. The first two steps, the preprocessing technique and the STRESS algorithm have been discussed already in Chapter 3 and Chapter 2 respectively. Here, we will discuss about the working criteria of our postprocessing technique.

[1] Most of the logical and coding part of this technique was developed during internship; the develeopment and coding continued during Adavnced colour Imaging course and it was finished in March, 2010. We submitted this work in the conference, EUVIP 2010, France and it got accepted.

Like any other image enhancement models, when STRESS is applied to an image, the output that is produced, doesn't preserve the basic properties like mean chromatic channel value, mean lightness or mean saturation etc [13]. So, the images produced from STRESS algorithm, seem to be oversaturated and of low lightness level comparing to the preprocessed image. Hence, we applied our postprocessing technique [13] to preserve the mean lightness and mean saturation level of the preprocessed image and pass it to the image after STRESS is implemented. The workflow of this stage, a modified form taken from [13], is depicted in Figure 18.

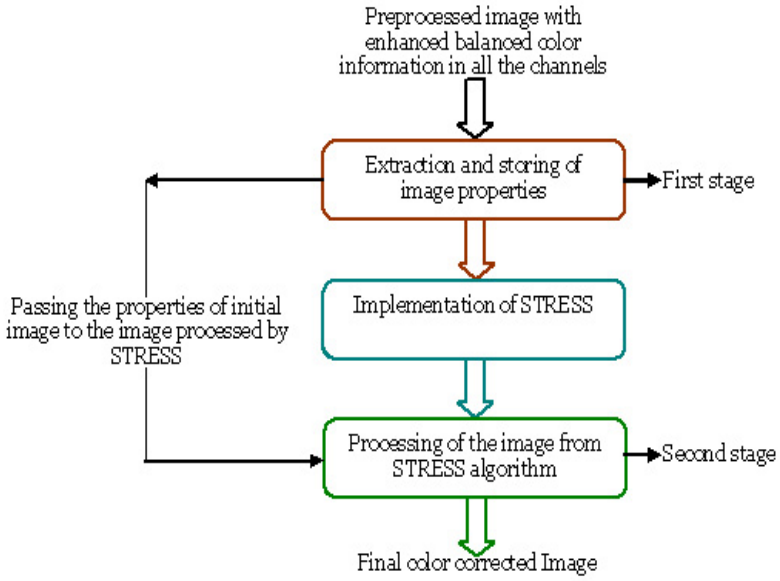

Figure 18: Work flow of postprocessing technique.

From Figure 18, we can see that, in the first stage, properties of images like mean lightness and mean saturation are extracted and stored. After that, the image is processed with STRESS algorithm. After implementation of STRESS, in the second stage of Figure 18, the resulting image is further processed with the properties stored in the first stage. The resulting image is the final image with colour correction. For postprocessing the image from STRESS, we used the KMG(Keep Mean by Gamma) technique [13] for preserving the mean lightness and saturation for the final output image. For extracting the saturation and lightness value, we have to convert the RGB colour values to HSL and then extract and store the values of saturation and lightness. We did not consider hue for this postprocessing. After extraction of values, we again convert back to RGB and implement the STRESS algorithm. Then, we convert RGB values of the image from STRESS model to HSL values and applied the KMG postprocessing technique and finally, image is converted back to RGB space.

The postprocessing method, KMG, for preserving the lightness and saturation value are illustrated in the diagram in Figure 19, modified form taken from [13]:

In Figure 19, in the left diagram, $\overline{L_o}$ and $\overline{L_S}$ indicate the mean value of lightness channel of the initial image and of the image from STRESS algorithm respectively. The left diagram in Figure 19 shows the final value, L_{out}, for lightness channel, after processing the saturation channel value, L_{stress}, from STRESS algorithm. L_{out} can be obtained by

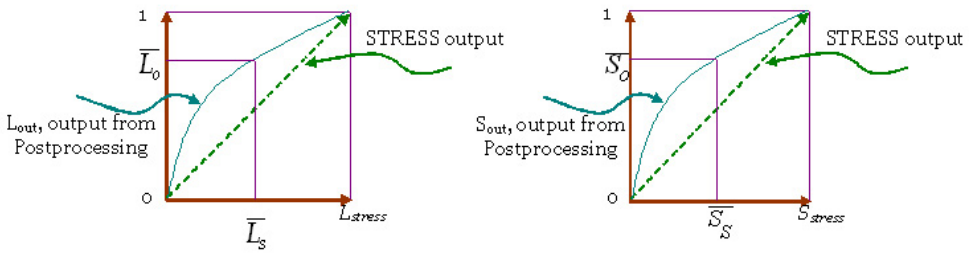

Figure 19: Diagram of KMG for lightness(left) and saturation(right).

the following equation, modified form, taken from [13]:

$$L_{out} = L_{stress}^{\gamma^L} \tag{4.1}$$

$$\gamma^L = \frac{\ln \overline{L_o}}{\ln \overline{L_s}} \tag{4.2}$$

The saturation value, S_{out} can also be derieved in the same process as L_{out} by the following equations:

$$S_{out} = S_{stress}^{\gamma^S} \tag{4.3}$$

$$\gamma^S = \frac{\ln \overline{S_o}}{\ln \overline{S_s}} \tag{4.4}$$

The results, obtained from this section, are discussed and depicted in the following section. Besides this, we arranged a psychophysical experiment to compare our final output image with an existing algorithm, ACE. After the preprocessing step i.e. after the colour information enhancement, we implemented ACE algorithm and then compared the result obtained with STRESS implementation. In that experiment, along with the original degraded image, we displayed the images from the method by ACE and the image from the method by STRESS. We did not inform the observers which image is from the method by ACE or which image is from the method by STRESS. Moreover, we changed the order to display the images from both the methods for different images. We tried to make the environment bias free.

We asked the observers to select one image from two different outputs from the two methods of the same image. On the monitor, total three images were showed at a time: the real degraded image and two different outputs from the two methods of the same degraded image. There were total 21 different images. The total number of observers were 10. So, there were total 210 observations for 21 images with 10 observers. A sample snapshot of the experiment window is depicted in Figure 20.

We can see in the experiment window in Figure 20 that there are total 3 images in the window. The top-middle one is the original degraded image and the rest two are the resulting images from the method with STRESS and ACE. Observers were asked to select anyone of the two images which they prefer most from the natural colour perspective. After selecting one, the next window with another image was presented. In this way, total 21 windows with 21 different images were displayed and the observers were asked to pick the most preferable one for him/her self. After the experiment was finished, we

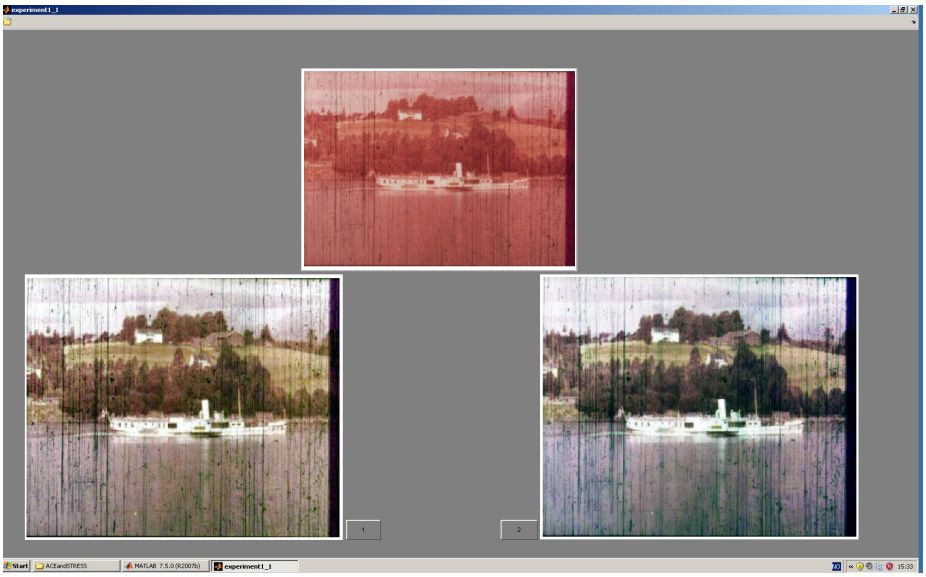

Figure 20: Snapshot of the window of the psychophysical experiment.

collected the results from the observers and analyzed them statistically. The result of this analysis is discussed in the following section.

4.2 Result

In this section, we have discussed about the outcome of implementing our technique for restoring the colour for degraded films. We will discuss the results in four subsections. Firstly, we will discuss the outcome from STRESS and postprocessing technique, then we will discuss the comparison result between images from STRESS algorithm and ACE algorithm and then we will present the result of psychophysical experiment. This psychophysical experiment was arranged to find out which image, STRESS or ACE, the observers liked mostly. Lastly, we will discuss the comparison result in case of computational time taken by STRESS and ACE algorithms for processing the images for colour correction. Hence, the subsections are:

1. Outcome from STRESS and postprocessing technique.
2. Comparison of images from STRESS and ACE.
3. Comparison result of computational time:STRESS vs ACE.
4. The outcome of psychophysical experiment.

4.2.1 Outcome from STRESS and postprocessing technique

In this subsection, at first we will present the results that are produced by implementing the STRESS algorithm on the preprocessed image. Then, we will depict the results that are produced by implemeting the STRESS and our postprocessing techniques on the preprocessed image. We will present degraded images and their colour restoration from two different old movies.

In Figure 21, we can observe that the image on the upper right position has some kind of burning effect, it seems that this image is oversaturated. Besides this, it looks a bit darker than the preprocessed image(lower left position).The reason behind this

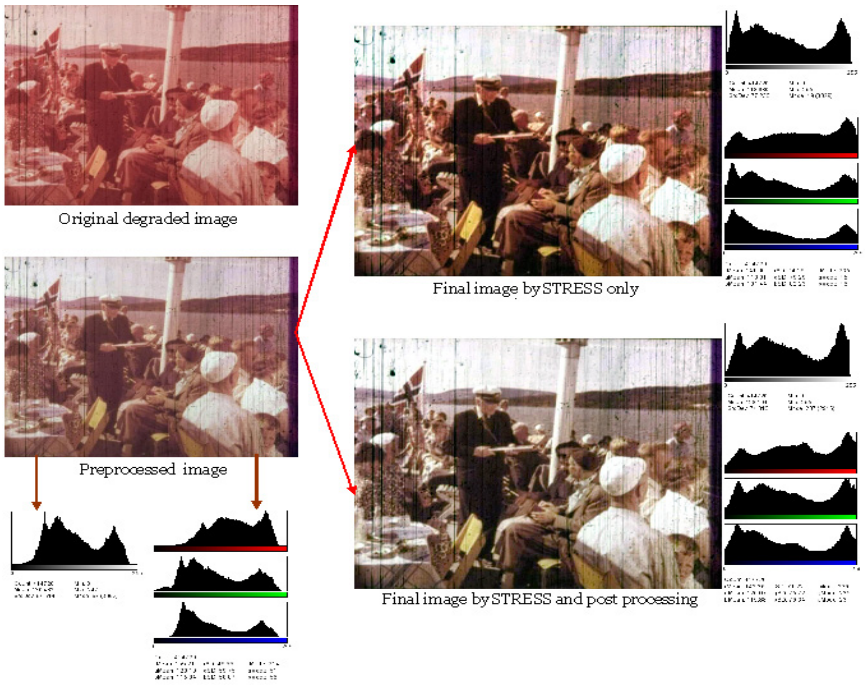

Figure 21: Effect of postprocessing on the STRESS image-1.

is the image enhancement model, STRESS. We know that after processing by STRESS, some basic properties of the image are lost in the final resulting image [13]. So, in order to maintain those basic properties like the saturation level and the lightness level, we implemented our postprocessing technique and finally, in the resulting image, we can observe the colour corrected image with expected saturation and lightness level. We can figure out this effect by looking at the intensity and colour histograms of the images on upper right and lower right position. Here, the upper right image is the image produced by only implementing STRESS on the preprocessed image.

We can observe similar effects of postprocessing technique, on the resulting image produced by STRESS, in Figure 22 and Figure 23 as well.

In all the three figures, Figure 21, Figure 22 and Figure 23, we can observe that the image, produced by only STRESS, is a bit more oversaturated and a bit less darker than the expected level. But, the image, which is produced by both STRESS and postprocessing technique, seems quite pleasant looking with better saturation and lightness level. Their corresponding intensity and colour histograms also state the similar observation.

We have applied our method on another degraded movie and found the similar colour correction result. The effect of our method on that movie is depicted in Figure 24, Figure 25 and Figure 26.

In Figure 26, we have presented several degraded images and their resulting images produced by only STRESS and by both STRESS and postprocessing. Here, in Figure 26, the left column contains the original degraded image, the middle column represents the resulting images produced by only STRESS and the right column shows the resulting images produced by both STRESS and postprocessing technique. In all the figures, Figure 24, Figure 25 and Figure 26, the resulting images produced by both STRESS and

Figure 22: Effect of postprocessing on the STRESS image-2.

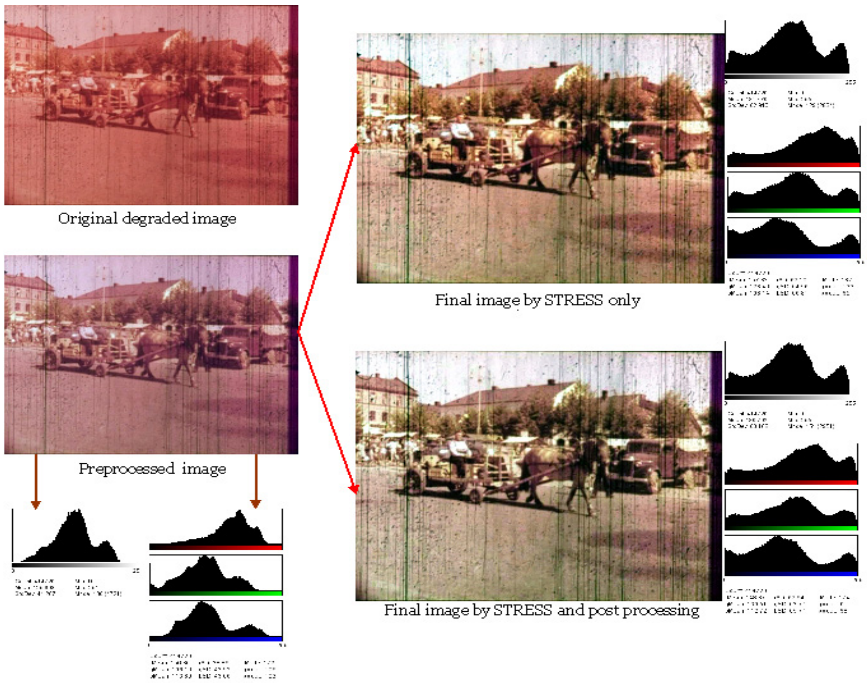

Figure 23: Effect of postprocessing on the STRESS image-3.

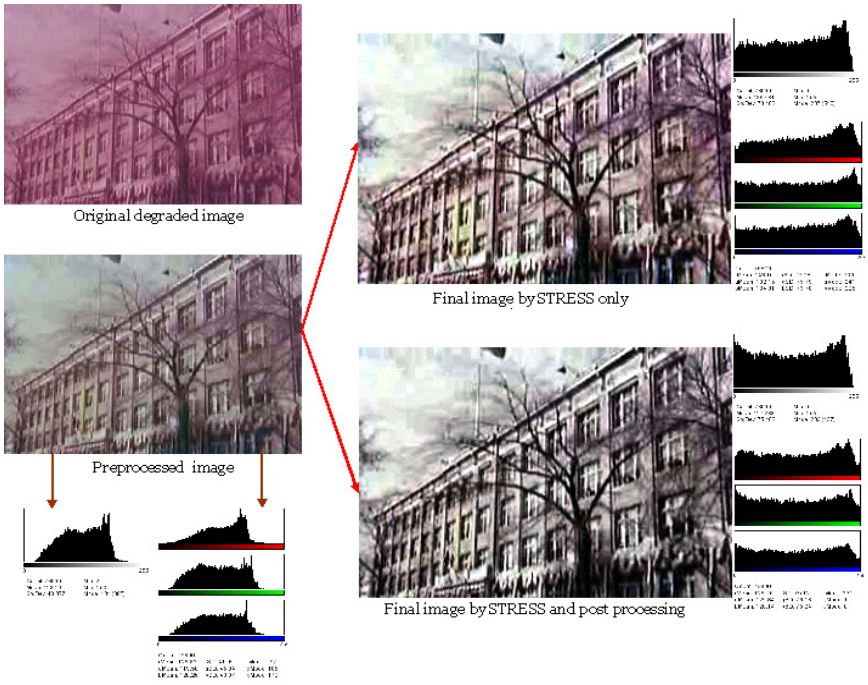

Figure 24: Effect of postprocessing on the STRESS image-4.

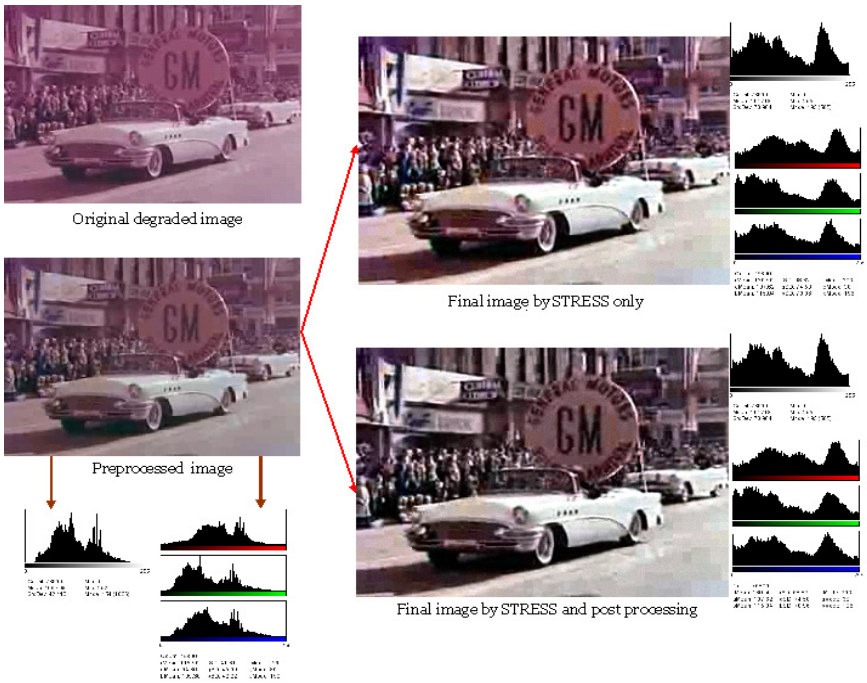

Figure 25: Effect of postprocessing on the STRESS image-5.

Figure 26: Effect of postprocessing on the STRESS image-6.

postprocessing technique, depicts more pleasant looking images, from both perspective of the saturation and lightness level.

4.2.2 STRESS and ACE images' comparison:dynamic range and histogram

In this subsection, we have presented the resulting images produced by our method with STRESS and method with ACE. We have gone through same processing steps both with STRESS and ACE. In case of the method with ACE, we just replaced the STRESS algorithm and then put ACE and performed similar operations that we did with STRESS algorithm.

In Figure 27, we have shown the two images produced by the method with STRESS and the method with ACE. The left image, in Figure 27, is the original degraded film, the upper right one is the image from the method with STRESS and the lower right one is the image from the method with ACE. We have presented the original degraded image and the resulting images with corresponding intensity histogram and the colour histogram as well.

From Figure 27, we can see that both the methods with STRESS and ACE have restored the colour; but having a closer look at the histogram of these methods, we observe that, the red channel in the colour histogram(marked with red ellipse) of the method with ACE, is almost cut off. But, the method with STRESS doesn't create such cut off in any of the channels in the colour histogram; STRESS enhances the dynamic range in a balanced way for all the three channels. We can also look at the intensity histograms and observe that, the method with STRESS covers more intensity information than the method with ACE. So, we found out that, the method with STRESS enhances the dynamic range while maintaining the basic shape of the histogram of the original image; but the method with ACE does not maintain the basic shape. We can observer a similar issue in

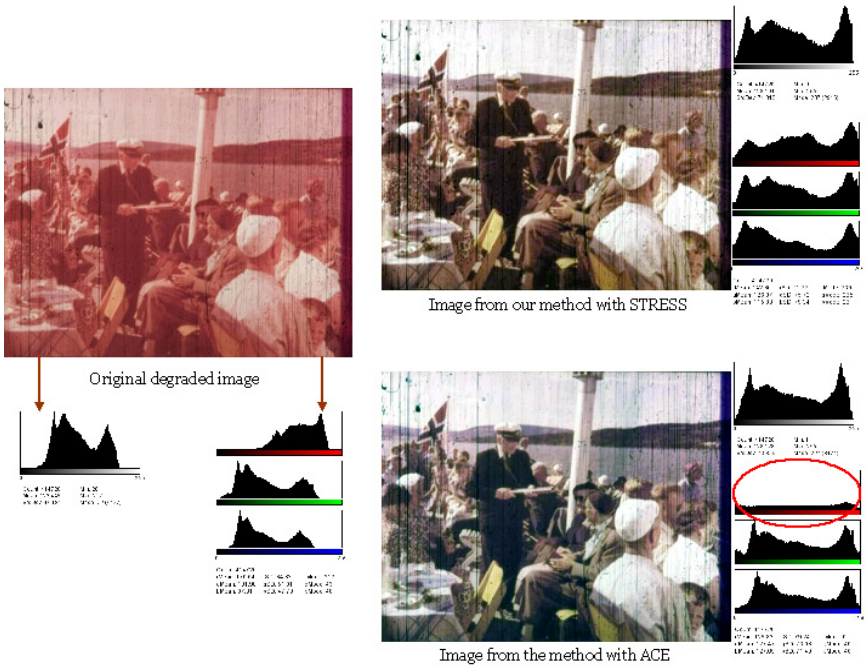

Figure 27: Resulting images-1 from the method with ACE and with STRESS.

Figure 28 as well.

In Figure 28, in addition to the cut off of the red channel in colour histogram with the method with ACE, in the intesity histogram, we can find a drop off of the intensity in the upper region(marked by red rectangle). As a result, the final image have some darkening effect in some areas in the image.

From Figure 29, we can see that, the method with ACE cuts off the lower region of the intensity histogram, while the method with STRESS doesn't make such cut off; rahter, the method with STRESS maintains the shape of the histogram of original image and also enhances the dynamic range to significant level. Besides this, the blue and green channels in the colour histogram of the method with ACE is also cut off in the lower region.

In Figure 30, we have taken a specific region of the resulting image from both the methods with STRESS and ACE and presented the effect of restoration on this particular area of the image by intensity and colour histogram.

We can clearly observe that, for the particular selected area, the method with STRESS enhanced the dynamic range while maintaining the shape of the histogram of the original image. It also covers more intensity information than the method with ACE. Besides this, in the colour histogram, we observe that there are cut off region(marked with red rectangles) in the green and blue channels with the method with ACE; but, the method with STRESS does not produce any such distortion in the histogram.

4.2.3 Comparison result of computational time:STRESS vs ACE

Computational time is a big factor in measuring the efficieny of any method or algorithm. Preserving a certain quality, if any algorithm can reduce the computational time down to a significant level, then it proves the computational efficiency of that particular algorithm. We implemented our methods: preprocessing, STRESS and postprocessing on

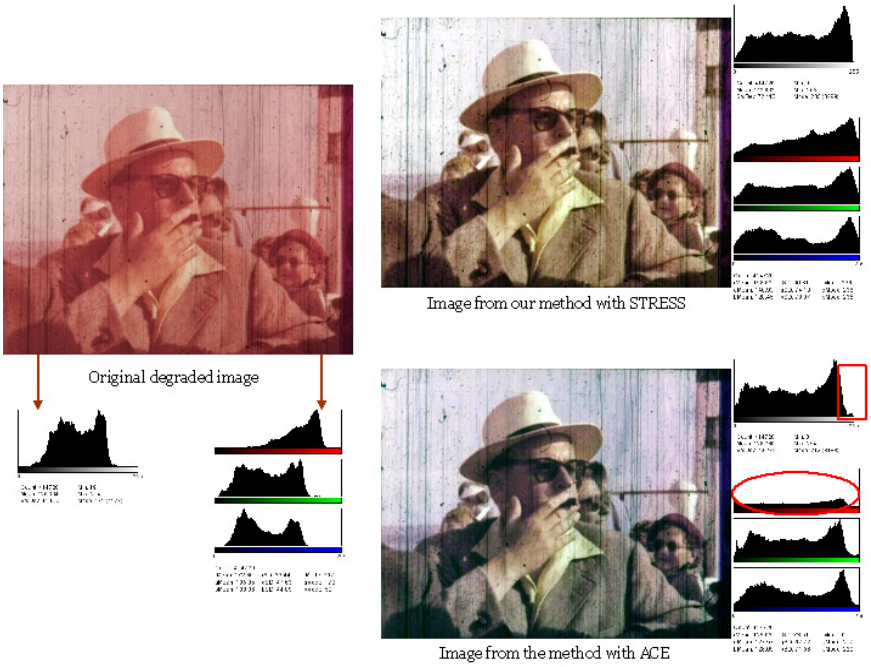

Figure 28: Resulting images-2 from the method with ACE and with STRESS.

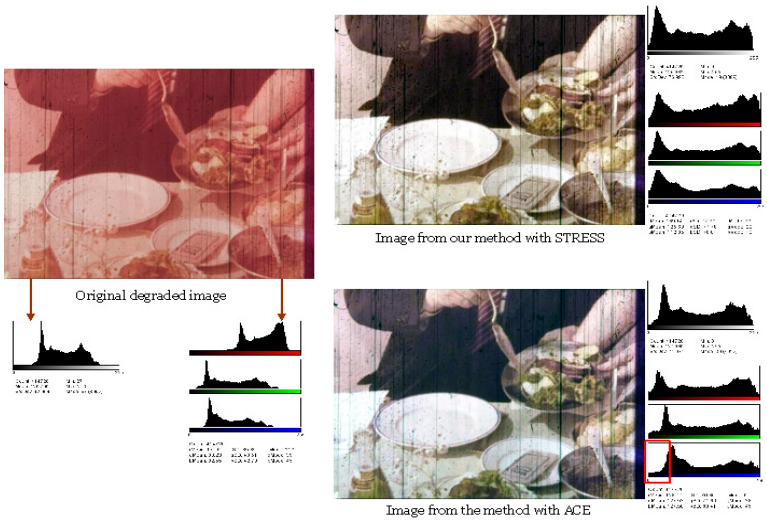

Figure 29: Resulting images-3 from the method with ACE and with STRESS.

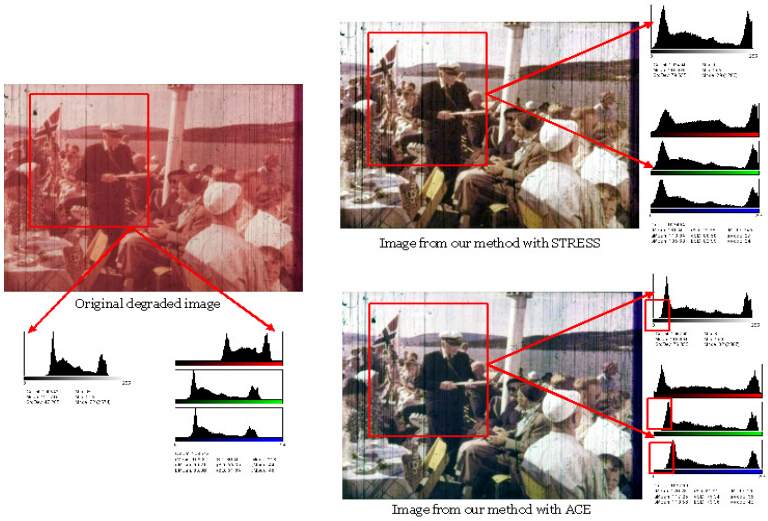

Figure 30: Resulting images-4 from the method with ACE and with STRESS.

a PC with the following configuration:

- `Processor`: Intel(R) Core(TM)2 Duo,
- `Processor speed`: 2.66 GHz,
- `RAM`: 4 GB

We measured the total time taken for restoring the colour of a degraded image by our method and we also implemented the ACE algorithm in place of STRESS and measured the required time taken for processing image. We used a degraded image with dimension of 720×576. The statistics of the time taken by both the methods with STRESS and ACE are represented in Figure 31.

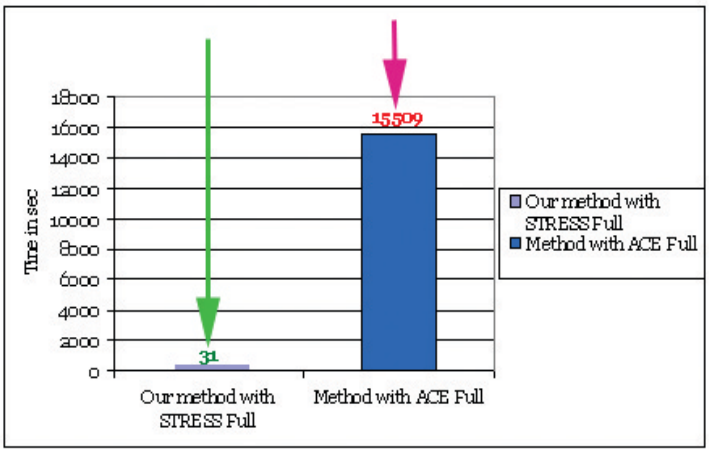

Figure 31: Computational time of the method with ACE and the method with STRESS.

We can have a clear idea about the difference between the required processing time with our method and the method with ACE. From Figure 31, we can see that, for the degraded image, our method outperforms the method with ACE.

We have also tested the method with ACE LLL(Local Linear LUT) method(appendix B) with a subsampling factor of 2. This LLL technique is applied to a small(sub-sampled) version of the original input image and then recomputed onto the full size input image using the Local Linear LUT method. The sub-sampling factor sets the resizing of the sub-sampled image; every axes of the input image is divided by sub-sampling factor. That means, with sub-sampling factor=2, every axes is subdivided 2 times and thus the sub-sampled image is 4 times smaller than the original input image. This option is faster than processing with the full image. The statistics of the method with ACE-LLL and our method with full version of STRESS is depicted below in Figure 32.

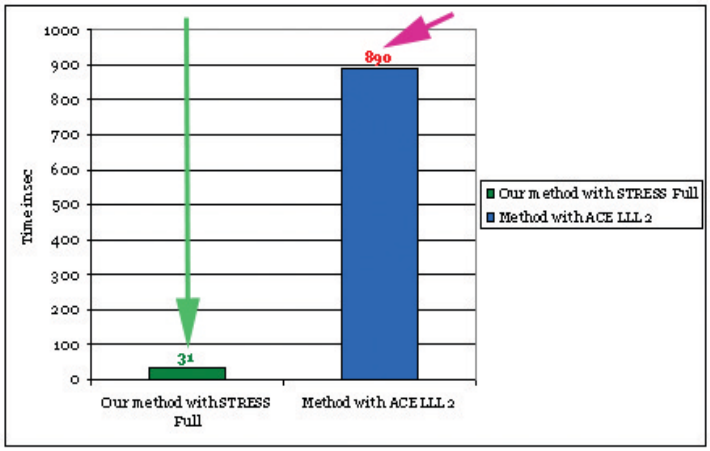

Figure 32: Computational time of ACE-LLL2 and full version STRESS.

Here, in this scenario in Figure 32, in case of ACE-LLL with subsampling factor of 2 as well, our method with full version STRESS outperforms the method with ACE-LLL. LLL method is a well documented method and it will certainly reduce the computational time of STRESS, if it is implemented on it. Figure 33 shows the computational time difference between our method with full version STRESS and ACE with LLL of subsampling factor of 4.

If the subsampling factor goes beyond 4, it produces images with artifacts. So, analyzing the results from Figure 31, Figure 32 and Figure 33, we can conclude that the method with full version STRESS outperforms the method with ACE full version, ACE-LLL2 and ACE-LLL4.

Now, we have presented the comparison on a smaller image. This original image has been resized so that the image becomes 30% of the original image. So, now the image dimension is 216×173. The comparison result is presented in Figure 34.

From Figure 34, we can observe that on the smaller image, our method with full version STRESS outperforms the method with full version ACE.

So, we can conclude that, irrespective of the picture size, our method with STRESS outperforms the method with ACE; even though LLL method is applied to ACE for speeding up the process, our method with full version STRESS still outperforms the method with ACE-LLL.

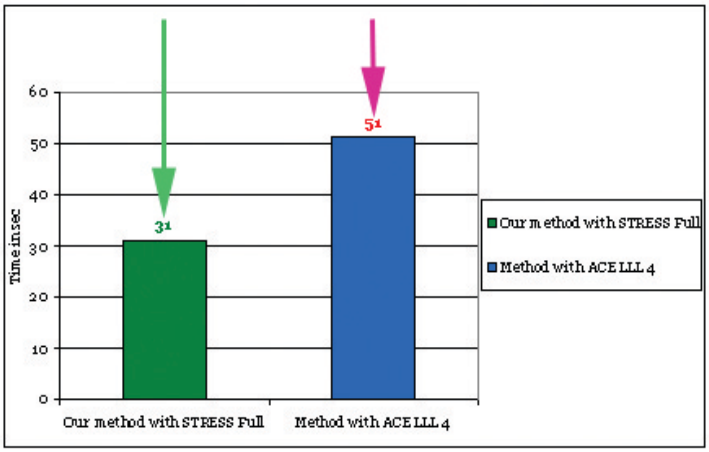

Figure 33: Computational time of ACE-LLL4 and full version STRESS.

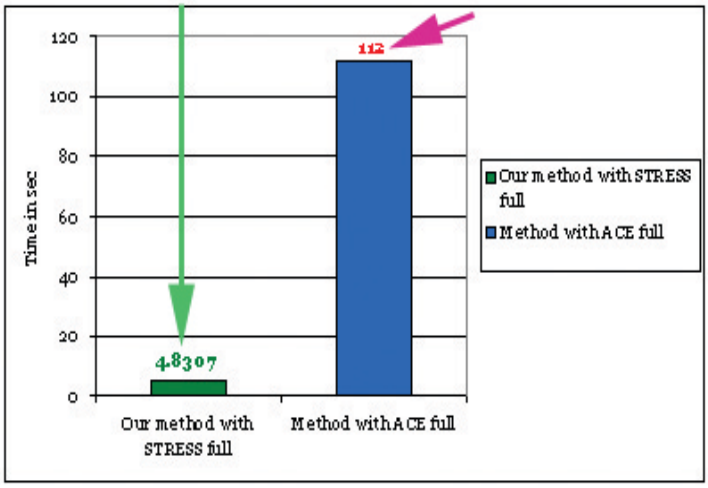

Figure 34: Computational time of ACE full and full version STRESS on smaller image.

4.2.4 The outcome of psychophysical experiment

For the analysis of our results from the psychophysical experiment, we used the method of statistical analysis. In order to prove any hypothesis in statistics, anyone have to define a `NULL` hypothesis and a signifinace level first. Then, the process of the analysis should be pursued in such a way that the `NULL` hypothesis could be rejected and eventually, the opposite of the `NULL` hypothesis i.e. the `alternative` hypothesis is accepted.

So, we define the NULL Hypothesis and the significance level of our experiment like below:

`NULL Hypothesis`= The observer finds both the images from ACE and STRESS equal and they can select any one of them with equal probability.
and
Significance Level= 0.05 i.e. 5%.

These defined parameters eventually mean that if the significance level i.e. tolerance level goes beyond 5% or becomes equal to 5%, we will not reject the `NULL Hypothesis` otherwise we will reject the `NULL Hypothesis` and eventually prove that such an incidence can occur by chance.

We got the experimental result which is depicted in Figure 35.

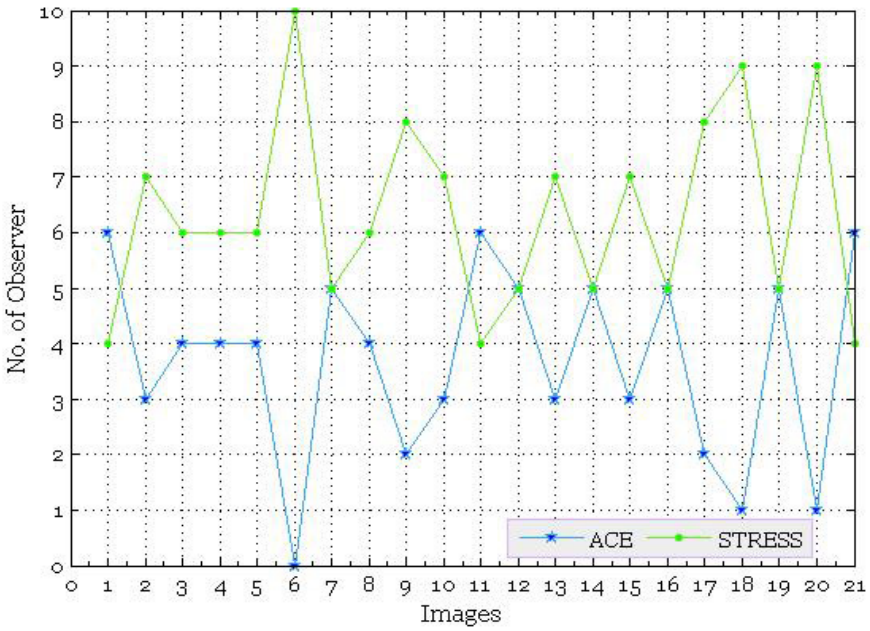

Figure 35: Statistical analysis of the psychophysical experiment for ACE vs STRESS.

In our experiment, we got 21 P values for 21 Images and 10 observers for each images. So, we have 210 observations. We found that for the first Image, P value = 0.7539. Since, P is above 0.05 for the first image, we cannot reject the `NULL Hypothesis`. It is to be noted here that, this is a double sided calculation of *binomial distribution*. We have calculated the P value according to the formula, Eq. 4.5:

$$P = 2 \times CDF(x, n, p) \tag{4.5}$$

Here p=0.5, n=10 and x=the outcome in each trial. CDF represents the Cumulative Distribution Function of *binomial distribution*.

Similarly, for the second image, we get $P = 0.3438$. Here also, P is above 0.05; so, we cannot reject the NULL Hypotheses.

The following Table 3 shows the results from the experiment with different images and their corresponding P values.

Table 3: Experiment results for different images and their corresponding P values.

Image no.	P value
Image 1	0.7539
Image 2	0.3438
Image 3	0.7539
Image 4	0.7539
Image 5	0.7539
Image 6	0.0020
Image 7	1.2461
Image 8	0.7539
Image 9	0.1094
Image 10	0.3438
Image 11	0.7539
Image 12	1.2461
Image 13	0.3438
Image 14	1.2461
Image 15	0.3438
Image 16	1.2461
Image 17	0.1094
Image 18	0.0215
Image 19	1.2461
Image 20	0.0215
Image 21	0.7539

So, from Table 3, we can see that 3 images have P values less than the significance level 0.05.

If we calculate the P value for all the images and for all the observers, we get the P value = 0.001356 which is less than the significance level 0.05. Here, x=77, n=210 and p=0.5.

So, we can reject the `NULL Hypothesis` and can conclude that the outcome we have observed is improbable due to chance.

4.3 Discussion

In this section, we have discussed about the effect of applying the STRESS algorithm and postprocessing technique on the preprocessed image. Then, we have discussed about the comparison of the result between our method with STRESS and the method with ACE. After that, we discussed about the computational time taken by the method with STRESS and the method with ACE. Lastly, we have discussed about the outcome of the psychophysical experiment to find out which image, from the method with STRESS or ACE, the observers liked most.

Firstly, we observed that, by applying the postprocessing technique, the image quality

is improved from the prespective of the HVS. The lightness level and the saturation level is preserved and the overall image gets more natural and pleasant outlook.

While comparing the result of the method with ACE and the method with STRESS, we found out that, the method with STRESS mainatins the shape of the histogram and enhance the dynamic range. But, in case of the method with ACE, the dynamic range is increased but the shape of the histogram gets distorted; sometimes some portions of some colour channels gets cut off. So, from this point of view, the method with STRESS performs better than the method with ACE in maintaining the shape of histogram while enhancing the dynamic range.

We performed a complete bias free psychophysical experiment to find out which image, from the method with STRESS or the method with ACE, the general people usually prefer. We found out that, most of the participants in the experiment preferred the image from the method with STRESS over the image from the method with ACE. Of course, for some images, the image from the method with ACE was preferred over the image from the method with STRESS. But, after calculatiing all the observation results for all the images and analyzing the results statistically, we found out that the images from the method with STRESS was preferred more than the images from the method with ACE.

Moreover, the method with STRESS outperforms the method with ACE in case of computational time. Both for the bigger image and smaller image, the method with STRESS took very less processing time than the method with ACE. Besides this, the method with full version STRESS outperforms the method with ACE-LLL as well in case of computational time.

So, after considering all the quality judgement factor, we can state that, out method with STRESS performs quite good in comparing with the existing methods like ACE: in case of enhancement quality, computational time and statistical result from psychophysical experiment.

5 Influence of Scratch Removal on Colour Restoration

While developing our method, we came across a well known degradation factor in old films, the scratch lines on the films. Scratch lines are vertical or almost vertical lines of the image with dark or bright intensity. These scratch lines vary in size; typically, the lines have the width from three to ten pixels or even sometimes more than ten pixels. Scratch lines might appear on the same position of subsequent frame sequences or sometimes, they appear just in random manner. Besides this, these lines might occupy a big portion of the degraded image or only a small part of the image.

The scratch lines degrade the visual quality of the movie frames. In our work, we have put our effort to remove the scratches by applying well known crack detection algorithm known as Top Hat crack detection [33] [34], binary morphology and inpainting method.

5.1 Method

For our work, we used the crack detection system, which is widely used in removing cracks in painting. Since, our principal goal is to enhance the colour information and scratch line removal is another direction of research, we decided to use one of the existing methods [35] for the scratch line detection and removal. For detecting scratches in the films, we used the general top hat crack detection from [35].

After identifying the scratches, we used morphological process like *dilation* to dilate the scratch line areas created by the scratches and then we have used Mean FMM Distance Transport Inpainting [35] process to remove the scratches.

We tested the scratch removal technique on the original degraded frames and after that we applied rest of our colour information enhancement technique and STRESS algorithm. Because, if we can remove the scratch before the colour enhancement and STRESS algorithm, the colour information from scratch lines will not be enhanced and the final outcome will be more pleasant looking. The work flow for the scratch line removal is depicted in Figure 36.

From Figure 36, we can see that the degraded image goes through a scratch line detection process. A sample degraded image with scratch line detected is depicted in Figure 37.

In Figure 37, the red lines in the bottom image are the scratch lines, which are detected by the Top Hat detection algorithm. Rectangular structuring element is used to detect the scratch lines on the film. After that, we applied a morphological operator called as *dilation*. This operator dilates the areas where the scratch lines are found.The result of dilation is depicted in Figure 38.

After the detected scratch lines are dilated, *inpainting* technique [35] is applied to fillup the dilated scratch line areas. The result of inpainting is like in Figure 39. From Figure 39, we can observe that most of the scratches have been removed from the degraded image.

From Figure 39, we can observe that still there are some scratches in the image. We could also remove this scratches by tuning the iteration parameter of the dilation process. If we increase the iteration of dilation, the scratch line area dilates more and the resulting

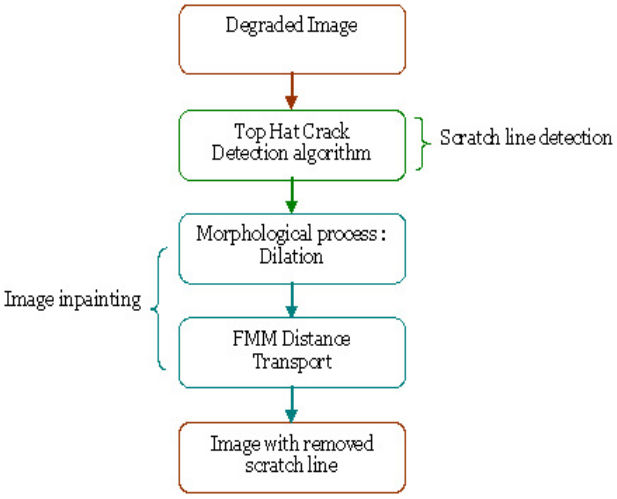

Figure 36: Work flow of scratch line detection and removal.

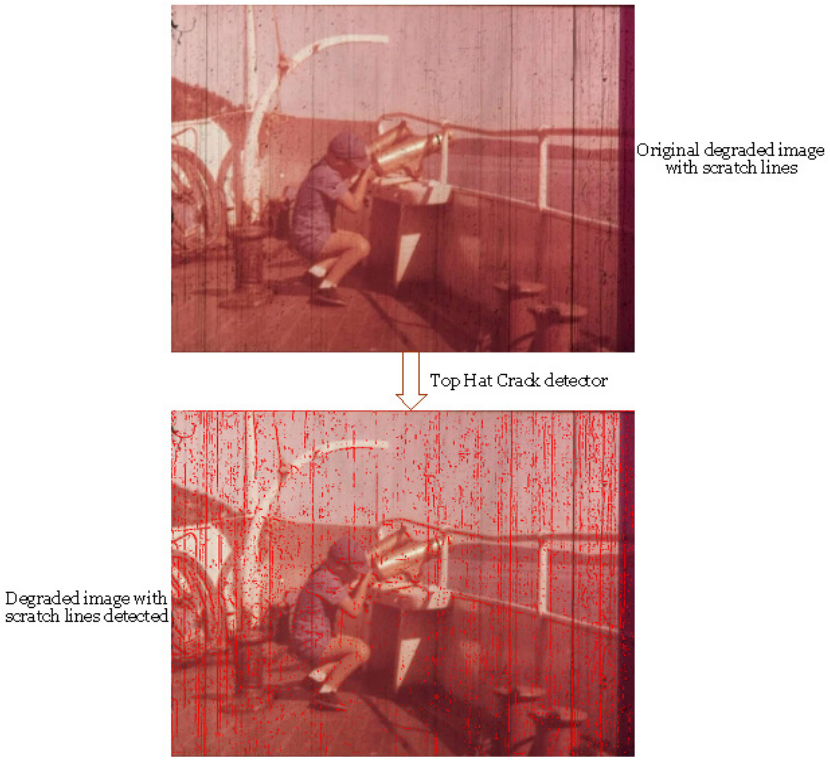

Figure 37: Degraded image with scratch lines detected in red colour.

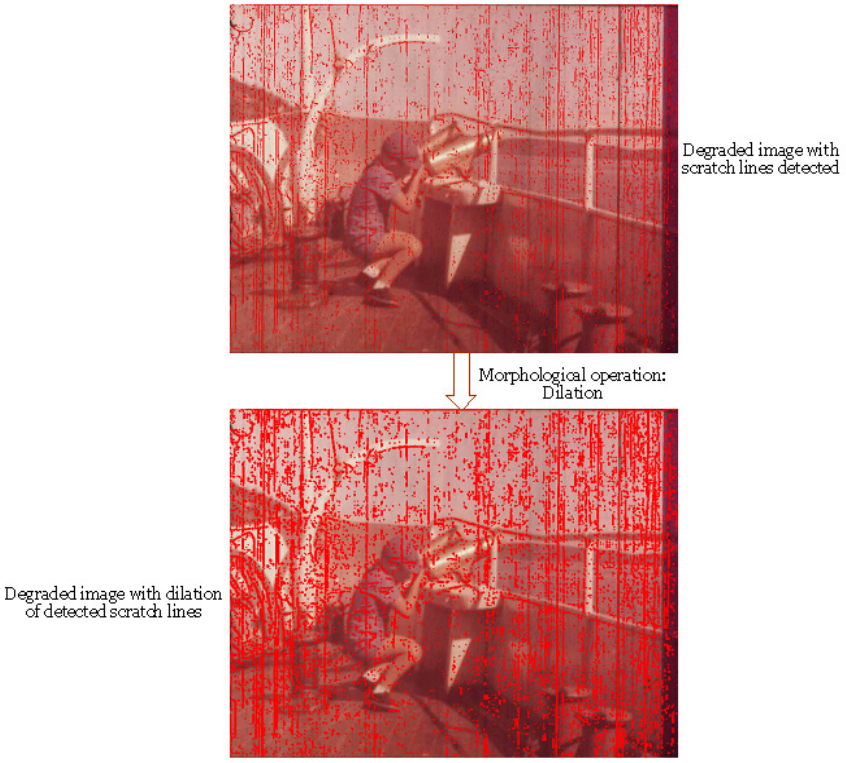

Figure 38: Degraded image with dilation of detected scratch lines in red colour.

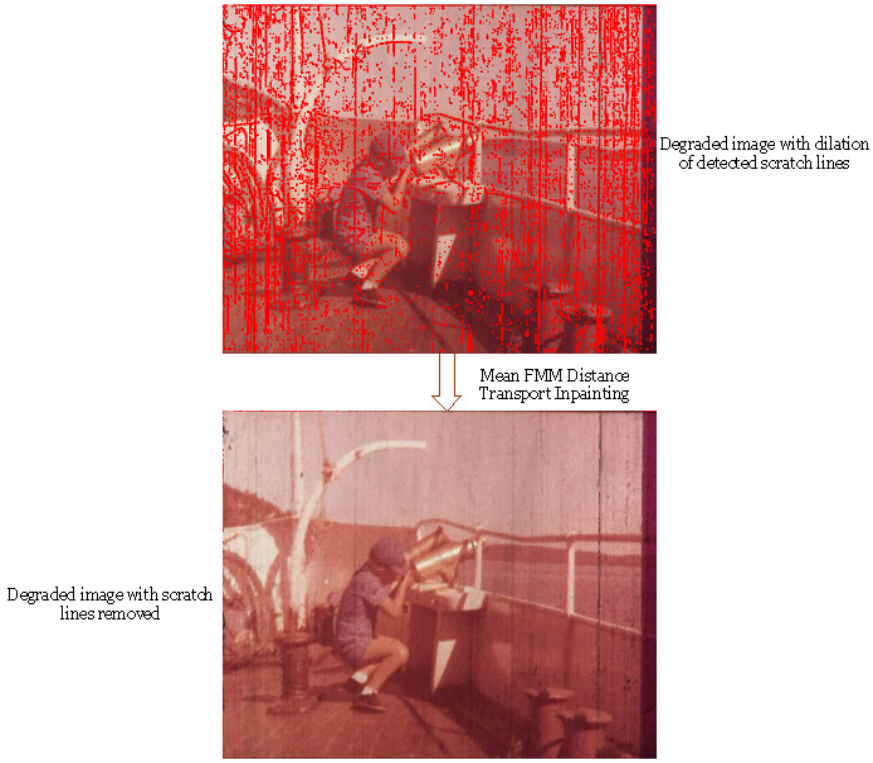

Figure 39: Inpainting result of the degraded image.

image from inpainting process gives result with some artifacts like blurring of some parts of the image or appearence of some sort of block areas in the image. The image with more number of iteration for dilation is depicted in Figure 40. The difference between two images in Figure 40 is clearly visible. The left image in Figure 40, has less number of dilation iteration; so the red area, which indicates the dilated area of detected scratch line, is less in the left image; But the red area is more in the right image.

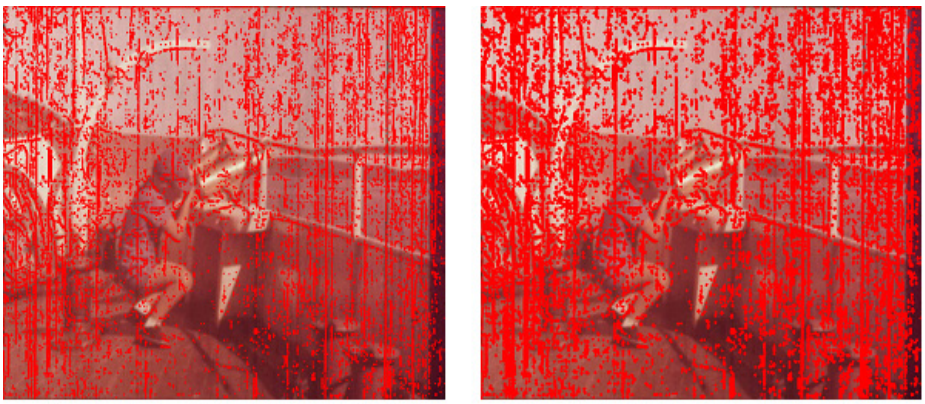

Figure 40: Scratch area dialted with less(left) and more(right) dilation.

The expected effect of using more number of dilation is, the less appearence of scratch lines on the resulting image from the inpainting process. But, it causes some sort of artifacts which are shown in Figure 41.

From Figure 41, it is clear that using more number of iteration helps to remove more scratches from the resulting image. But, it also creates some artifacts which are not desirable. In Figure 41, the area which is marked with red and green suffers from blurring effect. These areas loose their sharpness in the resulting image. In the red and green marked area, the top image is more sharper than the bottom one because of using less number of iteration in dilation process. The blue marked area in the bottom image in Figure 41 shows another artifact which is knows as blocking artifact.

The Figure 42 shows the total cycle of the scratch removal process with effect of each process on sample image.

5.2 Result

We have pointed out the effect of using scratch detection and removal methods in the previous section. We were able to remove most part of the scratch lines by using the methods described. Since, our concentration is on restoration of colour, we will discuss the effect of using these methods on the performance of colour restoration on the resulting image which is achieved by implementing our preprocessing, STRESS and postprocessing algorithm.

Figure 43 shows the effect of using and not using scratch line removal on colour restoration of degraded images. In Figure 43, the top image is the colour enhanced and restored image without scratch line removal method and the bottom one is the colour enhanced and restored image with scratch removal method. We can observe very less

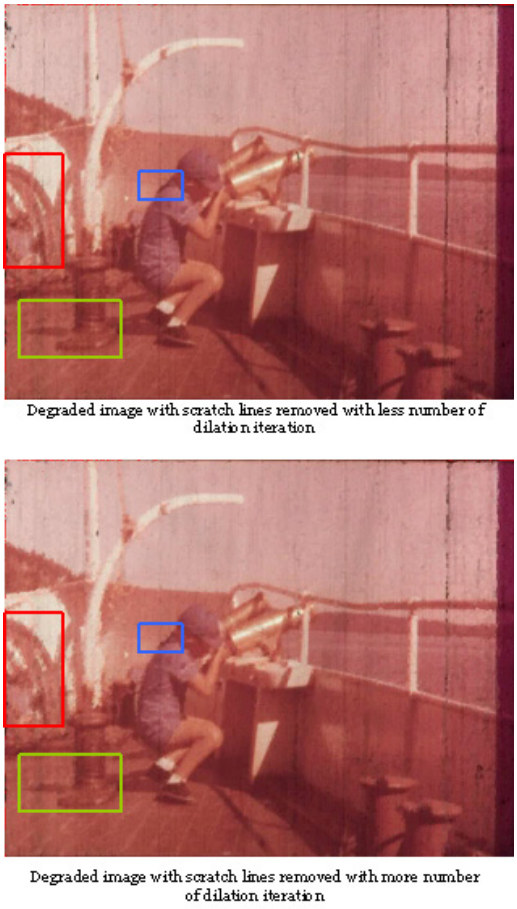

Figure 41: Effect of using less(top) and more(bottom) dilation on the inpainted image.

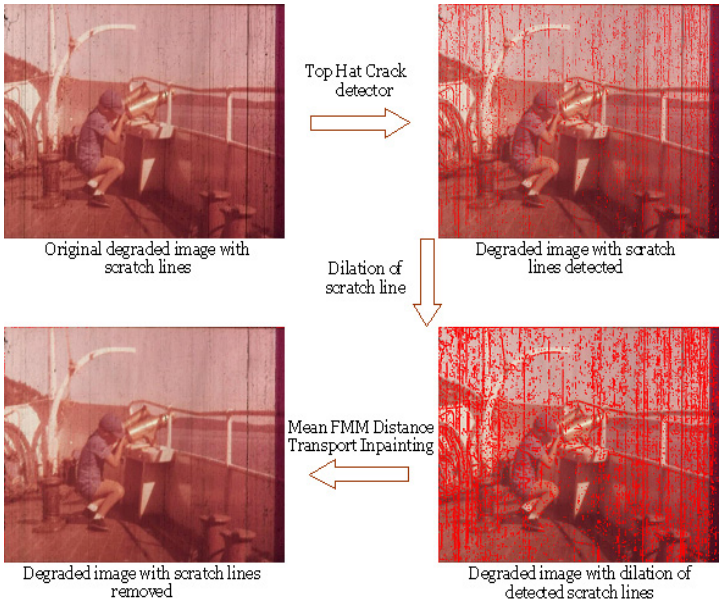

Figure 42: Scratch removal cycle.

difference between the intensity histogram of these two images. Besides this, by judging the colour histograms as well, we cannot perceive much difference between these two images. We can observe similar effect on Figure 44 as well.

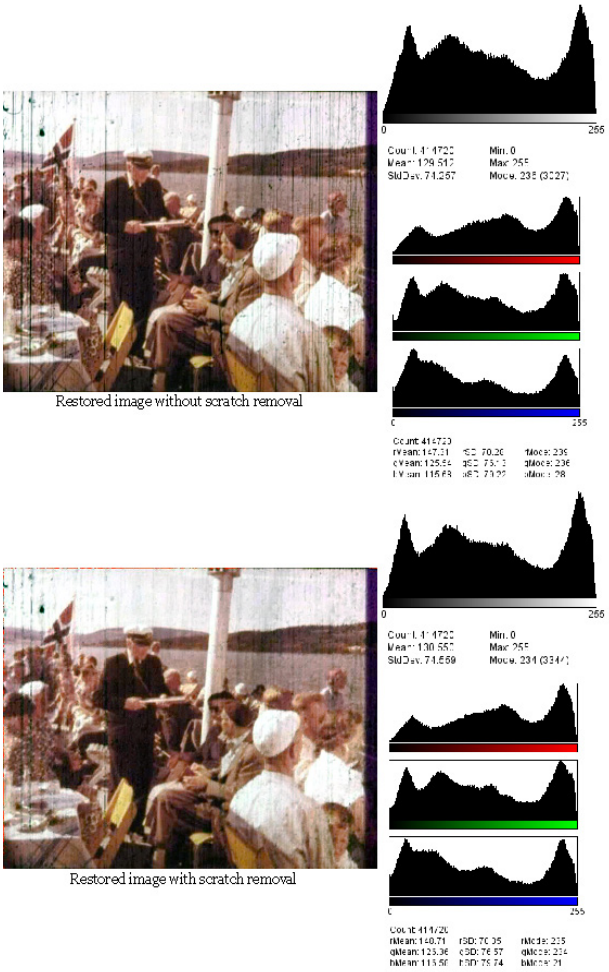

Figure 43: Effect of scratch line removal on colour restoration of degraded image-1.

We can have a more precise idea about the removal of scratch lines from the final output images by looking at the hot intensity heatmap in Figure 45 and Figure 46. The heatmap shows the variation of saturation intensity in the image by varying the colour from black through shades of red, orange, and yellow, to white. We can see that the intensity only varies in the scratch areas which are black. The other areas, which have no scratches, have similar intensity in both the figures.

From the intensity heatmap of Figure 45 and Figure 46, we can observe that in both the figures, the bottom image have less scratch than the top image. But, this does not affect the colour information restoration which is obvious from the colour histograms of Figure 43 and Figure 44.

The hsv heatmaps in Figure 47 and Figure 48, for these two images, also shows that the intensity only varies in the scartch area; it does not have any significant effect on the nonscratch areas. In hsv heatmap, hsv varies the hue component of the hue-saturation-

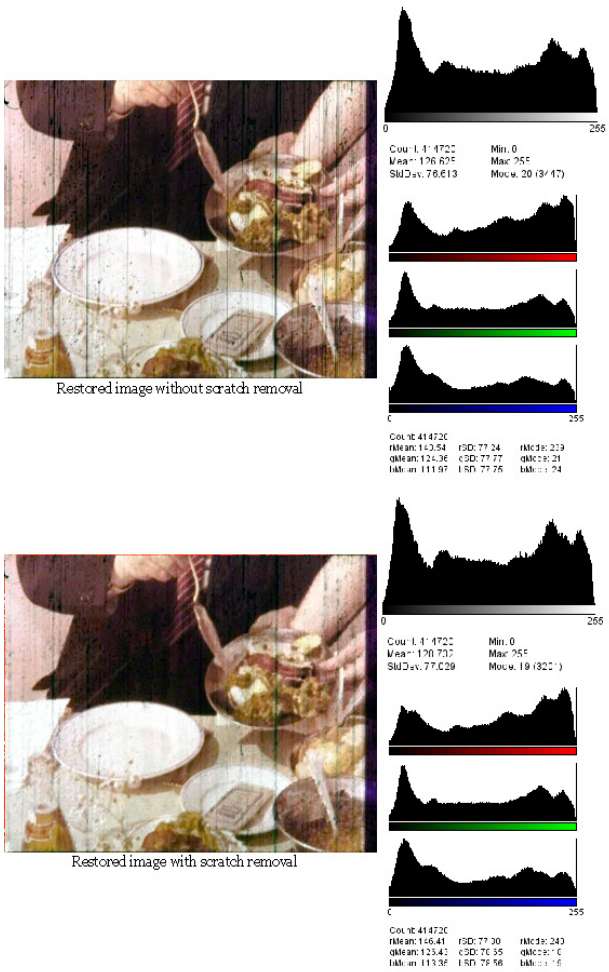

Figure 44: Effect of scratch line removal on colour restoration of degraded image-2.

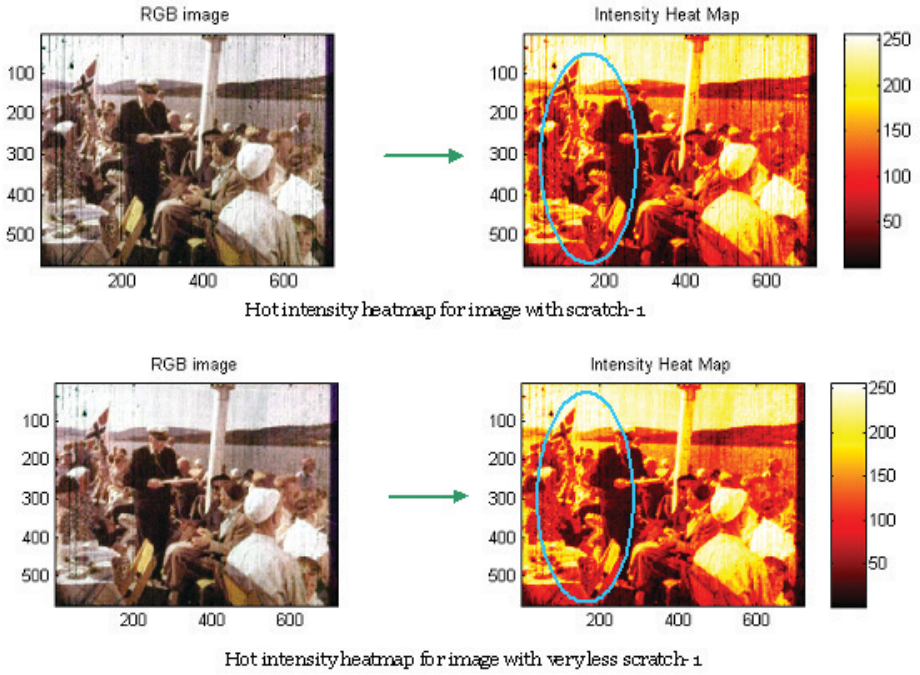

Figure 45: Hot intensity heatmap-1 of scratch line removal on colour restoration.

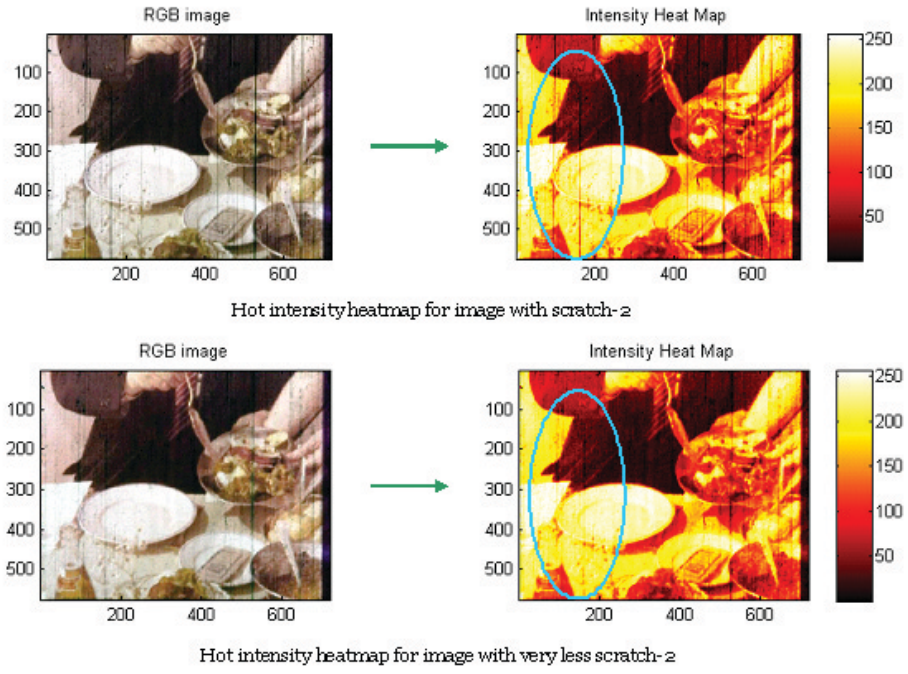

Figure 46: Hot intensity heatmap-2 of scratch line removal on colour restoration.

value color model. The colors begin with red, pass through yellow, green, cyan, blue, magenta, and return to red. So, we observe that the colour information of the total image is not affected much by the scratch removal.

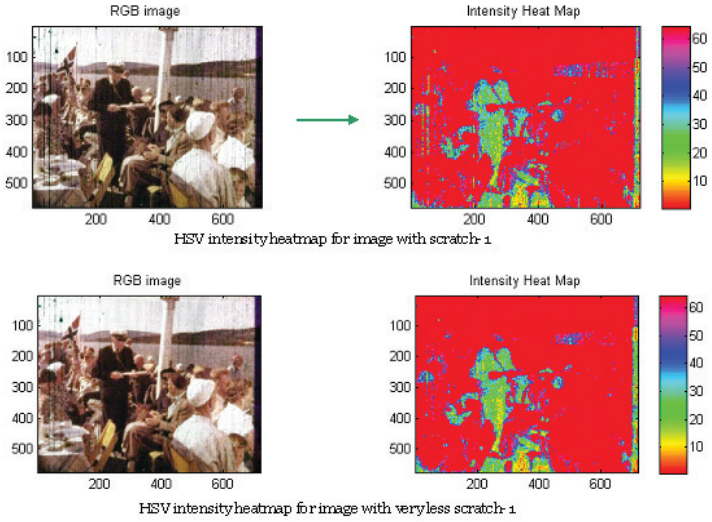

Figure 47: HSV intensity heatmap-1 of scratch line removal on colour restoration.

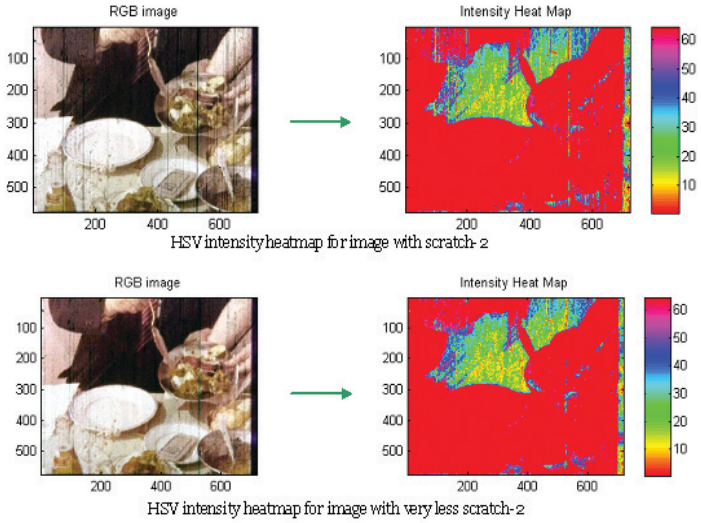

Figure 48: HSV intensity heatmap-2 of scratch line removal on colour restoration.

If we observe closely at some small area of the restored image with and without scratch line and compare the intensity histogram and colour histogram of that area in the image, we can comment more precisely about the effect of scratch removal on colour restoration. In Figure 49, a small area with degraded scratch line is pointed out by the red rectangle and the effect of applying scratch removal technique is closely observed. In Figure 49, the top image is the one generated without scratch removal method and the bottom one is generated with scratch removal method. We can see vey little difference between the histograms of the pointed area for both the images. Besides this, the co-

lour histogram also shows very insignificant difference between these two image in that marked area.

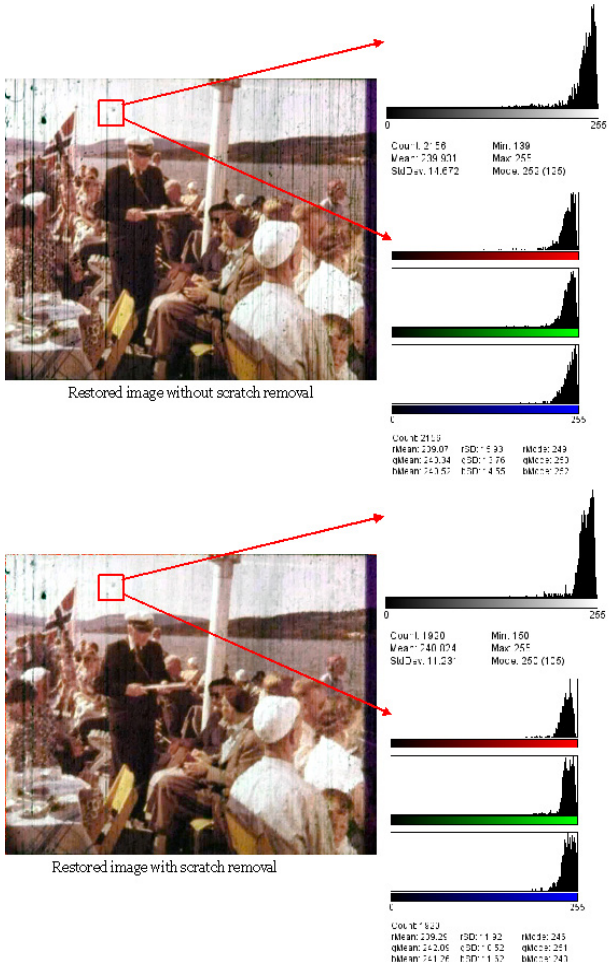

Figure 49: Effect of scratch line removal on colour restoration of degraded image-3.

Now, we observed one area in the image which has very less scratch lines and watch the effect of the scratch removal method on colour restoration. In Figure 50, the red marked area is almost without any scratch. In this case as well, the top image is the one generated without scratch removal method and the bottom one is generated with scratch removal method. We observe very little and insignificant difference between the intensity histogram of the images in Figure 50. Moreover, the colour histogram also shows very insignificant difference between these two marked area.

5.3 Discussion

Scratch line removal is an important area of research in old movie restoration. It has a great impact on the perceived quality of the images from the view of Human Visual System(HVS). But, in our work, we put our focus on the colour restoration of the old movie films and we are much more concerned about the perceived quality from colour perspective of HVS. Since, we found out that, removing the scratch doesn't affect or has

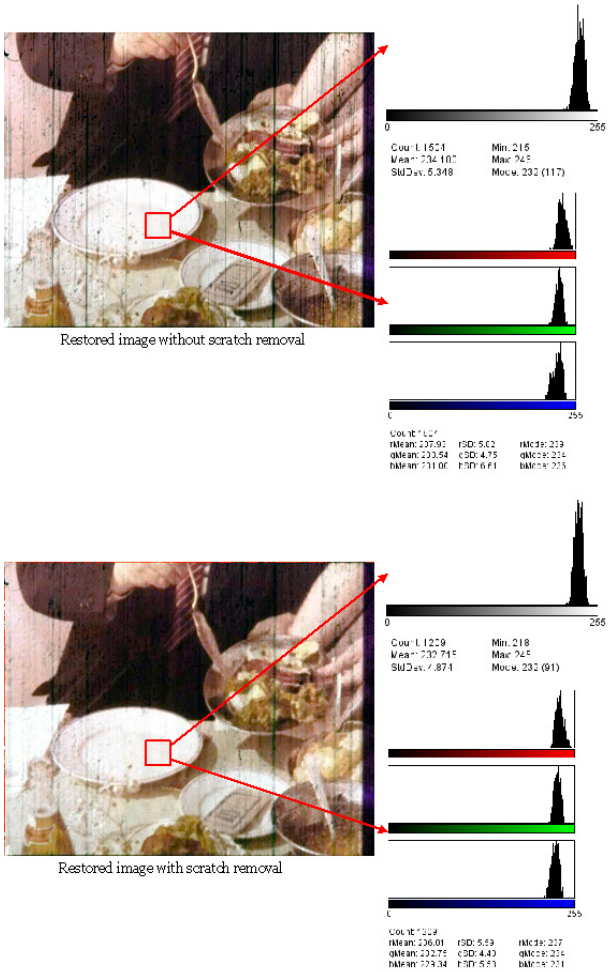

Figure 50: Effect of scratch line removal on colour restoration of degraded image-4.

a very insignificant effect on the performance of restoration of colour informarion, we can conclude that scratch removal doesn't play an important role in the colour correction phase for this particular scenario of test materials. Of course, it improves the visual perception of the frames. But, since, our main concentration is focused on the colour correction phase, we decided not to proceed with scratch removal technique. Rather we decided to proceed with the scratches on film which don't affect our method to a high level.

6 Temporal Domain STRESS Implementation and Postprocessing

Till this part of our work, we have been dealing with the spatial domain implementation of our method with STRESS and postprocessing. We were implementing our method on each frames and observing the reulting images and then we commented on the efficiency of our method. But, in a movie, we have huge number of frames. There are a number of shots, cuts etc in a movie. In each cut, there are number of frames. So, for correcting the colour of the whole movie, each of those frames should be considered. So, is there any efficient way to restore the colour of all the frames in minimum computational time - is a big issue in this area of research. We have suggested a temporal domain method which works on the frames in a cut. Our method performed quite well and was able to gain 80% reduction of computational time in the consecutive frames, comparing with the first frame in a cut. We have described our method in the next section. We have also presented the achieved resulting images and discussed about the achievement regarding the image quality and the computational time.

6.1 Method

In case of temporal domain implementation of our method, we have also used our preprocessing technique, described in Chapter 3, for enhancing the colour information in all the channels in a balanced way. After that, we implemented our new temporal STRESS model and our postprocessing method on each of the frames in a cut. But, while implementing our method on each frames, we followed a strategy for reducing the computational time. The workflow of our strategy is depicted in Figure 51.

In Figure 51, x is the weight factor. From Figure 51, we can observe that, we worked on degraded old film(topmost image). From this film, we worked on each cut(second image from top) in the movie. The cut might contain quite a number of frames and the background objects in a cut doesn't chnage very much. For the frames in a cut, we applied our preprocessing method(third image from top) for enhancing the colour information in all the three channels in a more balanced way. Then, we applied our temporal domain STRESS and postprocessing method on the frames of a cut. In [11], the final STRESS value, P_{stress} was calculated by using Eq. 2.10. Here, for temporal domain method, we stepped back from Eq. 2.10 and used the maximum and minimum envelopes for calculating the the final STRESS value, P_{stress}. So, P_{stress} is calculted by the formula in Eq. 6.1, a sample form of Eq. 2.9,

$$p_{stress} = \frac{p_0 - E_{min}}{E_{max} - E_{min}} \tag{6.1}$$

here, p_0, E_{min} and E_{max} are the current pixel and minimum and maximum envelopes respectively.

Now, for calculating the minimum and maximum envelopes for each pixels in each channels, we followed an efficient strategy to reduce the computational time at a very low level.

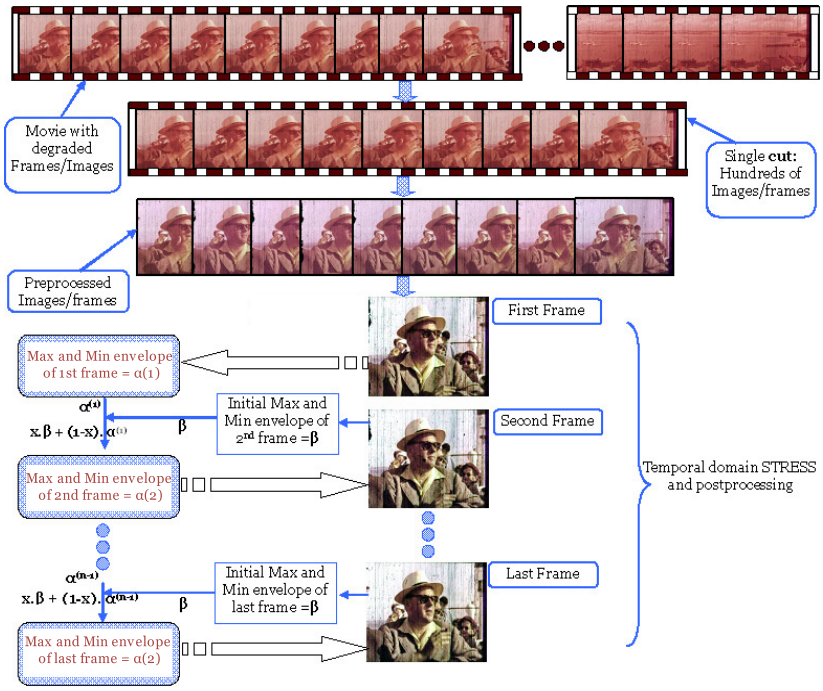

Figure 51: Workflow of the implementation of the temporal domain method.

For the first frame of the cut, we applied our method and calculated the minimum and maximum envelopes for each pixels in each channels and then we stored these envelopes in global variables for using them for the next frame. For the next frame, we firstly calculated it's own minimum and maximum envelopes for each pixels in each channels. Then, the minimum and maximum envelopes of the first frame are associated with the minimum and maximum envelopes of the current frame. We defined a weight factor for deciding how much information from the previous frame should be associated with the current frame. The envelope calculation and weight factor issue for the frames is depicted in the following Figure 52.

So, from Figure 52, we can see that the final minimum and maximum envelopes for the current frame are achieved by the summation of the weighted factor of the current frame and (1 - weighted factor) of the previous frame. Then, with these final envelopes, the resulting pixels, P_{stress}, for the current frame is calculated. These envelopes from current frame is stored in global variables for using for the next frame. The following set of Equations were used for calculating the envelopes and pixel value, P_{stress}.

For the first frame,

$$E_{min} = p_0 - \bar{v}\bar{r} \tag{6.2}$$

$$E_{max} = p_0 + (1 - \bar{v})\bar{r} = E_{min} + \bar{r} \tag{6.3}$$

$$p_{stress} = \frac{p_0 - E_{min}}{E_{max} - E_{min}} \tag{6.4}$$

here, p_0 is the current pixel and $\bar{v}$ and $\bar{r}$ are same as in Eq. 2.12 and Eq. 2.11 respectively.

Then, these envelopes, E_{min} and E_{max}, are stored in global variables,

$$GE_{min} = E_{min} \tag{6.5}$$

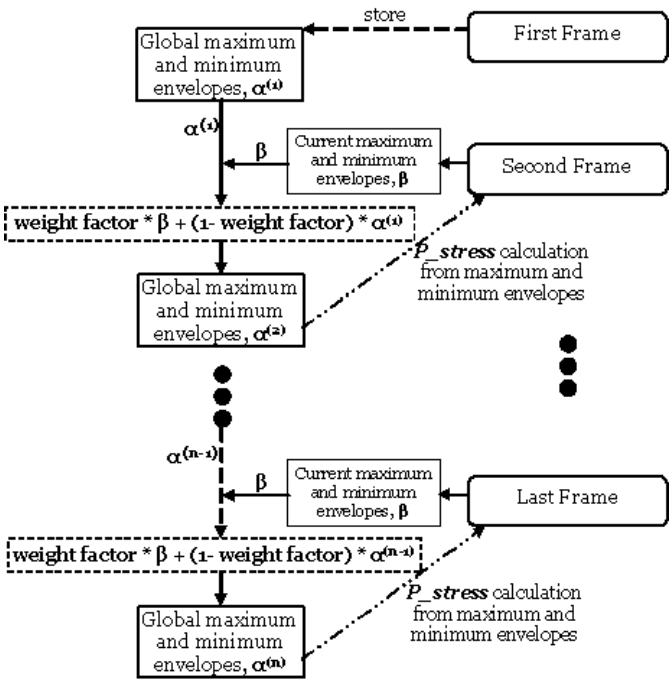

Figure 52: Min. and Max. envelope calculation for different frames and weight factor.

$$GE_{max} = E_{max} \tag{6.6}$$

here, GE_{min} and GE_{max} are the global variables for storing the envelopes.

Now, for the second frame, the envelopes are calculated by means of following equation,

$$E_{min} = \alpha \times E'_{min} + (1 - \alpha) GE_{min} \tag{6.7}$$

$$E_{max} = \alpha \times E'_{max} + (1 - \alpha) GE_{max} \tag{6.8}$$

$$p_{stress} = \frac{p_0 - E_{min}}{E_{max} - E_{min}} \tag{6.9}$$

here, α is the weight factor; E'_{min} and E'_{min} are the current initial envelopes of the current frame.

From Eq. 6.7 and Eq. 6.8, we can see that, for calculating the final envelopes of the current frame, the initial own envelopes are multiplied with the weight factor, α and then it is added to the envelopes of the previous frame, GE_{min} and GE_{max}, with a multiplicative coefficient of $(1 - \alpha)$. After calculating the P_{stress} value for each pixel in each channel, we store the envelopes of the second frame for using them in third frame and so on.

Now, in this strategy, we save a lot of computational time. Because, we appply our method with a high number of iteration only for the first frame for getting a noise free resulting image. For the next frame, we run our method for very few iterations. Usually, with a very few iteration, the resulting image should be very noisy. But, since, we are associating the envelopes of the previous first frame, which is noise free, we get a better image similar to the resulting image of the first frame.

We know that the envelopes are considered as the local reference maximum and minimum points. So, while calculating the envelopes of a particular point, the most nearby parts of the image influence strongly on the calculation. Hence, when we associate the envelopes of the previous frame with the current frame, the current frame gets the features of the envelopes of the previous frame.

How much information of the previous frame will be associated is decided by the weight factor. So, for example, if we set the weight factor as 0.4, then 60% information of the envelopes of previous frame will be associated with 40% information of the of the envelopes of the current frame. Hence, as much as the weight on the current frame will be increased, we need more iteration, because we are using very few iteration for the current frame. Similarly, as much as we put weight on the envelopes from previous frame, we need less iteration; since the previous frame is noise free and these envelopes are associated with small portion(small weighted portion) of the envelopes from current frame.

For extreme case, for weight factor of 0, we use no information of envelope from the current frame and use all the information of envelope from the previous frame; hence the iteration needed for the current frame is $\equiv 0$. Similarly, for weight factor of 1, we use no information of envelope from the previous frame and use all the information of envelope from the current frame; hence for producing a noise free image, the iteration needed for the current frame is equivaelnt to the iterations needed for the first frame and so on. If we use weight factor of 0, then some effect like, afterimage or trailing effect, might apprear in the current frame. Since, in case of 0 weight factor, we use all the information of envelopes from the previous frame, for a rapid motion on the foreground image of the previous frame might appear as an afterimage or like a shadow trail in the current frame. So, for avoiding this kind of effect, we need to set the weight factor to a certain level. We have depicted the implementation result of our method and ths effect of using extreme weight factor of 0 in the next section.

6.2 Result

In this section we have depicted the effect of using temporal doimain STRESS and postprocessing method on the colour restoration of movie frames. We have discussed the formation of envelopes for first frame and the next consecutive frames, the resulting images from temporal STRESS and postprocessing and the after image effect in this section.

6.2.1 Formation of the envelopes

We know that the maximum and minimum envelopes are the two functions which contain the image signal. These envelopes are the local reference darker and lighter points for each pixel in each chromatic channels [11]. Now, we produce the first frame of a cut of a movie with a large number of iteration which makes a noise free and good quality output image of the first frame. The envelopes of the first frame are also noise free and they are of good quality. The envelopes of the first frame are presented in Figure 53.

These envelopes in Figure 53 are stored for using with the initial envelopes of the second frame. For producing the next consecutive frames, we use very less number of iteration, hence it takes very less amount of time as well. The initial envelopes produced by the second frame is noisy and is not of good quality, since it is produced by using less number of iterations. The initial envelopes of the second frame is depicted in Figure 54.

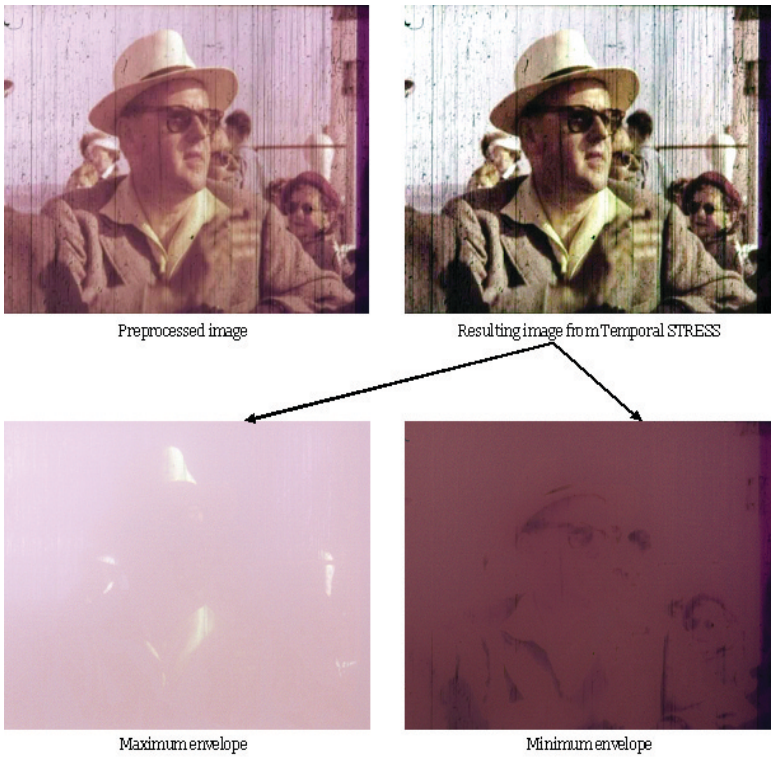

Figure 53: Max and Min envelope of the first frame.

The zoomed area of the image in Figure 54 shows the noise of the initial envelope.

Now, the stored envelopes of the first frame are added to the initial envelopes of the second frame by a specified weight factor. The addition of envelopes from first frame with the envelopes of second frame is depiected in Figure 55.

This makes the final envelopes of the second frame noise free and of good qulaity. The final envelopes of the second frame is presented in Figure 56. These envelopes of second frame is stored for using with third frame and so on.

The clear difference of the initial and final envelopes of the second frame can be observed from Figure 57. In this way, the resulting consecutive frames possess nice quality image as the first frame and a lot of computational time is saved.

6.2.2 Resulting images from temporal STRESS and postprocessing

This subsection describes various results of temporal STRESS and postprocessing. We have also showed the time required for different steps in colour restoration of degraded movie in this subsection. Besides this, we have shown the quality comparison of spatial and temporal STRESS ouput image by means of different intensity heatmaps. In Figure 58, we can observe the effect of this temporal domain method.

In Figure 58, the top left image is the first frame of the degraded film. The top middle image is the resulting image from spatial domain STRESS and postprocessing method. The top right image is the resulting image from temporal domain STRESS and postprocessing method. In this case, the output of the first frame is similar for both the spatial and temporal domain method, because in both methods same number of iteration is used for the first frame. The bottom left image is the second frame of the degraded

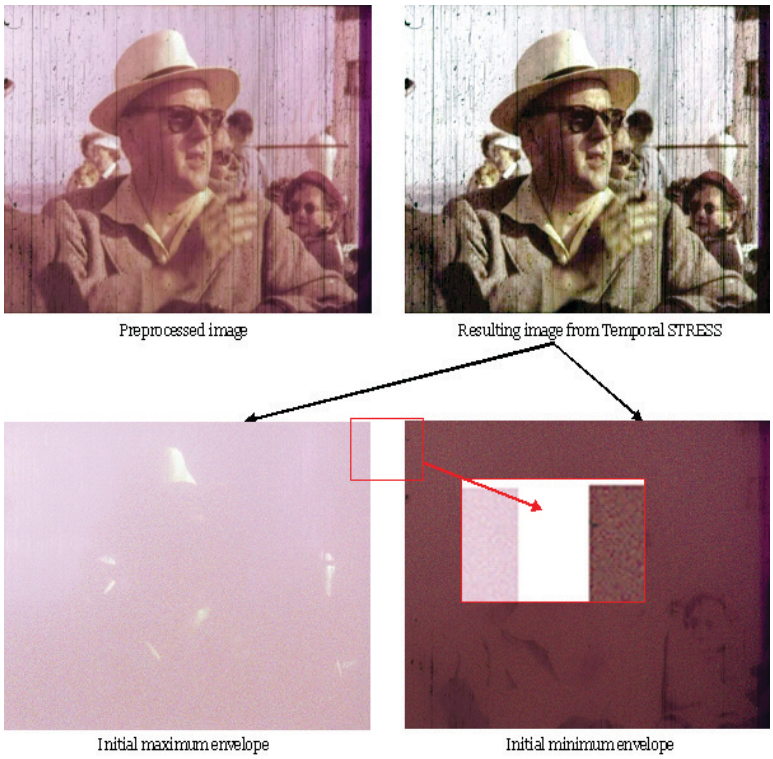

Figure 54: Initial Max and Min envelope of the second frame.

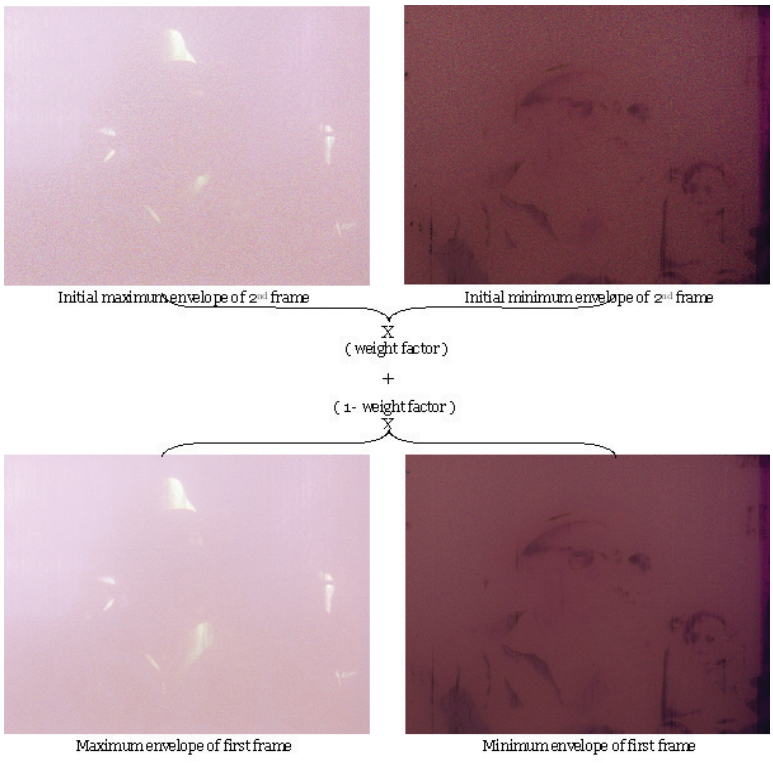

Figure 55: Addition of Max and Min envelope of the first and second frames.

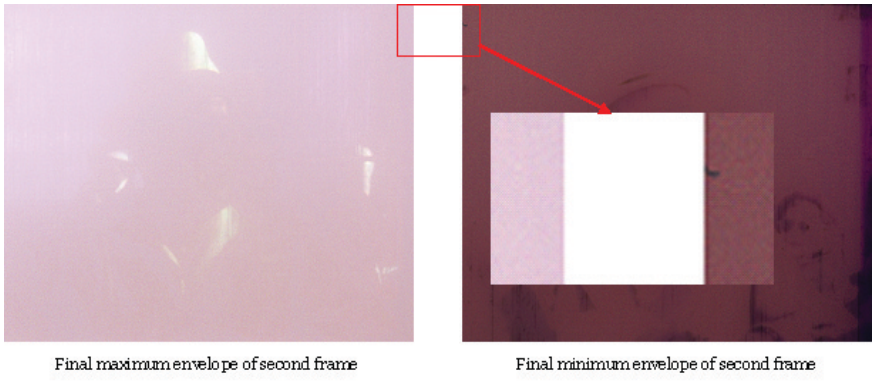

Figure 56: Final Max and Min envelope of the second frame.

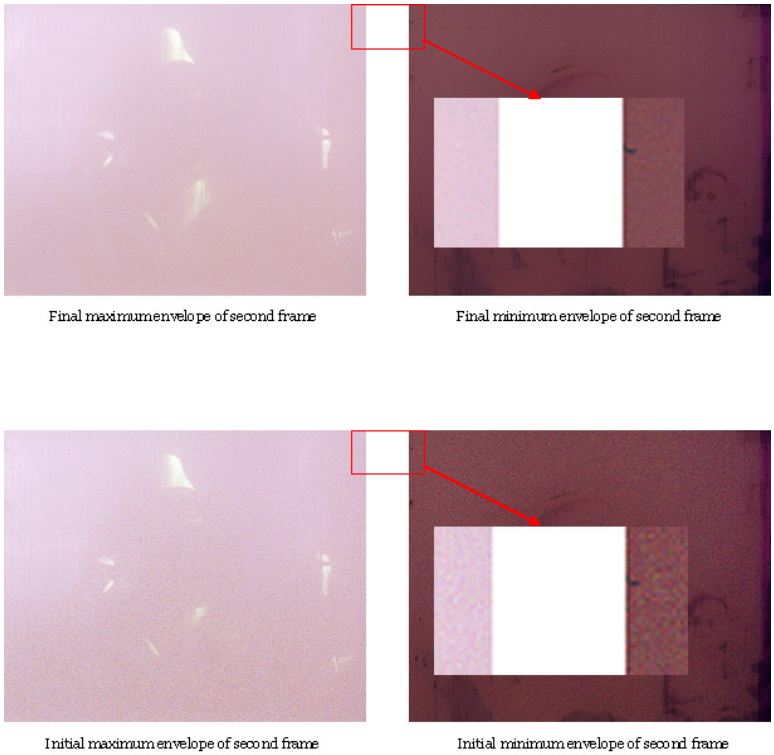

Figure 57: Difference of initial and final Max and Min envelope of the second frame.

Figure 58: Effect of temporal domain STRESS and postprocessing-1.

film. The bottom middle and bottom right images are the resulting images from spatial and temporal domain methods respectively. We cannot see any big perceived difference in image quality between the images from spatial and temoral STRESS for the second frame; though the temporal STRESS took very few iteration.

Now, in case of second frame of the degraded movie presented in Figure 58, if we set the weight factor=0.1, it means that the envelopes will be calculted by taking 10% information from the current second frame and rest 90% information from the first frame. So, we found out that the required number of iteration needed for second frame is only 8, while the required number of iteration for the first frame was 90. In the same way, for weight factor=0.3, the second frame only takes 10 iterations for producing a nice quality image as the first frame.

In case of this degraded movie, even with the low value of weight factor, like 0.1, we cannot find any after image effect in the resulting image. So, this means that, associating only 10% information from the current frame is enough and it eventually takes less time for producing the final image. So, if the second frame takes only 8 iterations while the first frame takes 90 iterations, the computational time is reduced to 91% for the second frame and also for the next consecutive frames.

For another degraded movie with more movement in the foreground, later presented in Figure 63, we found out that if we set the value of weight factor close to 0, like 0.1 or 0.2 or 0.3, we can see after image effect on the second and on the consecutive frames. We found out that, for weight factor=0.7, we don't see any such after image effect. So, if we set the weight factor as 0.7, it takes 15 iterations for producing a nice quality image without having the after image effect. So, the reduction of computational time is 83.33%.

So, after implementing temporal domain STRESS on both these degraded movies, we found out that, for frames with high contrast and much movement in the foreground, higher value of weight factor is needed and hence more iteration is required. For less contrast frames and less movement in the foreground, less weight factor and less iteration is needed and hence less time required. The required number of iteraions and time needed for first frame and the rest of the frames, for the later movie, are depiced in the

diagram in Figure 59.

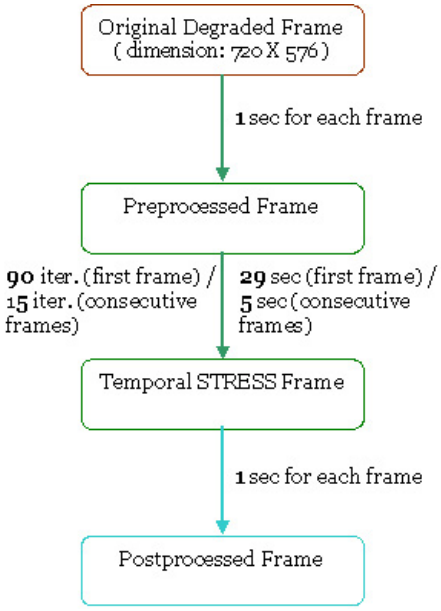

Figure 59: Required time for each step for temporal STRESS and postprocessing.

From Figure 59, we can see that, for the first frame it takes total 31 seconds(2 seconds for pre and post processing and 29 seconds for temporal STRESS), but for the later frame it takes 7 seconds in total. So, the reduction of time for the consecutive frames is more than 80 percent. In Figure 59, the frames are of 720×576 dimension. For another image, which is of 320×240 dimension, the process takes 25 seconds for the first frame and only 5 seconds for the consecutive frames. Again the reduction of computational time is more than 80 percent.

After processing by the temporal domain method we compared the two output from spatial and temporal domain methods for the second frame. We found out that the quality of the second frame is quite similar to the first frame, though the second frame only took 15 iterations i.e. the reduction of iteration was 83.33%. The image quality of the temporal domain method is shown with the comparison of the two methods on 2nd frame in Figure 60.

From Figure 60, by looking at the intensity histogram and from the perspective of the HVS, we found that two images are similar. So, our temporal domain methods works very effectively while reducing the computational time to a very significant low value.

We can also compare the image quality of the temporal domain by comparing the output with the spatial domain by considering the intensity of saturation. Figure 61 shows the intensity heatmap of the temporal output and spatial output. The heatmap shows the variation of saturation intensity in the image by varying the colour from black through shades of red, orange, and yellow, to white. We can see in Figure 61 that, both the images on the top and bottom have similar intensity heatmap. So, temporal output also preserves similar saturation intensity as the spatial output.

Besides this, we have also shown the two outputs from spatial and temporal STRESS by line intensity heat map. Line intensity heatmap shows the dark area of an image in different colour than the bright area. In Figure 62, we can see that blue part of the

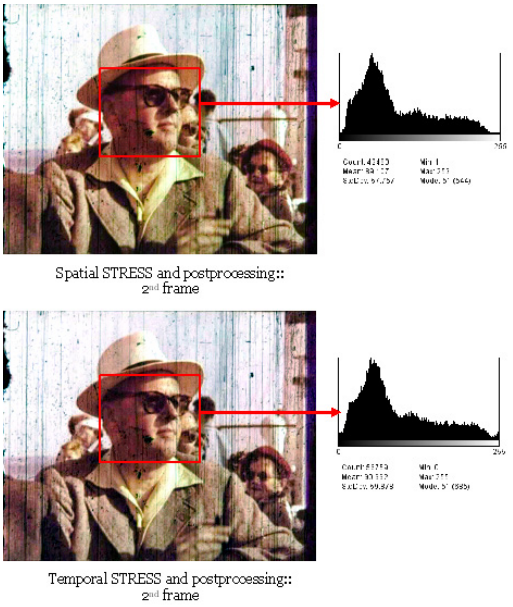

Figure 60: Spatial and temporal STRESS and postprocessing effect on the 2nd frame-1.

Figure 61: Hot intensity heatmap of spatial and temporal output-1.

intensity heat map image is the bright area and dark part is coloured with different colours. In Figure 62, we can see hardly any difference between the top and bottom intensity heatmaps, which again shows that the image quality from temporal STRESS is similar to the spatial STRESS; in which case, temporal STRESS takes very less time.

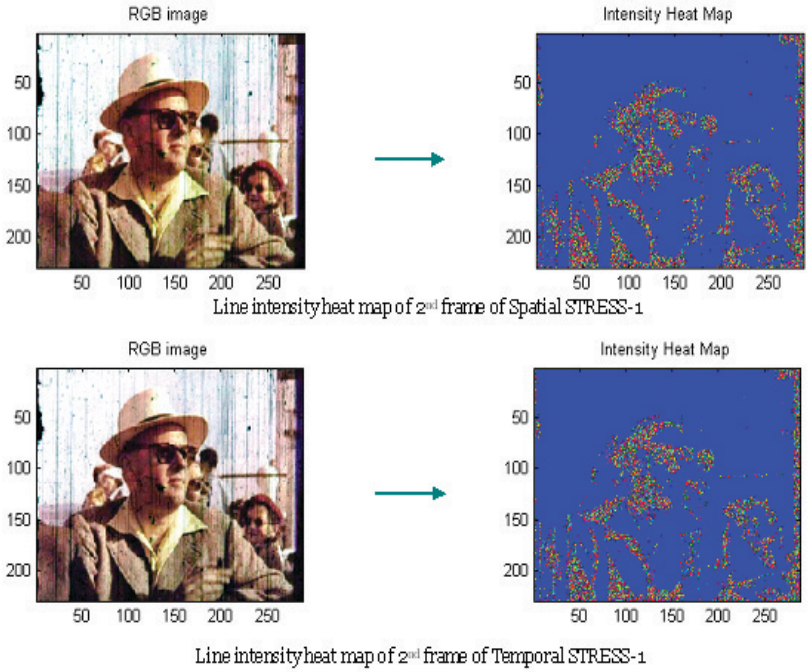

Figure 62: Line intensity heatmap of spatial and temporal output-1.

Another example of the effect of our temporal domain method is presented in Figure 63.

The comparison of the two output from spatial and temporal domain methods for the second frame of this movie is depicted in Figure 64. In this case as well, we found out that the quality of the second frame is fairly similar to the first frame, though computational time was reduced by more than 80% comapring to the computational time of the first frame.

The hot and line intensity heat map for this image is depicted in Figure 65 and Figure 66. From Figure 65 and Figure 66, we can clearly see that the temporal output image possesses similar saturation intensity as the spatial output image.

6.2.3 Removal of after image effect

While implementing the temporal domain method, we came accross a well known effect in the image, the after image effect. If we set the weight factor very less(close to 0), for example 0 or 0.1 or 0.2, then, we can see the after image effect on the current frame. In case of after image effect, some shadow part of the previous frame appears in the current frame. For exapmle, if a person moves his hand from one place to another, then, in the first frame the hand will be in one place and in the next frame, the hand will be in another position. So, if we process these two frames, then, the position of the hand in the first frame will appear at the same position in the second frame in a smoky way; a trail of the hand movement will be shown. This effect is depicted in Figure 67.

Figure 63: Effect of Temporal domain STRESS and postprocessing-2.

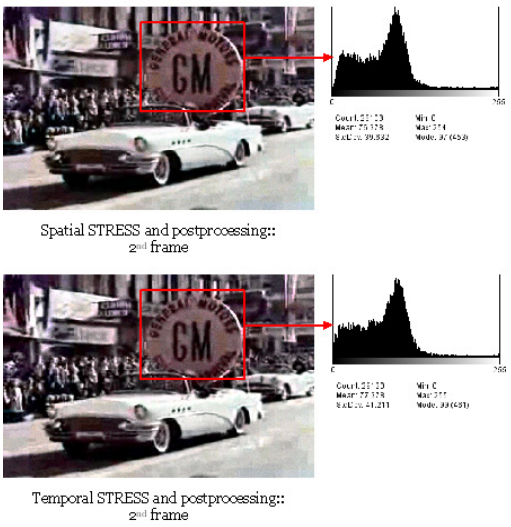

Figure 64: Spatial and temporal STRESS and postprocessing effect on the 2nd frame-2.

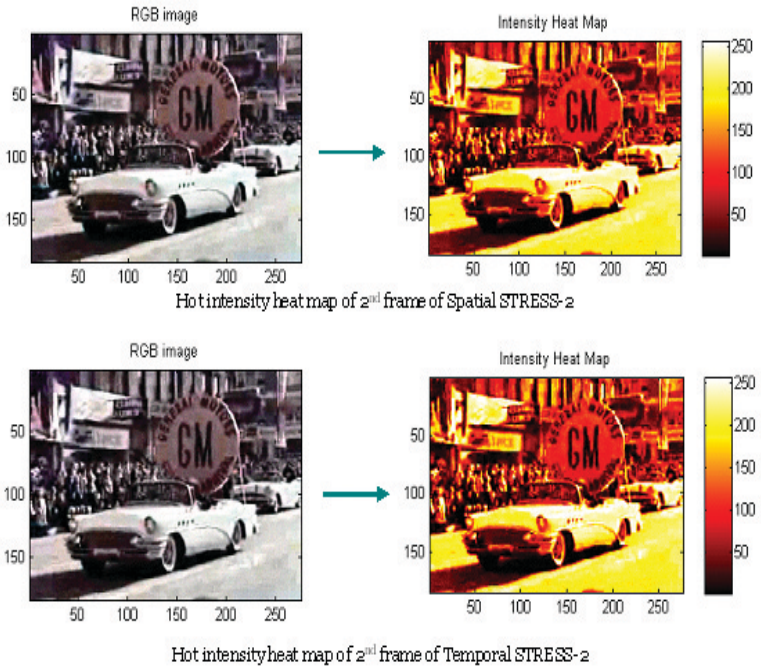

Figure 65: Hot intensity heatmap of spatial and temporal output-2.

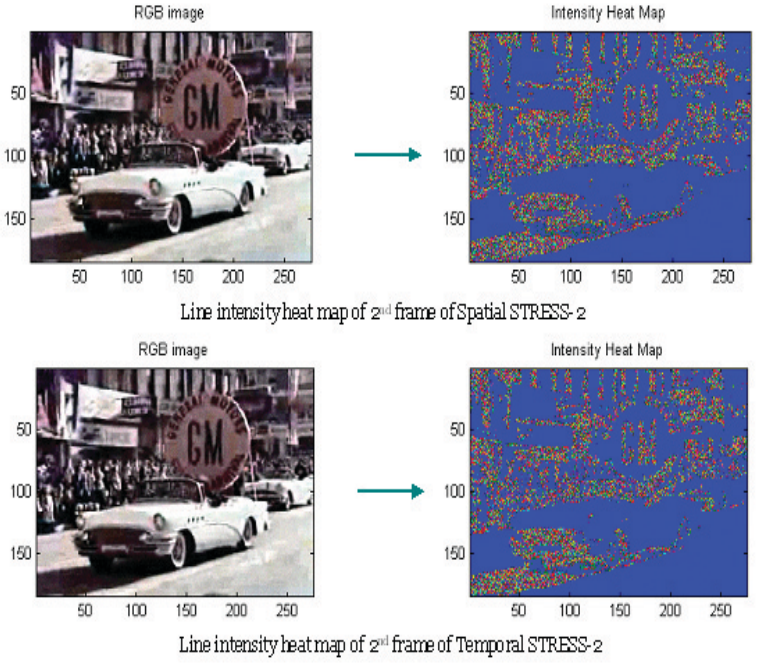

Figure 66: Line intensity heatmap of spatial and temporal output-2.

Figure 67: After image effect on video frame.

In Figure 67, we can observe that the position of the person has moved from left to right. So, the right hand of the person has also moved from left to right. We can see a shadow of the right sleeve(marked by red circle) of that person's coat in the second frame. The shadow is visible exactly at the same position where the sleeve was in the first frame. Besides this, we can also observe that the position of the circular object behind the person also moved from left to right. So, we can see a circular shape edge(marked by red rectangle) near the right end of the object. This circular edge is also exactly at the same position in the second frame as it was in the first frame. The reason behind this after image effect is the usage of less information of envelopes(weight factor is close to 0) from the current frame and more information of envelopes from the previous frame. If we increase the weight factor, for example weight factor is 0.7, then we do not see any after image effect on the current frame. Since, we have started to use more information from the current frame and less information from the previous frame, the effect of the envelopes of the previous frame is reduced and hence, the after image effect is also disappeared. We can observe the removal of after image effect by selecting the weight factor=0.7 in Figure 68.

We can observe in Figure 68, which contains the exact frames as in Figure 67, that, the after image effect is now removed from the second frame. So, we can comment that, by increasing the weight factor we can remove the after image effect in the video frames.

But, the issue here is that does this increase of weight factor also increases the computational time. The answer is yes, but, still the second frame's computational time is reduced by more than 80% comparing to the first frame. Because, if we have weight factor close to 0, then, only 1 or 2 iteration is enough for getting the similar quality of the image as the first frame. Since, we have increased the weight factor to 0.7, the required iteration for second frame is only 15(while for the first frame it is 90), hence the computational time for second frame is reduced by more than 80%.

The necessary number of iterations needed for producing a noise free image of the

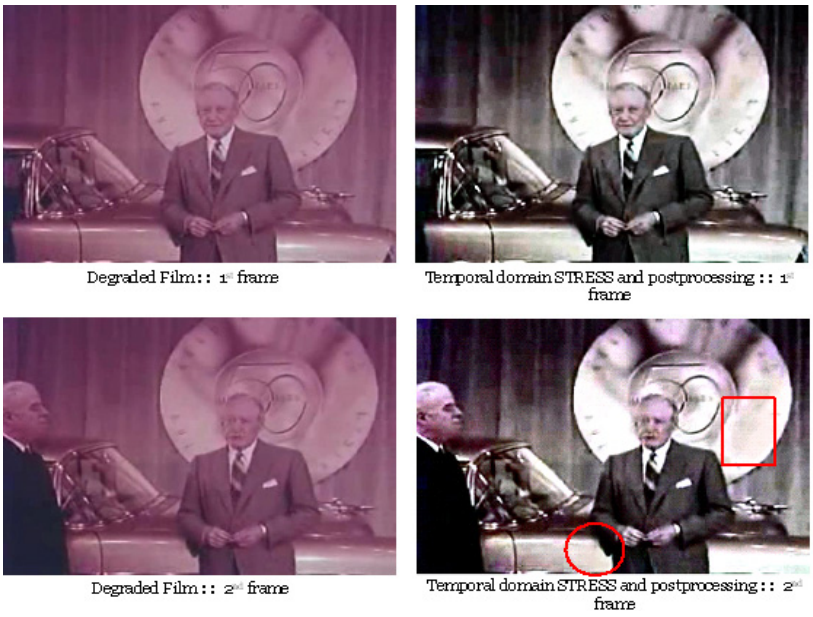

Figure 68: Removal of after image on video frame by increasing the weight factor.

current frame and having the similar image quality of the first frame depending on the different weight factors are presented in Table 4.

The relation between weight factor and required number of iteration is nonlinear, which is presented in Figure 69. This plot shows that we need more number of iterations as the weight factor is increased. The more number of iteration helps to reduce the noise in the current frame and make the image quality similar to that of first image. We can also observe that at weight factor 0.7, the required number of iteration is 15. At this point we can not see any after image effect, which we have shown in Figure 68.

The weight versus iteration plot in Figure 69 is achieved by considering the fact that, the resulting images for different weight factors were shown in the increasing order of weight factor. For example, we showed to the observers, the resulting images for the weight factor of 0.1 and then we showed the images for the weight factor of 0.2, 0.3 and so on. In this kind of situation, our assumption was that, if we increase the weight factor, then obviously the number of iteration would increase. Besides this, we also considered that for weight factor of 0, the required number of iteration is 0 and for weight factor of 1, the required number of iteration is 90; because, for weight factor of 0, all the information of the envelopes come from the previous frame which is noise free, so no iteration is required for getting the current frame. For weight factor of 1, all the information of the envelopes will be from the current frame, so more number of iteration is required and since the first frame took 90 iterations, the current frame also should take 90 iterations. Hence, we achieved the nonlinear relationship between the weight factor and the number of iterations required.

Now, we performed another test, totally blind test, by showing the images irrespective of the increasing order of the weight factor. For example, we showed the resulting images for weight factor of 0.4 and then we showed the images for the weight factor of 0.1, 0.5, 0.9, 0.8 and so on. In this case, the observer was not aware of the order of the weight factor. We found a correlation of 0.88 between the weight factor and the number of

Table 4: Different values of weight factor and number of required iterations.

Weight factor	Number of iterations
0.00	0
0.01	3
0.05	5
0.10	8
0.15	8
0.20	8
0.25	10
0.30	10
0.35	10
0.40	10
0.45	12
0.50	12
0.55	12
0.60	13
0.65	13
0.70	15
0.75	30
0.80	40
0.85	45
0.90	60
0.95	80
1.00	90

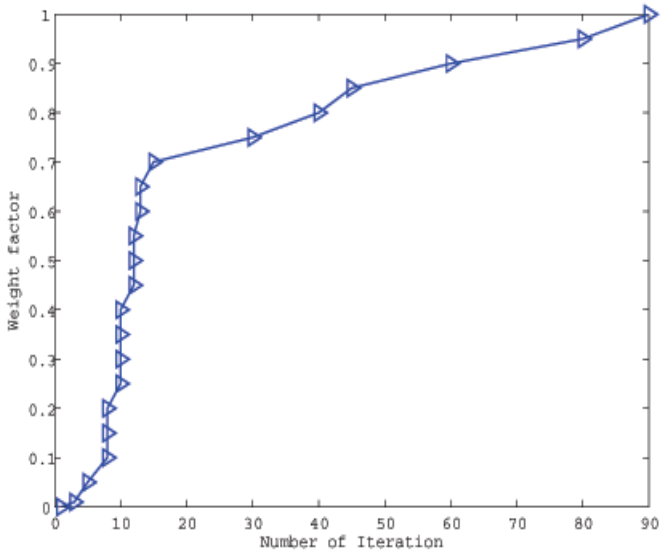

Figure 69: Relation between weight factor and required number of iterations-1.

iterations required. The necessary number of iterations required, for producing a noise free image of the current frame and having the similar image quality of the first frame depending on the different weight factors, for this blind test are presented in Table 5.

Table 5: Different values of weight factor and number of iterations:ideal case.

Weight factor	Number of iterations
0.00	0
0.10	9
0.20	9
0.30	18
0.40	18
0.50	9
0.60	27
0.70	27
0.80	18
0.90	36
1.00	45

Now, in this case as well, the number of iteration required, for producing a nice quality image, for next consecutive frames is 9 at weight factor=0.2 and for the first frame it is 45. So, the reduction of computational time is 80%. The plot for weight factor versus number of iterations required is depicted in Figure 70. It is worth to mention that, this is an ideal case which is not always achieved in any test. In case of psychophysical experiments, some sort of psychophysical bias of human mind always takes place.

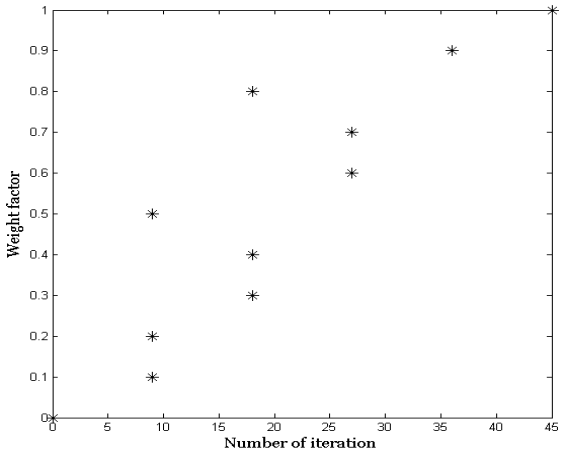

Figure 70: Relation between weight factor and required number of iterations-2.

Considering both the plot in Figure 69 and Figure 70, we can come to the decision that, the weight factor should be calculated by the estimated motion in the frames. For each frame, a map of weight factor should be calculated in which, the area in the image which have more motion should posses higher value of weight factor than the area with less motion. Then, considering this estimated motion information, the rest of the process should be performed. This would eliminate the necessity of psychophysical test and eventually make the process more faster. We have put this work of motion estimation as a future work.

6.3 Discussion

Analyzing the different results presented in the previous result section, we can come to the conclusion that our temporal domain method not only reduce the computational time for the consecutive frames, but it also maintains the similar perceived quality of the consecutive frames as well. We might come accross some effect like after image effct, but by increasing the weight factor, we can resolve this issue. Here, it is worth to mention that, increasing the weight factor to 0.7 gives us more than 80% reduction of computational time. We could have more reduction in computational time(as in the first movie), if there was not much movement in the foreground image in the cut of a movie so that the after image effect would not occur.

7 Conclusion and Future Perspective

7.1 Conclusion

We have proposed here an automatic colour correction technique which eventually automates the colour fading restoration process. The proposed method uses the STRESS model - an automatic image enhancement technique that deals with correcting colour images according to the Human Visual System. We tried to solve some basic problems that exist in colour restoration process. Firstly, we have proposed a preprocessing method which eventually helps to enhance the colour information in all the channels and thus the strong colour cast of the old movie is reduced to a very low level.

Besides this, we have offered a very effective postprocessing technique[1] for preserving the mean lightness and the saturation level of the preprocessed image after they are processed by the colour enhancing models like ACE or STRESS. We have also analyzed the effect of scratch line on old movies and implemented one existing technique to remove the scratch lines. We have analyzed the effect of removing the scratch line on the performance of colour restoration. We found out that removal of colour scratch does not affect the colour restoration much. Of course it improves the perceived image quality, but since our principal goal is to restore the colour of old movie; we decided to put the scratch line removal as an option.

Apart from this, the computational time is a major issue in processing the movie frames. There are large number of frames in a movie, so, processing time reduction affects the algorithm quality at a high level. We were able to significantly reduce(more than 80% reduction) the processing time for the consecutive frames in the cut of a movie. Besdies this, we have performed a psychophysical experiment to set the right parameter for the enhancement of colour information in preprocessing phase and we have also performed another psychophysical experiment for comparing the resulting image from our method and the method with ACE. The experiment was totally bias free and the result obtained was quite satisfactory. We found out that the images from our method are preferred by most of the observers in most of the images. Of course, for some test images, the resulting images from the method with ACE was preferred most, but the overall result for all the images was better for our method.

Finally, we can comment that our method produces resulting images which are pleasant from the perspective of the HVS, the processing time is very fast in comparing with the existing methods and the colour information in all the channels are well balanced after performing the method.

7.2 Future perspective

As a future plan, the following strategies might be explored for further efficiency and enhancement of this particular research area.

1. Implementation of Local Linear Lookup Table(LLL) method for STRESS. This will

[1] Accepted for presentation in the conference, EUVIP 2010, Paris, France.

certainly speedup the total processing time of our method; even though our method is faster than any other existing methods.

2. Visual saliency technique could be implemented to enhance the specific part of a frame rather than the whole image. This might speed up the total processing time even more; even though the current processing time is better than any other existing algorithms. In this strategy, the weight factor map of the whole image should be created and then, only associate the weight factor of those parts, which have higher importance regarding visual saliency.
3. Motion estimation and detection could be implemented to enhance the particular area where the change in motion occurs in a frame. This would certainly save a lot of computational time.

Bibliography

[1] Kennel, G. 1994. Digital film scanning and recording : The technology and practice. *SMPTE Journal (Society of Motion Picture and Television Engineers)*, 103(3), 174–181.

[2] CST. 1997. La restauration numérique des films cinématographiques. *CST (Commision Supérieure Technique de lImage et du Son) report*.

[3] Land, E. & McCann, J. 1971. Lightness and retinex theory. *Journal of Optical Society of America*, 61, 1–11.

[4] Barnard, K. & Funt, B. 1998. Investigations into multi-scale retinex. *In Proc. of Colour Imaging in Multimedia, Derby, UK*.

[5] Cowan, J. & Bressloff, P. 2002. Visual cortex and the retinex algorithm. *In Proc. SPIE Human Vision and Electronic Imaging, S. Jose, California, USA*, 4662(7).

[6] Huck, S. & Fales, C. 2000. Visual communication with retinex coding. *Applied Optics*, 39(11), 1711–1730.

[7] D.J. Jobson, Z. R. & Woodel, G. 1997. A multi-scale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transaction on Image Processing: Special Issue on Color Processing*, 6(7), 965–976.

[8] E. Provenzi, M. Fierro, A. R. L. D. C. D. G. & Marini, D. Random spray retinex: a new retinex implementation to investigate the local properties of the model. *IEEE Transactions on Image Processing*.

[9] Rizzi, A. & McCann, J. 2007. On the behavior of spatial models of color. *In Proc. of Electronic Imaging, S. Jose, California, USA*.

[10] Alessandro Rizzi, C. G. & Marini, D. 2003. A new algorithm for unsupervised global and local color correction. *Pattern Recognition Letters*, 24, 1663–1677.

[11] Øyvind Kolås, I. F. & Rizzi, A. 2009. Stress: A new human visual system inspired image processing method. *(submitted)*.

[12] Majed Chambah, B. B. & Courtellemont, P. 2002. Recent progress in automatic digital restoration of color motion pictures. *SPIE Electronic Imaging, San Jose, California, USA*, 4663, 98–108.

[13] Islam, A. T. & Farup, I. 2010. Enhancing the output of spatial color algorithms. *2nd European workshop on visual information processing, EUVIP, Paris, France*.

[14] DE A. ARAUJO Arnaldo, G. S. J. F. & C., C. G. 2001. A new approach for old movie restoration. *Proceedings of SPIE, the International Society for Optical Engineering*, 4308, 67–77.

[15] Silvio Jamil Ferzoli Guimaraes, A. D. A. A. & Cerqueira, G. 2000. Old movie restoration using opening by surface. *Proceedings of the 13th Brazilian Symposium on Computer Graphics and Image Processing*, 334.

[16] Ebner, M. 2002. A parallel algorithm for color constancy. *Technical Report 296, Universit ¨at W¨urzburg, Lehrstuhl f¨ur Informatik II, W¨urzburg, Germany*.

[17] Funt, B. & Cardei, V. 1999. Committee-based color constancy. *Proceedings of the IS&T/SID Seventh Color Imaging Conference: Color Science, Systems and Applications, Scottsdale, AZ*, 311–313.

[18] Majed Chambah, Alessandro Rizzi, C. G. B. B. & Marini, D. 2003. Perceptual approach for unsupervised digital color restoration of cinematographic archives. *SPIE Electronic Imaging, S. Clara, California, USA*, 5008(138).

[19] L.Khriji, M.Gabbouj, G. & E.D.Ferrandiere. 1999. Old movie restoration using rational spatial interpolators. *6th IEEE International Conference on Electronics, Circuits and Systems, ICECS-99, Cyprus*.

[20] Rizzi, A. Gatta, C. S. C. C. & G. Schettini, R. 2006. Unsupervised color film restoration using adaptive color equalization. *LECTURE NOTES IN COMPUTER SCIENCE*, 3736, 1–12.

[21] Lucia Maddalena, A. P. & Laccetti, G. 2009. A fusion-based approach to digital movie restoration. *Pattern Recognition*, 42(7), 1485–1495.

[22] tai Kim, K. & Kim, E. Y. 2010. Film line scratch detection using texture and shape information. *Pattern Recognition Letters*, 31(3), 250–258.

[23] T. Bretschneider, O. K. & Bones, P. 2000. Removal of vertical scratches in digitised historical film sequences using wavelet decomposition. *Proceedings of the Image and Vision Computing New Zealand*, 38–43.

[24] Silva, A. U. & Corte-Real, L. 1999. Removal of blotches and line scratches from film and video sequences using a digital restoration chain. *in Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP '99),Turkey*, 826–829.

[25] Giuliano Laccetti, L. M. & Petrosino, A. 2005. P-lsr: A parallel algorithm for line scratch restoration. *Proceedings of the Seventh International Workshop on Computer Architecture for Machine Perception*, 225–230.

[26] L. Joyeux, S. Boukir, B. B. & Buisson, O. 2001. Reconstruction of degraded image sequences. application to film restoration. *SPIE Electronic Imaging, San Jose, California, USA*, 19, 503–516.

[27] Raphaël Bornard, Emmanuelle Lecan, L. L. & Chenot, J.-H. 2002. Missing data correction in still images and image sequences. *Proceedings of the tenth ACM international conference on Multimedia*, 355–361.

[28] Ren, J. & Vlachos, T. 2007. Detection and recovery of film dirt for archive restoration applications. *IEEE ICIP*, 4, 21–25.

[29] Ren, J. & Vlachos, T. 2005. Segmentation-assisted dirt detection for the restoration of archived films. *in: Proceedings of BMVC, Oxford, England*, 359–368.

[30] Ren, J. & Vlachos, T. 2007. Efficient detection of temporally impulsive dirt impairments in archived films. *Signal Processing*, 87(3), 541–551.

[31] Marcelo Bertalmio, Guillermo Sapiro, V. C. & Ballester, C. 2000. Image inpainting. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 417–424.

[32] Y.I. Ohta, T. K. & Sakai, T. 1980. Color information for region segmentation. *Computer graphics and image processing*, 13, 222–241.

[33] Meyer, F. 1979. Iterative image transforms for an automatic screening of cervical smears. *The journal of histochemistry and cytochemistry*, 27, 128–135.

[34] Giakoumis, I. & Pitas, I. 1998. Digital restoration of painting cracks. *in Proc. IEEE International Symposium on Circuits and Systems*, 4, 269–272.

[35] Image restoration. http://restoreinpaint.sourceforge.net/index.html. (Visited February 2010).

[36] Cimg library. http://cimg.sourceforge.net/. (Visited January 2010).

[37] Mingw. http://www.mingw.org/. (Visited January 2010).

[38] Imagemagic. http://www.imagemagick.org/script/index.php. (Visited January 2010).

[39] Matlab colormap. www.mathworks.com/access/helpdesk/help/.../colormap.html. (Visited April 2010).

[40] C. Gatta, A. R. & Marini, D. 2006. Local linear lut method for spatial colour-correction algorithm speed-up. *IEE Proceedings - Vision, Image, and Signal Processing*, 153(3), 357–363.

A Library and Environment Used

A.1 CImg library

`CImg library:` CImg stands for *Cool Image*. CImg is a very pleasant toolbox for desiging image processing algorithms in C++. It is comparatively easier to use and it is very efficient as well. CImg defines a single image class which can represent datasets having up to four dimensions (from one dimension scalar signals to three dimension hyperspectral volumetric images), with template pixel types (bool,char,int,float,...). It can handle large image collections and image sequences as well. It can be applied to a wide range of image processing applications.

CImg can use functionalities of external tools/libraries such as ImageMagick, GraphicsMagick Magick++, OpenEXR etc. Moreover, it has a simple plug-in mechanism which allows an user to directly enhance the library capabilities according to the needs.

We have used CImg library for the postprocessing techniques. CImg is also used in developing the STRESS model. We have particularly used the classes like, CImg<T>, CImgDisplay of CImg for developing our methods. For more information about different CImg libraries anyone can look into [36].

A.1.1 Sample coding for "Hello World" with CImg

```
#include "CImg.h"
using namespace cimg_library;
int main() {
CImg<unsigned char> img(640,400,1,3); // Define a 640x400 color image with 8 bits per color component.
img.fill(0); // Set pixel values to 0 (color : black)
unsigned char purple[] = { 255,0,255 }; // Define a purple color
img.draw_text(100,100,"Hello World",purple); // Draw a purple "Hello world" a t coordinates (100,100).
img.display("My first CImg code"); // Display the image in a displa y window.
return 0;
}
```

Which can be also written in a more compact way as:

```
#include "CImg.h"
using namespace cimg_library;
int main() {
const unsigned char purple[] = { 255,0,255 };
CImg<unsigned char>(640,400,1,3,0).draw_text(100,100,"Hello World",purple).display("My first CImg code");
return 0;
}
```

Generally, a very small code can be written that performs complex image processing tasks. The CImg Library is very simple to use and provide a lot of interesting algorithms for image manipulation [36].

A.2 Environment:MinGW and Image Magic

A.2.1 MinGW

`MinGW:` MinGW stands for "Minimalist GNU for Windows". It is a port of the GNU Compiler Collection (GCC), and GNU Binutils, for use in the development of native Microsoft Windows applications. For more information about MinGW, anyone can look into [37].

We have used MinGW for compiling and and running our C++ source codes.

A.2.2 MinGW compiling and running the exe file

`MinGW compiling and running the exe file`: For compiling the C++ source codes and for running the resulting .exe file , we have used the following commands in MinGW

compiler:

Compiling in MinGW: g++ -o exefilename source filename -O2 -lgdi32. Example of compiling with sample source and exe file looks like this:

g++ -o helloworld.exe helloworld.cpp -O2 -lgdi32

Running .exe in MinGW: exe filename -parameters -i input file name -o output file name.

A.2.3 Image Magic

`Image Magic:`ImageMagick is a software which is used to create, edit, and compose bitmap images. It can read, convert and write images in a variety of formats including DPX, EXR, GIF, JPEG, JPEG-2000, PDF, PhotoCD, PNG, Postscript, SVG, and TIFF.

ImageMagick is used to translate, flip, mirror, rotate, scale, shear and transform images, adjust image colors, apply various special effects, or draw text, lines, polygons, ellipses and Bézier curves. For more information about Image Magic, anyone can look into [38].

We have used ImageMagic with CImg library for processing different images in C++.

A.3 Matlab colormap

We have used Matlab for the preprocessing method. Besides this, for showing different intensity heat map we have used the colormap function of Matlab. For more information about Matlab Colormap, anyone can look into [39].

B Lookup Table method: Global and Local Linear Lookup Table(LLL)

B.1 Global Lookup Table method

The idea underlying the look-up table transformation method is to apply the colour-correction algorithm to a small sub-sampled version of the original image and then to build LUT mapping functions between the two subsampled images, the input and the colour corrected. The sub-sampling factor is a trade-off between preserving filtering behaviour and minimising the computational time of the colour-correction stage. These mapping functions are then applied to the input full-size image to obtain the final colour-corrected image. The mapping functions operate on the values of each chromatic channel. The scheme of the global LUT technique can be studied in details from [40].

B.2 Local Linear Lookup Table method

The main idea of this method is to apply the colour-enhancement algorithm to a small sub-sampled version of the input image and to use a modified look-up table technique to maintain the local filtering effect of the colour enhancement algorithm. The method increases the speed of colour-filtering algorithms, reducing the number of pixels involved in the computation by sub-sampling the input image.

A local approach is required for overcoming the limitation of a global LUT technique. The LLL (Local Linear Lookup Table method) method uses a filter, same like in LUT, on a sub-sampled version of the original image to limit the number of pixels involved in the computation but with some improvements explained subsequently.

This method overcomes the limitations of the global approach, taking into account local information in image, sub-sampled version of the original image and the colour corrected image. The idea is to consider the pixel of a position (x, y) and then to generate LUT functions depending on the value of image in (x, y) and on the values of the pixels in suitable subsets of sub-sampled version of the original image and the colour corrected image. The scheme of the LLUT technique can be studied in details from [40].

C Preprocessing Codes and Output Images

C.1 Preprocessing code

For the preprocessing method, we used the Matlab. A portion of the related code for the preprocessing is given below:

```
im = imread('Image.jpg'); //reading the image
figure, imshow(im); // showing the image

max_R=max(R); //finding the maximum channel value for Red channel
max_G=max(G); //finding the maximum channel value for Green channel
max_B=max(B); //finding the maximum channel value for Blue channel

%%%% checking whether it is a 8 bit or 16 bit image
if max_R >255 && max_R <= 65535
divd=65535;
elseif max_R <=255 && max_G > 255 && max_G <= 65535
divd=65535;
elseif max_R <=255 && max_G <=255 && max_B > 255 && max_B <= 65535
divd=65535;
else
divd=255;
end

im = double(im)/divd;
%% Colour space transformation (sRGB to L*a*b*)
cform2lab = makecform('srgb2lab');
lab = applycform(im,cform2lab);

%% PCA implementation and enhancement
PCA_1=reshape(lab(:,:,2:3),size(lab,1)*size(lab,2),2);
PCA_mean=mean(PCA_1,1);
PCA_mean_rearrange=repmat(PCA_mean,[size(lab,1)*size(lab,2) 1]);
PCA_1=double(PCA_1)-PCA_mean_rearrange;

COV_MATRIX=(PCA_1'*PCA_1);%/(size(lab,1)*size(lab,2));


[V D]=eig(COV_MATRIX);

D=diag(D);
[D I]=sort(D,'descend');

V = V(:,I);
% projection onto the components
PCA_1 = PCA_1*V;

m=sqrt(D(1))/sqrt(D(2));

PCA_1(:,1) = PCA_1(:,1)*1.0;
PCA_1(:,2) = PCA_1(:,2)*m^0.5;

PCA_1 = PCA_1*V';

% Adding mean.....
PCA_1=PCA_1+PCA_mean_rearrange;

% new transfored image
f_im = reshape(PCA_1, [size(lab,1) size(lab,2) 2]);
lab(:,:,2:3)=f_im;
f_lab=lab;

Plot the a* vs b* figure for before and after colour balance.

%% Colour space transformation (L*a*b* to sRGB)
cform2lab = makecform('lab2srgb');
rgb = applycform(f_lab,cform2lab);

rgb_reshape=reshape(rgb,size(rgb,1)*size(rgb,2),3);
```

```
r_max=max(rgb_reshape(:,1));r_min=min(rgb_reshape(:,1));
g_max=max(rgb_reshape(:,2));g_min=min(rgb_reshape(:,2));
b_max=max(rgb_reshape(:,3));b_min=min(rgb_reshape(:,3));

rgb_reshape(:,1)=(rgb_reshape(:,1)-r_min)/(r_max-r_min);
rgb_reshape(:,2)=(rgb_reshape(:,2)-g_min)/(g_max-g_min);
rgb_reshape(:,3)=(rgb_reshape(:,3)-b_min)/(b_max-b_min);

f_rgb = reshape(rgb_reshape, [size(rgb,1) size(rgb,2) 3]);
figure, imshow(f_rgb);
% imsave
filename='Image_enhanced.jpg';
imwrite(f_rgb,filename,'jpg');
```

C.2 Some example images with colour balance output

Here in Figure 71, Figure 72, Figure 73, Figure 74, Figure 75, Figure 76 and Figure 77 , we have presented some more images which are colour balanced by the preprocessing method. The CIELAB space for the degraded image and colour balanced(preprocessed) image in the figures show the amount enhancement of colour information.

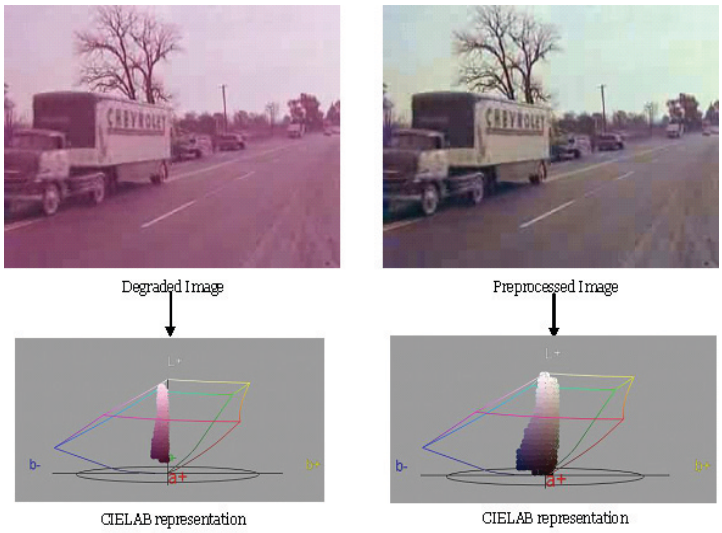

Figure 71: Preprocessing method on degraded image-1.

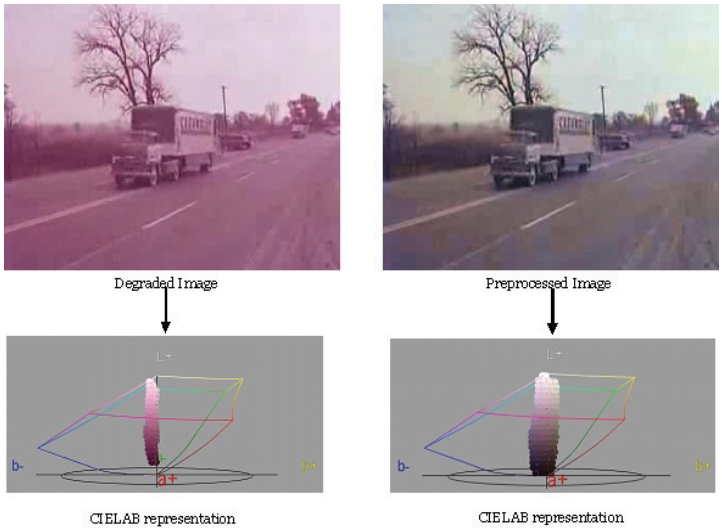

Figure 72: Preprocessing method on degraded image-2.

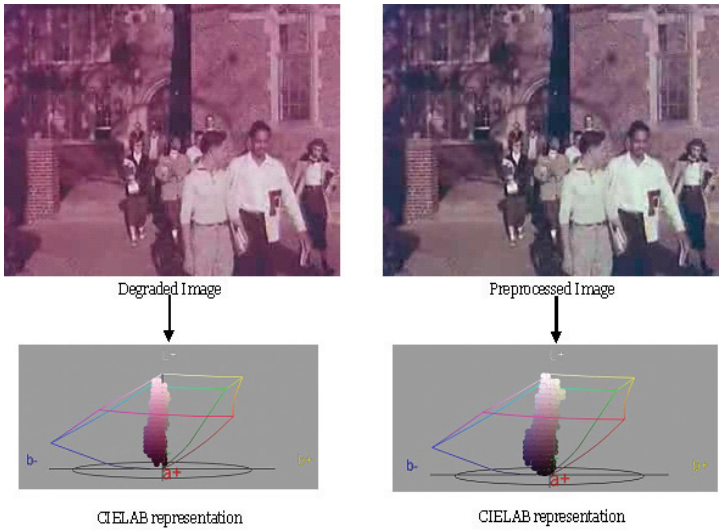

Figure 73: Preprocessing method on degraded image-3.

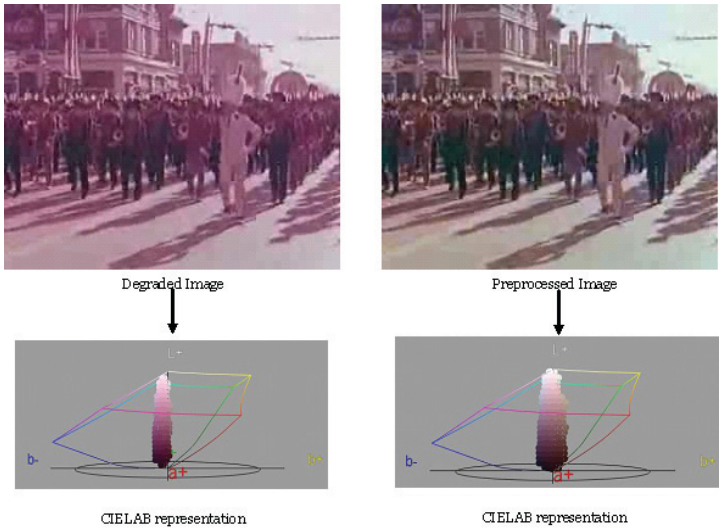

Figure 74: Preprocessing method on degraded image-4.

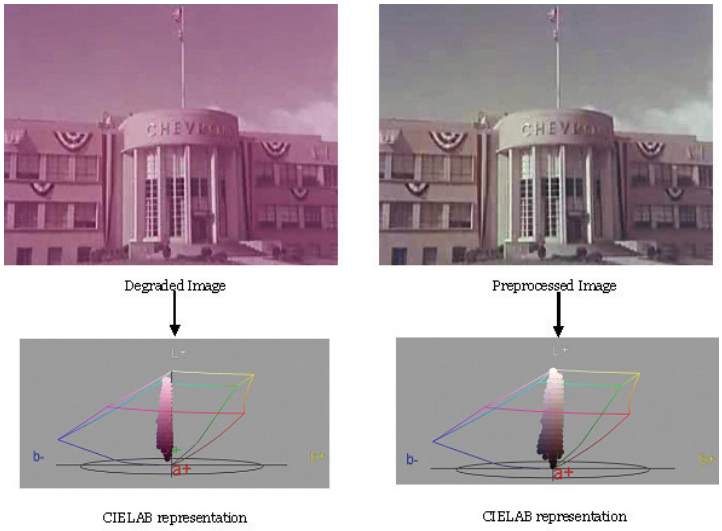

Figure 75: Preprocessing method on degraded image-5.

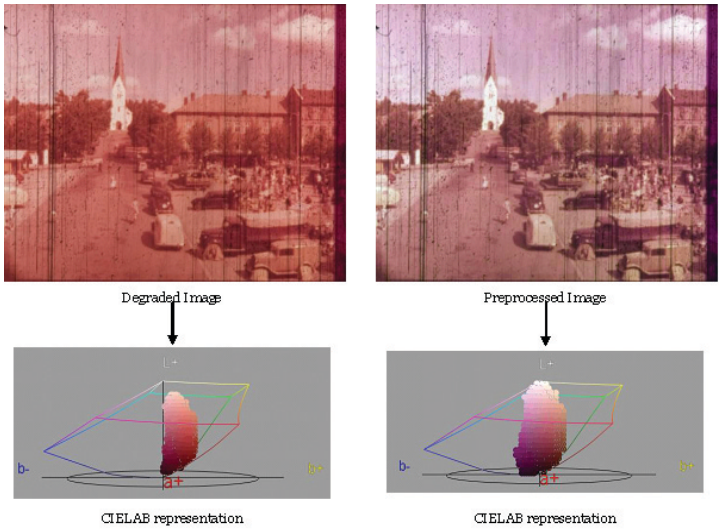

Figure 76: Preprocessing method on degraded image-6.

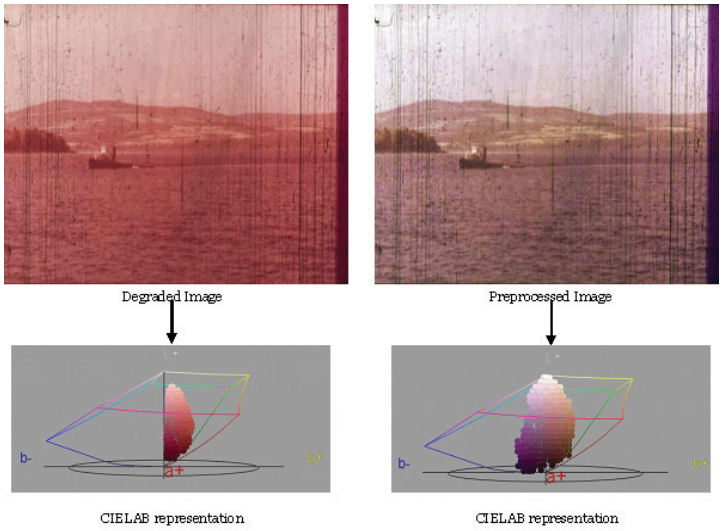

Figure 77: Preprocessing method on degraded image-7.

D Colour Space Conversion Formula

D.1 RGB to L*a*b*

There are no simple formulas to convert RGB values to L*a*b* values. At first, the conversion to CIEXYZ is required and then the resulting XYZ values can be converted to CIELAB space i.e. to L*a*b* values.

RGB to XYZ conversion:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [M] \begin{bmatrix} r \\ g \\ b \end{bmatrix} \qquad \text{(D.1)}$$

$$where, \quad r = R^{\gamma}, \quad g = G^{\gamma}, \quad b = B^{\gamma}$$

XYZ to L*a*b* conversion:

$$\begin{aligned} L^* &= 116 f\left(\frac{Y}{Y_n}\right) - 16 \\ a^* &= 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \\ b^* &= 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right] \end{aligned} \qquad \text{(D.2)}$$

$$where \quad f(t) = \begin{cases} t^{\frac{1}{3}} & t > (\frac{6}{29})^3 \\ \frac{1}{3}(\frac{29}{6})^2 t + \frac{4}{29} & otherwise \end{cases}$$

Here X_n, Y_n and Z_n are the CIE XYZ tristimulus values of the reference white point, the subscript n refers to *normalized*.

D.2 L*a*b* to RGB

Similarly, for the conversion of L*a*b* to RGB, we need first to transform from L*a*b* to XYZ and then the resulting values of XYZ will be converted to RGB.

L*a*b* to XYZ conversion:

This conversion requires a reference white X_r, Y_r, Z_r.

$$
\begin{aligned}
X &= x_r X_r \\
Y &= y_r Y_r \\
Z &= z_r Z_r
\end{aligned}
$$

where

$$
x_r = \begin{cases} f_x^3 & f_x^3 > \epsilon \\ (116 f_x - 16)/\kappa & f_x^3 \leq \epsilon \end{cases}
$$

$$
y_r = \begin{cases} ((L+16)/116)^3 & L > \kappa\epsilon \\ L/\kappa & L \leq \kappa\epsilon \end{cases}
$$

$$
z_r = \begin{cases} f_z^3 & f_z^3 > \epsilon \\ (116 f_z - 16)/\kappa & f_z^3 \leq \epsilon \end{cases} \tag{D.3}
$$

$$
f_x = \frac{a}{500} + f_y
$$

$$
f_y = (L+16)/116
$$

$$
\epsilon = \begin{cases} 0.008856 & Actual CIE Standard \\ 216/24389 & Intent of the CIE Standard \end{cases}
$$

$$
\kappa = \begin{cases} 903.3 & Actual CIE Standard \\ 24389/27 & Intent of the CIE Standard \end{cases}
$$

XYZ to RGB conversion:

Given an XYZ color whose components are in the nominal range [0.0, 1.0] and whose reference white is the same as that of the RGB system, the conversion to RGB is as follows:

$$
\begin{aligned}
R &= r^{1/\gamma} \\
G &= g^{1/\gamma} \\
B &= b^{1/\gamma}
\end{aligned} \tag{D.4}
$$

$$
\textit{where} \begin{bmatrix} r \\ g \\ b \end{bmatrix} = [M]^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}
$$

E Some Portion of Postprocessing Codes

Some relevant codes regarding image read, extraction of channel values and association of those values with the output of STRESS are presented in the following:

Reading the image:

```
CImg<double> img(ifname);
```

Check how much bit the image is: 8 bit or 16 bit

```
max_R=img.get_shared_channel(0).max();
max_G=img.get_shared_channel(1).max();
max_B=img.get_shared_channel(2).max();

if (max_R >255 && max_R <= 65535)
divd=65535;
else if (max_R <=255 && max_G > 255 && max_G <= 65535)
divd=65535;
else if (max_R <=255 && max_G <=255 && max_B > 255 && max_B <= 65535)
divd=65535;
else
divd=255;
```

Extraction of channel values like, max, min and mean of each channel:

```
    max_R = img.get_shared_channel(0).max()/divd;
max_G = img.get_shared_channel(1).max()/divd;
max_B = img.get_shared_channel(2).max()/divd;

min_R = img.get_shared_channel(0).min()/divd;
min_G = img.get_shared_channel(1).min()/divd;
min_B = img.get_shared_channel(2).min()/divd;

mean_R = img.get_shared_channel(0).mean()/divd;
mean_G = img.get_shared_channel(1).mean()/divd;
mean_B = img.get_shared_channel(2).mean()/divd;
```

Extraction of saturation and lightness channel values:

```
if (Keep_Saturation_Lightness)
{

cimg_forXYV(img,x,y,c)
{
img(x,y,c) =(img(x,y,c) /divd)*255;//dividing by divd (whicah has )
}

  img.RGBtoHSL();

  max_S=img.get_shared_channel(1).max();
  min_S=img.get_shared_channel(1).min();
  mean_S=img.get_shared_channel(1).mean();

  max_L=img.get_shared_channel(2).max();
  min_L=img.get_shared_channel(2).min();
  mean_L=img.get_shared_channel(2).mean();

img.HSLtoRGB();
}
```

Association of extracted values to the final output of STRESS image:

```
if(Keep_Lightness_Saturation_Gamma_post)
{

double max_Ls=0.0; double min_Ls=0.0; double mean_Ls=0.0; double gamma_L=0.0; double max_Lout=0.0;
```

```
double max_Ss=0.0; double min_Ss=0.0; double mean_Ss=0.0; double gamma_S=0.0; double max_Sout=0.0;

cimg_forXYV(res,x,y,c)
  {
 res(x,y,c) =res(x,y,c)*255;
  }

res.RGBtoHSL();

max_Ss=res.get_shared_channel(1).max();
min_Ss=res.get_shared_channel(1).min();
mean_Ss=res.get_shared_channel(1).mean();

max_Ls=res.get_shared_channel(2).max();
min_Ls=res.get_shared_channel(2).min();
mean_Ls=res.get_shared_channel(2).mean();

gamma_S=log(mean_S)/log(mean_Ss);
gamma_L=log(mean_L)/log(mean_Ls);

max_Sout=pow(max_Ss,gamma_S);
max_Lout=pow(max_Ls,gamma_L);

cimg_forXYV(res,x,y,c)
{

if (c==1)
res(x,y,c)=pow(res(x,y,c),gamma_S);
else if (c==2)
res(x,y,c)=pow(res(x,y,c),gamma_L);
}

res.HSLtoRGB();
cimg_forXYV(res,x,y,c)
{

res(x,y,c)=res(x,y,c)/255;

}

}
```

F Envelopes of Temporal STRESS and Postprocessing

We have depicted here another example of formation of envelopes by temporal STRESS and postprocessing. The total envelope formation is depicted in Figure 78, Figure 79, Figure 80, Figure 81 and Figure 82.

We know that the maximum and minimum envelopes are the two functions which contains the image signal. These envelopes are the local reference darker and lighter points for each pixel in each chromatic channels [11]. Now, we produce the first frame of the movie cut with a large number of iteration which makes a noise free and good quality image output of the first frame. The envelopes of the first frame are also noise free and they are of good quality. The envelopes of the first frame are presented in Figure 78.

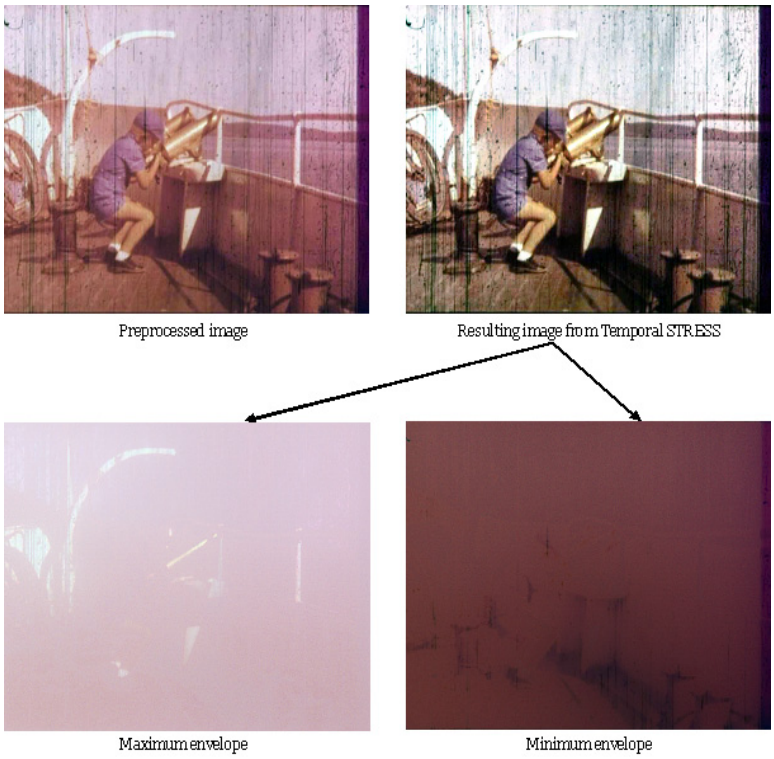

Figure 78: Max and Min envelope of the first frame.

These envelopes in Figure 78 are stored for using with the initial envelopes of the second frame. For producing the next consecutive frames, we use very less number of iteration, hence it takes very less amount of time as well. The initial envelopes produced by the second frame is noisy and is not of good quality, since it is produced by using less number of iterations. The initial envelopes of the second frame is depicted in Figure 79.

Now, the stored envelopes of the first frame are added to the initial envelopes of the second frame by a specified weight factor. The addition of envelopes from first frame with the envelopes of second frame is depiected in Figure 80.

This makes the final envelopes of the second frame noise free and of good qulaity. The final envelopes of the second frame is presented in Figure 81. These envelopes of second frame is stored for using with third frame and so on.

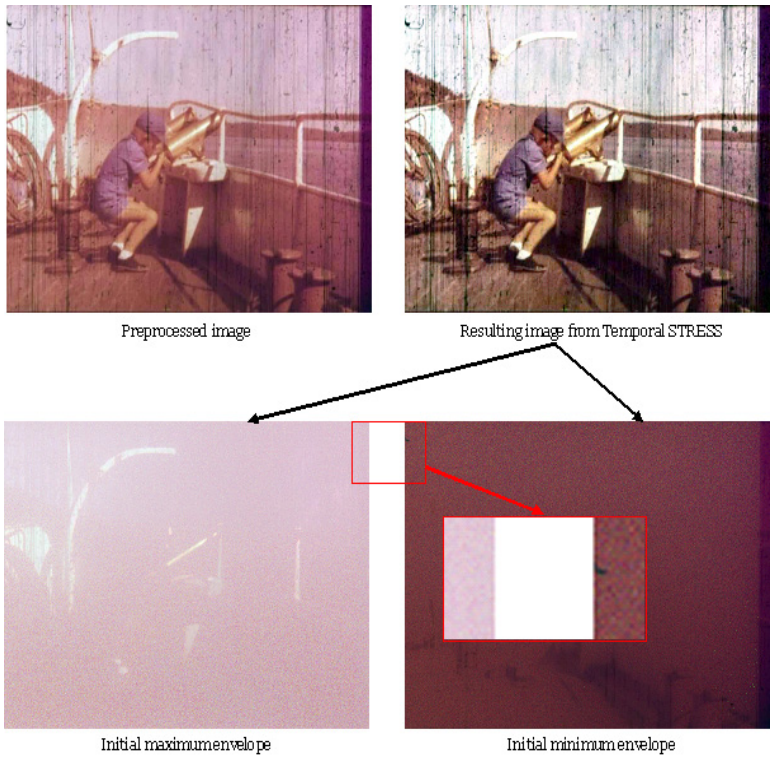

Figure 79: Initial Max and Min envelope of the second frame.

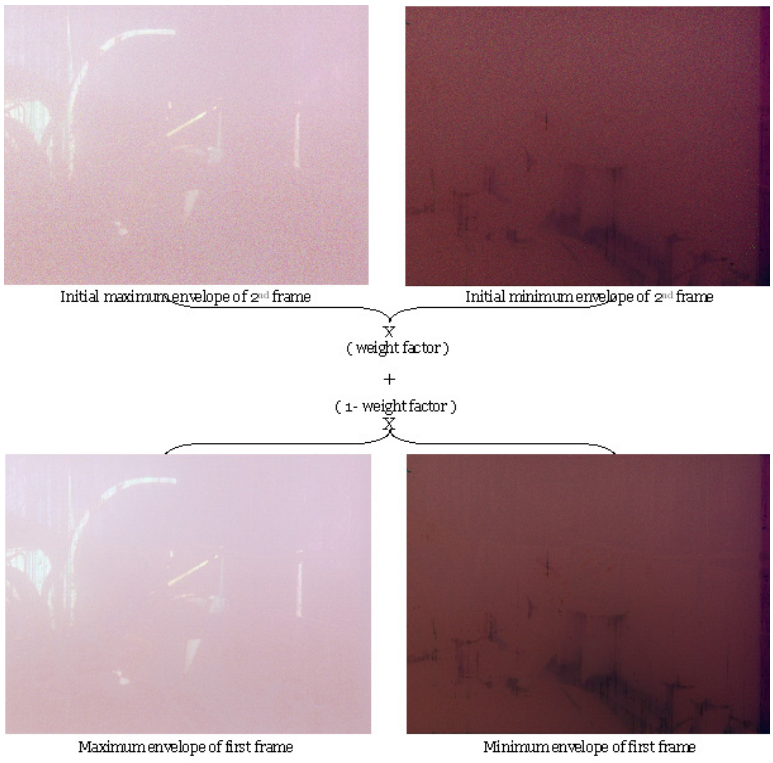

Figure 80: Addition of Max and Min envelope of the first and second frames.

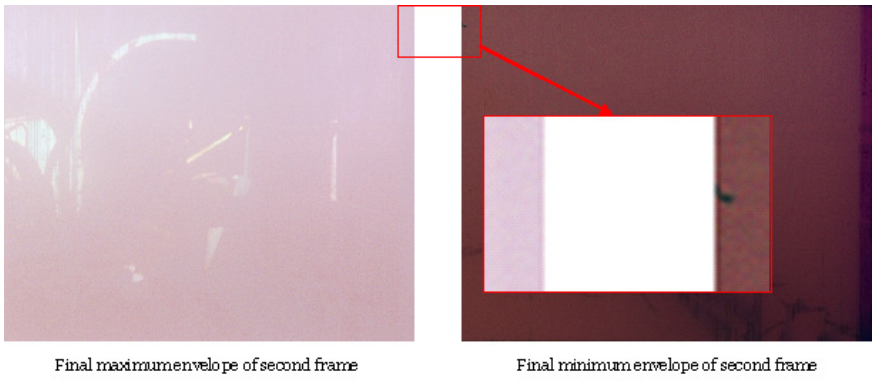

Figure 81: Final Max and Min envelope of the second frame.

The clear difference of the initial and final envelopes of the second frame can be observed from Figure 82. In this way, the resulting consecutive frames possess nice quality image as the first frame and a lot of computational time is saved.

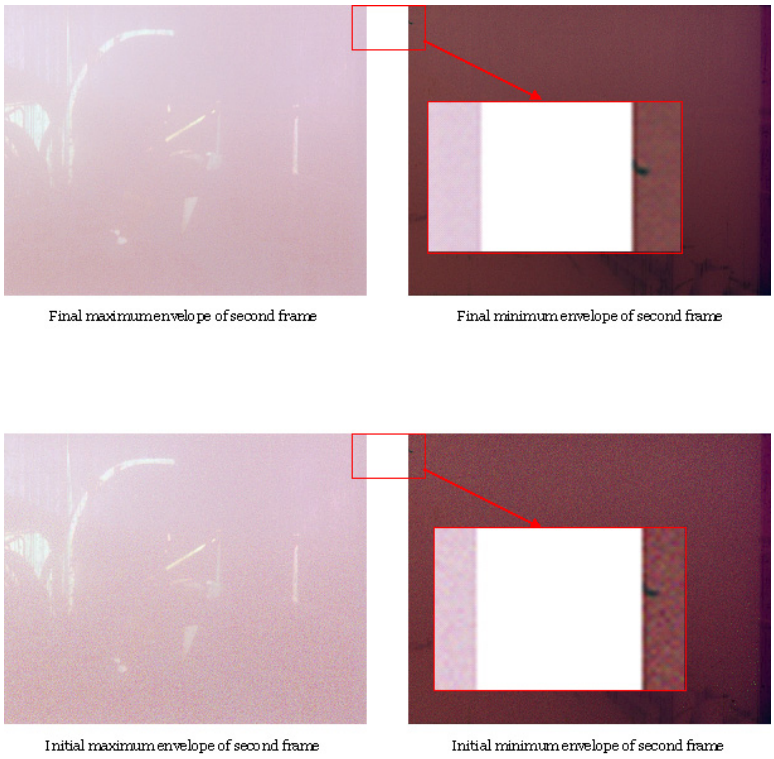

Figure 82: Difference of initial and final Max and Min envelope of the second frame.