Novel Video Completion Approaches and Their Applications

Ali Mosleh

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

September 2013

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Ali Mosleh

Entitled: Novel Video Completion Approaches and Their Applications

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Electrical and Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Rajamohan Ganesan — Chair

Dr. Jean Meunier — External Examiner

Dr. Wen-Fang Xie — External to Program

Dr. Yousef R. Shayan — Examiner

Dr. Mohammad Mannan — Examiner

Dr. Nizar Bouguila, Dr. A. Ben Hamza — Thesis Supervisor

Approved by

Chair of Department or Graduate Program Director

Dean of Faculty

Abstract

Novel Video Completion Approaches and Their Applications

Ali Mosleh, Ph.D.

Concordia University, 2013

Video completion refers to automatically restoring damaged or removed objects in a video sequence, with applications ranging from sophisticated video removal of undesired static or dynamic objects to correction of missing or corrupted video frames in old movies and synthesis of new video frames to add, modify, or generate a new visual story. The video completion problem can be solved using texture synthesis and/or data interpolation to fill-in the holes of the sequence inward. This thesis makes a distinction between still image completion and video completion. The latter requires visually pleasing consistency by taking into account the temporal information. Based on their applied concepts, video completion techniques are categorized as inpainting and texture synthesis. We present a bandlet transform-based technique for each of these categories of video completion techniques. The proposed inpainting-based technique is a 3D volume regularization scheme that takes advantage of bandlet bases for exploiting the anisotropic regularities to reconstruct a damaged video. The proposed exemplar-based approach, on the other hand, performs video completion using a precise patch fusion in the bandlet domain instead of patch replacement. The video completion task is extended to two important applications in video restoration. First, we develop an automatic video text detection and removal that benefits from the proposed inpainting scheme and a novel video text detector. Second, we propose a novel video super-resolution technique that employs the inpainting algorithm spatially in conjunction with an effective structure tensor, generated using bandlet geometry. The experimental results show a good performance of the proposed video inpainting method and demonstrate the effectiveness of bandlets in video completion tasks. The proposed video text detector and the video super resolution scheme also show a high performance in comparison with existing methods.

Acknowledgements

Foremost, I would like to express my greatest gratitude to my supervisors Dr. Nizar Bouguila and Dr. A. Ben Hamza for their help, guidance and encouragement throughout my PhD studies.

I would like to thank the members of the thesis committee for their encouragement, insightful comments, and hard questions at all levels of the research project. I thank my fellow lab-mates at Concordia University: Ali Sefidpour, Ali Shojaie, Mohamed Almashrgy, Taoufik Bdiri, Wentao Fan, Weijia Su, Parisa Tirdad, Sara Balar, Ola Ameyri, and Tarek Elguebaly for their support, motivation and all the good time we have had in the last four years. Also, I am thankful to my friend Reza Mohammadtaheri and my cousins Mazair Gomrokchi and Iman Owrangi for their endless support and help in the personal aspects of my life. I am also truly grateful to Dr. Farhad Zargari for enlightening me the first glance of research.

I am deeply thankful to my parents, sister, and brother for all the love and support they have provided me over the years. Nothing would be possible without them.

Last but not least, I would like to thank my wife, the love of my life, Shima Mohammadtaheri, who has brought hope, motivation, success, pleasure, and immense peace in my life since the earliest step she took into my world.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Video Completion

Digital completion is a technique commonly used for restoring damaged images or videos. It is often performed using interpolation and reconstruction of the missing parts. Generally speaking, completion techniques are divided into two main categories: inpainting, and texture synthesis. Inpainting-based methods employ interpolation techniques to reconstruct the missing data using the neighboring available structures. In the texture synthesis methods, a promising available part of the 2D or 3D data is selected and propagated to the missing parts[1]. It is however important to distinguish between image and video completion techniques.

Image completion is a technique for restoring damaged images or completing the area of removed objects, which are manually selected by a user to be removed. Typical applications of image completion include old painting restoration or removal of the scratches from pictures. One of the pioneering works in this field is [1], which applies non-linear partial differential equations (PDE) to perform image inpainting in an effort to imitate what artists manually do to fix old pictures. This idea is developed in [2] with a precise and efficient solution that applies fluid dynamic Navier-Stoke equations. Also, the PDE-based image inpainting is followed in [3] which derives a third order PDE equation based on Taylor

[1]Often, in literature, image and video completion techniques are called inpainting techniques, whether they are texture synthesis or inpainting based approaches.

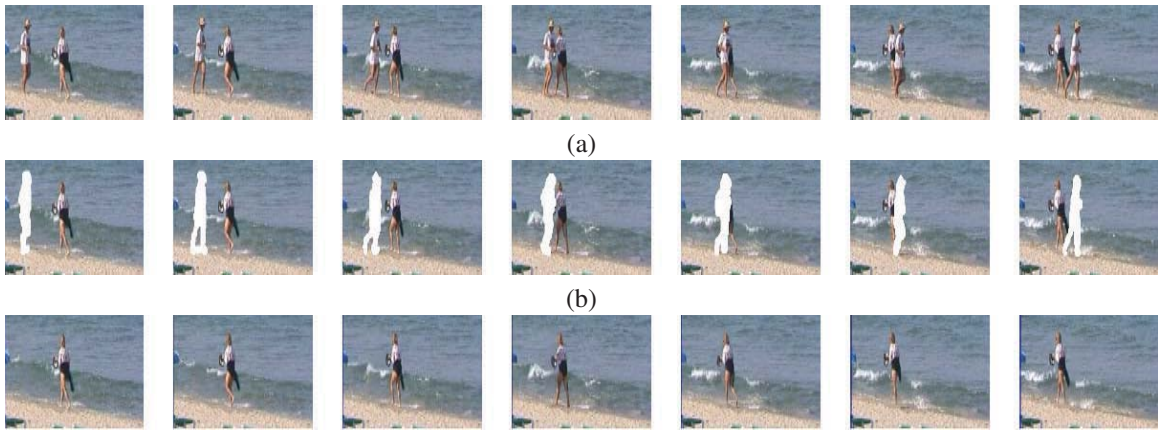


Figure 1.1: Samples of video completion provided in [9]. a) Original frames. b) Frames after object removal. c) Frames' hole completion result.

expansion to propagate the level lines. A similar idea of interpolation is used in other image inpainting methods such as [4, 5]. In the texture synthesis based image completion methods, texture synthesis is done by sampling a texture model in the undamaged part of the image, based on the geometrical patterns of the image, and fill-in the hole by the sampled texture pattern. One of the most promising methods is the exemplar-based method proposed in [6] which efficiently takes the property of the edges into account to perform completion. There are some other innovative methods that apply the structure of the image texture, such as [7], which uses curvelet structure analysis to perform the inpainting task. The technique in [8] applies a modified grouplet transform to find the geometrical structure of the image texture, and then performs image inpainting by a global optimization in the grouplet domain.

Video completion is defined as a problem of filling the missing parts of video frames caused by the removal of unwanted objects or restoration of the scratched frames. Figure 1.1 shows sample frames of a video sequence and the visually pleasing completion results after removing one object. There are important issues about video completion/inpainting that make it different from a single image completion. First, manually selecting a target area to be removed is almost impossible, in large part because we deal with many frames, and not only a single 2D image. Second, human recommended structural information,

such as texture samples, is not easy to obtain. Therefore, video completion involves a robust tracking and also an effective textural propagation. Third, one may consider video inpainting as an extension of image inpainting in the form of frame-by-frame image inpainting. However, video inpainting is a much more challenging task than what it seems. Due to sensitivity of human visual system to temporal aliasing, temporal consistency is of paramount importance in video completion tasks. This must be taken into consideration in addition to spatial consistency, which must be preserved for each frame in order to produce visually pleasing results.

Video completion methods can be broadly divided into two main categories: Inpainting-based methods, and exemplar-based methods. The inpainting-based video completion methods are similar to the image inpainting ones that perform filling-in process via interpolation of the available data mostly by applying PDEs. However, these methods are usually applied frame-by-frame on the video, and thus, most of them do not satisfy the temporal consistency. It is worth noting that this group of video completion methods suffer from smoothness effect in the task of inpainting large missing regions. However, inpainting-based video completion methods are very effective in the case of dealing with small missing data. As a result, they can be appropriate for the error concealment problem and also logo, text and scratch removal tasks that normally need to handle small and thin missing regions. The exemplar-based methods are quite similar to the texture synthesis based image inpainting methods. However, in these techniques, the texture (structure) sampling is carried out by considering all the frames, and also the moving objects of the frames, not just a single image as it is done in the texture synthesis based image completion techniques. Several exemplar-based video completion methods have been proposed in the literature; some of the most effective ones are discussed in this thesis. These methods are preferable in video completion of large missing regions generated after undesired object removal.

1.2 Contributions

In this thesis we look at the video completion task from a different view. To this end, we benefit from the advantages of new generations of wavelets in video completion. We introduce an efficient video completion approach by applying the bandlet transform [10], which is one of the latest developed generations of wavelets. The effectiveness of bandlets in both inpainting-based and exemplar-based video completion has been exploited in our research. First, an examplar-based video completion method that takes advantage of a bandlet-based patch fusion strategy is proposed. Then, an inpainting-based video completion that performs the task by means of an optimization in the bandlet transform domain is proposed. In fact, this is a 3D volume regularization algorithm that benefits from bandlet bases in exploiting the anisotropic regularities. The resulting video of the exemplar-based approach is provided to the 3D regularization in order to refine the completion results. In contrast to many of the state-of-the art video inpainting techniques, our method preserves the temporal consistency quite well. The structural information revealed by bandlets has an important role in the completion task. Our experimental results show that the proposed video completion technique maintains both the spatial and temporal consistency, and also demonstrate the effectiveness of the bandlet transform in video completion. To validate the importance of video completion in classical restoration problems, we studied two different video restoration problems: i) automatic video text/caption removal, and ii) video spatial super-resolution. Then, we developed a solution for each problem using video inpainting.

We present a two stage framework for automatic video text removal in order to detect and remove embedded video texts and fill-in their remaining regions by appropriate data. In the video text detection stage, text locations in each frame are found via an unsupervised clustering performed on the connected components produced by the stroke width transform (SWT). Since, SWT needs an accurate edge map, we develop a novel edge detector that benefits from the geometric features revealed by the bandlet transform. Next, the motion patterns of the text objects of each frame are analyzed in order to localize video texts. The detected video text regions are removed, and then the video is restored by the 3D volume regularization inpainting scheme. The experimental results demonstrate the effectiveness

of both our video text detection approach and the video completion technique in the case of text removal, and consequently the whole automatic video text removal and restoration process.

A new framework for single image and video super-resolution is also proposed in this thesis. The main idea is to employ the geometric details of the image in a way that minimizes the possible artifacts caused by the super-resolution process. To this end, we benefit from the bandlet transform that captures the image surface geometry quite effectively. The proposed single image super-resolution method is a two-stage scheme. In the first step, edges and high frequency details of the image are interpolated in the enlarged image. This interpolation uses a structure tensor, which is modified according to the geometry layer revealed by the image bandlets. In the second stage, the edge interpolated image is fed to the inpainting scheme in order to estimate the pixel values for the new spatial locations within the enlarged image. These locations are treated as missing pixels, and then a precise regularization in the bandlet domain is performed to fill-in the missing pixels. In a bid to avoid flickering artifacts after frame-by-frame spatially super-resolving videos, a pixel intensity refinement stage is added to the procedure which takes into account the motion flows of the frames. Due to the use of geometric details that bandlet transform captures and the edge interpolation stage, our method results in high quality, high resolution images with no over-smoothing or blurring effect while in the case of video sequences it also preserves temporal consistency.

The contributions of this thesis are summarized as follows:

- ☞ A comprehensive literature review is presented [11] and a video completion/inpaiting framework is proposed [12–14].

- ☞ An accurate video text detection technique is proposed and is employed in an automatic video inpaiting-based text removal scheme [15, 16].

- ☞ A single image and video super-resolution approach is proposed that employs an inpainting scheme and a bandlet-based structure tensor [17].

1.3 Thesis Overview

The organization of this thesis is as follows:

- ❏ In Chapter 2, the most practical existing video completion, both inpainting based and exemplar-based methods, are introduced and discussed. Also, an overview of the bandlet transform is presented.

- ❏ In Chapter 3, we propose a new method for video completion tasks by applying the bandlet transform. First, an exemplar-based video completion scheme is presented. Then, an inpainting-based video completion method is introduced to refine the completion results of the exemplar-based technique. Experimental results are provided to show the effectiveness of our method.

- ❏ In Chapter 4, we propose a novel technique for text localization in video sequences and introduce an automatic video caption removal scheme by employing our proposed video inpainting approach.

- ❏ In Chapter 5, we develop a new image/video super-resolution algorithm as another application of our proposed video completion technique.

- ❏ Finally, Chapter 6 provides concluding remarks and future work directions.

Chapter 2

Background

In this chapter, a comprehensive literature review of video completion techniques, including the interpolation methods and the exemplar-based ones, is presented. Since our work largely relies on the bandlet transform, an overview of the bandeltization processes is also presented.

2.1 Video Completion in Literature

2.1.1 Interpolation-based Methods

Interpolation-based techniques are mainly edge continuing methods, which apply data interpolation mostly using PDEs to pull the data (edge structure) from the boundary to the interior region of the missing region. These methods are often performed frame-by-frame and do not take the temporal information of the video sequences into account quite well. Therefore, temporal consistency is not satisfied by these methods. Besides, in the case of dealing with large missing pixels, these methods produce a blurring effect in the inpainting results. An example of the blurring effect is illustrated in Figure 2.1. However, they are appropriate for small missing areas, such as scratches in old movies.

(a) (b)

Figure 2.1: Blurring effect after inpainting. (a) Original frame. (b) Inpainted frame after removal of the bench and three people [12].

Navier-Stokes Fluid Dynamics Adaptation to Digital Inpainting

One of the pioneering PDE-based methods is the one introduced in [2] which was followed by [1]. The approach applies ideas from classical fluid dynamics to propagate isophote lines of the image continuously from the exterior into the inpainting zone. The basic idea is to assume the image intensity as a stream function for a 2-D flow. The Laplacian of the image intensity is considered as the vorticity of the fluid; it is pushed to the inpainting region by a vector field which is defined by the stream function. The algorithm is designed to continue isophote lines, while matching gradient vectors at the boundary of the region of the image to be filled-in. The approach directly uses the Navier-Stokes equations for fluid dynamics, which have the advantage of well-developed theoretical and numerical results while introducing the importance of propagating both the gradient direction (geometry) and gray-value (photometry).

The scheme of the PDE-based algorithm proposed in [1] which is designed to project gradient of the smoothness of the image intensity in the direction of the isophotes has lead to a discrete approximation of the PDE:

$$I_t = \nabla^{\perp} I . \nabla \triangle I \tag{2.1}$$

where $\nabla^{\perp}$ denotes the perpendicular gradient $(-\partial_y, \partial_x)$ and $\triangle$ denotes the Laplace operator $(\partial^2_{y^2} + \partial^2_{x^2})$ of the image intensity I. An additional anisotropic diffusion of the image produces a PDE in the form of:

$$I_t = \nabla^{\perp} I . \nabla \triangle I + \nu \nabla . (g(|\nabla I|) \nabla I) \tag{2.2}$$

There is a vital condition that, in the inpainting region of the image, the isophote lines

which are in the direction of $\nabla^{\perp} I$ must be parallel to the level curves of the smoothness $\triangle I$ of the image intensity. Therefore, here, the goal is to develop Eq. (2.1) or Eq. (2.2) to a steady state that satisfies this condition which for $\nu = 0$ becomes:

$$\nabla^{\perp} I . \nabla \triangle I = 0 \tag{2.3}$$

These equations can be adapted to the fluid dynamics. Equation (2.1) is a transport equation that converts the image intensity I along level curves of the smoothness $\triangle I$. This is true if one notes that Eq. (2.1) is equivalent to $DI/Dt = 0$ where D/Dt is the material derivative $\partial/\partial t + \boldsymbol{v}.\nabla$ for the velocity field $\boldsymbol{v} = \nabla^{\perp} \triangle I$. In fact, the velocity field $\boldsymbol{v}$ converts I in the direction of $\triangle I$.

Navier-Stokes equations govern the incompressible Newtonian fluids. These equations relate the velocity vector field $\boldsymbol{v}$ to a scalar pressure p:

$$\boldsymbol{v}_t + \boldsymbol{v}.\nabla \boldsymbol{v} = -\nabla p + \nu \triangle \boldsymbol{v}, \quad \nabla . \boldsymbol{v} = 0. \tag{2.4}$$

In the 2-D space, the stream function Ψ that satisfies $\nabla^{\perp} \Psi = \boldsymbol{v}$ belongs to the velocity field $\boldsymbol{v}$, which is divergence-free. By means of taking the curl of the first equation in Eq. (2.4) and applying some basic facts about the 2D geometry, a simple diffusion equation can be computed:

$$\boldsymbol{w}_t + \boldsymbol{v}.\nabla \boldsymbol{w} = \nu \triangle \boldsymbol{w} \tag{2.5}$$

where vorticity $\boldsymbol{w} = \nabla \times \boldsymbol{v}$ satisfies Eq. (2.4).

In fact, the vorticity is related to the stream function through the Laplace operator $\triangle \Psi = \boldsymbol{w}$. In absence of viscosity, $\nu = 0$, the Euler equations of inviscid flow are obtained. In terms of the stream function, based on Eq. (2.5) a steady state inviscid flow must satisfy:

$$\nabla^{\perp} \Psi . \nabla \triangle \Psi = 0 \tag{2.6}$$

This equation implies that the stream function, the vorticity and the Laplacian of the stream function must have the same level curves.

The stream function for inviscid fluids in two dimensions Eq. (2.6) satisfies the same equation as the steady state image intensity Eq. (2.3). Therefore, in order to solve the 2D

Table 2.1: Analogy of Navier-Stokes and image inpainting problems [2].

Navier-Stokes	Image Inpainting
stream function Ψ	image intensity I
fluid velocity $v = \nabla^{\perp}\Psi$	isophote direction $\nabla^{\perp} I$
vorticity $w = \triangle\Psi$	smoothness $w = \triangle I$
fluid viscosity ν	anisotropic diffusion ν

inpainting problem, one can find a stream function for inviscid fluid equation, which has very strong and developed solutions.

The problem is formulated to design a Navier-Stokes based method, using the vorticity-stream form Eq. (2.5), for the image inpainting application. This is done by considering Ω as the inpainting region and the image intensity I_0 as a smooth function with possibly large gradient outside Ω, and I_0 and $\triangle I_0$ be known on the boundary area $\partial\Omega$. Analogous quantities are summarized in Table 2.1.

Considering $\boldsymbol{w}$ as the smoothness of the intensity $\triangle I$, the transport equation Eq. (2.2) can be solved. Instead, using $\boldsymbol{w}$ the vorticity transport equation is solved:

$$\partial w/\partial t + \boldsymbol{v}.\nabla w = \nu\nabla.(g(|\nabla w|)\nabla w) \tag{2.7}$$

At the same time, the Poisson problem is solved to recover the image intensity I which defines velocity field ($\boldsymbol{v}$=$\nabla^{\perp} I$) in (2.7):

$$\triangle I = w, I|_{\partial\Omega} = I_0 \tag{2.8}$$

For $g = 1$ the numerical solutions of Eq. (2.7-2.8) lead to a classical way to solve both the dynamic fluid equations and to obtain a steady state [2].

This method does not take into account the temporal information of the video sequences and is performed frame-by-frame. Moreover, it is only appropriate for the narrow regions to be inpainted and causes blur effect for large regions. However, this method opened a new issue in the field of automatic digital image/video restoration and has been followed by many other methods in the literature.

Discrete Laplace Regularization-Based Method

An image and video inpainting method was introduced in [18], which benefits from discrete p-Laplace regularization on a weighted graph introduced in [19]. This method does not

perform frame-by-frame inpainting unlike many of the inpainting methods, but considers the whole video as a volume to perform the inpainting in all the frames to fill-in the missing regions.

A function $f^0 : V \to \mathbb{R}^m$ is considered as the image or a video (here, an image is taken into account as a single frame video). This function is defined over the set of vertices, V, of a weighted graph $G = (V, E, w)$. The subset of nodes of the missing region is denoted by $V_0 \subset V$. The purpose of this inpainting method is to interpolate the known values of f^0, $V \backslash V_0$, to the unknown region, V_0, which is solved as the discrete regularization using the weighted p-Laplace operator by minimizing the following equation:

$$f^* = \min_{f \in H(V)} \left\{ \frac{1}{p} \sum_{v \in V} |\nabla f(v)|^p + \frac{\lambda(v)}{2} ||f - f^0||^2_{H(V)} \right\} \tag{2.9}$$

where λ is the fidelity parameter, ∇f denotes the weighted gradient of the function f over the graph, $\nabla f(v)$ is the local variation of the weighted gradient operator of the function f at the vertex v.

$$\lambda(v) = \begin{cases} \lambda = \text{constant} & \text{if } v \in V \backslash V_0 \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

$$|\nabla f(v)| = \sqrt{\sum_{u \sim v} w(u, v)(f(v) - f(u))^2} \tag{2.11}$$

Equation (2.9) involves the main term, $\nabla f(v)$, which represents the weighted gradient over the graph. The problem of Eq. (2.9) has a unique solution for $p \geq 1$, which satisfies:

$$(\triangle_p) f(v) = \frac{1}{p} \sum_{u \sim v} \gamma(u, v)(f(v) - f(u)) \tag{2.12}$$

$$\gamma(u, v) = w(u, v)(|\nabla f(v)|^{p-2} + |\nabla f(u)|^{p-2}) \tag{2.13}$$

where $(\triangle_p f)(v)$ is the weighted p-Laplace operator on $H(V)$ with $p \in (0, +\infty)$ at the vertex v.

In the graph, $G_{k_1,k_2,k_3} = (V, E, w)$, $u \sim v$ denotes that a vertex u belongs to the neighborhood of v defined as:

$$N_{k_1,k_2,k_3}(v) = \left\{ \begin{array}{l} u = (i', j', t') \in V \backslash V_0 : \\ |i - i'| \leq k_1, |j - j'| \leq k_2, |t - t'| \leq k_3 \end{array} \right\}. \tag{2.14}$$

A patch in the form of 3D (spatial and temporal) centered at vertex v is a box with size $r_x \times r_y \times r_t$ is denoted by $B(v)$ and for each patch a feature vector is defined as:

$$F(f^0, v) = f^0(u), \quad u \in B(v), \quad u \in V \backslash V_0 \tag{2.15}$$

Then, the following two weight functions of the graph are considered:

$$w_L(u, v) = \exp\left(-\frac{|f(u) - f(v)|^2}{2\sigma_d^2}\right) \tag{2.16}$$

$$w_{NL}(u, v) = w_L(u, v) \exp\left(-\frac{||F(f^0, u) - F(f^0, v)||^2}{h^2}\right) \tag{2.17}$$

where h is the standard deviation based on variation of $||F(f^0, u) - F(f^0, v)||$

The regularization problem is solved using Gauss-Jacobi algorithm in [19], which is modified, in brief, as:

$$\left\{ \begin{array}{l} f^{(0)} = f^0 \\ \gamma^{(k)}(u, v) = w(u, v)(||\nabla f^{(k)}(v)||^{p-2} + ||\nabla f^{(k)}(u)||^{p-2}) \\ f^{(k+1)}(v) = \frac{\sum_{u \sim v} \gamma^{(k)}(u,v) f^{(k)}(u)}{\sum_{u \sim v} \gamma^{(k)}(u,v)} \end{array} \right.$$

The regularization is applied iteratively (k denotes the iteration number) for each node of the graph until the missing part is completely filled-in. To avoid error propagation, at each iteration, once the entire outer line is processed it is removed from the subset of missing region (V_0).

This method, however, has a serious computational bottleneck and similar to other methods which use PDEs, it only works well for small missing regions to be inpainted.

Rank Minimization Approach

Video inpainting is considered as a matrix rank minimization problem in the work presented in [20]. The scheme starts with finding a set of descriptors that encapsulate vital

information to restore the damaged pixels. Then an optimal estimate for the descriptors of the missing region is found. Finally, these descriptors are used to reconstruct the pixels.

The spatio-temporal descriptors are assumed to be generated by a stationary Gaussian-Markov random process. Therefore, the values of the descriptors of the k^{th} frame stored in a vector $\boldsymbol{f}_k$ are related to the descriptor values of the previous frames by an autoregressive moving average model with exogenous inputs model (ARMAX) model:

$$\boldsymbol{f}_{k+1} = \sum_{i=0}^{m-1} g_i \boldsymbol{f}_{k-i} + h_i \boldsymbol{e}_{k-i} \tag{2.18}$$

where g_i, h_i are fixed coefficients. $\boldsymbol{e}(.)$ denotes a stochastic input assumed to be always an impulse since the spectral density of $\boldsymbol{e}$ is absorbed in the coefficients g_i and h_i ([21], Chapter 10).

The model Eq. (2.18) is not simply a combination of surrounding pixels, it denotes the value of a descriptor of the actual pixels in a lower size space-time domain and relates the present and the past values. This model is used to find the descriptors of the uncorrupted regions and then it is used to inpaint the missing values. However, finding the missing values of each descriptor $\boldsymbol{f}$ is not necessarily done explicitly. A rank-minimization problem is adopted to estimate the missing values of each descriptor. The observed and missing descriptors are denoted by f_k^o and f_k^m, respectively. The idea is to find the values of f^m in such a way that they are maximally consistent with f^o, in the sense of the model introduced in Eq. (2.18). The minimum value of m such that Eq. (2.18) explains the observed data, is given by the rank of matrix constructed from measurements. The simplest model to explain this data can be a model in which the missing descriptors that minimize the matrix rank are corresponded. Instead of exhaustive rank minimization, the problems are solved by a convex semi-definite programming as follows:

1. Form the Hankel matrix as:

$$H_f = \begin{pmatrix} f_1 & f_2 & \cdots & f_{n/2} \\ f_2 & f_3 & \cdots & f_{n/2+1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n/2} & f_{n/2+1} & \cdots & f_{n-1} \end{pmatrix} \tag{2.19}$$

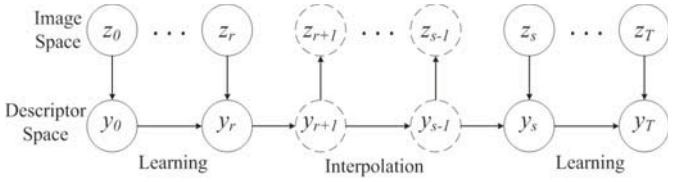


Figure 2.2: The general model of the rank minimization-based video completion.

where f denotes either the observed data f_k^o if the frame k is present, or the unknown value f_k^m if the frame contains missing pixels. The total number of frames is denoted by N.

2. Solve the following Linear Matrix Inequality (LMI) optimization problem, minimize $Tr(Y) + Tr(Z)$ subject to $H_f = \begin{bmatrix} Y & H_f \\ (H_f)^T & Z \end{bmatrix} \geq 0$ where $Y^T = Y \in \mathbb{R}^{n \times n}$, $Z^T = Z \in \mathbb{R}^{n \times n}$ and $H_f \in \mathbb{R}^{n \times n}$ This helps estimate the values of F^m to be consistent with f^o.

This minimization is employed in order to inpaint the damaged frames. The interpolated (and/or extrapolated) values of the descriptors are used to reconstruct the missing information using a nonlinear combination of the pixels. A general model for the video inpainting system, including dimensionally reduction (generating the descriptors), rank minimization, and non-linear reconstruction using the descriptors, is shown in Figure 2.2. The inpainting algorithm based on rank minimization is shown in Algorithm 2.1. The steps of the algorithm are iterated until no missing region remains in the sequence.

Applying this algorithm on several videos has shown promising results. The advantages of this method are: 1) It is computationally attractive since the algorithm optimizes the use of spatio-temporal and dynamic information. 2) The method is not restricted to the case of periodic motion, static or stationary cameras. 3) The method can be used to extrapolate frames which are used to generate and synthesize new textures from the same family.

Algorithm 2.1 Inpainting by means of rank minimization.

1: Given a sequence of fames I_t, $t = \{1, ..., T\}$, the target is occluded/corrupted in $r \leq t \leq s$. Thus, the target z_t is extracted from the unoccluded frames $1 \leq t \leq r$ and $s \leq t \leq T$.
2: For each t, $t = \{1, \ldots, r, s, \ldots, T\}$ map the pixels to the lower dimensional domain of descriptors using Localy Linear Embedding (LLE).
3: Apply the rank minimization algorithm using the Hankel matrix to find the descriptor values $\{y_r, \ldots, y_s\}$ corresponding to the missing pixels.
4: Reconstruct the pixels z_t from y_t by means of inverse mapping i.e. descriptor to pixel domain.
5: Use a robust tracking to keep the track of the centroid of the target in the occluded frame.
6: Fill theses locations by the reconstructed pixels.

Complex Ginzburg-Landau Equation in Inpainting

Ginzburg-Landau equation was originally developed to describe phase transitions in superconductors near their critical temperature. A solution to this equation develops homogenous regions (in one, two or three dimensions) with phase transition edges. These homogenous areas can be, for instance, constant gray value intensity regions in an image. Its analytical properties and adaptability to restore higher than 2-dimensional data has motivated the use of Ginzburg-Landau equation in image and video inpainting [22].

Ginzburg and Landau derived the following estimations for thermodynamic potential energy function of semiconductors by considering the kinetic and potential energy:

$$F(u, \nabla u) := \frac{1}{2} \int_{\Omega} \underbrace{|-i\nabla u|^2}_{\text{kinetic term}} + \underbrace{\alpha|u|^2 + \frac{\beta}{2}|u|^4}_{\text{potential term}} \tag{2.20}$$

where $-i$ is a negligible factor which comes from quantum mechanics, α and β are physical constants. These factors are set to $\alpha = -\frac{1}{\varepsilon^2}$ and $\beta = -\alpha$. Then the state of minimal energy satisfies the Euler-Lagrange equation of $F(u, \nabla u)$ as follows:

$$\frac{\delta F}{\delta u} = \Delta u + \frac{1}{\varepsilon^2}(1 - |u|^2)u = 0 \tag{2.21}$$

Transition region is a transient that separates different phases. The coherence length which correlates with the width of transition region is denoted by ε, in physics. An analytical

formula to solve Eq. (2.21) is given by:

$$u(x) = \frac{e^{\frac{\sqrt{2}}{\varepsilon}x} - 1}{e^{\frac{\sqrt{2}}{\varepsilon}x} + 1} \tag{2.22}$$

where the order function $u : \mathbb{R} \to \mathbb{R}$ satisfies the boundary condition $\lim_{x \to \pm\infty} u(x) = \pm 1$. The complex Ginzburg-Landau equation has been solved in the domain of inpainting in order to reconstruct the missing pixels.

Considering D as the image and Ω as the inpainting subset of D, $\overline{u}^0 :\to [-1, 1]$ is the gray level values scaled to $[-1, 1]$ i.e., -1 and 1 corresponds to white and black, respectively. The real part of the complex valued function $u^0 : D \to \mathbb{C}$ identifies the function $\overline{u}^0$ and the imaginary part is selected as $I(u^0) = \sqrt{1 - (R(\overline{u}^0))^2}$ where $|u^0(x)| = 1$ for all $x \in D$.

The real part of the solution of Eq. (2.21) contains any value in $[-1, 1]$, which is an estimation of the inpainting region while the imaginary part has an absolute value of 1 for each pixel. In the case of color images, each color component can be inpainted separately. In order to reduce the artifacts, the color images can be inpainted by generalizing the equation Eq. (2.21) to vector value functions ($u : D \to \mathbb{C}^n$). Hence, the problem of inpainting reduces to finding $u : \Omega \to \mathbb{C}$ ($\mathbb{C}^3$ for color images) which satisfies Eq. (2.21) and the Dirichlet boundary condition $u|_{\partial\Omega} = u^0|_{\partial\Omega}$.

The complex Ginzburg-Landau equation can be considered as a system of PDEs. In order to find a solution for Eq. (2.21) with the boundary condition, the steepest descent method (or any relaxation method) can be applied to the following PDE:

$$\frac{\partial u}{\partial t} = \Delta u + \frac{1}{\varepsilon^2}(1 - |u|^2)u. \tag{2.23}$$

In the case of real valued u, this equation can be considered as a variant of Allen-Cahn equation:

$$u_t = \Delta u + \psi'(u). \tag{2.24}$$

If u is complex or vector-valued and $\psi(u)$ has a stable minimum for $|u| = 1$, which is true in the case of inpainting problem, equation Eq. (2.24) is called Ginzburg-Landau. An explicit forward in time finite-difference approach is used to integrate Eq. (2.23). The

explicit forward in time finite-difference is done by discretizing (2.23) in space and time, which has the following form:

$$u_{i,j}^{t+1} = u_{i,j}^{t} + \delta t(\Delta u_{i,j}^{t} + \frac{1}{\varepsilon^2}(1 - |u_{i,j}^{t}|^2)u_{i,j}^{t}) \tag{2.25}$$

where $\Delta u_{i,j} = u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}$. Here, $u_{i,j}$ denotes the color vector intensity at the pixel location (i, j). Also, the time step is such that $\delta t < \varepsilon^2$. It is worth noting that this explicit forward in time finite difference scheme does not work on rectangular inpainting areas as efficiently as it works on irregular regions like scratches and text regions.

The Ginzburg-Landau equation can be generalized to any number of dimensions. Therefore, it can be applied to inpaint the three-dimensional gray valued intensity functions, specifically video sequences considered as three dimensional volumes. Although the method is very straightforward and flexible to be applied in the 3D space and satisfies the temporal consistency, it leads, however, to a smoothing effect in the highly textured regions of the video volume.

Projection Based Inpainting Using Wavelets

The frequency-spatial representation provided by wavelet transform is used in a projection onto convex sets (POCS) scheme [23] in order to find and apply the correlation between the missing pixels and their neighbors [24]. The main assumption in the restoration process is that the lost block of pixels very likely contains a continuation of its surrounding data. Wavelets help determine the frequency details of the neighboring data. The algorithm works based on the constraints applied on the missing pixels in both the wavelet and spatial domains. The process of projection onto image and wavelet domain and applying the constraints are formulated as:

$$I_{n+1} = C_i P_{wi} C_w P_{iw}(I_n) \tag{2.26}$$

where I_{n+1} is the inpainted window after $n + 1$ iterations, I_n is the current inpainting window, P_{iw} is the wavelet transform of the image considered as the projection from image to frequency-spatial domain, and C_w represents the constraints applied in the wavelet

domain. Projection from wavelet to image domain is represented by P_{wj}. Finally, C_i indicates the constraints applied in the image domain. As for the wavelet constraint C_w, the maximum and minimum of wavelet coefficients in the known region are found. Then, the reconstructed pixel corresponding to the lost pixel in the inpainting region is forced to be between these maximum and minimum values. The constrained wavelet coefficients are then projected onto the image domain (P_{wi}) using the inverse wavelet transform. The newly grown pixels are constrained (C_i) too. Both wavelet and signal domain constraints satisfy the criteria that the new pixels must be i) as sharp as the surrounding pixels, ii) on the continuation of prominent edges along the missing region and, iii) matching the surrounding area texture wisely.

In order to deal with videos, the frames are stacked as a 3D volume. Then, the missing part of the video is considered as a 3D hole and 3D wavelet and inverse wavelet transforms are applied in the algorithm.

2.1.2 Exemplar Based Methods

The exemplar-based methods use a texture model from the undamaged parts of the image/video and try to reconstruct the missing parts using this model. Many of the exemplar-based video completion methods are based on the exemplar-based image (spatial) completion algorithm proposed in [6]. These methods try to adapt this image inpainting method to the temporal information, which is available and useful in video sequences. Therefore, this section begins with a brief review of this well-known and efficient exemplar-based image inpainting method [6].

Criminisi's Exemplar-Based Image Inpainting

The exemplar-based spatial inpainting [6] is an order-based algorithm, in which the best matching source patch (block) is transferred inward the filling area. The algorithm is severely dependent on the order of filling-in, determined by two factors: confidence and priority. In fact, the algorithm gives high priority of synthesis to the regions (patches) of the target (missing) area which lie on the continuation of image structures and are surrounded

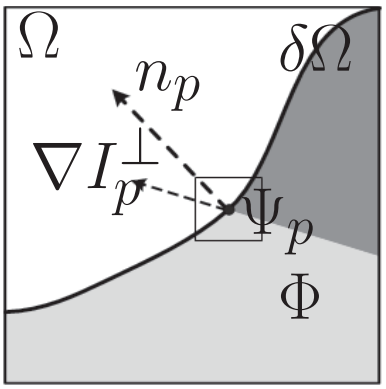


Figure 2.3: Notation diagram of the exemplar based image inpainting in [6].

by high-confidence pixels. The confidence value of each pixel reflects the confidence and reliability in the pixel's value. The algorithm uses three main steps:

1) Computing Patch Priorities: The priority computation is biased toward those patches which are on the continuation of the strong edges determined by data term $D(p)$ and surrounded by high-confidence pixels approximated by $C(p)$. The priority of each patch Ψ_p centered at p is computed using:

$$P(p) = C(p) \times D(p) \tag{2.27}$$

where the confidence $C(p)$ and the data $D(p)$ value are obtained as follows:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (I-\Omega)} C(q)}{|\Psi_p|} \tag{2.28}$$

$$D(p) = \frac{|\nabla I_p^{\perp}.n_p|}{\alpha} \tag{2.29}$$

where Ψ_p is a patch centered at p and $|\Psi_p|$ is its area in the video frame I. α is the normalization factor which is 255 for gray-scale images and n_p is the normal vector of the border $\delta\Omega$ of the missing are Ω and the source region $\Phi = I \backslash \Omega$ in the point p. Isophote (direction and intensity) at point p is denoted by $\nabla I_p^{\perp}$. Figure 2.3 illustrates the notation of the inpainting problem. It is worth noting that the initial value of $C(p)$ is set to $C(p) = 0$ $\forall_p \in \Omega$ and $C(p) = 1$ $\forall_p \in I - \Omega$.

2) Propagating Texture and Structure Information: Once the patch to be filled with the highest priority is found ($\Psi_{\hat{p}}$), it is filled with the extracted data directly sampled from the

source (Φ) region. The best match to be extracted from the source is found based on the lowest Sum of Squared Differences (SSD):

$$\Psi_{\hat{q}} = \arg\min_{\Psi_{\hat{q}\in\Phi}} SSD(\Psi_{\hat{p}}, \Psi_q). \tag{2.30}$$

Whenever the best patch in the source region is found, its area which corresponds to the missing area of the target patch is replicated in the missing area. This propagates both the structure and the texture information of the source Φ to the target Ω.

3) Updating Confidence Value: After the patch Ψ_p has been filled with the new pixels, the confidence value for the border around Ψ_p is computed as:

$$C(\hat{p}) = C(p), \quad \forall p \in \Psi_p \cap \Omega. \tag{2.31}$$

This confidence update reduces the confidence color values of the pixels near the center of the missing region, which is reasonable.

The above three steps are repeated until the hole in the image is fully filled. This algorithm not only has a very high performance to complete the missing regions of still images, but also has been adopted in many of the video completion methods to produce highly reliable results. Some of these video completion methods are discussed in the next sub-sections.

Patawardhan's Occluding and Occluded Object Inpainting

Patawardhan *et al.* [25] proposed an effective video inpainting method which performs the video completion task in two cases: 1) Inpainting damaged regions or moving objects in the static background, and 2) inpainting the moving objects which are partially occluded by other moving objects. The method is very similar (in fact, adapted) to the well-designed exemplar-based image inpainting method proposed in [6] (Section 2.1.2). The method performs segmentation based on optical flow, as the pre-processing stage, to separate the moving objects of the video from the static background.

The stationary background is filled-in by available temporal information of the corresponding inpainting zone in undamaged frames, as the first process. Then, the priority based spatial inpainting [6] is performed to fill the remaining unfilled region after temporal filling.

Temporal filling of the background requires computation of priority of pixels. First, a confidence value $C(p)$ is assigned to each pixel p in every frame, which is set to zero for the pixels in the moving or damaged regions and is initialized to one otherwise. The second relevant parameter is the data value $D(p)$, which is based on the availability of temporal information at location p:

$$D(p) = \frac{\sum_{p \in \partial\Omega, t=-\delta n \cdots \delta n} M_t(p)}{\beta} \tag{2.32}$$

where Ω is the hole to be inpainted, $\partial\Omega$ is the boundary, and $M_t = 0$ if p is damaged or is moving, else $M_t = 1$; t indicates the relative index of frames to the current frame which is denoted by $t = 0$ (p belongs to current frame). β is equal to $2n + 1$, where n indicates the total number of considered frames. The priority value of the filling-in at $p \in \partial\Omega$ is calculated using Eq. (2.27). This priority value determines the damaged pixel location to be first filled. Then, the temporal information patches having the highest confidence value should be copied from the temporally nearest location to the location p. The confidence value of all the previously damaged pixels in the patch Ψ_p is updated as in Eq. (2.28), once the patch is copied to the highest priority location p.

Whenever $D(p) = 0, \forall p \in \partial\Omega$ there is no temporal information available to be copied. Therefore, a priority-based spatial inpainting of the hole is performed, exactly similar to the three steps mentioned in Section 2.1.2.

The method, in the next step, completes the moving objects which are partially occluded by other objects. This process is done independently from background filling process, as described in the following steps:

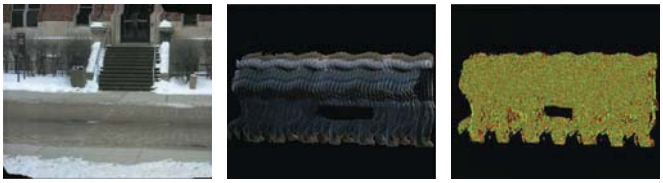
1. Find the highest priority location, using Eq. (2.27), to fill-in the moving object in the current frame.

2. Search for the best moving patch from the undamaged parts of the video sequences using Eq. (2.30).

3. Copy only the moving pixels of the best found match to the current fame and update the confidence value Eq. (2.28).

4. Set the priority of the not moving pixels in the copied patch to zero.

(a)



(b) (c) (d)

Figure 2.4: Preprocessing performed in [26] to enhance the completion results of Patawardhan's method. (a) Sample input video frames. (b) Background mosaics. (c) Foreground mosaics. (d) Optical flow mosaics.

5. Repeat 1 to 4 until all the damaged pixels get zero priority.

In this process, the priority value is computed as Eq. (2.27) but its data term is computed differently. In Eq. (2.29), I is replaced by a motion confidence image M_c, which is zero if p belongs to the background and one if p belongs to moving objects to compute the data term.

This video completion satisfies, firstly, filling-in the background while maintaining temporal consistency and, secondly, filling-in the moving foreground while keeping the motion globally consistent. However, this method is performed only under stationary scene (i.e. fixed camera).

Patawardhan *et al.*'s method was followed in [26] to improve the inpainting scheme while permitting some camera motions, however the priority and confidence parameters are applied the same way as described in the aforementioned paragraphs. This method performs a preprocessing segmentation which results in three frame-mosaics [26]: foreground, background and optical flow, as shown in Figure 2.4. This segmentation leads to a better performance and reduces the search space and consequently provides a faster implementation.

Shih's Exemplar-Based Method

Another exemplar-based video inpainting method was introduced in [27], which is also based on the spatial inpainting algorithm proposed in [6] with some modifications. This method uses a modified patch matching, which incorporates the edge features. The data term used in the priority calculation is also redefined. In this algorithm, a simple video tracking is applied to produce a foreground video.

The method follows the notation introduced in Figure 2.3 similar to the method discussed in Section 2.1.2. A simple region segmentation in the CIE Lab color space is applied to convert I to a set of segments $\acute{I}$. Considering p_i and p_j as the pixels and s_i and s_j the segments, the following conditions are satisfied:

1. $\forall p_i \in I, \forall p_j \in I, p_i$ and p_j are adjacent pixels.

2. $SSD_{CIELab}(p_i, p_j) < \delta_c \Rightarrow$ puts p_i and p_j in the same segment.

3. $\forall s_i \in I, \forall s_j \in I, s_i$ and s_j are adjacent segments.

4. $pn(s_i) - pn(s_j) < \delta_{pn} \Rightarrow$ makes s_i and s_j in the same segment.

where $pn(s)$ computes the number of pixels in the segments and $SSD_{CIELab}(p_i, p_j)$ calculates the sum of squared differences in the CIE Lab color space. An appropriate value for δ_c is 3 based of the experiments and δ_{np} is usually chosen between 50 to 100 for different video sequences, which should be selected manually. The results of the segmentation ($\acute{I}$) are then converted to binary format, BI, using Crispening algorithm. Therefore, $\Phi_\varepsilon \subset BI$ is the edge map of the corresponding area in $\Phi \subset I$ (source area).

After the segmentation step, the inpainting is done using a revised version of the algorithm introduced in Section 2.1.2; the initial confidence value $C(p)$ of each pixel in I is computed as in [6]; i.e. $C(p)$ is set to $C(p) = 0\ \forall p \in \Omega$ and $C(p) = 1\ \forall p \in \Phi$. Let Ψ_p be a patch centered at $p \in \Omega$ then the confidence value is computed using equation Eq. (2.28). It is obvious that the confidence value computes the number of the useful pixels of each patch. So far, the terms are exactly similar to those of [6]. The main difference of this method and the one in [6] is calculation of the data term. Instead of using isophote [6],

percentage of edge pixels in the patch is computed in this method to obtain the data term, in addition to the color variation in the patch:

$$\forall p \in \delta\Omega, D(p) = \min\{1, (\Sigma_{q \in (\Psi_p \cap \Phi_\varepsilon)} c)\} \times var(\Psi_p)/|\Psi_p| \tag{2.33}$$

where $var(\Psi_p)$ is the variance of the pixels in Ψ_p. Then the priority is obtained using Eq. (2.27). Since both terms $C(p)$ and $D(p)$ contain no structural information, a more sophisticated patch matching strategy, called patch template matching, is used in this method.

Let $\Psi_{\widehat{p}}$ be a patch with the highest priority, which is defined as:

$$\Psi_{\widehat{p}} = \arg\max\{P(p), p \in \delta\Omega\}. \tag{2.34}$$

The larger the patch, the better the chance to find the best match. Therefore, the neighboring pixels are considered, in addition to only using the useful pixels of the patch. A patch template in his method is defined as:

$$\Gamma_{\widehat{p}} = \bigcup_{\pm\Psi_{\widehat{p}}} (\Psi_{\widehat{p}} \cap \Phi) \neq \emptyset \tag{2.35}$$

where $\emptyset$ is the empty set and $\pm\Psi_{\widehat{p}}$ indicates the patch $\Psi_{\widehat{p}}$ plus its surrounding patches (eight neighbors). Then, the best patch template can be found by:

$$\Gamma_{\widehat{q}} = \min_{\Gamma_q \in \Phi_q^r} d(\Gamma_{\widehat{p}}, \Gamma_q) \tag{2.36}$$

$$d(\Gamma_{\widehat{p}}, \Gamma_q) = SSD_{CIELab}(\Gamma_{\widehat{p}}, \Gamma_q) \times \min\{1, (\sum_{q \in (\Gamma_q \cap \Phi_\varepsilon)} c)\} \tag{2.37}$$

where Φ_q^r is a region of the source image centered at q by a distance of r pixels and the constant c is the weight of the useful pixels in the patch template. In fact, the distance function Eq. (2.37) takes the SSD and the number of useful pixels into account. In the last step of the algorithm, the confidence value can be updated using Eq. (2.31) as in [6]. But, the update strategy is revised to incorporate the degree of similarity into the confidence map:

$$C(p) = C(\widehat{p}) \times (d(\Psi_{\widehat{p}}, \Psi_{\widehat{q}})/\alpha), \forall p \in \Psi_{\widehat{q}} \cap \Omega \tag{2.38}$$

The normalization value of α sets the range of $d(\Psi_{\widehat{q}}, \Psi_{\widehat{q}})/\alpha$ to $(0, 1]$.

To recover a background in a stationary video, the method does not perform the above inpainting directly frame-by-frame. Instead, the frame-difference of the current background frame and all the background frames of the video is calculated to find the proper temporal information to fill the background. After this temporal filling-in, the remained hole in the background is filled using the aforementioned spatial inpainting method and the found best matches are propagated to all the background frames.

In the case of non-stationary video, a simple motion estimation algorithm is performed to find the global motion vectors and compensate the motion for slow foreground movements and a more complicated algorithm for faster foreground [27].

Exemplar-Based with Ghost Shadow Removal

The video completion may cause "ghost shadows", i.e, due to the temporal discontinuity of inpainted area, flickers may be produced after video completion. This problem is partially solved in [28] using a modified video inpainting method applying a motion segmentation mechanism. However, it is still challenging to deal with more complicated videos. This problem was addressed in [29] in detail.

The inpainting parameters in the method introduced in [29] are exactly the same as those in [27] (Section 2.1.2). The main difference is the object tracking algorithm which is applied in [29] to remove the ghost shadow effect. 4SS motion estimation algorithm [30], which is defined for block-based motion estimation in video coding is used to compute the motion vectors. This motion estimation method maps the moving objects in different layers and the non-stationary background. The blocks with similar motion in this method are placed in the same layer. Whenever the objects in each layer are tracked and removed, the video inpainting process is performed on the bottom layer to the top one to fill-in the holes after object removal. The motion segmentation, in brief, is performed in the steps of Algorithm 2.2. For more detail about this motion segmentation one can refer to [29]. After division of the frames into motion layers, based on different motions, a target object is supposed to be selected by the user and it should be tracked in the entire video sequence. The tracking is performed by Algorithm 2.2.

To avoid the ghost shadow effect, the inpainted area of the previous frame needs to

Algorithm 2.2 Motion segmentation algorithms.

1: Apply the 4SS motion estimation algorithm in the HSI color space to compute the motion map and edge detection.
2: Merge the segments (Closing and Opening).
3: The remaining blocks which are not processed in the previous step are merged using similar average motion vectors.

Algorithm 2.3 Tracking algorithm in the motion layers.

1: The bounding box, Box_i, of the object to be removed is selected by the user.
2: The motion segment which has the largest number of overlapping pixels with Box_i and its average motion vector is found.
3: Using the average motion vector, another bounding box in the next frame, Box_{i+1} is found.
4: The image inpainting method of section 2.1.2 is done on each Box_x and results in Box_y in the entire sequence.
5: For each Box_x the gray-scale difference with its corresponding Box_y is computed.
6: The tracked object is specified by thresholding the result of the difference in the previous step.

be considered. Moreover, since objects may occlude each other, their relationship, which is complicated, should be seriously taken into consideration. The completion problem is shown in Figure 2.5 with the definitions summarized in Table 2.2.

Here, the goal is how to fill-in the hole at frame $t+1$. To this end, a transformation function $T(\Omega, \Delta)$ is defined, which takes the hole Ω and the average motion vector Δ to locate the object in the next frame. For example, for a panning video, the transformation function can be a simple translation such that $\Omega^{t+1} = \text{Tarnslate}(\Omega^t, \delta_{xy})$. In general, the inpainting scheme is $\acute{\Omega} = Inpaint(\Omega)$, where $\acute{\Omega}$ is the completed hole. Therefore, the two functions can be combined as:

Table 2.2: Definitions in Figure 2.5.

Ω^t	hole to be completed at frame t
$\acute{\Omega}^{t+1}$	hole of Ω^t completed at frame $t+1$
Ω_E	hole to be completed at an upper layer
w^t	surrounding block of Ω^t, $\Omega^t \subseteq w^t$
w^{t+1}	surrounding block of $\acute{\Omega}^{t+1}$, $\acute{\Omega}^{t+1} \subseteq w^{t+1}$
$\triangle_{xy}$	average motion vector of background
δ_{xy}	average motion vector of foreground
Ω^{t+1}	hole to be completed at frame $t+1$

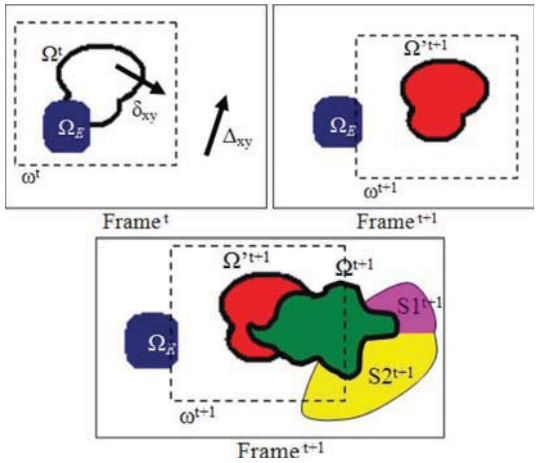


Figure 2.5: Separation of holes in consecutive frames [29].

$$Translate(Inpaint(\Omega^t), \Delta_{xy} + \delta_{xy}) = Translate(\acute{\Omega}^t, \Delta_{xy} + \delta_{xy}) = \acute{\Omega}^{t+1}$$

This transformation function can be generalized for different camera motions such as tilting and perspective video which takes the scaling factor S_{xy} and the rotation factor θ, $T(\Omega, \Delta_{xy}, S_{xy}, \theta)$ in addition to average motion vector.

The inpainting area in each frame contains several regions considering the motion segmentation:

$$\Omega_A = \acute{\Omega}^{t+1} \setminus \Omega^{t+1}$$

$$\Omega_B = \acute{\Omega}^{t+1} \cap \Omega^{t+1}$$

$$\Omega_C = (\Omega^{t+1} \setminus \acute{\Omega}^{t+1}) \cap w^{t+1}$$

$$\Omega_D = (\Omega^{t+1} \setminus w^{t+1})$$

For each of the above regions the inpainting strategy is different as follows:

$$Inpaint(\Omega_A) = w^{t+1} \cap \acute{\Omega}^{t+1}$$

$$Inpaint(\Omega_B) = Inpaint + (\Omega_B, \acute{\Omega}^{t+1})$$

$$Inpaint(\Omega_C) = Inpaint + (\Omega_C, w^{t+1})$$

$$Inpaint(\Omega_D) = Inpaint + (\Omega_D, (S1^{t+1} \cup S2^{t+1}) \setminus w^{t+1})$$

Since the original surrounding of Ω^{t+1} is kept, region Ω_A at frame $t+1$ should be discarded. To maintain temporal continuity, patches from $\acute{\Omega}^{t+1}$ are used to inpaint Ω_B. Ω_C is a new area so the patches from w^{t+1} are used to inpaint it. The patches outside w^{t+1} and inside frame $t+1$ of the same motion segment (e. g., $S1^{t+1}$ or/and $S2^{t+1}$) are used to inpaint Ω_D.

This method exhibits a very high performance in terms of video completion while

avoiding ghost shadows. Besides, it can be applied in the presence of many camera motions.

Jia's Tacking and Fragment Merging

In [31] another exemplar-based video completion method very similar to the previous methods was introduced, but applies the trackability information to find the highest priority. This method uses frame fragments in the process to find a proper target and a source data. The method performs the completion task in three main steps iteratively, similar to the previous methods:

1. Find the most promising target pixel at the boundary of the missing hole and define a space-time fragment around it.

2. Find the source fragment most similar to the target fragment in the search region of the video.

3. Merge the source and the target fragments to reduce the hole size.

There is a special way to find the merit of each of the target fragments in this method, which takes both trackability and the available information into account. In addition, tracking of the moving objects is done based on the well-known "mean shift" algorithm, to reduce the search space. The hole is filled-in, fragment-by-fragment, with a "graph cut" algorithm to merge the source and the target fragments in a way that retains the fine details especially at the borders.

In order to select the best target fragment T the term merit, O_T, is defined and the largest merit value indicates the best target fragment candidate:

$$O_T = I_T + kC_T \tag{2.39}$$

where I_T and C_T are the info map and trackability map of the target fragment, T respectively, and an optimum choice for k is 2 or 3 to give larger weight to the trackability of the fragments [29]. The info map and trackability map are defined as:

$$I_T = \sum_{v \in T} M_v \tag{2.40}$$

$$C_T = \sum_{v \in T, M_v = 0} \tau_v \tag{2.41}$$

where M_v is the matte value of the pixel $v(x, y, t)$ and τ_v is a Boolean value indicating whether the pixel v is trackable or not. In fact, for a target fragment, which contains unknown pixels, C_T is the number of trackable pixels. An unknown pixel is trackable if and only if there is an adjacent known neighborhood that contains an object that can be tracked in the video. This information gives priority to those rare objects which are trackable in the entire video sequence.

After finding the best target fragment T using the merit value, an appropriate source fragment S should be found. The method applies tracking to avoid unnecessary fragments to reduce the search space. Trackable and non-trackable target fragments in the video sequence are treated differently in this method. A non-trackable target fragment has unchanging texture and color through the whole video. Moreover, global temporal consistency, which is a basic need in the video completion scheme, tells us that the non-trackable fragment should be filled in a same way in each frame. Therefore, a global search in whole the video to fill-in a non-trackable fragment is pointless. Instead, one can use the current frame's source region information to fill-in the non-trackable fragment. On the other hand, for trackable target fragments a known trackable neighborhood N usually overlaps the target, which is active through the entire video. Obviously, the target belongs to a moving object which is separate from the stationary background. In this case, N is tracked using the mean shift tracking algorithm through the video which results in a set of small windows including the moving object N. The search is performed on this set of windows for the trackable target fragment.

After determination of both T and S, these two fragments should be combined in a way that does not lead to an obvious joint edge. To circumvent this limitation, a graph cut method is applied to find the "least visible seam" (i. e. the one for which, pixel differences across the seam are as small as possible) between the target and the source overlap area. The overlap region of the target and the source is defined as $O = T \cap S$. In this method, the color difference value c_o is needed at each pixel in the overlap area O:

$$c_o = ||c_t - c_s|| \tag{2.42}$$

where $o(x,y,t), t(x,y,t), s(x,y,t) \in O, T, S$ and c_o is expressed as an r, g, b vector.

Then, using the pixels of the overlap area, O, an undirected weighted graph, G, is built, in which, for each connected pair of pixels, o_i and o_j with color differences, c_{o_i} and c_{o_j}, the weight is computed as:

$$w_{ij} = \begin{cases} k\left(1 - \exp(-\frac{||c_{o_i}||+||c_{o_j}||}{2\sigma^2})\right) & \text{if } N(o_i, o_j) \\ \infty & \text{otherwise} \end{cases} \tag{2.43}$$

where k and σ are constant values about 10 and 5, respectively. If the pixels are six-connected $N(.,.)$ returns 'true'. The lower weight implies that the adjacent pixels in T and O are similar and desirable to have the least visible seam. Hence, the least visible seam is the one that gives a minimum cut for the graph.

The experiments presented in [31] show that this method is very efficient and fast enough, with very visually pleasing results, due to the use of the efficient tracking and graph cut algorithms. However, this method is designed only for simple stationary videos and does not work properly in the presence of camera motions.

Motion Layer Based Object Removal

Zhang *et al.* focused on motion layer segmentation to remove the moving objects and fill-in the resulted missing areas [32]. This method maintains the consistency based on the motion compensation and a graph cut algorithm for video completion. The algorithm assumes that the overlapping order of the motion layers in all the frames of the video sequence remains the same. Based on this assumption, the motion layers are first extracted using a level set representation and a graph cut algorithm, and consequently the occluded and occluding pixels and, also, the layer orders are determined. The video completion technique can be summarized in the following steps:

1. Motion layer segmentation of the video frames and determination of the layers' orders as shown in Figure 2.6a.

2. Removing the undesired object (motion layer), which results in a missing region in each layer (Figure 2.6b).

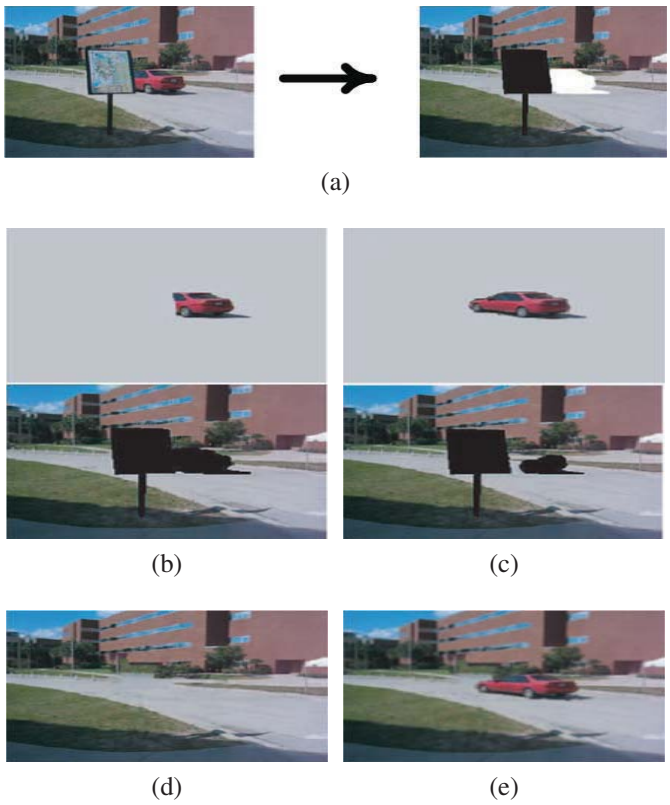


Figure 2.6: Steps of the motion layer segmentation based video completion. (a) Generating different motion layers. (b) All the motion layers of the video and marking the layer to be removed. (c) Synthesizing layers by means of the motion information of all the frames. (d) Remaining missing pixels are filled by the region completion method. (e) The completed layers are warped to generate the resulting frame.

3. Filling-in the missing area of each layer by means of motion compensation (Figure 2.6c).

4. Filling-in the remaining missing area by using the region completion approach (Figure 2.6d).

5. Warping the completed layers of the reference frame to every frame of the sequence to generate the final frames (Figure 2.6e).

The motion layer segmentation is based on the graph cut algorithm proposed in [33] to obtain the motion layers and explicitly the occluded pixels (refer to [32] and [33] for details). After applying the graph cut scheme and obtaining the motion layer segmentation and the occlusion information, the layer ordering is extracted which is assumed to be the same in the whole video sequence. It is worth noting that it is not necessary to perform this

step on all the frames only a few number of the first frames of the sequence can be used in step 1 of this video completion method.

Since each object belongs to a layer and the layer segmentation has been already performed, it is easy to remove the undesired objects by removing the corresponding layer. After removing the layer, all the other layers may have missing regions in some of the frames. In step 2 each incomplete layer the motion parameter is first found by a motion model. Then, for each layer, a compensated frame is generated, which is the result of all the warped frames together with their motion parameters. If there is still any uncompleted region, a graph cut based single image completion method is applied to fill-in the missing area similar to the other patch exemplar based methods. Similar to the other exemplar-based methods, this image completion is based on a non-parametric texture synthesis, therefore the priority of the target patch is very important. Here, the priority of the target patch is decided based on its number of available pixels. After the target patch, Ψ_t, is determined, a search for the source patch, Ψ_s, is done. The search range for the source patch is reduced to the locations near the previous found source due to the locality issue. Again, in this method, finding the source patch is based on the sum of squared differences (SSD) given in Eq. (2.30).

An interesting aspect of this video completion method is the graph cut algorithm applied to combine the found source patch, Ψ_s, and the target patch, Ψ_t, with overlapping area denoted by Ψ_o. Considering v_i as the location in the overlapping regions, and $C_t(i)$ and $C_s(i)$ as the color values in Ψ_s and Ψ_t respectively, the graph of the locations, v_i and v_j, with the edge weight of $W(v_i, v_j)$ is generated as follows:

$$W(v_j, v_j) = \begin{cases} \|C_t(v_i) - C_s(v_i)\| + & \text{if } v_i \text{ and } v_j \\ \|C_t(v_j) - C_s(v_j)\| & \text{are 4-adjacent} \\ & \text{neighbors} \\ \infty & \text{otherwise} \end{cases} \tag{2.44}$$

A small value of weight means that if the cut runs between the two locations, the resulting 4 color pairs, $C_t(v_i) - C_s(v_i)$, $C_s(v_i) - C_t(v_j)$, $C_t(v_i) - C_t(v_j)$, $C_s(v_i) - C_s(v_j)$, are very similar and lead to the least noticeable seam.

After the layer compensation and completion step, the synthesized layer of the reference frame is projected, using the layer motion parameters, to render each frame. This projection can be easily done by warping it into the corresponding position in the target frame since the motion parameters of the layers are computed with respect to the reference frame.

This layer compensation and modified graph cut based method preserves the consistency of the video frames very well, albeit the method relies heavily on an accurate motion layer segmentation.

Video Completion as a Global Optimization

One of the most effective video completion methods is the one introduced in [34], which treats the inpainting task as a global optimization with a well-defined objective function in the 3D volume of the frames. The objective function satisfies two important conditions: i) For every local space-time patch in the video sequence, there should be some similar patches in the remaining parts of the video, and ii) all the patches must be globally consistent with each other, both, spatially and temporally. In other words, the objective function is defined to rank the quality of the completion.

As mentioned earlier, a video is treated as a space-time volume. A pixel in the frame t is denoted by $p = (x, y, t)$, where (x, y) denotes the spatial position of the pixel. A small space-time fixed-sized window, W_p, around the pixel p is defined. It is clear that, in the 3D space, several windows can contain p. For example, W_p is centered around p, and W_p^i can be the ith window containing p, which is centered around the ith neighbor of p in the volume. Let the dataset D be the available parts of the video, S be the input video sequence, and $H \subset S$ be the hole region in the video sequence, one can say S is visually coherent with some other sequence D if every patch in S can be found somewhere in D. The goal is to complete the missing space-time region H with new data H^* such that the new resulting S^* is coherent with D. Therefore, the following objective function is defined, which should be minimized by the found S^*:

$$Coherence(S^*|D) = \Pi_{p \in S^*} \max_{q \in D} sim(W_p, V_q) \tag{2.45}$$

where windows in the data set D are denoted by V and p, q, run over all the space-time

points. The patches used here are of size $5 \times 5 \times 5$. $Sim(.,.)$ is a function to measure the similarity between two space-time patches. A typical measure to find the similarity between two patches is SSD. Since SSD does not consider the temporal information, it is modified in this method to produce a better similarity measure.

In the similarity measure modification, two parameters are considered in addition to the space-time R, G, B values of each pixel. First, at each point the spatial and temporal derivatives (Y_x,Y_y,Y_t) are computed. Then, $u = Y_t/Y_x$ and $v = Y_t/Y_y$ are defined to capture the temporal motion of each point along x and y directions, respectively. Therefore, a five-dimensional representation is used to express each point, both spatially and temporally, (R, G, B, u, v). The SSD, as $d(W_p, V_q)$, is applied to find the distance between two windows W_p and V_q, employing the 5D values for each point in each window. Finally, the distance measure is converted to a similarity measure using:

$$sim(W_p, V_q) = e^{-\frac{d(W_p, V_q)}{2\sigma^2}} \tag{2.46}$$

The value of σ controls the smoothness of the induced error so it is carefully chosen as the 75% of all distances in the current search in all locations.

So far, the well-behaved objective function is defined. Now, the optimization is done using this objective function. The optimization must satisfy two conditions for any space-time point p:

1. All the windows $W_p^1 \dots W_p^k$ containing p appear in the data set D: $\exists V^i \in D, W_p^i = V^i$

2. All those $V^1 \dots V^k$ agree on the color value c at the location p: $c = V^i(p) = V^j(p)$

The first condition, which is the reliability of each W_p^i, is satisfied by the lower obtained distance $d(W_p^i, V^i)$. In fact, $sim(W_p^i, V^i)$ measures the degree of reliability of the patch W_p^i. The value of color c at the point p should minimize the variance of the colors, $c^1 \dots c^k$, obtained by $V^1 \dots V^k$ at p. In other words, the color value at p should minimize $\sum_i s_p^i (c - c^i)^2$ in order to satisfy the second condition of the optimization, where $s_p^i = sim(W_p^i, V^i)$. Therefore, the best value for c is obtained by:

Algorithm 2.4 Global optimization-based video completion.

Input: video S, hole $H \subset S$, data set D
$t \longleftarrow 0$, $S^t \longleftarrow S$
repeat
  for all $p \in H$ **do**
    Let $\{W_p^i\}_{i=1}^k$ be all the windows containing $p \in W_p^i$
    Find $\{V^i\} \subseteq D$ maximizing Eq.(2.45)
    Let $c^i \in V^i$ be the appropriate colors.
    Set $\omega_p^i = \alpha_p^i.sim(W_p^i, V^i)$.
    $S^{t+1}(p) \longleftarrow c^i$ using Eq.(2.48)
  End for
  $t \longleftarrow t + 1$
until convergence
Output: S^t

$$c = \frac{\sum_i s_p^i c_i}{\sum_i s_p^i} \tag{2.47}$$

This color value calculation is modified in [9] to find a more likely reliable color value at point p. In [9], the quantity α_p is associated, as a weight, to each point which determines the confidence value of the point p. This weight ensurers that the total error inside the hole is less than the total error on the hole boundary; $\alpha_p = \gamma^{-dist}$, where $dist$ is the distance transform and $\gamma = 1.3$. Then, the color value is determined by:

$$c = \frac{\sum_i w_p^i c^i}{\sum_i w_p^i} \tag{2.48}$$

where $\omega_p^i = \alpha_p^i.s_p^i$. The whole optimization iteration is summarized in Algorithm 2.4.

Although, as expected, the method is computationally very complex and time consuming, the video completion results of this method reveal its high performance [9] [34], which made it as one of the most popular techniques in the field of video completion.

Tang's Video Completion via Maintaining Spatiotemporal Continuity

A technique that performs video completion in the motion and spatial domains separately was proposed in [35]. The method takes advantage of a global and local motion estimation scheme quite well.

In the first step, for each frame a motion map is constructed. The motion map is obtained by means of global motion estimation (GME) and local motion estimation (LME). The applied GME is an enhanced version of the Lucas-Kanade optical flow computation technique [36]. The GME results are initial motions for a proposed correlation-based LME algorithm. The final resulting motion vectors produce the motion map for each frame. Since the method is specialized for digitized vintage films that normally suffer from a lack of constant illumination, intensity normalization is performed as a preprocessing step before motion map construction. Then the video completion is performed in two steps: motion completion and frame completion. Motion completion is a task similar to Exemplar-Based with Ghost Shadow Removal video completion discussed in Section 2.1.2. But, instead of patch pasting, the motion vectors of the source patch are assigned to the border patch. The priority of each patch $\Psi_{(p,t)}$ to be filled-in is computed by:

$$P(p,t) = C(p,t) \times D(p,t) \times W(F_t) \tag{2.49}$$

where the confidence C and data D values of a patch in frame t centered at p are computed using equations Eq. (2.28) and Eq. (2.33), respectively. A new parameter called weighting factor $W(F_t)$ is used in the priority computation that measures the percentage of the source area available in each frame. A higher value of $W(F_t)$ indicates that the frame F_t has more source data, thus it has a higher priority.

Patch searching is carried out for the 3D-patch template, $\Gamma_{(\hat{p},t)}$ of $\Psi_{(\hat{p},t)}$. The following equation is used to find a patch with the least distance to $\Gamma_{(\hat{p},t)}$:

$$\begin{aligned} d(\Gamma_{(\hat{p},t)}, \Gamma_{(\hat{q},t)})) = SSD(\Gamma_{(\hat{p},t)}, \Gamma_{(\hat{q},t)})) \times \\ \max(1, \textstyle\sum_{(q,\acute{t})\in(\Gamma_{(\hat{q},t)}\cap\Phi_{\varepsilon})} c) + fd(t,\acute{t}), \end{aligned} \tag{2.50}$$

where $fd(t,\acute{t})$ is the temporal distance.

Once the best match is found, its motion vectors, already determined in motion map construction step, are assigned to the missing patch ($\Psi_{(\hat{p},t)}$). Since the missing areas have motion information, they can be tracked in the neighboring frames.

In the frame completion stage, the possibly same patches in the neighboring frames are stacked to produce a space-time volume. Then the available data in each patch in the

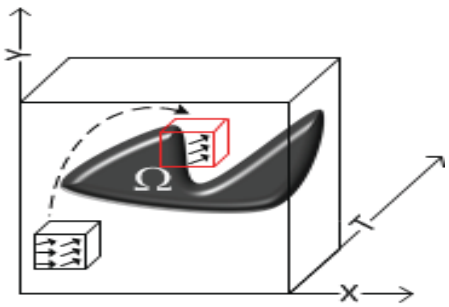


Figure 2.7: Motion filed transfer scheme.

volume is propagated to the corresponding missing pixel in the volume. This technique shows good completion results and visually pleasant. The disadvantage of this method is the applied preprocessing motion estimation technique, which is quite computationally complex.

Completion by Motion Field Transfer

Unlike most of the exemplar-based methods, the approach introduced in [37] fills the missing parts of the video by sampling the spatio-temporal patches of local motions instead of color and/or intensity values. Color propagation is done only after the estimation of the missing motion information of the inpainting region. In fact, the method is inspired by the video completion approach introduced in Section 2.1.2, but performs sampling of motion vectors instead of color and intensity (Figure 2.7). It assumes that the motion information is sufficient to fill missing parts in a video sequence. The entire completion process is summarized in three main steps: local motion estimation, motion field transfer, and color propagation.

Local motion field is estimated using Lucas-Kanade optical flow computation method [36]. Each motion vector $(u, v)^T$ is estimated by minimizing the following error function:

$$\arg\min_{(u,v)} \sum_{x,y,t} \left(u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} \right), \tag{2.51}$$

where $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are image derivatives along spatial and temporal directions. The motion information of each point $p = (x, y, t)^T$ in the video is represented by $(u(p), v(p))^T$.

In the next step, the algorithm searches for the most similar source patch given the target patch. Then, the source patch motion vectors are assigned to the missing motion vectors of

the target patch. A search for the optimal source patch $\hat{P}_s$ is obtained by:

$$\hat{P}_s(\hat{x}_s) = \arg\min_{P_s(x_s)} d(P_s(x_s), P_t(x_t)), \tag{2.52}$$

where P_s and P_t represent a source patch and a target patch, respectively. The position of source and target patch are denoted by X_s and X_p respectively. The dissimilarity measure between P_s and P_t is calculated as follows:

$$d(P_s(x_s), P_t(x_t)) = \frac{1}{|\mathcal{D}|} \sum_{p \in \mathcal{D}} d_m(m(p + x_s), m(p + x_t)), \tag{2.53}$$

where $\mathcal{D}$ represents the set of available (valid) pixels in the target patch and $|\mathcal{D}|$ denotes its cardinality, p represents the relative position from the center of each patch. As the 2D motion vector can be seen as a 3D vector considering the temporal information, the motion vectors are defined as $\boldsymbol{m} = (ut, vt, t)^T$. The distance between two 3D motion vectors is calculated by the angular difference of the vectors:

$$d_m(\boldsymbol{m}_0, \boldsymbol{m}_1) = 1 - \frac{\boldsymbol{m}_0.\boldsymbol{m}_1}{\|\boldsymbol{m}_0\| \|\boldsymbol{m}_1\|} = 1 - \cos\theta, \tag{2.54}$$

where θ is considered as the angle between the two motion vectors. Hence, an optimal choice of the source patch is the one that minimizes Eq. (2.52). It is worth noting that the order of filling is determined by the number of available pixels in a target patch. A higher number of non-hole pixels in a target patch means a higher priority to fill-in.

The assigned motion vectors to the missing pixels indicate the relationship between missing pixels with their neighbors. The motion vectors of each spatio-temporally adjacent pixel can be considered as undirected edges in an adjacency graph. The weight factor ω is assigned to each edge:

$$\omega(p, q) = r(p, , q)s(p, q), \tag{2.55}$$

where $r(p, , q)$ is the reliability value for the edge and it is calculated by the inverse of the dissimilarity measure defined in Eq. (2.54). The edge originating from the pixel p may point to a fractional location in the adjacent frame. Similarly, a point q in the previous frame may be connected to a fractional location in pixel p. Therefore, the size of the overlapping areas of these two pixels is used as $s(p, q)$ in the calculation of the edge weight. The color

of the pixel p is calculated as:

$$c(p) = \frac{\sum_{q \in \mathcal{N}} \ w(p,q)c(q)}{\sum_{q \in \mathcal{N}} \ w(p,q)}. \tag{2.56}$$

In fact, the color value $c(p)$ of the pixel p is a weighted average of the color values at the pixels q which belong to the set $\mathcal{N}$ of connected neighboring pixels.

Suppose there are n hole pixels. For each pixel $p_i; i = 1, ..., n$, Eq. (2.56) is obtained. Therefore, in the case that there are m boundary pixels $\{p_j^b : j = 1, ..., m\}$ with known color values, the following linear system is formed using the n equations:

$$C = [W|W_b] \begin{bmatrix} C \\ C_b \end{bmatrix}, \tag{2.57}$$

where $C = [c(p_1), ..., c(p_n)]^T$ is a $3 \times n$ matrix, $C_b = [c(p_{b_1}), ..., c(p_{b_m})]^T$ is a $3 \times m$ matrix. W and W_b are $n \times n$ and $n \times m$ matrices defined as:

$$W = \begin{pmatrix} 0 & w_{12} & \dots & w_{1n} \\ w_{21} & 0 & \dots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & 0 \end{pmatrix},$$

$$W_b = \begin{pmatrix} w_{11}^b & w_{12}^b & \dots & w_{1m}^b \\ w_{21}^b & w_{22}^b & \dots & w_{2m}^b \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1}^b & w_{n2}^b & \dots & w_{nm}^b \end{pmatrix},$$

where w_{ij} denotes the normalized weight factor. The values in C are obtained by rewriting (2.57) as:

$$C = (I - W)^{-1} W_b C_b, \tag{2.58}$$

where I is an $n \times n$ identity matrix.

This motion field transfer strategy in video completion works based on the assumption that the object motion is continuous in the video. Therefore, the motion information is sufficient to fill the holes. This is particulary effective when dealing with periodic motions like a walking person and can be less likely practical in completion of stationary objects and backgrounds.

Homography Blending, Sampling and Alignment in Completion

Background and foreground completion is performed independently in the method introduced in [38]. The background completion is done with the image inpainting method introduced in [39]. However, the image inpainting approach is extended by using layer segmentation and homography blending in a way to preserve temporal consistency. The foreground completion is performed in a two-step scheme: First, motion vectors are sampled and regularized by 3D tensor voting in order to maintain temporal coherence and motion periodicity. Second, the sampled motion data are aligned spatially and temporally to infer the missing moving pixels.

The background completion consists of four steps: Layer propagation, generating the layered mosaics, frame repairing, and homography blending. In the first step, layer information is propagated to the entire video by means of the mean shift algorithm. In a video frame, a layer is considered as a 3D image patch with similar features. The background is segmented into similar layers with depth ordering. Therefore, each layer has an optical flow which helps achieve a reasonable temporal consistency when the scene is complex. Afterwards, the different layers in the video are stitched together in order to make the layer mosaics. Layer mosaic generation is performed automatically by using a phase correlation [40] to produce a good translation matrix. Once the layered mosaics of the background video are aligned and constructed, large holes may exist in the background. The image inpainting method introduced in [39] is directly used on each frame. But, projecting the layer on the reference view produces inconsistency in the layer boundaries. Therefore, a homography blending approach is applied as presented in Algorithm 2.5 to remove the flickering effect at the boundary locations.

The foreground completion consists of two steps: sampling and alignment. In the sampling phase, motion data are sampled and then regularized to preserve the temporal coherence. In the alignment step, the missing pixels are inferred by spatio-temporal alignment of the sampled pixels.

The periodic motion is sampled in the movel sampling stage. A *sample movel* is a video that contains at least one cycle of a periodic motion. If there is a damaged frame in

Algorithm 2.5 Homography blending algorithm.

1: A layered reference mosaic K is made by choosing a reference frame and then all the mosaics constructed for all the layers are warped to it while each layer has its own optical flow.
2: Suppose M_1 and M_2 are respectively foreground and background layers in the video with an overlapping region M_3 as shown in Fig. 2.8. The homographies H_1^{-1} and H_2^{-1} are to warp $M_1 \bigcup M_3$ and $M_2 \bigcup M_3$ with regards to the reference mosaic K. M_3 is produced by the blend function $H_3 = \alpha H_1 + (1 - \alpha) H_2$, where α is the blending coefficient.
3: The repaired frame is warped back by projecting the mosaic K to layer M_i using the transformation matrix $H_i, i = 1, 2, 3$.

the movel then it is called *damaged movel*. The moving pixels are detected first by learning the background. Then, the connected components are constructed for the moving pixels. Next, a video mask is produced, which is used to sample a movel. The first and last frames of the sample movel must be the same to avoid "pop-up" effect. Therefore, the movels need to be warped up. The first 5 frames and the last 5 frames of the movel are chosen. Then, the 2D boundaries of the character in all these frames are found using the video mask. The set of boundary locations in all these frames is used as the input to 3D tensor voting [39] to infer the in-between boundary locations. Finally, a morphing method is applied to infer the color of the found pixels in the in-between frames. The movels are regularized afterwards to preserve the temporal consistency. The centroid of each frame of the movel is found individually so the corresponding path along their axes is not smooth. In order to make them coherent, the 3D tensor voting is used again to smooth the trajectory in the spatio-temporal domain. In the final step of the sampling process, the movel is normalized. In fact, the pixels in each frame are translated in a way that each centroid moves to the center of the frame. This yields a faster convergence in the alignment process and also the total number of voxels to be processed in the moved alignment can be reduced.

The alignment phase is carried out by employing a homography transformation function

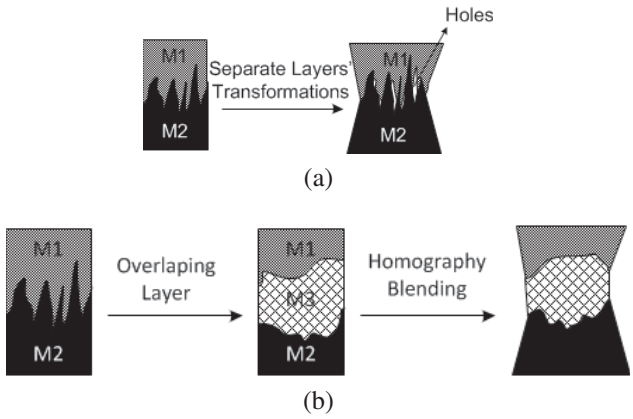


Figure 2.8: Homography blending. (a) Remaining small holes after layer projection. (b) Homography blending by making a new overlapping layer.

in the spatio-temporal space:

$$\begin{pmatrix} \acute{x} \\ \acute{y} \\ \acute{t} \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} h_0 & h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 & h_7 \\ h_8 & h_9 & h_{10} & h_{11} \\ h_{12} & h_{13} & h_{14} & h_{15} \end{pmatrix}}_{H} \begin{pmatrix} x \\ y \\ t \\ 1 \end{pmatrix} \tag{2.59}$$

where (x, y, t) and $(\acute{x}, \acute{y}, \acute{t})$ represent the sample and aligned movel coordinates respectively. H is the transformation matrix which relates the sample and aligned movels. The problem is now the estimation of $h = h_k, 0 \leq k \leq 15$. Depending on the type of transformation, the matrix can be further reduced. For example if the transformation involves only a translation, the upper 3×3 matrix turns into an identity matrix. The intensity squared errors for the pixels of the aligned movel must be minimized. Given I is the damaged movel, and $\acute{I}$ is the aligned sample movel, the error term in the overlapping volume between I and $\acute{I}$ is defined as:

$$E = \sum [\acute{I}(\acute{x}, \acute{y}, \acute{z}) - I(x, y, z)]^2 \tag{2.60}$$

The minimization is performed by Levenberg-Marquardt iterative algorithm. For each voxel, the intensity gradient is computed $(\frac{\partial \acute{I}}{\acute{x}}, \frac{\partial \acute{I}}{\acute{y}}, \frac{\partial \acute{I}}{\acute{z}})^T$. A Hessian matrix $A = [a_{kl}]$ and a weighted gradient vector b are computed as well:

$$a_{kl} = \sum \frac{\partial e}{\partial h_k} \frac{\partial e}{\partial h_l} \tag{2.61}$$

Algorithm 2.6 Voxel alignment algorithm.

1: (x_i, y_i, t_i) is computed using (2.59).
2: E is computed using (2.60).
3: Intensity gradient is computed $(\frac{\partial \acute{I}}{\partial \acute{x}}, \frac{\partial \acute{I}}{\partial \acute{y}}, \frac{\partial \acute{I}}{\partial \acute{z}})^T$.
4: Considering $e_i = \acute{I}(\acute{x}, \acute{y}) - I(x, y)$, the partial derivative of e_i with respect to $h_k, 0 \leq k \leq 15$ is computed:

$$\frac{\partial e_i}{\partial h_k} = \frac{\partial \acute{I}}{\partial \acute{x}} \frac{\partial \acute{x}}{\partial h_k} + \frac{\partial \acute{I}}{\partial \acute{y}} \frac{\partial \acute{y}}{\partial h_k} + \frac{\partial \acute{I}}{\partial \acute{z}} \frac{\partial \acute{z}}{\partial h_k} \tag{2.65}$$

5: A and b are computed using (2.65).
6: $h^{(m+1)}$ is computed using (2.63) and (2.64).
7: These steps are iterated until e becomes smaller than a threshold value.

$$b_k = -\sum e \frac{\partial e}{\partial h_k} \tag{2.62}$$

and h is updated by δh:

$$\delta h = (A + \alpha I)^{-1} b \tag{2.63}$$

$$h^{(m+1)} \leftarrow h^{(m)} + \delta h \tag{2.64}$$

Then, an iterative algorithm to perform the alignment for each voxel i at (x_i, y_i, t_i) is as in Algorithm 2.6.

This algorithm is performed on the 3D Gaussian pyramid form of the sample and damaged movel in order to start with a reasonable initialization and guarantee the convergence of H in fewer iterations.

Each of the background and foreground completion tasks is carried out individually. Then, the reconstructed foreground is placed in the restored background. It is worth noting that since the foreground completion is based on the periodic motions, it is restricted to only a subclass of camera and object motions. For instance, a rotated face cannot be sampled in a sample movel.

Posture Mapping and Analysis in Foreground Motion Completion

Object-based video completion can be considered as a specific case of exemplar-based techniques, which concentrate on filling-in the missing pixels of moving objects like a moving person. The object-based technique introduced in [41] [42] performs the completion task

by a virtual contour construction. Then, a key-posture selecting and mapping is performed. Finally, in the case of lack of proper postures in the video, synthetic posture generation is carried out. The method assumes that the object is already separated from the background and the completion is only performed to complete the missing regions of the object.

The virtual contours are constructed after object extraction. First, the video is sampled into 2D spatio-temporal slices at different Y and/or X values. This captures the horizontal and/or vertical trajectories of the moving object. Note that a motion that is not purely horizontal leads to various sizes for the object. Therefore, posture alignment and normalization is needed. In this case, the largest posture is chosen as the reference to align and normalize the other postures of the object. A correspondence between each contour point of the adjacent contours is generated, then the affine transformation parameters between the largest posture and the other postures are found using the least squares optimization scheme. Thus, all the postures are aligned and normalized with regard to the largest posture of the object. Before composing a virtual contour, the missing regions of the object trajectories must be restored. Therefore, the image completion technique introduced in Section 2.1.2 is performed directly on each already sampled 2D spatio-temporal slice. Afterwards, the Sobel edge detector is employed on each slice to find the boundary of the object's trajectory. Then, the completed slices are combined to make a virtual contour which is used later in a posture mapping and retrieval process. In fact, a virtual contour can provide quite precise information about the posture and the filling location of an occluded object.

After finding a sequence of virtual contours, they are used to match the most similar posture sequence in the available set of postures in the entire video. Therefore, a set of key postures that are the most representative postures need to be selected as the key postures. The postures need practical descriptors and a measure to find out if they are good matches. The descriptor and a dissimilarity measure is defined as in [43]. The silhouette of the posture is described by a set of feature points. Then, a circle centered at the feature point with radius r is considered to generate a local histogram. Figure 2.9 shows the steps to generate the local histogram for a posture. The circle is partitioned into N_{bin}, then the number of feature points in each partition is assigned to each histogram bin. Hence, for each posture a set of such histograms is used as the posture's descriptor. In order to match

(a) (b) (c) (d)

Figure 2.9: Extracting the local context of a posture. (a) The original object. (b) The silhouette of the object. (c) Significant points in the silhouette the object. (d) Local histogram of a significant feature point.

two different sampled points in two different postures, the following cost function is used:

$$F(a_i, c_i) = \frac{1}{2} \sum_{k=1}^{N_{bin}} \frac{[h_{a_i(k)} - h_{c_i(k)}]^2}{h_{a_i(k)} + h_{c_i(k)}}, \tag{2.66}$$

where a_i and c_i represent two sampled points and h_{a_i} and h_{a_i} denote the kth bin of their histograms. The best match can be found by minimizing the following matching function:

$$H(\pi) = \sum_j F(a_j, c_{\pi_{(j)}}), \tag{2.67}$$

where π is a permutation of $1, 2, ..., N_{bin}$. Finally the shape matching function in terms of a dissimilarity measure for two postures A and C with N_A and N_C number of sample points is defined as:

$$F_{sc}(A, C) = \frac{1}{N_A} \sum_i F(a_i, c_{\pi_i}) + \frac{1}{N_C} \sum_j F(a_j, c_\pi(j)). \tag{2.68}$$

A posture is selected as a key posture if its dissimilarity to all the other key postures is smaller than a threshold value $TH_{posture}$ set to 0.08. Each posture in the set of key postures is labeled with a unique number. Then, the virtual contour of each occluded posture is matched with the key posture that has the most similar context using Eq. (2.68). A special label is assigned to a virtual contour which does not have a match in the key posture set. This way, the whole problem is converted to a string matching in order to find a sequence of postures for a missing or occluded object. Given an input encoded segment, the objective is to look for the most similar substring in the long string of codes. After finding the proper substring, the associated postures to the string are used to fill-in the occluded part of the target object. All the aforementioned steps help preserve a reasonable spatial and temporal consistency and avoid any flickering effect. However, a lack of key postures can

result in a poor quality of the video reconstruction since there is not enough number of available postures in the entire video to generate an appropriate key-posture set. This can happen especially in the case when the object is occluded in the video for a large period of time. Therefore, the key-posture set needs to be enriched to cover more sophisticated motions and objects. The synthesis technique [41] divides a human body into three parts: the head, torso, and legs. Then it generates new postures using other postures. The posture synthesis scheme proposed in this method is very case-specific and specialized only for human body. Therefore, we avoid further discussion and refer the reader to [41] for more detailed information.

2.2 Bandlet Transform

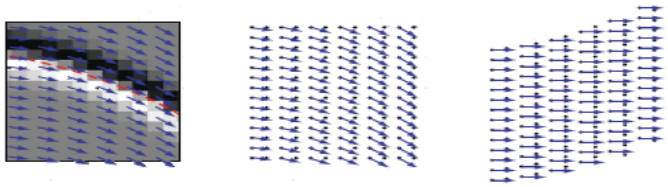
Wavelets are mathematical functions that cut up data into different frequency components, and then study each component with a resolution matched to its scale. Wavelets have advantages over traditional Fourier methods in analyzing physical situations in which the signal contains discontinuities. However, conventional wavelet bases are not optimal to approximate natural images due to their disability to take advantage of geometrical regularity of image structures. In fact, wavelets have a square support. So, they are not adapted to anisotropic regularity of geometrical elements including edges. Several new bases such as curvelets [44], contourlets [45] and wedgelets [46] have been proposed to take advantage of the anisotropic regularity of geometrical image structures. One of the most effective ones is bandlet transform [47].

An image can be differentiable in a direction parallel to the tangent of an edge curve even though, very likely, the image may be discontinuous across the contour. Fig. 2.11a shows the geometric flow in the direction of edges in the hat of Lenna. Such an anisotropic regularity is exploited by the bandlet transform which constructs orthogonal vectors that are elongated in the direction of the maximum regularity of the function. Hence, the bandlet transform is considered as an effective tool to capture the geometric properties of an image.

The first bandlet bases are introduced in [47], [48] having optimal approximation results

(a) (b)

Figure 2.10: (a) Geometric flows in the direction of edges. (b) Image geometric segmentation.



(a) (b) (c)

Figure 2.11: (a) Wavelet coefficients and geometric flow. (b) Geometric flow and sampling position. (c) Warped sampling. [10]

for geometrically regular functions. Then, the bandlet bases have been improved by a multi-scale geometry defined over the coefficients of wavelet bases [49], [50]. In the following subsections a brief overview of bandlets is provided. For a detailed explanation the reader is referred to [10].

Orthogonal Bandlets Approximation

The polynomial approximation by means of a thresholding in an orthogonal Alpert basis is computed for the bandlet approximation. The Alpert transform can be considered as a polynomial wavelet transform adapted to an irregular sampling grid. It is obtained by orthogonalizing multi-resolution space of polynomials defined on the irregular sampling grid. An example of such sampling grid is shown in Fig. 2.11c. The resulting vectors have vanishing moments on this irregular sampling grid, which is the basic need to approximate the warped wavelet coefficients. A few vectors from Alpert basis can efficiently approximate a vector corresponding to a sampling of a function with an anisotropic regularity. This kind of bandletization of wavelet coefficients is done by an Alpert transform defines a set of bandlet coefficients. These coefficients can be written as inner products $\langle f, b^k_{j,l,n} \rangle$ of the original image f with bandlet functions that are linear combinations of wavelet functions

$$b^k_{j,l,n}(x) = \sum_p a_{l,n}[p] \psi^k_{j,p}(x) \tag{2.69}$$

where the $a_{l,n}[p]$ are the coefficients of the Alpert transform. These coefficients depend on the local geometric flow, since this flow defines the warped sampling locations, as it is exemplified in Fig. 2.11c. The bandlet function is defined at some location n and wavelet scale 2^j. Another scale factor $2^l > 2^j$ is introduced by the Alpert transform which defines the elongation of the bandlet function. Also, the bandlet inherits the regularity of the mother wavelets $\psi^k_{j,p}$.

Geometric Flow Segmentation Approximation

The family of orthogonal bandlets depends on the local adapted flow over small squares for each scale 2^j and orientation k. This parallel flow is characterized by an integral curve, such as the depicted one as the dashed red plot in Fig.2.10a. Due to the bidimensional regularization performed by the smoothing of the wavelet $\psi^k_{j,p}$, i.e, $f_i = f * \psi^k_{j,p}$, this integral curve does not need to be strictly parallel to the contour.

One needs to segment the set of wavelet coefficients in squares S, in order to approximate the geometry by a polynomial flow. This segmentation is obtained for each scale 2^j and orientation k of the wavelet transform using a recursive subdivision in dyadic squares of various sizes. This subdivision results in a quadtree that specifies if a square S should be further subdivided in four sub-squares with twice smaller size or not. There is no geometric directional regularity to exploit, if there is no specific direction of regularity inside a square. This is the case either in uniformly regular regions or at the vicinity of edge junctions. Thus, it is not necessary to modify the wavelet basis. A sample of such quadtree segmentation is shown in Fig. 2.12. Obviously, only for the edge squares, the adaptive flow is required to be computed in order to produce the bandlet bases which exploit the anisotropic regularity of an image.

Through scales the geometric structures of an image evolves. Therefore, a different geometry Γ^k_j can be chosen for each scale 2^j and orientation k. The set of all geometries is noted as $\Gamma = \{\Gamma^k_j\}^k_j$ that consists of all the adapted flows of the quadtree segmentation squares.

In our work, due to the high complexity of quadtree segmentation, we employed a fixed size for all the squares instead of dynamically finding the size of each square. Fig. 2.10b

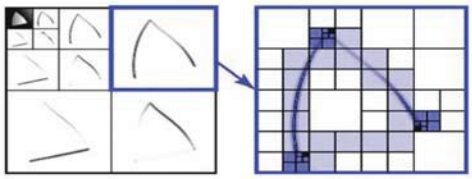


Figure 2.12: Example of quadtree segmentation on scales of the wavelet transform of an image [10].

shows the result of finding the geometric flow for each same-sized square of the quadtree, on the finest scale of the wavelet transform. We propose a solution for spatio-temporal inpainting by a bandlet-based regularization. The proposed regularization is in fact an optimization in the ℓ^1 norm of bandlet coefficients. The revealed geometry by bandlet transform is also employed in our research work in a patch fusion scheme to blend the matching pathes in an exemplar-based filling-in process.

Chapter 3

Bandlet-based Video Completion

3.1 Using Bandlets in Inpainting

The bandlet framework can achieve an effective geometric representation of texture images. It is essential in sparse regularization and spatial or spatio-temporal data reconstruction for digital inpainting purposes.

The image (spatial) inpainting problem may be formulated as follows. An image I contains a set of missing pixels indicated by Ω and a source ($\Phi = I \setminus \Omega$) area. The goal is finding an image $\acute{I}$ such that $\acute{I}(x)$ is equal to $I(x)$ for the pixels that belong to Φ, i.e., $\acute{I}(x) = I(x)\ \forall x \notin \Omega$ while the overall geometry of $\acute{I}$ has the same geometrical regularity as that of I in Φ. In the presence of additive noise ω we have the image f with missing pixels as $f = \theta I + \omega$ where

$$\theta I(x) = \begin{cases} 0 & \text{if } x \in \Omega \\ I(x) & \text{if } x \in \Phi. \end{cases} \tag{3.1}$$

A sparsity-based regularization solution for the inverse problem $f = \theta I + \omega$ was proposed in [51] as

$$\acute{I} = \arg\min_g \frac{1}{2}||f - \theta g||^2 + \lambda \sum_k |\langle g, \psi_k \rangle|. \tag{3.2}$$

This minimization has been used with the orthogonal wavelet bases ψ_k for denoising [51] where the value of λ is chosen based on the level of noise and can be set to 1 for a noise-free image. Considering the bandlets as anisotropic wavelets warped along the geometry flow,

we substitute the conventional wavelet bases of Eq. (3.2) with the bandlet bases introduced in Eq. (2.69) as

$$\acute{I} = \arg\min_g \frac{1}{2}||f - \theta g||^2 + \lambda \sum_{j,l,n,k} |\langle g, b^{\Gamma}_{k,j,l,n}\rangle|. \tag{3.3}$$

where similar to Eq. (2.69), k and j are the number of orientations and scales of the wavelets, and l, n are the sampling grid parameters in the Alpert transform employed in the bandlet transform. As discussed in the next section, our video inpainting scheme is subject to reconstructing the missing part of the frames generated due to occlusions and/or undesired object removal. Therefore, we avoid the noise level in the above equations (i.e., $\omega = 0$) and rewrite Eq. (3.3) as

$$\acute{I} = \arg\min_g \sum_{j,l,n,k} |\langle g, b^{\Gamma}_{k,j,l,n}\rangle|. \tag{3.4}$$

This equation is indeed minimizing the ℓ^1 norm of the bandlet image representation by which we achieve a solution for the spatial inpainting problem. In the next section, we utilize this idea to develop a 3D video volume regularization algorithm as well as the effectiveness of bandlets for blending the matching results of a best match search approach in the video completion task.

3.2 Spatio-temporal Video Completion

An important task of video completion is to fill in large missing regions produced by object occlusion or undesired object removal. The large missing region completion cannot be carried out well by simply applying PDE, regularization, or other interpolation based methods. On the other hand, in the exemplar-based methods, finding a reliable area around the missing parts and also finding a proper match in the source frames toward the end of the process reduces the accuracy of the results. Therefore, a video inpainting technique is proposed here that benefits from both an exemplar-based patch matching and a sparsity regularization scheme. The process starts looking for best candidates that match a patch Ψ_p on the border of the missing region. The N best retained matching patches in the whole sequence (Fig. 3.1) are then fused and the resulting data replaces the missing part of the border patch. In case there is no proper match for the border patch, i.e., $N = 0$, the border

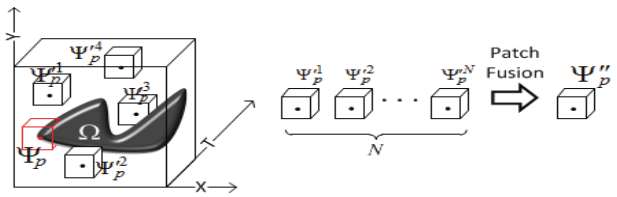


Figure 3.1: Fusion strategy of patch matching results in a 3D volume video. Ψ_p lies on the missing region border. $\acute{\Psi}_p^1$,...,$\acute{\Psi}_p^N$ are the N most similar patches to Ψ_p and Ψ_p'' is the patch fusion result.

patch is kept unchanged for a further process by the 3D video volume regularization to generate the final inpainting result.

A 3D patch centered at p on the border $\partial\Omega$ of the source Φ and missing Ω regions is denoted by Ψ_p as depicted by red in Fig. 1. We search for the best match of Ψ_p in the Φ of the whole frames. The best match $\acute{\Psi}_p$ is found using sum of squared differences (SSD)

$$\acute{\Psi}_p = \arg\min_{\Psi_q \in \Phi} SSD(\Psi_q, \Psi_p), \tag{3.5}$$

$$SSD(\Psi_q, \Psi_p) = \sum_{(x,y,t)} ||\Psi_p(x,y,t) - \Psi_q(x,y,t)||^2, \tag{3.6}$$

where for each RGB pixel located at (x, y) in the source region (Φ) of frame t we have a vector containing 5 elements (R, G, B, u, v). Considering (Y_x, Y_y, Y_t) as spatial and temporal derivatives of gray-scale video Y, $u = Y_t/Y_x$ and $v = Y_t/Y_y$ represent instantaneous motions in x and y directions respectively [9]. The motion information is involved in the space-time patch matching in order to preserve the motion consistency.

Unlike many of the exemplar-based methods, we do not simply replace the missing portion Ω of Ψ_p by the corresponding pixels in $\acute{\Psi}_p$. Instead, the best N matches $B_p = \{\acute{\Psi}_p^1, \acute{\Psi}_p^2, \acute{\Psi}_p^3, ..., \acute{\Psi}_p^N\}$ are fused using the bandlet transform as described in Section 3.2.1, then the fusing result pixels are copied into the missing part Ω of Ψ_p. The idea behind using several top similar patches instead of a single patch in image inpainting was presented in [52] and [53] by using nonlocal means and a linear blending of the patches spatially, respectively. The reason for employing a fusion framework in our video completion scheme stems from the fact that, for other border patches $\Psi_{\acute{p}}$ spatio-temporally near Ψ_p that have many pixels in common with Ψ_p the resulting set $B_{\acute{p}}$ would have many matching patches in common with B_p of Ψ_p. Therefore, their results of fusion can be very similar. Consequently, the results of inpainting for spatio-temporally close regions become reasonably

consistent both spatially and temporally.

The value of N is determined using a threshold value τ. If SSD of a patch $\acute{\Psi}_p$ and Ψ_p is lower than τ, B saves $\acute{\Psi}_p$. The value of τ should not be too large to filter out many patches and at the same time it should not be too small to keep so many of them. Based on our observations we choose 0.85 as a good value for this threshold. Also, N should not be too large to avoid unnecessary fusions. In our experiments N is limited to $N \leq 10$. It is worth noting that the number obtained for N indicates the degree of reliability of the best matching patches found for Ψ_p. A lower value of N means Ψ_p is not frequently repeated in the entire frames and consequently the obtained matches are not quite reliable for Ψ_p. This case happens frequently in inpainting of scenes captured by a static camera where the goal is reconstructing the missing region after a stationary object removal. Therefore, we leave a border patch Ψ_p intact once the length N of its B_p set is 0 (i.e., $\forall \Psi_q \in \Phi,\ SSD(\Psi_p, \Psi_q) > \tau$).

The priority of filling-in process is very important in the exemplar patch matching. We give the highest priority to a border patch Ψ_p that contains more reliable pixels, lies on the continuation of textures and also lies on the moving regions of the video comparing to other patches. The reliability of pixels in the border patch is measured by the confidence value given by

$$C(p) = \Bigg(\sum_{q \in \Psi_p \cap \Phi} C(q) \Bigg) / |\Psi_p|. \tag{3.7}$$

This parameter is adopted from [6] for the 3D patches, where $|\Psi_p|$ is the volume size of Ψ_p. In this equation and the equations that appear hereafter, $\Psi_p \cap \Phi$ indicates pixels of the border patch Ψ_p that lie in the source pixels Φ of the video. In the initialization, the confidence value is set to 1 for the pixels in the source region and 0 for the pixels in the missing area, i.e. $C(p) = 0\ \forall p \in \Omega$ and $C(p) = 1\ \forall p \in \Phi$. A patch centered at p on the border $\partial\Omega$ with already more filled-in pixels has a larger confidence than those of other patches. The number of edge pixels can be used to measure the structural information contained in the patch. This is obtained by means of the already computed spatial derivatives Y_x and Y_y. Suppose $\acute{Y}_x$ and $\acute{Y}_y$ represent 0-1 maps of thresholded horizontal and vertical derivatives of the entire frames, respectively. Instead of manually defining threshold values to generate

these two binary maps, Otsu's method can be used to find a proper threshold. Then, the structural data value of Ψ_p is defined as

$$D(p) = \Bigg(\sum_{q \in \Psi_p \cap \Phi} \acute{Y}_x(q) \vee \acute{Y}_y(q) \Bigg) / |\Psi_p|. \tag{3.8}$$

Similarly, $\acute{Y}_t$ that contains 0-1 maps of temporal derivatives is used to determine the motion data value of a border patch,

$$M(p) = \Bigg(\sum_{q \in \Psi_p \cap \Phi} \acute{Y}_t(q) \Bigg) / |\Psi_p|. \tag{3.9}$$

A high value of D means that the patch is placed on the continuation of a highly textured region. Also, a large value of M indicates a large number of moving pixels with large motion vectors in the border patch. The priority of a border patch is obtained as follows

$$P(p) = C(p) \times D(p) \times M(p). \tag{3.10}$$

A border patch Ψ_p with the highest $P(p)$ is chosen from the whole frames to be filled-in first. Once the patch matching is carried out, the confidence value is updated as $\acute{C}(p) = \alpha C(p)$ where $0 < \alpha < 1$. The derivative matrices $\acute{Y}_x$, $\acute{Y}_y$ and $\acute{Y}_t$ are also updated by copying the derivative values of $\acute{\Psi}_p$ into the corresponding locations in $\Psi_p \cap \Omega$. Then the process is repeated for a new highest priority border patch until there is no border patch unprocessed.

The resulting video sequence containing unfixed regions (i.e. those with unreliable matches) are passed to the sparsity regularization inpainting stage for further processes as discussed in Section 3.2.2.

3.2.1 Patch Fusion

Multi-scale decomposition (MSD) based image fusion schemes, especially wavelet-based ones, have a great performance compared to regular methods [54]. However, as discussed in Section 3.1, due to its capability to capture more complicated geometric flows and structural information in images, the bandlet transform is much more appropriate than wavelet transform for analysis and synthesis of edges and textures [55]. Hence, we design a fusion scheme based on bandlets to blend the best patch search results.

Figure 3.2 shows the proposed image fusion scheme. Consider I_1 to I_M as M images of a single scene captured from M different sources (e.g., cameras, sensors, etc.), the bandlet transform is applied on each I_i to obtain the geometric features Γ_i in the form of real numbers and bandlet coefficients C of each image. Now we need to generate a fused set of geometry flows and bandlet coefficients.

The fused geometry flow set Γ_F is computed as follows

$$\Gamma_F = \big(\sum_{i=1}^{M} j_i \Gamma_i\big) \big/ \big(\sum_{i=1}^{M} j_i\big), \tag{3.11}$$

where j_i is 0 if mean μ_i of the values of Γ_i is lower than a threshold σ. The value of σ is chosen as the mean of all $\mu_1, \mu_2, ..., \mu_M$. Indeed, this thresholding leads to applying only the highly structurally similar source images to produce the fused geometry. The most similar Γ of the source images are selected and their mean value generates Γ_F. The fused bandlet coefficients' set is calculated as

$$C_F = \big(\sum_{i=1}^{M} C_i\big)/M. \tag{3.12}$$

It is worth mentioning that the bandlet coefficients C and the geometric features Γ are produced for l, n, j, k scales and orientations of Eq. (2.69).

The inverse bandlet transform is performed on Γ_F and C_F in order to generate the fused image from the M source images. Fig. 3.3(d) shows an example of the bandlet based fusion result for 3 source images, where Barbara's image is manually blurred and the resulting images are considered as the source images depicted in Fig. 3.3(a)–(c).

Now consider the set B_p of the N best matching patches obtained for Ψ_p in the proposed video inpainting technique in Section 3.2. Each $\acute{\Psi}_p^i$ of B_p has a size of $X \times Y \times T$. The corresponding spatial planes of patches in B_p are fused using the aforementioned fusion

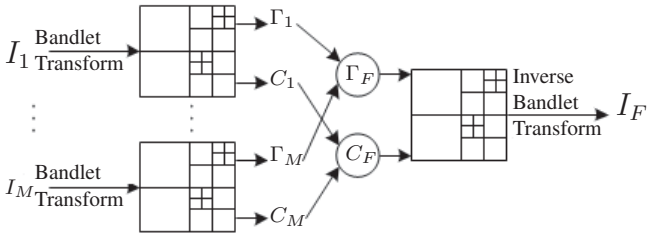


Figure 3.2: Bandlet-based fusion framework for M source images.

(a) (b) (c) (d)

Figure 3.3: Fusion result for 3 different source images of Barbara. a-c) Source images. d) Resulting fused image.

method to produce the resulting inpainting patch Ψ''_p, i.e.,

$$\begin{aligned}
\Psi''_p(t_1) &= \text{fuse}\left(\Psi'^1_p(t_1), \Psi'^2_p(t_1), ..., \Psi'^N_p(t_1)\right) \\
&\vdots \\
\Psi''_p(t_i) &= \text{fuse}\left(\Psi'^1_p(t_i), \Psi'^2_p(t_i), ..., \Psi'^N_p(t_i)\right) \\
&\vdots \\
\Psi''_p(t_T) &= \text{fuse}\left(\Psi'^1_p(t_T), \Psi'^2_p(t_T), ..., \Psi'^N_p(t_T)\right)
\end{aligned} \tag{3.13}$$

where $\Psi_p(t_i)$ represents all the $X \times Y$ pixels at time index t_i ($1 \leq t_i \leq T$) in the patch Ψ_p. This fusion scheme takes more structural information into account than simply copying the source (Φ) pixels of the best match $\acute{\Psi}^1_p$ to produce the final inpainting result. Besides, as mentioned earlier such patch fusion strategy followed the introduced search process, helps gain more visual consistency.

3.2.2 Spatio-temporal Regularization Using Bandlets

As a result of the N best patch matching strategy, the unreliable border patches (i.e. those that less likely have a match in the whole sequence or those less frequently are repeated in the frames) are recognized by the inpainting system. These kinds of patches

Algorithm 3.1 Bandlet-based 3D video volume inpainting.

1: $i = 0$ and $V^{i=0} = y$
2: **while** $|V^{(i+1)} - V^{(i)}| > \varepsilon$ **do**
3: Find $\acute{V}^i$ using Eq. (3.15)
4: **for** $z = 1 \rightarrow X \times Y \times T$ **do** {Update the estimate; $V^{i+1} = T_B(\acute{V}^i)$ }
5: Bandlet transform on $\acute{V}_z^i$
6: Soft-thresholding Eq. (3.16) on $\acute{V}_z^i$ bandlet coefficients
7: Generate V_z^{i+1} by inverse bandlet transform
8: **end for**
9: $i \leftarrow i + 1$
10: **end while**

remain unchanged in the first inpainting stage and are passed to the 3D regularization procedure introduced in the following paragraphs.

Considering the 2D minimization problem introduced in Eq. (3.4) as an exhaustive optimization, we adopt the *soft-theresholding* algorithm which has been used as a solution for multi-scale wavelet representation inverse problems such as denoising [56].

The overall geometry is supposed to be fixed for an estimate of the original video. The soft-theresholding function is carried out iteratively for the minimization of Eq. (3.4) for each plane in the 3D volume video. At each iteration, the estimate video V^{i+1} is updated as follows

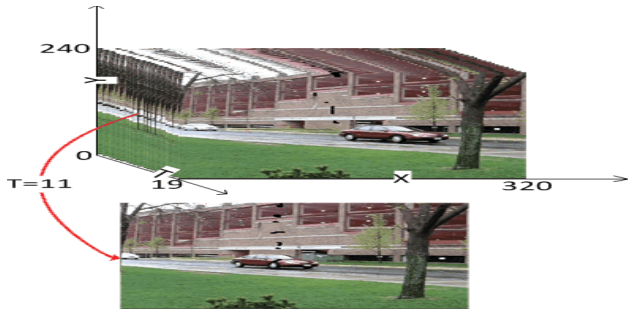
$$V^{i+1} = T_B(\acute{V}^i), \tag{3.14}$$

$$\acute{V}^i = \begin{cases} V^i(x) & \text{if } x \in \Omega \\ y(x) & \text{if } x \in \Phi. \end{cases} \tag{3.15}$$

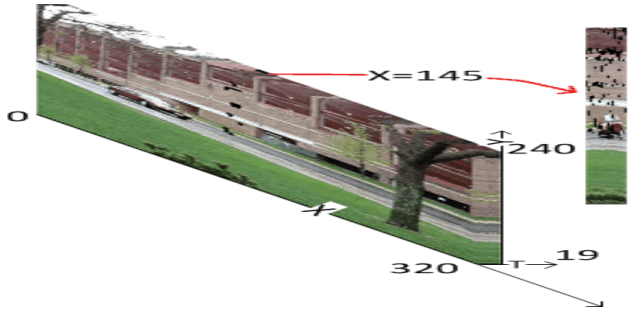
Pixels of the original video volume are represented by $y(x)$ in the above equation. T_B denotes the soft-thresholding function performed in the bandlet domain for each existing plane in $\acute{V}^i$ defined as

$$T_B(f_z) = \sum_{j,l,n,k} t_\lambda(\langle f_z, b_{j,l,n,k} \rangle).b_{j,l,n,k} \ , \tag{3.16}$$

where f_z denotes each existing plane in the video volume. For a 3D volume consisting of T frames of $X \times Y$ pixels, we consider T planes along the time, X planes along horizontal and Y planes along vertical directions. $t_\lambda(x) = \max(0, 1 - \frac{\lambda}{|x|})x$ and the value of λ goes to 0 as the iteration number increases. $b_{j,l,n,k}$ represents the bandlet functions of various scales and orientations as in Eq. (2.69).



(a)



(b)

Figure 3.4: A damaged video volume from different views. a) X-Y planes view. b) T-Y planes view.

(a) (b) (c) (d) (e) (f) (g)

Figure 3.5: Various iteration results of Algorithm 1 on the 11^{th} frame of the video of Fig. 4.7. For a better illustration the images are cropped from left, right and bottom.

Algorithm 3.1 presents the details of the minimization procedure to inpaint a video volume. This algorithm stops once the difference between two consecutive estimates is less than a small value ε. One may think of applying this algorithm on each frame independently as the inpainting task. Obviously, in a video sequence the flow of motions and trajectories is very important and needs to be considered in the inpainting task to preserve the consistency. Figure 4.7(a) displays the resulting video of the exemplar-based repair stage done on the original video of Fig. 3.7(c). This video contains black holes representing unfixed patches. Rotating the video volume around the Y axis, one can see the video volume T-Y planes. As seen for example in the T-Y plane of $X = 145$ in Fig. 4.7(b), pixels of the missing region do not only lie on the spatial geometric flows but also those along the time direction. As a consequence, in each iteration of Algorithm 3.1, the regularization is carried out on

(a) (b) (c)

Figure 3.6: The 2-stage proposed video completion method shown for a sample frame. (a) Original frame. (b) Stage 1 result: Exemplar-based patch fusion step (Sec. 3.2.1). (c) Stage 2 result: bandlet-based regularization on the result of stage 1 (Sec. 3.2.2).

planes X-Y, T-X and T-Y denoted by $\acute{V}_z^i$. Due to limitations of a 3D illustration, the inpainting result for only a single frame is shown in Fig. 4.8.

3.3 Experimental Results

Several video sequences, including some that are provided in [29], [26] are used to evaluate the proposed video inpainting method. This set of videos contains sequences captured by both static and moving camera. The resolution of each video sequence is 320×240. The intermediate results of the proposed two-stage video completion technique performed on a sample video sequence for one of its frames are presented in Fig. 3.6. In the implementation, the following settings are used:

- The size of each patch is $9\times9\times5$ in the patch matching process.
- α is set to 0.5 for confidence update.
- τ is set to 0.85 to choose the N top matching patches.
- Gray-scale values of the RGB frames are found by $(R+G+B)/3$ whenever needed like instantaneous motion calculation.
- Considering a border patch Ψ_p centered at $p=(x,y,t)$, the search range is reduced to $x-50<x<x+50$, $y-50<y<y+50$ and $t-7<t<t+7$ in the video sequence in order to avoid unnecessary search. This does not negatively affect the patch matching result, since most likely the best patches for an arbitrary patch exist in its adjacent space and time locations.

The details of the bandlet transform applied in our technique are as follows:

- Number of scales j, on which geometry is computed, is set to 3.

- The introduced scale factor l by Alpert transform in the bandletization (Section 3.1) is set to 4.
- Orthogonal wavelets are used in the bandetization.
- In the wavelet transform, Daubechies wavelets are employed.
- A fixed size 8×8 segmentation is employed instead of the complex dyadic segmentation introduced in Section 3.1.

Fig. 3.7 depicts the results of our video inpainting scheme on different sequences. These videos are selected from TV, video games, and also captured by a digital camera. The objective in the sequence of Fig. 3.7(a) is to remove the stationary object and fill-in its missing region with proper data. Since the camera and the removed object are static, as discussed in Section 3.2, there is not much information about what was behind the object in the whole sequence. Therefore, the inpainting result is mostly produced by 3D regularization rather than patch matching. Other examples illustrated in Fig. 3.7 depict inpainting results of videos containing camera motions. In all cases, the proposed method performs the completion task quite well. In order to gain insight into the effect of each step of the proposed video completion scheme, several analyses are next presented as well as a comparison with two state-of-the-art methods.

3.3.1 Effects of Patch Fusion and 3D Regularization

As mentioned before, the N best patch sorting and fusion results in a better performance in comparison to conventional patch replacement. We show this by means of a quantitative comparison.

A manual damage is generated on an original video sequence. Then, the damaged video is completed by the spatio-temporal video completion approach presented in Sec. 3.2. The completion is performed once without patch fusion, i.e, replacing the missing parts of a border patch by the corresponding pixels of the best matching patch. The spatio-temporal completion is carried-out once again by applying the introduced patch fusion technique. However, since the second stage of our proposed method (i.e., 3D regularization) is not applied in this experiment, we simply avoid the threshold τ (used to find N) and set $N = 5$. Then, for both cases, the difference of the completion result of the damaged video and the

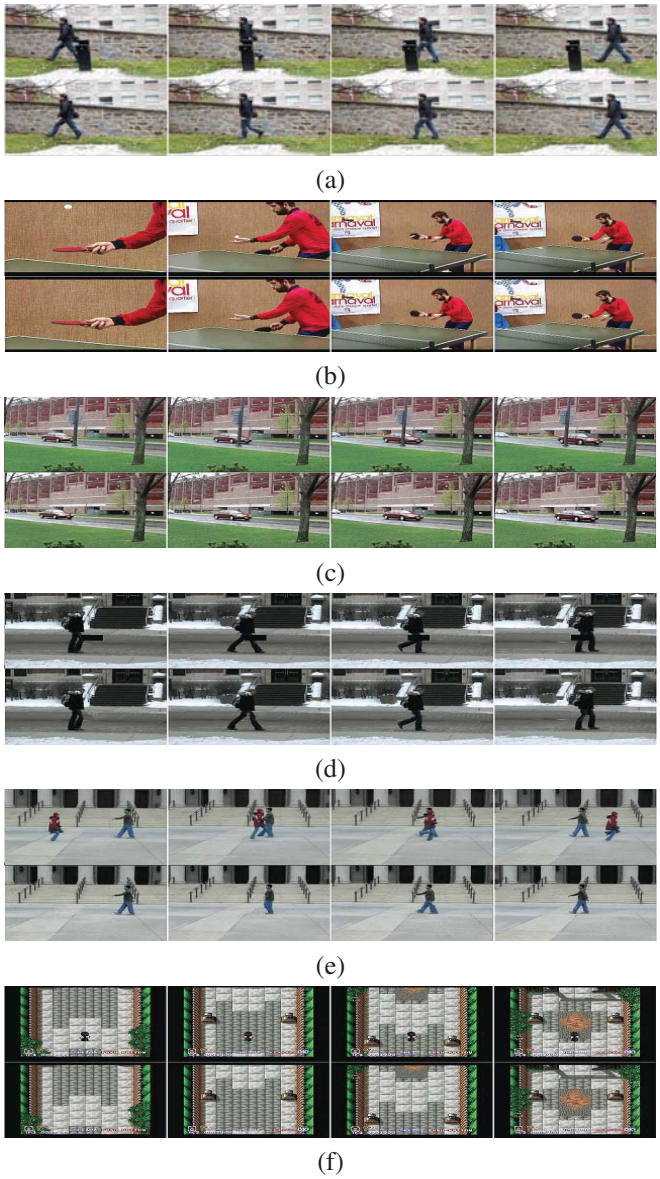


Figure 3.7: Completion results for different video sequences. In each sub-figure, the top row shows the original frames and the bottom row demonstrates the corresponding inpainting results.

original video sequence is observed by computing the MSE value for the corresponding frames of the original and the completion result video sequences. Fig. 4.11(a) shows a frame of the video chosen for evaluation which is damaged as in Fig. 4.11(b) and then completed as in Fig. 4.11(c) and Fig. 4.11(d).

The plot indicated as "Exemplar Bandlet-based Patch Fusion" in Fig. 4.12 shows mean square error (MSE) graph of all the 50 frames of the original video and the spatio-temporal completion result sequence using the bandlet based patch fusion. Obviously, the MSE

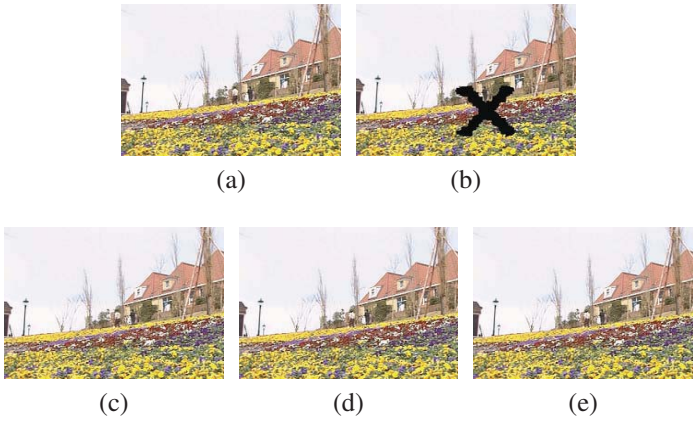


Figure 3.8: (a) Original frame. (b) Damaged frame. (c) Regular exemplar-based inpainting result (Frame number 13, MSE=19.13). (d) Patch fusion exemplar-based inpainting result (Frame number 13, MSE=18.4). (e) Two-stage (exemplar patch fusion-based method followed by the bandlet-based 3D regularization) inpainting result (Frame number 13, MSE=11.86)

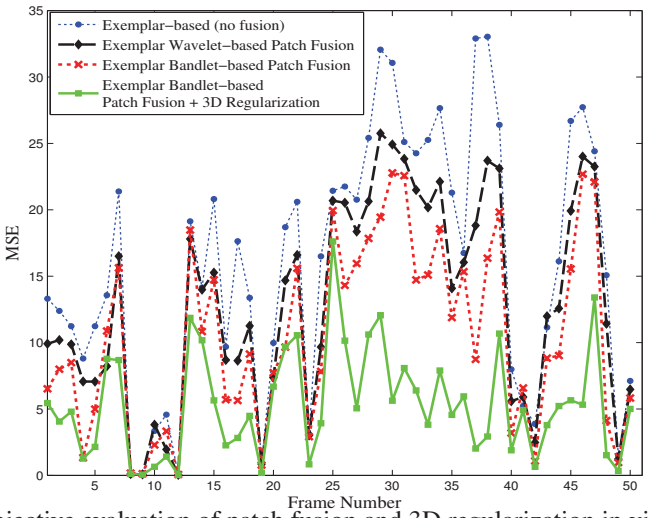


Figure 3.9: Objective evaluation of patch fusion and 3D regularization in video inpainting.

value of the fusion-based completion for almost all the frames is lower than that of the conventional exemplar-based completion scheme labeled as "Exemplar-based" in Fig. 4.12. In order to show the performance of the proposed bandlet-based patch fusion technique in video completion tasks, the experiment is performed another time using another image fusion technique. A patch fusion scheme similar to Sec. 3.2.1 is considered for a well-known image fusion technique based on wavelets introduced in [57]. Then, the completion task is performed by means of the exemplar-based platform applying this fusion technique. Similar to the wavelet stage of the bandlet transform, Daubechies wavelets are employed in this wavelet-based patch fusion scheme. The resulting MSE values of all the generated frames using this method are presented as the "Exemplar-based Wavelet patch Fusion" plot

in Fig. 4.12. The plots shown in Fig. 4.12 indicate visually pleasing completion results for the bandlet-based patch fusion scenario compared to simply replacing the missing region by the best matching patch and also using an effective fusion method [57] based on wavelets.

Similar experiments are carried out in order to evaluate the effectiveness of bandlet-based 3D regularization in the inpainting task. This time, the proposed two-stage video inpainting method is carried-out for the video sequence of Fig. 4.11. In other words, the damaged video of Fig. 4.11(b) has been inpainted using spatio-temporal patch-fusion followed by the 3D regularization step in order to refine the results and also to preserve the visual consistency (Fig. 4.11(e)). The corresponding MSE plots in Fig. 4.12 show a higher performance for the proposed video completion method compared to using solely the patch fusion scheme or the convectional exemplar-based video inpainting technique presented in Sec. 3.2. It is worth mentioning again that the regularization methods are not practical for large regions due to the blur effect they impose on the resulting frames [6]. However, as presented here a precise combination of a regularization-based method and an exemplar-based method can result in a higher accuracy.

3.3.2 Comparison with State-of-the-art Methods

The performance of video inapinting/completion methods is generally evaluated subjectively. However, we use MSE to evaluate the effectiveness of our method as done in our previous experiments. A manual damage is produced on an original video sequence. Then, the result of the completion method on the damaged video is compared with the original video sequence by computing the MSE value for the corresponding frames of the original and the completion result video sequences. Fig. 3.10(a) shows a frame of the video chosen for evaluation which is damaged as in Fig. 3.10(b) and then completed as in Fig. 3.10(c). The green plot in Fig. 3.11 shows the MSE graph of all the 47 frames of the original video and the completion result sequence using the proposed method. For almost all the frames, the MSE value is low, indicating visually pleasing completion results.

We compared our approach to two well-known video completion methods introduced in [26] and [35]. Fig. 3.12 shows a sample frame of a video sequence processed by these two methods as well as by our technique. We performed the same MSE graph generation

(a) (b) (c)

Figure 3.10: (a) Original frame. (b) Damaged frame. (c) Proposed method completion result (Frame number 22, MSE=8.18).

i.e., computing MSE for the completion results and the original sequence. The produced graphs are depicted in Fig. 3.11. The graphs and the computed average MSE values of all the frames indicate a high performance for our proposed method compared to these two methods. Despite the crucial importance of temporal consistency in video completion, to the best of our knowledge, none of the existing techniques have been evaluated objectively in the literature in this sense. This is due to the fact that there is no standard temporal quality measurement framework designated for video inpainting. Here, we employ the spatio-temporal most apparent distortion (STMAD) model to analyse our approach with regards to temporal consistency [58]. A low value for STMAD indicates that the video is temporally consistent [58]. In fact, the extension of the still image-based most apparent distortion (MAD) model [59] by taking the motion information between frames into account is the main idea of STMAD. Table 5.2 presents STMAD values obtained for the completed videos by the three different techniques. It should be mentioned that the obtained values are normalized to the range of 0 to 1 and then they are subtracted from 1. Hence, a higher value in the table indicates a better consistency. The STMAD is calculated between the inpainted video and the original one of Fig.3.10. As the table indicates, our approach has the highest value for STMAD and consequently the best temporal consistency among the other methods. This high performance is largely credited to the effective role of bandlets in the patch-fusion scheme in the spatio-temporal completion and the 3D regularization and a good combination of these two different stages.

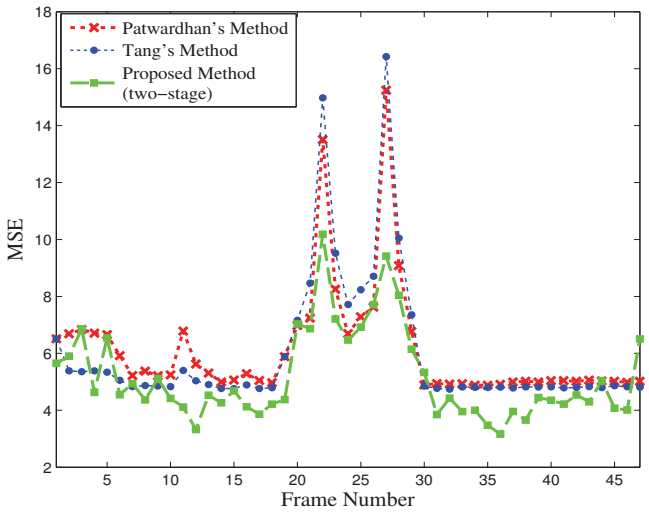


Figure 3.11: Objective evaluation of the proposed video completion method. Average frame MSE is 6.11, 6.02, 5.1863 for Patwardhan [26], Tang [35], and the proposed two-stage method, respectively.

(a)

(b) (c) (d)

Figure 3.12: A sample frame inpainted by three differen methods. (a) Damaged frame. (b) The proposed algorithm result. (c) Completion result of [26]. (d) Completion result of [35]. (For a better illustration the images are cropped from left, right and bottom)

Table 3.1: Temporal consistency evaluation. STMAD obtained for each resulting video using different video completion techniques.

Method	Patwardhan [26]	Tang [35]	Ours
STMAD	0.501	0.484	0.601

Chapter 4

Video Text Removal

4.1 Introduction

Embedded text in a video sequence provides valuable information of paramount importance. Texts usually appear as logos, subtitles, captions or banners in the video sequence. Examples of such informative embedded texts can be largely found in the news and other popular television broadcastings. Although texts provide additional information, not all of them are necessary as they may occlude important portions of a video. Consider the case, for instance, when indirect advertisement is not permitted but it is already included within a frame sequence in the form of a caption. Hence, there should be a way to erase the unwanted text from the video. This motivates the need of an automatic approach to remove undesired texts from a video.

Roughly speaking an automatic video text removal scheme involves two main stages: i) an automatic video text detection, and ii) an effective video completion/restoration after the text removal. Existing video text completion techniques rarely cover both of these aspects in a single platform. The proposed methods in [60–63], for instance, deal with only the restoration stage from the video inpainting perspective. The proposed scheme in [64] utilizes a support vector machine (SVM)-based text detection method to localize texts in the video, then performs the inpainting method introduced in [1] in order to restore the parts occluded by the removed texts. In [65] the video caption detection is performed

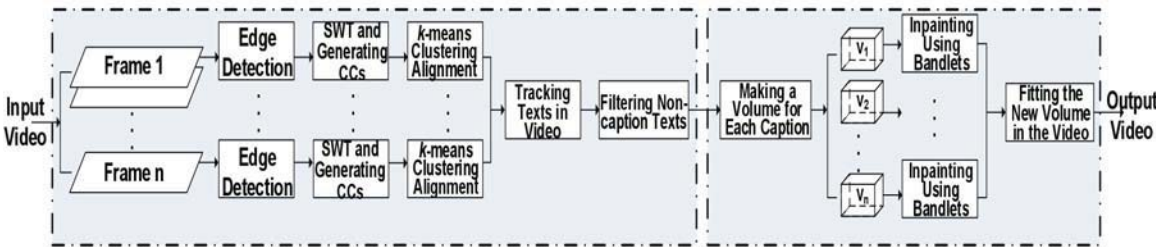


Figure 4.1: Main stages of the proposed video text detection and removal.

by a multi-layer perceptrons scheme and genetic algorithms. The removal task is carried out by modeling it as an optimization problem whose cost function is made by isophote constraints and minimized by genetic algorithms.

In this chapter, we propose a video text completion approach which consists of an accurate video text localization technique and an effective restoration stage. Fig. 4.1 illustrates different steps of the proposed framework. In the video text detection stage, embedded texts in each frame are localized then tracked in the entire sequence and removed eventually. The most challenging step in video text detection algorithms is finding the text locations in each frame before tracking them. Therefore, we developed a precise single frame/image text detection algorithm using SWT and unsupervised classification. A set of feature vectors is generated for the connected components (CC) produced by SWT. Then, the feature vectors are employed in a k-means clustering to distinguish text CCs from non-text ones. Since SWT requires accurate edge locations to process, we also introduce an effective bandlet-based edge detector. The restoration task after removing the texts from the video is performed by applying spatio-temporal geometric flows extracted by bandlets to reconstruct the missing data. To that end, the inpainting algorithm, i.e. the 3D volume regularization algorithm introduced in Chapter 3 is employed. The two stages of our video text completion framework are utilized independently. The main challenge of our video text detection approach is to detect the text regions in each frame. Therefore, a review of related image text detection techniques is first provided in this chapter.

4.2 Single Frame Text Detection Related Works

Text in video, especially the superimposed text, is the most reliable information to be considered in video indexing research work. Many research works have addressed the task of extracting text regions from videos [66–69]. Our proposed video text detection approach is based on tracking the detected texts in each frame [15]. In such tracking-based text detection methods [70, 71] the accuracy of detected text locations in each frame significantly affects the performance of the video text detector. To circumvent this limitation, we develop an accurate single frame (image) text detector as the base of our video text detection approach. Existing image text detectors may be broadly classified into two main groups [72, 73]: texture (also called region) based and CC-based methods.

Texture-based methods scan the image at a number of scales and consider the embedded text as a particular texture pattern that is distinguishable from other parts of the image and its background. Basically, features of various regions of the image are retained. Then, the presence of text is identified by either a supervised or an unsupervised classifier. Finally, the neighboring text region candidates are merged based on some geometric features to generate text blocks. As examples of such methods, the technique introduced in [74] applies Sobel edge detector in all Y, U, and V channels, then invariant features such as edge strength, edge density, and edge's horizontal distribution were considered. The method presented in [75] produces a statistical-based feature vector using the Sobel edge map and applies k-means algorithm to classify image regions into text and non-text parts. Assuming that the horizontal gradient value of text regions is higher than that of other parts of the image, the method in [76] thresholds the variance of gradient values to identify text-regions. A support vector machine (SVM) classifier is used in [77] to generate text maps from the gray-level features of all local areas. The method extracts the features through each layer of image pyramids. The method in [78] also takes advantage of image pyramids to find local thresholds to detect text areas. The frequency domain is shown to be practical in text-region classifications. For example, classification is applied in wavelet domain in [79] and [80] in order to detect aligned texts in an image. In the same vein, the proposed method in [81] applies frequency domain coefficients obtained by the discrete cosine transform (DCT) to

extract features. By thresholding filter responses, text-free regions are discarded and the remaining regions are grouped as segmented text regions.

CC-based methods stem from the observation that text regions share similar properties such as color and distinct geometric features. At the same time, text regions have close spatial relationship. Therefore, based on such properties they are grouped together and form CCs. The method introduced in [82] finds candidate text regions by utilizing Canny edge detector, then a region pruning step is carried out by means of an adjacency graph and some heuristic rules based on local components features. Candidate CCs are extracted by the method in [75] based on edge contour properties, then text-free components are pruned by analysis of wavelet coefficients. In order to find CCs, an adaptive binarization is applied in [83]. Statistical analysis of text regions is performed to determine which image features are reliable indicators of text. This is done by considering a large training set which consists of text images. In fact, the feature response of the candidate CCs must be similar to the text images. A useful operator is defined in [84] to find stroke width of each image pixel. The SWT image is generated by shooting rays along the direction of each edge pixel's gradient. Then, the SWT values are grouped based on their ratios in order to produce CCs. The text-candidate CCs are selected by applying some rules such as aspect-ratio, diameter and variance of stroke width of each component. In [85] the CCs are found by k-means clustering in the Fourier-Laplacian domain. Then, the candidate CCs are filtered by test string straightness and edge density features. This method is not only practical for horizontally aligned texts but also for any arbitrary oriented text. A CC-based algorithm is introduced in [86], which employs Maximally Stable Extremal Regions (MSER) as the basic letter candidates. Then, by using geometric and stroke width information non-text CCs are excluded.

A number of existing methods are not categorized in the aforementioned two groups. As an example, the method in [87] is a hybrid technique whose first step detects text regions in each layer of image pyramid and projects the text confidence and scale information back to the original image followed by a local binarization to generate candidate text components. A CRF model filters out non-text components and then a learning-based minimum spanning tree (MST) is used to link the CCs. Sparse representation is also applied in the field of

text detection. The introduced method in [88] benefits from two learned discriminative dictionaries [89], one for document images, and another for natural images to distinguish between text and background regions in an input image.

The general scheme of our proposed image text detection method consists of producing the image edge map and then finding CCs based on SWT guided by the generated edge map. Next, precise feature vectors are formed using the properties of CCs from SWT and pixel domain. An unsupervised clustering is performed on the image CCs to detect the candidate text CCs. Finally, text candidate components are linked to form text-words. The method is considered as a CC-based technique whose contribution is twofold: 1) Since accurate edge maps drastically enhance SWT results, a precise edge detection approach adaptive to text-regions is proposed by employing the bandlet transform. 2) A feature vector based on text properties and stroke width values is employed in k-means clustering in order to detect text CCs.

4.3 Video Text Detection

The proposed video text detection performs a motion analysis and tracking for the text objects found in each frame in order to detect the actual video texts and distinguish them from the rest on the video. The frame text detector is first presented and then the tracking scheme is discussed in this section.

4.3.1 Text Detection in Each Frame

The frame text detector is in fact an image text localization technique based on CCs that benefits from SWT and a k-means clustering which in turn requires accurate edge locations. The scheme contains three main stages as follows: 1) Edge detection, 2) SWT and generating CCs, and 3) k-means clustering of the CCs.

Edge Detection Using Bandlets

As discussed earlier, the bandlet transform effectively represents the geometry of an image. We take advantage of this representation and propose an edge detection algorithm that can

be used effectively in text-detection techniques. On the other hand, it has been shown in [90, 91] that finding local maxima of wavelet transform coefficients is similar to the multi-scale Canny edge detector operator. Since the image coefficients are all warped along local dominant flows in the bandlet transform, the final bandlet coefficients generated for each segmentation square S have the form of approximation, and high-pass filtering values appear in the wavelet transform of a 1D signal. We benefit from the bandlet-based resulting 1D high-pass frequency coefficients that are adapted to the directionality of the edge that exists in each segmentation square S in order to find a binary map of the edge positions in the image.

The bandlet transform is performed on the original image, and for each segmentation square S the bandlet coefficients are generated. For each S, the resulting coefficients are grouped in low-pass (approximation) and high-pass filtering results similar to the 1D wavelet transform. Since the approximation part consists of coarse information of the original signal, we discard it and only process the high-pass coefficients. The first-order derivatives of the fine-detail bandlet coefficients are computed. By applying a contextual filer, we find local maxima of the resulting gradient signal since many meaningful edges can be found in the local maxima of the gradient not only in the global maxima. Then, in order to improve the quality of the edge image a two level thresholding is employed.

For each point x_i in the gradient signal, we check if x_i is a local maximum and its value is greater than a threshold T_G. If so, x_i is kept as an edge point coefficient otherwise it will be discarded. Hence, a window with size $2L + 1$ centered at x_i is set. Then, the binary indicator of edge points in the gradient signal is generated as follows:

$$M_i = \begin{cases} 1 & \text{if } g_i > T_G \ \wedge \ g_i > g_j, \ \forall j \in [i-L, i-1] \wedge \\ & \quad g_i > g_j, \ \forall j \in [i+1, i+L] \\ 0 & \text{otherwise,} \end{cases} \tag{4.1}$$

where g_i represents the gradient value for x_i and g_j indicates gradient value of neighboring pixels of x_i that exist is the window. M is a map of local maxima of the gradient signal. The corresponding locations of 0's of M in the bandlet fine (high-pass) coefficients are set to 0, for all the bandlet squares S. Then, the inverse bandlet transform is performed in order to have the final edge locations of the original image.

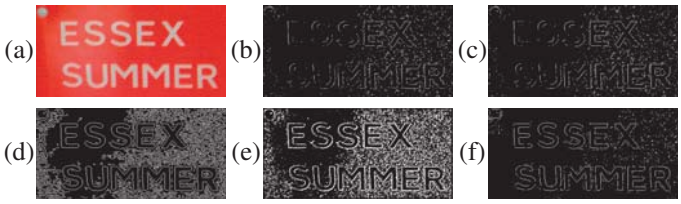


Figure 4.2: Edges by different methods. a) Original image. b) Sobel. c) Prewitt. d) Canny. e) Wavelet-based. f) Bandlet-based.

Obviously, the quality of the edge map depends on the value of the threshold T_G. In order to ensure a high quality, a two-level thresholding is employed. First, the edge detection is performed using a low value for T_G and the edge image E_l is produced. The algorithm is performed another time utilizing a higher value for T_G to generate the edge image E_h. Apparently, E_l includes more edge pixels than E_h, which only includes significant edges. Also, all the edge pixels of E_h exist in E_l. A combination of E_h and E_l leads to more reasonable results. For each edge component C_{eh} that exists in E_h we inspect E_l and check if there is an edge component C_{el} in E_l that overlaps C_{eh}. If so, C_{el} is taken from E_l and saved in the final image edge map.

Considering the bandlet transform structure strictly adapted to strong local pixel flows through a geometry-based dyadic segmentation, this edge detection scheme reveals reliable edge pixels. Moreover, since the regions consisting of sparse singularities such as noisy and foliage pixels, and the regions with various pixel intensities are eliminated in the bandlet geometric segmentation, the resulting edges are quite appropriate to localize text-edges embedded in the image. Fig. 4.2 shows the results of four different edge detection methods including Sobel, Prewitt, Canny, wavelet and the proposed bandlet-based technique. The input image includes a text and noisy pixels. Our proposed approach shows considerably better results compared to the other methods.

Stroke Width Transform

The SWT value of each pixel is roughly the width of the stroke that contains the pixel. A stroke is defined as a part of the image that forms a band of constant width. In the first step, we find the edges of the input image using the proposed edge detection method (Sec.

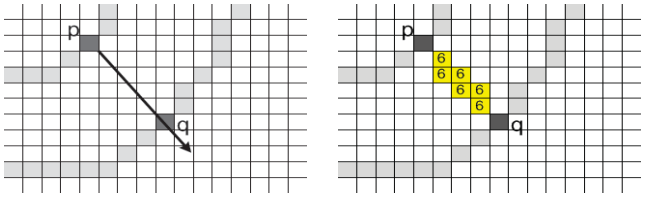


Figure 4.3: Stroke width transform. Finding the gradient value of edge pixel p and shooting a ray in its direction and finding an edge pixel q with opposite gradient direction on the ray (left). Assigning the stroke width value to each pixel that lies on the ray (right).

Figure 4.4: The original image and the SWT output using bandlet-based edges are shown at left and right, respectively.

4.3.1). Then, the gradient direction d_p of each edge pixel p is determined. A ray starting from p with the direction of d_p is considered and followed until it meets another edge pixel q. If the gradient direction d_q at edge pixel q is approximately opposite to d_p, the distance value of p and q is assigned to all the pixels that lie on the ray. Fig. 4.3 shows SWT values of sample pixels that lie on a ray. SWT of a sample image whose edge map generated using bandlets (Sec. 4.3.1) is computed and shown in Fig. 4.4. This figure shows how effective SWT can be in finding text regions in images.

Neighboring pixels are grouped together and form CCs if they have similar stroke width values. The traditional CC algorithm is not performed on a binary mask but on the SWT values with a different connection criterion. In the CC algorithm, 4-neighboring pixels are considered. Adjacent pixels are grouped if the ratio of their stroke width values is higher than 0.3 and lower than 3. Features of the produced CCs are used to find text candidates.

Unsupervised Classification and Refinement

We need to identify components that very likely contain text. Thus, we employ a set of rules and assumptions in order to make a feature vector for each component. Then, the

feature vectors are fed to k-means clustering to identify text components.

The variance V_{SWT} of stroke width values in all the text components is not too large. A high value of V_{SWT} for a CC means that the component consists of the pixels of a foliage region. The mean μ_{SWT} and median M_{SWT} values of each CC are also considered in order to find text components with the same stroke width, since almost all the characters of a word would have the same stroke width. Another important feature of a text component is that it is neither too long nor thin. Therefore, the ratio R_s of the component diameter and its median stroke width M_{SWT} are added to the feature vector.

Considering a sample character as a text CC, one may observe that the gradient directions of edge pixels of the component vary significantly. In other words, a text component can have edge pixels with gradient directions ranging from 0° to 90° for character "I" for instance or 0° to 180° for "O", indicating a large range of directionality. So, we calculate the variance V_G of gradient directions of all the edge pixels of a CC and save it in the feature vector. Also, a text component has almost a symmetric distribution for the gradient directions of the edge pixels. This is due largely to the fact that a character has at least two sets of edge pixels roughly parallel to each other with opposite gradient directions. Therefore, we estimate having a symmetric distribution for the direction of edge pixels by computing the skewness SK_G for the gradient directions and add it to the features:

$$SK_G = \frac{\mu_3}{\sigma^3} = \frac{\frac{1}{n}\sum_{i=1}^{n}(g_i - \mu_G)^3}{(\frac{1}{n}\sum_{i=1}^{n}(g_i - \mu_G)^2)^{3/2}}, \tag{4.2}$$

where n is the total number of edge pixels in a CC, g_i is the gradient direction at edge pixel i and μ_G is the mean of gradient directions of the edge pixels of the CC. In fact, in this equation μ_3 and σ are the third moment about the mean and standard deviation of the gradient directions, respectively.

An important feature attributed to texts in images is their relatively high contrast with the background compared to other regions of the image. This is due to the nature of utilizing texts i.e, catching one's sight and conveying information. A scene text or a caption text in a video frame, for example, must have a strong contrast with the background since the producer of the text wanted them to stand out clearly. Thus, we consider this important property and use it in the feature vector. Typically, contrast is estimated by Weber formula:

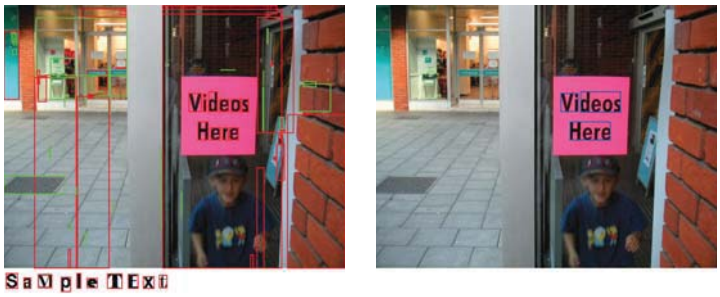


Figure 4.5: Clustering of CCs. Text and non-text CCs identification (left). Merging text CCs to generate the final result (right).

$C = (L_o - L_b)/L_b$, where L_o and L_b are the luminance of the object and its surrounding background, respectively. More complex contrast analysis can be found in [92, 93] by employing discrete cosine transform and wavelets. We simply use the local mean μ_L and standard deviation σ_L of the image intensity to estimate the contrast value of a CC with its background [94]; $C_L = \sigma_L/\mu_L$. C_L is computed for the intensity pixels that exist in the bounding box of a CC and added to its feature vector.

Finally, the bounding box itself must have a reasonable aspect-ratio for a text CC. Normally, the height of a text component is larger than its width and their aspect-ratio is not too large. So, we find the aspect-ratio R_{asp} of the bounding box of each CC and use it in the feature vector. The final feature vector of each CC has the following form:

$$\overrightarrow{\boldsymbol{F}} = \{V_{SWT},\ \mu_{SWT},\ M_{SWT},\ R_s,\ V_G,\ SK_G,\ C_L,\ R_{asp}\} \tag{4.3}$$

The produced vectors $\overrightarrow{\boldsymbol{F}}$ of all the CCs of the image are fed to a k-means algorithm with $k = 2$ and consequently clustered into two groups, non-text and text components as shown at left in Fig. 4.5 for instance. In order to identify which cluster is associated to the texts and which is not, at the beginning of the process we append a sample text to the end of each input image. Hence, the resulting cluster that contains the sample text components is considered as the group of text components and the rest of the components are discarded. In the last step, the remaining text components which are horizontally aligned and have reasonable distance to each other, for example as far as a character width, are grouped together and form the word components as shown at right in Fig. 4.5.

Algorithm 4.1 Video text detection.

1: **for** $f_i = 1 \rightarrow f_N$ (f_N:total number of frames) **do**
2: Find the edge map for f_i (Sec. 4.3.1)
3: Generate the STW of f_i (Sec. 4.3.1)
4: Unsupervised classification using Eq. (4.3) (Sec. 4.3.1)
5: Text components alignment (Sec. 4.3.1)
6: Save all the spatio-temporal locations of the texts
7: **end for**
8: Find the video global motion μ_G (Eq. (4.4))
9: **for** all the detected frame texts **do**
10: Track the text using CAMSHIFT algorithm
11: Find the local motion field μ_O of the tracked text (Eq. (4.4))
12: Find the dissimilarity of μ_O and μ_G (Eq. (4.5))
13: Mark the tracked text as a moving text if it satisfies Eq. (4.6)
14: **end for**

4.3.2 Text Detection in Video Sequence

Once the text locations are detected within each frame of the video sequence, a mechanism is needed to distinguish the video texts from the natural texts that may exist in a frame. Considering that an embedded video text appears in a consequence of frames with specific motion properties compared to the rest of the video, we employ a tracking and motion analysis scheme in order to specify the video text regions.

The detected text locations in each image are considered as different objects and CAMSHIFT algorithm [95] is performed on each of them. It is worth noting that CAMSHIFT starts from a text object of the current frame if the text object has not been already tracked in the sequence. Therefore, the large set of text locations of all the frames is reduced to a set of tracked text objects in the video. For each text object we have the spatial and temporal locations. In the next step, the local motion field of each text object and the global motion field of the video are estimated using Lucas-Kanade optical flow computation algorithm [36]. Each motion vector $(u, v)^T$ is estimated by solving the following optimization problem:

$$\arg\min_{(u,v)} \sum_{x,y,t} \left(u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} \right), \tag{4.4}$$

where $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are image derivatives along spatial and temporal directions. The motion information of each point $\boldsymbol{p} = (x, y, t)$ in the video is represented by $(u(\boldsymbol{p}), v(\boldsymbol{p}))$.

An embedded video text like a caption or a subtitle can be distinguished from the natural text regions by analyzing the motion fields. An embedded video text has a distinct local motion compared to the global motion field of the video. Besides, a video text usually has a dominant motion which can have one of the three patterns: 1) along horizontal direction like a caption which enters from the right down corner and leaves from the left down corner of the screen, 2) along vertical direction like a text line which rolls down from top to the bottom of the screen, 3) stand-still like a logo on top corner of the screen or a subtitle that appears at bottom of the screen. Hence, the mean value of the global motion vectors $(\bar{u}(p_G), \bar{v}(p_G))$ and the mean value of the motion vectors associated to each text object $(\bar{u}(p_O), \bar{v}(p_O))$ are calculated. Then, the dissimilarity between these two vector mean values is calculated as follows:

$$d = \|\mu_G - \mu_O\|, \tag{4.5}$$

where $\mu_G = \bar{u}(p_G), \bar{v}(p_G)$ and $\mu_O = \bar{u}(p_O), \bar{v}(p_O)$. Once the motion field of a text object satisfies the following condition:

$$d > T_m \wedge \big(\bar{v}(p_G) = 0 \vee \bar{v}(p_G) = 0\big), \tag{4.6}$$

the text object is considered as an embedded video text. The condition in (4.6) indicates that a video text is distinguishable from the rest of the video specifically the natural text regions if its local motion dissimilarity with the global motion is larger than a threshold value T_m and at the same time its local motion is only horizontal, vertical or it is totally static $\big(\bar{v}(p_G) = 0 \vee \bar{v}(p_G) = 0\big)$. The steps of the proposed video text detection scheme are listed in Algorithm 4.1. Fig. 4.6(b) shows the detected embedded video texts in a sample frame which are distinguished well from the other text objects that exist in the frame shown in Fig. 4.6(a). Once the video text is located in the entire video, its associated CCs in all the frames are marked and removed from the sequence (see Fig. 4.6(c)).

4.4 Video Text Removal

The resulting video after removing the caption text needs to be inpainted to be filled with appropriate pixels. Interpolation-based video inpainting schemes are generally practical to

Figure 4.6: Output results of video text detection. a) Original frame. b) Video text detection result. The video text objects are differentiated from the other text objects by another color. c) Detected video text is removed and masked.

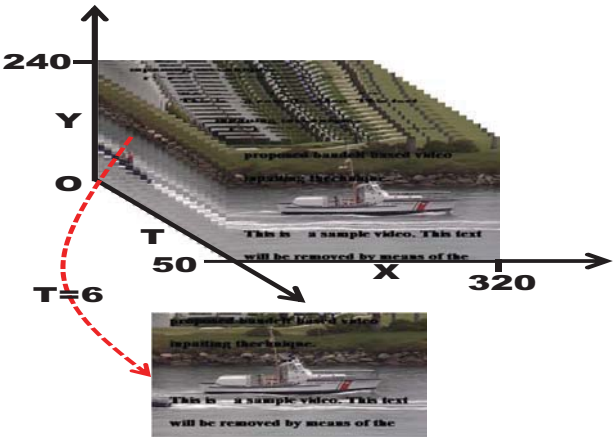
restore videos that contain small and thin missing regions such as the areas occluded by text pixels. At the same time, these types of methods do not require sophisticated pre-processing steps like segmentation, tracking, and motion vector estimation. Image sparse representation is effectively adapted to the local image properties, specifically the textural ones. Therefore, we employ the inpainting scheme introduced in Chapter 3 to address the restoration task. Considering that the resulting video after text localization and removal as a video that consists of T frames with $X \times Y$ pixels for each, there are T, X and Y planes along the time, horizontal and vertical directions, respectively, the iterative steps of the minimization process introduced in Algorithm 3.1 in order to inpaint the video volume is used.

Fig. 4.7(a) displays a sample video after text removal to be restored by the proposed inpainting algorithm. The inpainting result of a single frame of the sequence is shown in Fig. 4.8.

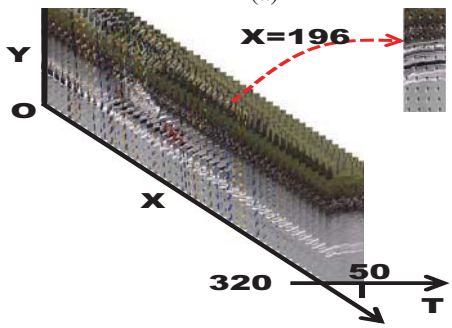
4.5 Experimental Results

Several video sequences are used to evaluate the proposed video text removal method. The set of videos contains sequences captured from TV, movies and video games. The resolution of each video sequence is 320×240. In the implementation of the text detector and the video inpainting scheme, the following settings are used:

- Gray-scale values of the RGB frames are found by $(R+G+B)/3$ whenever needed.



(a)



(b)

Figure 4.7: A video volume from different views. a) X-Y planes view. b) T-Y planes view.

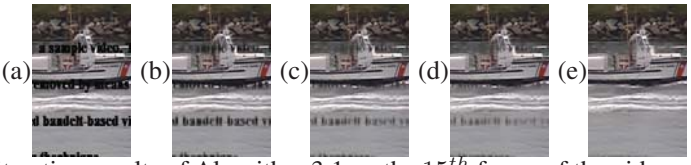


Figure 4.8: Various iteration results of Algorithm 3.1 on the 15^{th} frame of the video of Fig. 4.7. (The images are cropped from left, right and bottom.)

- In order to reduce the size of the inpainting volume, instead of performing the inpainting task on the entire video volume for a text which has a bounding volume size of $w \times \ell \times h$, the bounding box is confined to $w + 50 \times \ell + 10 \times h + 50$ (w and h, respectively, represent width and height of the volume in pixels and ℓ represents the number of frames in the volume).
- In the bandlet transform the number of scales j, on which geometry is computed, is set to 3.
- The introduced scale factor l by Alpert transform in the bandletization (Sec. 2.2) is set to 4.

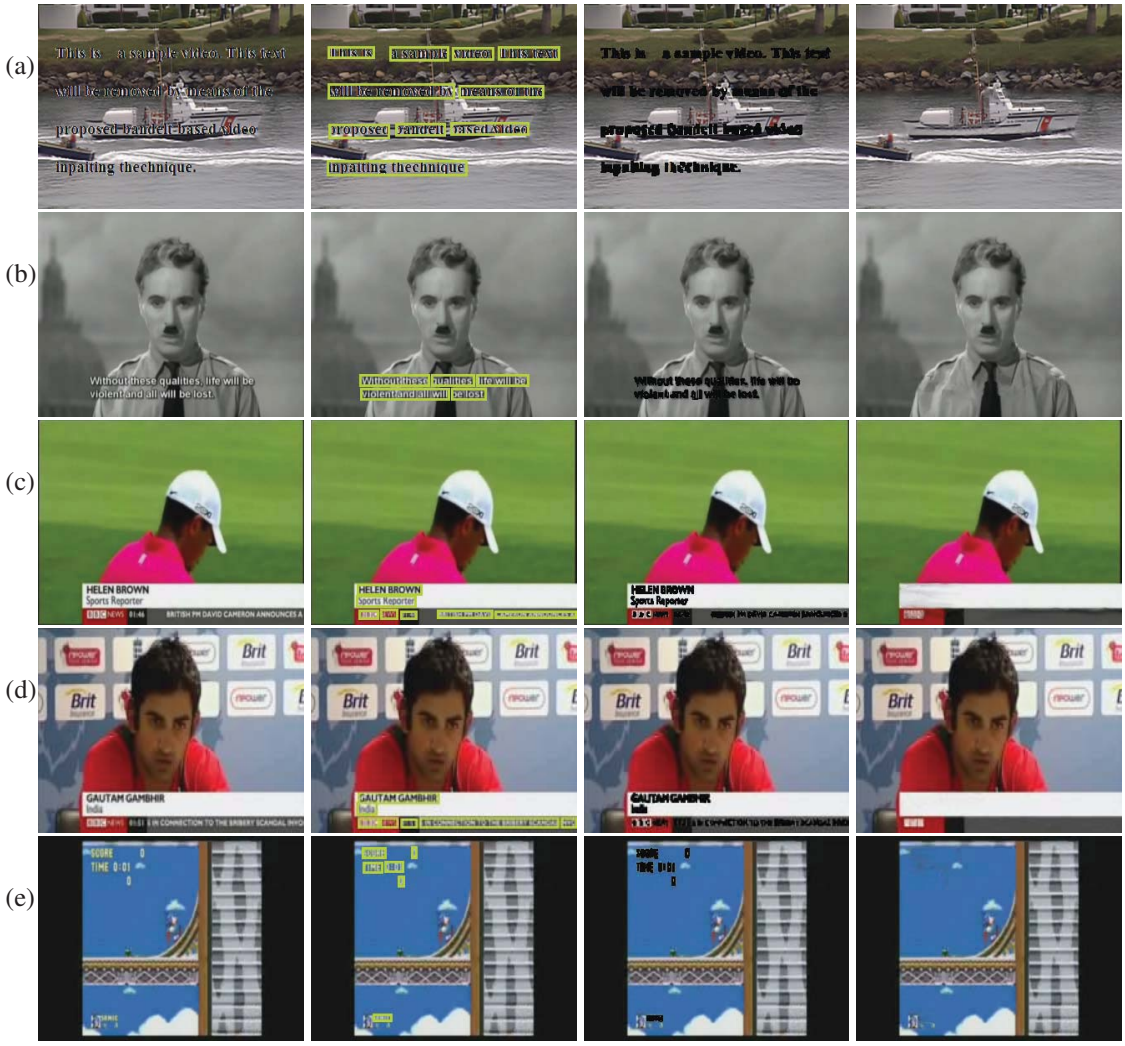


Figure 4.9: Sample automatic video text removal results.

- Orthogonal wavelets are used in the bandetization.
- A fixed size 8×8 segmentation is employed instead of the complex dyadic segmentation introduced in Sec. 2.2.

Fig. 4.9 depicts the results of our video text removal and restoration scheme on different sequences. The objective is to detect and remove the text rolling over the sequence shown in Fig. 4.9(a), the static subtitle that appears in the video of Fig. 4.9(b), the static and sliding captions in Fig. 4.9(c) and Fig. 4.9(d), and the texts appear in the video of Fig. 4.9(e). Although the sequences involve various camera motions and no matter whether the embedded text is static or moving, the text detection is performed well and the removed text pixels are restored visually pleasantly using the proposed inpainting scheme. In order

Figure 4.10: Sample text detection results using the proposed technique on the ICDAR dataset.

to gain further insight into the effectiveness of each stage of the propped automatic text removal scheme, various analyses are presented as follows.

4.5.1 Evaluation of the Video Text Detection

As mentioned before, an important contribution of our work is the proposed image text detector which is used in the single frame text detection to locate the potential text regions in the entire sequence before tracking them. Therefore, this subsection starts with analyses of the proposed image text detection technique. Next, a performance analysis of the entire video text detection steps is presented.

Single Image Text Detection

We evaluated our proposed image text detection approach introduced in Sec. 4.3.1 on the ICDAR text locating contest dataset [96]. In Fig. 4.10 sample text detection results of our approach on ICDAR dataset are presented. The dataset contains 251 color images of various sizes ranging from 307×93 to 1280×960. Along with the images, the dataset provides ground-truth locations of the texts that exist in the images, called targets, to have a precise evaluation of the results of text detection techniques. The result of a text detection method in the form of a rectangle that bounds a text in the image is called estimate hereafter. We followed the same evaluation scheme by means of *Precision* and *Recall* used in ICDAR competitions [96, 97]. *Precision* is the number of correct estimates divided by the total number of estimates. A method has a low precision if the number of text bounding rectangles is too large. *Recall* is defined as a ratio of the number between correct estimates and the total number of targets. Hence, a method that results in a large number of incorrect rectangles has a low Recall score. The results of a text locating system are not as exact

Table 4.1: Performance of the Proposed Image text Detection Method using Other Edge Detectors.

Method	Precision	Recall	f
Bandlet edges	0.76	0.66	0.71
Wavelet edges	0.71	0.59	0.65
Canny edges	0.67	0.51	0.58
Sobel edges	0.53	0.56	0.53

Table 4.2: Performance of Different Image Text Detectors on the ICDAR Dataset.

Method	Precision	Recall	f
Our Method	0.76	0.66	0.71
Zhao *et al.* [88]	0.64	0.65	0.65
Epshtein *et al.* [84]	0.73	0.60	0.66
Gllavata *et al.* [79]	0.44	0.46	0.46

Table 4.3: Performance of Different Video Text Detectors.

Method	Precision	Recall	f
Our Method	0.70	0.61	0.65
Lyu *et al.* [67]	0.66	0.60	0.63
Kim *et al.* [69]	0.58	0.56	0.57

as human tagged locations. Therefore, a match m_p between two rectangles defined as the area of their intersection divided by the area of the minimum bounding box containing both rectangles is used. The value of m_p is zero for two rectangles without any intersection and one for exactly alike rectangles. For each rectangle in the set of estimates, the closest match in the set of targets is found, and vice versa. Hence, the best match $m(r; R)$ for a rectangle r in a set of rectangles R is defined as

$$m(r; R) = \max\{m_p(r; \acute{r})|\acute{r} \in R\}. \tag{4.7}$$

Then, Precision and Recall are defined as

$$\text{Precision} = \frac{\sum_{r_e \in E_s} m(r_e, T_r)}{|E_s|}, \quad \text{Recall} = \frac{\sum_{r_t \in T_r} m(r_t, E_s)}{|T_r|}, \tag{4.8}$$

where T_r and E_s are the sets of target (ground truth) and estimated boxes, respectively. These measures are combined into a single measure f with a weight factor α set to 0.5:

$$f = \frac{1}{\frac{\alpha}{\text{Precision}} + \frac{1-\alpha}{\text{Recall}}}. \tag{4.9}$$

In the first experiment, we employed other edge detection methods in our text detection scheme instead of the proposed bandlet-based edge detection approach (Sec. 4.3.1).

Table 4.1 shows $\mathrm{Precision}$, Recall and f values obtained by our text detection approach for the images of ICDAR dataset using Sobel, Canny, conventional wavelets and bandlet transform edge detection techniques. In this table the highest values of $\mathrm{Precision}$, Recall and f are attributed to the method which employs our proposed bandlet-based edge detector. The result of our proposed approach has been compared with other methods as well. Table 4.2 shows the list of methods used for comparison and their $\mathrm{Precision}$, Recall and f values for ICDAR dataset images. The proposed method has a better performance compared to the other listed methods. Specifically, our approach outperforms the SWT-based method introduced in [84] already shown to have a good performance compared to several other existing methods [84] including the participating algorithms in ICDAR 2003 [97] and ICDAR 2005 [96].

Video Text Detection

The whole video text detection technique needs to be evaluated. Therefore, we performed a similar Precision-Recall performance analysis to find out the effectiveness of our proposed video text detection approach. Instead of an image dataset, we considered this time the video sequences of Fig. 4.9(a) and Fig. 4.9(b) as the benchmark. These two video sequences have 168 frames in total. For each frame we manually located target texts' locations and marked them as ground-truth text locations. Then, after performing the video text detection, Precision, Recall and f values are calculated using (4.8) and (4.9) for the video text detection results. Sample results are provided in Fig. 4.9. The performance of the proposed method has been compared to that of other existing video text detection methods. Table 4.3 presents the Precision, Recall and f of the methods introduced in [67] and [69]. The methods are applied on the same benchmark as our proposed approach. The table shows that our method distinctly outperforms the other two methods.

4.5.2 Evaluation of the Inpainting Method for Video Text removal

The restoration results using the proposed inpainting scheme is illustrated on sample videos in Fig.4.9. In all cases, the proposed method performs the inpainting task quite well. In

(a)  (b) 

(c) 

Figure 4.11: a) Original frame. b) Damaged frame. c) Completion result by our method (Frame number 11, MSE=8.16).

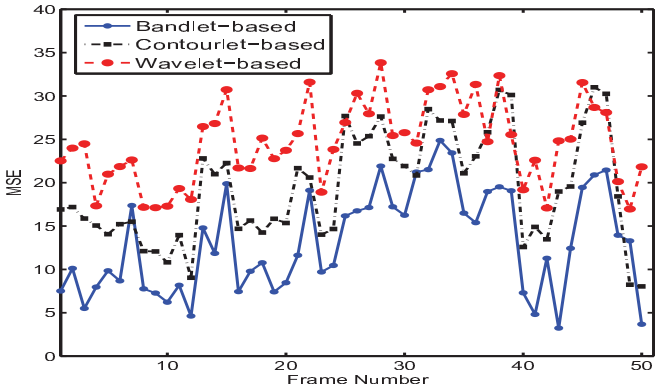


Figure 4.12: Objective evaluation of the proposed video inpainting approach using different image sparest representations.

this subsection we evaluate the performance of the proposed video inpainting technique. First, the effectiveness of the bandlet transform in comparison with two well-known image sparse representation platforms, wavelet and contourlet transforms is evaluated. Next, a quantitative comparison with two existing video inpainting techniques is provided.

Effectiveness of Bandlet Transform in Video Inpainting

The bandlet transform is an effective image representation which is strictly adapted to the local geometry of the image. This feature can be so practical in the case of spatio-temporal inpainting as we deal with continuation of geometric structures that lie within an image or a video. Therefore, we evaluated the effectiveness of this transform and compared it with other sparse representations.

A manual text is generated on an original video sequence. Then, the resulting video after text removal is completed by the proposed bandlet-based video inpainting approach presented in Sec. 4.4. The completion is performed once again using conventional wavelets in Eq. (3.4) and consequently in Eq. (3.16). Also, the inpainting is performed another time by applying the contourlet transform [45] instead of bandlet transform. Then, for all

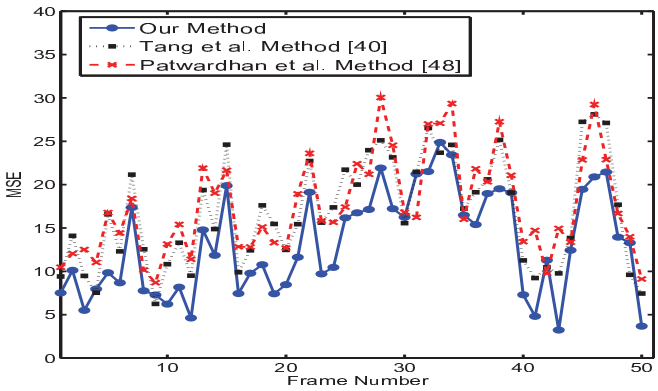


Figure 4.13: Objective evaluation of the proposed video inpainting method compared to Patwardhan [26] and Tang [35] methods.

the three cases, the difference of the completion result of the text-removed video and the original video sequence is observed by computing the mean square error (MSE) value for the corresponding frames of the resulting and the original video sequences. Fig. 4.11(a) shows a frame of the video chosen for evaluation to which a text is added manually as in Fig. 4.11(b) and then completed as illustrated in Fig. 4.11(c).

The plot indicated as Bandlet-based in Fig. 4.12 shows MSE graph of all the 50 frames of the text-removed video and the completion result sequence using the bandlets. Both of Contourlet-based and Wavelet-based graphs, which indicate MSE values, resulted by respectively employing wavelets and contourlets in the proposed inpainting scheme are above Bandlet-based. This implies that the bandlet transform is more effective than the other sparse representations once used in our proposed regularization-based video inpainting approach.

Comparison with Existing Video Inpainting Techniques

Our video inpainting approach is compared with two video completion methods introduced in [26] and [35]. We performed the same MSE graph generation i.e., computing MSE for the inpainting results of the text-removed video sequence (Fig. 4.11(c)) and the original sequence (Fig. 4.11(a)). The output graphs are depicted in Fig. 4.13. These graphs indicate a high performance for our proposed approach compared to these two methods. This high performance is largely credited to the effective role of bandlets in the spatio-temporal inpainting.

Chapter 5

Video Super Resolution

5.1 Introduction

A high resolution (HR) image contains more details compared to the same image with a low resolution (LR) because of the higher pixel density of the HR image. As a result, images with high resolution are desirable in many applications. For example, analysis of satellite images to detect objects, moving object detection and recognition in surveillance videos, and online video streaming are other applications that rely heavily on HR images. Due to the constrains of image acquisition systems and/or storage limitations, it is not always feasible to have HR images. Hence, single image super-resolution (SR) was introduced for increasing the resolution of a given image to provide more visual information after up-scaling the image.

The most common methods for single image SR are based on interpolation [98, 99] since they have a low complexity. However, these methods generate artifacts in the resulting image such as blurring effect and zigzagging edges. To enhance the interpolation-based techniques, a reconstruction constraint is enforced in [100, 101] in order to have smoothed and/or down-sampled versions of the HR image close to the original image. There are several works that followed the idea presented in [102, 103] to preserve the edges in the SR process. The method introduced in [104] employs local image background/foreground

patches to reconstruct sharp discontinuities between them. The method in [105] benefits from a gradient profile prior for local image structures to be applied in SR. Machine learning based techniques, such as the ones proposed in [106], estimate the high-frequency details of the target image by a learning process in a set of natural images. Also, the method in [107] uses a Markov Random Field (MRF) to learn the prediction from LR to HR. The primal sketch priors are employed in [105] to enhance burred edges and corners even though the method relies on a large number of images as the training set. The scheme in [108] is performed via the sparse representation of the image generated by the image dictionary learning strategy introduced in [109]. This method trains two dictionaries for LR and HR patches, then, enforces the similarity between the LR and HR image patch pairs with respect to their own dictionaries. The approach presented in [110] analyzes the patch redundancy within different scales to recover at each pixel its best possible resolution increase.

Another category of SR techniques are performed in a transform domain (e.g., wavelet) of the image. The algorithm proposed in [111] first estimates the edge regularities by analyzing the decay of wavelet coefficients across scales and then the regularity is preserved by extrapolating a new sub-band used in synthesising HR image. In the same vein, the method introduced in [112] employs wavelet transform for image interpolation. Since the wavelet transform does not provide enough directionality and regularity details of the image, recently developed image transforms have been used for SR purposes. For instance, the SR technique proposed in [113] predicts the edge regularity of the HR image by analyzing the directionality details of the LR image well-captured by the contourlet transform. Also, the approach introduced in [114] employs the grouplet transform geometric details of the LR image to find the edge details and predict their behaviors in the HR image by analyzing the grouplet structure tensor.

The generalization of image SR to the problem of video SR is not trivial, since the temporal consistency needs to be preserved. A simple frame-by-frame application of the static image approach leads to unacceptable flickering artifacts. Therefore, efforts were directed to address the video SR task [115–118].

In this chapter, a new SR method based on the bandlet transform is proposed. Our

technique benefits from the advantages of the bandlets in effort to exploit the geometrical details of images. The new pixels in the up-scaled image are estimated by means of a regularization in the bandlet domain representation of the image. In fact, this scheme involves an optimization in the bandlet transform domain in order to inpaint the unknown pixels that lie among the original pixels of the enlarged image. Since spatial regularization-based techniques tend to cause blurring or over-smoothing effect in inverse problems, particularly in inpainting [2, 6, 13], we propose in this work an effective edge preserving scheme based on a structure tensor that can be generated by applying the geometric details revealed by the bandlet transform. Since the edge details are interpolated accurately based on their directionality, our method produces HR images produced after the inpaitning process with a considerable quality. The proposed SR method is discussed in Section 5.2. Then, the method is extend to the video frames in Section 5.3. Finally, in Section 5.4, the experimental results are presented.

5.2 Bandlet-based Image Super-resolution

Preserving high frequency details such as edges in an image is a challenge in a SR task. To tackle this problem, we propose a two-stage SR scheme. In the first stage, the edge information of the original image is captured by means of a structure tensor analysis which benefits from the local geometric details obtained by the bandlet transform. Then, in the up-scaled image the edge pixels are propagated to the missing neighboring pixels by considering the direction of the edge. This interpolation task is performed on the edge pixels of the image in order to avoid over-smoothing in the next step of the SR process. In the second stage, a regularization is performed in the up-scaled image. This spatial regularization is in fact an optimization in the bandlet domain representation of the resized image that approximates the new pixels in an inpainting fashion.

5.2.1 Edge Pixels Interpolation

A structure tensor is a matrix representation of partial derivatives information. In the spatial domain, it is typically used to represent the gradient or edge information of an image since

it provides a more powerful description of local patterns compared with the directional derivative [119]. The structure tensor is given by the matrix:

$$S = \begin{pmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{pmatrix} \tag{5.1}$$

where I_x and I_y denote the first order partial derivatives of image I along x and y, respectively. Eigen-decomposition of S yields eigenvalues (λ_1, λ_2) and eigenvectors (e_1, e_2). The vector e_1 represents a normal vector directed to the gradient edge and the vector e_2 is the tangent. In turn, the certainty of the gradient structure along the associated eigenvectors is indicated by the eigenvalues. These new gradient features provide a precise description of the local gradient characteristics.

Since the structure tensor relies on the local gradient which in turn needs accurate geometric (directionality) details of the textures, we modified Eq. (5.1) using the bandlet geometric details in order to benefit from more precise directionalities. As mentioned in Sec. 2.2, before the bandletization process a quad-tree segmentation is performed on the image based on the dominant regularity that exists in each region (Fig. 2.10b). Therefore, in this segmentation stage the regularity direction of each pixel flow is determined and represented by Γ (Sec. 2.2). Given this directionality information for each segmentation square, indicated by θ_Γ as an angle ranging from -90° to 90°, we modified Eq. (5.1) as follows:

$$\acute{S} = \begin{pmatrix} I_x^2 \cos^2(\acute{\theta}_\Gamma) & I_x I_y \cos(\acute{\theta}_\Gamma)\sin(\acute{\theta}_\Gamma) \\ I_y I_x \sin(\acute{\theta}_\Gamma)\cos(\acute{\theta}_\Gamma) & I_y^2 \sin(\acute{\theta}_\Gamma) \end{pmatrix} \tag{5.2}$$

where $\acute{\theta}_\Gamma$ is the 90° rotated version of the directionality degree of each segmentation square θ_Γ. The values $\cos(\acute{\theta}_\Gamma)$ and $\sin(\acute{\theta}_\Gamma)$ in Eq. (5.2) give a weight to the edge pixels with a regularity along θ_Γ in each region. Thus, the eigen-decomposition of $\acute{S}$ results in characteristics of edges corresponding to the geometrical flows that exist within the image. Hence, edges and high frequency details are characterized more effectively. The norm of $\acute{S}$ in terms of eigenvalues, i.e, $||\acute{S}|| = \sqrt{\acute{\lambda_1} + \acute{\lambda_2}}$ identifies the edge pixels in the image as shown in Fig. 5.1b. The edge map of the image can then be found by thresholding $||\acute{S}||$.

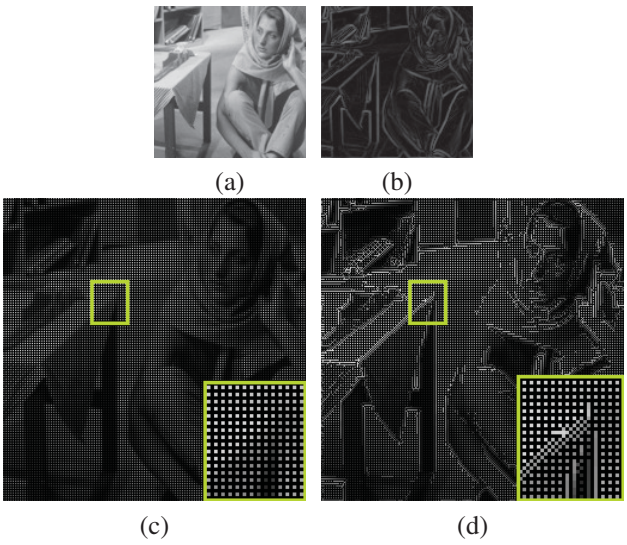


Figure 5.1: (a) Barbara image. (b) $||\acute{S}||$ of structure tensor (Eq.(5.2)). (c) Up-scaled image to be filled in with appropriate pixels. (d) Edge pixels interpolation result.

Now the task is to assign appropriate values to the new pixels of the up-scaled image that lie on the edges and high frequency image components. Let $\acute{I}$ be the scaled image by 2 (Fig. 5.1c), then $\acute{I}(2x, 2y) = I(x, y)$ and $\acute{I}(2x-1, 2y-1) = 0$. If the corresponding value of the pixel $\acute{I}(2x, 2y)$ in the edge map is 1, then this edge pixel needs to be properly continued to the adjacent pixels. From the eigen-decomposition of Eq. (5.2) we have the tangent vector of the edge pixel at $I(x, y)$. Given the tangent vector as a degree value ranging from -90 to 90 denoted by α, the pixel value at $\acute{I}(2x, 2y)$ is assigned to any of its 5 neighboring pixels as follows:

$$\begin{array}{ll} \acute{I}(x-1, y) = \acute{I}(x, y) & \text{if } -25 \leq \alpha(x, y) \leq 25 \\ \acute{I}(x+1, y) = \acute{I}(x, y) & \text{if } -25 \leq \alpha(x, y) \leq 25 \\ \acute{I}(x-1, y+1) = \acute{I}(x, y) & \text{if } -60 \leq \alpha(x, y) \leq -30 \\ \acute{I}(x, y+1) = \acute{I}(x, y) & \text{if } -90 \leq \alpha(x, y) \leq -65 \text{ or} \\ & 65 \leq \alpha(x, y) \leq 90 \\ \acute{I}(x+1, y+1) = \acute{I}(x, y) & \text{if } 30 \leq \alpha(x, y) \leq 60 \end{array} . \tag{5.3}$$

This edge continuing scheme ensures the preservation of high frequency details of the image once the image is enlarged, as can be seen in Fig. 5.1d. In the next step, the values of remaining pixels of the enlarged image are estimated via an inpainting process.

Algorithm 5.1 Bandlet-based image inpainting algorithm.

1: $i = 0$ and $Y^{(i=0)} = \acute{I}$
2: Find $\acute{Y}^{(i)}$ using Eq. (3.4)
3: Apply bandlet transform on $\acute{Y}^{(i)}$ (Sec. 2.2)
4: Update the estimate; $Y^{(i+1)} = T_{\lambda}^{B}(\acute{Y}^{(i)})$ (applying Eq.(3.16) on the bandlet coefficients of $\acute{Y}^{(i)}$ and performing inverse bandlet transform to generate $Y^{(i+1)}$)
5: $i \leftarrow i + 1$, if $|Y^{(i+1)} - Y^{(i)}| < \varepsilon$ stop, else go to step 2

5.2.2 Image Inpainting

As shown in Fig. 5.1d, many of the pixels of the enlarged image are not estimated yet, after the edge pixel interpolation step. These pixels can get new intensity values by inpaitnting the up-scaled image. An inpainting task is performed quite effectively if it employs the geometrical features of the image. Since bandlets provide us with practical local geometric characteristics, we take advantage of them to inpaint the enlarged image.

The input of an inpainting process is an image Y that contains some missing pixels. The missing pixels are often connected and make a blank region (or several regions). As defined in Sec. 3.1, let Ω be the set of missing pixels and $\Phi = Y \setminus \Omega$ be the source pixels. The goal is to find a new image $\acute{Y}$ such that $\acute{Y}(x, y)$ is equal to $Y(x, y)$ for the pixels that belong to Φ, i.e., $\acute{Y}(x, y) = Y(x, y)\ \forall (x, y) \notin \Omega$. At the same time, the overall geometry of $\acute{Y}$ is supposed to have the same geometrical regularity as the original image Y in Φ. Considering the new unassigned pixels of the enlarged image as the missing pixels, we modified the saptio-temporal video inpainting scheme introduced in Algorithm 3.1 to be applied for single images as presented in Algorithm 5.1 to approximate the unassigned pixels. Therefore, in the inpainting process we have $Y = \acute{I}$.

5.3 Video Super-Resolution

The direct application of the proposed static image super-resolution scheme of Sec. 5.2 on videos in a frame-by-frame fashion results in disturbing artifacts. It was applied on several video sequences, and as expected there was a strong flickering effect in the static parts of the video frames. This kind of flickering occurs, largely because of the intensity

difference between the estimated corresponding pixels in two consecutive frames. Looking at each independently (frame-by-frame) super resolved frame, there is a visually pleasant result and no artifact is perceived by human eye. However, since each frame is processed individually, the resulting intensity values for corresponding pixels in these frames are different. Therefore, looking at the frames as in a sequence with an average frame-rate would cause flickering effect due to the lack of temporal consistency. We address this problem by taking into account the motion information among the frames and refining the super resolved pixels by using the exploited motion vectors.

5.3.1 Computing Motion

In order to find the motion vectors between two frames, the optical flow is computed based on the objective function proposed in [120], [121]. While it has discontinuities agreeing with object boundaries, the resulting flow field is supposed to be smooth. Let I_1 and I_2 be two images (frames) and the flow vector be $w(p) = (u(p), v(p))$ at $p = (x, y)$ which is the grid coordinate of images. Assuming there are L possible states for $u(p)$ and $v(p)$ (i.e., horizontal and vertical flows, respectively) and considering ϵ as a set of the four spatial neighbors, the optical flow energy function is defined by data term, small displacement term, and smoothness term as follows:

$$E(w) = \sum_{p} \min(||I_1(p) - I_2(p + w(p))||_1, t) \tag{5.4}$$

$$+ \sum_{p} \eta(|u(p)| + |v(p)|) \tag{5.5}$$

$$+ \sum_{(p,q)\in\epsilon} \min(\alpha|u(p) - u(q)|, d) \tag{5.6}$$

$$+ \min(\alpha|u(p) - v(q)|, d)), \tag{5.7}$$

The pixels to be matched along with the flow vector $w(p)$ are constrained by the data term Eq. (5.4). The flow vectors are constrained to be as small as possible by the small displacement term defined in Eq. (5.5). In addition, adjacent pixels are supposed to have

similar flow vectors. Therefore, they are constrained by the smoothness term, Eq. (5.6) and (5.7) along with a smoothing (regularization) factor of $\alpha > 0$. In the data term and the smoothness term the outliers are matched to the theresholds t and d, respectively. The use of coarse-to-fine search for optimization and the incorporation of stronger local constraints on the motion, result in impressive optical flow estimates [121].

5.3.2 Pixel Intensity Refinement

Once each frame is super-resolved individually using the image super-resolution technique introduced in Sec. 5.2, we need to take the motion information into account and refine the intensity values of the pixels that lie in the static regions of the frames to avoid flickering artifacts. With respect to the notation used for the still image super-resolution procedure, let I_t be a frame in the original video, and $\acute{Y}_t$ be its corresponding super-resolved frame, where $\acute{Y}_t(2x, 2y) = I_t(x, y)$. For $p = (2x-1, 2y-1)$, $p = (2x-1, 2y)$ or $p = (2x, 2y-1)$ in Y_t as the location of an estimated pixel, we consider $p_1 = (x, y)$, $p_2 = (x-1, y)$, $p_3 = (x-1, y-1)$ and $p_4 = (x+1, y)$ in I_t as the four adjacent neighboring locations of p at $\acute{Y}_t$. If there is no motion at p_1, p_2, p_3 and p_4 regarding the pervious frame I_{t-1}, one can consider that p belongs to a static region in the frame I_t. Therefore, the intensity of the estimated pixel in the enlarged (super-resolved) image must be similar to that of the corresponding pixel in the previous frame. To that end, we first calculate the sum of the magnitudes of the motion vectors obtained for p_1, p_2, p_3 and p_4 as follows:

$$M = |w(p_1)| + |w(p_2)| + |w(p_3)| + |w(p_4)|, \tag{5.8}$$

where w represents the motion vector found in Sec. 5.3.1. It should be noted that in the motion flow computation, the previous frame I_{t-1} is considered as the reference frame. Then, if the value of M is smaller than a small value δ, there assumed to be no motion at p. Hence, the pixel value at p is updated as: $\acute{Y}_t(p) = (\acute{Y}_{t-1}(p) + \acute{Y}_t(p))/2$.

Algorithm 5.2 presents all the steps required to generate a super-resolved video. As discussed in Sec. 5.4, this algorithm results in a high quality spatially enlarged video with a high temporal consistency.

Algorithm 5.2 Video spatial super-resolution.

Video sequence: $\{I_1, I_2, ..., I_t, ..., I_N\}$
Generate $\acute{I}_1$ (Sec.5.2.1)
Generate $\acute{Y}_1$ (Sec.5.2.2)
for t=2 to N **do**
  Generate $\acute{I}_t$ (Sec.5.2.1)
  Generate $\acute{Y}_t$ (Sec.5.2.2)
  for all x and y in $I(x, y)$ **do**
    if $p = (2x - 1, 2y - 1)$ **or** $p = (2x - 1, 2y)$ **or** $p = (2x, 2y - 1)$ in Y_t **then**
      if $M < \delta$ Eq.(5.8) **then**
        $\acute{Y}_t(p) = (\acute{Y}_{t-1}(p) + \acute{Y}_t(p))/2$
      end if
    end if
  end for
end for

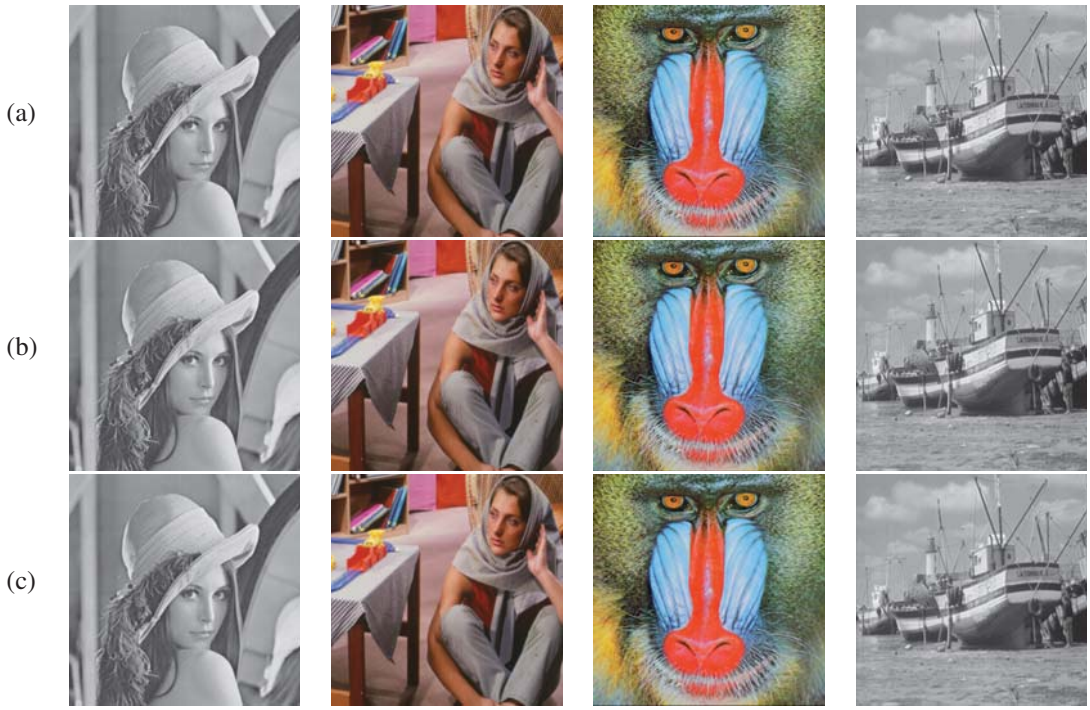


Figure 5.2: SR result on different images. (a) Original images. (b) HR results obtained for 256×256 inputs. (c) HR results obtained for 128×128 inputs.

5.4 Experimental Results

5.4.1 Image Super-Resolution

Our image SR method discussed in Sec. 5.2 is tested on several standard images, including Lena, Barbara, Baboon and Boat each of which has a size of 512×512. In order to evaluate

the accuracy of the proposed approach, the original image is once down-sampled by a factor of 2 and another time by a factor of 4; then the down-sampled images are enlarged using the SR technique to the original size (512×512). This way, after the SR task for each sample image we have 2 different 512×512 versions, one is the original image, the other is the resulting HR image. The effectiveness of our algorithm is evaluated by calculating the PSNR. Fig. 5.2 illustrates the result of our algorithm on four different images. It is worth mentioning that, in the case of dealing with a color image, the image is first transformed from RGB to YCbCr, then since the Cb and Cr components contain only low-frequency details, these components are interpolated (bi-cubic). Thus, the SR algorithm is carried out only on the Y (intensity) component. It is also noteworthy that for an image to be up-scaled by a factor of 2 the algorithm, i.e, the two steps discussed in Sec. 5.2.1 and 5.2.2, is performed once. Therefore, for larger scales it can be done iteratively until we get the desired image size so was it done to enlarge the 128×128 images in our experiments.

To gain further insight into the effectiveness of our proposed SR technique, we compared it with two other methods ([110] and [105]). These methods were used to enlarge the sample images shown in Fig. 5.2 in a same way done by our approach. PSNR values are calculated for the results of each method and reported in Table 5.1. As the table indicates, our technique outperforms the other two methods, which is quite reasonable since our approach benefits from precise local image geometric details revealed by an effective image representation, i.e, bandlet transform.

5.4.2 Video Super-Resolution

The proposed spatial video SR technique (Sec. 5.3) is tested on several video sequences. Similar to the experiments carried out for the still images, we first down-sampled the original videos by 2 and 4, then performed Algorithm 5.2 on the down-scaled videos to produce HR videos with the same size as that of the original videos. Fig. 5.3 illustrates sample frames of two tested videos and the resulting HR frames. The original size of these videos was 352×288, so the SR task was performed on the 176×144 and 88×77 frames. Since we have both the original 352×288 frames and the 352×288 super-resolved frames we can use PSNR to evaluate our approach. The sample video shown in Fig. 5.3(d) was tested by the

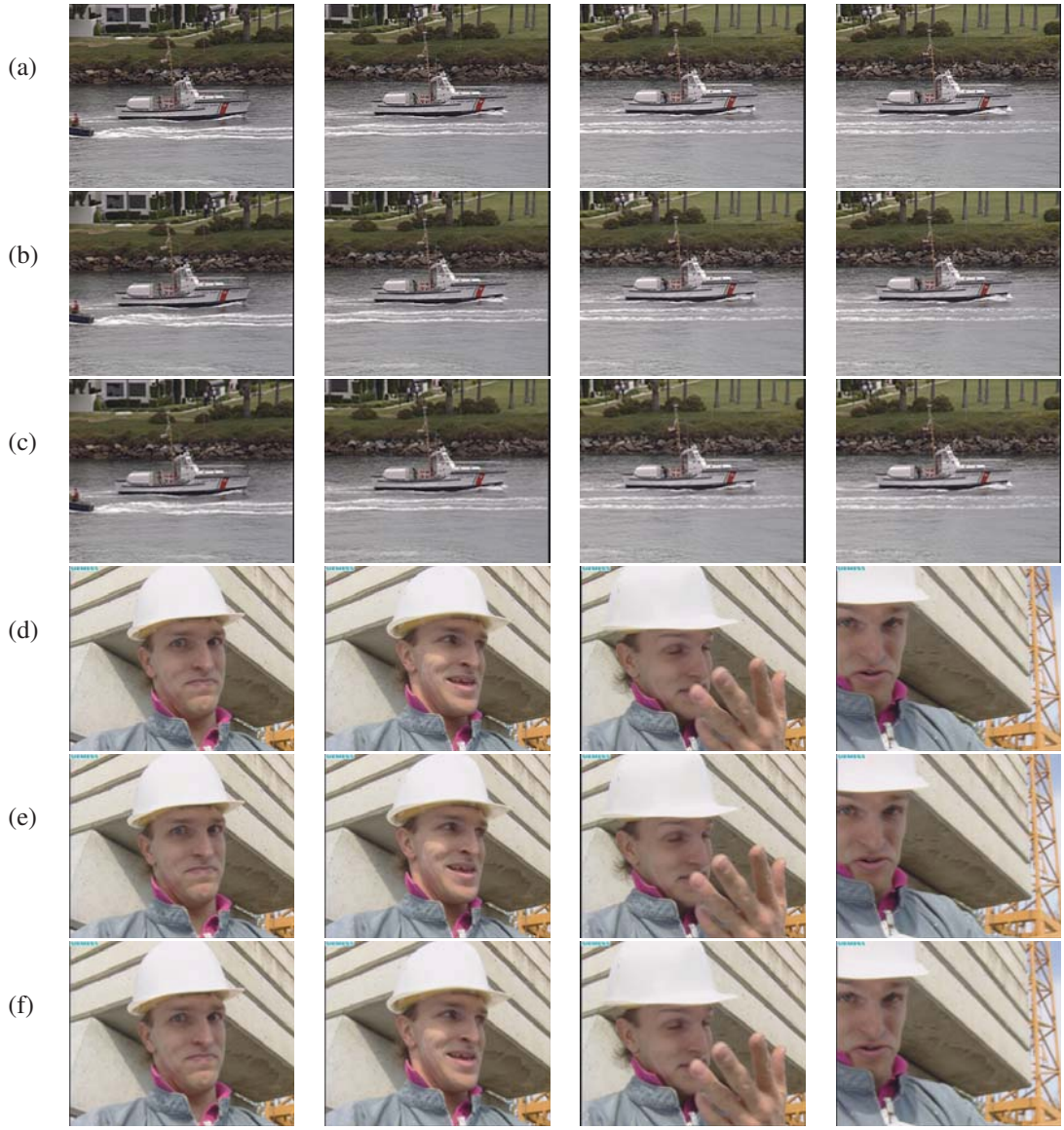


Figure 5.3: SR result for different frames of two video sequences. (a)(d) Original frames. (b)(c) HR results obtained for 176×144 inputs. (d)(e) HR results obtained for 88×77 inputs.

video spatial SR methods introduced in [115, 116] to compare their results with the results of our algorithm. The PSNR graphs of all the resulting $\times 2$ enlarged (HR) frames are shown in Fig. 5.4 for all the three methods. As these graphs indicate, our SR technique has a better performance in terms of PSNR compared with the other two techniques. As discussed in Sec. 5.3, having reasonable visual pleasant results is largely due to preserving the temporal consistency in the resulting super resolved videos. Therefore, we evaluated the temporal consistency of the HR videos. We employ the spatio-temporal most apparent distortion

Table 5.1: PSNRs of HR images obtained by different methods.

	PSNR Values (dB)		
Image	Our Method	[110]	[105]
Lena (256×256)	37.69	32.45	31.89
Barbara (256×256)	26.66	24.90	25.02
Baboon (256×256)	25.67	23.32	23.61
Boat (256×256)	32.97	29.10	28.63
Lena (128×128)	34.09	30.38	29.31
Barbara (128×128)	24.28	22.73	21.82
Baboon (128×128)	23.07	22.91	20.74
Boat (128×128)	29.93	27.11	27.23

Table 5.2: Temporal consistency evaluation. STMAD obtained for each resulting video using different SR techniques.

Method	Bishop [115]	Shan [116]	Ours
STMAD (176×144)	0.492	0.551	0.671
STMAD (88×77)	0.413	0.484	0.601

(STMAD) model to analyze our approach with regards to temporal consistency [58, 59]. Table 5.2 presents STMAD values obtained for the super-resolved videos by the three different techniques. The STMAD is calculated between the resulting videos and the original one of Fig. 5.3(d). The obtained values are normalized to the range of 0 to 1 and then they are subtracted from 1. Hence, a higher value in the table indicates a better consistency. According to this table, our approach has a better STMAD compared to the other methods. Consequently, the best temporal consistency is provided by our proposed method. This high performance both spatially and temporally (in terms of PSNR and STMAD, respectively) is largely credited to the effective role of bandlets in the edge preserving and spatial inpainting schemes along with taking into account the motion information of frames in order to preserve the temporal consistency.

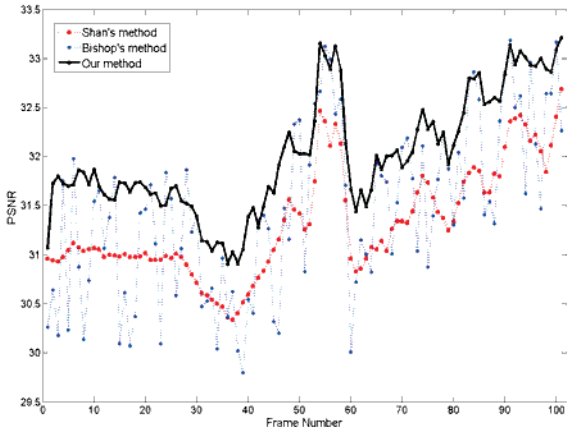


Figure 5.4: Objective evaluation of the proposed, Bishop [115] and Shan [116] video SR methods.

Chapter 6

Conclusions

We presented a video inpainting approach that effectively benefits from the geometric features represented by bandlets. The conventional exemplar-based video completion is modified and followed by a 3D regularization in order to perform the inpainting task. The patch search is carried out using the pixel values and instantaneous motion information. Then, the best matching patches are blended by a bandlet-based fusion framework to fill in the border patch. The fusion procedure employs the geometric flows and texture structures revealed by the bandlet transform. Afterwards, since some patches remain unchanged in the generated video, a 3D regularization based on bandlets refines the inpainting results. This is performed by enforcing the sparseness of the bandlet image representation through a minimization over the bandlet coefficients. The minimization is done iteratively by a soft-thresholding scheme in the video volume. Unlike many existing video completion methods, our approach does not require background/foreground segmentation, decomposition of motion layers, tracking and/or optical-flow mosaics computation. Moreover, the experimental results show a high performance of our video inpainting approach in preserving the spatio-temporal consistency, and consequently in reconstructing the videos visually pleasingly.

As an application of video completion, we presented an automatic video text removal scheme which involves an automatic video text detection and a practical inpainting stage. The video text detection technique strictly relies on an accurate text detector for single

frames. We proposed a connected component-based image text detection developed as an unsupervised clustering scheme. A feature vector based on properties extracted from stroke width transform connected components, distinct characteristics of text components that exist in the image and their general geometry is formed. Since the accuracy of the image text detector depends on precise edge locations to generate the connected components, we employed the properties of bandlet transform in representing local image geometry and introduced a novel edge detection approach which is quite adapted to edge locations of texts embedded in various types of images. The detected text regions in all the frames are tracked in the entire video sequence in order to locate the video text and distinguish it from the other parts of the video. The text is removed and the video is restored using the proposed inpainting technique which performs 3D regularization in the bandlet domain. The experimental results indicate a high performance of our video inpainting approach in the case of text removal and crucial role of bandlets in reconstructing the videos visually pleasingly. Also, the results indicate a considerable performance for both the bandlet-based edge detector and video text detection scheme.

As a second application of video completion, we proposed a single image SR technique that benefits from the bandlet transform in representing geometric features of images. The main stage of our proposed scheme is the inpainting task performed via the the image regularization technique to estimate the unknown pixels in the up-scaled image. In order to avoid over-smoothing in the HR images, prior to the inpainting process, the edge pixels and high frequency details of the image are interpolated in the up-scaled image. The edge propagation is performed using a structure tensor modified according to the image surface geometry captured in the bandlet transform diadic segmentation step. In order to maintain the temporal consistency in the HR videos, we took the motion information of the frames into account and performed a pixel intensity refinement to avoid the flickering effect in the static regions of the frames. Our approach performs the SR task with a considerable performance, as demonstrated by the experimental results.

The vast majority of the run-time of our video completion algorithm is spent on computing the bandlet transform, which lacks an optimized implementation since it is relatively new. Therefore, our first future work direction is to find possible solutions for optimizing

bandlet transform and reducing its complexity, specifically in the inpaiting task.

The completion task of the remaining holes or occluded regions after an undesired moving object removal can be accomplished more accurately if the moving object is separated from the background frames. In this case, the background can be filled-in easily using the available temporal information. Also, the occluded moving object can be completed much faster and more accurately since it is already separated from the background. This happens because the patch matching or interpolation process needs to be applied only on the foreground sequence, and an unnecessary search for the best match is avoided. Therefore, as another future work, we plan to apply motion layer segmentation or moving object tracking schemes, as done for example in [26] [29] [31] [32] [33] in conjunction with the proposed video completion method.

Sparse coding (modelling data vectors as sparse linear combinations of basis elements) and dictionary learning (learning basis set) were proven to be very effective for signal reconstruction and classification in the image processing domain [122–124]. Dictionary learning is performed by an optimization based on stochastic approximations to scale up to large datasets with millions of training samples. This can be effectively employed in video completion task. A future work direction in this field may yield practical and high performance solutions for spatio-temporal inpainting problems.

List of References

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proc. of 27th annual conference on Computer graphics and interactive techniques. (SIGGRAPH)*, pages 417–424, 2000.

[2] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (CVPR)*, pages I355–362, 2001.

[3] M. Bertalmio. Strong-continuation, contrast-invariant inpainting with a third-order optimal pde. *IEEE Transactions on Image Processing*, 16:1934–1938, Jul. 2006.

[4] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10:1200–1211, Aug. 2001.

[5] S. Masnou. Disocclusion: a variational approach using level lines. *IEEE Transactions on Image Processing*, 11:68–76, Feb. 2002.

[6] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13:1200–1212, Sep 2004.

[7] M. Elada, J.-L. Starckb, P. Querreb, and D.L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis. *Applied and Computational Harmonic Analysis*, 19:340–358, Nov. 2005.

[8] G. Peyre. Texture synthesis with grouplets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:733–746, Apr. 2010.

[9] Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:463–476, Mar. 2007.

[10] S. Mallat and G. Peyre. A review of bandlet methods for geometrical image representation. *Numerical Algorithms*, 44:205–234, March 2007.

[11] A. Mosleh, N. Bouguila, and A. Ben Hamza. Video completion methods: a state-of-the-art survey. *Artificial Intelligence Review*, Submitted on October 2012.

[12] A. Mosleh, N. Bouguila, and A. Ben Hamza. A video completion method based on bandlet transform. In *Proc. of IEEE International Conference on Multimedia and Expo. (ICME)*, pages 1–6, Barcelona, Spain, Jul. 2011.

[13] A. Mosleh, N. Bouguila, and A. Ben Hamza. Video completion using bandlet transform. *IEEE Transactions on Multimedia*, 14(6):1591–1601, 2012.

[14] A. Mosleh, N. Bouguila, and A. Ben Hamza. Bandlet-based sparsity regularization in video inpainting. *Journal of Visual Communication and Image Representation*, Submitted on July 2012, Revised on July 2013.

[15] A. Mosleh, N. Bouguila, and A. Ben Hamza. Image text detection using a bandlet-based edge detector and stroke width transform. In *Proc. of the British Machine Vision Conference*, pages 63.1–63.12, Sep. 2012.

[16] A. Mosleh, N. Bouguila, and A. Ben Hamza. An automatic inpainting scheme for video text detection and removal. *IEEE Transactions on Image Processing*, 22(11):4460–4472, Nov. 2013.

[17] A. Mosleh, N. Bouguila, and A. Ben Hamza. Image and video spatial super-resolution via bandlet-based sparsity regularization and structure tensor. *IEEE Transactions on Multimedia*, Submitted on August 2013.

[18] M. Ghoniem, Y. Chahir, and A. Elmoataz. Geometric and texture inpainting based on discrete regularization on graphs. In *Proc. 16th IEEE International Conference on Image Processing. (ICIP)*, pages 1349–1352, 7-10 Nov 2009.

[19] A. Elmoataz, O. Lezoray, and S. Bougleux. Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. *IEEE Transactions on Image Processing*, 17:1047–1060, Jul. 2008.

[20] T. Ding, M. Sznaier, and O.I. Camps. A rank minimization approach to video inpainting. In *Proc. of IEEE 11th International Conference on Computer Vision (ICCV)*, pages 1–8, Oct. 2007.

[21] R. S. Pena and M. Sznaier. *Robust Systems Theory and Applications*. Wiley & Sons, Inc., 1998.

[22] H. Grossauer and O. Scherzer. Using the complex ginzburg-landau equation for digital inpainting in 2d and 3d. In *Scale Space Methods in Computer Vision*, volume 2695 of *Lecture Notes in Computer Science*, pages 225–236. Springer Berlin / Heidelberg, 2003.

[23] A. N. Hirani and T. Totsuka. Combining frequency and spatial domain information for fast interactive image noise removal. In *Proc. of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 269–276, 1996.

[24] K.A. Patwardhan and G. Sapiro. Projection based image and video inpainting using wavelets. In *Proc. of IEEE International Conference on Image Processing. (ICIP)*, pages I857–I860, Sept. 2003.

[25] K. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting of occluding and occluded objects. In *Proc. IEEE International Conference on Image Processing. (ICIP)*, pages II69–72, 11-14 Sep 2005.

[26] K. A. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting under constrained camera motion. *IEEE Transactions on Image Processing*, 16:4545–553, Feb. 2007.

[27] T. K. Shih, N. C. Tang, W-S. Yeh, T-J. Chen, and W. Lee. Video inpainting and implant via diversified temporal continuations. In *14th annual ACM international conference on Multimedia*, pages 133–136, 2006.

[28] T. K. Shih, N. C. Tang, and J. N. Hwang. Ghost shadow removal in multi-layerd video inpainting. In *Proc. IEEE International Conference on Multimedia & Expo. (ICME)*, pages 1471–1474, Jul 2007.

[29] T. K. Shih, N. C. Tang, and J-N. Hwang. Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity. *IEEE Transactions on Circuits and Systems for Video Technology*, 19:347–360, March 2009.

[30] L. M. Po and W. C. Ma. A novel four-step search algorithm for fast block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:313–317, Jul. 1996.

[31] Y-T. Jia, S-M. Hu, and R. R. Martin. Video completion using tracking and fragment merging. *The Visual Computer*, 21:601–610, Sep. 2005.

[32] Y. Zhang, J. Xiao, and M. Shah. Motion layer based object removal in videos. In *Proc. Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTIONS'05)*, pages 516–521, Jan. 2005.

[33] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cut. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (CVPR)*, pages II972–979, 27 June-2 July 2004.

[34] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (CVPR)*, pages I120–127, 27 June-2 July 2004.

[35] N.C. Tang, C-T. Hsu, C-W. Su, T.K. Shih, and H.-Y.M. Liao. Video inpainting on digitized vintage films via maintaining spatiotemporal continuity. *IEEE Transactions on Multimedia*, 13:602–614, Aug. 2011.

[36] B. D Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conf. on Artificial Intelligence*, pages 674–679, 1981.

[37] T. Shiratori, Y. Matsushita, Xiaoou Tang, and Sing Bing Kang. Video completion by motion field transfer. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (CVPR)*, pages 411–418, june 2006.

[38] J. Jia, T-P. Wu, Y-W. Tai, and C-K. Tang. Video repairing: inference of foreground and background under severe occlusion. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (CVPR)*, pages I364–371, 27 June-2 July 2004.

[39] J. Jia and C-K. Tang. Image repairing: robust image synthesis by adaptive nd tensor voting. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (CVPR)*, pages I643–650, June 2003.

[40] J. Davis. Mosaics of scenes with moving objects. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (CVPR)*, pages 354–360, 1998.

[41] C-H. Ling, C-W. Lin, C-W. Su, Y-S. Chen, and H.-Y.M. Liao. Virtual contour guided video object inpainting using posture mapping and retrieval. *IEEE Transactions on Multimedia*, 13:292–302, April 2011.

[42] C-H. Ling, C-W. Lin, C-W. Su, H.-Y.M. Liao, and Y-S Chen. Video object inpainting using posture mapping. In *Proc. 16th IEEE International Conference on Image Processing. (ICIP)*, pages 2785–2788, 7-10 Nov 2009.

[43] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, apr 2002.

[44] E. Cands and D. Donoho. *Curvelets: A Surprisingly Effective Nonadaptive Representation of Objects with Edges*. Vanderbilt University Press, 1999.

[45] M.N. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on Image Processing*, 14(12):2091–2106, dec. 2005.

[46] D. Donoho. Wedgelets: Nearly minimax estimation of edges. *The Annals of Statistics*, 27(3):859–897, June 1999.

[47] E. Le Pennec and S. Mallat. Sparse geometric image representations with bandelets. *IEEE Transactions on Image Processing*, 14:423–438, April 2005.

[48] E. Le Pennec and S.Mallat. Bandelet image approximation and compression. *SIAM Multiscale Model. Simul.*, 4:992–1039, 2005.

[49] S. Mallat and G. Peyre. Surface compression with geometric bandelets. *ACM Trans. Graphics*, 24:601–608, July 2005.

[50] S. Mallat and G.Peyre. Orthogonal bandlets bases for geometric image approximation. *Commun. Pure Appl. Math.*, 61:1173–1212, July 2008.

[51] David L. Donoho and Jain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[52] A. Wong and J. Orchard. A nonlocal-means approach to exemplar-based inpainting. In *Proc. of IEEE International Conference on Image Processing. (ICIP)*, pages 2600–2603, Oct. 2008.

[53] Z. Xu and J. Sun. Image inpainting by patch propagation using patch sparsity. *IEEE Transactions on Image Processing*, 19(5):1153 –1165, May 2010.

[54] Z. Zhang and R.S. Blum. A categorization of multiscale-decomposition-based image fusion schemes with a performance study for a digital camera application. *Proceedings of the IEEE*, 87(8):1315–1326, Aug. 1999.

[55] X. Q. J. Yan, G. Xie, Z. Zhu, and B. Chen. A novel image fusion algorithm based on bandelet transform. *Chinese Optics Letters*, 5:569–572, Oct. 2007.

[56] J.L. Starck, M. Elad, and D. Donoho. Redundant multiscale transforms and their application for morphological component separation. *Advances in Imaging and Electron Physics*, 132:287–348, 2004.

[57] G. Pajares and J. M. de la Cruz. A wavelet-based image fusion tutorial. *Pattern Recognition*, 37(9):1855–1872, 2004.

[58] P.V. Vu, C.T. Vu, and D.M. Chandler. A spatiotemporal most-apparent-distortion model for video quality assessment. In *Proc. of IEEE International Conference on Image Processing (ICIP)*, pages 2505–2508, 2011.

[59] E. C. Larson and D. M. Chandler. Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, 19(1):011006–011006–21, 2010.

[60] T.-H. Tsai and C.-L. Fang. Text-video completion using structure repair and texture propagation. *IEEE Transactions on Multimedia*, 13(1):29–39, feb. 2011.

[61] M.N. Favorskaya, A.G. Zotin, and M.V. Damov. Intelligent inpainting system for texture reconstruction in videos with text removal. In *Proc. of International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 867 –874, oct. 2010.

[62] C.-L. Fang and T.-H. Tsai. Advertisement video completion using hierarchical model. In *Proc. IEEE International Conference on Multimedia and Expo. (ICME)*, pages 1557–1560, April 2008.

[63] W.-Q. Yan and M.S. Kankanhalli. Erasing video logos based on image inpainting. In *Proc. IEEE International Conference on Multimedia & Expo. (ICME)*, pages 521–524 vol.2, 2002.

[64] C. W. Lee, K. Jung, and H. J. Kim. Automatic text detection and removal in video sequences. *Pattern Recognition Letters*, 24(15):2607 – 2623, 2003.

[65] J.B. Kim, H.J. Kim, and S. Wachenfeld. Restoration of regions occluded by a caption in tv scene. In *Proc. of Conference on Convergent Technologies for Asia-Pacific Region*, volume 2, pages 817–820, Oct. 2003.

[66] D. Chen, J.-M. Odobez, and H. Bourlard. Text detection and recognition in images and video frames. *Pattern Recognition*, 37(3):595 – 608, 2004.

[67] M.R. Lyu, J. Song, and M. Cai. A comprehensive method for multilingual video text detection, localization, and extraction. 15(2):243 – 255, feb. 2005.

[68] L. Agnihotri and N. Dimitrova. Text detection for video analysis. In *Proc. of IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL)*, pages 109 –113, 1999.

[69] W. Kim and C. Kim. A new approach for overlay text detection and extraction from complex video scene. 18(2):401 –411, feb. 2009.

[70] H. Li, D. Doermann, and O. Kia. Automatic text detection and tracking in digital video. *IEEE Transactions on Image Processing*, 9(1):147 –156, jan 2000.

[71] R. Lienhart and W. Effelsberg. Automatic text segmentation and text recognition for video indexing. *Multimedia Systems*, 8:69–81, 2000.

[72] K. Jung, K. I. Kim, and A. K. Jain. Text information extraction in images and video: a survey. *Pattern Recognition*, 37(5):977 – 997, 2004.

[73] J. Liang, D. Doermann, and H. Li. Camera-based analysis of text and documents: a survey. *International Journal on Document Analysis and Recognition*, 7(2):84–104, Jul. 2005.

[74] M. Cai, J. Song, and M.R. Lyu. A new approach for video text detection. In *Proc. of IEEE International Conference on Image Processing (ICIP)*, pages I–117–I–120, 2002.

[75] C. Liu, C. Wang, and R. Dai. Text detection in images based on unsupervised classification of edge-based features. In *Proc. of Eighth International Conference on Document Analysis and Recognition*, pages 610–614, 2005.

[76] E. K. Wong and M. Chen. A new robust algorithm for video text extraction. *Pattern Recognition*, 36(6):1397–1406, 2003.

[77] K.I. Kim, K. Jung, and J. H. Kim. Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. 25(12):1631 – 1639, Dec. 2003.

[78] M.R. Lyu, J. Song, and M. Cai. A comprehensive method for multilingual video text detection, localization, and extraction. 15(2):243–255, Feb. 2005.

[79] J. Gllavata, R. Ewerth, and B. Freisleben. Text detection in images based on unsupervised classification of high-frequency wavelet coefficients. In *Proc. of International Conference on Pattern Recognition (ICPR)*, volume 1, pages 425–428, Aug. 2004.

[80] H. Li, D. Doermann, and O. Kia. Automatic text detection and tracking in digital video. 9(1):147–156, Jan 2000.

[81] Y. Zhong, H. Zhang, and A.K. Jain. Automatic caption localization in compressed video. 22(4):385–392, Apr 2000.

[82] H. Takahashi and M. Nakajima. Region graph based text extraction from outdoor images. In *Proc. of Third International Conference on Information Technology and Applications (ICITA)*, volume 1, pages 680–685, July 2005.

[83] X. Chen and A.L. Yuille. Detecting and reading text in natural scenes. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages II–366–II–373, June-2 July 2004.

[84] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2963–2970, June 2010.

[85] P. Shivakumara, T. Q. Phan, and C. L. Tan. A laplacian approach to multi-oriented text detection in video. 33(2):412 –419, Feb. 2011.

[86] H. Chen, S.S. Tsai G., Schroth, D.M. Chen, R. Grzeszczuk, and B. Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *Proc. of IEEE International Conference on Image Processing (ICIP)*, pages 2609–2612, Sep. 2011.

[87] Y-F. Pan, X. Hou, and C-L. Liu. A hybrid approach to detect and localize texts in natural scene images. 20(3):800 –813, Mar. 2011.

[88] M. Zhao, S. Li, and J. Kwok. Text detection in images using sparse representation with discriminative dictionaries. *Image and Vision Computing*, 28(12):1590–1599, 2010.

[89] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2008.

[90] S. Mallat and S. Zhong. Characterization of signals from multiscale edges. 14:710–732, 1992.

[91] W.G. Zhang, Q. Zhang, and C.S. Yang. Edge detection with multiscale products for sar image despeckling. *Electronics Letters*, 48(4):211–212, 16 2012.

[92] J. Tang, J. Kim, and E. Peli. Image enhancement in the jpeg domain for people with vision impairment. 51(11):2013–2023, nov. 2004.

[93] E. Peli. Contrast sensitivity function and image discrimination. *Journal of Optical Society of America*, 18(2):283–293, nov. 2001.

[94] E. Reinhard, P. Shirley, M. Ashikhmin, and T. Troscianko. Second order image statistics in computer graphics. In *Proc. of the 1st Symposium on Applied perception in graphics and visualization*, pages 99–106, 2004.

[95] G.R. Bradski. Real time face and object tracking as a component of a perceptual user interface. In *Proc. of Fourth IEEE Workshop on Applications of Computer Vision (WACV)*, pages 214 –219, oct 1998.

[96] S.M. Lucas. Icdar 2005 text locating competition results. In *Proc. of Eighth International Conference on Document Analysis and Recognition (ICDAR)*, pages 80–84, Aug.-Sep. 2005.

[97] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. Icdar 2003 robust reading competitions. In *Proc. of Seventh International Conference on Document Analysis and Recognition (ICDAR)*, pages 682 – 687, Aug. 2003.

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

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

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

[101] M. Ben-Ezra, Zhouchen Lin, and B. Wilburn. Penrose pixels super-resolution in the detector layout domain. In *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[102] K. Jensen and D. Anastassiou. Subpixel edge localization and the interpolation of still images. *IEEE Transactions on Image Processing*, 4(3):285–295, 1995.

[103] J. Allebach and P.W. Wong. Edge-directed interpolation. In *Proc. International Conference on Image Processing*, volume 3, pages 707–710, 1996.

[104] S. Dai, M. Han, Wei Xu, Y. Wu, and Y. Gong. Soft edge smoothness prior for alpha channel super resolution. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[105] J. Sun, J. Sun, Z. Xu, and H.-Y. Shum. Image super-resolution using gradient profile prior. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[106] Q. Wang, X. Tang, and H. Shum. Patch based blind image super resolution. In *in Proc. IEEE International Conference on Computer Vision*, volume 1, pages 709–716, 2005.

[107] W. Freeman, E. Pasztor, and O. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, 2000.

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

[109] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[110] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *Proc. of IEEE International Conference on Computer Vision,*, pages 349–356, 2009.

[111] W.K. Carey, D.B. Chuang, and S.S. Hemami. Regularity-preserving image interpolation. *IEEE Transactions on Image Processing*, 8(9):1293–1297, 1999.

[112] D.D. Muresan and T.W. Parks. Prediction of image detail. In *Proc. IEEE International Conference on Image Processing*, volume 2, pages 323–326 vol.2, 2000.

[113] C. V. Jiji and Subhasis Chaudhuri. Single-frame image super-resolution through contourlet learning. *EURASIP J. Appl. Signal Process.*, 2006:235–235, January 2006.

[114] A. Maalouf and M. C Larabi. Colour image super-resolution using geometric grouplets. *IET Image Processing*, 6(2):168–180, 2012.

[115] C.M. Bishop, A. Blake, and B. Marthi. Super-resolution enhancement of video. In *Proc. Artificial Intelligence and Statistics*, volume 3024, pages 25–36, 2003.

[116] Qi Shan, Zhaorong Li, Jiaya Jia, and Chi-Keung Tang. Fast image/video upsampling. In *Proc. ACM SIGGRAPH Asia 2008 papers*, pages 153:1–153:7, 2008.

[117] V. Cheung, B. Frey, and N. Jojic. Video epitomes. *International Journal of Computer Vision*, 76(2):141–152, 2008.

[118] M. Ben-Ezra, A. Zomet, and S.K. Nayar. Video super-resolution using controlled subpixel detector shifts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):977–987, 2005.

[119] S. Di Zenzo. A note on the gradient of a multi-image. *Comput. Vision Graph. Image Process.*, 33(1):116–125, January 1986.

[120] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proc. European Conference on Computer Visiom*, volume 3024, pages 25–36, 2004.

[121] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[122] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. *CoRR*, abs/0809.3083, 2008.

[123] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008.

[124] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing,*, 15(12):3736–3745, 2006.