AN APPLICATION OF INPAINTING-BASED STILL IMAGE REANIMATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ECE SELİN BÖNCÜ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2017

Approval of the thesis:

AN APPLICATION OF INPAINTING-BASED STILL IMAGE REANIMATION

submitted by **ECE SELİN BÖNCÜ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** ____________

Prof. Dr. Tolga Çiloğlu
Head of Department, **Electrical and Electronics Eng.** ____________

Prof. Dr. Gözde Bozdağı Akar
Supervisor, **Electrical and Electronics Eng. Dept., METU** ____________

Examining Committee Members:

Assoc. Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering Department, METU ____________

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering Department, METU ____________

Prof. Dr. Çağatay Candan
Electrical and Electronics Engineering Department, METU ____________

Assist. Prof. Dr. Fatih Kamışlı
Electrical and Electronics Engineering Department, METU ____________

Prof. Dr. Pınar Duygulu Şahin
Computer Engineering Department, Hacettepe University ____________

Date: September 25, 2017

ABSTRACT

AN APPLICATION OF INPAINTING-BASED STILL IMAGE REANIMATION

Böncü, Ece Selin
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Gözde Bozdağı Akar

September 2017, 150 pages

A single-input system that converts still paintings into short video sequences is developed. The system is composed of four separate blocks that are Digital Matting, Inpainting, Motion Modelling & Synthesis and Rendering Blocks. Within the scope of this thesis, the performance analysis of all are realized and the input images are restricted to paintings so as not to cope with the difficulty of achieving photorealism that accompanies the usage of photographs.

The input image is partitioned into object-background layers in Digital Matting Block. For the extraction of alpha channels the algorithm proposed in [135] is used and it is tested for performance, especially on highly textured data, which paintings mostly tend to be, coupled with different trimap extraction methods.

Inpainting Block is used to cover up the holes in the background after object removal. A detailed analysis and experimentation is done on image inpainting techniques and their performance in the image-to-video converter system is discussed. Within this literature review, a classification based on which key elements are preserved through inpainting is stated, and by experimental verification, it is shown that the geo-texture preserving methods outperformed the methodology based on texture-preservation and geometry-preservation. A further comparison among geo-texture preserving method are made through visual performance and timing analysis in object removal, as well as tests on reconstruction of the known regions of several images, where the abilities of the algorithms in rebuilding the images are tested using several image quality metrics such as PSNR, SNR and SSIM.

Motion Modelling & Synthesis Block is responsible for creating an artificial motion field for the object layers. This bi-step procedure involves the sculpting the movement of certain group of objects into mathematical models that are produced by spectral filtering of random noise and generating time dependent displacement maps for each pixel in the image frame. Besides, in order to preserve the harmony within the image frame 2D adaptation of the 3D wind field in [174] is realized. The models implemented within the framework is restricted to passive objects, such as trees, clouds, bodies of water, etc. that move in harmonic oscillations as a reaction to the external drag forces such as wind and gravity.

In the ultimate step, Rendering Block, merges the layers back to create the frames of the video by using the transparency maps so as to obtain seamless composite images. The ordering of the frames in the temporal axis provides the flow of the video sequence as the final output.

Keywords: Inpainting, Reanimation, Object Removal, Motion Modelling

ÖZ

SAYISAL ROTÜŞ TABANLI BİR TABLO CANLANDIRMA UYGULAMASI

Böncü, Ece Selin
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Gözde Bozdağı Akar

Eylül 2017 , 150 sayfa

Sabit görüntüleri video dizilerine çeviren tek girdili bir sistem geliştirilmiştir. Sistem Sayısal Tabakalama, Sayısal Rötuş, Hareket Modelleme & Sentezleme ve Görsel Kaplama olmak üzere 4 ayrı bloktan oluşmaktadır. Bu tez kapsamında, her bir bloğun performans analizi yapılmıştır. Imge girdileri, fotoğraf kullanımının beraberinde getireceği foto-gerçeklik ilkelerinin başarımının zorluğunun önüne geçmek adına tablo imgeleriyle sınırlandırılmıştır.

Girdi imgesi Sayısal Tabakalama Bloğu'nda obje ve arkaplan olmak üzere tabakalara ayrılır. Alfa kanllarının hesaplanması sırasında [135]'da önerilen metod kullanılmıştır. Metodun performansı özellikle tablo imgelerinin de çoğunlukla ait olduğu zengin dokulu imgeler grubunda farklı tabaka haritası oluşturma yöntemleri kullanarak test edilmiştir.

Sayısal Rotüş Bloğu objelerin görüntüden çıkarılmasından dolayı oluşan boşlukların doldurulmasında kullanılır. Sayısal rötuş teknikleri üzerine detaylı bir inceleme sunulmuş, inceleme deneylerle desteklenmiş ve sabit görüntüleri video dizilerine dönüştüren bu sistemdeki performansları değerlendirilmiştir. Bu kaynak taraması içerisinde, sayısal rötuş yöntemlerinin uygulama sırasında imgenin hangi yapı taşını muhafaza ettiğini baz alan bir sınıflandırma yapılmış ve deneysel doğrulamalarla yapı-doku muhafaza eden yaklaşımların doku muhafaza eden ve yapı muhafaza eden yöntemlere kıyasla sayılsal rötuşta daha iyi sonuçlar verdiği gösterilmiştir. Yapı-doku muhafaza eden yöntemleri arasında obje yok etme uygulamaları kullanılarak görsel performans ve zaman analiz karşılaştırılması yapılmıştır. Ek olarak, aynı metodların imgenin bilinen kısımlarını yeniden oluşturma kabiliyetleri, Doruk Sinyal Gürültü Oranı (DSGO), Sinyal Gürültü Oranı (SNR), Yapısal Benzerlik Ölçütü (YBÖ) gibi bazı imge nitelik ölçütleri kullanılarak karşılaştırılmıştır.

Hareket Modelleme & Sentezleme Bloğu obje tabakaları için yapay hareket sentezlemekten sorumludur. Bu iki basamaklı işlem, rastgele gürültü alanının spektral süzgeçlenmesiyle oluşturulan hareket modelinin matematiksel olarak ifade edilmesini ve imge karesi içindeki her bir piksel için zamana bağlı bir yerdeğiştirme haritasının oluşturulmasını içerir. Bunun yanı sıra, imge karesinde ahengin sağlanması adına [174]'de verilen 3 boyutlu rüzgar modelinin 2 boyuta uyarlanması sağlanmıştır. Bu çalışmada gerçekleştirilen modeller ağaç, bulut, su kütlesi gibi, harici bir çekim kuvvetinin etkisi altında harmonik salınımlar şeklinde hareket eden pasif objelerle sınırlandırılmıştır.

Son aşamada, Görsel Kaplama Bloğu tabakaları, alfa kanallarını da kullanarak pürüzsüz bir şekilde tekrar bir araya getirerek video dizisinin karelerinin oluşturulmasını sağlar. Bu karelerin zaman ekseninde sıralanasıyla son çıktı olarak video dizisi oluşturulur.

Anahtar Kelimeler: Sayısal Rotüş, Canlandırma, Obje Yok Etme, Hareket Modelleme

$n\prime, \quad n^i, \quad n^i_\ell \quad \mathcal{N}\{\mathcal{A}\}$

When n^4 reaches infinity...

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my thesis supervisor Prof. Dr. Gözde Bozdağı Akar for her support, wisdom and endless patience. With her guidance and her experience, I was able to become acquainted with research areas that I was delighted to work in and topics that boosted my enthusiasm. I always felt privileged to be able to work as her student throughout the entire studies.

I would also like to thank Prof. Dr. A. Aydin Alatan for his technical insight and making room for me in his lab, Multi Media Research Group which allowed me to work with talented researchers during the the course of my thesis work. It has been a great pleasure to be a member of a highly qualified academic community where I had to opportunity to work with and meet people that are very dear to me. First of all, I would like to thank Emin Zerman for his mentorship since the day I was introduced to the lab, his unceasing support and for sharing his appreciation of life as a whole, Yeti Ziya Gürbüz for his words of wisdom [28], his frankness and always cleaning up after me when I fail, Akın Çalışkan and Özümcan Demir Çalışkan for always sparing a coach for me in their lovely flat where I always felt at home and for cheering me up when in need, Ömürcan Kumtepe for all the laughs and weird moments that we shared, Emrecan Batı for his omnipotence as an engineer and being there for technical and non-technical support, to İlker Buzcu for clearing things out when we are desperately in need of explanations, Oğul Can for being my partner in crime, Enis Kobal for being eager to listen to every story that I tell him and for always standing beside me, Melis Özateş Gürbüz for being wiser than the wise man and the thesis template which still holds her name, and Harika Başpınar Zerman for hand-crafting me a Boticelli which indeed was one of the main inspirations of this work. Moreover, I also would like to acknowledge Dr. Alper Koz for his immense knowledge and İzlen Geneci, Aybüke Erol, Bahar Taşkesen and Emre

Can Kaya as great teammates.

Apart from my fellow labmates, there are certain people who have helped me throughout the whole phase of writing. I can't thank enough to Selahaddin Topselvi for lending me his own gadgets of writing, to Sinan Yakupoğlu for providing me a second home whenever I needed, to Cem Mehmethanoğlu for being the saviour at the eleventh hour, to Tuğçe Şeref for hosting me the day before the presentation and calming me down and İlhan Yaman for staying up all night with me in the final throes.

Furthermore, there are several other people that who provided the utmost support that I required while achieving determination to continue, I cannot pass without mentioning them, though not explicitly by name, but through some *mise en scene.* I would like to express my heartfelt thanks to the ones that we called each other *"home"*, to the ones with whom we always ended up going on *"our last vacation"*, and to the ones that we had all the summertime sadness together. Also, the ones who were there in all car crushes, in significant measurements and in dreams of existence, the ones with whom we climbed to the rooftops, the ones who prefer to eat *cosmetics* along with sushi, the ones who proved to the true band of brothers, the ones with whom we shared a bench, a stage or a toolbox and the ones who suffered from being *pisces*, I am absolutely grateful to you. I repeat once again that I appreciate and cherish all the moments we have shared so far and the ones that are yet to come.

Last but not the least, I dedicate my thesis to four people that mean a lot to me, namely to my family. I declare my gratitude to my grandfather, **N**ail Kumcu, with whom I had very little chance to share moments, but it was his memoirs that gave me the inspiration to achieve what I have done up to this moment. I would like to thank to my one and only uncle Akgün **N**ecati Kumcu, for his endless support in whatever I intend to do and being one of the funniest people I have ever met, to my dearest grandmother, **N**imet Kumcu, who passed away in the course of the thesis work, for being the joy of my life and still making me smile while recalling her memory. My final and utmost gratitude is to my mother, **N**ilgün **N**ilay Kumcu, who has always been my very best friend from the day I was born. My appreciation of her unconditional love, her support and her efforts to help me both during the thesis and in life as a whole is beyond words.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BV	Bounded Variation
CSH	Coherence Sensetive Hashing
CM	Criminasi Method
EM	Expectation Maximization
FH	Free Hand
I-CHS	Inpainting with CSH
I-IM	Inpainting with Image Melding
I-PM	Inpainting with Patch Match
IS	Intelligent Scissors
IM	Image Melding
MRF	Markov Random Field
MSE	Mean Square Error
NN	Nearest Neighbour
LS	Least Squares
PCA	Principal Component Analysis
PDE	Partial Differential Equation
PM	Patch Match
POV	Point of View
ROI	Region of Interest
SSD	Sum of Squared Distances
TV	Total Variation
UI	User Interface

CHAPTER 1

INTRODUCTION

1.1 Introduction to Computer Animation

The earliest computer animation dates back to 1940s and 1950s that coincide with the pioneering experiments on computer graphics, and hereafter, with the outspread launch of the digital computers throughout the world, numerous innovatory fields flourished for the development of computer animation [81]. Even though initially computer graphics and animation were mainly captured the attention of the academic researches of the natural sciences or engineering upbringings, it did not take long for them to become the quintessence of artistic purposes. Eventually, by the end of 70s, animated scenery had already come to be frequent in public and mass media. From then on, there have been an ever-growing interest to achieve more on this artificially created sequences.

The foremost emphasis on the amelioration of the computer graphics and animation, *in genere*, was the pursuit to achieve photo-realism. With the exponential improvement of the computational power of the devices used in this industry, further plausible visuals were possible to be produced. In the first place, basic computer-aided special effects were started to be used in feature films, there blossomed the opportunities to generate movies in full animation out of no footage at all, but with the computer aided techniques that has been developed so far. It is also possible to claim that the gigantic empire of computer games has co-evolved with computer animation. Nowadays, video games absolutely delicate graphics and embody versatile motion of characters so as to make the player hallucinate the surrounding in a more intensive way. In nutshell, it is to be said that, in the time being, that is to say, computer animation evolved from being a short-length sequence of individually sketched black & white frames in to long duration videos in which motion and the appearance of what is present in is hard to distinguish from real world.

Throughout the journey of computer animation in accordance with photo-realism, two fundamental aspects of a video footage play a huge role to imitate the real world. First of which is to necessity to design the objects present in the scene, from the human body, animals even to buildings, to household furniture, and to pieces of nature, totally depending on the desired content of the animation in a proper manner. Without loss of generality, each and every object in the frame are expected to comply with the rudimentary features of their real life archetypes, in terms of shape, colour, texture, etc. Additionally, these scene objects are expected to be in a harmony with the rest of the objects, to obey the principle of perspective, *id est.*, each frame to form the animation sequence are to be a separate realistic representation of a real life scene. Precisely, at this very point here comes the second aspect to play the part, exactly in the transition between the animation frames that creates, indeed, the motion. The motile objects of the animation, which will be called as figures of action, are required to replicate the authentic motion of their counterparts so that the animation does not look unnatural. While modeling the motion of a figure of action, its physical properties, its body structure, motile and still parts of the body, the type of motion and its interaction with the rest of the scene can be counted as some items to be paid attention to.

Computer animation has its origins in cartoon animation in which only 2 dimensional illustrations are utilized, however, as the computational powers increased and the researches on the area broadened, as a means of increasing photo-realism in the animated scenery, the 3rd dimension is introduced in the newer works. From then on, instead of drawing the objects, the animators began to sculpt them. Using 3D modeling on computer animation provided the scene to appear more than a projection on the image frame, but rather fed the completeness of each object present in the frame, with its enhanced geometry and shading.

Although seeking photo-realism has whipped up the interest of the researchers in the specific study, contributed to the procreation of astonishing animated visuals in movies, computer games, augmented reality applications, etc. and hence prompted computer animation to be involved in every part of the daily life. Regarding the diversity of the applications, some developers tend not to achieve extremely realistic animation due to the limitations of media to broadcast or display the output. Likewise, it may prefer to be eliminated in the output and even additional special effects may be presented for merely artistic purposes. Within the scope of this thesis, an application of single frame animation will be discussed throughout the chapters. The main idea is to create short animated videos merely using single frame images as inputs and limit the human interaction in the production procedure as much as possible.

The fundamental motive behind the application is to imitate the interaction between the human visual system and memory. For humans beings, the semantic classification of the images is done with further ease than the classification of than words [122]. When a single frame photograph is seen, although the image stands still, displaying only a single instance of the plot of events in that setting, the brain has the capability compensating for the missing information using its past experiences. Briefly, the human brain has the tendency to gather the elements, namely objects, persons, environment, etc. that are present in a scenery, draw a connection between these elements, figure out the setting and the occasion, hallucinate the motion that happen to be taking place and generate a short story out of all these material. For instance, when a glance is taken of the photograph in Figure 1.1, one is prompted immediately to assume that the woman is jogging on the edge of a cliff on a windy day. How this conclusion is derived is quite trivial, indeed, the posture of the woman and the position of her legs leads to the idea that she is running; however; she does not seem to be terrified to be trying to escape from anything coming behind. Also, her clothes suggest the fact that she is performing some branch of sports at the moment. Moving on the environment, the way that the way that the trees and bushes are bent allow the observer to do the weather forecast, it is windy out there!

Along with the entire effort devoted to seek photo realism, as computer ani-

Figure 1.1: An illustration of perception

mation began to capture the eyes of the artists, computer animation was commenced to be incorporated into a multiplicity of branches of art. Reanimation of still images can be counted as one of these artistic interests. A major and essential example was the exhibition that toured around Europe in 2015, which was called Van Gogh Alive, where his paintings were digitally manipulated to turn them into short animation pieces to be projected on the walls of the exhibition hall.

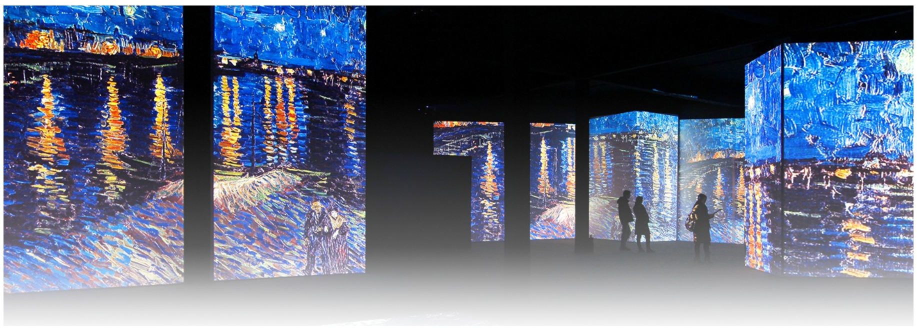


Figure 1.2: An instance from the exhibition *Van Gogh Alive* in 2015

The Exhibition of Van Gogh Alive was one of the most prominent influences to cater the idea behind this work, after observing how beautifully these works of art can contribute to the birth of brand-new pieces of art. Immediately, it sparked the idea that many other paintings could possibly be the domain for reanimation.

Taking the story-telling characteristic of the human neural system as the basis, and the aforementioned exhibition as the main inspiration, a system that generates stories out of single images is tried to be developed. Nevertheless, this goal is a tough one containing numerous problems to be attacked. Concerning the fact that what is tried to be achieved is a single-input-single-output system, it is almost impossible, unless one is bestowed with a magic wand, to animate it at a step. The obstacles to prevent this are listed as follows:

An image is a whole entity, containing a number of objects both in the background and foreground, and these objects are required to be separated from the rest of the image and from each other, since each and every object tends to obey distinct motion characteristics. Even a certain number of them may be preferred to be kept as still.

When the objects are layered individually, reminding the fact that these layers will have motion inserted on them, there occurs the necessity of filling the holes that is left due to the removal of the objects in accordance with the remainder image so that no blank pixels are apparent on the final output frames.

To create the animation, a motion model is needed to be synthesized for the objects of interest, however; the type of motion and structure of the moving body are to be considered while fitting the design to make the action in the video realistic. The separated and animated layers of the image are to be juxtaposed once again seamlessly so as to fake that they have never been layered.

In order to fulfill the requirements of the problem, a converter model is used to be used that consists of blocks, amending the stated concerns, respectively.

1.2 Scope and Outline

In this thesis, a system that converts still paintings into short video sequences is developed. The block diagram of the system having four main steps, namely, Digital Matting, Inpainting, Motion Modelling & Synthesis, and Rendering, is given in Figure 1.3. A similar framework for 2D animation is proposed in [37]; however, in this research, we provide a detailed analysis on each block.

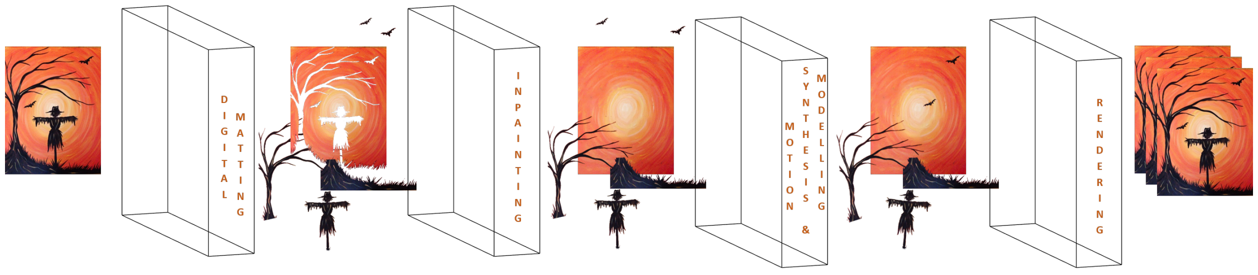


Figure 1.3: The Block Scheme of the Still-Image to Video Conversion System

The input, namely the single frame image enters the converter system, is processed initially by the **Digital Matting Block**. This very section is the part that the image is converted into a layered representation of foreground and background objects. In Chapter 2, the necessity of a layered representation is discussed, a brief introduction to detailed image matting and its difference from segmentation is given and the details of the used algorithm [135] is provided.

After matting of the image is completed, the individual layers enter the second block,namely the **Inpainting Block**, where the image refilling procedure takes place. In this phase of the conversion, the blank parts of the background layers are painted with a content aware means so that the newly generated regions appear to be coherent with the remainder image. The entanglement of inpainting in reanimation applications is one of the core problems to be discussed throughout the thesis. In the corresponding chapter, Chapter 3, primarily the problem is defined properly and its vitality this image-to-video converter system is asserted. This very chapter functions as a detailed survey on how inpainting approaches have developed so far. The analysis also includes the implementation of a number of approaches in literature and experimentation on their performances based on quantitative and qualitative metrics is provided.

The penultimate block, **Motion Modelling & Synthesis Block** of the system is responsible for the bistep method for the creation of the sense of movement in the video sequence. Within the MMS Block, modelling motion is done through spectral filtering and for the motion synthesis part, the generation of a motion armature in the form of a 5D displacement map, $DM(x, y, t, \partial x, \partial y)$, is explained. Chapter 4 is dedicated to the explaination of how these kinetic models are determined, and then the method how the motion is applied on the images. Within the chapter, the approach to true modelling of motion and the representational power of stochastic animation is discussed first. The implemented models are explained and also this chapter includes the generalization of the coherency preservation method of Zhang et al.'s algorithm in harmonious forest animation in 3D [174], which is adapted to 2D animation to be used in the scope of this thesis work.

The **Rendering Block** is the forth and the last of the blocks to be visited before the conversion is completed. The layers that belong to the same animated video frame is remerged to a single image frame that is seamless enough to look unpartitioned and the procedure is repeated for each time instance in the displacement map. When the entire group of video frames are tiled along the time axis, they form the video sequence itself. Chapter 5 details the procedure of rendering, both in formation of a single frame and then the frame ordering in the video sequences as a whole.

Yet the overall setup is determined to achieve a conversion from still images to video sequences, to facilitate the implementation, certain restrictions are needed to be applied on the images that will be allowed as inputs to the system. As the algorithm has the intention of making paintings *come alive*, the application of reanimation on photographs is not taken into the scope of this thesis work. Undoubtedly, the photographs in the input domain require detailed processing and longer intervals of computation to achieve a desirable visual due to the before explained concern of photo-realism. In addition to that, true modeling of all types of motion that happen to exist in the world, and inevitably, in images requires an utterly complicated algorithm and computational power. Consequently, the figures of action in this work are limited to the objects that demonstrate movements in an oscillatory manner or in forms of simple translational and rotational motion. On the other hand, the application domain of paintings are easier to animate in terms of motion, but hard to segments when they are highly textured.

CHAPTER 2

DIGITAL MATTING

Image segmentation involves the procedure of partitioning the digital image into multiple layers of interest where pixels are allocated into clusters to form pertinent image regions. The output of segmentation may be salient surfaces on the image, objects or compact parts of the objects. Regardless of the type of segmentation, a desirable result demands that the alike parts of the image are grouped in a single connected component and separated from the reminder of the image parts based on a some similarity criterion [38].

Image segmentation is a hotspot of image processing and computer vision research, owing to the fact that a diversity of applications of these field depend on the merits of segmentation. Evidently, segmentation simplifies or alters the image representation seeking to obtain more meaningful interpretation of the data and ease in analysis.

In literature, image segmentation has been used in object detection [57], [58], recognition [156] and tracking [120] algorithms along with many others. Many image processing require image or video segmentation as a subproblem to be attacked, and it is generally ill-posed and tend not to have a general solution. In this thesis work, image segmentation is used to layer objects of interest to animate, as well as still ones, into non-overlapping regions. As an image editing application, the first step of the image to video converter highly depends on the segmented layers of objects and object parts for further manipulation. By revisiting the block scheme of the system, it can be recalled that **Digital Matting Block** is the first step of the converter system and it determines how the rest of the animation will proceed.

An image is merely an aggregate of pixels without any manipulation, fortunately by means of segmentation, it is possible to obtain the object information in the image. The coherency of an animated layer is crucial for a visually pleasing output, in which the object boundaries are to be respected to keep the wholesome look of each object, namely each figure of action. As stated earlier in the problem statement, each body in nature has diverse motion dynamics, that can be a result of their physical state, structural properties, inherent behaviour or even due an interaction with the other bodies or nature. Conclusively, it is vital to have a well functioning segmentation methodology in this system.

Moreover, a proper layering is not necessary only because of object extraction from the image, but also it plays a significant part in rendering the video. When all the animated layers make their way to the ultimate block for rendering, (the procedure will be explained in the upcoming to chapters, Chapters 3 & 4), they require region masks for a proper blending of layers in the final composite image [37]. These masks are indeed the alpha channels that determine the transparency levels of an image layer and can be obtained through a matting procedure as an output of probabilistic procedure. Indeed this is the key point where, the problem diverges from object segmentation to digital image matting.

Digital image matting is a term used for computing the transparency maps of the objects to be used in composite images and videos for a seamless finish. That's why the focus is on the matting algorithms that can provide this gray-level maps instead of black & white regions of interest.

In this thesis, after a brief introduction to the methodology of digital matting, the details of the used algorithm, Weighted Colour and Texture Sample Selection for Image Matting, [135] will be provided. It is chosen in this system, thanks to its strength in layering textured images and its high ranking performance in [128], a known database for quantitative comparison of digital matting algorithms.

2.1 Related Works

Let B denote the background and F, the foreground, on an image I then, the alpha matte can be formulated in a closed form expression as follows:

$$I_{p^i} = \alpha_{p^i} F_{p^i} + (1 - \alpha_{p^i}) B_{p^i} \tag{2.1}$$

where p^i denotes any pixel $\in I$, and the range for α is [0,1].

Similar to a segmentation problem, digital matting tends to be utterly ill-posed as well, hence most of the application, especially in textured or crowded images, additional constraints are required to guide the algorithm to the region of interest [135]. These limitations may be provided as marks on the image in the form of a trimap, which is a 3-level mask drawn to indicate the foreground, background and roughly the boundary within an error margin; user scribbles that are smaller seed regions or pixels that indicate the presence of foreground-background on that specific location, or a bounding box in which the region of interest is encapsulated.

A main approach to the extraction of α is make local colour smoothness assumptions to compute the matte using the following equation:

$$\alpha_p = \frac{(I_p - B)(F - B)}{||F - B||^2} \tag{2.2}$$

The solution may include a parametric [130], [36] or non-parametric proposition. Without the parametrization, the selection of samples from the known background and foreground and the credibility of the selection remains as a tough challenge.

A secondary group exploits the local statistics of image and employs an affinity matrix based formulation [97], [146]. The solution for the matrix gives the α values for the whole image and it enhances the propagation of information from the known regions to the unknown. Another group of methodology adopts the benefits of both of these two, such as robust matting [158], [127] and iterative matting [157].

The above mentioned techniques have two common shortcomings:

- In certain image domains, the background and the foreground may demonstrate overlapping colour distributions, hence the alpha computed based on colour turns out to be erroneous.
- In regions containing rich textures, barely a colour based representations may fail to demonstrate the affinity in the adjacent pixels, as the large local gradients that are present in these regions may prevent the alpha values to be propagated.

In [37], which has a similar animation framework to this research, employs Bayesian Matting [36], a parametric method, assuming smooth colour distributions, paired with Intelligent Scissors for trimap generation.

2.1.1 Intelligent Scissors

Intelligent Scissors [110] is a method for selecting a region of interest with the aim to remain true to object contours, in general image edges and higher order features of the image, as much as possible. The boundary definition, indeed is a graph search problem, in which an initial seed node helps the algorithm to find the optimal path to a group of candidate goal nodes, by minimizing a cost function. In this algorithm the cost function to be minimized is the sum of local edges, and inevitably the optimum patch is enforced to lie along the strong gradients (edges). In implementation, edges are extracted by using Laplacian zero-crossings and the solution is obtained via Dijkstra's algorithm.

2.1.2 Bayesian Matting

Proposed in [36], Bayesian Matting dedicated to solve for an alpha value to extract the foreground from the background in a Bayesian framework, as the name indicates. For the solution of the problem, a probabilistic modeling of the background and foreground is assumed and they are taken to be distributed by superpositions of spatially-varying Gaussians. For the calculation of the unknown boundary region, a sliding window is used to get the neighbourhood information and the order computation is towards the boundary. That is, with the apriori information given or computed values in the adjacency of the current pixel, a MAP estimation is calculated to obtain the final opacity value.

Bayesian Matting is rather an old method and requires more strict trimaps, such as the output of the Intelligent Scissors, compared to user scribbles for its final outcome to be pleasing. Especially, in the domain of this thesis work where images are highly textured, it is expected to perform poorly as mentioned earlier. Indeed, it is intended to employ a method that will also decrease the need for user-interaction with the converter to remain on the more automatic side.

2.2 Used Method

The methodology that is employed in the the system is proposed by Shahrian et al. in [135] to layer objects and compute alpha mattes. It promotes the effectiveness of an additional texture-based representation along with colour-based representation in image matting algorithms. The algorithm uses sampling of these two feature spaces to optimize the foreground, F and background B, values.

The main contribution of the paper [135] is the proposition of a texture feature to compute a true alpha matte value and the motivation of selecting this method in the thesis framework is its success in rough textured images.

2.2.1 Weighted Colour and Texture Sample Selection for Image Matting

2.2.1.1 Texture Feature

A 36-dimensional feature vector is defined as follows

$$\mathcal{V}_T = (A^{\nabla}_{\ell,c}, A^{\mu}_{\ell,c}, A^{\sigma}_{\ell,c}, H^{\mu}_{\ell,c}, V^{\mu}_{\ell,c}, D^{\mu}_{\ell,c}) \tag{2.3}$$

where A, H,V, D correspond to the approximate, vertical, horizontal and diagonal subimages obtained out of a 2-level Haar Wavelet Decomposition [45] and the superscripts $(\star)^{\nabla}$, $(\star)^{\mu}$, $(\star)^{\sigma}$ stand for gradient, mean, variance and ℓ,c for level & colour channel indices, respectively. While extracting these features, the subimages are resized to the initial resolution of the input image by using bicubic interpolation.

Haar Wavelet Kernels

2D Haar Wavelet Transform allows the stair-case decomposition of an image into layers of detail by downsampling and filtering. In the discrete case, for an image I, the diagram is as follows:

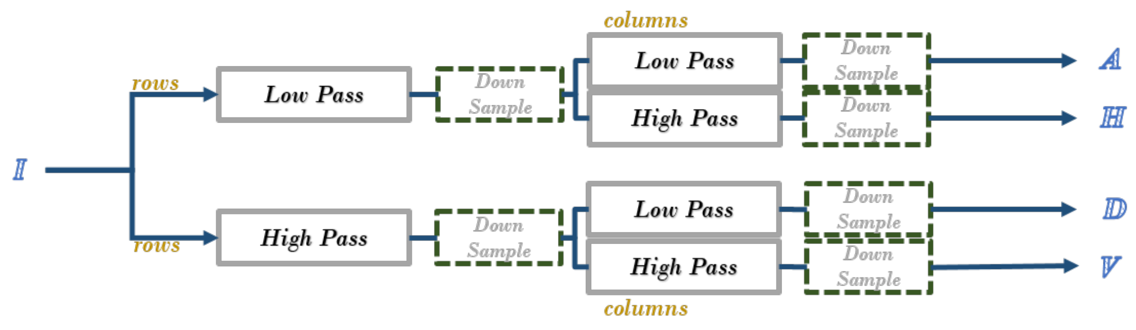


Figure 2.1: Schematization of 2D Haar Wavelet Decomposition.

In a more verbose explanation, it can be claimed that the decomposition gives the base image and image details along the specified directions. For the sake of clarity an example decomposition is provided in Figure 2.2.

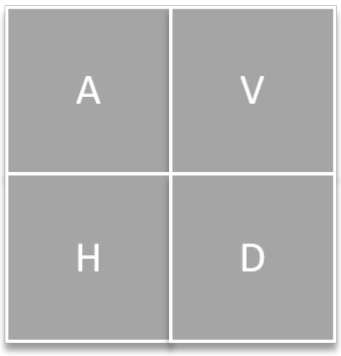
However, due to the computational complexity of handling a high dimensional vector, 2-level dimensional reduction is applied using *Principal Component Analysis (PCA)* and *Linear Discriminant Analysis.*

Principal Component Analysis

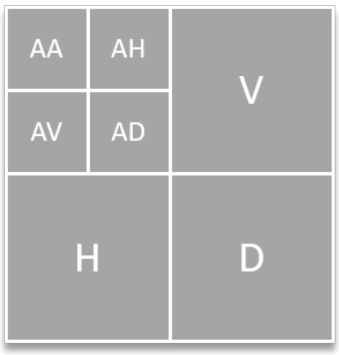
It is a basis for multivariate data analysis, that provides the approximation of the data while retaining a significant amount of its energy. It can be used as dimensional reduction technique to emphasize variation and bring out strong

(a)

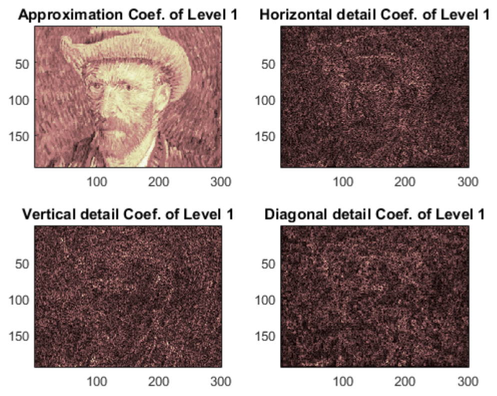
(b)



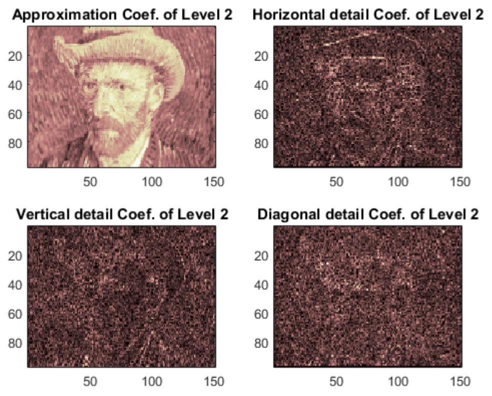
(c)



(d)



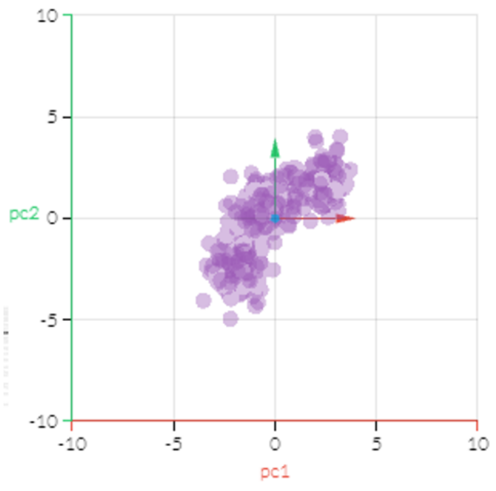
(e)



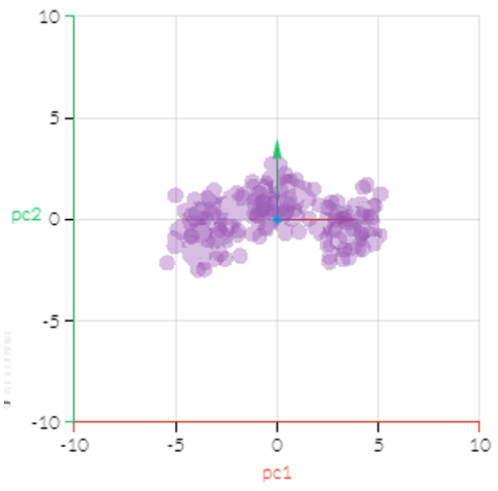
(f)

Figure 2.2: 2D Haar Wavelet Decomposition: (a) Input Image. (b) Input Image in Grayscale. (c) Single Layer Decomposition Kernels. (d) Double Layer Decomposition Kernels. (e) Single Layer Decomposition Images. (f) Double Layer Decomposition Images

features in a data set by projecting the data on the axes where the distribution is sparser. The components are obtained by the eigenvectors of the covariance matrix of the raw data. The eigenvector corresponding to the largest eigenvalue indicates the orientation of the axis that most of the data energy can be conserved.



(a) Raw Data



(b) Dimensionally Reduced Data

Figure 2.3: An example illustration of obtaining dimensionally reduced data using PCA. (a) Raw Data. (b) Dimensionally Reduced Data.

Linear Discriminant Analysis

LDA is a supervised data classification method that computes the component axes which maximize the class separation. The in-class, S_W and between-class, S_B scatter matrices are computed for the training data and $S_W^{-1} S_B$ is solved as a generalized eigenvalue problem. The idea is to maximize the inter-spacing between different classes constrained on keeping the intra-class data spacing minimum. The k eigenvectors that correspond to the largest eigenvalues are the linear axes components for this separation. By the concatenation of these eigenvectors, a transform matrix is obtained to apply on the data samples for the classification of the experiment data.

2.2.1.2 Computing The Unknown Region Using Samples

In this methodology, the criterion for an unknown pixel to be labeled as a foreground pixel is given as follows:

$$((||I_p - I_n|| \leq Th_T) \cup (||I_p - I_n|| \leq Th_C) \cap (D_{ist}(p, n) \leq Th_D) \tag{2.4}$$

That is to say, the unknown spatial neighbours of foreground pixel, that satisfy the distance condition Th_D, and are below colour and texture similarity thresholds Th_C & Th_D are eligible as foreground pixels. An analogous criterion applies for the background as well. Henceforth, the pixels that are in the smaller vicinities of the known regions are computed directly. The rest of the pixels that cannot fit into either region, are the ones that require the α-value computation which involve colour and texture sampling from both background and regions.

Inspired by [64], an information propogation method using the known background and foreground pixels is applied. Initially, the image is tiled into square blocks each having vertices of $v \times v$. Then, from each of these blocks, ℓ number of lines are scattered towards several orientations phase difference $2\pi/\ell$ and these rays are forced to hit the closest 6 blocks of known background and foreground regions, 3 of each. These 6 blocks will function as the region samples to perform a best pair voting that determines the optimal separation of foreground and background regions. Shahrian et al. state that it is critical to increase the number of rays to augment the amount of alterations in the colour and texture features [135],in the implementation for the thesis, a concentric growing circular region of interest is used until the required amount of blocks are sampled. The mean value of both texture and colour plays a part in the computation of alpha values in the unknown region, therefore by making combinations in pairs, the set P_{FB}, composed of 9 (F, B) pairs, is obtained for each unknown block. The procedure is illustrated in Figure 2.4.

Best (F, B) pair voting involves the maximization of the confidence measure defined as

$$Q = (C_\alpha)^{\omega_c} \times (T_\alpha)^{\omega_t} \tag{2.5}$$

where C_α determines the colour dissimilarity of an (F, B) pair and it is deter-

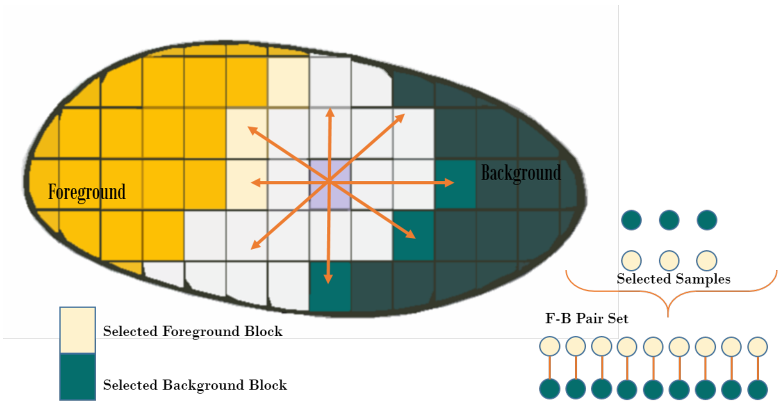


Figure 2.4: Sampling of F-B Pairs

mined by manipulation of the closed form expression for composite images, as in equation 2.5 and T_α denotes the compatibility of texture and colour of an alpha value, provided in the following equation 2.6.

$$C_\alpha = e^{\frac{-||I_p - (\alpha\prime F_p + (1-\alpha\prime)B_p)||}{\frac{1}{\mathfrak{C}}\sum_{(F_p,B_p)\in P_{FB}} ||I_p - (\alpha\prime F_p + (1-\alpha\prime)B_p)||}} \tag{2.6}$$

where $\mathfrak{C}$ is the cardinality of the block $\imath$ in P_{FB} and $\alpha\prime$ is computed for a specific (F, B) using equation 2.1.

$$T_\alpha = \alpha\prime \times SF_p^T + (1 - \alpha\prime) \times SB_p^T \tag{2.7}$$

where

$$SF_p^T = ||B_T - T_p||/(||B_T - T_p|| + ||F_T - T_p||) \tag{2.8}$$

$$SB_p^T = ||F_T - T_p||/(||B_T - T_p|| + ||F_T - T_p||) \tag{2.9}$$

and F_T,B_T are texture samples for the background and foreground regions.

Presumably, T demonstrates how reliable the colour information is for the alpha estimation. That is, if the computed alpha out of colour information approaches 1, it is highly probable that it is a foreground pixel, and the compatibility is incremented.

2.2.1.3 Weight Coefficients

In this algorithm, when the colour distributions of the two regions, F and B, tend to behave similarly, texture features begin to have more contribution in the alpha computation. The weighting coefficients in the objective function are computed by exploiting the colour histograms to measure any colour distribution resemblance. The amount of overlap in histograms is computed as shown

$$O(Hist^F, Hist^B) = \frac{\sum_{\imath}^{n} Hist^F(\imath) \times Hist^B(\imath)}{\sum_{\imath}^{n} (Hist^F(\imath)^2 \times Hist^B(\imath)^2)/2} \tag{2.10}$$

in which the normalized histograms of F & B, are given as $Hist^F(\imath)$ & $Hist^B(\imath)$, respectively, and n is the number of bins in the histogram. Then, ω_C and ω_T are formulated as

$$\omega_C = e^{-\frac{O_C}{(O_T+O_C)}} \quad \omega_T = e^{-\frac{2\times O_T}{(O_T+O_C)}}, \tag{2.11}$$

O_C,O_T are the overlaps of F and B distributions of the colour and texture features. When they are entirely overlapped in the colour space, $\omega_C = e^{-1}$, $\omega_T = 1$. In the other way round, when the complete overlap occurs in the texture space, $\omega_C = 1$, $\omega_T = e^{-2}$. It is to be remarked that, bare texture representation is not credible since it is determined in block resolution.

2.2.1.4 Matte Refinement

The alpha matte generation involves block processing, which means the values are kept constant block-wise in initial computation. In order to achieve a smoother transition between the adjacent pixels, a refinement procedure is proposed so as to provide that neighbouring pixels with high confidence values reinforce coherency on the alpha values. One-step iteration of the refinement procedure is described as

$$\alpha_p^{ref} = \frac{1}{\sum_{n\in\mathcal{N}_p} U_p(n)} \sum_{n\in\mathcal{N}_p} U_p(n)\alpha_n, \tag{2.12}$$

in which α_p^{ref} is the refined version of the computed α_p for some pixel p. $U_p(n)$ is the contribution of the $\mathcal{N}_p$, the neighbourhood of p. The contribution $U_p(n)$,

$$U_p(n) = A_p^c(n) \times A_p^t(n) \times Q(n) \tag{2.13}$$

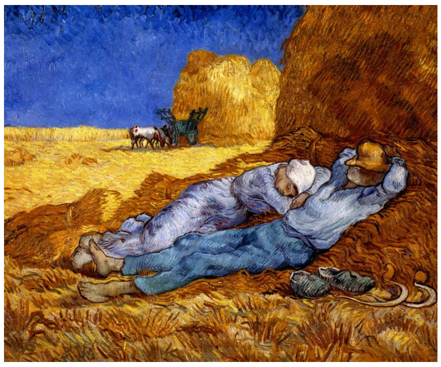
and

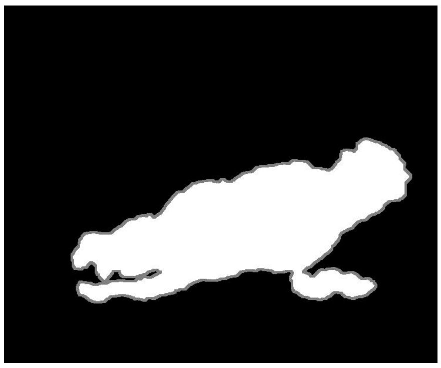
$$A_p(n) = e^{-\frac{||I_p - I_n||}{\frac{1}{\mathcal{C}ard(\mathcal{N}_p)} \sum_{n \in \mathcal{N}_p} ||I_p - I_n||}} \tag{2.14}$$

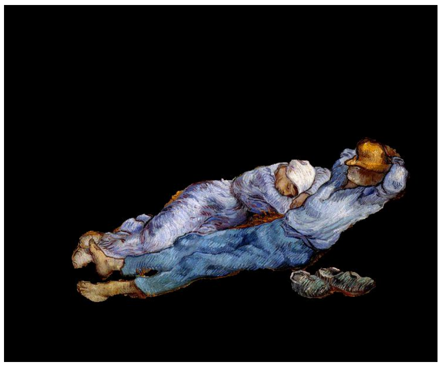
and $Q(n)$ is the normalized number of known values in the neighbourhood of n to its number of total numbers.

2.2.2 Test Results

The threshold values for the algorithm are finely tuned for each segmentation. The implementation includes a feature that the objects to be extracted can be marked using a free hand drawing tool. An example result of the algorithm is provided below in Figure 2.5


(a) Image


(b) Trimap


(c) System Output-Layer

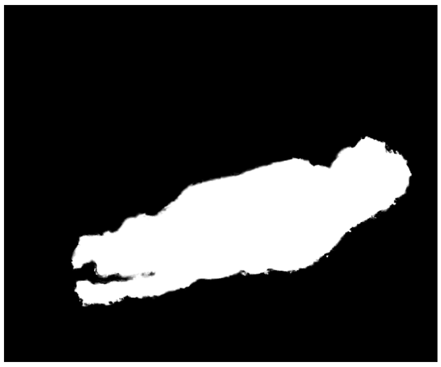
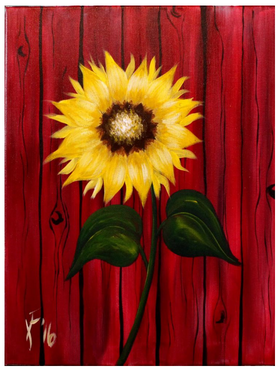
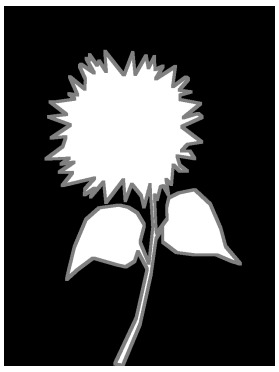

(d) System Output-Layer

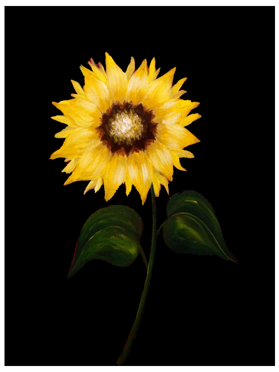
Figure 2.5: An example alpha extraction & layering as an output of the implementated algorithm [135].

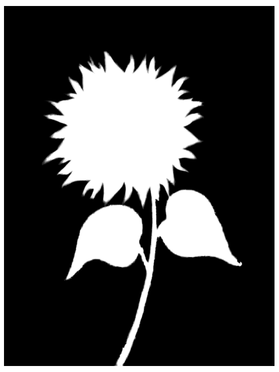
The comparison of the implemented method with different trimap extractors are provided in Figure 2.6. It has to be remarked that some morphological operations are done to extract a region boundary from the Intelligent Scissors (IS) output as it merely returns a binary output, where as the freehand tool,like a paintbrush has its own adjustable thickness.

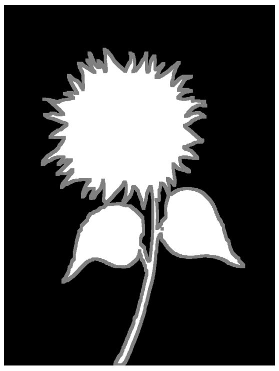
The comparative results of the two trimap extractors demonstrate that a freehand drawing tool is sufficient to extract a required quality alpha layer out of the image. For the cases, where exact layering is not achieved, further regional thresholding and morphological operations are applied in the **Rendering Block**, to be discussed in Chapter 5.

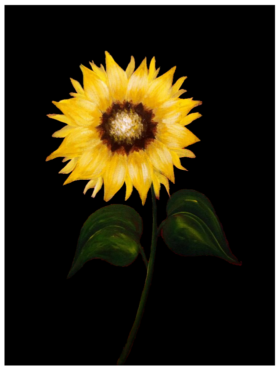

(a) Image


(b) Trimap-FH


(c) Layer-FH


(d) Alpha Matte- FH


(e) Trimap-IS


(f) Layer-IS

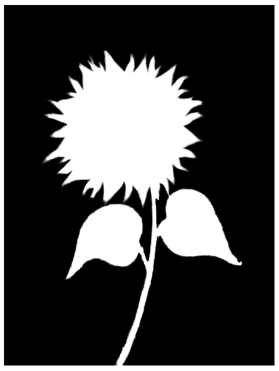

(g) Alpha Matte-IS

Figure 2.6: Comparison of alpha extraction & layering with freehand drawing tool and Intelligent Scissors [110] as an output of the implementated algorithm [135].

CHAPTER 3

INPAINTING

Inpainting is a term that dates back to Renaissance and is related to art restoration and it refers to the a cumulation of activities to keep the medieval art pieces to date [20]. This very term was first introduced to telecommunications area to name the error concealment applications in making up for the corrupted data. Later on, in last two decades, it has started to be used as the given name to removal of foreground objects and handling of disocclusion by completing the background pixels in digital images.

The main objective of using inpainting techniques in digital image processing is to manipulate pixel data on a certain region X on an image such that this X of interest has similar features and appear in accordance with the rest of the image. Consequently, the reconstructed parts of the image are obtained in such a visual continuation with the remainder image so that a random observer cannot grasp the fact the images have been retouched by any artificial means. The motives behind digital image inpainting include object removal, crack filling, texture synthesis and error concealment in general, however; concerning the entire work devoted for this thesis, it is made use of as a means of depth enhancement on the image. The procedure of depth enhancement will be explained in detail later in this chapter.

An image may said to be composed of structure, which is the rough sketch of the object boundaries; texture, the surface characteristic of the objects on the image and; the colour information, and for certain, these are what to be figured out thoroughly and imitated while generating pixels for the undesired locations [161].

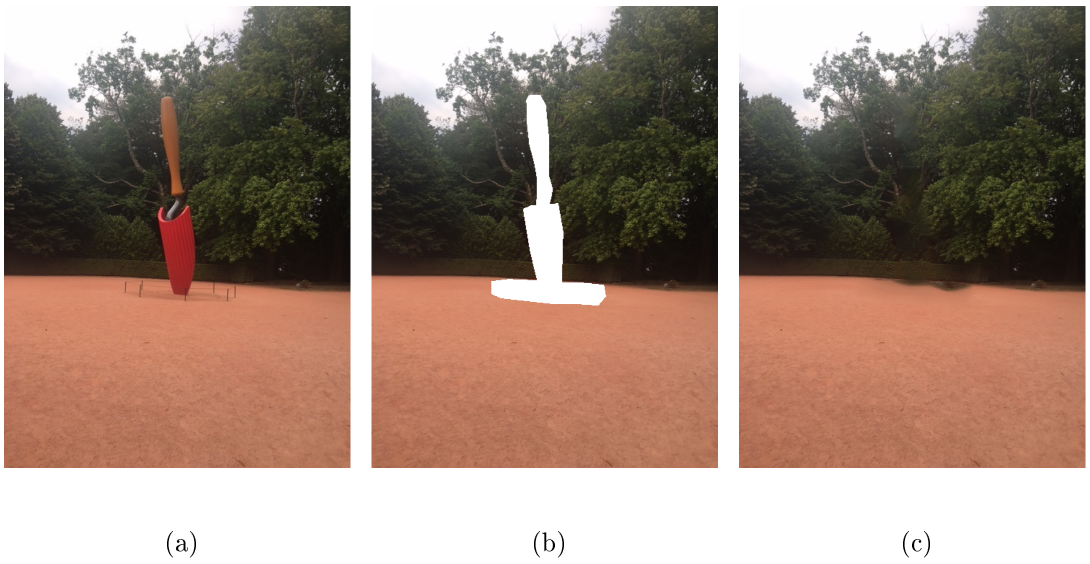


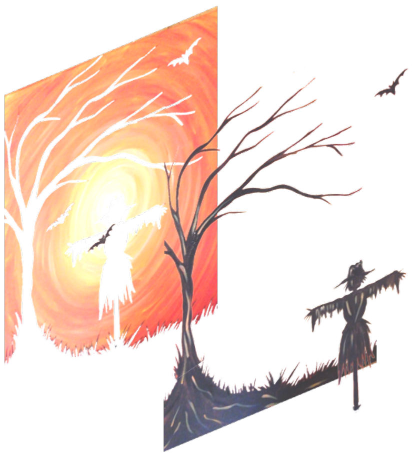
Figure 3.1: (a) Original Image. (b) Region Mask X. (c) Inpainted Image.

Any of these three assets may have more strength in the representation of separate images and clearly, needed to be prioritized among the others. Henceforth, regarding this fact as a basis, numerous approaches have been developed to realize inpainting goals, each taking one or two of the aforementioned assets to be more vital.

3.0.1 Necessity of Inpainting

Inpainting, as mentioned beforehand, is employed in this application of animation for above anything else, for depth enhancement. To be broad on the concept, the fundamentals of video animation and the experimental setup used to achieve this are to be revisited. When an animation is considered regarding the aspects of photorealism, perspective is an essentiality to be paid attention to. Due to the structural and morphological differences of the motile objects, the image is preferred to be partitioned into different layers in the previous block of the experimental setup. This partitioning outputs a number of layers in which of each there exists an object that is supposed to show a certain type of motion and the rest of the layer remains transparent. If the image frame is taken to be the x-y plane, then extracted layers can be translated along the z-axis to arrange the objects in foreground and the background, as shown in Figure 3.2.

(a) Single Image


(b) Layering without Inpainting

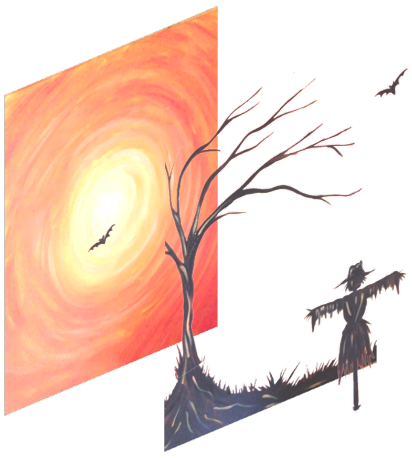

(c) Layering with Inpainting

Figure 3.2: The necessity of inpainting for perception of depth.

If the observation angle is taken as to be perpendicular to the image plane, from where the frame will look as an unlayered image, when the observation point is rotated around this plane, the separation between the layers will be visible. Consequently, a rough perspective can be said to be introduced into the video frame by the matting block of the setup.

For a complete perspective and perception of depth in the video frame, there exists the obstacle of blank pixels that occur due to the motion of any layer. Since these layers are formed of a group of pixels of a single image, even with the slightest spatial dislocation of any pixel, with the artificially created motion, brings forward the loss of data in its former position. More precisely, that pixel information, once occluded by a foreground object will be disoccluded with unknown intensity values, and it is to be admitted that the larger the motion is, the greater the dislocated blob becomes, as does the loss of pixel data, inevitably, leading to the formation of vast blank blobs in the video frame. The blank pixels/blobs do not only result in the deterioration of the seamless video frame, but also weaken the notion of proximity-remoteness of the objects with respect to the viewer. Hence, there rises the necessity of making up for the now-disoccluded pixels by determining appropriate intensity values so as to maintain the coherence of the image, through expanding the background region of the image. By applying this method, a proper appreciation of depth is obtained by the human visual system at the object boundaries.

Depth enhancement is a widely used concept in multi-view geometry, which has many applications such as controller-free gaming [69], stereoscopic image problems [96], etc. used to obtain higher quality depth maps that are more accurate on localizing objects and more eliminated in terms of system noise. Depending on the type of application, the desired accuracy of the depth may alter. In this application rather than extracting a depth map from the image, which is an ill-posed problem and computationally costly, the depth enhancement is realized by filling-in the aforementioned blank pixels/blobs directly, with the extraction procedure skipped. Precisely, the inpainted layers function as cut-slices of the x-y plane from different z-values, and instead of using a continuous depth map that is not obtained, these 'painted layers are used as its discrete samples, sampled at different z(depth).

3.1 Classification of Inpainting Techniques

This classification is based how the information of the region to inpainted and its background is provided to the algorithm. Since [51] which is widely accepted as one of the pioneers of image inpainting, although the name is fathered by Bertalmio et al.'s paper [21], a vast variety of research has been devoted to the area. Hence, it is not fair to classify all the work in terms of a single criterion, but it would be wise to investigate them with respect to different aspects.

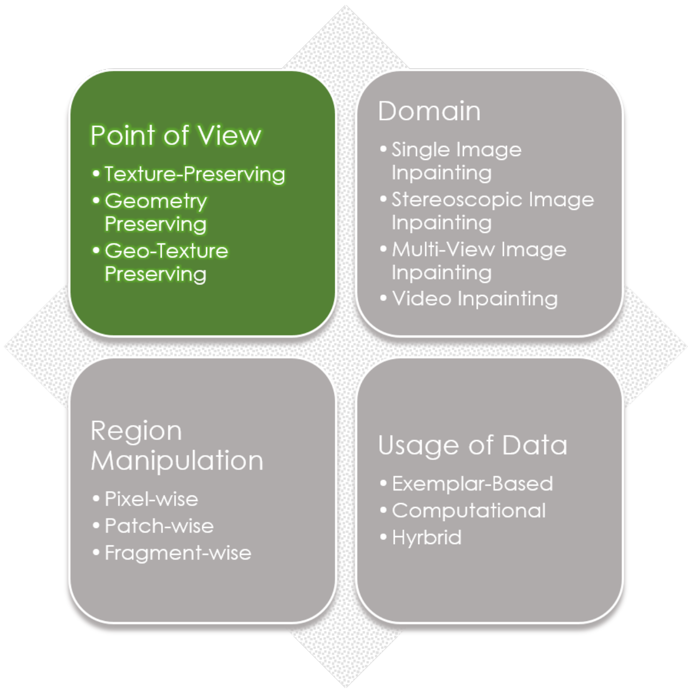


Figure 3.3: A classification of Inpainting approaches from four different aspects.

3.1.1 Point of View

This classification is based on which core element of the image is tried to be prioritized among the others for the specific in painting application and is one of the most critical categorizations on how the algorithm is expected to behave [29]. Eventualy it is the most widely used in literature [50], [29], [41].

- Texture-Preserving
- Geometry-Preserving
- Geo-texture Preserving

There are other aspects to classify the inpainting techniques developed so far. These classifications apart from the POV include the domain, region manipulation and usage of information. For the sake of clarity they can be summarized as follows:

Inpainting Domain

Single Image Inpainting: This group refers to the algorithms that are devoted to filling region on an input image with no prior information or input.
Stereoscopic Image Inpainting: The input for painting is a stereoscopic image pair.
Multi-view Inpainting: The region to be filled and its background is provided from many different points of observation.
Video Inpainting: The input and (most often) the output are available in the from of video frame sequences. May be used for error concealment.
Essentially is it to be clarified in the first place that, due to the single input nature of the framework, stereoscopic [111], [80] or multi-view [119], [153], [10] inpainting techniques and video inpainting applications [142], [163] are beyond the scope of this thesis work.

Region Manipulation
Pixel-wise: In general, the pioneering methods tend to implement pixel-wise processing algorithms. [51], [162], [5], [98] are works devoted to texture synthesis by using individual pixel information from the image. [21], [32], [11], [19] are variational approaches in image inpainting where pixel values are computed separately by solving a constrained equation. These methods are slower compared to the other groups due to individual manipulation requirements.
Patch-wise: The retouching of the image is realized through copying or calculating of pixels in arbitrary but fixed sized groups. Patch-based processing enables the reduction in number of the iteration of the algorithm; however, algorithmically patch based methods are more complex to allow parallel processing of data.
Fragment-wise: In these methods, the information of pixel values is not conveyed in fixed sizes but rather in irregular chunks. Drori et al. propose copying of data in the form of "peels" through a coarse-to-fine image decomposition procedure [50], so does Lubin et. al in the improved version of the algorithm [102]. In [34], [92], [83], [100] employ segmentation prior to region filling such that fragments are filled by using connected image regions.

Usage of Information

Exemplar Based Methods: Examplar-based inpainting depends on the usage of self-contained information from the valid part of the image to fill one that is unknown. [42], [41], [14], [15], [90]. The exemplars may be pixels, texture elements refers as *texels*, patches or even fragments of input image depending on the application. Typically, the method includes an optional pre-processing of the exemplar [90], [114], followed by a similarity search mechanism, brute-force [52], KD-trees [162], hashing [90], [82], random search [14], [15] etc. or a global optimization method [89], [91], [95] for pasting to provide pixel information.

Computation Based Methods: These approaches include mathematical model fitting on the whole geometry of the image to make up for the missing pixel information. In literature, the acclaimed modes are most often variational [21], [18], [11] or describe natural phenomena [19], [105], [55]. The completion procedure involves solutions of complex equations constrained on the image, that is to say, the replication of the known values is eliminated and each pixel is computed from scratch.

Hybrid Methods: This final bunch is the amalgamation of the former two groups, where the provided information is directly employed in some regions accompanied with mathematical manipulation for correct propagation of the image structure. Chaos mosaic [66] is one of those methods where small (in size) but numerous (in quantity) patches are created out of an exemplar, then merged into a single vast image. The merger of the small patches include inter-patch manipulation in order to avoid blocking-effect in the output. [56], [3] treat the target region as the weighted sums of the patches in the source region as a solution of variational constraint. Segmentation based inpainting methods [83], [100], [34] require interpolation methods for a seamless finish in the output.

Although the mentioned classifications are important in understanding the nature of the methods proposed, the literature analysis in this chapter will be based on the Point of View criterion, therefore the development of **Texture-Preserving**, **Geometry-Preserving** & **Geo-Texture Preserving** methods will be explained in detail.

3.2 Texture-Preserving Viewpoint

Counted as one of the core elements on an image, textures can describe a vast variety of repeated surface characteristics with a degree of randomness to represent natural or unnatural objects. Texturing is a core process for computer graphics applications [161] where synthesized textures are commonly used to imitate the real world objects and their features. Texture-preserving methods gather the algorithms that are dedicated to synthesizing images out of a sample texture or mapping a texture on a given image. The benefits of this texture-oriented group of methods involve the capability to generate vast areas of images and/or tileable textures out of a small sample inputs by straightforward techniques [5].

3.2.1 Inpainting with Texture-Oriented Methods

Particularly, the employment of texture-oriented in inpainting applications have the common motive of extending the present texture models in the image to cover up the undesired or damaged regions on the image. This extension is realized by duplicating certain image regions to fit appropriately into target region to be inpainted.

In a more broad explanation, it can be said that the intensity value of each every pixel to be inpainted is copied from a matching pixel in the source image. Finding the source-target match pairs involves sampling of the input source image for data that is compatible with those in the location of their target region correspondents.

Conclusively, the whole approach can be summarized as a similarity search, which contains an assembly of objects represented with certain features located in the high-dimensional space of image. With the queries provided, it is required to figure out the closest object to the desired query [44]. In texture synthesis, modelling the image and sampling are the two crucial factors to be considered, since the more accurate the model is, the more the visual fidelity of the synthetic images become compared to the raw image, and meanwhile yielding higher efficacies during sampling will reduce the computational cost of the whole synthesis operation [161]. These emphasized concerns determine the evolution of texture-oriented methods, *id est.*, the decision on how these samples would be selected, with respect to selection of candidates, neighbourhood size, search order and the *"likeliness measure"* create the change in the texture synthesis algorithm.

Indeed, this approach is very compatible with the human visual system, which has the potential to hallucinate the unknown information with what is already available. Obviously, it is fair enough that synthesizing textures is one of the pioneering techniques to be used not in only inpainting applications, but with further processing for object detection, recognition application as well [71].

3.2.2 Texture-Oriented Approaches in Literature

In this section a brief overview of the previous work on texture-oriented methods, limited to 2D image textures, will be provided. An important notice is to be made that, texture synthesis and texture-oriented inpaintings are separate problem, but they can be formulated in a similar framework when the approach is basic. In literature, it is possible to find the adaptations of the same method for both problems with simple algorithmic changes.

3.2.2.1 Non-parametrical Texture Syntesis & Pixel-Based Evolution

[52] is one of the state-of-the-art texture synthesis methods to be used in inpainting applications that outperformed its previous work visually and it is a pioneer in this very specific field such that several new algorithms have been developed taking it as the core idea of inspiration. The algorithm can be comprehended with ease; therefore, a broad explanation of the method worth the effort for the reader to grasp the flow in the development of ideas in time.

Markov Random Field

It is a widely proposed idea in the image processing and vision applications that by using Markov Random Field, *MRF*, model the images are divided into a collection of nodes, where nodes represent either pixels or agglomerations of

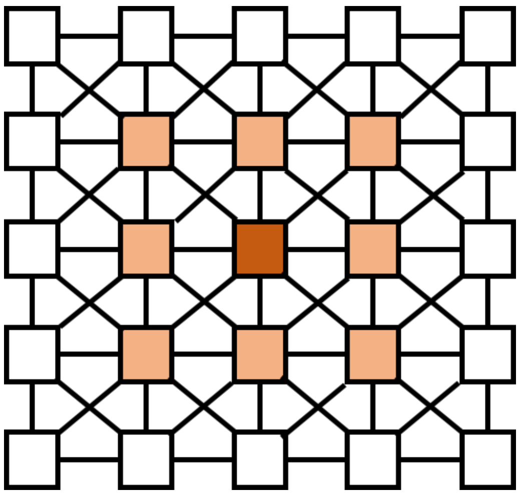


Figure 3.4: Graph representation for a Random Markov Field (MRF) modelling of an image using 3x3 neighbourhood.

pixes [23]. In [52], image textures are modelled as a *MRF*, to generate textured large images by probability sampling.

Precisely, it has the key assumption that the probability distribution of pixel intensity of a pixel, given those values of its spatial neighbourhood, is independent of the remainder image. In other words, two pixels that share the same neighbourhood information are expected to be the same. Based on this corollary, the goal of texture synthesis can be formulated as follows: given an input texture, synthesize an output texture so that for each output pixel, its spatial neighbourhood is similar to at least one neighbourhood at the input. The solution to satisfy the probability assumption is to seek the best matching neighbourhood in the image.

Given in Figure 3.5, let image I be consisting of two regions, namely Λ, the region to be inpainted and $I - \Lambda$, the region to be kept as it is. The pixels belonging to the region Λ are filled in by copying the pixel intensity values of the $I - \Lambda$ region in terms of similarity with respect to the adjacent pixels. Precisely, the algorithm assumes that pixels that have alike neighbouring pixels tend to have the same intensity values. Therefore, for each pixel in Λ the best fitting pixel is searched in the $I - \Lambda$ domain using the similarity metric of Sum of Squared Distances, SSD,

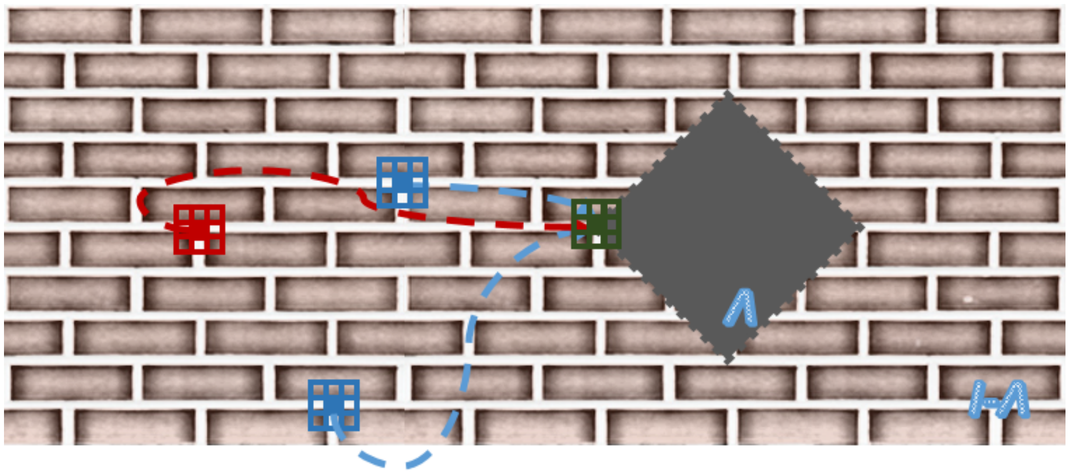


Figure 3.5: Filling Method proposed by Efros et al. [52]. The image, I is searched for the candidates best matching the green pixel, in λ by comparing neighbourhood information. The most similar value in $I - \lambda$, that of the red pixel is copied.

$$d(\psi_t, \psi_c) = \sum_k \sum_l |\psi_t(k,j) - \psi_c(k,j)|^2 \tag{3.1}$$

where, ψ_t is the neighbourhood of pixel t, $\psi_t \in \Lambda$, ψ_c is the neighbourhood of pixel c, $\psi_c \in I - \Lambda$,

that computes the normalized Euclidean distance between the neighbours of the candidate pixels that are already retouched or desired to be kept unchanged. The best match is constrained on:

$$c =_{argmin} d(\psi_t, \psi_{c_i)} \tag{3.2}$$

where c_i is the candidates for t for $i = 1, 2, ...N$

Consecutively, then t is inpainted with the value of c.

The size of the neighbourhood is a user-specified parameter and should be compatible with the largest unit repeated pattern in the image to keep the texture energy within the neighbourhood. It is a rather computationally costly method, nearly the whole image is searched for each pixel to be determined [52], reaching the *O(N²)* complexity, due to the fact that sampling of the image can be described as non-local, such that a single search is performed in the entire remainder image.

In order not to favour the gradients or object edges along any principle direction in the image plane $+/-x$ or $+/-y$, Λ region is not filled using straight lines of order. Instead, the boundary pixels are inpainted first in a concentric means so as to propagate the texture towards the center of the lambda region. This method of prioritization is called the onion-peel ordering that allows the features to be continuous in all directions possible more effectively than a linear filling method. In a nutshell, the method is very simple but fundamental one to build on with a single parameter to determine the neighbourhood size [161].

In [162], Markov Random Field assumption is kept and the texture synthesis applications are regarded as realizations of a stationary and local random processes. The major problems of textures synthesis are defined as the estimation of the entire stochastic process of the image by a finite sample and the challenge to obtain a search method to be considered as efficient. Starting from a random noise field, to capture the randomness factor in the real patterns, each pixel is manipulated by verbatim copying individually to make the output similar to the input texture. The first problem is handled with the locality assumption, where they mentioned the importance of neighbourhood similarity and preferred the neighbourhood size to be arbitrary so as to provide flexibility on textures with different sizes of regular structure. Furthermore, to be loyal to the structure of the input, a causality principal is adopted where an L-shaped neighbourhood is searched, in a raster scan order, by comparing the similarity only with the already assigned pixels using the typical SSD metric.

In texture synthesis, The WL algorithm [162], despite being successful in mimicking natural textures, especially smoother ones without well defined edges, unfortunately, when sample textures that form the seed of texture synthesis appear to have sharper transitions or visually distinct features, its performance of replication in the generated image is not that jaw-dropping as the edges appear to be blurred. A similar visual degradation is evident in [66], where a bunch of relatively small texture regions are generated and gathered via random pasting. The post processing to fulfil the transition between the tiles impels the occurrence of blurry edges. This drawback is due to the utilization of the simple an smooth norm L_2, which fails to acquire the behaviour of edges, corners or higher

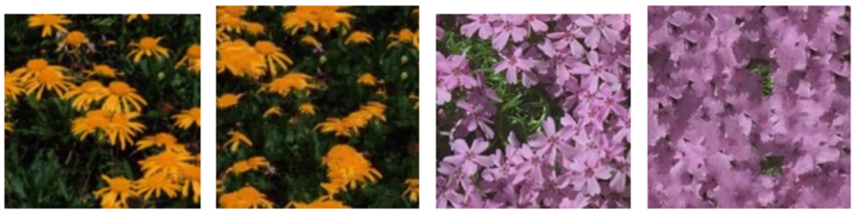


Figure 3.6: Image blurring effect of the L_2 norm illustrated in the synthesized textures using [162]. Image retrieved from the corresponding paper.

order features [5]. Human visual perception is very conformal to edges [155] and it can be deluded provided by characterization of the image by preserved edges. [124] tackles with the blurriness issue with a trade-off between the quality and rapid computation that was achieved in [66].

3.2.2.2 Coherence and Region-Growing Effect

Both [52] and [162] suffer from low-quality and speed as query methods. In term of the quality of the output image, [162] gives noisy results and [52] irrelevantly textured images. In [5], Ashikhmin et al. rely on the representational prowess of WL and propose some modifications to prevail the smoothing effect of the system. A critical observation that forms the basis of this novel algorithm is that the computed pixels of the output image have neighbourhoods in the input texture, which are the shifted versions of the neighbourhoods of the pixels yet to be painted. In other words, the candidates for the current pixel to be painted are the iso-positioned neighbours, in the original texture, of its already painted neighbours in the output. A better explanation is provided visually in Figure 3.7.

As a consequence, by using the effect of coherency, instead of starting each search for the best match from scratch, the vicinity information is further shifted from the defined L-neighbourhood that is already filled. This requires extra memory to keep track of the input image coordinates for the output image duplicates, but fortunately it effectively shortens the run-time of generation. During implementation, a random initialization of these locations are preferred. Owing to its information propagation property, the results demonstrate a visual region-

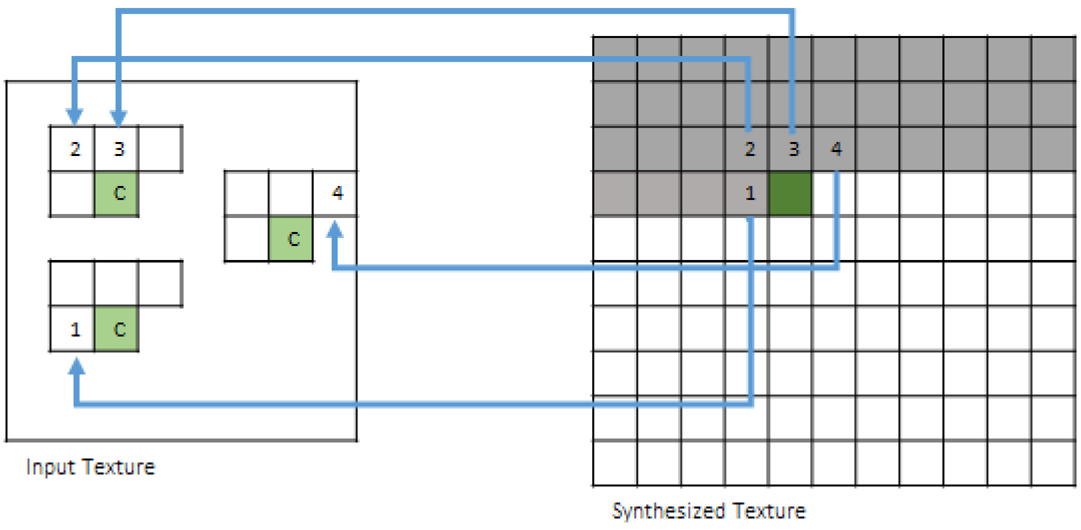


Figure 3.7: Filling Method proposed by Ashikhmin [5]. The neighbourhood information of the previous matches are propagated to the neighbours for a faster implementation.

growing effect where the tiles are not formed as rectangular but in irregular chunks, which makes the outcome look more seamless. Due to its straight scan-line order, a toroidal neighbourhood treatment is evident, which is not much of a consideration in horizontal tileability; however, depending on the input texture, the algorithm may be trapped in distinct features located in the bottom section, causing a more frequent show up of the feature. A random candidate assignment method is proposed, to overcome this shortcoming, if applicable, during implementation.

[5] allows user interaction to determine the general properties of the generated texture. This flexibility ranges from the adjacency boundaries to the initialization of the best match locations. Different from [162] larger neighbourhood only helps the algorithm in [5] to have more candidates in seeking the most similar, whereas, it is vital for [162] mirror the low frequency characteristics of the texture in output image. [5] is proposed to be not only a texture synthesis method but rather a texture mapping method, in which a fractional blend of the frequency characteristics of two separate images is achieved to produce images that demonstrate features belonging to both. The user-interaction in the initialization phase for the best match location map leads to the production of

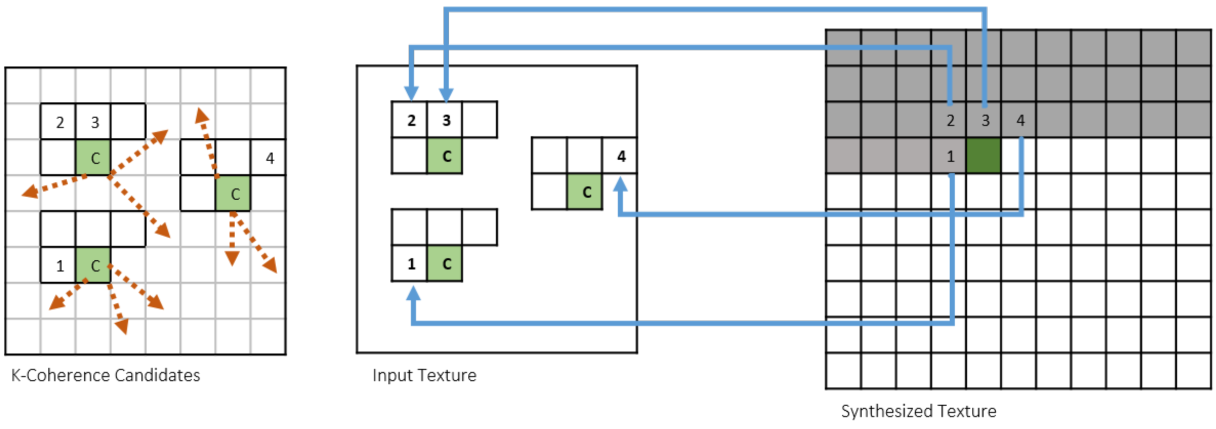


Figure 3.8: K-Coherence Search: As an improvement to the method proposed in in [5], the search domain is extended with the addition of k nearest neighbours of the coherence candidates

amalgamated images.

3.2.2.3 Accelerations on the Search Methods

Other proposals to accelerate the search mechanism were [162] by tree-structured vector quantization (TSVQ), [91] using kd-trees, [172] by jump maps. Inspired by [5], [154] defines k-coherence where similarity information is propagated as in [5] using k-best matches, as illustrated in Figure 3.8. A recent research is devoted to the acceleration of the [52] algorithm using Principal Component Analysis, namely PCA.

3.2.2.4 From Pixels to Patches

Due to timing considerations and the struggle to have higher quality output images, the novel approaches, the unit of search and data transfer is altered from pixels to patches. However, the introduction patches to the algorithm domain introduced two problems. The former, indeed, being not necessarily a problem but a clarification, is that although the required query iterations are decreased to accelerate the algorithms, the parallel processing of within-patch information is still a burden. Moreover, to state the latter problem, it is to be admitted that patch pasting brings in the concern of overlapping patches.

The solutions proposed to the latter concern have a diversity among the literature, where in [124] the patches are let to be overpainted, which causes blocking artefacts on the output image due to mismatched features and in [98] weighted blending of the regions is preferred. Unfortunately, the sharpness of the edges is eventually lost and the final image comes in a blurred version of the sample.

3.2.2.5 Cut-primed Copying

Efros et al., in [51], so as to preserve the completeness of the image, prior to insertion of a selected patch into the target region, compute the error of the overlap between the already placed blocks and the current one. The boundary of the two blocks for vertical overlapping is determine by the following error function:

$$Err_{(k,l)} = err_{(k,l)} + min(Err_{(k-1,l-1)}, Err_{(k-1,l)}, Err_{(k-1,l+1)}) \tag{3.3}$$

where

$$err = (\psi_t^{OVLP} - \psi_t^{OVLP})^2 \tag{3.4}$$

and ψ_t^{OVLP},ψ_t^{OVLP} are the overlapping regions that belong to patches centred at pixels t & c, respectively.

A similar function can be formulated for horizontally oriented overlaps or both by combination of the two. The dramatic increase in the E value determines where the mismatch occurs between the patches and by tracing back the values it is possible to figure out where the cut between the patches is to be made. This method is implemented with dynamic programming of Dijkstra's algorithm [49].

In [92], the extension of the idea of seam finding is present. In this method, image pixels are represented as the nodes in a graphical model and the similarity measure between two adjacent pixels, t and c is defined as

$$C'(t, c, \psi_\alpha, \psi_\beta) = \frac{C(t, c, \psi_\alpha, \psi_\beta)}{||Grad\psi_\alpha(t)|| + ||Grad_{\psi_\beta}(t)|| + ||Grad_{\psi_\alpha}(c)|| + ||Grad_{\psi_\beta}(c)||} \tag{3.5}$$

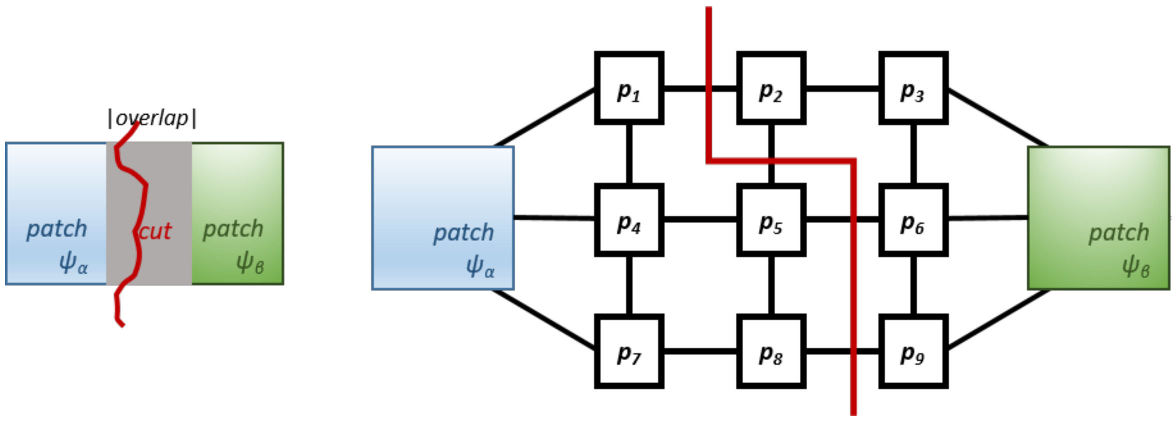


Figure 3.9: Filling Method proposed by Kwatra et al. [92]. Overlapping regions for the patches are cut using $minimum - cut/maximum - flow$ approach.

$||.||$ standing for the norm operator and ψ_α, ψ_β for the old and the new patches, respectively and using

$$C(t, c, \psi_\alpha, \psi_\beta) = ||\psi_\alpha(t) - \psi_\beta(t)|| + ||\psi_\alpha(c) - \psi_\beta(c)|| \tag{3.6}$$

where $Grad(.)$ performs directional gradient computation.

Indeed [92] is the application of the $minimum - cut/maximum - flow$ problem [60], [134] on texture synthesis, and the solution determines the path to separate the given nodes, namely pixels.

In [126] the energy function is restated the function by adding extra constraints on the similarity of of the patch to be placed wth the already placed patches. The main difference of the two cost functions, with [92], is that the latter involves a patch placement cost where in the former all the patch placements are made randomly. Eventually, the newer one eliminates the necessity of including query & paste heuristics to preserve the overall structure of the input sample.

3.2.2.6 Texture Optimization

A texture, in terms of statistical description, is the outcome of a random process. Hence, in [75] Heeger et. al modified an initial random noise field to match the histogram of the noise field to that of a texture sample image, by using a pyramidal approach, where steerable Laplacian pyramids are used to extract information via subband transformations at multiple scales and orientations. [26] is a recent extension to [75] to apply the method on coloured textures using PCA. [46] has a similar approach to [75], but the algorithm initialized with the down-scaled version of the input texture instead.

In [91], Kawatra et al. propose a joint optimization formulation for texture synthesis that allow gradual refinement and retouching on the generated texture. The problem is defined as an energy minimization approach solved through Expectation-Maximization [109].

Unlike the pixel growing techniques in literature [5], [154] to preserve coherence solely in localities, this methodology gathers the neighbourhood based assumptions into a global measure to metricate the quality of the generated image. Based on the premise that each neighbourhood on the synthesized image must have a corresponding similar neighbourhood in the input sample, the energy to be minimized is formulated as follows:

$$G(t, c_p) = \sum_{p \in T^*} |t_p - c_p|^2 \tag{3.7}$$

where T^* refers to the active subset T, the output image.

Algorithm 1 TS with Global Optimization [91]

```
  Initialization: $c_p^0$ = arbitrary $c_p^*$ in $C$, $\forall p \in T_*$
  function EM($C, T$)
      for $iteration \tau = 1$ to $N$ do
          $t^{\tau+1} \leftarrow argmin_t G(t, c_p^\tau)$
          $c_p^{\tau+1} \leftarrow$ NN of $t_p^{\tau+1}$ in $C$, $\forall p \in T_*$
          if $c_p^{\tau+1} == c_p^\tau, \forall \in T_*$ then
              break
          end if
      end for
  end function
```

The E-step of the EM is where energy function E is computed, keeping the candidate set of input neighbourhood fixed and in the M-step the set of compatible neighbourhoods are found via tree search. The two interleaved steps are reiterated until a convergence criterion is reached.

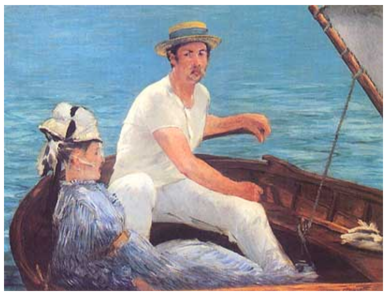
[70] ameliorated this method with respect to two-aspects of the output, efficiency and quality. The infamous L_2 norm is utilized in the previous method [91], which is empirically known to be one of the primal causes of blur, whilst [70] utilizes k-coherence approach for both energy calculation (expectation-phase) and the query of best matches (maximization-phase). Consequently, as the refinement on the image is omitted by direct copying of pixel values the formation blur is prevented, and the operation speed is fixed to a constant value rather than being $O(log(N))$.

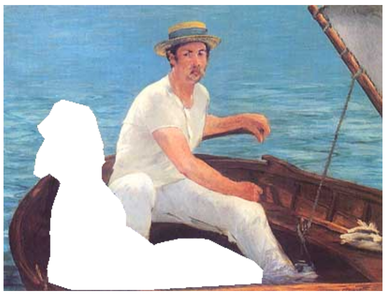
3.2.3 Tests on Texture-Oriented POV

It has been discussed that texture synthesis methods are useful in completing a region by replicating the features of a sample image. Object removal, however in most cases include more than replication of a single pattern, therefore texture synthesis methods are not suitable for the completion of holes left after large objects [50]. This fact will be illustrated by test results on this group of methodology.

Synthesizing Natural Textures The selection of this method for testing owes to the fact that it performs pixel-based synthesis as opposed to all the other methods implemented within the scope of the research [5]. As explained in Section 3.2.2.2, it has emphasis on coherency propagation in best match queries. The test result for image completion using this method is provided in Figure 3.10. As observed the inpainted image contains chunks of pixel information from irrelevant parts of the image as result of the image. This is due to two reasons the neighbourhood size is not capable of capturing the true amount of information and the coherence propagation increased the aftermath of propagation of incorrect information. Also the output is blurry due to the utilization of a small unit of synthesis, namely a pixel.

Image Quilting Proposed by Efros et al. [51], this method is one of the pioneers


(a) Original Image


(b) Region Mask X

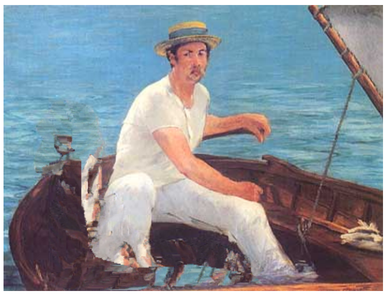
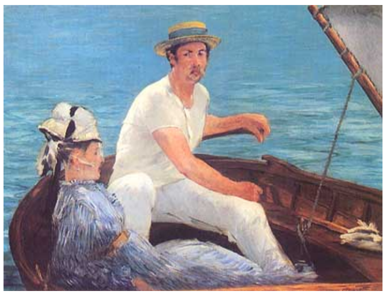

(c) Inpainting Results

Figure 3.10: Inpainting Result on 305x405 pixel image using Natural Texture Synthesis [5]

of patch based synthesis and it allows the patches to be overlapped by calculating an error tolerance on the overlap region. The methodology is explained in Section 3.2.2.4.

As the algorithm is formulated in raster scan for patch it requires elongated time intervals, therefore in our implementation, the input texture is tiled into distinct regions larger than the patch size. Hence, once match occurs, the tile that the patch is located is saved. For the next searches, patch is compared with the earlier patches, if the SSD is smaller than a threshold, the search space is restricted to that tile. Indeed, the implementation is somewhere midway between the k-d tree approach of pixel based synthesis in [162] and Image Quilting. For the computation of boundary cut Dijkstra's algorithm is used.The similarity threshold for the image is the tolerance value [51],taken to be 0.01 as indicated. The result for image inpainting given in Figure 3.11, displaying a result with blocking effects especially on the region boundaries. Even though, overlapping of patches is allowed, the method fails to capture correct geometry in objects boundaries.


(a) Original Image

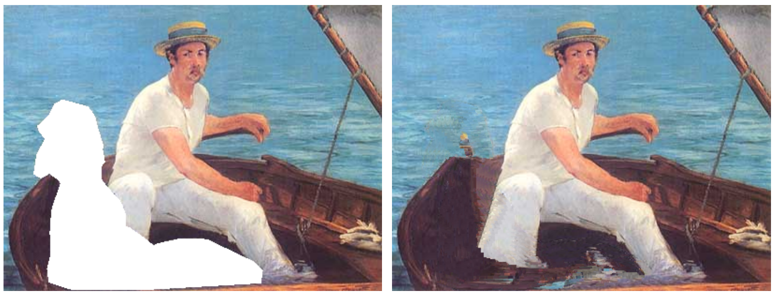

(b) Region Mask X (c) Inpainting Results

Figure 3.11: Inpainting Result on 305x405 pixel image using Image Quilting [51]

3.3 Geometry-Preserving Viewpoint

When the removal of objects in the foreground is considered, it is abundant that the background objects are partially or completely occluded by the very foreground object itself. In partial occlusion cases, the apparent parts of the objects act as clues to be interpolated in continuum with the geometric shape of the background objects [29]. Human visual system has the capability of intuitively recognizing and hallucinating the disguised parts objects as complete [121]. For digital inpainting problems, this merit is inspirational and there is struggle mimic this well-functioning facility.

3.3.1 Inpainting Using Geometry-Oriented Methods

The methods that fall into this group try to complete the image using a continuity assumption, and figure out the missing pixel values by solving for a partial differential equations. A PDE can be derived based on the image minimizing a variational cost function or by using a phenomenological modelling. The priority of variational inpainting is to preserve the global structure present in the image while reconstruction the undesired/corrupted region [29].

With a geometry-based point-of-view, Partial Differential Equations, PDE-based methods differ from the techniques that attempt to fill in the corrupted or desired part of the image by using samples from the remainder image, but rather compute for the pixel values regarding the model constraints. Therefore, primarily the locating the prominent features of remainder image becomes crucial, since the object contours are required to be completed for proper depiction of background object. Then, how to formulate the continuation is to be decided, which is the main criterion that determines the alterations between the variational image inpainting approaches.

3.3.2 Geometry-Oriented Approaches in Literature

3.3.2.1 Level Lines

Ogden et. al states an early example of interpolation based filling of the missing pixels in the images. [115] proposes constructing a Gaussian pyramid that denotes the image with varying spatial resolution values, from the coarsest to the finest, G_0 and G_n, respectively. The method consists of passing the image through a Gaussian filter, and successively up and down sampling the image.

[107] is a yet another method for interpolation where the isophates (the adjacent same-intensity pixels) of the image are tried to be preserved. This basic method, attacking the problem of disocclusion, has inspired the basis of plethora of algorithms devoted to the area.

$$\int_{[\xi,\Xi]} \int_{\mathcal{C}_\ell} (\alpha + |\kappa \mathcal{C}_\ell|) ds d\ell \tag{3.8}$$

The energy function in equation 3.8 that the image is constrained on is Euler's elastic model, using L_1 norm, which draws a parallelism between the disocclusion problem and the brain's ability to extrapolate the cracked edges in its sight. The range $[\xi, \Xi]$ corresponds to the pixel intensity values that the inpainting domain

λ in image I can have, $\mathcal{C}_\ell$ is the non-crossing curves that are expected to coincide with the image isophates in the ϵ-vicinity of the hole boundary. α, β are tunable and context-dependant.

1

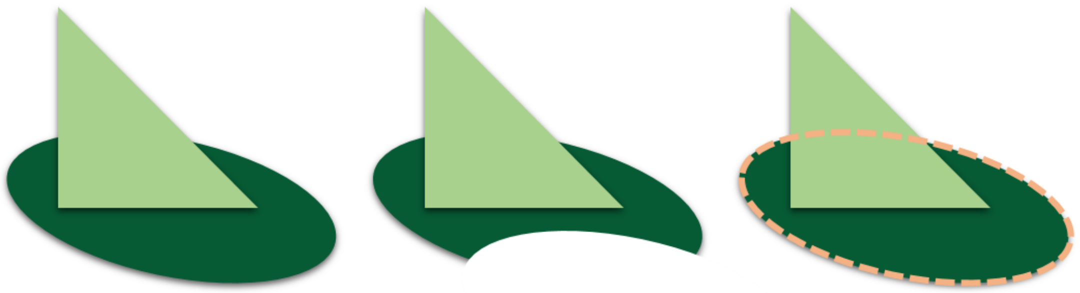


Figure 3.12: An illustration of the perception of incomplete edges by the human visual system as a motivation to use elastica model. (a) The complete object. (b) The occluded object. (c) Interpolation of the curve model to reform the shape of the object.

3.3.2.2 PDE-based Methods

A differential equation consists of a function of single or multiple equations and its derivatives, hence it is comprehensible that they are the epitome of image inpainting, since most of the image features can be described by arbitrary superpositions of its high order moments.

The inpainting is fathered by the work in [21] and Bertalmio et al. make significant definition of the inpainting problem, where no information for the target region is provided, is not to be taken for a denoising problem, in which the data values are present in corrupted manner.

What is tried to be imitated here is the means how the art restorators work. The global appearance of the image is to be kept while, the gaps are needed to be filled in via prolonging the contour lines that arrive the region boundary. That principle defines the nature of PDE-based or variational inpainting in attacking the problem: Unlike the texture-oriented methods that were presented n the previous subsections, these methods are local, and only the boundary information is considered during operation.

Let ω be the region to be inpainted in the image I, on the domain ω then each instance τ of the filling algorithm can be modelled as in equation 3.9.

$$I^{\tau} = \nabla^{\perp} I \cdot \nabla(\triangle I), \quad \text{in} \quad \omega \tag{3.9}$$

where $\triangle *$ defines a smoothness operator on I, the image. Under this operator, the points that yield constant values on the orientation induced by the field $\nabla^{\perp} I$ are defined as stationary. The model is combined with the anisotropic-diffusion equation to form

$$I^{\tau} = \rho(x)|\nabla I| \nabla \cdot \frac{\nabla I}{|\nabla I|}, \tag{3.10}$$

where ρ, a smooth cut-off function, enforces the operation to be limited to the boundary of the image hole $\partial\omega$.

In order to account for the linearity-enforcing property, the BV model is substituted for the following one in another work by Chan et. al [31].

$$I^{\tau} = \nabla \cdot \mathcal{C}(\kappa) \frac{\nabla I}{|\nabla I|} + \lambda(I^0 - I) \tag{3.11}$$

where $\mathcal{C} : B \leftarrow [0, +\infty)$ and B stands for to a bunch of curvature functions.

But it is easily comprehended from the λ coefficient in the equation that large curves upon image completion is penalized. To be precise, the conductivity coefficient must be examined. In the previous work, $\frac{1}{|\nabla I|}$ leads to the diffusion to be only dependent on the strength of the gradient. The addition of $\mathcal{C}(\kappa)$ obliterates strong curves and favours smaller ones. Hence, the CCD model propagates information in a way that is perpendicular to the algorithm in [21] where smoothness is prioritized along the isophates.

3.3.2.3 Introducing Fluid Dynamics

Bertalmio et al. developed the previous work in [19] by modelling a new diffusion model. The reason behind is to be more strict on the boundary conditions such that the flow is kept within the target region. An optional solution is to employ stronger boundary conditions [19]. In previous works on fluid dynamics demonstrate that time-dependent solutions of the Navier-Stokes equations with isotropic viscosity exist and are unique [104].

Within this method, the flow of a compressible fluid in 2D is used to formulate the changes in the pixel intensities. The former expression of image smoothness in [21] now relates to the vorticity of the fluid, and the isophates are considered as the riverbed and the diffusion function is the viscosity. Eventually, the problem turns into an analogy of how a stream would fill its basin and the solution is reached by a variational approach. In this subsequent work, the model includes two PDEs of second-order, corresponding to gradient orientations and the grayscale intensities.

3.3.2.4 Euler's Elastica Model

As mentioned, Elastica Model was previously introduced in Masnou et al.'s model for disocclusion [107]. In the subsequent works , [138], [137], [139] Chan et al. propose a variational approach using Masnou et al.'s elastica assumption. The motivation is derived from the previous mathematical approaches used for image restoration and removal of noise.

$$I^{\tau} = \nabla \cdot \frac{\nabla I}{|\nabla I|} + \lambda(f - I) \tag{3.12}$$

where λ refers to a penalization term, and f is the corrupted image. The model in can be interpreted as the joint minimization *total variation, (TV)* inside ω and the fidelity measure outside. Overall, it is a *bounded variation, (BV)* and the existence of the solution comes inherently. That is, a BV model requires the existence of data and the prior information. In this model, the data is used from the known part of the image and the elastica-curves function high-order (second) geometric information. The output images demonstrate straight line connections of across the gap, which makes the model only feasible for filling small gaps. Indeed, this model further extends the model to exploit the benefits of [21] and [30], however it suffers from converging to forced linear geometry [133].

3.3.2.5 Active-Contours

Proposed by Esedoglu et al., [55] uses the variational Mumford-Shah-Euler model, as a combination of the MS model [112] that is widely used for image segmentation and the long-discussed elastica curves. This method is proposed as an improvement to [139] due to its above mentioned shortcomings.

Th former MS object-line model

$$E[I,\Gamma] = \frac{\gamma}{2}\int_{\Omega\backslash\Gamma}^{a} |\nabla I|^2 dx + \alpha length(\Gamma) \tag{3.13}$$

is optimal in simple image denoising and segmentation algorithm where the image complexity is low, nevertheless, it lacks quality in inpainting applications, such that yields solutions with shortest possible isophates such that the inpainting forced on linear structures and fails to incorporate curvi-linearities.

$$\begin{aligned} E(\Gamma) = \int_{\Gamma}(\alpha + \beta\kappa^2)ds = \alpha length(\Gamma) + \int_{\Gamma}\beta\kappa^2)ds \\ + \frac{\gamma}{2}\int_{\Omega\backslash\Gamma}^{a} |\nabla I|^2 dx + \alpha length(\Gamma) \end{aligned} \tag{3.14}$$

The MSE model is more successful in adapting to the curvilinear geometries of the objects with the addition of the elastica model and its efficient numerical realization based on the Gamma-convergence approximations by [47].

3.3.2.6 Coherence Transport

PDE-based or variational methods are mostly infamous for their elongated operation intervals and computational costs. This is the aftermath of the fact that equations with high-order derivatives require more iteration for the correct solution, whereas low-orders are possible to be stabiized non-iteratively. To prevail as a rapid method, Bornemann et al. [24] propose the utilization of coherence transport in image inpainting applications. The core idea is based on the Telea et al.'s algorithm to inpaint holes in a single pass [150] by *Fast Marching Method (FFM).* Telea equation how to inpaint a pixel

$$I(x) = \frac{\sum_{q\in B_\epsilon(x)} w(x,q)[I(q) + \nabla I(q)(x-q)]}{\sum_{q\in B_\epsilon(x)} w(x,q)} \tag{3.15}$$

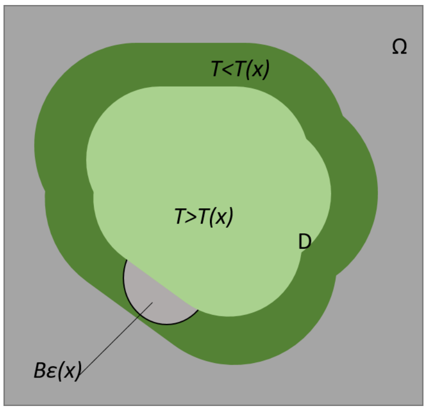


Figure 3.13: Telea's ordering of pixels on D domain based on their distances to the boundary.

where x is a pixel on the image I and B_e is an open ball around x, representing its known neighbourhood, and $w(x, q)$ is a weighting function of the intensities and local gradients.

In order to pain the whole region, the boundary between the source-and the target is needed to be narrowed, by advancing the inpainting of pixels with an ascending order with respect to their distance to the contour. By the employment of the Eikonal equation:

$$|\nabla T| = 1 \quad \text{on} \quad \omega, \quad \text{with} \quad T = 0 \quad \text{on the boundary,} \quad \partial\omega. \tag{3.16}$$

the distance map is obtained. In [24], an extra constraint of coherence transport is introduced. This first-order quasi-linear modelling enforces the propagation of the outside-boundary information along the dominant edges and features on the image, determined by the coherence field. The computation of the field is done by taking the eigenvector corresponding to the minimum eigenvalue of the image structure tensor. Similar to the Eikonal equation, the employment Dirichlet constraint allow the FFM to be realized. Hence, the pixels, one at a time, are filled according to the closeness to the boundary, using again the Euclidian distance. In this newer model, the weighing function is stated as

$$w(x, q) = \frac{\pi}{2} \frac{\mu}{|x - q|} e^{-\frac{\mu^2}{2\epsilon^2}|C^{\perp}(x)\cdot(x-q)|^2} \tag{3.17}$$

and the differential-equation is given by

$$I_t = -\nabla^{\perp} \triangle I_{\sigma} \cdot I, \quad I_{\sigma} = GH \star I \tag{3.18}$$

where HG is a Gaussian heat kernel with standard deviation of σ. The structure tensor for a grayscale image is given as

$$J_\rho(\nabla I_\sigma) = GH_\rho \star (\nabla I_\sigma \bigotimes \nabla I_\sigma) \tag{3.19}$$

which is positive semi-definite 2x2 matrix at each point p of the image. The eigenvector $C(x)$ of the minimal eigenvalue, $\lambda_1(x)$ define the flow direction, and the field strength is computed through

$$\mu = \begin{cases} 1 & if \quad \lambda_1(x) = \lambda_2(x) \\ 1 + \kappa e^{\frac{-\delta_{quant}^4}{(\lambda_2(x)-\lambda_1(x))^2}} & otherwise \end{cases} \tag{3.20}$$

where δ_{quant} is the difference of two successive gray levels to stand for the resolution quantization.

[105] is a subsequent work where the distance functions are adapted to a new metric of harmonic interpolation for more visually desirable effects, and once again the framework is improved by adding more strict constraints on the image geometry as a whole for the sake of well-posedness of the problem with the introduction a more recent work [106].

3.3.2.7 Other Works

Other studies on variational inpainting include methods based on joint interpolation of grayscale isophates [11], crack modelling with Neumann boundary condition [9],heat transfer model [6], strong continuation [18]. More recent work is focused on curvilinear structure propagation [175], nonlinear second order diffusion [13], hybrid second-forth order diffusions [12] topological gradient analysis and contour detection [7], [8].

3.3.3 Tests on Geometry-Oriented POV

The brief review on the geometry-based approaches demonstrated that even if the modeled natural phenomena or the PDE model changes to capture the continuum of the isophates and edges, one thing is common. They propagate known the pixel information and produce the novel pixel values for the unknown by continuous smoothing and mathematical manipulation due to locality which in case of large holes reaches an increased amount and results in extremely blurred outputs. Though the literature results demonstrate that they are more appropriate for crack and stain repairing approaches, for the sake clarity in this research, some example methods will be tested for performance.

Bertalmio et al.'s *Image Inpainting*

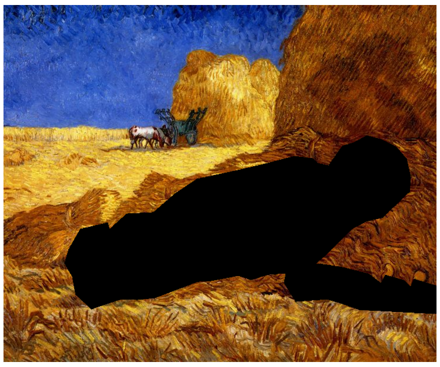
Image Inpainting [21] is chosen for being a state-of-the-art method and fathering the name of the problem domain. The formulation of the partial differential equation in the equation 3.9 is provided sequentially in the original paper. The method, except for the image and the region mask, accepts no input for parameter tuning. The only required user-specified value is the number of iterations for inpainting as in each iteration the values of the outside boundary pixels are propagated inside to paint the missing pixels gradually. As it is an iteration-based method, the inpainting operation takes elongated intervals. The performance of the algorithm on one of the test images used in the system is provided in Figure 3.14.

The run time for the given image of 607x376 pixels is 15 minutes for nearly no apparent coverage of the hole at all. Only a closer look reveals that a thin slice close to the boundary is completed, with an apparent blur in the colours. It is obvious that the method is not eligible for large object removal tasks.

Fast Coherence Transport Inpainting

The underlying reason for the selection of this algorithm is the non-iterative one shot approach employed, even though it is a PDE based method. However, FCT Inpainting has the shortcoming of the requiring very fine tuning of its user-given parameters, which are κ to determine the sharpness of the image, ϵ for data propagation neighbourhood, σ and ρ for pre & post values for smoothing the image. The usage of these parameters in the explanation of the algorithm in 3.3.2.6 and the inpainting result is in Figure 3.15.

(a) Original Image


(b) Region Mask

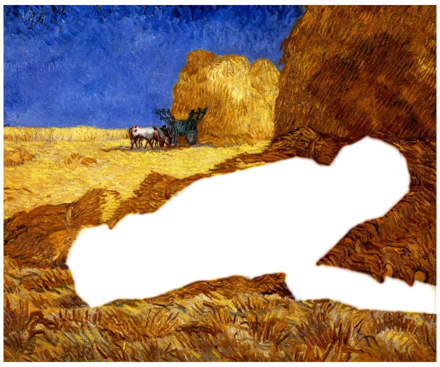
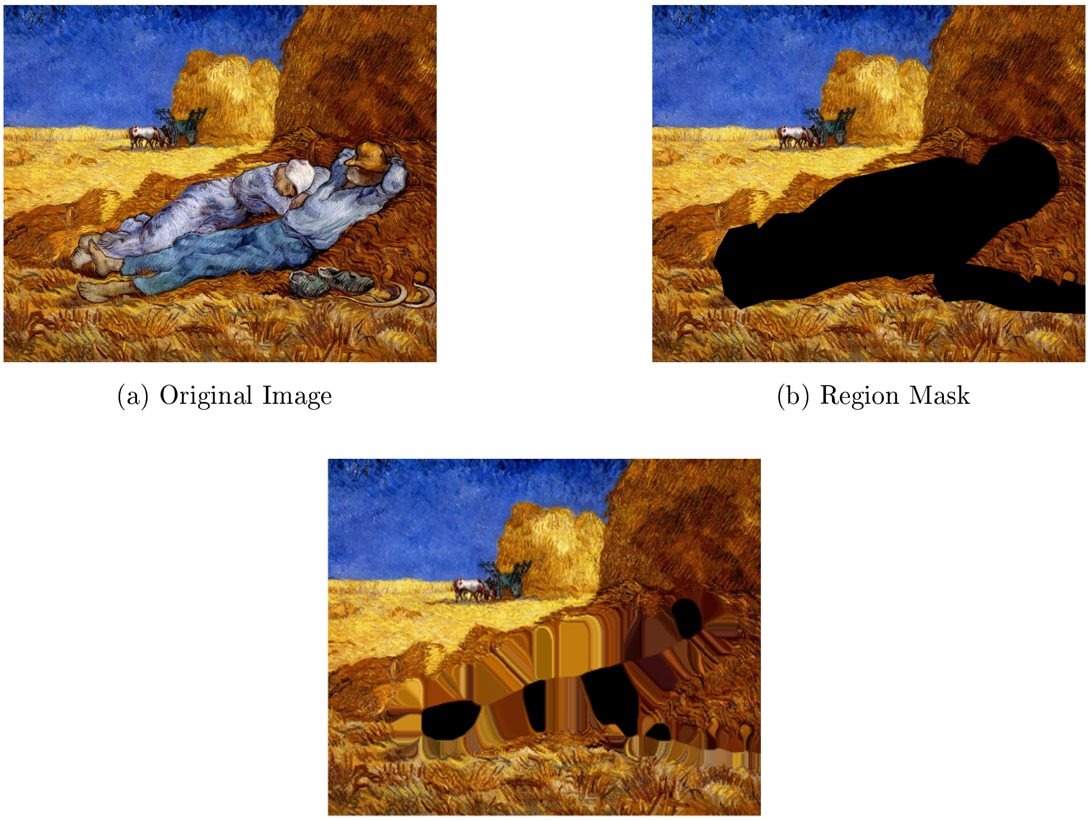

(c) 2000 iterations

Figure 3.14: (a) Original Image. (b) Region mask Λ. (c) Inpainted Image for 2000 iterations on 607x736 pixels using Bertalmio Image Inpainting [21]

3.3.4 Extra Results on Crack Filling

As explained in the literature anlaysis and demonstrated in the experimental results, the pure geometry-oriented inpainting algorithms are infamous for blurring the image significantly, especially for object removal issues [50]. They are more appropriate for crack filling and stain removal applications and so as to provide an optional feature to the image-to-video converter system in this thesis, the autographs on the paintings are regarded as cracks. In these regions, the ratio of *Perimeter/Area* is larger to allow the propagation of boundary data more easily. The results of the application of the Bertalmio et al.'s Image Inpaiting on example images are provided in Figure 3.16 & 3.17 and the results of the FCT Inpainting are illustrated in Figures 3.18 & 3.19.

In Figure 3.16, the region mask 3.16b is made to cross the image boundaries on purpose to demonstrate how the inpainting model works. Without the isophate


(a) Original Image

(b) Region Mask

(c) $\epsilon = 25$ $\kappa = 5$ $\rho = 3$ $\sigma = 2$

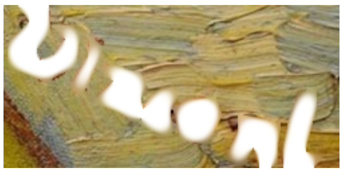
Figure 3.15: (a) Original Image. (b) Region mask Λ. (c) Inpainted Image for 2000 iterations on 607x736 pixels using Fast Coherence Image Inpainting [24].

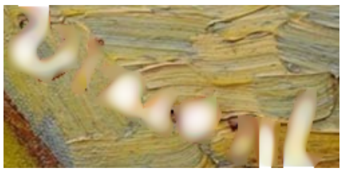
information outside the mask, the image could not be completed. This one as well as 3.17 demonstrates how this algorithm works slowly and blurs the image even in smallers gaps.

The image completion lasts in a couple of second for test images acclaimed, but the undesired blurring effect is still present although not that apparent as those of Bertalmio Image Inpainting [21].

(a)

(b)


(c) 100 iterations


(d) 1000 iterations

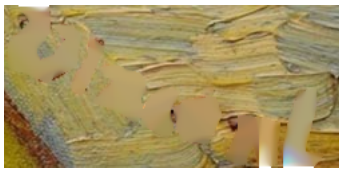
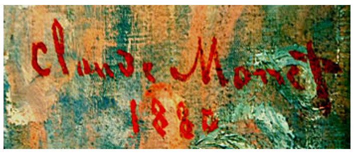
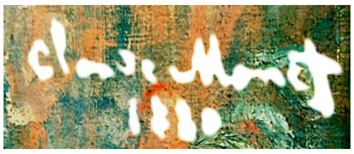

(e) 7500 iterations

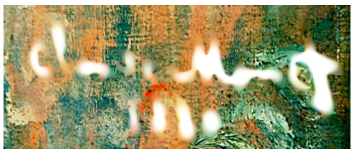
Figure 3.16: (a) Original Image. (b) Region mask Λ. (c-e) Inpainted Image for different numbers of inpainting iterations on 121x251 pixels using Bertalmio Image Inpainting [21]


(a)

(0.5)

(b)


(c) 100 iterations


(d) 1000 iterations

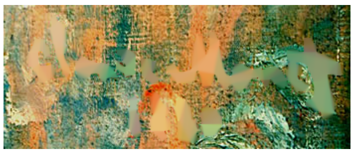
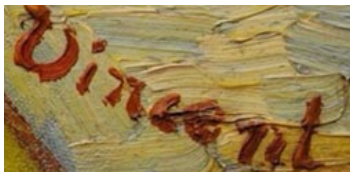
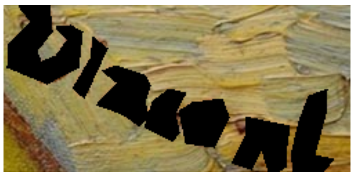

(e) 7500 iterations

Figure 3.17: (a) Original Image. (b) Region mask Λ. (c-e) Inpainted Image for different numbers of inpainting iterations on 145x349 pixels using Bertalmio Image Inpainting [21]


(a)


(b)

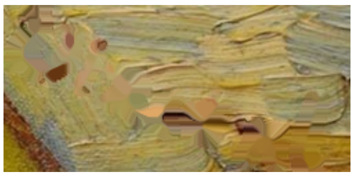

(c) $\kappa = 22\epsilon = 2\rho = 3\sigma = 2$

Figure 3.18: (a) Original Image. (b) Region mask Λ. (c-e) Inpainted Image for 121x251 pixels using Fast Coherence Transport Inpainting [24]

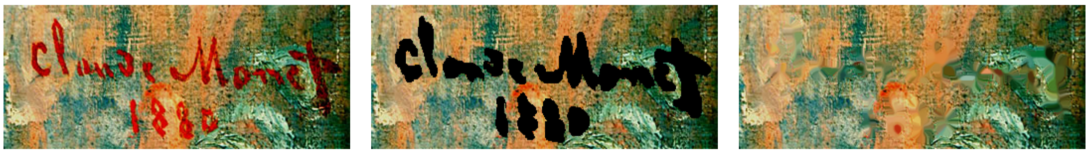

(a) (b) (c) $\kappa = 25 \epsilon = 2 \rho = 4 \sigma = 2$

Figure 3.19: (a) Original Image. (b) Region mask Λ. (c-e) Inpainted Image for 145x349 pixels using Fast Coherence Transport Inpainting [24]

3.4 Geo-texture Preserving Methods

It is discussed in the previous subsections that texture synthesis methods are self-sufficient in retouching structured or stationary regions, and similarly, PDE based methods are capable of handling partial occlusions through the use of geometric continuum assumptions in cases of corruption of the image, for instance small scratches and stains [50]. On the other hand, when the removal of larger objects is concerned, it is often the case that the background does not barely consist of the repeated tiles of the same pattern or the geometry is not that simple for basic modeling.

In real life scenery, most often the objects that have an increased degree of randomness in the outline compared to tile-textured ones with the slightest changes in the repeated pattern. De facto, in real world images, at least, composite textures, that consist of a spatial blending of several textures, are present in general [171], or at least distinct patterns are present in various regions of the frame. In structure-wise thinking, the object contours are have irregularities, are non-symmetrical and even the shapes of the objects may not be well defined. Even in the simplest case, curvilinear bodies may be coupled with sharp edges in the image frame as illustrated in Figure 3.20. Additionally, the diffusion process or the other flow mechanisms that the geometry-preserving methods obey introduce blur to the restored image and it is most evident while filling larger regions.

Subsequently, direct assignment of a texture synthesis method or PDE based model fitting for inpainting is mostly likely to end up with unnatural-looking completion of the background regions. Geo-texture methods are mainly derived

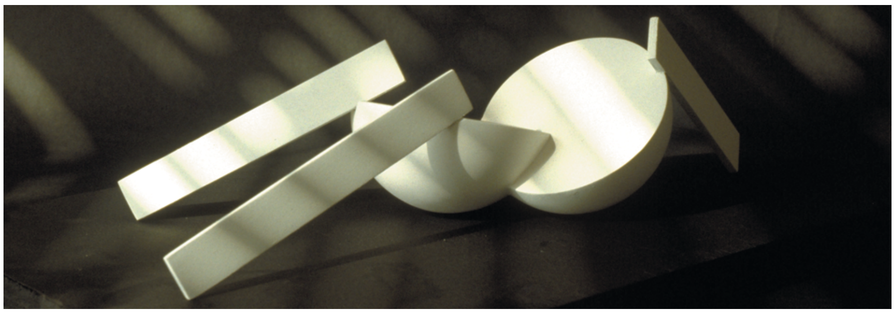


Figure 3.20: The diversity of real world object shapes.

from the both of these approaches [41], [42], exploiting its prowess on analysing the texture component of the images and inserting emphasis on the structure as well, allowing the algorithm to adapt the changes in the background, including non-repeated or non-symmetrical object edges, corners, etc. The nomenclature is trivial and as the name suggests puts an emphasis on the two significant elements of images, both the structure and the texture.

The methodology of this group is the thrive to remedy the deficiencies of the other groups of algorithms in inpainting applications by using self-information chunks and the approaches that gather under this roof tend to realize the goal by attacking two main problems of Ordering and Blending. [16].

3.4.1 Inpainting with Geo-Texture Preserving Techniques

Overall, it is possible to divide the geo-texture oriented methods into three where exemplars (group of information retrieved from the source image) are placed in an order based on pre-declared heuristics on orientation, by physical a continuity model, or by direct computations on the geometry. Within this chapter 4 algorithms that will be subjected to the major performance test will be explained in detail along with brief mentions of some other methods that are essential.

3.4.1.1 Early Prominent Models

Entropy Metric for Filling Order

One the earliest methods in exemplar-based object removal is the work in [73], where a metric of "texturedness" is defined in order to determine the filling-order of the pixels. This metric is modeled using the entropy of pixels and similarity of the pixels makes use of a weighted version the Manhattan Distance:

$$D_{isp}(s,t) = \Gamma|f_{\alpha}(s) - f_{\beta}(t)| + \sum_{t+\epsilon\in\Omega_t} \kappa(\epsilon) d_{Manh}(I(s+\epsilon), O(t+\epsilon)) \tag{3.21}$$

where $I(s)$,$O(t)$ denote two corresponding pixels in the source image, I and target image, O, respectively. Ω_t denotes the neighbourhood of pixel t; Γ, $\kappa(\epsilon)$ are weighting coefficients; and f_{α},f_{β} are uniform but arbitrary functions.

User-Aided Lines

In [147], the image completion is aided by the user-specified curves to guide the overall geometry. The provided curves are sectioned into graph nodes to determine the priority of the hole regions to be filled with priority by minimizing the energy function given below:

$$\mathcal{E}_{total}(\Omega) = \sum_{m\in\vartheta}(\beta_s E_s(p_m) + \beta_i E_i(p_m)) + \sum_{m,n\in\Phi} E_c(p_m, p_n) \tag{3.22}$$

where

$$E_s(p_m) = d(C_m, C_{p_m}) + d(C_{p_m}, C_m) \tag{3.23}$$

is a structural correspondence metric to measure the curvilinear similarity of ψ_m, the source patch and the curve segment indexed with m, using
$d(C_m, C_{p_m}) = \sum_s \in C_m ||d_{ist}(C_m(s), C_{p_m}||^2$ and $E_i(p_m)$ computes the consistency patch with the boundary pixels based on the overlapping region and lastly,

$$E_c(p_m, p_n) = |\psi_m - \psi_m|^2 \tag{3.24}$$

determines the coherency between to synthesized patches, ψ_m & ψ_n on the curve, using the normalizes SSD metric.

The rest of the hole is filled by using Belief Propagation. This method is extended for automatic detection of curves in [35] by Euler' Elastica. Interestingly, the method uses the priority metric in Criminisi [42] (a method to be explained in detail) and graphcut textures [92] explained in 3.2.2.5 to fill small gaps to accomplish a high-blend of the state-of-the-art.

Irregular Patches

In [83], a two-step inpainting method is proposed, where unsupervised image segmentation is succeeded by a tensor-voting step. The segmentation step provides the exemplars to be connected already for visual accuracy and the latter voting for the seamless blending of the regions across the holes. Despite the benefit of having the advantage of transferring compact curved structures as a hole, the algorithm is computationally costly and has little flexibility on the decision of texture/object contours, which makes it challenging to decide on the boundary of two textures.

The Emphasis on Sparsity

As being smaller components of the image and containing, at least partially, the textural and structural, inevitably colour, properties of the entire image, the patches are proved to be highly functional representative elements of an image [20]. Hence, with the success of the texture synthesis and exemplar-based methods, which are highly dependent on the use of patches, the idea of constructing dictionaries out of image patches have been blossomed.

An image representataion method by decomposition using sparce coefficients to represent geometry and texture components of the image, which are converted into an overcomplete dictionaries is put foward in [53]. It was also asserted that this dictionary representation model could be taken granted for image inpainting. According to the model,

$$I = L_g m_g + L_t m_t, \tag{3.25}$$

where L_t and L_g represent the dictionaries for texture and geometry, composed of elements with sizes $\ell \times q_g$ an $\ell \times q_t$, respectively for an image I to be shown in the vector form $\Re^M$ and $m_g \in \Re^{q_g}$ and $m_t \in \Re^{q_t}$ s a decomposition of the image. Then,

$$\underset{(m_g,m_t):I=L_g m_g + L_t m_t}{min} ||m_g||_p + ||m_t||_p, \quad p = 0, 1. \tag{3.26}$$

becomes a sparse representation of the image I.

In with the introduction of a total variation (TV) penalty function into the image decomposition and by regularizing the dictionary components of the representation, the model turns out to become a convex optimization problem.

$$\underset{(m_g,m_t)}{min} ||m_g||_1 + ||m_g||_1 + \lambda||I - L_g m_g - L_t m_t||_2^2 + \gamma TV(L_g m_g) \tag{3.27}$$

where $TV(\star)$ denotes the total variation, $\lambda, \gamma > 0$. In this model, $I - L_g m_g - L_t m_t$ is interpreted as the image noise and λ as a penalty parameter that is inversely correlated with the noise power. Inpainting problems are regarded equally as image denoising problem to be attacked in many works in literature [22], therefore the undesired part may be regarded as a region where the magnitude of the noise is extremely large. For an inpainting application, the cost function is slightly altered with the introduction of brand-new variable, M

$$\underset{(m_g,m_t)}{min} ||m_g||_1 + ||m_g||_1 + \lambda||M(I - L_g m_g - L_t m_t)||_2^2 + \gamma TV(L_g m_g) \tag{3.28}$$

to be taken binary, that could either be equal to 0 or 1, the ROI for inpainting and the remainder image that the data for completion to be retrieved are determined.

The solution of the above explained network still requires more assumptions on the model, yet the interested reader may refer to [53]. The above explained model is adapted to model dictionaries that use image patches in [1], namely the K-SVD method,where K-means clustering is merged dictionary training. The method has numerous image processing applications beyond inpainting, such as demosaicing and image denoising. Mairal et al. used a similar method inclusing multi-scale dictionaries [103] for more quality in inpainting [20].

Other dictionary based inpainting techniques include [165] , [166] where as well utilize sparse linear combinations of the image patches for inpainting, [168], [136], [59] also exploit the sparse representations of the examplars. One of the most recent dictionary based approaches using hand-crafted dictionaries is proposed by Ogawa et al. in [114]. This method alters the similarity metric in [1] with one that is more compatible with the human visual system, the structural similarity index, or SSIM.

3.4.1.2 *Criminisi*'s Method

[41] and its further detailed extension [42] form what is called to be the state-of-art methodology of exemplar based inpainting, as well as in entire study on inpainting itself. The algorithm has the driving force from texture synthesis algorithms and inpainting techniques dedicated to smaller gaps (approximation of the PDE based solutions). By exploiting strengths of both group of techniques that precede, it formulates a well-defined merger of the two, and expectedly, it does not only perform well on their application domain, but further prove to be useful in an extended problem domain: removal of larger objects from the image scene.

In [42], the approach can be summarized as the propagation of the textural information along the given structure of the objects in the image. That is, the pixel information is to be filled in, as patch groups, uses a strict ordering method which is defined by the confidence values of the pixels coupled with the data values. Notably, the rudimentary part of the modeling is the linear structures on the image, also referred as isophates, which are the key elements used to determine the filling priority of the pixels. To be broad, the algorithm becomes an exemplar-based texture synthesis method that is limited by the constraints imposed by the isophates on the image.

There are two main claims of the algorithm, the former being that no extra means other than exemplar-based synthesis is required, and the latter the filling order is crucial for image coherency preservation.

Figure 3.23 illustrates how a patch centered by a pixel c is filled in [42]. The pixels to be filled lie adjacent to the boundary that separates the source, Λ and the target region $I - \Lambda$, namely the fill front. If the bounded box BB_c, centred by c is to be filled, the most similar exemplar definitely will be found along the edge, ψ_{match}, the region where the bounded box extends to the source region, already located on a strong gradient. Following the filling of BB_c, it can be observed that the orientation of the corresponding ishopate is preserved as desired. Different from the classic texture synthesis methods explained in

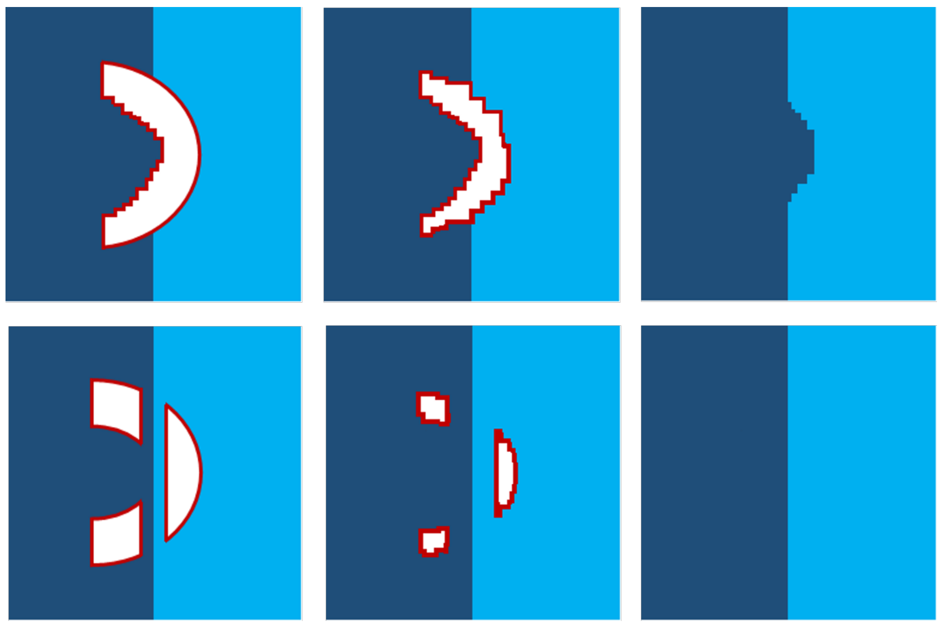


Figure 3.21: Significance of Filling Order: First row illustrates three arbitrary filling instances of the same image using the concentric filling, a.k.a onion peel method. The second row is the corresponding instances for a desired completion.

Sections 3.2.2.1,3.2.2.2,3.2.2.3, exemplars are arbitrary sized full patches, that is instead of the usage of single pixels as texture building elements. This provides the algorithm to be flexible on the synthesis procedure where single pixels may more often fail to capture the whole features of the textures in complex scenes [20].

Though it is possible to maintain the linear structures of the image as shown above, the extent of preservation is dependent on how the prioritization of filling is determined. In order to achieve high quality in the completed image, the filling procedure is to give higher cardinality to the neighbourhoods on the linear continuum of the fill front. Figure 3.22 illustrates how this prioritization produces more robust operation against variations in the shape of the objects. For the sake of clarity, the proposed filling order is compared with the concentric-filling.

As observed, the former shows more strength in the preservation the edges, as the onion-peel method results in a more curvilinear filling, contrary to what is desired. Furthermore, the proposed algorithm is expected to prevent blocking-effect artefacts that is caused by unnecessary protrusions, which is the case for the concentric filling of the image regions.

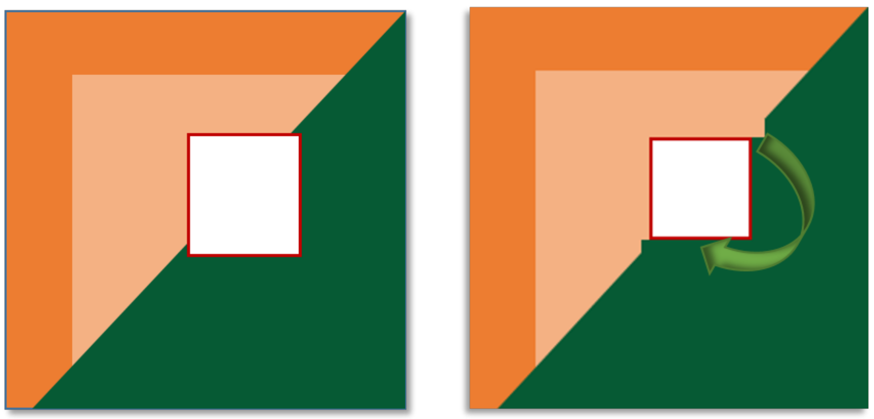


Figure 3.22: Formation of Protrusions: The undesired region overgrowing due to the onion-peel filling order. (a) Initial Image & Gap. (b) An arbitrary iteration instance in clock-wise concentric filling.

Filling Algorithm

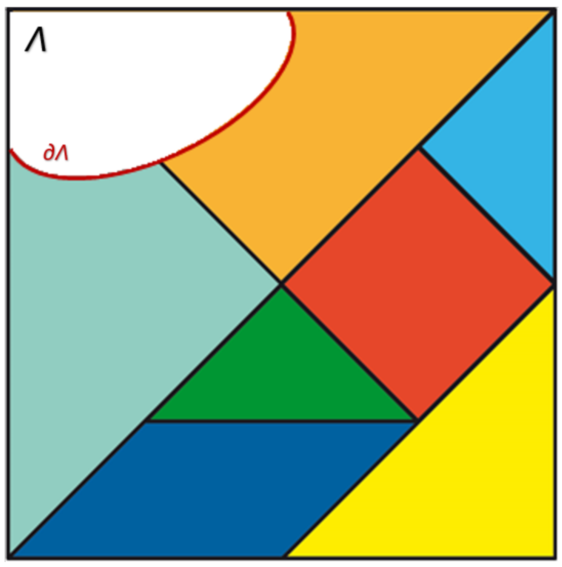
For each pixel in the image, two information is considered essential, initially for prioritization and then the filling procedure. Colour information is the exact intensity value of that pixel, which is taken equal to 0 for the target region and the confidence information that is computed by a proposed metric that determines the filling order.

The filling procedure is composed of two steps, that are iterated until all the target region is filled, in an interleaved manner: *Priority Computation* and *Sampling*.

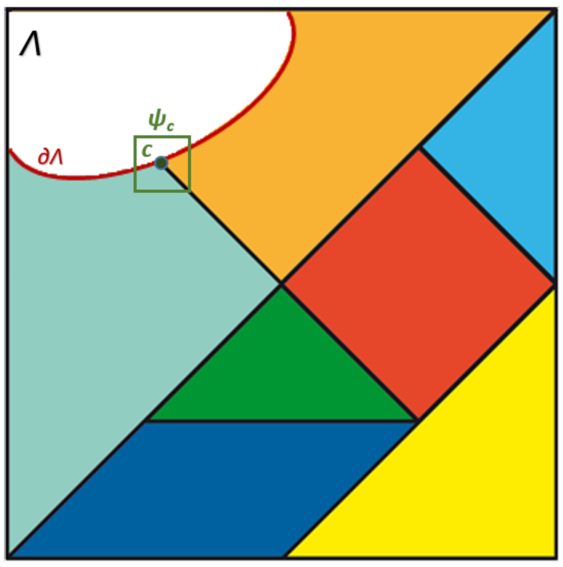
Priority Computation

Succeeding the determination of the fill front, which is computed by the bi-directional Gaussian filtering of the mask, denoting the region to be filled, the prioritization of the candidate patches is realized. From the previous discussions on the assumptions of the algorithm, it is evident that the pixels that lie along the linear continuum of sharp edges that intersect the source-target region contour are prioritized for the sake of the preservation of the image isophates, and so do the ones that are located in the adjacency of the high-confidence pixels, namely the ones whose neighbourhood information is nearly complete.

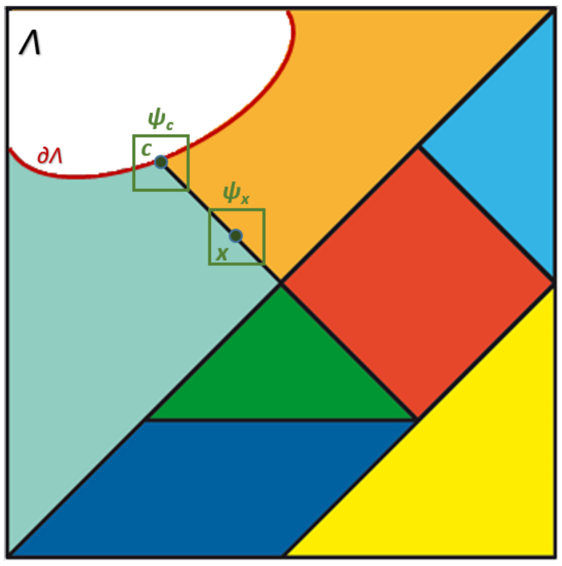
Let ψ_x be a patch centred around the x, a pixel on I, then the priority metric



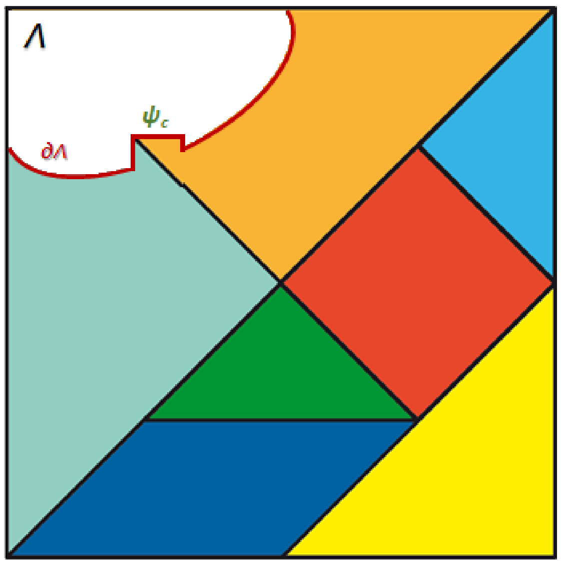
(a)



(b)



(c)



(d)

Figure 3.23: Proposed Filling Method [42]: (a) Initial Image. (b) The prioritization of pixels on the boundary. (c) Computation of the Best Matches. (d) Corresponding filling iteration.

to compute $\mathcal{P}(x)$ is defined as

$$\mathcal{P}(x) = \mathcal{D}_x \star \mathcal{C}_x \tag{3.29}$$

where $\mathcal{D}_x$ is the data and $\mathcal{C}_x$ is the confidence term assigned for x and its containing patch, ψ_x.

$$\mathcal{D}_x = \frac{|\nabla I_x^{\perp} \cdot n_x|}{\gamma} \tag{3.30}$$

$$\mathcal{C}_x = \frac{\sum_{\xi} \mathcal{C}_{\xi}}{\mathcal{A}_{\psi_x}} \tag{3.31}$$

where $\xi \in \psi_x \cap (I - \Lambda)$, γ is constant or normalization, $\mathcal{A}_{\psi_x}$ is the area of the patch psi_x, and n_x denotes the unit vector orthogonal to the fill front, $\partial\Lambda$, at x.

The initialization of the confidence terms at the very beginning of the algorithm are realized as follows,
$\mathcal{C}_x = 0, \forall x \in \Lambda$, and $\mathcal{C}_x = 0, \forall x \in I - \Lambda$. Since each one the patches whose centres are located on the source-target region contour are candidates to be eligible to be inpainted first, the priority computation is done for all and then they are sorted in a descending order.

If it is expected to comment to the functionality of the priority metric, it is possible to state that the confidence term, $\mathcal{C}_x$ inserts bias towards the presence of solid data for further computation and $\mathcal{D}_x$ favours regions that coincide with the pathway of high order features intersecting the fill front. It provides morphological closure and completion of image edges in accordance with "Principle of Connectivity" [85]. When combined, the two measures subdue each other for the desired equilibrium of constrained inward growth.

Sampling, Propagation & Update
Priority sorting of the candidate patches has demonstrated which one is to be filled before all the others. Let ψ_t be have the utmost priority, then data propagation towards inpainting area is achieved by sampling the $I - \Lambda$ region where an exemplar patch, ψ_m is searched for in order to satisfy the following constraint:

$$\psi_m = argmin_{\psi_{m'} \in I-\Lambda} = d(\psi_t, \psi_{m'}) \tag{3.32}$$

where the sum of squared distances is used as the similarity criterion. It is important for the algorithm to function correctly that the entire exemplar must be enclosed by the source region. The best exemplar in the source region is used to copy the corresponding pixel information from the source to the target region. Pixel data is copied in the form of rectangular blocks called texels as opposed to single pixels as was the case in texture synthesis methods [52], [162], [5] discussed, there occurs the possibility of overpainting a pixel for several pasting instances. The overpainting is already time and memory consuming and its results in a patchy, artificially filled region due to visual blocking effect. So as to prevent these shortcomings, only the pixels that belong to ψ_m, $p_\imath \in \psi_m \cap \Lambda$, $\imath = 1, 2.., patchsize$ that overlap the target region is filled.

Once the filling is completed, confidence map of the image is to be updated for the processed pixels. During the priority computation phase, the patches at the fill front are assigned with temporary confidence values; however, when the filling of any pixel is completed, the corresponding confidence values are fixed to be 1 until the whole procedure is completed. Precisely, it is the exact same procedure of propagating the confidence map patch corresponding to the exemplar towards the patch to be filled:

$$\mathcal{C}_t = \mathcal{C}_m, \quad p_\imath \in \psi_m \cap \Lambda, \imath = 1, 2.., patchsize \tag{3.33}$$

Frankly, in order to rapid up the implementation, gradients to compute data values may also be copied to the target region using an appropriate mask to omit the pixels that are not filled in this iteration.

The proposed two-fold filling method proceeds until all the pixels in the target region are filled with appropriate information, or in other words, until no contour separating the source and the target regions is evident. It is rudimentary to notice that, in each instance of inpainting, the fill front is to be recomputed, since with the previous pasting instances the fill front changes its orientation, so do the data terms accordingly. In addition to that, the usage of bigger patches prevents the direct employment of the onion-peel prioritization for the pixels in the lambda region. Therefore, the algorithm adopts a distinct pasting method that it skips the pixels that are already filled.

The method that is proposed by Criminasi et. al is such an acclaimed method in inpainting field, a state-of-the-art, there is a plethora of work based on its core idea, even algorithms named after it. The methods include, [113], [79] with new metric, [170] metric watershed and segmentation, [72] fractional derivatives.

3.4.1.3 PatchMatch

PatchMatch [14] is indeed not an algorithm to be used as a sole image inpainting method, but rather as an interactive tool to edit images to by rapidly determining the approximate nearest neighbours matches for the patches of an image. The key advantage of the method is that it reduces the cost functions to be computed such that it enables interactivity. The main motive behind the method is the possibility of achieving proper matches in the image via random sampling and propagating these matches to the areas that surrounds the initial one by an assumption on the natural coherency of the image. Apart from inpainting, it can be used for the high-level editing applications such as retargeting and shuffling [14].

The vitalities of image editing applications, as inpainting and image completion are two of them, are the flexibility and rapidity. Patch sampling in the image is known method to satisfy the first criterion as the result of the previous works, [161], [77], [143] as they implement the dense sampling at various scales using small image patches, and are capable of generating similar images to the input with respect to structure and texture. Nevertheless, they are not eligible to be used interactively due to their elongated computation intervals, which indeed fails the latter criterion. PatchMatch is faster than these methods even in orders of magnitude [14].

Non-parametric sampling [52] of the image includes searching the entire group of target patches, in a repeated fashion, so as to reach the utmost similarity for each and every patch in the group domain patches. Let S be the domain of image patches and T be target region. For each patch ψ_s in the domain there is a mapping to ψ_t , a patch in T, under some predefined metric $\mathcal{M}$. In [52], this one-by-one matching is named as the $NearestNeighbourField$, (NNF).

In non-parametric patch sampling methods, the search step is the phase that slows down the operation, *id est.*, the bottleneck. Attacking this problem with different search algorithms offer different complexity functions, such that brute force is rather computationally costly, that reaches $O(pP2)$ in time, where the image has p pixels and is composed of patches of size P. Tree-based search acceleration structures, [162] tend to have at least $O(M)$ utilization of the memory, preventing these structures to be optimal on high-resolution image editing.

Due to that complexity in achieving a fast algorithm to compute NNF mapping, Barnes et. al put forward three postulations to form the backbone of this search method.

- Dimensionality of Offset Region: Unlike, what precedes in literature, this very method spans the 2-D region of patch offsets, still yielding to be highly rapid and memory-efficient, whereas many related previous work include handle with the lack of speed in operation by manipulating the dimensionality of the domain of patches by employing tree structures, such as K-d trees or by direct dimensional reduction methods, Principal Component Analysis, etc.
- Natural structure of images: Pixel-wise search methods fail to grasp the coherency properties of the image, however; texture synthesis algorithms that are based on patch-sampling produce output images that have adherent lumps of information retrieved form the original image itself [5]. Hence, it is beneficial to assume continuity in the neighbouring patches, such that more reasonable mapping are obtained by seeking the matches in the adjacency of the corresponding pixels.
- The law of large numbers: As the image dimension is increased, the probability of achieving reasonable matches diminishes, hence, some non-arbitrary initial assignment may be useful for better results.

Approximate Nearest Neighbour Search

The overall output of the search algorithm is the *approximate nearest-neighbour field (ANNF)* that is mapping from the patches ψ_s of the domain image, S, to the corresponding best matches ψ_t in the target image, T. Precisely, $ANNF$ may be regarded as a mapping function, Φ

$$\Phi : T \to \Re^2 \tag{3.34}$$

where $\Phi(\psi_s)$ is the offset of the nearest-neighbour patch centre, namely centre of ψ_t, with respect to the centre of ψ_s , making $\Phi(\psi_s) = Centre(\psi_s) - Centre(\psi_t)$. Therefore, $\Phi(S)$ refers to a matrix that gives the distances of $\forall\psi_s$ to the $\exists\psi_t$ for the defined metric. The algorithm to obtained the final mapping is 3-fold and can be explained as follows:

Initialization
The initialization is accepted to be performed by either random value assignment to the Approximate Nearest-Neighbourhood Field, or by inclusion of some apriori information. It is to be noted that when it is the case to make use of random values, it may be accompanied with a pyramidal gradual processing of the images.

Iteration: Propagation & Query
The latter step of the algorithm is the repeated iterations performed to ameliorate the Approximate Nearest Neighbour Field. For satisfying correspondence values in $ANNF$, propagation towards the adjacent pixels is applied, and it is succeeded by random search so that the best match among these could be taken as the update value for the corresponding $ANFF$ entry. The reason behind propagation of the data is to take advantage of the coherent nature of the image in matching and to yield higher computational efficiency, whilst the random query is used to increase the probability of a better match with a diversity of candidates and to prevent the matches being trapped in local extrema. In the iteration phase, for each patch ψ_{s_ℓ}, $\mathcal{P}_\ell$ and $\mathcal{Q}_\ell$ steps are handled in an interleaved manner such the given order is followed until all the patches are processed: $\mathcal{P}_1$ $\mathcal{Q}_1$ $\mathcal{P}_2$ $\mathcal{Q}_2$ ..... $\mathcal{P}_\ell$ $\mathcal{Q}_\ell$ .... $\mathcal{P}_\mathcal{L}$ $\mathcal{Q}_\mathcal{L}$. The default scanning order is from left to right, from top to bottom.

Algorithm 2 PatchMatch

```
    function PM($S, T, M$)                ▷ $S, T$ images, $M$ NNF from $S$ to $T$
2:      $InitializeM^0$: randomly or use apriori mapping
        while $M^\tau != M^{\tau-1}$ do
4:          for $\ell = 1 : \mathcal{L}$ do
                $Propogate :$
6:              if $\tau \mapsto odd$ then
                    $\Phi(\psi_s^{\ell' x-1}, \psi_t^\ell)$, $\Phi(\psi_s^{\ell' y-1}, \psi_t^\ell)$ are propagated to $\Phi(\psi_s^{\ell'}, \psi_t^\ell)$
8:                  $dist(\ell) = argmin(\Phi(\psi_s^{\ell' x-1}, \psi_t^\ell), \Phi(\psi_s^{\ell' y-1}, \psi_t^\ell)), \Phi(\psi_s^{\ell'}, \psi_t^\ell)$
                    $M^\tau(\ell) = dist(\ell)$
10:             end if
                if $\tau \mapsto even$ then
12:                 $\Phi(\psi_s^{\ell' x+1}, \psi_t^\ell)$, $\Phi(\psi_s^{\ell' y+1}, \psi_t^\ell)$ are propagated to $\Phi(\psi_s^{\ell'}, \psi_t^\ell)$
                    $dist = argmin(\Phi(\psi_s^{\ell' x+1}, \psi_t^\ell), \Phi(\psi_s^{\ell' y+1}, \psi_t^\ell)), \Phi(\psi_s^{\ell'}, \psi_t^\ell)$
14:                 $M^\tau(\ell) = dist(\ell)$
                end if
16:   ▷ (Please note that propagation is realized in the reverse direction in the
    even iterations)
                $Query :$
18:             $\nu = dist(\ell)$
                while $\omega^\jmath \gamma \beta_\jmath \leq 1$ do
20:                 $\upsilon^\jmath = \nu + \omega \gamma^\jmath \beta_\jmath$                ▷ $\omega$ determines the
    upper limit for the search radius, $\gamma$ is used to shrink the search window in
    successive queries with a fixed ratio, and $\beta$ stands for the shift value which
    is uniformly distributed in [-1,1]×[-1,1]
                    $dist = \Phi(\psi_s^\ell, \psi_t^{\ell+\upsilon^\jmath})$
22:                 if $dist < \nu$ then
                        $M^\tau(\ell) = dist(\ell)$
24:                 end if
                end while
26:         end for
            $\tau + +$
28:     end while
    end function
```

The method to use PatchMatch in inpainting problems will be explained in section 3.5.1.

3.4.1.4 Coherency Sensitive Hashing

[90] is a merger of [44], Locality-Sensetive Hashing, LSH, an approach to the computation of Approximate Nearest Neighbour Fields and the well-functioning patch-matching algorithm in [14], PatchMatch, to propose a novel method for estimating ANNFs between two images. Though it is not stated explicitly as an inpainting method, in this thesis work it is further extended to be used in region-filling.

Coherency-Sensitive Hashing (CSH), replaces the traditional tools, such as KD-trees [4] or hashing algorithms with a novel approach to compute the similarity between two patches in close accuracy to these methods but in more rapid means. In the query mechanism, it exploits the functionality of LSH method, hence it will be wise to take a brief look on how it is formulated.

Revisiting Locality-Sensitive Hashing

Let $\mathcal{U}$ represent the sample space of objects. A similarity metric can be defined as in relation with $\mathcal{U}$ as

$$\mathcal{S} : \mathcal{U} \times \mathcal{U} \leftarrow [0, 1] \tag{3.35}$$

An example of LSH for $\mathcal{S}$ is a probability distribution over a set $\mathcal{H}$ of hash functions

$$\mathcal{P}_{rob}^{h \in \mathcal{H}}[h(x) = h(y)] = S(x, y) \tag{3.36}$$

where for $\forall x, y \in \mathcal{U}$

In other words, the probability of elements x and y being hashed in to the same hash-bin is defined to be equal to the similarity between x & y, with respect to the given similarity function, $\mathcal{S}$. Therefore, the hash collision instances occur proportionately to element similarities. A further definition can be made on both similarity and dissimilarities to make a “gap definition”:

$(r, R, \pi, \Pi)_{LSH}$ with respect to S on H can be defined as

$$\mathcal{S}(x, y) \geq R \Rightarrow \mathcal{P}_{rob}^{h \in \mathcal{H}}[h(A) = h(B)] > \Pi \tag{3.37}$$

$$\mathcal{S}(x, y) < \rho \Rightarrow \mathcal{P}_{rob}^{h \in \mathcal{H}}[h(A) = h(B)] < \pi \tag{3.38}$$

for each $x, y \in \mathcal{U}$, where $r < R$ and $\Pi > \pi$

LSH method is used ANNF computation in [65], [44], [27] in literature and the implementation consists of a bi-step procedure with stages dedicated to indexing and query. The details of both of these steps will be provided through the explanation of CSH algorithm.

CSH for Approximate Nearest Neighbour Neighbourhood Computation

The algorithm is bi-folded as in the case of LSH based computation, the mother algorithm where the search elements, namely the non-overlapping patches of size $\ell - by - \ell$ are regarded as ℓ^2 in Euclidean space.

Indexing

Projection: The primary phase of indexing step of the algorithm may also be referred as the dictionary-building phase, where the image patches are manipulated by selected functions to form sparse projection dictionaries out of them. In CSH [90], these functions are specified as 2D Walsh-Hadamard [17] functions that have the following kernels:

$$WH_0 = 1, WH_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, WH_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{bmatrix}, \cdots$$

$$WH_n = \begin{bmatrix} WH_{n-1} & WH_{n-1} \\ WH_{n-1} & -WH_{n-1} \end{bmatrix}$$

The work in [17], [76] prove that the projections on 2D WH kernels are incredibly potent in representing the image, when arranged in an order of ascending frequency. This representational power owes to the fact that when they are arranged optimally to form a projection line [129], the scattering of the projected patches are maximized. The dispersion is a critical measure in determining the

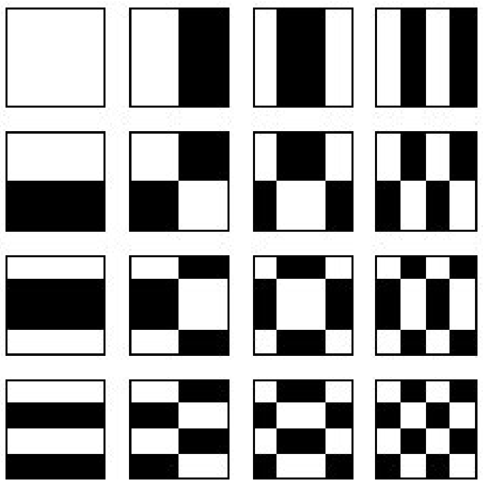


Figure 3.24: A visual of the ordered 2D Walsh-Hadamard Kernels up to 2^{nd} order.

patch similarities since, the fundamental thrive is enlarge the gap between the related patches and unlike ones.

Hashing: The eigenvalues of the covariance matrix, containing each image patch, is computed and the largest one is taken. The bins are generated by repeating the procedure for kernels concatenated in increasing frequency. Hence, image patches that are contained in the same hash-bin are indicated with an identical index value. These bins form what is called as the hash table for the searching algorithms.

2D WH kernels are a substitute for primitive hash functions in LSH, which are apt to a linear form defined as:

$$h_{\alpha,\beta}(x) = \frac{\alpha x + \beta}{\rho} \tag{3.39}$$

where ρ is an arbitrary integer, $\beta \in [0,\ \rho]$ and a is a p-dimensional vector consisting of random samples from a Gaussian distribution. However, this basic model is not satisfyingly effective in amplification of scattering in separate bins.

Query:
Although hashing the patches using modified-LSH method provides candidates for the image patches to be matched, their diveristy is very limited. In order to profit from the spatial arrangement and principle of coherency via data propagation [14], the generation of brand-new candidates on these bases is required.

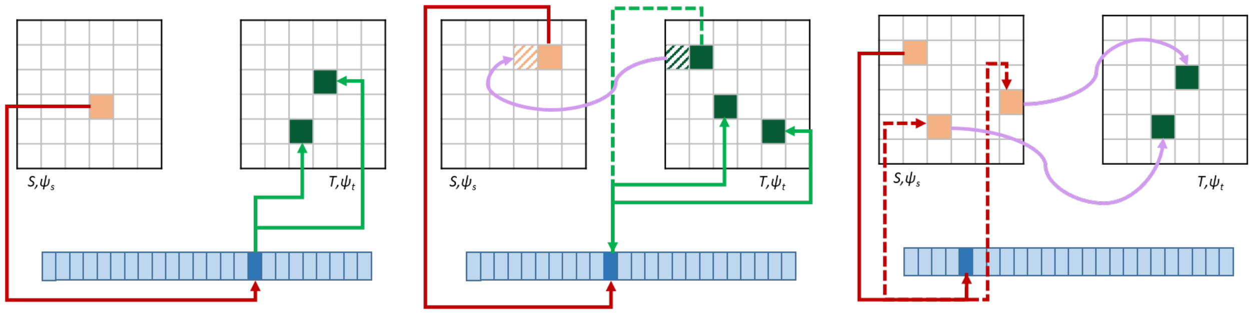


Figure 3.25: The three group of candidates, $\mathcal{C}_{and}(\psi_s)$ generated based on the observations. (a)$\mathcal{C}^{I}_{and}$: LSH driven candidates. (b)$\mathcal{C}^{II}_{and}$: PatchMatch driven candidates. (c)$\mathcal{C}^{III}_{and}$: CSH driven candidates.

Candidate Selection: Once the hashing of each patch in both images are completed, certain observations on the matching patches are made. For image patches ψ_s & ψ_t in images S & T respectively, under the hash function $\mathcal{H}$ provides the following:

1. $\mathcal{H}_S(\psi_s) = \mathcal{H}_T(\psi_t)$, then $\psi_t \in \mathcal{C}_{and}(\psi_s)$
2. ψ_t is a candidate for ψ'_s and $\mathcal{H}_S(\psi_s\prime) = \mathcal{H}_S(\psi_s\prime\prime)$, then $\psi_t \in \mathcal{C}_{and}(\psi_s\prime\prime)$
3. $\psi_t\prime$ is a candidate for ψ_s and $\mathcal{H}_T(\psi_t\prime) = \mathcal{H}_T(\psi_t\prime\prime)$, then $\psi_t\prime\prime \in \mathcal{C}_{and}(\psi_s)$
4. This observation is two-fold:
   - If ψ_t is a candidate for $Left(\psi_s)$, then $Right(\psi_t)$ is a candidate for ψ_s and vice versa
   - If ψ_t is a candidate for $Top(\psi_s)$, then $Bottom(\psi_t)$ is a candidate for ψ_s and vice versa

Figure 3.25 illustrates the candidate set,$\mathcal{C}_{and}(\psi_s)$, corresponding to the source patch, ψ_s can be partitioned into 3 groups. $\mathcal{C}^{I}_{and}$ is what LSH based query enforces, where candidates are only limited to the ones that are hashed into the same bin, manipulating direct similarity between patches in different images. $\mathcal{C}^{II}_{and}$ has its inspiration from [14] in which image coherency through propagation is the core element to build ANNFs. Last but not the least, $\mathcal{C}^{III}_{and}$ is a proposition of this method, and it examines the possibility of finding best matches in terms of hash collisions.

Candidate Elimination: In order to avoid the shortcomings of the widely used L_2 basis [5] in direct utilization and due to the necessity of processing an increased number of candidates with timing considerations, [90] uses the following ranking method among the candidates proposed in [76]. A secondary projection of candidates on the 2D WH kernels realized, with the constraint that in this step are the used ones are the first few kernels to be able to preserve most of the patch energy. Finally, the candidates are discarded one at a time if their projected SSDs with patch ψ_s fail decrease the lower bound for the corresponding ANNF entry.

The application of CSH to inpainting problem will be discussed in secton 3.5.1.

3.4.1.5 Image Melding

Proposed by Dabari et al. [43], Image Melding is a more generalized method devoted to not only to inpainting but also to other image editing applications. However, for the sake of precision, the explanation will merely focus on the image completion algorithm. The superiority of this current method over the other focus algorithms is that this exemplar-based method takes the deformation incidents of patches into consideration, which is indeed an inherent means to involve the structure element within image inpainting applications.

Let S and T be the source and the target images, the joint energy function to be minimized is defined as follows:

$$\mathcal{E}_n(S,T) = \sum_{t \in T} min_{s \in S}(d_{ist}(\psi_{s'}, \psi_t) + \lambda d_{ist}(\nabla\psi_{s'}, \nabla\psi_t) \tag{3.40}$$

where ψ_s,ψ_t are patched located in S,T images with center pixels s & t, respectively and $\psi_{s'}$ corresponds to the deformed version of ψ_s obtained by exposing ψ_s geometric and/or photometric transformations. Denoted by $\mathcal{D}$, these transformations involve translation as was in the other methods, along with rotation, reflection and scaling with non-uniform ratio. Conclusively, the query for the best match involves the restricted deformation of the source patches to fit onto the target.

The implementation is on the Cie-Lab colour space where spatial uniformity is assumed. In equation 3.40 the derivative terms, namely, $\nabla\psi_{s'}, \nabla\psi_t$ stand for the gradients of the luma channel, $\mathcal{L}$ in the selected colour space, to emphasize the local features on the patches in the energy function, with a penalty term λ. This enhances the algorithm in a way that the it becomes more robust in capturing high frequency representations and embodying the local geometry continuum.

A inpainting method similar to [14], [90] is implemented in a coarse-to-fine approach where in each scale the algorithm performs two interleaved operations:

Query

The search algorithm is similar to [14] in coherence propagation and random search; however the similarity of the deformed patches under the aforementioned set of transformations, $\mathcal{D}$ is also tested using the SSD similarity metric. It is important to remind that the search in the algorithm uses sliding windows for query instead of distinct patch divisions to prevent overlaps.

Additionally, so as to make the algorithm more versatile to adjust the slight changes in exposure and illumination, a confidence interval is obtained. This method originates from [68] that discusses image enhancement metrics. The interval is obtained by computing two terms, bias β ,and gain γ.

$$\gamma(\psi_{s'}^{\imath}) = min\{max\{\sigma(\psi_t^{\imath}) - \sigma(\psi_{s'}^{\imath}), g\}, G\} \tag{3.41}$$

$$\beta(\psi_{s'}^{\imath}) = min\{max\{\mu(\psi_t^{\imath}) - \gamma(\psi_{s'}^{\imath})\mu(\psi_{s'}^{\imath}), b\}, B\} \tag{3.42}$$

where the predefined range for γ and β are $[g, G]$ and $[b, B]$, respectively, $\imath$ stand for each of the channels L,a,b in the colour space, and μ, σ define the mean and the standard deviation moments. The confidence adjustment is by γ and β:

$$\psi_{s'adj} = \gamma(\psi_{s'})\psi_{s'} + \beta(\psi_{s'}) \tag{3.43}$$

Voting

Output image is obtained by solving the optimization problem defined on the energy function in 3.44. In more formulated manner,

$$O = argmin_S(d_{ist}(S, \overline{T}) + \lambda d_{ist}(\nabla S, \overline{\nabla T}) \tag{3.44}$$

where

$$\overline{T}(m,n) = \sum_{\imath \leq patchHeight} \sum_{\jmath \leq patchWidth} \frac{\mathcal{NN}(\psi_t^{m-\imath,n-\jmath})(\imath,\jmath)}{\mathcal{A}_{rea}(\psi_t)}, \tag{3.45}$$

$$\overline{\nabla T}(m,n) = \sum_{\imath \leq patchHeight} \sum_{\jmath \leq patchWidth} \frac{\nabla\mathcal{NN}(\psi_t^{m-\imath,n-\jmath})(\imath,\jmath)}{\mathcal{A}_{rea}(\psi_t)} \tag{3.46}$$

In 3.45 $NN(\star)$ corresponds to the best match for the corresponding pixel. Indeed, the minimization of energy is realized not by direct pasting of patch information but by sampling pixel information in overlapping patches to achieve a weighted average to be used as the new channel values in the output image.

3.4.1.6 Other Methods to Mention

Other important inpainting techniques include inpainting with Shift-Map [125], an image editing tool that utilizes a labelling method to compute how each pixel is changed after a transformation operation.

The work in [3], [56] merge the variational approach and exemplar based inpainting to solve the mapping of patches on the new image as a bounded variation problem.

The most recent approaches to inpainting generally include learning based approach which are mostly based on deep architecture [88], [117], [169], [99], [140], however these methods are kept out of the scope of this thesis due to the single-input nature of the system and adopt a hand-crafted generation scheme.

3.5 Implementation & Test Results

3.5.1 Object Removal

The still image to video converter mainly requires the objects to be removed from the foreground so as to allow the separate animation of the background. The literature analysis demonstrated that the most effective means to object removal is to utilize geo-texture preserving methods. Therefore, within this part of the experiments, the implementation of 4 methods are realized: Criminisi's Method [42], PatchMatch [14] based inpainting, inpainting using Coherency Sensitive Hashing [90] and inpainting using Image Melding [43].

Results for Criminisi

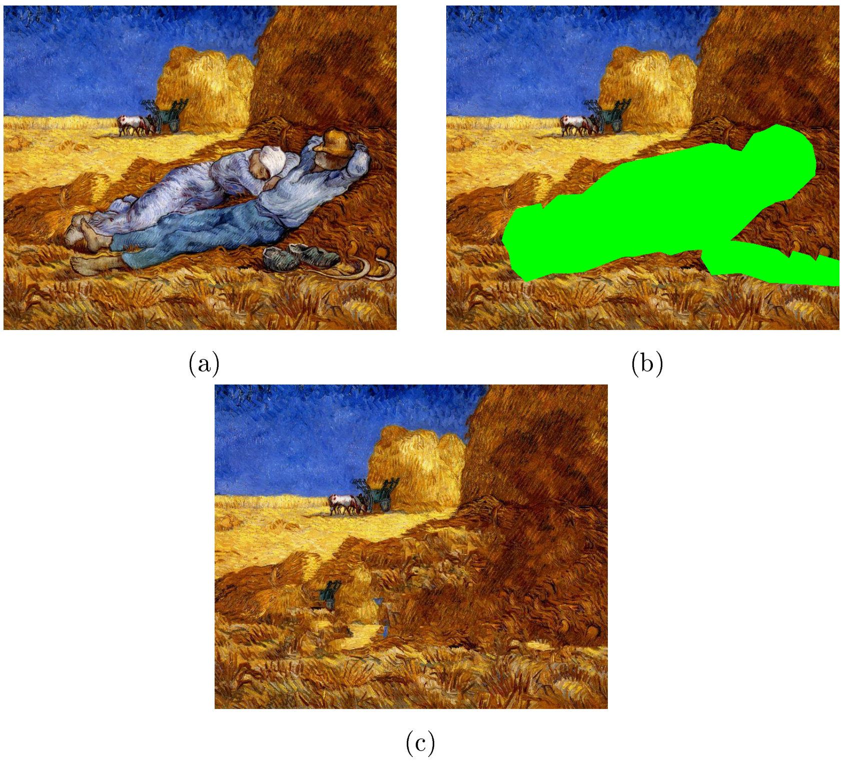
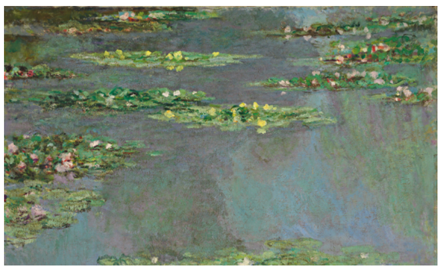


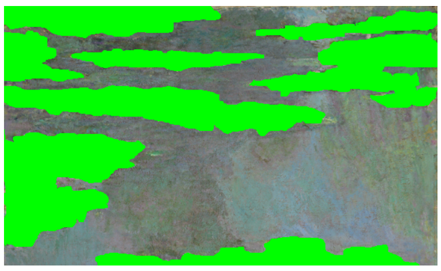
Figure 3.26: (a) Original Image. (b) Region Mask Λ. (c) Inpainted Image for 11x11 patch size on 607x736 pixel image using Criminisi [42] Method.

Since Criminisi is a straight-forward approach, it does not require too much parameter tuning. Indeed, as the filling order and sampling is well-defined, the only parameter that affects the inpainting is the patch size. As stated before, patch size is an essential parameter that determines how the energy of the initial image is transferred to the inpainted, as seen in Figure 3.29.

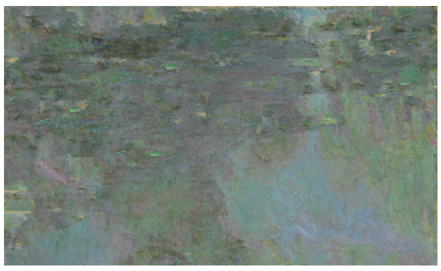
Criminisi involves naïve brute-force search for the best match query to paste the exemplars in the appropriate locations. Indeed, this also rapids up the inpainting procedure, compared to pixel based methods, as a whole since, from then on the requirement to make a search for each and every missing or undesired pixel in the Λ region is eliminated. Nevertheless, it is essential to remark that as the patches tend to grow, the similarity check for each patch requires an increased number of



(a)

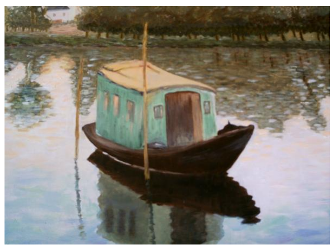


(b)

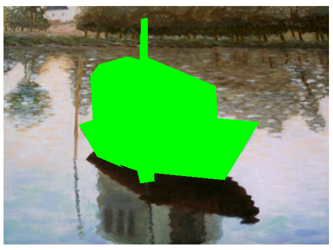


(c)

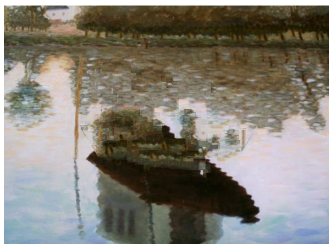
Figure 3.27: (a) Original Image. (b) Region Mask Λ. (c) Inpainted Image for 9x9 patch size on 347x580 pixel image using Criminisi [42] Method.



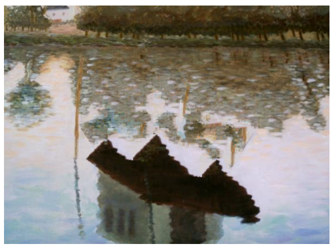
(a)



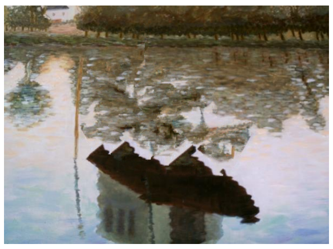
(b)



(c) 5x5 patch size



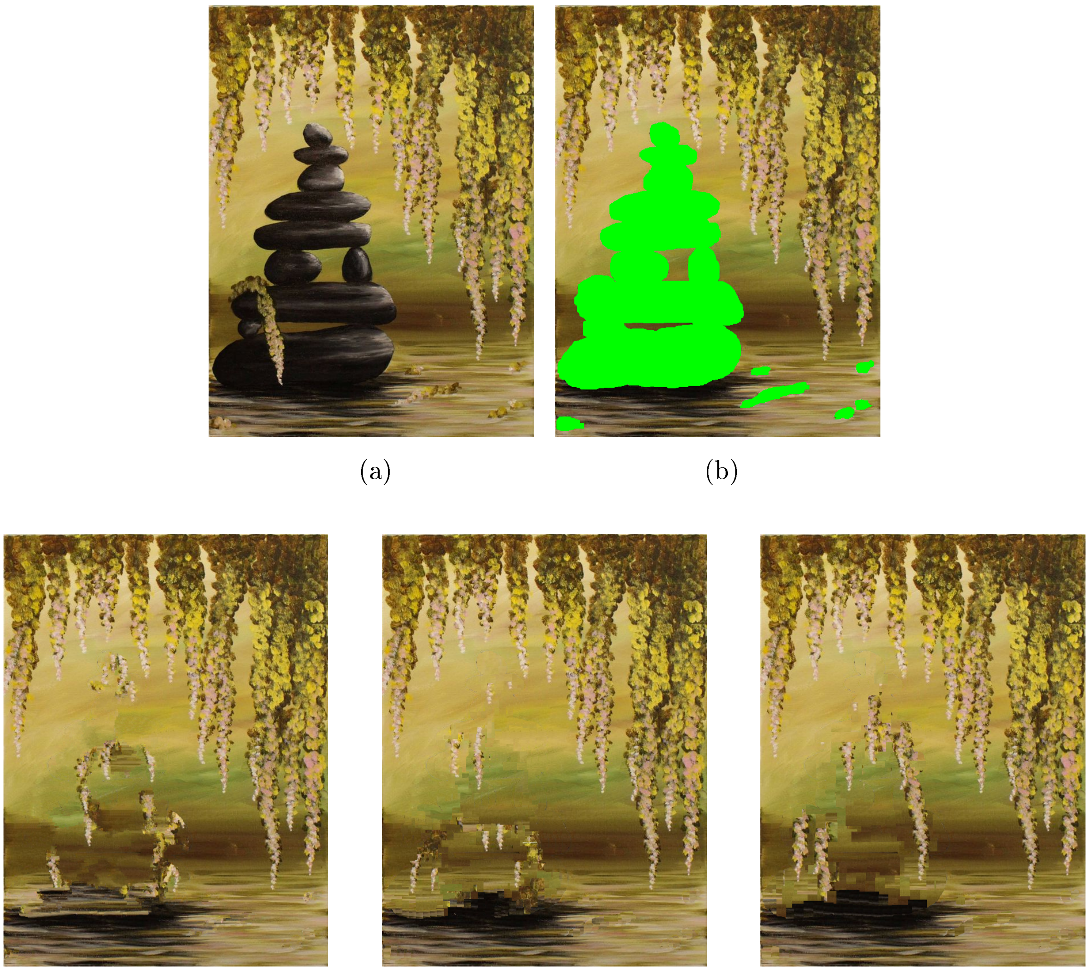
(d) 13x13 patch size



(e) 25x25

Figure 3.28: (a) Original Image. (b) Region Mask Λ. (c-e) Inpainted Image for different patch sizes on 439x600 pixel image using Criminisi [42] Method.

computations for the neighbouring pixels. In extreme cases, the patches begin to contain information more than necessary that will prevent the correct samples from the $I - \Lambda$ region to be selected. It may result in blocking artefacts on the image, as seen in Figure 3.30. Eventually, this becomes a trade-off between



(a) (b)

(c) 5x5 (d) 25x25 (e) 41x41

Figure 3.29: (a) Original Image. (b) Region Mask Λ. (c-e) Inpainted Image for different patch sizes on 751x564 pixel image using Criminisi [42] Method.

faster inpainting and more accurate filling of pixels and in certain cases the user is required to repeat the procedure with a variety of patch sizes to obtain visually pleasing results.

It is a method that prioritizes the strong high order features and inherently propagates filling order along their continuum. Therefore, such undesired results as illustrated in Figure 3.30 are likely to occur, especially when the region of interest is not properly chosen and contains traces from the object to be removed.

Results for Inpainting using PatchMatch

PatchMatch [14] is proposed as a fast match finder algorithm, and the paper

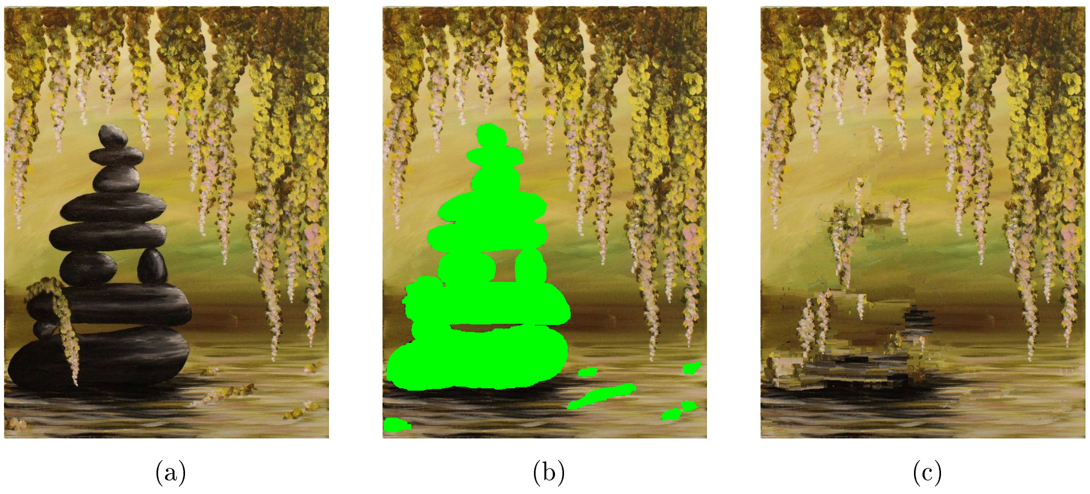


Figure 3.30: (a) Original Image. (b) Region Mask Λ with traces from the boundary of the object (c) Inpainted Image for 21x21 patch size on 751x564 pixel image using Criminisi [42] Method.

implements a method for inpainting based on guidance curves to initiate the NNF map. In this thesis work, the main approach is to minimize the user-interaction in each block, therefore a coarse-to-fine image completion method, which includes upscaling and down scaling the input and the generated texture using Gaussian pyramids, as in Figure 3.31, is used by starting with random seeds.

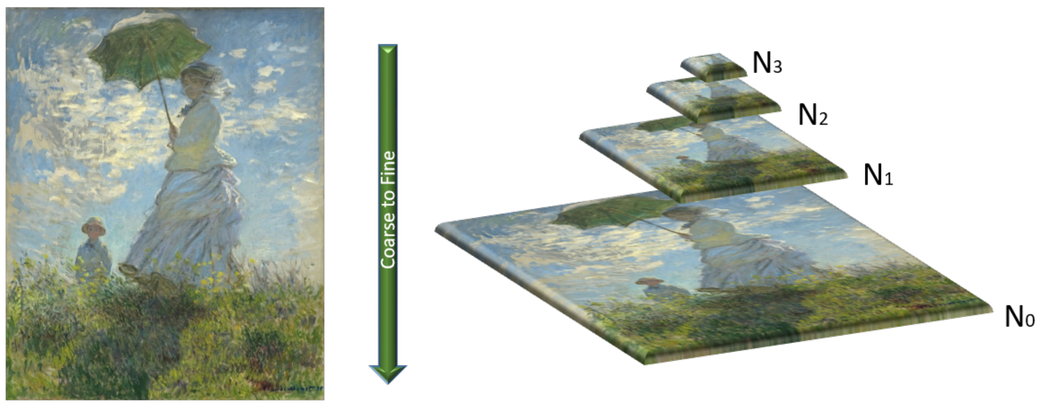


Figure 3.31: Image processing in a coarse to fine manner by using Gaussian Pyramids.

The image and the mask are downscaled to the coarsest level, and the boundary information is dilated to take the geometry into account by favouring the largest gradients crossing the boundary. ANNF randomly initialized by selecting from the dilated boundary of the image and the mask. For the coarsest level, in order to have more room for an accurate match, the ANNF is pre-iterated for convergence via increasing the number of random searches. At each level of the pyramid, in each EM iteration, for each pixel, the best matching patches are placed and averaged on the overlaps, using the normalized similarity on the patch using L_2 norm as a weight for each value. The best match searches are iterated until a maximum number is reached, then the images, the mask and ANNF is upscaled to a finer level until the finest is reached and processed.

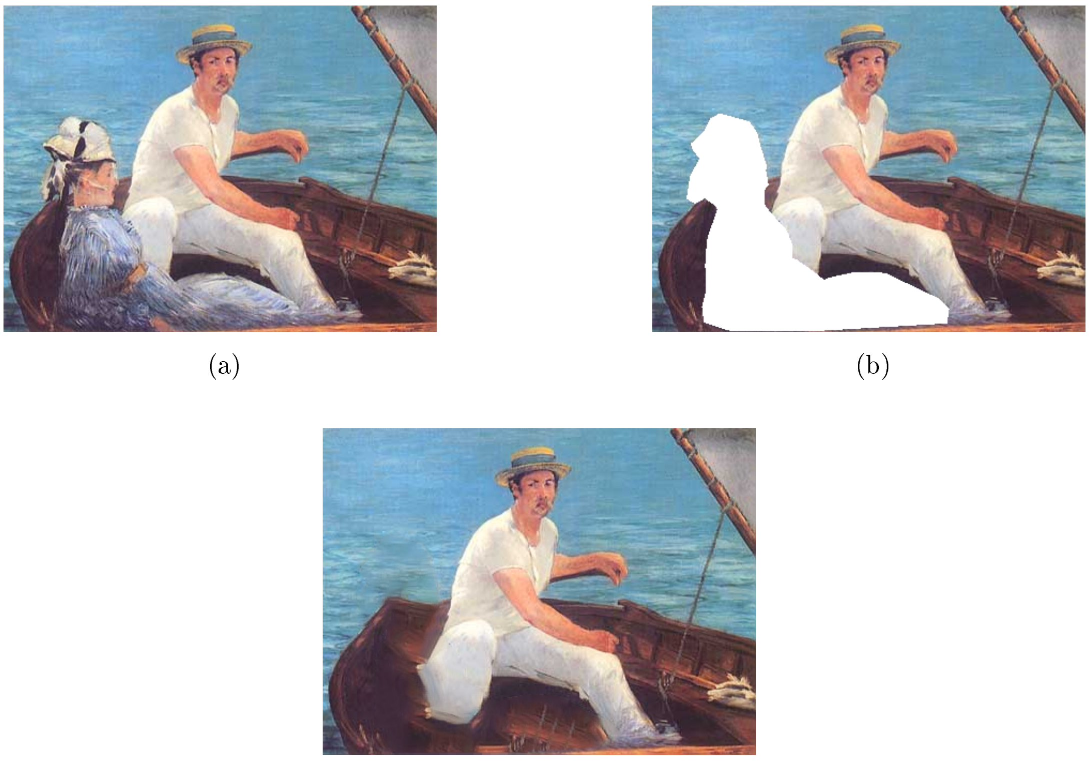
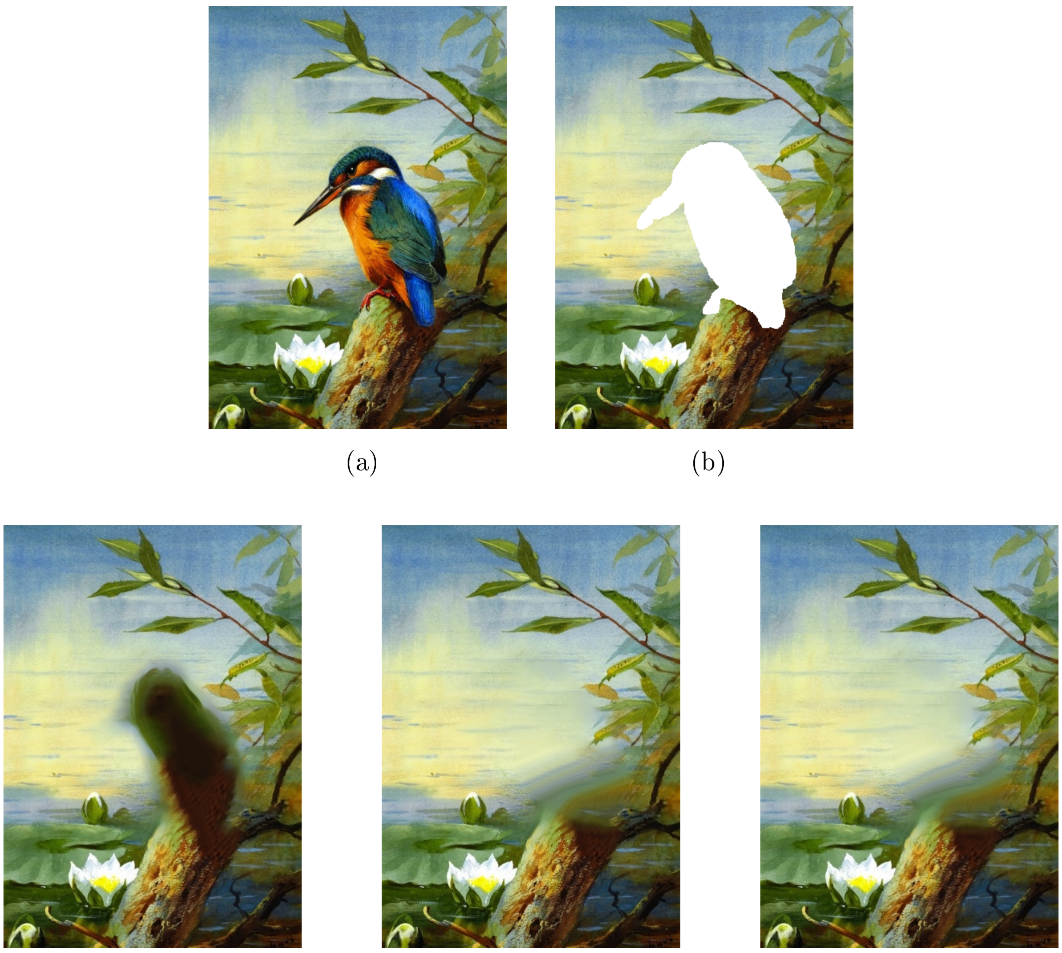


Figure 3.32: (a) Original Image. (b) Region Mask Λ. (c) Inpainted Image for 11x11 patch size on 305x405 pixel image using coarse-to-fine inpainting with PatchMatch [14] Method.

Indeed by implementing an inpainting method with PatchMatch equals to limiting the search domain to smaller area and accelerates the matching time interval, but since this is realized in multiple-scales a single run of inpainting lasts definitely longer than matching the patches of two images.



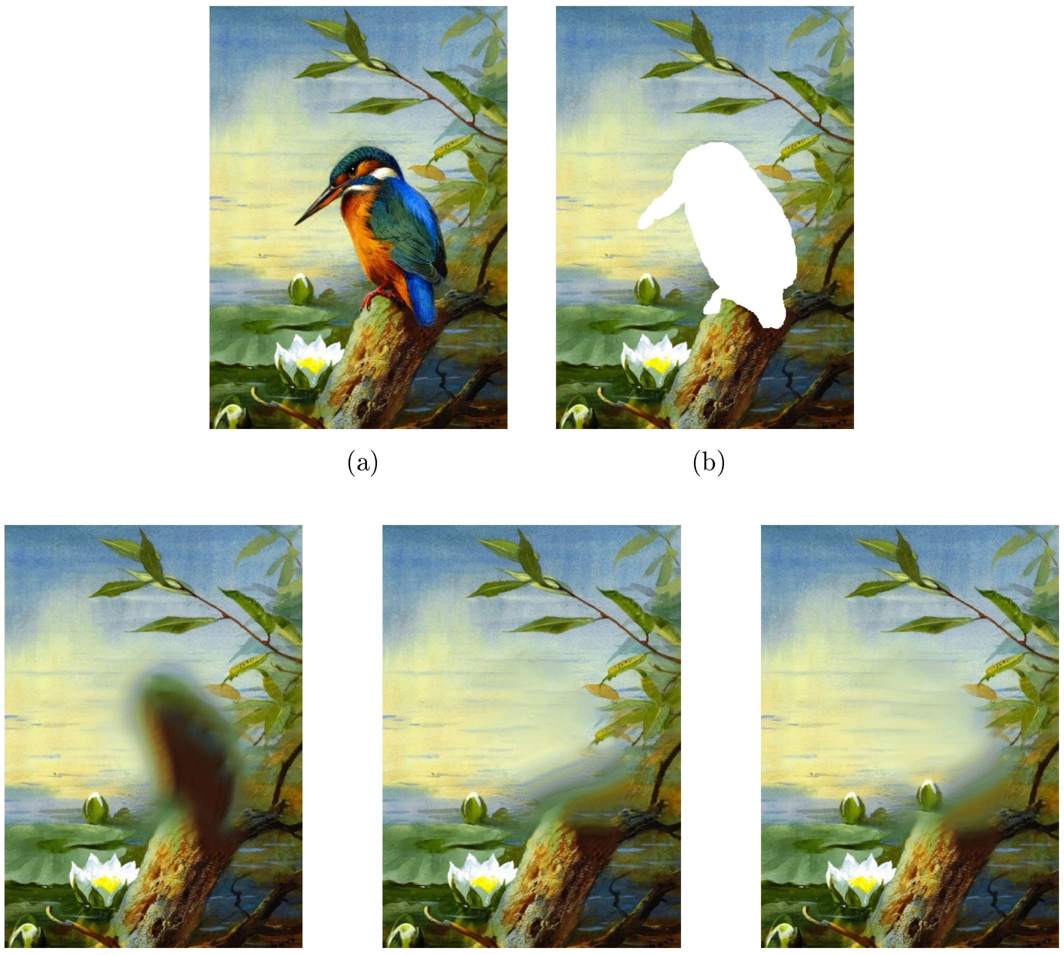
(a) (b)

(c) 3 level (d) 5 level (e) 8 level

Figure 3.33: (a) Original Image. (b) Region Mask Λ. (c-e) Inpainted Image for different levels of detail and 5x5 patch size on 523x371 pixel image using inpainting with PatchMatch [14] Method.

Figure 3.33 illustrates the fact that the level of pyramids is crucial when the method of completion is formulated as a coarse to fine method. With the same patch size on all the levels, the reduction of the number of pyramids results in re-utilization of the same limited source location to cover a large area, therefore the image is blurred and the completion looks unrealistic.

Figure 3.34 demonstrates the output under the change of patch size parameter once again for this algorithm. It is possible to grasp that with the use of smaller patch size than the available texture results the best matches to be trapped in


(a) (b)

(c) patchsize 5x5 (d) patchsize 11x11 (e) patchsize 15x15

Figure 3.34: (a) Original Image. (b) Region Mask Λ. (c-e) Inpainted Image for different patchsizes using an image pyramid of 8-levels on 523x371 pixel image using inpainting with PatchMatch [14] Method.

local minima, and as the pyramids are upscaled for a finer resolution the corresponding ANNF fields start to account for unfavourable matches. As the patch size increases, the staircase effect on the hole is lessened; however this results in replication of nearby objects in the covered region. Visually the completion is not undesirable, can even be considered to be functional especially in this converter system implementation; nevertheless it is odds on to imagine any waterlily standing on top of a tree on its own.

Inpainting Using Coherency Sensitive Hashing Similar to PM, CSH is again an ANNF building method that can be applied to different image processing applications. In [90] that the method was proposed, neither the mention of its utilization in inpainting applications , nor the approach for it is present. Therefore, in this work, an appropriate algorithm for its involvement in the field is provided. Based on the general idea of a coarse-to-fine approach which is used in many previous studies, [5], [51], [50], [115] is used for its adaptation on the inpainting problem domain. Also, the selection of a gradual processing is to provide a compatibility with the roughly explained inpainting approach in [14], and the hole filling method in [43] throughout the experiments.

Algorithm 3 CSH-Inpainting

```
1: function $I - CSH(S, T, M)$    ▷ $S$ - Target Image with Hole (0 valued), $B$ -Source Image , $M$-Region Mask X
2:     Let $O$ be the number of scales, and $d$ be the number of iterations
3:     Downscale $A, B$ & $M$ to the coarsest level
4:     $A(M) = w(\mu, \sigma)$
5:     for $i = 1$ to $O$ do
6:         for $j = 1$ to $d$ do
7:             $CSH(A, B, M)$
8:             Let $P[1...t]$ be the best patch matches to fill $A$
9:             $A[M] = P[1...t]$ by voting and normalize the effect
10:        end for
11:        Upscale $A, B, M$ to a finer level
12:    end for
13:    Return $A$
14: end function
```

Within this method, the best match is not determined by the L_2 norm, but refiltering on the less strict WH kernels by a applying a rejection method in a sliding manner. However, in order to paste the best matches on the target image seamlessly, similarity metric is necessary as well. Therefore, a voting method is employed, where each pixel in a patch has a say in the filling. That is, briefly, at each level of the pyramid, the ANNF is re-built for a maximum number of iterations or until a convergence criterion is reached so that the image to indicate that the image is no longer changed significantly within consecutive iterations.

For each pixel c, separately, the corresponding best match is checked for SSD with L_2 norm. The blending is realized as follows:

$$\psi_{output}(c) = \psi_{output}(c) + \omega * \psi_{input}(c\prime) \tag{3.47}$$

where

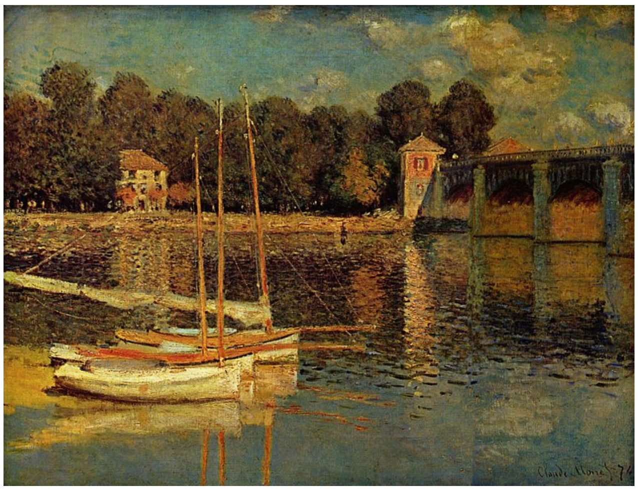
$$\omega = e^{-[SSD_{L_2}(\psi_{output}(c), \psi_{output}(c))^2)/\sigma^2} \tag{3.48}$$

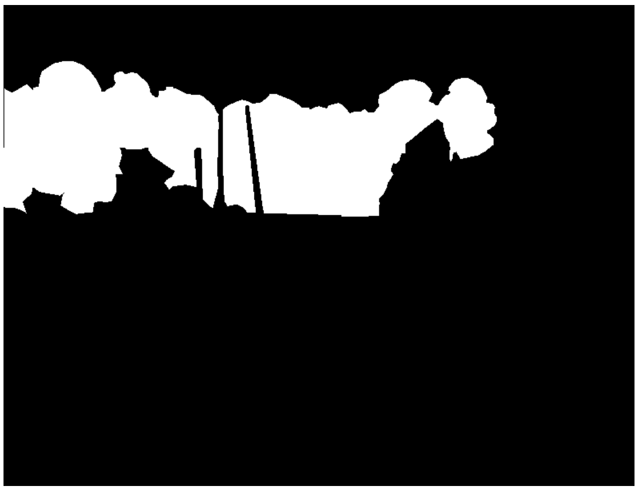
, σ is a regularization term taken to be 0.1.

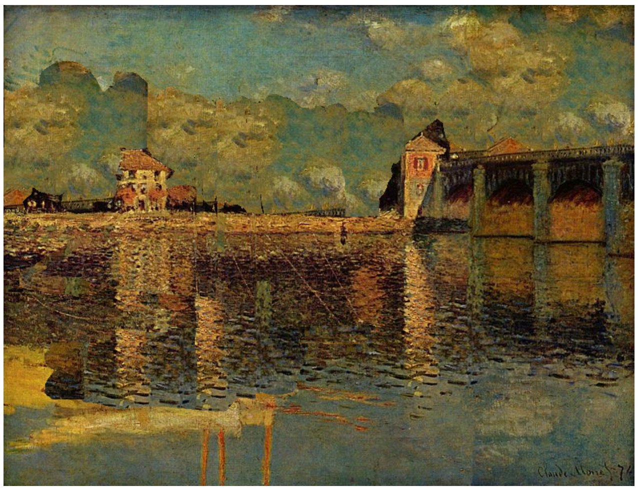
Patch size is yet the parameter that determines the fate of the inpainted output, as was the case in the previous implementations. Interestingly, in this method, the patch size selection is limited to the orders of 2, due to the ordering of WH Kernels therefore, in certain cases this limitation may result into suboptimal fillings as illustrated in Figure 3.35. Since patches tend to contain more information than required in certain cases the completed region features from irrelevant locations of the image. When it is fixed to a small value to retain in a local neighbourhood, the filling appears to be blurred.

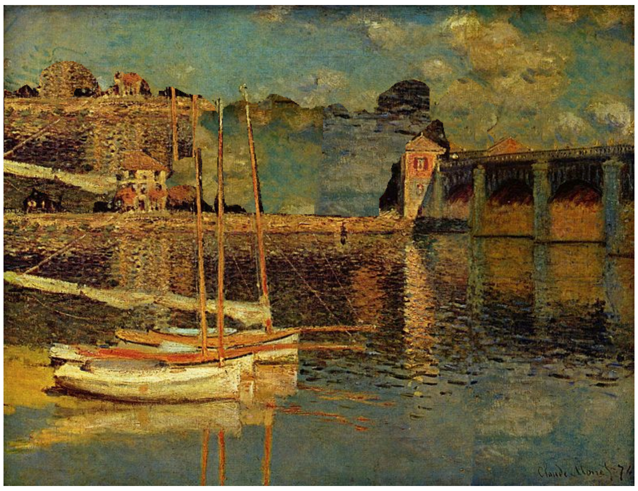
As the name suggest, CSH provides the propagation of coherency within the image in an increased way, especially when the number of k-nearest-neighbours are increased. Figure 3.36 illustrates this fact, the lines are more properly propagated for a smoother completion even the patch size is small in a less featured image. However, when combined with an incorrect patch size on a more diversely featured image, the completion may not be that pleasing. Figure 3.37 demonstrates the effect of k in K-NN with larger than requires patch size for larger patch size than necessary. It is possible to realise that when k may diverge the image from a truer completion if combined with not finely tuned parameters.

In the original paper [90] where CSH is proposed, the implementation was realized on the YCbCr colour domain. The motive behind is to represent the image in a more robust fashion to capture the change in luma and chroma colours separately. The hashing bins for the luma channel is 8 times more than the chroma channels. Regarding the methods shortcoming on the limited patch size to capture the energy, this ratio is changed for the experiments illustrated in Figure 3.38.


(a)


(b)


(c) patchsize 4x4


(d) 8x8

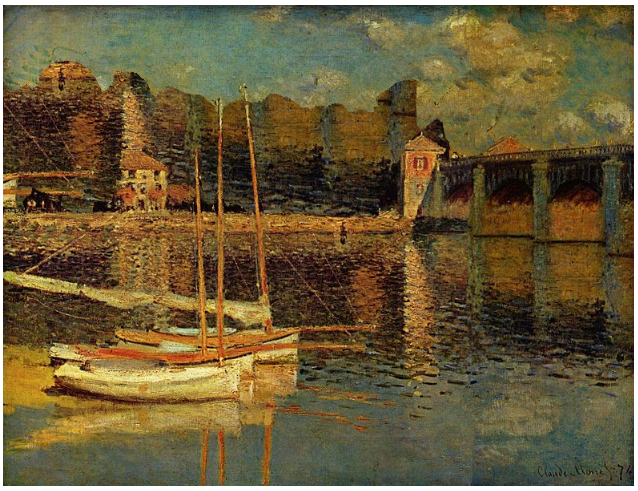
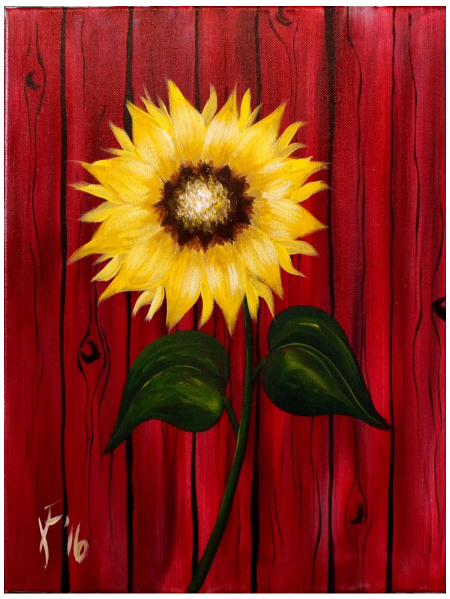
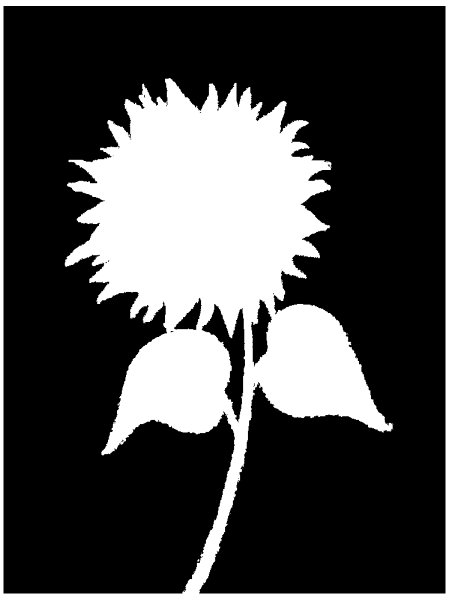
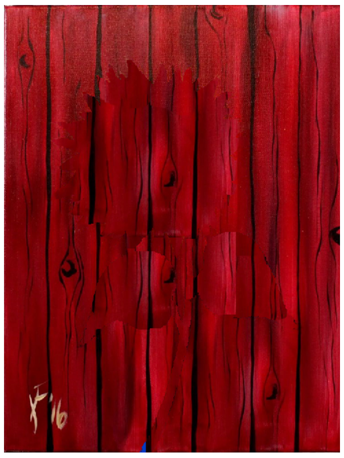
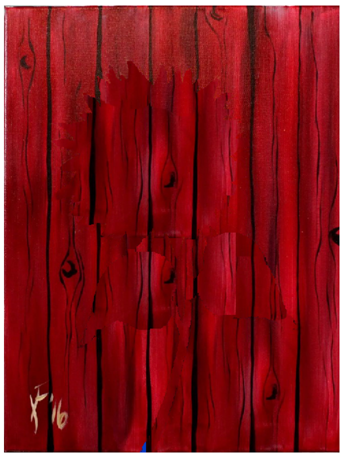

(e) 16x16

Figure 3.35: (a) Original Image. (b) Region Mask Λ. (c-d) Inpainted Image for 6 levels of pyramid with different number of k nearest neighbours on 679x768 pixel image using inpaing via CSH [43] Method.


(a)


(b)


(c) k=1


(d) k=2

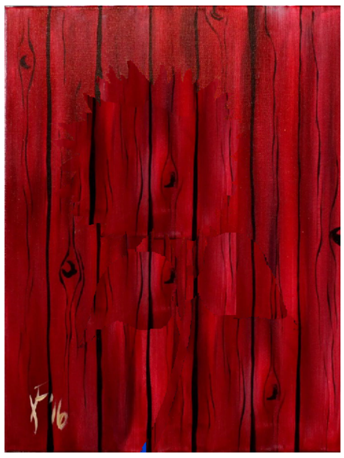
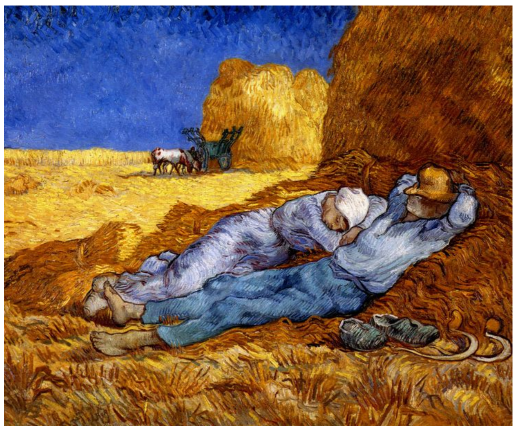
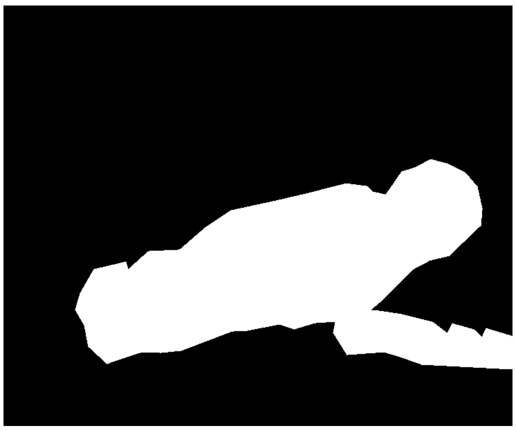
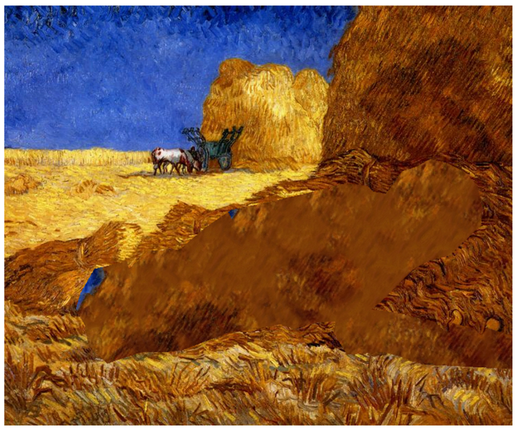
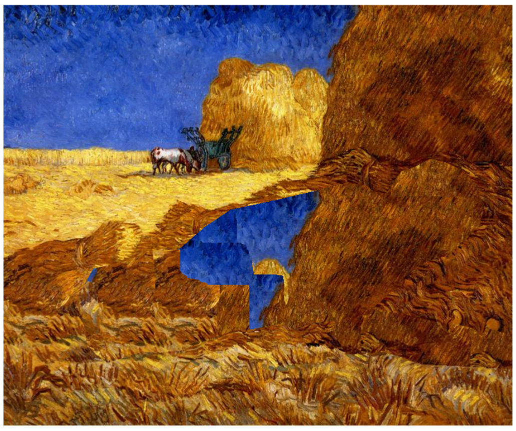

(e) k=3

Figure 3.36: (a) Original Image. (b) Region Mask Λ. (c-e) Inpainted Image for 6 levels of pyramid with different number of k nearest neighbours on 978x736 pixel image using inpaing via CSH [43] Method.


(a)


(b)


(c) k=1


(d) k=2

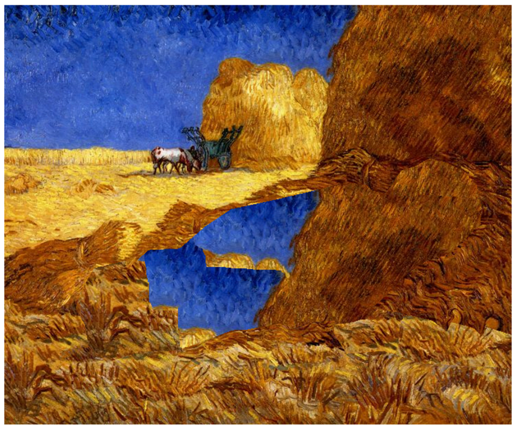
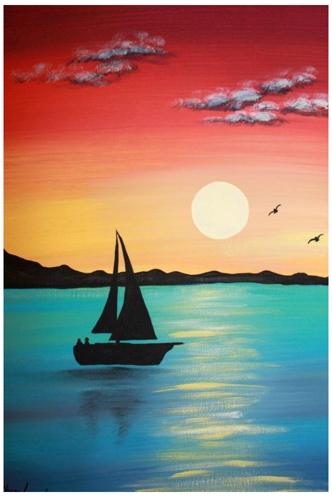
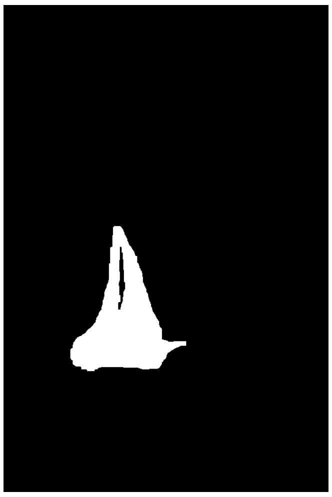
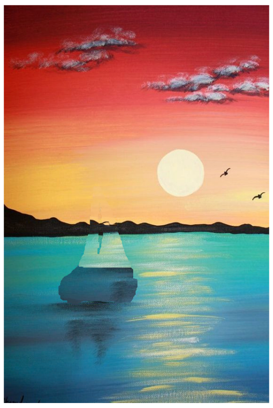
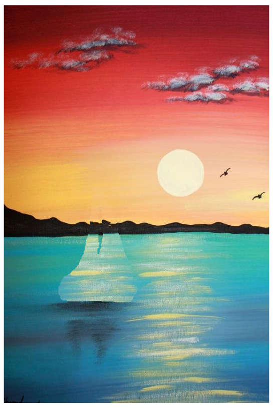

(e) k=3

Figure 3.37: (a) Original Image. (b) Region Mask Λ. (c-e) Inpainted Image for 6 levels of pyramid with different number of k nearest neighbours on 607x736 pixel image using inpaing via CSH [90] Method.


(a)


(b)


(c) Y/CbCr HashBin=8


(d) Y/CbCr HashBin=4

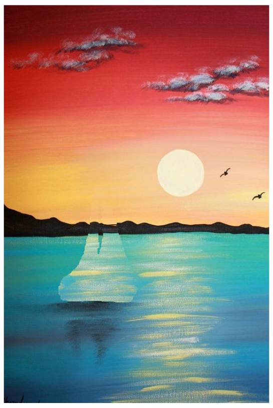

(e) Y/CbCr HashBin=1

Figure 3.38: (a) Original Image. (b) Region Mask Λ. (c-d) Inpainted Image for 6 levels of pyramid with different ratio for luma/chroma hashbins in YCbCR colourspace on 846x564 pixel image using inpaing via CSH [90] Method.

Overally, I-CSH implemented here is a method that requires very fine parameter tuning for an effective performance on image completion.

Inpainting with Image Melding

It is another image editing tool that that will be applied to image inpainting. Dabari et. al. has proposed the usage of discrete screened Poisson equation [163] for pixel voting as explained in Section 3.4.1.5. It is again implemented in a coarse to fine fashion for a better comparison with inpainting using PatchMatch [14] and inpainting using CSH [90]. For the sake of clarity, the effect on number pyramidal levels on image quality is illustrated in Figure 3.39.

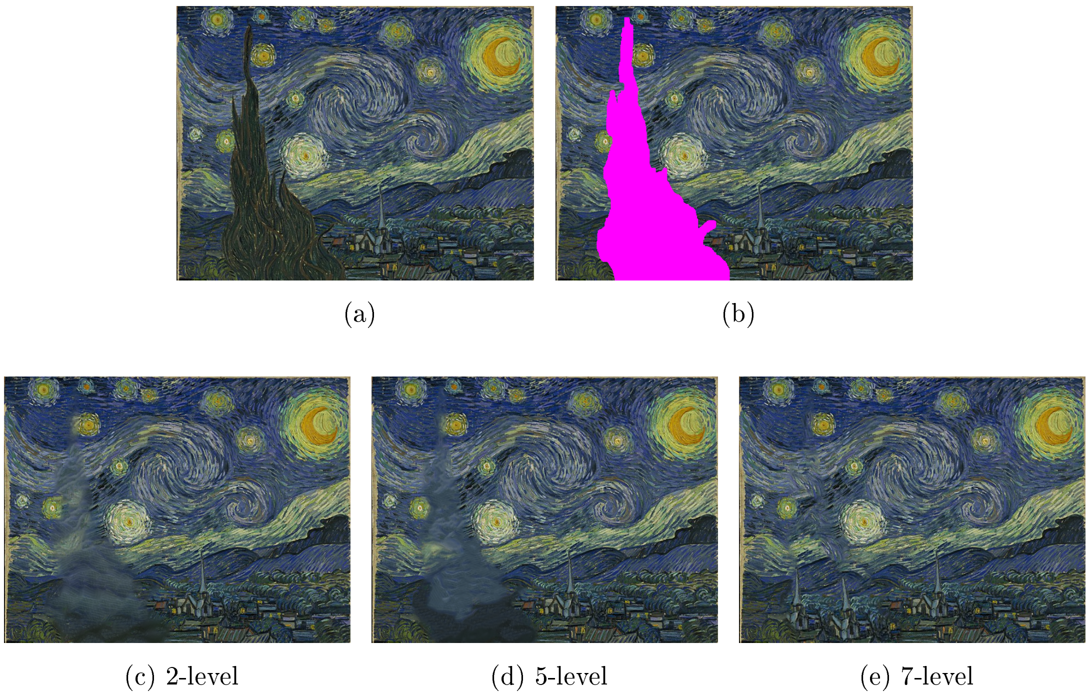
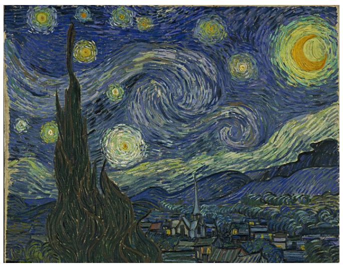


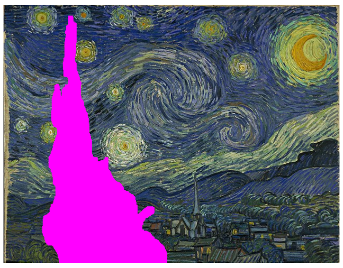
Figure 3.39: (a) Original Image. (b) Region Mask Λ. (c-e) Inpainted Image for levels of pyramids with fixed patch size (10x10) and levelwise EM iterations 751x564 pixel image using inpainting with Image Melding [43] Method.

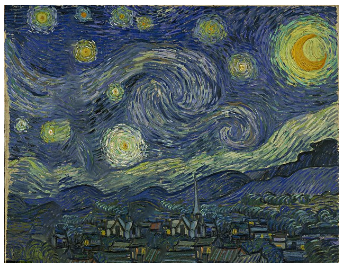
As the method includes patch comparisons with a range of scaling, rotation and refection based transformations, the test cases are built upon them. Also, the inclusion of gain and bias terms are also manipulated to overcome the variation in the patterns due to brush strokes and colourings in the paintings. .

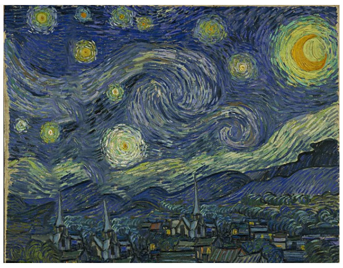
The second result in Figure 3.41 illustrates the removal of two distinct objects from the same image sequentially. Compared with the inpainting output of the Criminisi Method in Figure 3.30, it is to be deduced that the erroneous selection of ROI of the object (to an extent) is not crucial for an Image Melding operation, as the filling order which is omni-directional and iterative, it can omit the existence of the contour traces in the scene, unlike its isophate following, single query&paste driven rival [42].

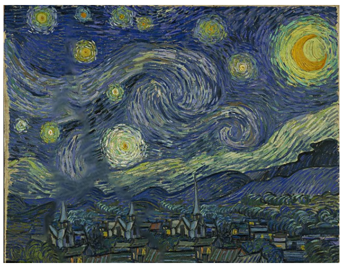
The result demonstrates that I-IM is quite effective in filling the image holes in a visually plausible way however; its operation time is inevitably longer than those of the others due to more detailed computations on each patch.

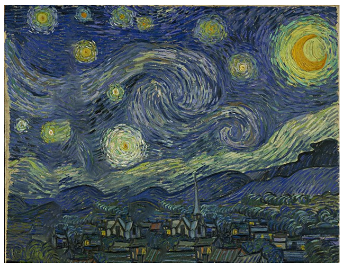

(a)

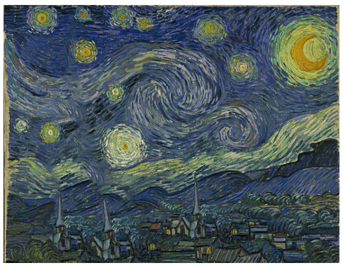

(b)


(c) no scale


(d) [0.9-1.2] scale range


(e) [0.5-2] scale range


(f) no rotation


(g) $[-\pi/4, +\pi/4]$ range

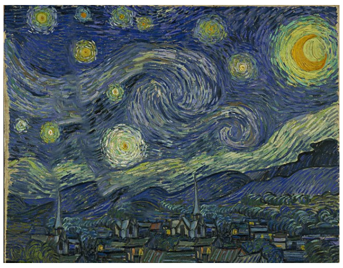
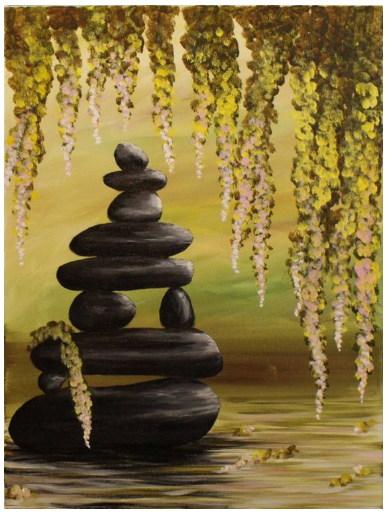

(h) $[-\pi/2, +\pi/2]$ range

Figure 3.40: (a) Original Image. (b) Region Mask Λ_I. (c-e) Inpainted Image for Λ_I with difference allowances on the patch scaling. (f-h) Inpainted Image for Λ_I with difference allowances on the patch rotation. Operation is completed with fixed patch size (10x10) and fixed number of EM iterations (30 at coarsest) on each level on 487x634 pixel image using inpainting with Image Melding [43].

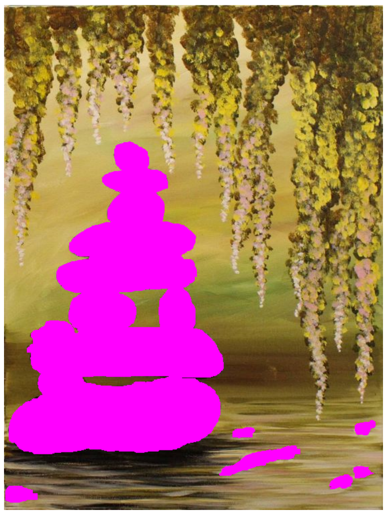
3.5.1.1 Comparative Results for Object Removal

These experiments provide the a visual comparison of how the methods behave on the same object removal cases. The comparison will be based on their visual quality, timing and memory requirements and functionality in the this image-to-video converter system.

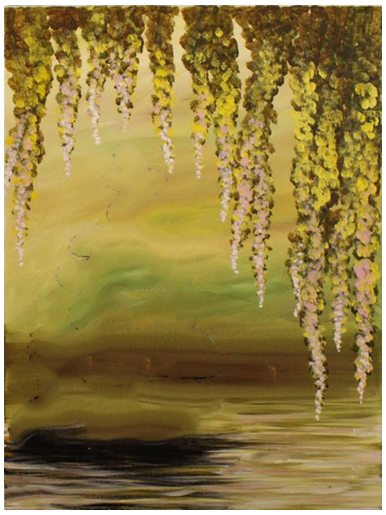
Figure 3.42 illustrates a toy example where the image scene contains a similar pattern in the entire $I - \Lambda$ domain. Therefore, the visual performance of all



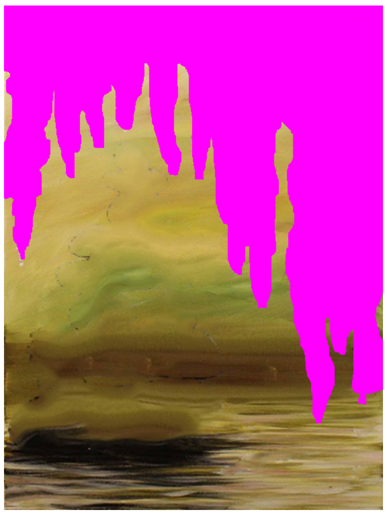
(a)



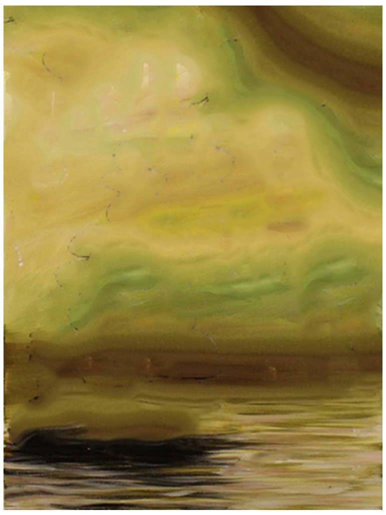
(b)



(c)

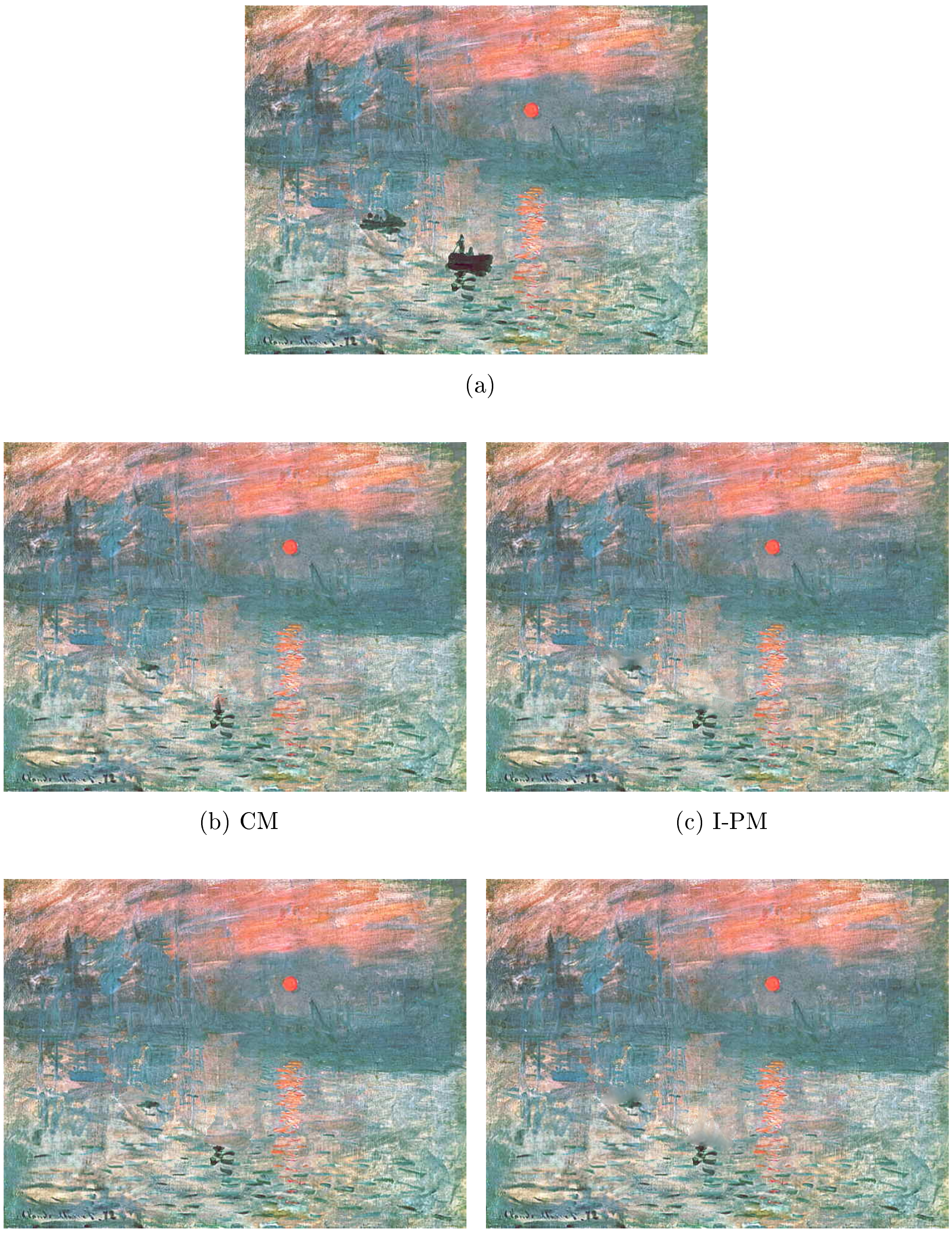


(d)



(e)

Figure 3.41: (a) Original Image. (b) Region Mask Λ_I. (c) Inpainted Image for Λ_I. (b) Region Mask Λ_{II}. (c) Inpainted Image for Λ_{II} for patch size 10x10 with image pyramid of 7-levels on 751x564 pixel image using inpainting with Image Melding [14] Method.



(a)

(b) CM

(c) I-PM

(d) I-CSH

(e) I-IM

Figure 3.42: (a) Original Image. (b-e) Object Removal Results for different inpainting techniques on 487x634 pixel image.

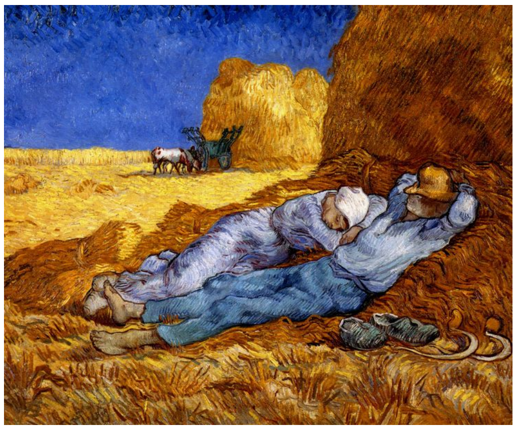
four of the algorithms is satisfying. The timing-wise analysis points out that inpainting with I-CSH in 3.42c is the fastest choice for such an image.

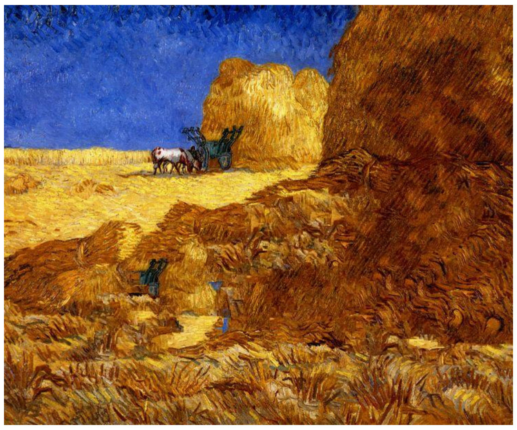
Moving onto another image, Figure 3.43 exemplifies a more complex case. The image has this time, more significant texture and colour information different parts of the image. The CM in 3.43a performed, nicely without blurring, but with minimal protrusions to the irrelevant parts of the image due isophate continuum. In this application the output image can still be used if the object removed will be kept immotile or move slightly. The visual quality of I-IM in 3.43d outperforms the others, where no blurring is present, and it is followed by I-PM in which a slight blur is observed. I-CSH in 3.43c, though not blurred at all, has an undesired image part in the central region. Though this image is supervised, as explained in Section 3.4.1.4, with boundary information, I-CSH when used with increased number of k-neighbourhoods, it becomes stronger on the regional continuum hence the region corresponding to the sky popped in the center of Λ region. A highly blurred version using the hay patches is possible, with different sets of parameters, in Figure 3.37c; however the latter result is more acceptable to be used in the video generation.

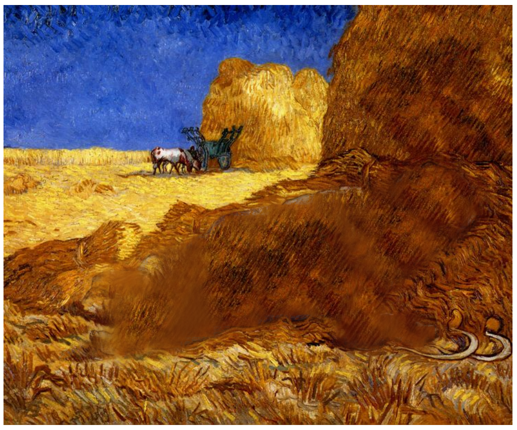
The example in Figure 3.45 is a case where the removal of an object from the junction of two regions is the issue. The CM is successful in completing the edges of the shadow into a closed form as seen in 3.44b. The coherency-insisting completion of I-CSH is useful in such a boundary of regions and provided a plausible completion. I-PM in 3.44c and I-IM in 3.44e are plausible, they are computationally more costly.

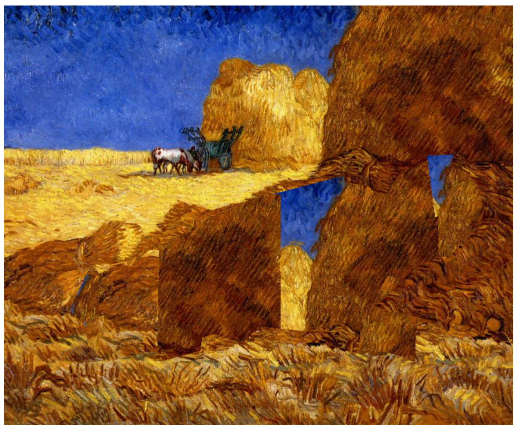
The final comparison is on o bi-step removal of objects where a group is initially removed, and then using the completed image another completion task is handled for the latter group. Concurrent removal is a more facile task for these exemplar based methods since as the number of candidates to use are more limited; however in sequential removal it is more probable to observe the formation of undesirable blobs in the Λ_I and Λ_{II} regions. Figure 3.45a is the initial image (C. Monet's Argentuil) where first, the vegetation(trees) and the boats will be deleted from the scene.

These illustrations indeed prove once again the ill-posedness of an inpainting problem. In 3.45b shows the replication of small image elements, the houses etc. within the inpainted region. This is a similar occurrence to the example


(a)


(b) CM


(c) PM


(d) CHS

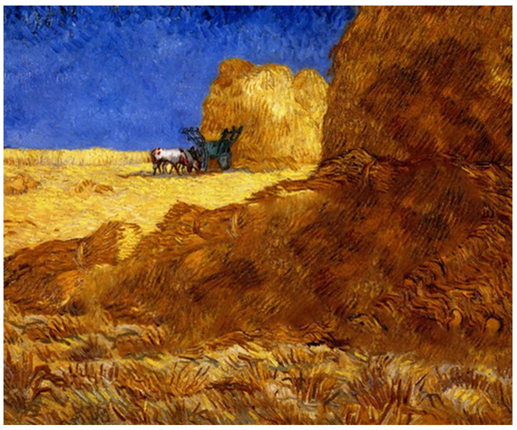
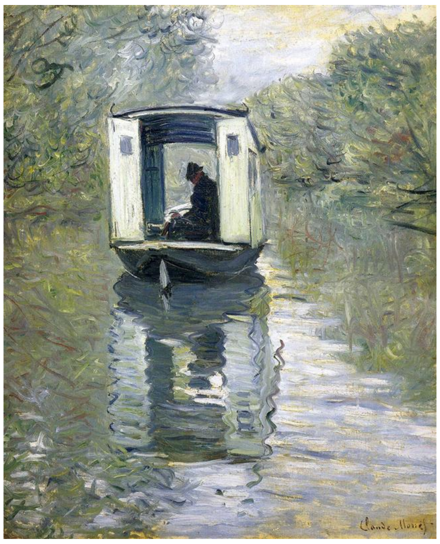
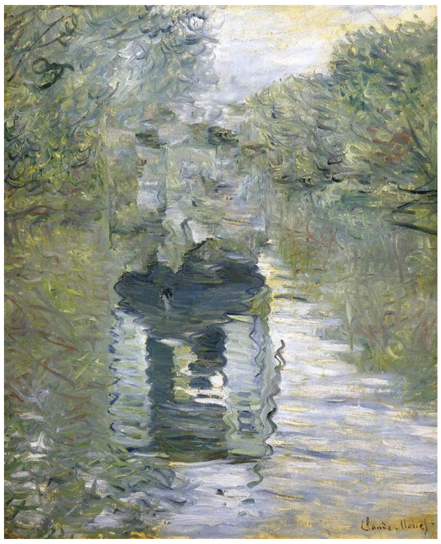
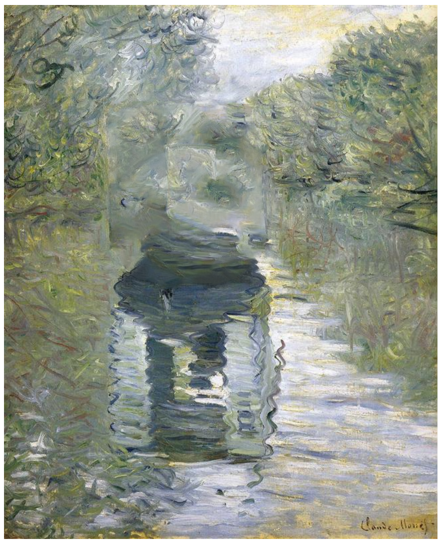

(e) IM

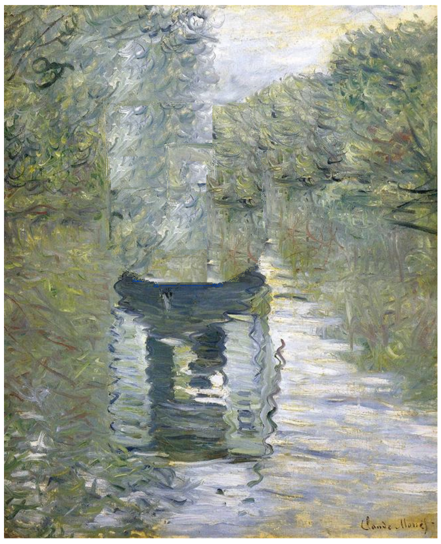
Figure 3.43: (a) Original Image. (b-e) Object Removal Results for different inpainting techniques on 607x736 pixel image.

where the effect of incomplete selection of the region of interest; nevertheless in this case the boundaries of other objects leads to the formation of partial


(a)


(b) CM


(c) PM


(d) CHS

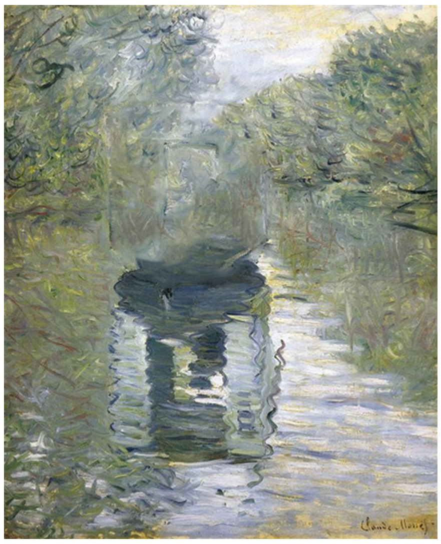
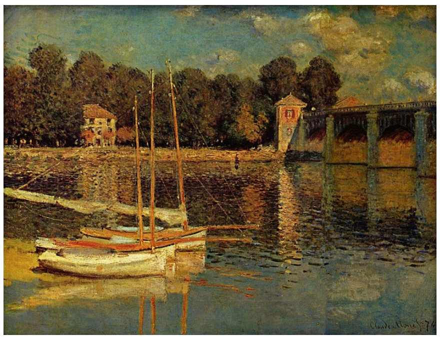
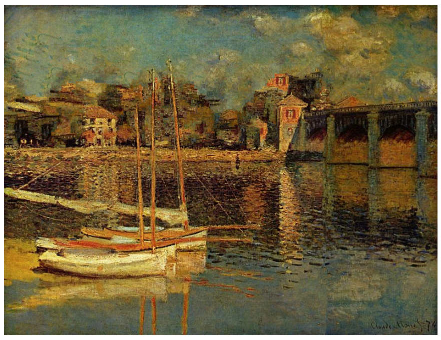

(e) IM

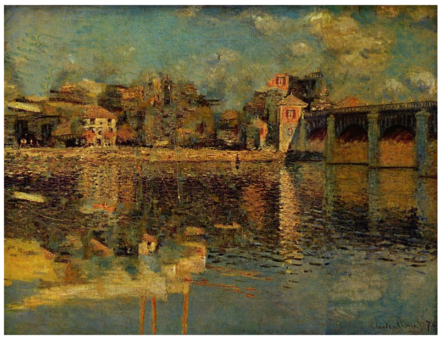
Figure 3.44: (a) Original Image. (b-e) Object Removal Results for different inpainting techniques on 800x649 pixel image.



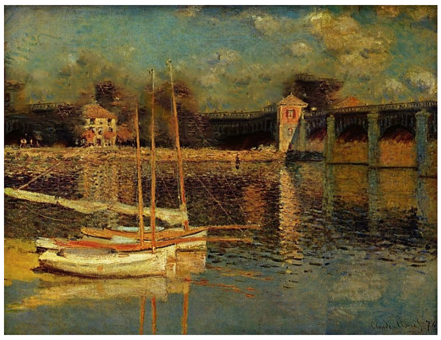
(a)



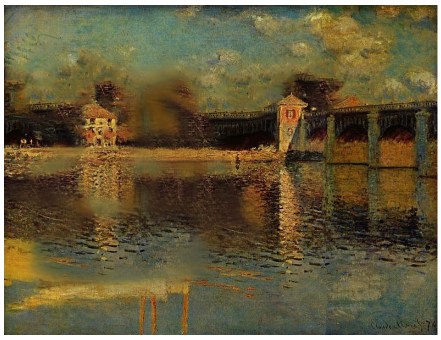
(b) CM_I



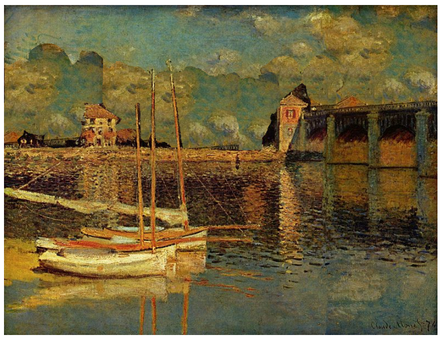
(c) CM_{II}



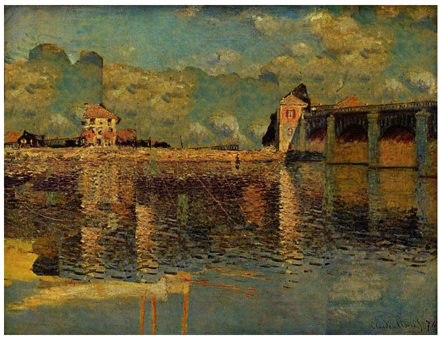
(d) $I - PM_I$



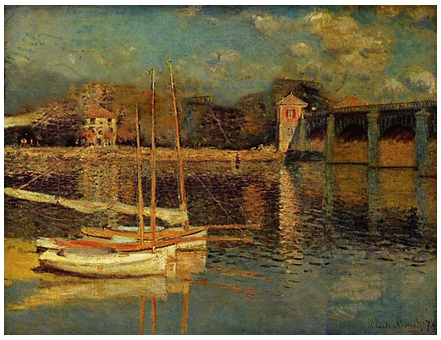
(e) $I - PM_{II}$



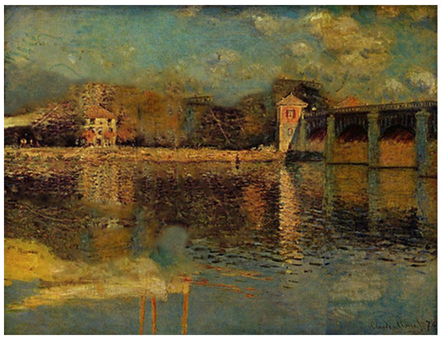
(f) $I - CSH_I$



(g) $I - CSH_{II}$



(h) $I - IM_I$

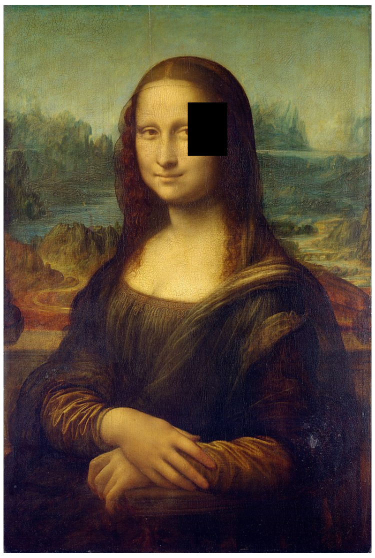


(i) $I - IM_{II}$

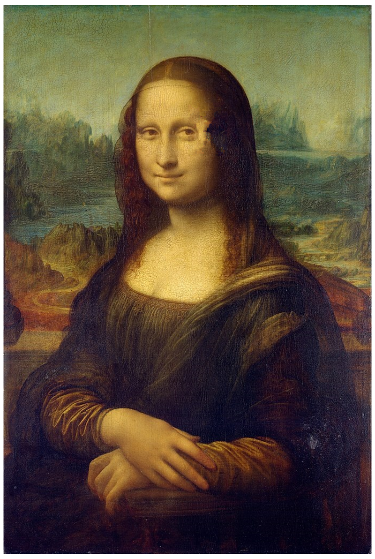
replicas of the image features. Eventually, these isthmuses are transferred to the latter image in 3.45c as well. For I-PM, the replication of undesired image features is not that evident. The sincerest concern is the sincerest concert is the selection of patch size. As the object to be removed in the latter step in 3.45g has an asymmetrical geometry, the ropes of the boat are replaced with more blur whilst the hull is completed as sharp as the initial image. This is due to the fact that I-PM only accounts for regional weighted averaging of features on the boundary regions. It is apparent that some of the blur is actually is not created but directly pasted from the former completion in 3.45g which result in a cumulative blur. For a desirable completion, fine parameter tuning is required for the two distinct region by I-CSH due to the initial assumption of not interfering with the building of the ANNF table. As the initial step requires the completion to be kept in a local area, the k-NN neighbourhoods are decreased by setting k=1 and patch size is lessened for the matches to be trapped in local minima. Therefore, the first completion is again on the blurrier side, where as the second needs more coherency propagation between different textures the aforementioned parameters are both increased for a plausible effect. Finally, I-IM eliminated all the shortcoming of the diversely featured image regions by employing the scale and rotation and discrete Poisson voting on the global image not on edges but on the entire image.

3.5.1.2 Quality Analysis on Reconstruction

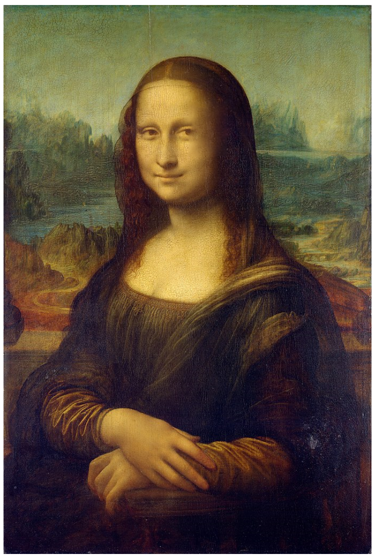
In this group of experiments, the strength of the inpainting methods in reconstructing the known parts of the image will be analysed. In contrast to what has been tried before, which is guessing the background, that is never known exactly, the methods are forced to rebuild what is available already. On a group of images rectangular blank patches are placed to obtain the initial image. This blank patch is the region mask Λ that has been long discussed. The resulting images are undergone quality measurements using widely used metrics such as, P(SNR) and SSIM. The image results for two examples are found in Figures 3.46, 3.47 and the numerical results in the following figures.



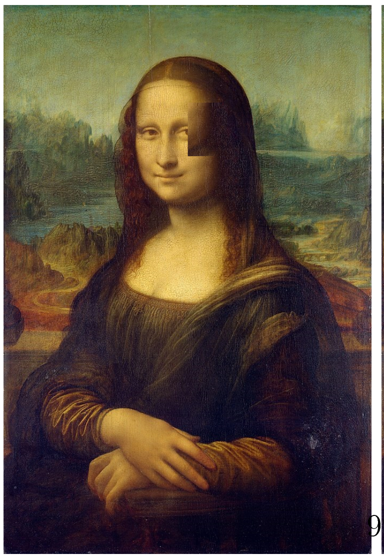
(a) Test Image. Mask Size: 74x100



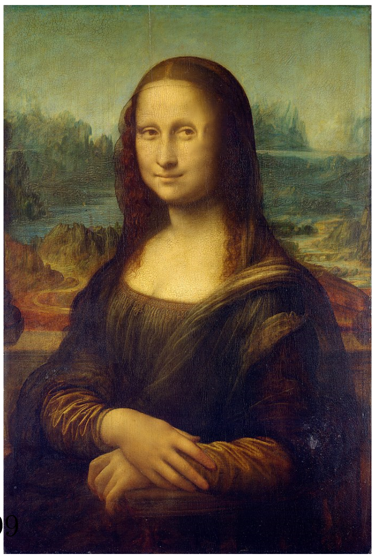
(b) CM



(c) I-PM

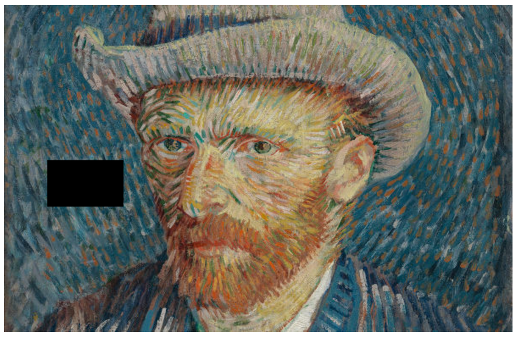


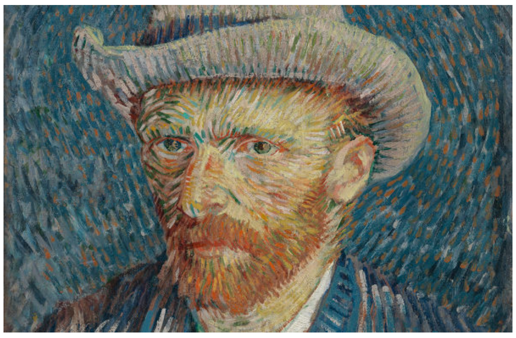
(d) I-CSH

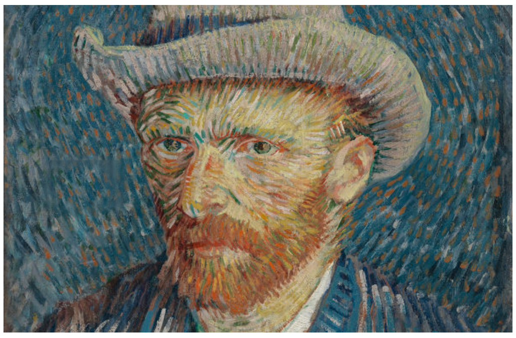


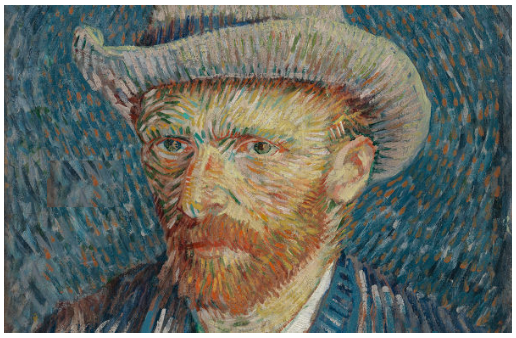
(e) I-IM

Figure 3.46: The strength of inpainting methods to reconstruct a known part of the image. (a)Test Image (b-d) Reconstruction results.


(a) Test Image: Mask Size: 90x55


(b) CM


(c) I-PM


(d) I-CSH

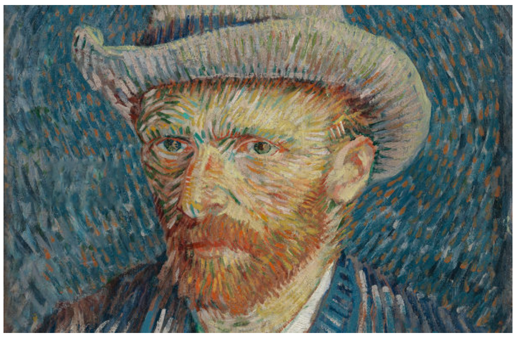

(e) I-IM

Figure 3.47: The strength of inpainting methods to reconstruct a known part of the image. (a) Test Image (b-d) Reconstruction results.

The test images have a diversity in the availability for construction where in Figure 3.46 it is harder to construct Λ due to the its location, which is residing near a complex geometrical surface, whereas the high order features in Figure 3.47 make an incorrect completion highly indetectable due to the increased texturedness of the region. Indeed, that is why the qualitative tests are utilized. The metrics are capable of measuring what the human visual system ignores. The first metric is the Peak Signal-to-Noise Ratio, or PSNR.

$$PSNR = 10log_{10}(peakval^2/MSE) \tag{3.49}$$

where MSE is the mean square error, and peakval for the digital images is 255. PSNR is expressed in decibels and SNR is

$$SNR = 10log_{10}(\frac{P_{signal}}{P_{noise}}), \tag{3.50}$$

where P(·) denotes power. Figure 3.48 illustrates the PSNR and SNR analysis of the 4 methods to test the reconstruction strengths on RGB colour scale.

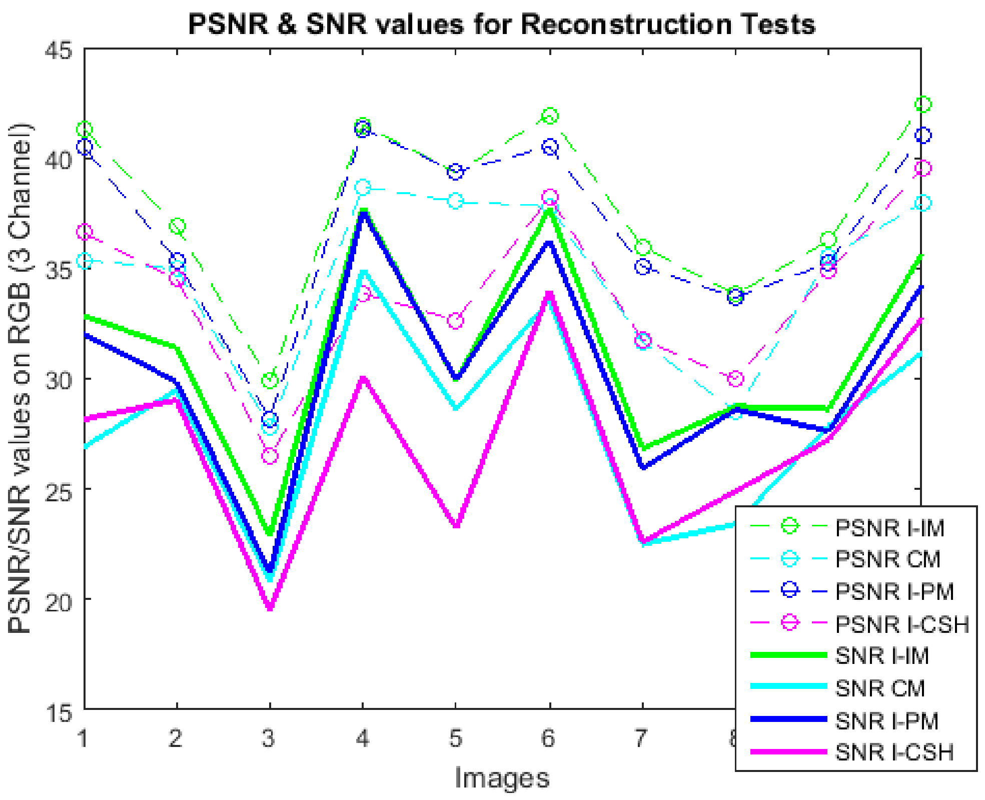


Figure 3.48: PSNR and SNR analysis on the reconstructed images in RGB colourspace.

The reconstruction by $I-IM$ maximizes the PSNR in most test cases, followed closely by $I-PM$, which means that the image is reconstructed in higher quality with these methods compared to the other ones.

In literature, indeed it is common to apply the PSNR measurement in colourspaces other than RGB, on the ones which seperate the chroma and luma information from each other based on the fact that they are more compatible with the human visual system. [42]. Therefore, the tests are repeated on the YCbCr colour space using only the luma channel, Y. Both of the results demonstrated using $I-IM$ and $I-PM$ promises higher quality results.

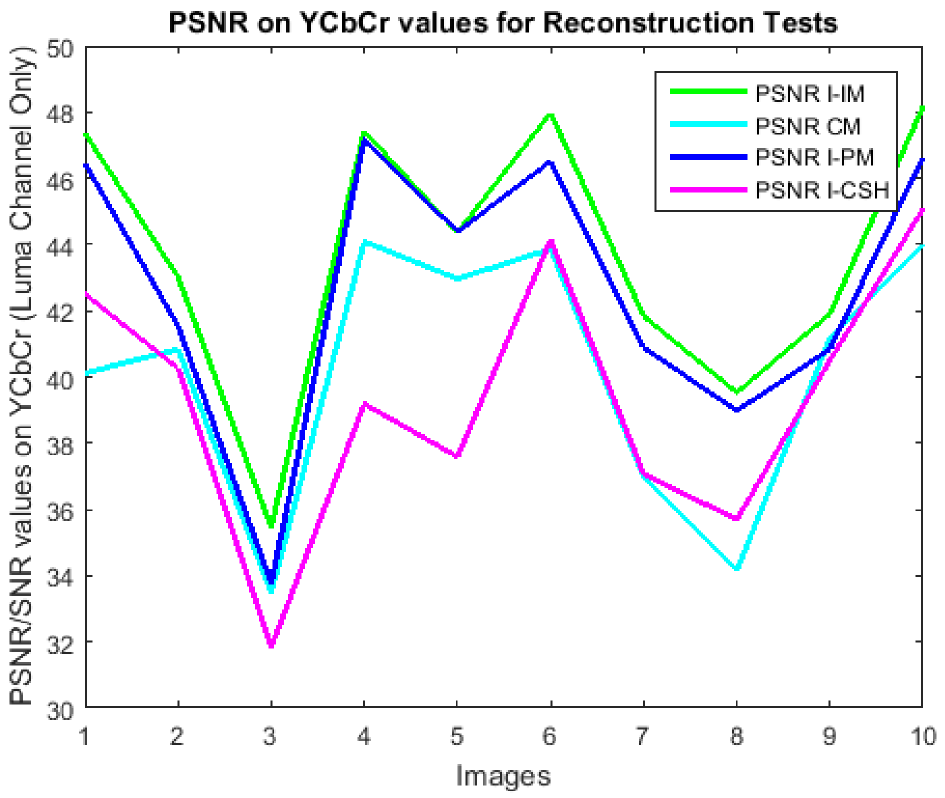


Figure 3.49: PSNR analysis on the reconstructed images on the luma channel of YCbCr colour space.

Another image quality metric that is proposed to be compatible with human visual system in perceiving information is Structural Similarity Metric, also known as SSIM. This metric is made use of for building dictionarities in the inpainting algorithm proposed in [114]. It is calculated as

$$SSIM(x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma} \tag{3.51}$$

where

$$l(x, y) = \frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \quad c(x, y) = \frac{2\sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3} \tag{3.52}$$

corresponding to the luminance, contrast and structural respectively, with $\mu_\star$ and $\sigma_\star$ representing the local mean and standard deviations. In this metric, the choice of α, β and γ are arbitrary but within these tests they are taken to be 1. Figure 3.50 demonstrates the quality test results using SSIM. Similar to those with the other metrics, $I - IM$ outperformed all the other methods in quality and $I - CHS$ gave the least quality images as expected from the image results. The mean results for each of the tested metric is illustrated in Table 3.1 .

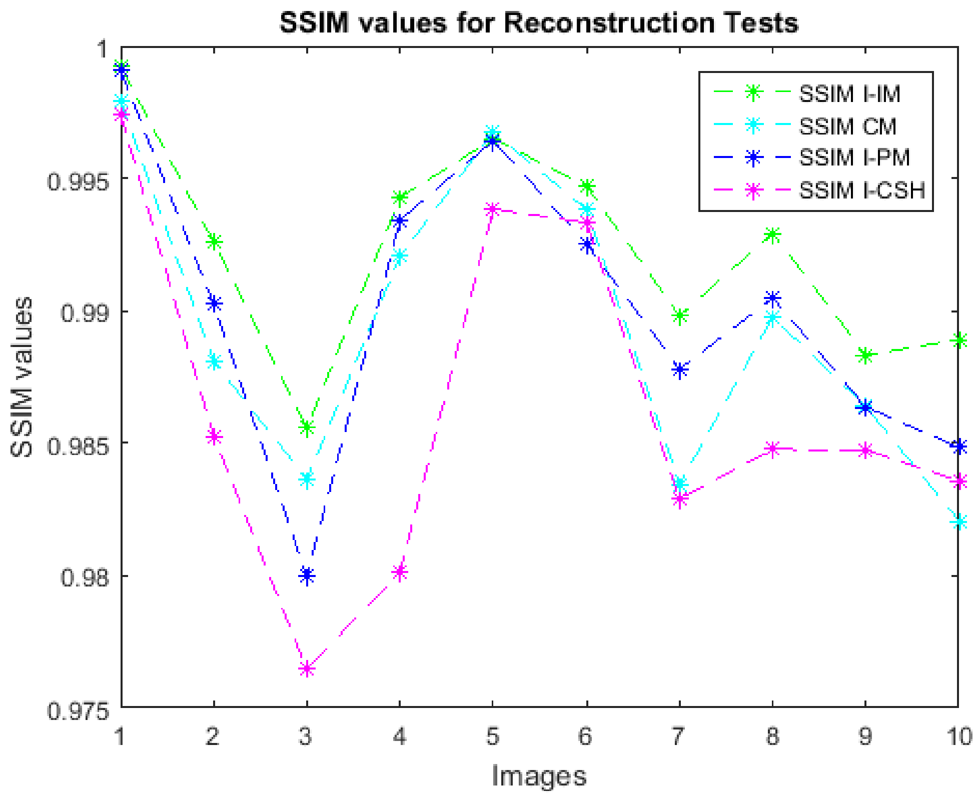


Figure 3.50: SSIM Image Quality analysis on the reconstructed images.

$Method$	$PSNR_{RGB}$	SNR_{RGB}	$PSNR_{Y}$	$SSIM$
CM	34.625	27.936	40.166	0.98939
I-PM	37.021	30.332	42.719	0.99012
I-CSH	33.845	27.156	39.387	0.98624
I-IM	37.928	31.239	43.704	0.99228

Table3.1: Mean results for each quality metric for the implemented algorithms.

3.5.1.3 Timing Analysis on Reconstruction

Another important criterion that determines the success of image inpainting is the operation duration of the methods. Timing is dependent on many parameters such as the size of the image, size of the missing region, the number of iterations, one-step fill amount, the scale(multi or single) of the method. Therefore, the mean filling durations of each algorithm on different test images are provided in Figure 3.51.

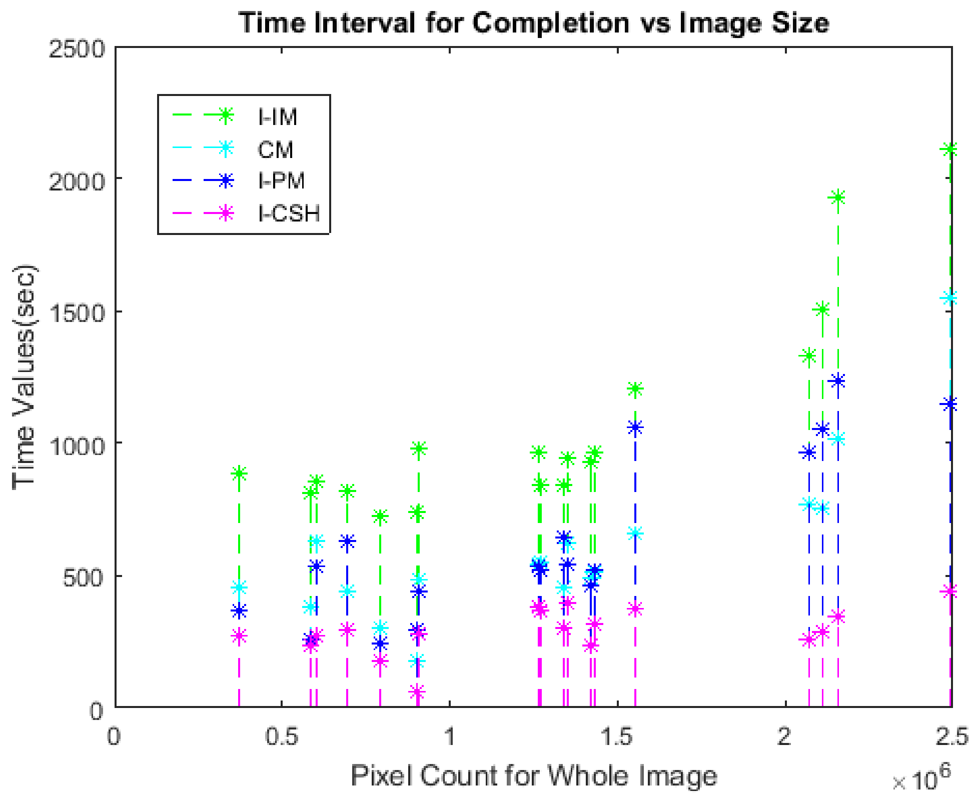


Figure 3.51: Mean results for a number of images with different pixel numbers.

As it is observed from the graphic that slowest performing result is $I-IM$, due to increased number of operations by using geometric transformations on the image patches and using additional computations during voting by also adding the gradient terms. The $I-PM$ without all these extra constraints operates much faster than $I-IM$ as trade of for the quality. Although, PM acts as faster computational method in NNF computation than brute force scan used CM, in a single scale, when the problem is domain is inpainting, the coarse-to-fine pyramidal method used in $I-PM$ causes the whole approach to be slower or sometimes equal. CM applies brute force search; however the fact that it is non-iterative, which means no blending patches is involved, and it is a single scale may result in faster completion of images. The reason behind the rapid operation of $I-CSH$ is that it highly depends on hash collisions and reduces the need for reiteration for random searches that are done in $I-PM$ and $I-IM$ and it also eliminates eliminates the computation of SSD during NNF tables.

3.5.1.4 Usage in the System

As discussed in the analysis, each algorithm has its own benefits and drawbacks with regard to some criteria. For the inpainting applications in this system, for the objects that are highly motile and reveal background in a significant amount, or the image captures a diversity of different textures and high order features locally, $I-IM$ is preferred to be used based on quality considerations. However, when the background is plain and the object movement is slight $I-CSH$ is a better option for faster operation. Therefore, in the flow of the the converter system, the user is allowed select the algorithm to be used for inpainting.

CHAPTER 4

MOTION MODELLING & SYNTHESIS

4.1 Introduction to Motion Modelling

The old-school virtual environments were surrounded by barely static objects such as buildings and mountains [84] but; hopefully, nowadays motion modelling is an increasingly popular area in the computer graphics and animation society, focusing on regenerating the movement of many figures of action in the real world, especially taking the human motion modelling as a core element. The analysis and implementation of human motion is a subject that has been the interest of researchers since 1960s [33] since it has its applications on a diversity of areas ranging from film industry to training activities for sports, from computer games to physical rehabilitation treatments, and recently the human-robot interaction has become a hot topic that make use of the human movements. Human motion is affected by several factors including the setting of the action, physical properties of the body, even cultural upbringing and the state of emotions at the course of the action [33]. It is interesting to note that the very same motion may be performed by the same individual in distinct ways each time. Hence, concerning the vast range of body types that exist and the complexity of the human actions, let alone modeling humans as figures of action, even recognition of motion is a tough challenge.

Along with human motion modelling, there exist billions of other motion models and the analysis of these active figures are also challenging, they possess similar difficulties such as the increased number of degrees of freedom and independent parts of the motile body. Animating these figures requires representational animation, which is a hardtask [159]. For instance, let's imagine the flight of a bird. In general, the motion consists of a translation with the flapping action of the wings; nevertheless when considered in detail to achieve more accurate visuals, one has to take more elements into account. Admittedly, the anatomical knowledge is indispensable for the utmost accuracy [123]. Although, the flapping action is done using the wings, the muscles of the wings are hinged to the body of the bird such that each flap results in synchronous contraction and relaxation of the chest as well. Moreover, the translational motion is not always on a straight trajectory, a bird may tend to have different postures on the take-off, during the flight and while landing. Some fluctuation in the air is observable, along with the changes in its flying pace due to obstacles, such as trees, hills, etc. The birds rotate their bodies around their trajectories such that they look inclined towards the direction that they are making a turn. Concisely, the brief action of flying is hard to model without paying attention to a bunch of morphological and behavioural criteria.

4.2 Specifications of the Block

As the penultimate block, the **Motion Modeling and Synthesis Block** is where the simulations of the objects are realised. The overall system lacks an object recognition capability to animate the layers automatically, due to complexity considerations therefore, the user is expected to specify the motion types to be applied to each layer. This block handles a bi-step procedure of reanimation, the former is responsible for sculpting the model and the latter for and creating a 5D motion armature to allow filtering the still image with the function generated by the armature. The limitations and the capabilities of the motion representations are provided in the upcoming subsections.

4.3 Motion Modelling Domain

4.3.1 Object Motion

In this application of animated video generation, the figures of action that obey complex rules of motion are omitted from modeling, since they require more interaction from the user for a smooth visual flow. The image domain to enter to the system is therefore restricted to objects that show a full body motion, with no separate parts involved and only having limited degrees of freedom. Motion modeling and synthesis part of the system, henceforth, is focused on the objects with harmonic patterns, clearly, the motion present is a repeated oscillatory movement of the objects. These models are easy to figure out and applied to the image layers such that the required continuity in the frame sequences is automatically achieved, with very minimal user inputs where necessary.

Harmonic motion is abundant in nature especially in passive bodies, such as clouds, trees and vegetation, bodies of water, boats, etc., namely the masses that does not demonstrate an active motion of their own, but move due to being exposed to the drag forces that are present in nature, namely wind, gravitation etc.

4.3.2 Weather Effects

Some weather conditions, such as rain, snow etc. can be visually applied onto the video frames with addition random of noise, simple filtering or some basic transformations of the image regions, without seeking a deep sense of photo realism.

4.3.3 Modeling Approach

In [148] Sun et. Al. assert that it is quite often that the natural motion models are observed to be harmonic oscillations. In early applications of computer graphics for especially in computer games, natural phenomena was preferred to be roughly modelled by arbitrary superposition sinusoid functions, the complete depiction of the whole motion requires a very fine tuning of the parameters or synthetic looking visuals are obtained [37].

Despite the name suggests, these reacting motion groups never occur in full periods, rather they avoid strict repetition. Every non-synthetic repeated pattern has an inherent degree of randomness, as a result of being part of a conjoined network of things, nature!, whose elements are, significantly or insignificantly, correlated with the remainder. As a result of this butterfly effect, it is predictable that any swaying tree branch under the howling wind will diverge from its course, even in the slightest manner, when an eagle flies over it, let alone the cases where the weather conditions are rapidly changing. Trivially, it is assumed that modeling natural flow requires the principal of arbitrariness.

Likewise, in signal processing, an analogous randomness is the case due to the presence of any type of noise. In [37], it is proposed that the irregularity of the natural phenomena can be imitated by the introduction of noise term in the motion model. It is further stated that, instead of the spatio-temporal domain, where the artificially produced noise has the aftermath of abnormal and unrealistic outcome, the optimal approach is to insert noise in to the frequency domain, such that the system model is molded to mimic the inherent frequency responses and intrinsic periodical behaviours of the real-life counterparts.

This methodology is referred as spectral filtering in literature, where distinct domain-specific spectral filters contribute to the generation of models for various applications. Spectral methods are the-state-of-the-art techniques in stochastic animation. The synthesis of the stochastic field to be used in motion model generation for natural phenomena is three-folded and is described as follows:

- Generate a complex random noise field, preferably Gaussian, in the frequency domain
- Pass the signal through a domain-specific spectral filter system
- Take inverse Fourier Transform to obtain the spatio-temporal representation

4.4 Motion Models

4.4.1 Trees & Flora

4.4.1.1 Related Works

Trees and branched plants cannot be quite easily modeled due to their inherent geometrically complex nature [149], that is composed of a main trunk, branches that divide into uncountable more and the leaves that are attached to them. The methods to render swaying motion on trees have three general approaches. Simulation-oriented methods exploit the area of fluid dynamics [101] to visualize the interaction between the drag-force and the object. [2] employs Navier-Strokes, [131] independently animate tree segments in the form of rigid sticks, [160] develops a streamlined-mechanics basis for modelling and [48] uses wind-projections for faster implementation. However, these methods are computationally costly and too detailed for our framework.

The second approach is stochastic modelling where a spectral-formulation of wind is used. First proposed by [141], and implemented in [37], [67], [174] for different domains, ranging from large forest modelling to 3D-animation applications. In [116] the wind turbulence effect is modeled by Perlin noise by incorporating a mass spring model. [145] preferred white noise to be synthesized in the frequency domain and computed a displacement map for the tree particles. [118] is a model that mimics Lissajous curve model in spatio-temporal domain with randomness inserted by three-dimensional $1/f^{\beta}$ noises. Finally, motion capture methods [101], [48], [167] collect data using video input or direct capturing to collect movement and position data to reconstruct tree geometry and motion. These techniques are not applicable to this system due to input restrictions to single image and the requirements of additional hardware.

4.4.1.2 Motion Model - Sway

The motion model used for animating trees and flora is a blend of Zhang et. al's algorithm [174], and the simplified version of the method in [148] that was implemented in [37] by Chuang et al. The reason behind the utilization of both is to generate a model that is simple to be implemented in fast-rendering framework, as well as to be capable of reflecting a realistic swaying motion. The method of Zhang will be used to generate the entire wind field in the scene so that the motion models can be build upon it. Therefore, it will be used water modeling as well.

Sculpting a tree motion is composed of three main steps in this implementation

- Modeling The Wind
- Modeling The Motion of The Branch Tip
- Modeling the Whole Body

For the sake of clarity, the methodology will be explained in a reverse order.

Modelling the trunk of a piece of flora can be considered analogous to modelling the branches on it [37], [148], and they can be regarded mechanical systems that have motion behaviours determined by their mass along with other properties like stiffness and damping. This physical system and its inherent motion can be described as follows:

Each trunk/branch on a trunk is pivoted to a non-oscillatory surface from which the trunk/branch grows and elongates until the tip of the trunk/branch. The swaying motion is roughly the motion of an inverse pendulum that is hinged to that surface. Unlike an inverse pendulum, the tip of a branch is far thinner and more flexible than the branch root, hence the motion is visually more evident along the stem from the root to towards the tip. In a simpler geometrical interpretation, it can be concluded that the parts near to the branch root draw infinitesimally small arcs, whilst the tip creates vaster arcs in a concentric fashion. In [148], the simplified version for this observation is modelled as

$$d_{isp}(\ell,t) = [\frac{1}{3}\ell^4 - \frac{4}{3}\ell^3 + 2\ell^2]d_{isp}^{BT}(t) \tag{4.1}$$

where $d_{isp}^{BT}(t)$ determines the movement of the branch tip in the spatio-temporal domain. Revisiting the assumption that the fundamental reason behind the motion trees begin the drag-force applied by the wind, a damped harmonic oscillator formulation is adopted. The steady state equation is given as

$$\ddot{d}_{isp}^{BT}(t) + \frac{c}{m}\dot{d}_{isp}^{BT}(t) + 4\pi^2 f_0^2 d_{isp}^{BT}(t) - \frac{\vartheta(t)}{m} = 0 \tag{4.2}$$

The conversion to the frequency domain by applying the Fourier Transform and the solution is given respectively in equations:

$$4\pi^2 f^2 \mathfrak{D}_{isp}^{BT}(f) - \jmath 2\pi\frac{c}{m} f \mathfrak{D}_{isp}^{BT}(f) - 4\pi^2 f_0^2 \mathfrak{D}_{isp}^{BT}(f) - \frac{V(f)}{m} = 0 \tag{4.3}$$

$$\mathfrak{D}_{isp}^{BT}(f) = \frac{V(f)e^{\jmath 2\pi\theta}}{2\pi m[[2\pi(f^2 - f_0^2)]2 + (c/m)^2 f^2]^1/2} \tag{4.4}$$

where $f_0 = \sigma/m$ corresponds to the natural frequency of the oscillator determined by the aforementioned stiffness property, σ, of the system and c is the damping factor. θ is explicitly computed as

$$\theta = tan^{-1} = \frac{cf/m}{2\pi(f^2 - f_0^2)} \tag{4.5}$$

to include the phase-shift.

$V(f)$ corresponds to the spectral approximations of the wind. From this point on, the the computation of the wind field using Zhang's method [174]. This formulation provides more spatial coherency for wind formulation in the image scene, instead of assuming a constant velocity at each point. The method is actually proposed is devoted to 3D applications, henceforth, a 2D truncation will be provided for the specified framework.

Zhang et al. partitions the wind field to have three orthogonal components, namely longitudinal, lateral and elevation vectors. The lateral components is kept out of consideration to model 2D wind velocities. Hence, the formulation is focused on the computation of modeling $L = L_m + L_f$ corresponding to the DC and the fluctuating parts the longitudinal wind vector and E, the elevational vector field with no DC part.

Modeling the wind field starts with the computation of the Cross-Spectral Power Density for each direction. The fluctuating parts of the velocity fields are regarded as the outputs of a Gaussian process. $CSPD$ is formulated as a matrix as shown with the used of allowing equations:

$$S(f) = \begin{bmatrix} s_{\alpha_1\alpha_1}(\omega) & s_{\alpha_1\alpha_2}(\omega) & s_{\alpha_1\alpha_3}(\omega) & \cdots & s_{\alpha_1\alpha_n}(\omega) \\ s_{\alpha_2\alpha_1}(\omega) & s_{\alpha_2\alpha_2}(\omega) & s_{\alpha_2\alpha_3}(\omega) & \cdots & s_{\alpha_2\alpha_n}(\omega) \\ s_{\alpha_3\alpha_1}(\omega) & s_{\alpha_3\alpha_2}(\omega) & s_{\alpha_3\alpha_3}(\omega) & \cdots & s_{\alpha_3\alpha_n}(\omega) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{\alpha_n\alpha_1}(\omega) & s_{\alpha_n\alpha_n}(\omega) & s_{\alpha_n\alpha_3}(\omega) & \cdots & s_{\alpha_n\alpha_n}(\omega) \end{bmatrix}$$

where $s_{\alpha_m\alpha_l} = \sqrt{s_{\alpha_m\alpha_m}(\omega)s_{\alpha_l\alpha_l}(\omega)}\mathcal{C}_\omega(\vartheta_m, \vartheta_l)$ and using the formulation in [144]

$$s_{\alpha_m\alpha_m} = \frac{\nu_*^2 \kappa_\alpha \mu}{(\omega/2\pi)[1+\Gamma_\alpha]^5} \tag{4.6}$$

and the coherence function $\mathcal{C}_\omega$,

$$\mathcal{C}_\omega(\vartheta_m, \vartheta_l) = \frac{\omega \sum_r (C_r|r_m - r_l|)}{2\pi L_m} \tag{4.7}$$

in which κ,Γ being arbitrary coefficients, ν_* denoting the shear velocity, and μ representing the Monin metric to be calculated as $\mu = \omega y L_m/2\pi$. C_r corresponds to a decay factor, where r are the image frame coordinates. The parameters are fixed to the values as indicated in the original paper [174].

The wind field is calculated using the the Cholesky [164] decomposition of the $CPSD$ matrix,assuming that the generated matrix is mostly real and symmetric.

$$S(\omega) = A(\omega)A^*(\omega)^T \tag{4.8}$$

Using the fact that $A(\omega)$ is a lower-triangular matrix, evaluating $A(\omega_\jmath), \jmath = 1, 2, ..N$, L_f can be evaluated in the spatial domain as

$$L_f = \sum_\jmath \sum_{k=1}^{\imath} a_{\imath k}(\omega_\jmath)\sqrt{\nabla\omega} \tag{4.9}$$

and E in a similar manner by decomposing the $CPSD$ for the vertical direction. L is computed for the main wind direction, and finally an affine transformation is applied using the orientation of the mean velocity vector to obtain the velocity values for the image coordinates.

Now that the wind velocity at each pixel is present, this velocity maps,V_x & V_y will be used to animate not only the flora but all the objects on the image. For the trees, the sway motion modeling will be proceeded by using corresponding velocity value in the map to be the mean velocity, v_{av} for that object. This mapping provides a visual harmony within the video frame.

The $V(f)$ field obtained by using domain-specific filter, $T(f)$

$$V(f) = N(f)T(f) \tag{4.10}$$

where $T(f)$ is defines as

$$T(f) = \sqrt{\frac{v_{av}}{(1 + Bf/v_{av})^{5/3}}} \tag{4.11}$$

and the multiplication by a random Gaussian noise field $N(f)$ in temporal frequency for the aforementioned desire of arbitrariness. Going back to equation 4.3 the computed velocity fields in the frequency domain are inserted to compute for the motion field of the tip in the frequency domain and it is succeed by the inverse Fourier transformation that yields the displacement map in the spatio-temporal domain.

The procedure is finalized by the propagation of the displacement values for all the pixels along the branch/trunk. If the skeleton [48] of the branch is taken to be the line that connects the pivot and the tip, the pixels that lie along the same orthogonal line with respect to the skeleton are taken to have the same displacement values to keep the layer compact.

4.4.2 Bodies of Water

4.4.2.1 Related Works

Modelling bodies of water is another tough challenge due to its non-rigid structure and fluidity. However, owing to its omnipresence in the impressionist paintings, it is a must have model to be implemented within this framework. However, the independent fluid animation is kept out of the scope of the thesis work due to the degree of freedom [54], [87] and the shallow/coastal waves [132], [173] are not considered due to their complex interaction with the shores [108]. This thesis focuses on the modeling of deep water waves, such as ocean waves and river flows that are created in the presence of natural drag forces, the gravitation and the wind.

Since the superposition of sinusiods to represent ocean waves in [62], a great deal of contribution has been made to this animation field. A certain group of modeling methods are based on the laws of fluid mechanics [61], [40] or its numerical applications [86], [39], [93]. Other methods are based on yet again spectral filtering, where Mastin et al. [108] make use of Pierson-Moskowitz Spectrum, Thon et al. employ ocean waves spectrum [152] and Tesseldorf et al. propose the utilization of Phillips Spectrum [151]. More recent spectral models include Texel, Marson and Arsole(TMA) in [25], [94] and JONSWAP in [74]. Additionally, [63], [78] rely on geometrical modelling of waves using meshes.

4.4.2.2 Motion Model - Ripple

The model chosen for the animation of surfaces on the bodies of water is based on the application Philips Spectrum that was proposed in [151], in which a spatial random noise field is filtered by the ocean-wave generating filter to create the animation.

$$O_P(\sigma) = \frac{e^{-1/(\sigma\kappa)^2}}{\sigma^4} |\sigma\prime \cdot v_{av}\prime| \tag{4.12}$$

$$D_0 = \Gamma \sqrt{O_p}(\sigma)\mathcal{N}(\sigma)/\sqrt{2} \tag{4.13}$$

where $\mathcal{N}(\sigma) = N_{re}(\sigma) + \jmath N_{im}(\sigma)$, $\quad \kappa = v_{av}^2/g$ acts as a gravitational term and $\sigma\prime$,$v_{av}\prime$ correspond to normalized 2D matrices. It is be recalled that v_{av} is taken from the velocity map calculated for the animation of trees from the spatial location of the approximate centre of the fluid body on the image.

The computed D_0 corresponds to the initial 2D height-map of the water surface and by making use of the following expression this initial surface state is turned into time-varying field to simulate the fluctuations on the surface. By taking the inverse FFT in 2D, the height map in the time domain is obtained. The procedure proceeds mathematically as in equation 4.14.

$$D(\sigma, t) = D_0(\sigma)e^{\jmath\sqrt{g\sigma}t} + D_0^*(-\sigma)e^{-\jmath\sqrt{g\sigma}t} \tag{4.14}$$

where the exponential term refers to the dispersion function between spatial frequency & phase velocity.

$$d_{isp}(x, y, t) = \mathcal{IFFT}_{2D}D(\sigma, t) \tag{4.15}$$

4.4.3 Boats

4.4.3.1 Motion Model - Heaving

For the animation of the boats, when the boat is anchored, which means it remains the same location; however it performs a heaving motion due to the water surface ripples,the model is made selection of two points (a line) on below the boat hull and rotating the boat depending on the change in the orientation of the line. If the motion includes an active movement as well, then the initial selected line is translated towards the boats trajectory. In this manner, the it is provided that the boat show a heading and heaving motion at the same time.

4.5 Motion Synthesis

When the models are generated, which are time dependent complex functions, it is time to synthesize the motion out of these models by building a motion armature. This armature is indeed a 5D field $DM(x, y, t, \partial x, \partial y)$ that guides all the pixels of the image in the spatio-temporal domain to displace them in the correct amount and orientation to achieve expected visual changes. In order to build the map, the given spectral filters and eventually the filter outputs are discretized in spatial coordinates for within frame displacement and in temporal axis to determine the displacement of each pixel in time. This manifold will be applied on the image, as well as the alpha matte layers to produce the final output.

4.6 Results & Implementation Details

There is an important remark to be made before diving into the subject, which is, the results shown within this chapter do not only contain to not show a single object moving in void but rather a scene that the object is contained in. Indeed, these results are the outputs of the ultimate block, namely the **Rendering Block**. Nevertheless, the illustration of motion without the surrounding would look incomplete, that is why the complete animation scenes are provided within the scope of this chapter.

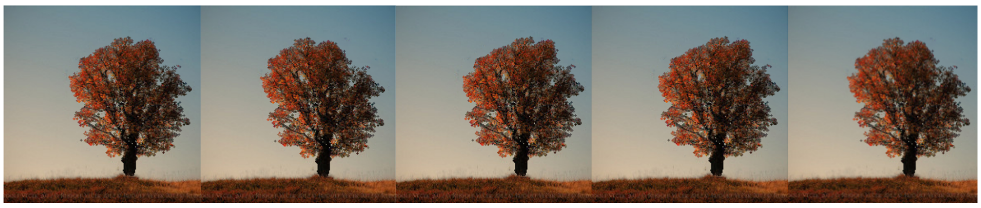
4.6.1 Trees & Flora

For each tree or branch that is required to be animated, the orientation is either drawn by the user, if not the body is assumed to be perpendicular to the to the y-axis of the image frame, and the bounding box extraction is used to determine the parallel lines of motion. Then the mentioned filter in subsection 4.4.1 is applied. The controllable (meaningful) parameters are stiffness, dampness which are fixed for this application, and the wind which effects the entire system is used from the generated wind field. Figure 4.1 illustrates the effect of v_{av}, average wind velocity on the motion of a single tree. For large trees the mass parameter m is taken to be 500 and for small trees it is 100.

4.6.2 Bodies of Water

A height map is generated by using the spectral filtering method for the motion water bodies. Nevertheless, this height map cannot be embedded to the 5D motion armature to synthesize the sequence of frames. This is because of the fact that, the water surface is plane that does not lie parallel on the image frame.

The proposed solution in [37], [174] is to manipulate the water pixels, w_2^p to be located in a 3D-domain by using an affine transformation determined by a


(a) $v_{av} = 5$

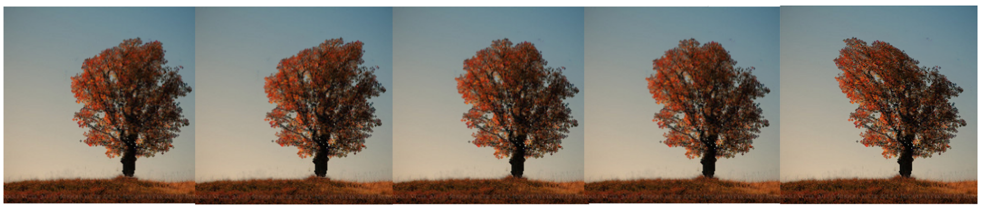

(b) $v_{av} = 20$

Figure 4.1: Effect of average velocity on the motion of tree, with stiffness and damping parameters are kept fixed.

user-specified horizon.

$$T_{form} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \alpha & \beta & \gamma \end{bmatrix}$$

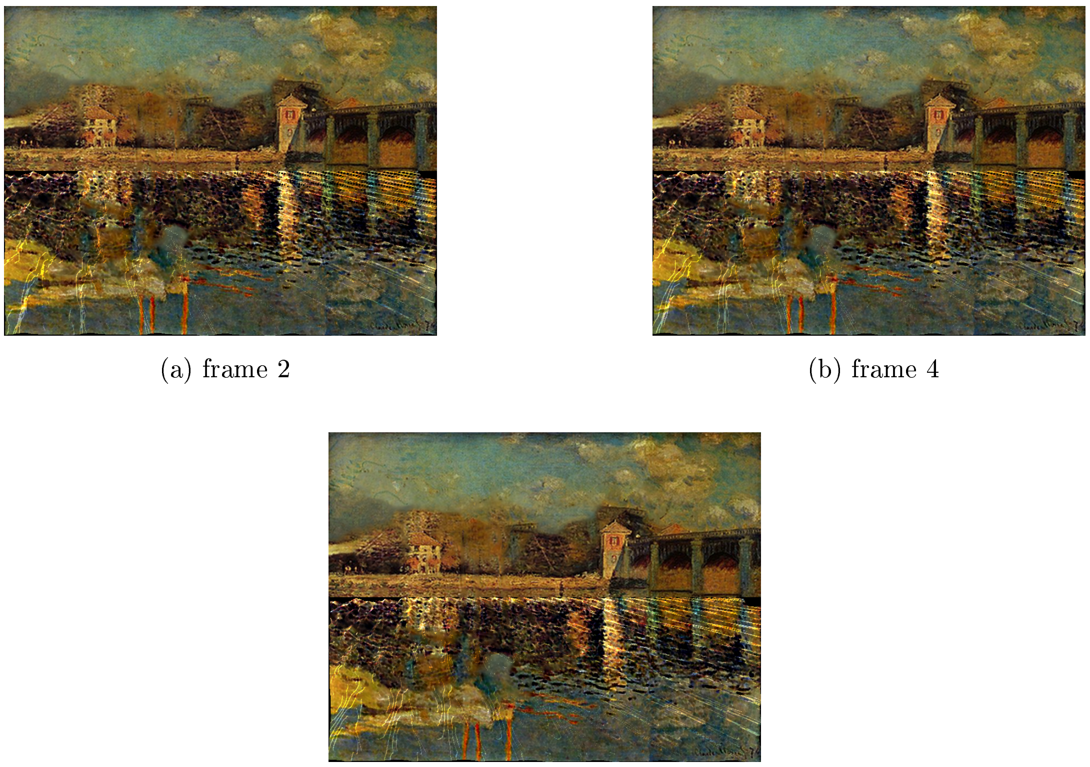
Using the horizon-line $\alpha x + \beta y + \gamma = 0$ obtain the 3D coordinates, w_3^p as

$$w_3^p = T_{form} w_2^p \tag{4.16}$$

On this map the pixels are added as the vertical dimension as the elevation information and re-projected to the image frame using the inverse transformation. It is be cleared that for more vivid look, the water surface is filtered in the frequency domain by using a Gaussian High Pass filter. Figure 4.2 illustrates an example animation of the surface of water.

4.6.3 Boats

The heaving line of the boats may be taken as a user-provided parameter, but it is also not illogical to use the bounding box information, obtain in the **Digital Matting Block**, to act as the rim of the boat hull. An example heaving is illustrated in Figure 4.3.


(a) frame 2

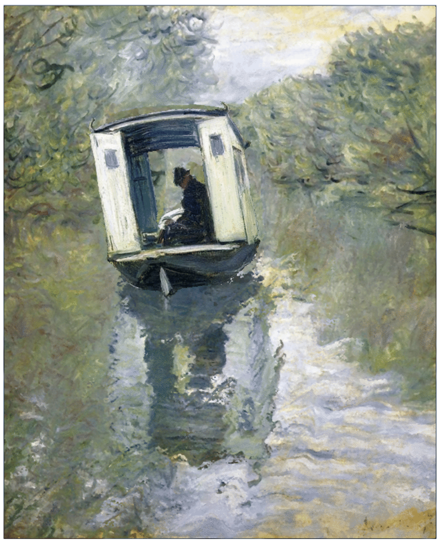
(b) frame 4

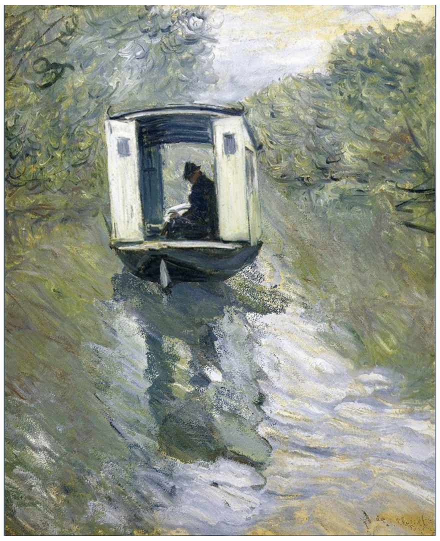
(c) frame 7

Figure 4.2: An example animation of a water surface by Phillips Spectrum [151].

4.6.4 Other Objects and Effects

The study of augmenting still images with motion can be regarded as a means to capture the dynamics of an instance. Hence, for further enhancement of the images for better visuals, there are occasional additions of animating active objects. The implementation of these objects contain more heuristic manipulation of the image layer such as combinations of translational and rotational motion models that are implemented with trial and error. Hence, for the visuals the reader is kindly invited to refer animation results in Chapter 5. Also for extra motion in the background, especially in the sky and clouds, one of the implemented texture synthesis methods for the tests on texture-oriented point of view is used.


(a) frame 2


(b) frame 4

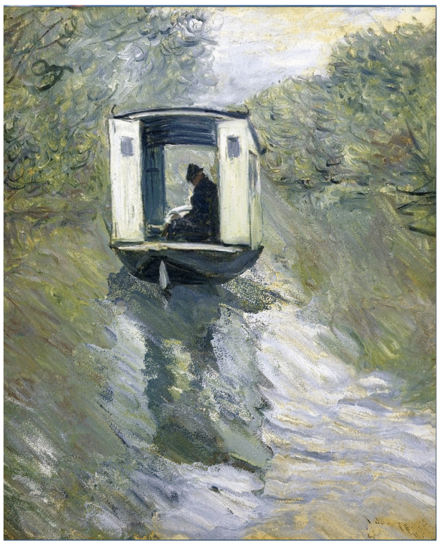

(c) frame 7

Figure 4.3: An example animation of an anchored boat showing the heaving motion

CHAPTER 5

RENDERING & ANIMATION RESULTS

Rendering Block is the ultimate block of the system before the final output is reached. The purpose of the block is very straightforward and simple, that is to apply the 5D motion armature $DM(x, y, t, \partial x, \partial y)$ synthesized in the preceding block. 5D provides the information of how each current pixel would behave directly.

5.0.1 Synthesis of a Single Frame

The reason for keeping this block separate is beacuse of the fact that this armature cannot be directly applied to the image. The layering that is done in the second block is to be respected to prevent the boundary violation of different layer pixels. Therefore, while processing the image under the motion armature, the extracted alpha mattes are also processed similarly. The alpha value at a specific image point guides the fractional blending of the intensity values at that location by using the foreground information with the new background. This provides the production of composite images with a seamless finish. Additionally, it also discards the necessity of user-supervision to order the layers with respect to the perspective information when the motion is slight.

When the motion is large, and the layer overlaps are significant that foreground pixel values,(with large alpha values) end up in the same pixel, a priority assistance is needed. This is the case especially when a translation is present, for instance in the clouds. The user provided information of the perspective helps the conservation of visual coherence in the scene during synthesis.

During the rendering process, it is frequent that some pixels reach spatial coordinates that are outside the image frame. The gap that is created by the absence of this pixel information is filled by interpolating its spatial neighbours in any synthesized frame.

5.0.2 Synthesis Along the Time Axis

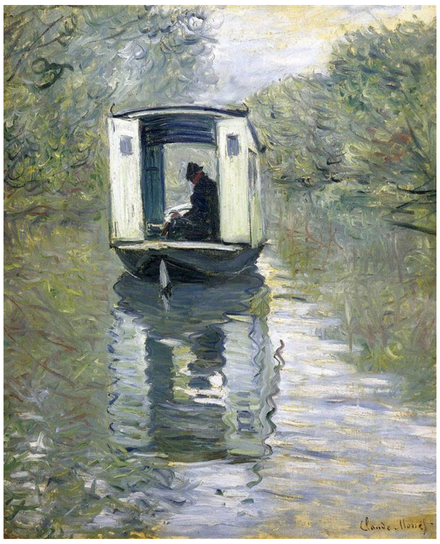
The output of this final block, which is also those of the entire system, is a video sequence that is composed of image frames. Apart from all the parameters that have been discussed on the animation, the quality of motion is also very correlated with the frame rate.

Single frame synthesis was realized by keeping the value of t fixed. Now, it is time to produce frames with different t's, which is a parameter of the motion armature, DM. Then, these separate frames are written to video file in the order of chronology. An important remark is to be made that, for a fixed frame rate in the video production, if the step size between each t, namely $\triangle t$ is taken to be large, the motion apparent on the video will be faster, in extreme cases resulting in an effect similar to fast forward mode in multi-media players.

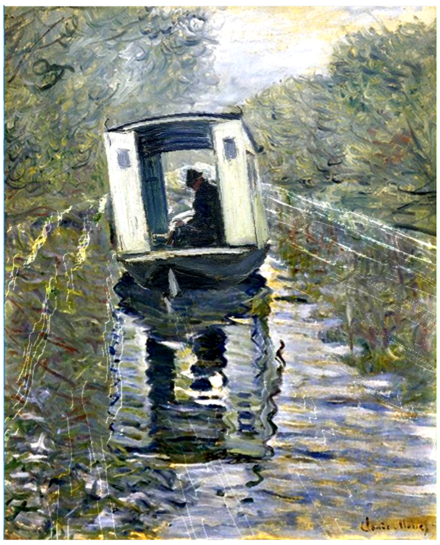
5.0.3 Animation Results

In this subsection, the video results are presented by displaying some frames that belong to the videos and explaining how they are animated. Also a comparison table is provided at the end of the example video results to provide timing information.

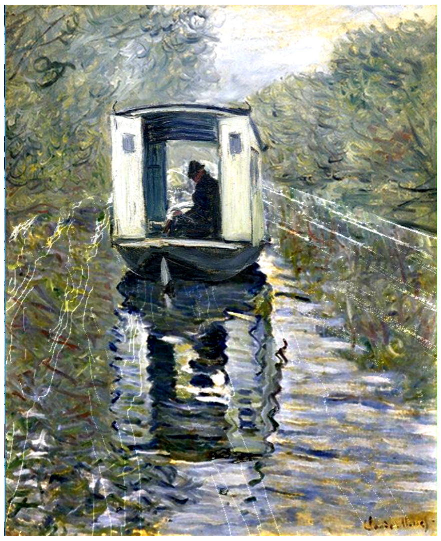
Monet's Boatstudio It is one of the simplest animation examples generated within the system. Includes 2 layers of animation, heaving and ripple, a long with a still layer. The results are given in Figure 5.1.



(a)



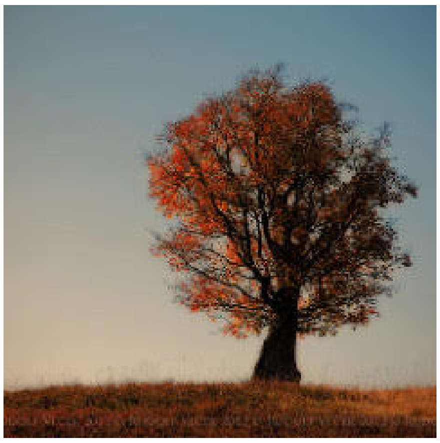
(b) frame 7

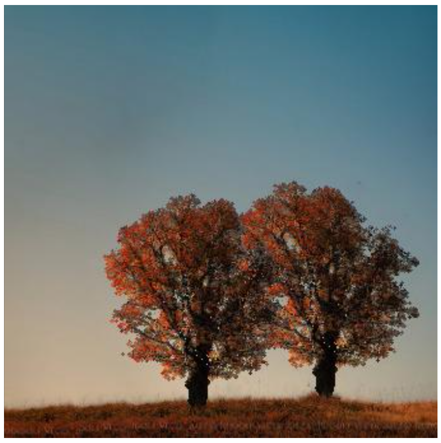


(c) frame 15

Figure 5.1: (a) Original Image. (b-c) 2 Frames of the Animated Video

A Tree with a Friend This example is rather different than the other ones as it includes artificially replicated objects of interest. After the layering and the inpainting procedures are completed, the background layer undergoes a texture synthesis operation to enlarge the image frame. The tree, the object of interest in this example, is duplicated, and well as its alpha matte to have a more crowded scene, even a forest can be generated by increasing the number of trees. In the motion model,the essential idea to apply is initializing each of the trees using different noise fields to provide the off-phase look as visible in 5.2.


(a)


(b) frame 5

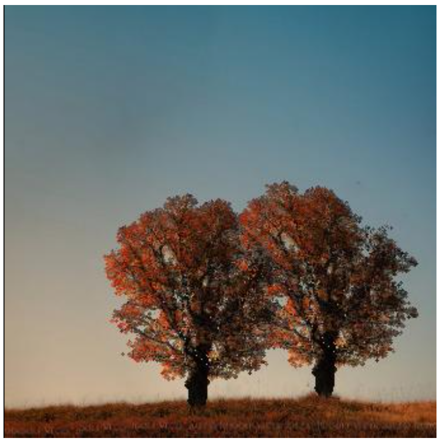
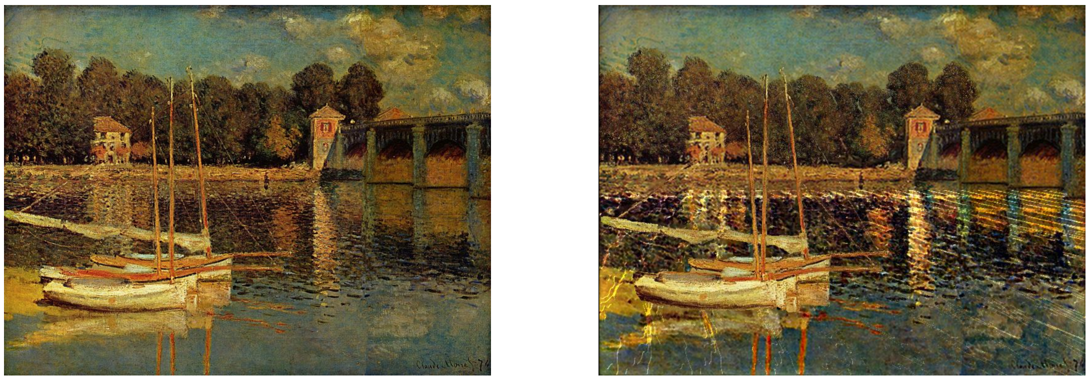

(c) frame 12

Figure 5.2: (a) Original Image. (b-c) 2 Frames of the Animated Video

Monet's Argentuil The importance of this example is the complexity of the scene for each and every block within the system. Image contains 9 large trees, 3 boats, a river and clouds to be animated. Therefore, in the synthesis of this video, layering and inpainting parts take significant amount of time. As layering is repeated for the increased number of objects, the inpainting method not only plays a role in object removal but also disocclusion, and it is repeated for the removal of each boat separately. Since the boats do heave at the presence of waves, and their motion is to be again off-phase with each other. The generated frames are given in Figure 5.3.


(a) (b) frame 35

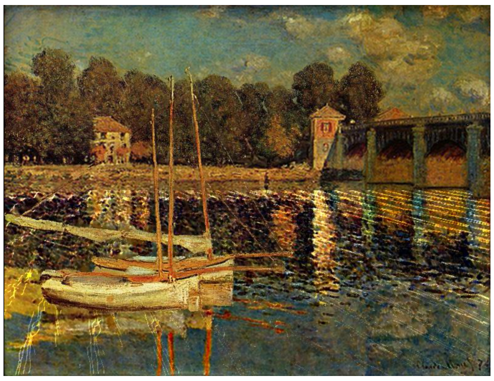
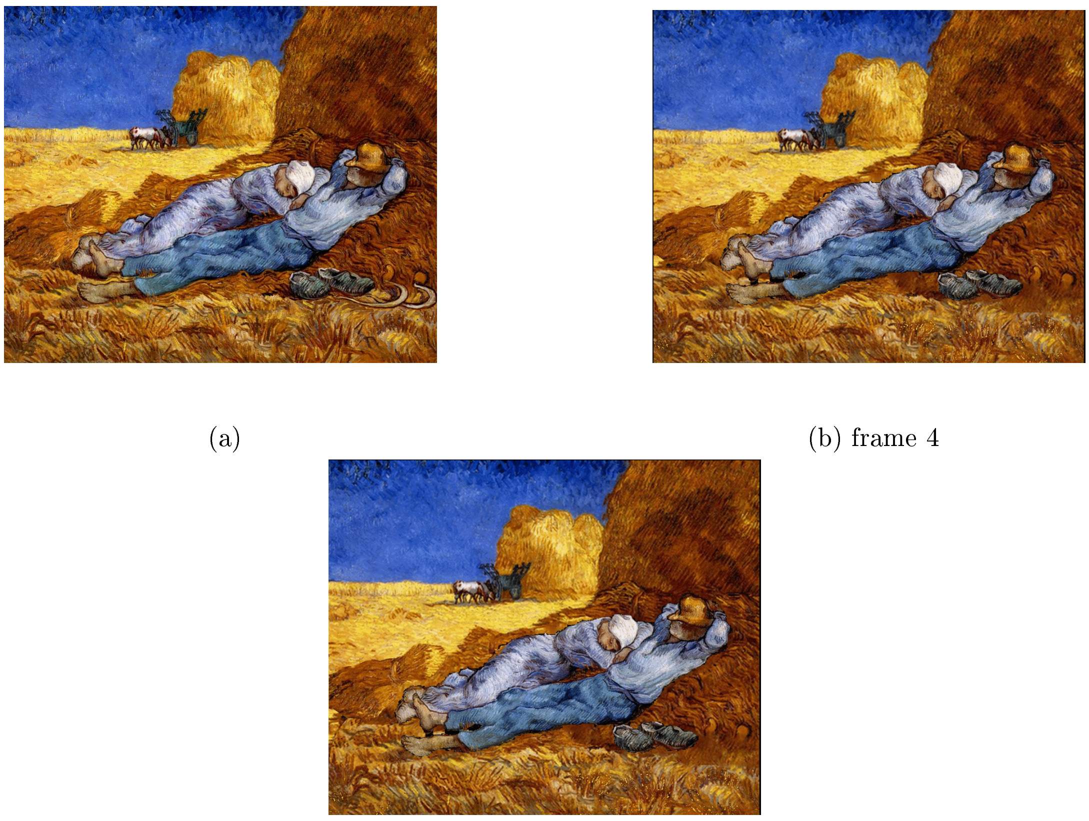

(c) frame 42

Figure 5.3: (a) Original Image. (b-c) 2 Frames of the Animated Video

Van Gogh's Wheat Field This animation is based on experimentation of the adaptability of the motion models to different objects that the models are not intended for. For this, once again a level inpainting for disocclusion is used, first to remove the couple,then the foreground wheat layer in front of them. The sway motion model is manipulated in this animation to replicate the the look of shiver in the wheatfield. Also, a very finely tuned version of the same model again is used to model the in-sleep quiver of the human body. The hay stack near the feet of the couple is anchored to the shivering wheat for shivering effect on the stack as well. The translational motion of the clouds are enhanced by enlarging the region using texture synthesis. The frames are given in 5.4.



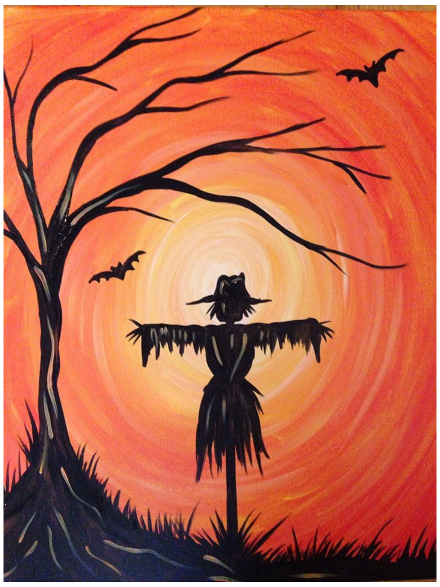
(a)

(b) frame 4

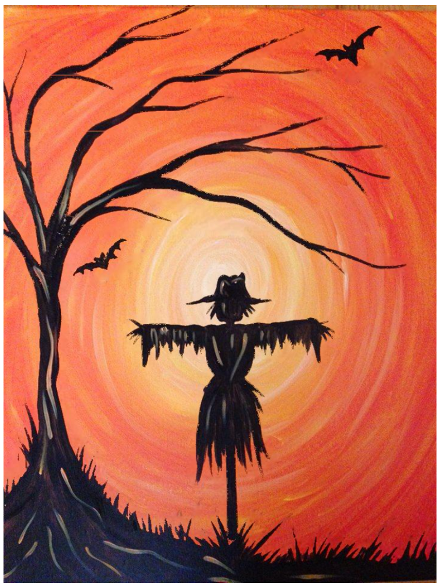
(c) frame 16

Figure 5.4: (a) Original Image. (b-c) 2 Frames of the Animated Video

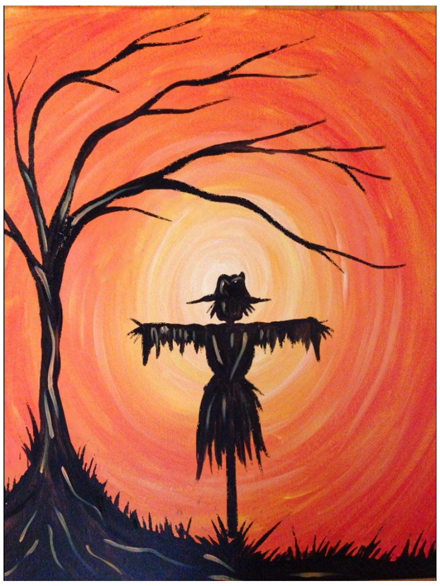
The Scarecrow This model includes many foreground objects of motion; however the inpainting is realized in a single-shot. The motion of the scarecrow is modelled as a swaying motion and the bats are made rotate slowly around an out of boundary reference point as observed in 5.5.



(a)



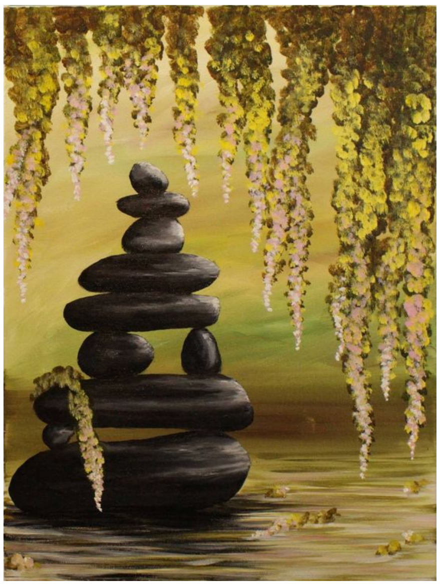
(b) frame 13



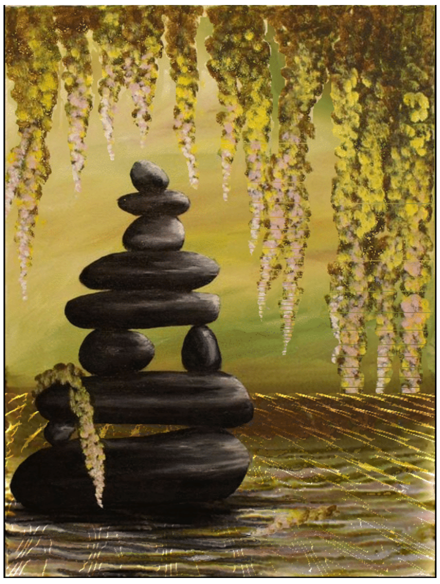
(c) frame 62

Figure 5.5: (a) Original Image. (b-c) 2 Frames of the Animated Video

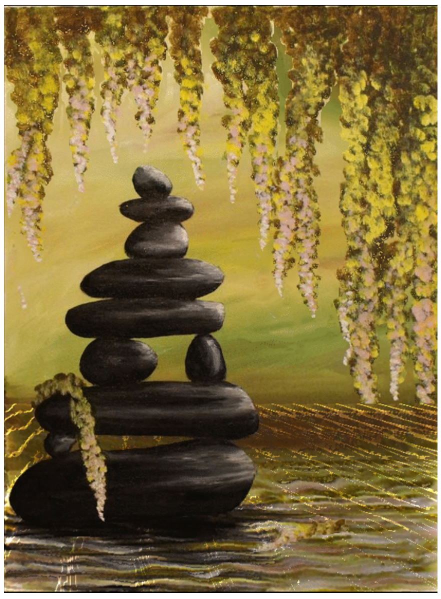
The Zen Garden The motion of the ivy is regarded as the sway motion that is pivoted to a point above as contrary to the trees which are pivoted below. Also, the motion of the leaves on the surfaces of the shallow water mimics those of the boats heaving on water, as shown in 5.6.



(a)



(b) frame 10



(c) frame 21

Figure 5.6: (a) Original Image. (b-c) 2 Frames of the Animated Video

The results in 5.1 demonstrate that after the completion layering and inpainting problems the creation of the animated scenes is quite rapid. The animations that are signed with $*$ demonstrate that texture synthesis is also included in the inpainting part. The changes in the frame generation rate owes to the difference

Animation Comparison					
Image	Segmentation Layers	Segmentation Duration(m)	Inpainting Layers	Inpainting Duration(m)	Motion Synthesis& Rendering Rate (sec/frame)
Argentuil	15	20	4	35	2.5
BoatStudio	3	4	1	11	0.8
Scarecrow	6	9	1	15	1.3
TwoTrees	2	1.3	2∗	7	0.7
Wheat Fields	6	8	3∗	22	1.3
Zen	6	12	1	16	1.7

Table5.1: Animation details of example produced video sequences.

in the number of layers to be merged as well as the size of the initial image.

CHAPTER 6

CONCLUSION

In this thesis, the main goal was two-fold, one of which was to implement a framework for semi-automatic still image animation. The fundamental steps of producing animation with such an approach included four consecutive tasks of image matting, image inpainting, motion modeling & synthesis and rendering. The latter goal of this thesis was to analyse the image inpainting techniques & evaluate their performance within the context of still image inpainting.

Primarily, focusing on the synthesis task as a whole, it is concluded that image-to-video animation is not an easy task to fulfill such that it demanded the implementation of distinct operational blocks. The domain of animation was limited to paintings to avoid the necessity of retouching for the seek of photo-realism and the motion domain was mostly restricted to passive objects in nature for the sake of ease.

For the image matting step, the implemented method had been proposed by Shahrian et al. [135] which proved to be effective in computing alpha channels to allow the composition of seamless images even in textured regions. This emphasis was essential for our focus domain of paintings that was likely to feature highly textured areas. The inclusion of a texture feature in accompaniment with the colour vector allowed the extraction of the foreground from the background even though the colour histograms were overlapping. The experimental results proved that the algorithm had the required strength in extracting satisfying transparency maps without the usage of too strict guidance.

Besides, focusing on the second goal in the thesis work, it was realised that quality of a proper image inpainting is essential in this framework for increasing the perspective and perception of depth. A wide research was conducted on different the inpainting methodology, and a main classification of approaches was made based on the point of view to preserve the image elements. As an image is composed of texture, structure and colour information, texture-preserving methods favoured the replication of the texture whilst geometry-preserving methods prioritized the continuum of structure above all. Within this classification the research had shown that the geo-texture preserving methods outperformed the others in the applications of object removal.

Therefore, 4 different geo-texture preserving methods was implemented and tested for their performance in object removal and region reconstruction. The first one that is known as Crimini's Method [42] employed heuristic on the order of filling to preserve the continuum of image isophates and performed single shot patch-wise filling of the image hole using exemplars from the known region. The experiments demonstrated that this method is highly effected by the selection of patchsize for a visually plausible filling as it may be trapped in local minima if it is too small due to not being able to encapsulate the energy of the image features, or may cause blocking effects if the selection is larger than required.

In addition to that, the other 3 methods were indeed proposed to be ANNF building methods but were utilized in a coarse-to-fine approach to be adapted to image inpainting. The initialization of the NNFs in all methods depended on the interpolation of the boundary to take the image gradients and geometry into account. The adaptations were similar in general, only small changes were included. The application of inpainting via PatchMatch [14] is dependent on again the selection of patch sizes as well as the number of logarithmic scales and number of random iterations. It was observed that since it does not search the whole image in a brute for manner, as the number of random searches through iterations were increased, the matches began to converge to more accurate values. As the implementation depends on weighted averaging of the patch values, if the NNF is trapped in local minima the output appears to be significantly blurred.

Moreover, inpainting with CSH depends on the patch similarities using the projections on 2D Walsh Hadamard kernels, both in candidate selection and rejection. The inpainting application provided sharp completion of the image; however, as it is highly dependent of hash similarities and coherence propagation, if it is not finely tuned for correct parameters the completion may result in undesired results. Number of logarithmic scales is an important parameter, as well as the patch size, but within this method the patch size selection is restricted to orders of 2 due to the utilization of WH kernels as bases. Throughout the experiments, the number hashbins in the chroma and luma channels effected the performance of the completion algorithm where edge information was not strong, and increased in the number of k-nearest neighbours led the inpainted region to have information from different parts of the image frame.

Furthermore, hole filling with image melding involves the preservation of structure inevitably by employing geometric and photometric transformations on the image patches while query phase. Additionally, while pasting through the discrete screened Poisson equation, the minimization on the gradient of the luma channel is also employed so as to put more emphasis on the geometry. Results showed that presumably, the visual performance of the method is affected by the patch size, number of logarithmic scales and number of iterations, that are similar to other methods, along with the changes in the ranges of scaling and rotation, inclusion of reflection and image enhancement parameters that augment the plausibility of the completion. In visual comparison, I-IM proved to give the best results in image completion.

Precisely, for the comparison of these methods, certain cases of object removal were tested, also for the sake completeness in quantitative analysis of performance, the known parts of the images were masked and the 4 methods were tested for their ability to reconstruct these known regions using PSNR, SNR and SSIM metrics, where for each I-IM outperformed its rivals and I-CSH scored the lowest quality values on average. On the other hand, the timing analysis revealed that the slowest inpainting was realized by I-IM, whilst I-CSH achieved a rapid filling rate. The experimental results demonstrated that depending on the expectation of the outcome each method tested had their own benefits and drawbacks. In order to solve this ill-posed problem, it is required to prioritize expectations very clearly as the task becomes a trade-off between faster operation and higher quality outputs.

In the motion modeling & synthesis part, the restrictions in the modeling relieved the system from the burden of modeling utterly complex motion models. The problem is approached by expressing harmonic motion domain via spectral filtering methods that produced satisfying visuals without the requirement extreme parameter tuning and user-interaction. The inclusion of the random noise field captured the required arbitrariness that is present in nature and the conversion of the models was realized with ease by discretization of the models in the spatial and temporal axes.

The rendering was done through emerging of the layers by using the transparency information produced in the matting block. The usage of these alpha channels proved to be significant in a seamless blending within the animated video frames. For temporal arrangement of the video frames, the frame rate and the time step parameters were significant to obtain a high quality flow in the artificial motion.

To conclude, the framework provided the synthesis of high quality short videos out of single images. However, it was observed that the quality of the output animation video is inter-dependent on each of these blocks. The experiments demonstrated that an incomplete matting procedure may result in undesirable inpainting and an incomplete inpainting layer may look worse than before when animated, that is to say, the error here is cumulated and multiplied at each block. Clearly, it is necessitated to improve the performance of all the subsystems within the still-image to video converter so as to achieve satisfying final output videos.

For future work, it is expected to transform the framework into one with deep architecture that is capable of automatically recognizing a group of objects and animate them. With the employment of this methodology, it is also expected to work with vaster variety of motion models and it is believed that photo-realistic results even with the usage of photograph will be achieved.

REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein. *rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[2] Y. Akagi and K. Kitajima. Computer animation of swaying trees based on physical simulation. *Computers & Graphics*, 30(4):529–539, 2006.

[3] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro. A variational framework for exemplar-based image inpainting. Technical report, MINNESOTA UNIV MINNEAPOLIS INST FOR MATHEMATICS AND ITS APPLICATIONS, 2010.

[4] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.

[5] M. Ashikhmin. Synthesizing natural textures. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 217–226. ACM, 2001.

[6] M.-F. Auclair-Fortier and D. Ziou. A global approach for solving evolutive heat transfer for image denoising and inpainting. *IEEE Transactions on Image Processing*, 15(9):2558–2574, 2006.

[7] D. Auroux, L. J. Belaid, and B. Rjaibi. Application of the topological gradient method to color image restoration. *SIAM Journal on Imaging Sciences*, 3(2):153–175, 2010.

[8] D. Auroux, L. D. Cohen, and M. Masmoudi. Contour detection and completion for inpainting and segmentation based on topological gradient and fast marching algorithms. *Journal of Biomedical Imaging*, 2011:4, 2011.

[9] D. Auroux and M. Masmoudi. A one-shot inpainting algorithm based on the topological asymptotic analysis. *Computational & Applied Mathematics*, 25(2-3):251–267, 2006.

[10] S.-H. Baek, I. Choi, and M. H. Kim. Multiview image completion with space structure propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 488–496, 2016.

[11] 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(8):1200–1211, 2001.

[12] T. Barbu. Hybrid variational restoration technique based on second-and fourth-order diffusions. In *Development and Application Systems (DAS), 2016 International Conference on*, pages 1–5. IEEE, 2016.

[13] T. Barbu. Variational image inpainting technique based on nonlinear second-order diffusions. *Computers & Electrical Engineering*, 54:345–353, 2016.

[14] C. Barnes, D. B. Goldman, E. Shechtman, and A. Finkelstein. The patchmatch randomized matching algorithm for image manipulation. *Communications of the ACM*, 54(11):103–110, 2011.

[15] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized patchmatch correspondence algorithm. In *European Conference on Computer Vision*, pages 29–43. Springer, 2010.

[16] C. Barnes and F.-L. Zhang. A survey of the state-of-the-art in patch-based synthesis. *Computational Visual Media*, pages 1–18, 2017.

[17] G. Ben-Artzi, H. Hel-Or, and Y. Hel-Or. The gray-code filter kernels. *IEEE transactions on pattern analysis and machine intelligence*, 29(3), 2007.

[18] M. Bertalmio. Strong-continuation, contrast-invariant inpainting with a third-order optimal pde. *IEEE Transactions on Image Processing*, 15(7):1934–1938, 2006.

[19] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[20] M. Bertalmío, V. Caselles, S. Masnou, and G. Sapiro. Inpainting. In *Computer Vision, A Reference Guide*, pages 401–416. 2014.

[21] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[22] A. L. Bertozzi, S. Esedoglu, and A. Gillette. Inpainting of binary images using the cahn–hilliard equation. *IEEE Transactions on image processing*, 16(1):285–291, 2007.

[23] A. Blake, P. Kohli, and C. Rother. *Markov random fields for vision and image processing*. Mit Press, 2011.

[24] F. Bornemann and T. März. Fast image inpainting based on coherence transport. *Journal of Mathematical Imaging and Vision*, 28(3):259–278, 2007.

[25] E. Bouws, J. Ephraums, J. Ewing, P. Francis, H. Gunther, P. Janssen, G. Komen, W. Rosenthal, and W. De Voogt. A shallow water intercomparison of three numerical wave prediction models (swim). *Quarterly Journal of the Royal Meteorological Society*, 111(470):1087–1112, 1985.

[26] T. Briand, J. Vacher, B. Galerne, and J. Rabin. The heeger & bergen pyramid based texture synthesis algorithm. *Image Processing On Line*, 4:276–299, 2014.

[27] J. Buhler. Efficient large-scale sequence comparison by locality-sensitive hashing. *Bioinformatics*, 17(5):419–428, 2001.

[28] I. BUZCU. *REPRESENTING IMAGES AND REGIONS FOR OBJECT RECOGNITION*. PhD thesis, MIDDLE EAST TECHNICAL UNIVERSITY, 2015.

[29] V. Caselles. Exemplar-based image inpainting and applications. *SIAM News*, 44(10):1–3, 2011.

[30] T. F. Chan and J. Shen. Morphologically invariant pde inpaintings. 2001.

[31] T. F. Chan and J. Shen. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449, 2001.

[32] T. F. Chan, J. Shen, and H.-M. Zhou. Total variation wavelet inpainting. *Journal of Mathematical imaging and Vision*, 25(1):107–125, 2006.

[33] L. Chen, H. Wei, and J. Ferryman. A survey of human motion analysis using depth imagery. *Pattern Recognition Letters*, 34(15):1995–2006, 2013.

[34] X. Chen, Y. Shen, and Y. H. Yang. Background estimation using graph cuts and inpainting. In *Proceedings of Graphics Interface 2010*, pages 97–103. Canadian Information Processing Society, 2010.

[35] X. Chen, B. Zhou, F. Xu, and Q. Zhao. Automatic image completion with structure propagation and texture synthesis. *International Journal of Software Engineering and Knowledge Engineering*, 20(08):1097–1117, 2010.

[36] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2001.

[37] Y.-Y. Chuang, D. B. Goldman, K. C. Zheng, B. Curless, D. H. Salesin, and R. Szeliski. Animating pictures with stochastic motion textures. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 853–860. ACM, 2005.

[38] B. Cirkovic. Image segmentation as a classification task in computer applications. *Center for Quality*, 2014.

[39] J. Cohen and S. Belcher. Turbulent shear flow over fast-moving waves. *Journal of Fluid Mechanics*, 386:345–371, 1999.

[40] G. Compère, J.-F. Remacle, and E. Marchandise. Transient mesh adaptivity with large rigid-body displacements. *Proceedings of the 17th International Meshing Roundtable*, pages 213–230, 2008.

[41] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2003.

[42] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004.

[43] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Trans. Graph.*, 31(4):82–1, 2012.

[44] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM, 2004.

[45] I. Daubechies and B. J. Bates. Ten lectures on wavelets. *The Journal of the Acoustical Society of America*, 93(3):1671–1671, 1993.

[46] J. S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 361–368. ACM Press/Addison-Wesley Publishing Co., 1997.

[47] E. De Giorgi, M. Carriero, and A. Leaci. Existence theorem for a minimum problem with free discontinuity set. *Archive for Rational Mechanics and Analysis*, 108(4):195–218, Nov 1989.

[48] J. Diener, M. Rodriguez, L. Baboud, and L. Reveret. Wind projection basis for real-time animation of trees. In *Computer graphics forum*, volume 28, pages 533–540. Wiley Online Library, 2009.

[49] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[50] I. Drori, D. Cohen-Or, and H. Yeshurun. Fragment-based image completion. In *ACM Transactions on graphics (TOG)*, volume 22, pages 303–312. ACM, 2003.

[51] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001.

[52] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999.

[53] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (mca). *Applied and Computational Harmonic Analysis*, 19(3):340–358, 2005.

[54] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 736–744. ACM, 2002.

[55] S. Esedoglu and J. Shen. Digital inpainting based on the mumford–shah–euler image model. *European Journal of Applied Mathematics*, 13(4):353–370, 2002.

[56] V. Fedorov, G. Facciolo, and P. Arias. Variational framework for non-local inpainting. *Image Processing On Line*, 5:362–386, 2015.

[57] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 30(1):36–51, 2008.

[58] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *International journal of computer vision*, 87(3):284–303, 2010.

[59] M. Filipović, I. Kopriva, and A. Cichocki. Inpainting color images in learned dictionary. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 66–70. IEEE, 2012.

[60] L. R. Ford Jr and D. R. Fulkerson. *Flows in networks.* Princeton university press, 2015.

[61] N. Foster and D. Metaxas. Realistic animation of liquids. *Graphical models and image processing*, 58(5):471–483, 1996.

[62] A. Fournier and W. T. Reeves. A simple model of ocean waves. *ACM Siggraph Computer Graphics*, 20(4):75–84, 1986.

[63] M. N. Gamito and F. K. Musgrave. An accurate model of wave refraction over shallow water. *Computers & Graphics*, 26(2):291–307, 2002.

[64] E. S. Gastal and M. M. Oliveira. Shared sampling for real-time alpha matting. In *Computer Graphics Forum*, volume 29, pages 575–584. Wiley Online Library, 2010.

[65] A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529, 1999.

[66] B. Guo and Y.-Q. Xu. Chaos mosaic: Fast and memory efficient texture synthesis. 01 2000.

[67] R. Habel, A. Kusternig, and M. Wimmer. Physically guided animation of trees. In *Computer Graphics Forum*, volume 28, pages 523–532. Wiley Online Library, 2009.

[68] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. *ACM transactions on graphics (TOG)*, 30(4):70, 2011.

[69] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Transactions on Cybernetics*, 43(5):1318–1334, Oct 2013.

[70] J. Han, K. Zhou, L.-Y. Wei, M. Gong, H. Bao, X. Zhang, and B. Guo. Fast example-based surface texture synthesis via discrete optimization. *The Visual Computer*, 22(9-11):918–925, 2006.

[71] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.

[72] A. S. Hareesh and V. Chandrasekaran. Exemplar-based color image inpainting: a fractional gradient function approach. *Pattern Analysis and Applications*, 17(2):389–399, 2014.

[73] P. Harrison. A non-hierarchical procedure for re-synthesis of complex textures. 2001.

[74] D. Hasselmann, M. Dunckel, and J. Ewing. Directional wave spectra observed during jonswap 1973. *Journal of physical oceanography*, 10(8):1264–1280, 1980.

[75] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238. ACM, 1995.

[76] Y. Hel-Or and H. Hel-Or. Real-time pattern matching using projection kernels. *IEEE transactions on pattern analysis and machine intelligence*, 27(9):1430–1445, 2005.

[77] A. Hertzmann. Paint by relaxation. In *Computer Graphics International 2001. Proceedings*, pages 47–54. IEEE, 2001.

[78] D. Hinsinger, F. Neyret, and M.-P. Cani. Interactive animation of ocean waves. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 161–166. ACM, 2002.

[79] Z.-x. Hou, Y.-q. He, and W. Xu. A fast algorithm of image inpainting. *Journal of image and Graphic*, 12(10):1909–1912, 2007.

[80] J. Howard, B. Morse, S. Cohen, and B. Price. Depth-based patch scaling for content-aware stereo image completion. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 9–16. IEEE, 2014.

[81] G. M. Hunter. Computer animation survey. *Computers & Graphics*, 2(4):225–229, 1977.

[82] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.

[83] J. Jia and C.-K. Tang. Image repairing: Robust image synthesis by adaptive nd tensor voting. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2003.

[84] Y. Kabata, H. Nishino, K. Utsumiya, and K. Korida. An interactive modeling method for dynamic natural objects. In *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, volume 4, pages 212–217. IEEE, 1999.

[85] G. Kanizsa. *Organization in vision: Essays on Gestalt perception.* Praeger Publishers, 1979.

[86] M. Kass and G. Miller. Rapid, stable fluid dynamics for computer graphics. In *ACM SIGGRAPH Computer Graphics*, volume 24, pages 49–57. ACM, 1990.

[87] B. M. Klingner, B. E. Feldman, N. Chentanez, and J. F. O'brien. Fluid animation with dynamic meshes. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 820–825. ACM, 2006.

[88] R. Köhler, C. Schuler, B. Schölkopf, and S. Harmeling. Mask-specific inpainting with deep neural networks. In *German Conference on Pattern Recognition*, pages 523–534. Springer, 2014.

[89] N. Komodakis and G. Tziritas. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Transactions on Image Processing*, 16(11):2649–2661, 2007.

[90] S. Korman and S. Avidan. Coherency sensitive hashing. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1607–1614. IEEE, 2011.

[91] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra. Texture optimization for example-based synthesis. *ACM Transactions on Graphics (ToG)*, 24(3):795–802, 2005.

[92] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003.

[93] A. T. Layton and M. van de Panne. A numerically efficient and stable algorithm for animating water waves. *The Visual Computer*, 18(1):41–53, 2002.

[94] N. Lee, N. Baek, and K. W. Ryu. Real-time simulation of surface gravity ocean waves based on the tma spectrum. In *International conference on Computational Science*, pages 122–129. Springer, 2007.

[95] S. Lefebvre and H. Hoppe. Parallel controllable texture synthesis. In *ACM Transactions on Graphics (ToG)*, volume 24, pages 777–786. ACM, 2005.

[96] J. Lei, C. Zhang, Y. Fang, Z. Gu, N. Ling, and C. Hou. Depth sensation enhancement for multiple virtual view rendering. *IEEE Transactions on Multimedia*, 17(4):457–469, April 2015.

[97] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242, 2008.

[98] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics (ToG)*, 20(3):127–150, 2001.

[99] H.-t. Lim, H. G. Kim, and Y. M. Ro. Learning based hole filling method using deep convolutional neural network for view synthesis. *Electronic Imaging*, 2016(14):1–5, 2016.

[100] Y. Liu and V. Caselles. Exemplar-based image inpainting using multi-scale graph cuts. *IEEE transactions on image processing*, 22(5):1699–1711, 2013.

[101] J. Long, C. Reimschussel, O. Britton, A. Hall, and M. Jones. Motion capture for a natural tree in the wind. *Motion in Games*, pages 158–169, 2010.

[102] B. Lubin and G. Tewari. Faster fragment-based image completion.

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

[104] A. J. Majda and A. L. Bertozzi. *Vorticity and incompressible flow*, volume 27. Cambridge University Press, 2002.

[105] T. März. Image inpainting based on coherence transport with adapted distance functions. *SIAM Journal on Imaging Sciences*, 4(4):981–1000, 2011.

[106] T. März. A well-posedness framework for inpainting based on coherence transport. *Foundations of Computational Mathematics*, 15(4):973–1033, 2015.

[107] S. Masnou and J.-M. Morel. Level lines based disocclusion. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, pages 259–263. IEEE, 1998.

[108] G. A. Mastin, P. A. Watterberg, and J. F. Mareda. Fourier synthesis of ocean scenes. *IEEE Computer graphics and Applications*, 7(3):16–23, 1987.

[109] G. McLachlan and T. Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[110] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198. ACM, 1995.

[111] T.-J. Mu, J.-H. Wang, S.-P. Du, and S.-M. Hu. Stereoscopic image completion and depth recovery. *The Visual Computer*, 30(6-8):833–843, 2014.

[112] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989.

[113] A. Nan and X. Xi. An improved criminisi algorithm based on a new priority function and updating confidence. In *Biomedical Engineering and Informatics (BMEI), 2014 7th International Conference on*, pages 885–889. IEEE, 2014.

[114] T. Ogawa and M. Haseyama. Image inpainting based on sparse representations with a perceptual metric. *EURASIP Journal on Advances in Signal Processing*, 2013(1):179, 2013.

[115] J. M. Ogden, E. H. Adelson, J. R. Bergen, and P. J. Burt. Pyramid-based computer graphics. *RCA Engineer*, 30(5):4–15, 1985.

[116] H. Ono. Practical experience in the physical animation and destruction of trees. In *Computer Animation and Simulation'97*, pages 149–159. Springer, 1997.

[117] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[118] S. Ota, M. Tamura, T. Fujimoto, K. Muraoka, and N. Chiba. A hybrid method for real-time animation of trees swaying in wind fields. *The Visual Computer*, 20(10):613–623, 2004.

[119] A. Owens, C. Barnes, A. Flint, H. Singh, and W. Freeman. Camouflaging an object from many viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2782–2789, 2014.

[120] K. E. Papoutsakis and A. A. Argyros. Integrating tracking with fine object segmentation. *Image and Vision Computing*, 31(10):771–785, 2013.

[121] L. Pessoa, E. Thompson, and A. Noë. Filling-in is for finding out. *Behavioral and brain sciences*, 21(6):781–796, 1998.

[122] M. Potter and B. Faulconer. Time to understand pictures and words. *Nature*, 253:437–438, 1975.

[123] M. Pratscher, P. Coleman, J. Laszlo, and K. Singh. Outside-in anatomy based character rigging. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 329–338. ACM, 2005.

[124] E. Praun, A. Finkelstein, and H. Hoppe. Lapped textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 465–470. ACM Press/Addison-Wesley Publishing Co., 2000.

[125] Y. Pritch, E. Kav-Venaki, and S. Peleg. Shift-map image editing. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 151–158. IEEE, 2009.

[126] G. Ramanarayanan and K. Bala. Constrained texture synthesis via energy minimization. *IEEE Transactions on Visualization and Computer Graphics*, 13(1), 2007.

[127] C. Rhemann, C. Rother, and M. Gelautz. Improving color modeling for alpha matting. In *BMVC*, volume 1, page 3, 2008.

[128] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. A perceptually motivated online benchmark for image matting. pages 1826–1833, 06 2009.

[129] D. L. Ruderman. The statistics of natural images. *Network: computation in neural systems*, 5(4):517–548, 1994.

[130] M. A. Ruzon and C. Tomasi. Alpha estimation in natural images. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 18–25. IEEE, 2000.

[131] T. Sakaguchi. Botanical tree structure modeling based on real image set. In *ACM SIGGRAPH 98 Conference abstracts and applications*, page 272. ACM, 1998.

[132] C. Schneggenburger, H. Günther, and W. Rosenthal. Spectral wave modelling with non-linear dissipation: validation and applications in a coastal tidal environment. *Coastal Engineering*, 41(1):201–235, 2000.

[133] C.-B. Schönlieb. *Partial Differential Equation Methods for Image Inpainting*, volume 29. Cambridge University Press, 2015.

[134] R. Sedgewick. *Algorithms*. Pearson Education India, 1988.

[135] E. Shahrian and D. Rajan. Weighted color and texture sample selection for image matting. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 718–725. IEEE, 2012.

[136] B. Shen, W. Hu, Y. Zhang, and Y.-J. Zhang. Image inpainting via sparse representation. pages 697–700, 04 2009.

[137] J. Shen and T. Chan. Variational restoration of nonflat image features: Models and algorithms. *SIAM Journal on Applied Mathematics*, 61(4):1338–1361, 2001.

[138] J. Shen and T. F. Chan. Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2002.

[139] J. Shen, S. H. Kang, and T. F. Chan. Euler's elastica and curvature-based inpainting. *SIAM Journal on Applied Mathematics*, 63(2):564–592, 2003.

[140] X. Shen, Y.-C. Chen, X. Tao, and J. Jia. Convolutional neural pyramid for image processing. *arXiv preprint arXiv:1704.02071*, 2017.

[141] M. Shinya and A. Fournier. Stochastic motion—motion under the influence of wind. In *Computer Graphics Forum*, volume 11, pages 119–128. Wiley Online Library, 1992.

[142] T. Shiratori, Y. Matsushita, X. Tang, and S. B. Kang. Video completion by motion field transfer. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 411–418. IEEE, 2006.

[143] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[144] G. Solari and G. Piccardo. Probabilistic 3-d turbulence modeling for gust buffeting of structures. 16:73–86, 01 2001.

[145] J. Stam. Stochastic dynamics: Simulating the effects of turbulence on flexible structures. In *Computer Graphics Forum*, volume 16. Wiley Online Library, 1997.

[146] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 315–321. ACM, 2004.

[147] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum. Image completion with structure propagation. *ACM Transactions on Graphics (ToG)*, 24(3):861–868, 2005.

[148] M. Sun, A. D. Jepson, and E. Fiume. *Video input driven animation (VIDA)*. IEEE, 2003.

[149] P. Tan, G. Zeng, J. Wang, S. B. Kang, and L. Quan. Image-based tree modeling. In *ACM Transactions on Graphics (TOG)*, volume 26, page 87. ACM, 2007.

[150] A. Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34, 2004.

[151] J. Tessendorf et al. Simulating ocean water. *Simulating nature: realistic and interactive techniques. SIGGRAPH*, 1(2):5, 2001.

[152] S. Thon, J.-M. Dischler, and D. Ghazanfarpour. Ocean waves synthesis using a spectrum-based turbulence function. In *Computer Graphics International, 2000. Proceedings*, pages 65–72. IEEE, 2000.

[153] T. Thonat, E. Shechtman, S. Paris, and G. Drettakis. Multi-view inpainting for image-based scene editing and rendering. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 351–359. IEEE, 2016.

[154] X. Tong, J. Zhang, L. Liu, X. Wang, B. Guo, and H.-Y. Shum. Synthesis of bidirectional texture functions on arbitrary surfaces. In *ACM Transactions on Graphics (ToG)*, volume 21, pages 665–672. ACM, 2002.

[155] C.-S. Tseng, C.-T. Lin, C.-W. Lin, and J.-H. Wang. Perceptual edges preservation conformal to human vision perception. In *Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference on*, pages 173–178. IEEE, 2012.

[156] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders. Segmentation as selective search for object recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1879–1886. IEEE, 2011.

[157] J. Wang and M. F. Cohen. An iterative optimization approach for unified image segmentation and matting. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 936–943. IEEE, 2005.

[158] J. Wang and M. F. Cohen. Optimized color sampling for robust matting. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[159] A. H. Watt and M. Watt. *Advanced animation and rendering techniques.* ACM press New York, NY, USA:, 1992.

[160] J. P. Weber. Fast simulation of realistic trees. *IEEE Computer Graphics and Applications*, 28(3), 2008.

[161] L.-Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR*, pages 93–117. Eurographics Association, 2009.

[162] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488. ACM Press/Addison-Wesley Publishing Co., 2000.

[163] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2004.

[164] J. H. Wilkinson. *The algebraic eigenvalue problem*, volume 87. Clarendon Press Oxford, 1965.

[165] B. Wohlberg. Inpainting with sparse linear combinations of exemplars. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 689–692. IEEE, 2009.

[166] B. Wohlberg. Inpainting by joint optimization of linear combinations of exemplars. *IEEE Signal Processing Letters*, 18(1):75–78, 2011.

[167] E. Wu, Y. Chen, T. Yan, and J. Feng. Reconstruction and physically-based animation of trees from static images. In *Computer Animation and Simulation'99*, pages 157–166. Springer, 1999.

[168] Z. Xu and J. Sun. Image inpainting by patch propagation using patch sparsity. *IEEE transactions on image processing*, 19(5):1153–1165, 2010.

[169] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539*, 2016.

[170] H. Ying, L. Kai, and Y. Ming. An improved image inpainting algorithm based on image segmentation. *Procedia Computer Science*, 107:796–801, 2017.

[171] A. Zalesny, V. Ferrari, G. Caenen, and L. Van Gool. Parallel composite texture synthesis. In *Texture 2002 workshop-ECCV*, 2002.

[172] S. Zelinka and M. Garland. Jump map-based interactive texture synthesis. *ACM Transactions on Graphics (TOG)*, 23(4):930–962, 2004.

[173] H. Zhang, H. A. Schäffer, and K. P. Jakobsen. Deterministic combination of numerical and physical coastal wave models. *Coastal engineering*, 54(2):171–186, 2007.

[174] L. Zhang, C. Song, Q. Tan, W. Chen, and Q. Peng. Quasi-physical simulation of large-scale dynamic forest scenes. *Advances in Computer Graphics*, pages 735–742, 2006.

[175] H. Zhou, L. Wei, D. Creighton, and S. Nahavandi. Inpainting images with curvilinear structures propagation. *Machine vision and applications*, 25(8):2003–2008, 2014.