2010

DIGITAL INPAINTING ALGORITHMS AND EVALUATION

Vijay Venkatesh Mahalingam
University of Kentucky, mvijay@engr.uky.edu

ABSTRACT OF DISSERTATION

Vijay Venkatesh Mahalingam

The Graduate School

University of Kentucky

2010

DIGITAL INPAINTING ALGORITHMS AND EVALUATION

ABSTRACT OF DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering
at the University of Kentucky

By

Vijay Venkatesh Mahalingam

Lexington, Kentucky

Director: Dr. Sen-ching Samson Cheung , Professor of Electrical Engineering

Lexington, Kentucky

2010

ABSTRACT OF DISSERTATION

DIGITAL INPAINTING ALGORITHMS AND EVALUATION

Digital inpainting is the technique of filling in the missing regions of an image or a video using information from surrounding area. This technique has found widespread use in applications such as restoration, error recovery, multimedia editing, and video privacy protection. This dissertation addresses three significant challenges associated with the existing and emerging inpainting algorithms and applications. The three key areas of impact are 1) Structure completion for image inpainting algorithms, 2) Fast and efficient object based video inpainting framework and 3) Perceptual evaluation of large area image inpainting algorithms.

One of the main approach of existing image inpainting algorithms in completing the missing information is to follow a two stage process. A structure completion step, to complete the boundaries of regions in the hole area, followed by texture completion process using advanced texture synthesis methods. While the texture synthesis stage is important, it can be argued that structure completion aspect is a vital component in improving the perceptual image inpainting quality. To this end, we introduce a global structure completion algorithm for completion of missing boundaries using symmetry as the key feature. While existing methods for symmetry completion require a-priori information, our method takes a non-parametric approach by utilizing the invariant nature of curvature to complete missing boundaries. Turning our attention from image to video inpainting, we readily observe that existing video inpainting techniques have evolved as an extension of image inpainting techniques. As a result, they suffer from various shortcoming including, among others, inability to handle large missing spatio-temporal regions, significantly slow execution time making it impractical for interactive use and presence of temporal and spatial artifacts. To address these major challenges, we propose a fundamentally different method based on object based framework for improving the performance of video inpainting algorithms. We introduce a modular inpainting scheme in which we first segment the video into constituent objects by using acquired background models followed by inpainting of static background regions and dynamic foreground regions. For static background region inpainting, we use a simple background replacement and occasional image inpainting. To inpaint dynamic moving foreground regions, we introduce a novel sliding-window based dissimilarity measure in a dynamic programming framework. This technique can effectively inpaint large regions of occlusions, inpaint objects that are completely missing for several frames, change in size and pose and has minimal blurring and motion artifacts. Finally we direct our focus on experimental studies related to perceptual quality evaluation of large area image inpainting algorithms. The perceptual quality of large area inpainting technique is inherently a subjective process and yet no previous research has been carried out by taking the subjective nature of the Human Visual System (HVS). We perform subjective experiments using eye-tracking device involving 24 subjects to analyze the effect of inpainting on human gaze. We experimentally show that the presence of inpainting artifacts directly impacts the gaze of an unbiased observer and this in effect has a direct bearing on the subjective rating of the observer. Specifically, we show that the gaze energy in the hole regions of an inpainted image show marked deviations from normal behavior when the inpainting artifacts are readily apparent.

KEYWORDS: Digital Inpainting, Occluded Symmetry Structure Completion, Object Based Video Inpainting, Dynamic Programming, Eye-tracker based Perceptual Evaluation

(Vijay Venkatesh Mahalingam)

(April 2010)

DIGITAL INPAINTING ALGORITHMS AND EVALUATION

By

Vijay Venkatesh Mahalingam

Dr. Sen-ching Samson Cheung

(Director of Dissertation)

Dr. Stephen Gedney

(Director of Graduate Studies)

April 2010

(Date)

RULES FOR THE USE OF DISSERTATIONS

Unpublished dissertations submitted for the Doctoral degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgements.

Extensive copying or publication of the dissertation in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

A library that borrows this dissertation for use by its patrons is expected to secure the signature of each user.

Name	Date

DISSERTATION

Vijay Venkatesh Mahalingam

The Graduate School

University of Kentucky

2010

DIGITAL INPAINTING ALGORITHMS AND EVALUATION

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering
at the University of Kentucky
By

Vijay Venkatesh Mahalingam

Lexington, Kentucky

Director: Dr. Sen-ching Samson Cheung , Department of Electrical and Computer

Engineering

Lexington, Kentucky

2010

Amma, Appa, Priya and Vasu

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere gratitude towards my advisor Prof. Sen-ching Samson Cheung for his valuable guidance, encouragement and continuous financial support throughout my doctoral study and dissertation work. During all these years, I have had the great opportunity to closely work with Samson and his work ethic and brilliance has been an inspiration to me in all aspects of my life. He has been very supportive through the hard times and has always encouraged me to aim higher and do better. I will always be indebted to his mentoring and it has been really a fruitful association that I would cherish throughout my entire life. Thank you Samson, you have been a wonderful advisor and a great mentor!

Next, I would like to thank Prof. Kevin D. Donohue who has been always available for me with a good word and wise counsel. I have learned to balance my work and always treat difficulties with a little humor and his support throughout my master's program has been a blessing. I would also like to thank Prof. YuMing Zhang, Prof. Ruigang Yang and Prof. Cheng for taking time to read my dissertation and providing valuable comments.

I would like to thank my friends and my lab mates in MIALAB for being very helpful and supportive. Jithendra, James, Edwin, Viswajith, Ying and all my lab mates have been wonderful company. I would like to thank people at the Center of Visualization and Virtual Environment for providing excellent educational environment conducive for students. I also appreciate volunteers who participated in my experiments to complete this work.

Above all, I would not be here without the help and support of my family. There is nothing I could do to repay their faith in me and through all these years they have been there for me always and it is the only thing I trust. My younger brother Vasu has been like my elder brother and without him I don't think that I would be at this stage. My Sister Priya has been the one I turn to when times are difficult and just talking to her cheers me up a great deal and there is no kinder soul that I know. My greatest gift is Amma and Appa. It is their unconditional love, support and blessings along with the mercy of the almighty that I am here today. I love you all, you are simply the best.

Table of Contents

List of Tables

List of Figures

List of Files

1. VijayVenkateshPhDdissertation.pdf

Chapter 1

Introduction

Digital inpainting is the technique of filling in the missing regions of an image using information from surrounding area. In the parlance of digital inpainting, the missing region is often referred to as *hole*, and is usually provided by the user in the form of mask or can be obtained by automatic or semi-automatic means. Some of the earlier nomenclature referred small region filling as inpainting and large area inpainting as image or video completion. In this work however, we do not make any such distinctions and these techniques are commonly referred as Digital Image and video inpainting algorithms. Digital inpainting has found widespread use in many applications such as restoration of damaged old paintings and photographs, removal of undesired objects and writings on photographs, transmission error recovery in images and videos, computer-assisted multimedia editing and replacing large regions in an image or video for privacy protection. The goal of the inpainting technique is to modify the damaged region in an image or video in such a way that the inpainted region is undetectable to a neutral observer. Alternately, as described in [2], the objective of inpainting is to reconstitute the missing or damaged portions of the work, in order to make it more legible and restore its unity. Based on the context of operation, the goal of the inpainting can range from making the damaged image or video appear as close to the original to completely providing an alternate completion which is virtually unnoticeable to human observer.

In this dissertation, we identify and explore three main challenges that are associated with existing and emerging digital inpainting algorithms. First, to improve the structure completion capability of image inpainting algorithm; second, to design a fast and efficient video inpainting algorithm that can perform robustly under varying operating conditions, and third, to analyze the perceptual quality of large area image inpainting algorithms. We propose new techniques and algorithms to tackle these challenges and demonstrate their effectiveness under various operating conditions. Before embarking on a detailed description of our contributions, we first provide an introduction to the concept of digital inpainting, discuss emerging applications where they are useful and elaborate on the research challenges associated with our goals.

1.1 Digital Inpainting

Historically, inpainting has been manually performed by artists to restore damaged paintings and photographs with small defects such as small cracks, scratches, red-eye and dust spots. At this juncture, it is pertinent to note that inpainting is fundamentally different from classical de-noising algorithms. In case of denoising algorithms, the underlying signal is corrupted by noise and de-noising algorithms try to recover the the original signal by modeling the noise and statistical estimation. Typically, a noise model, such as Additive White Gaussian Noise (AWGN) or speckle noise, and an underlying Image model such as piecewise continuity is assumed. There are several spatial and transform domain filtering techniques that are used to remove the noise and recover the true image. On the other hand, in inpainting, the regions of missing information in the original signal are large and inpainting algorithm attempts to re-create it using the surrounding information. Hence, denoising algorithms are not applicable in inpainting applications. The examples representing de-noising and inpainting are presented in Figures 1.1(a) and Figures 1.1(b) respectively.

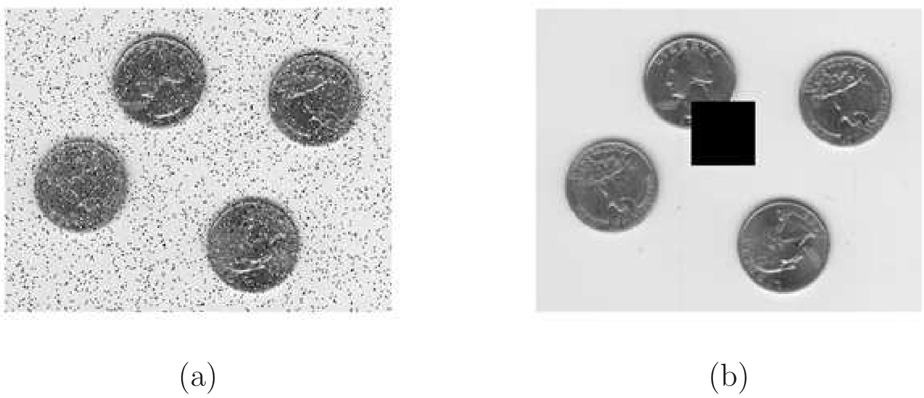


Figure 1.1: *De-noising versus Inpainting: (a) Image corrupted by noise for de-noising application. (b) Damaged input image with missing region or "hole" for image inpainting.*

1.1.1 Digital Inpainting in Image Restoration

To illustrate the use of inpainting in restoration type applications, we present an example original image with defects such as cracks and scratches is shown in Figure 1.2(a), and the corresponding user specified mask hole in Figure 1.2(b). Given these inputs, an inpainting algorithm then strives to fill the hole based on the statistics of the rest of the image. The result of applying inpainting based on our implementation of the algorithm proposed in [3] is presented in Figure 1.2(c).

1.1.2 Digital Inpainting in Privacy Protection Application

There has been a recent surge of interest in obfuscating privacy information in visual information system. In video surveillance front, the PrivacyCam surveillance

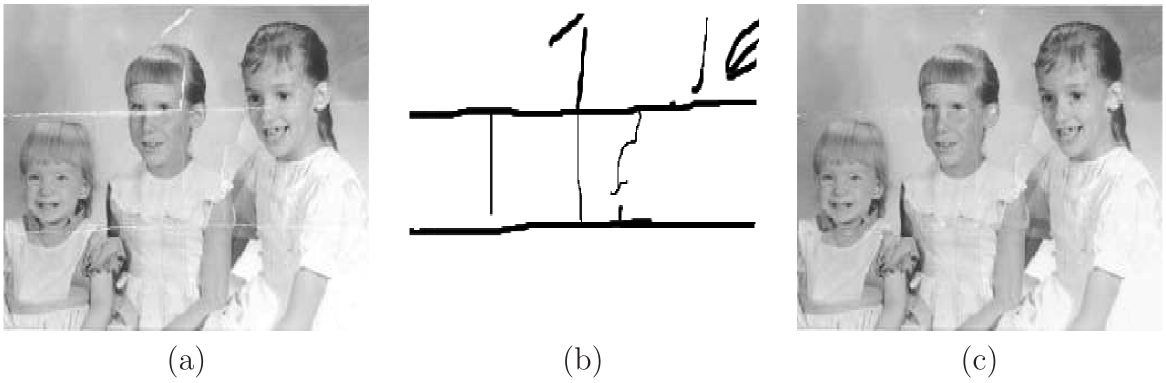


Figure 1.2: *Example of Digital image inpainting : (a) Original Image. (b) Damaged regions or "hole" in the form of mask. (c) Inpainted image based on algorithm proposed in [3].*

system developed at IBM protects privacy by revealing only the relevant information such as object tracks or suspicious activities [4]. Such a system is limited by the types of events it can detect and may have problems balancing privacy protection with the particular needs of a security officer.

Alternatively, one can modify the video to obfuscate the appearance of individuals for privacy protection. There are a large variety of such kinds of video obfuscation techniques, ranging from the use of black boxes or large pixels (pixelation) in [5–7] to complete object removal in [8,9]. For example, we see that the identifiable private information such as license plates of vehicles, human faces etc, are blurred in visual maps provided by publicly available search engines. New techniques have also been proposed recently to replace a particular face with generic face [10] or a body with a stick figure [11]. The use of black boxes or pixelation has been shown to be inadequate in fully protecting a person's identity [10]. Face or body replacement require precise position and pose tracking which are beyond the reach of current surveillance technologies. On the other hand, complete removal of private objects by image and video inpainting techniques provide a good solution for full privacy protection while preserving a natural-looking video amenable to further vision processing. The technical challenges that are involved in recreating occluded objects and motion after the removal of individuals offer significant research potential for inpainting applications.

Let us describe a typical scene from an indoor video surveillance environment capable of privacy protection. To preserve the privacy of authorized individuals, we would like to remove those authorized people from the video information and record the movements of the unauthorized visitors. Readily we observe that, there are two distinct scenarios 1) Instances where there are no overlaps between the two parties - non-occluding case and 2) Instances when there are overlaps among the two parties, or occluding scenarios. In indoor video surveillance conditions, it is reasonable to assume that the background models are available and hence background replacement can be used efficiently as an inpainting method to protect the privacy in non-occluding conditions. Figure 1.3(a) presents a sample original frame in which the person of interest, whose privacy we need to protect, is shown carrying an object. We extract moving foreground objects, by comparing each frame with the existing background model and pixels that are substantially different from the background are classified as foreground. These foreground objects are created based on a connected component grouping, expanded to form bounding boxes and finally tracked based on the overlap of objects from frame to frame as displayed in Figure 1.3(b). Figure 1.3(c) shows the inpainted frame by background replacement using an adaptive background model based on Kalman filter [12] and replacing it over the extracted foreground. We also apply a de-blocking filter to smooth the transition between fill-in and rest of the image to make it look more natural.

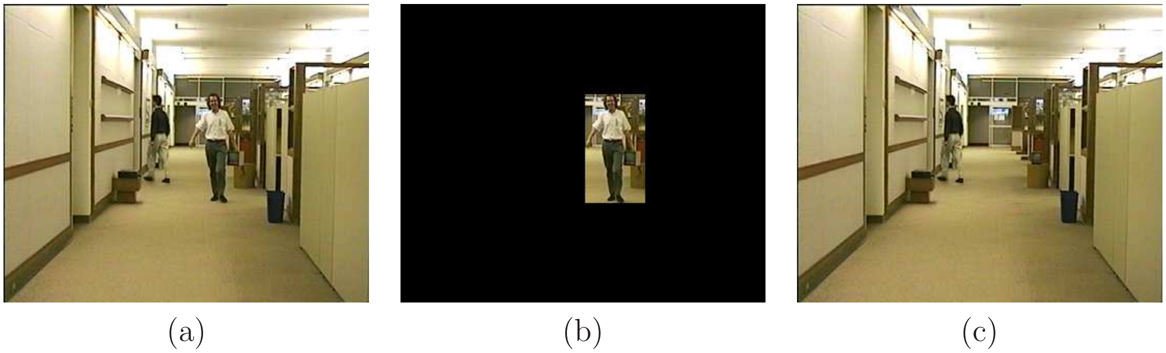


Figure 1.3: *Video inpainting by simple background replacement: (a) Original video frame. (b) Selected foreground object to be erased. (c) Result of our proposed algorithm.*

The more challenging situation occurs in handling the occluding conditions and this is where video inpainting truly offers a powerful alternative. By using the existing spatio-temporal data present in the unoccluded regions, video inpainting attempts to reconstruct the missing foreground regions thereby recovering the occluded information. The advantage of using in video inpainting here is that the processed videos maintain their homogeneity. As a result, these videos can be used for further processing by other computer vision algorithms along the pipeline.

From the two examples cited above, it is fairly clear that as the application arena expands from restoration type changes to multimedia object editing in which users typically need to remove an entire object, inpainting algorithms face a formidable challenge of filling in a significant portion of an image. In particular, we can observe that there are two related but fundamentally challenging operations in inpainting; 1. Object removal and 2. Object creation. In the former, we are trying to remove an entire object and hence the hole region can be filled by models of background or by texture replication methods. In the latter, however, we are dealing with inpainting partially available objects and it presents an entirely different set of difficulties which require a more precise structure and region filling capabilities to arrive at a perceptually good completion. Hence traditional approaches based on simple de-noising or interpolation, in which small-size holes are assumed fail to apply and large region inpainting in images and videos potentially offer a larger impact.

1.2 Our Contributions

While significant advancements have been made in improving the capability of digital inpainting algorithms, they still face many challenges to be used under different circumstances in various applications. In this section, we identify three main issues that face existing inpainting algorithms, discuss our motivation to address them and present a concise overview of our approach.

1.2.1 Structure Completion for Image Inpainting Algorithms

Currently existing image inpainting techniques perform reasonably well in tasks wherein the region to be inpainted is textureless and small as shown in Figure 1.2. When the area to be inpainted is relatively large and requires completion of areas spanning boundary regions, they do not perform well. This is due to the fact that existing inpainting applications endeavor to fill the hole by a smoothing process which tends to blur the regions and do not handle the boundary regions explicitly. To illustrate this point, we present an example which shows the difficulties faced by the state of the art algorithm such as in [13], on an image rich in features. In Figure 1.4(a) and Figure 1.4(b) we have an unmodified original image along with an user introduced synthetic hole. We have the result of applying algorithm in [13] in 1.4(c). The result does not appear to be satisfactory as we can observe that the technique is hampered by its inability to inpaint large regions and recreate curved boundary structures. An intuitive way to complete the missing regions is to segment the image into different regions, complete the boundaries of the respective regions and fill in the individual regions with appropriate texture information. While it may be fairly easy for humans to mentally imagine the "fill-in" component inside the hole, it is still a formidable challenge for computer vision algorithm to automatically estimate the missing regions.

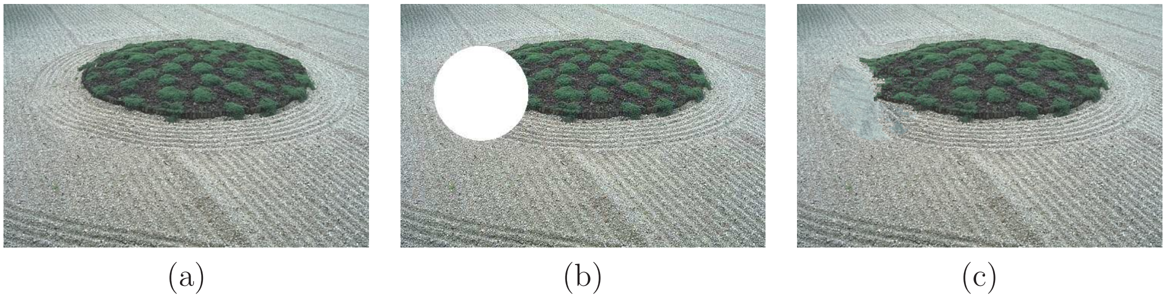


Figure 1.4: *Challenges in large area inpainting (a) Original Image. (b) Image with a hole. (c) Inpainted Image by using algorithm in [13].*

From this example, it can be clearly understood that completing the boundaries of the respective regions inside the hole, also referred to as *structure completion*, plays a vital role in providing a perceptually consistent inpainting. Hence to produce a more effective boundary completion, we will have to take into account other important global structural cues that may be present in the image. It is conceivable that using mid-level image cues such as depth, repeating object patterns, symmetries and t-junctions can improve the global completion capability which would subsequently enhance the image inpainting quality. In Chapter 3, we introduce a contour completion algorithm for filling the missing boundaries of a symmetric object under severe occlusions [14]. This non-parametric algorithm is capable of completing the boundaries of rotationally symmetric objects without any prior information on the order of symmetry by utilizing the invariant nature of curvature. This technique can be used in the structure completion stage of global image inpainting algorithm.

1.2.2 Fast and Efficient Object Based Video Inpainting

Earlier video inpainting techniques evolved as an extension of the image inpainting algorithms and were originally used to fix small scratches or blotches in vintage videos [15]. While being useful, they are severely limited by the size of the hole they were able to handle and by the inordinate amount of time for processing. Due to the computationally intensive nature of inpainting process, it is clear that for more advanced and interactive applications such as privacy protection, they are practically unsuitable. Some video inpainting methods are capable of inpainting simple periodic repetition motions such as human walking and running, however, there are are no video inpainting techniques that can inpaint human motion under changing pose conditions. Another important challenge in video inpainting is to make it suitable to work under different constraints so that it can be useful in many practical conditions. Even advanced video inpainting techniques such as [16], are adapted to extend the exemplar based techniques used in image inpainting for video data. We believe that video offers a great amount of useful information which can be exploited to make the inpainting faster and more perceptually coherent. While existing methods offer limited advantages, use of templates of human motion by continuously observing the patterns of human activity could provide more potential both in terms of speed and efficiency. They also offer promise to handle large occlusions including instances where the entire objects are missing for several time instances. In chapter 4, we propose a fundamentally different object based framework by combining templates of human motion for video inpainting [17, 18]. Our system utilizes simple background replacement and occasional image inpainting for static background inpainting. We introduce a novel sliding-window based dissimilarity measure in a dynamic programming framework to inpaint moving objects. This computationally efficient technique can effectively handle large regions of occlusions, inpaint objects that are completely missing for several frames, operate under change of pose and size of humans, inpaint videos captured in fixed as well as moving camera conditions and has minimal blurring and motion artifacts.

1.2.3 Subjective Evaluation of Large Area Image Inpainting Techniques

Evaluating the quality of an inpainting technique is a challenging task as it is fundamentally different from the general notion of the image quality. An image/video which is free of visual artifacts could still end up being classified as unsatisfactory inpainting implementation. For example, in synthetic images, completion of the geometric attributes of the image to present a visually plausible image could be considered to be more satisfactory than the appropriate correct texture replication. Similarly for natural images, continuation of texture inside natural edge boundaries might indicate a better inpainting algorithm. Thus we are facing a scenario in which the texture, structure, motion (for video), size of the hole and other geometric attributes of the underlying signal play a crucial role in the determining the effectiveness of an inpainting algorithm. The presence of such diverse attributes makes the evaluation of inpainting techniques harder.

In Section 2.3 we will review a number of techniques that analyzed the image inpainting quality. Among the methods discussed, the work by Hays et.al [1] and Ardis et.al [19] is of particular relevance to our study. While they attempt to rank the quality of inpainted images with motivation to determine which algorithm is better, they do not provide useful information on the underlying reason behind those human subjectivity. Extensive studies in vision science have shown that interesting objects are visually salient and these regions provide *surprises* which attract visual attention [20, 21]. In the context of image inpainting, we can extrapolate that techniques that cannot fill the hole in a consistent manner will introduce artifacts on the inpainted image. We hypothesize that these artifacts could capture the attention of HVS and cause a change from its normal behavior. This deviation from normalcy can be quantified by measuring the changes in gaze pattern using an eye-tracker and can be related to the perceptual quality of inpainting. Even though attention processes are partly driven by higher level cognitive processes, low and mid level visual process plays an important role in determining the perceptual quality of image inpainting. In Chapter 5, we design and conduct extensive subjective experiments and measure the changes from normal behavior in gaze pattern of HVS using an eye-tracker [22]. By analyzing those gaze pattern we establish a strong connection between gaze pattern and inpainted image quality. We believe that this relation, combined with a computational model in measuring visual saliency by using low and mid-level vision features in inpainted regions, will be useful for predicting inpainted image quality.

1.3 Organization

This dissertation is organized as follows: Chapter 2 provides a detailed review of existing approaches in inpainting and also discuss relevant work on evaluating the perceptual quality of image inpainting algorithms. In Chapter 3, we introduce a novel contour completion algorithm for completing rotationally symmetric objects under severe occlusion. We explain our motivation and our approach in using the invariant nature of the curvature under similarity transform to estimate the fundamental angle of rotation and the centroid, two quantities that characterize the nature of rotationally symmetric objects. We introduce a cost function and a candidate selection strategy to identify appropriate choices for completing the missing regions of the occluded object. We demonstrate the usefulness of this symmetric shape completion algorithm in structure completion stage of global image inpainting algorithm with practical examples.

Chapter 4 discusses our proposed fast and efficient object based video inpainting algorithm. We explain the design of this modular approach which offers a unified framework to address inpainting under both static and limited moving camera conditions. Our key contribution is a computationally-efficient object-based inpainting algorithm capable of inpainting partially- and completely-occluded objects and providing global motion consistency by using sliding-window registration and dynamic programming. We demonstrate the effectiveness of our algorithm by inpainting different human subjects of varying pose, size and motion in video sequences captured through both static and moving cameras.

Chapter 5 is devoted to explaining our human subjective experiments and analysis of perceptual quality of image inpainting algorithm. Using extensive eye-tracking experiments, we show that there is a strong correlation between inpainting quality and visual attention. We also introduce a novel ranking mechanism to rate the subjective quality of inpainted images in those experiments. By comparing human attention in the form of gaze, within and outside the hole regions of inpainted images, we show that discernible artifacts due to inpainting attract an unusual amount of visual attention and corroborate well with subjective rankings.

Chapter 6 summarizes the results in this dissertation along with suggestions for improvements for future work.

Chapter 2

Literature Review

In this section we provide a detailed review on the different categories of image inpainting techniques and explain their approach. We then discuss the various algorithms proposed for video inpainting applications. This is followed by a discussion on relevant work on evaluating the perceptual quality of image inpainting algorithms.

2.1 Digital Image Inpainting

Presently there are different approaches to digital image inpainting and can be broadly classified into several different categories as listed below

1. Texture synthesis based inpainting
2. Partial Differential Equation (PDE) based inpainting
3. Exemplar and search based inpainting
4. Hybrid inpainting
5. Semi-automatic and Fast Digital Inpainting.

2.1.1 Texture synthesis based inpainting

One of the earliest modes of image inpainting was to use general texture synthesis algorithms to complete the missing regions. The texture synthesis algorithms synthesize new image pixels from an initial seed and strive to preserve the local structures of the image. Earlier inpainting techniques utilized these methods to fill the holes by sampling and copying pixels from neighboring areas [23–28]. For example, in [23], Markov Random Field (MRF) is used to model the local distribution of a pixel and new texture is synthesized by querying existing texture and finding all similar neighborhoods. Their differences lay mainly in how continuity is maintained between the inpainted hole and the existing pixels. These synthesis based techniques perform well only for a select set of images where completing the hole region with homogenous texture information would result in a natural completion.

Later this effort was extended to a fast synthesizing algorithm [24] by stitching together small patches of existing images referred to as *image quilting*. Heeger and Bergen developed a parametric texture synthesis algorithm which can synthesize a *matching* texture, given a *target* texture [26]. This was done by matching first order statistics of a linear filter bank which roughly match to the texture discrimination capabilities of Human Visual System (HVS). Igehy et.al included a composition step to the above method to generate synthetic and real textures [28]. A multi-resolution texture synthesis method which can generate texture under varying brightness conditions was introduced for inpainting by Yamauchi et.al [29]. Recently, a fast multi-resolution based image completion based on texture analysis and synthesis was introduced by Fang et.al [30]. In their method, the input image was analyzed by a patch based method using Principal Component Analysis (PCA) and a Vector Quantization (VQ) based technique was used to speedup the matching process of the texture inside the hole region. Various texture synthesis methods discussed here differentiate among themselves in their ability to create textures with different statistical characteristics and to generate textures under gradient, color or intensity variations. There are innumerable texture synthesis methods other than the aforementioned, but we shall restrict ourselves to illustrate those texture synthesis techniques specifically used for inpainting. While the texture synthesis based inpainting perform well in approximating textures, they have difficulty in handling natural images as they are composed of structures in the form of edges and have complex interactions between structure and texture boundaries. In some cases, they also require the user to specify what texture to replace and the place to be replaced. Hence while appreciating the use of texture synthesis techniques in inpainting, it is prudent to understand that these methods address only a small subset of inpainting issues and are not suitable for a wide variety of applications.

2.1.2 PDE based inpainting

A Partial Differential Equation (PDE) based iterative algorithm proposed by Bertalmio et.al [3] paved the way for modern digital image inpainting. The result of applying this algorithm to example images is shown in Figure 1.2. Borrowing heavily from the idea of manual inpainting, this iterative process propagates linear structures (edges) of the surrounding area also called *Isophotes*, into the hole region denoted by Ω, using a diffusion process given by.

$$\mathbf{I}^{n+1}(i,j) = \mathbf{I}^{n}(i,j) + \triangle t \cdot \mathbf{I}_t^{n}(i,j), \forall (i,j) \epsilon \Omega \tag{2.1}$$

where n is the iteration time, (i,j) are pixel co-ordinates, $\triangle t$ is the rate of the change of inpainting, $\mathbf{I}_t^n(i,j)$ is the update factor on the image $\mathbf{I}^n(i,j)$.

The update factor in the above equation, is a smoothed image obtained by applying a Laplacian operation in the direction perpendicular to the gradient in an iterative fashion. The PDE form of this process is expressed a

$$\mathbf{I}_t = \nabla(\triangle \mathbf{I}).\nabla^{\perp}\mathbf{I} \tag{2.2}$$

where $\nabla^{\perp}\mathbf{I}$ is the isophote direction and $\nabla(\triangle \mathbf{I})$ is the Laplacian smoothness operation on the gradient. One of the main drawbacks of this technique is that it underperforms in the replication of large textured regions due to blurring artifact of the diffusion process and the lack of explicit treatment of the pixels on edges. Inspired by this work, Chan and Shen proposed the Total Variational (TV) inpainting model which uses Euler-Lagrange equation and anisotropic diffusion based on the strength of the isophotes [31]. Let D be the inpainting region and E be the adjoining region around the hole, the variational inpainting model finds a function u on the extended inpainting domain adjoining the hole boundary $E \cup D$, such that it minimizes a regularity functional $R(u)$ under the denoising constraint on E defined below:

$$R(u) = \int_{E \cup D} r(|\nabla u|)dxdy \tag{2.3}$$

where r is an appropriate real function which is nonnegative for nonnegative inputs. This technique performs reasonably well for small regions and noise removal applications but it neither connects broken edges nor creates texture patterns. The TV model was extended to Curvature Driven Diffusion model (CDD) in [32] which included the curvature information of the isophotes to handle the curved structures in a better manner. Tschumperle et.al [33] introduced another PDE based technique referred to as vector valued regularization under anisotropic diffusion framework. These algorithms were focused on maintaining the structure of the inpainting area and hence could not perform as well in texture filling due to blurring artifacts.

2.1.3 Exemplar and search based Inpainting

The exemplar based approaches constitute an important class of inpainting algorithms and have proved to be very effective. An algorithm for handling large fill areas which combines the use of texture synthesis and Isophote driven Inpainting by a priority based mechanism in a unified framework was proposed by Criminisi et.al [13]. In this algorithm the region filling order is determined by a priority based mechanism and is presented in Figure 2.1. Points which lie on the path of edges have a higher priority and hence are filled earlier than other pixels. Figure 2.1(b) shows a point P with high priority lying on the contour of the hole boundary.

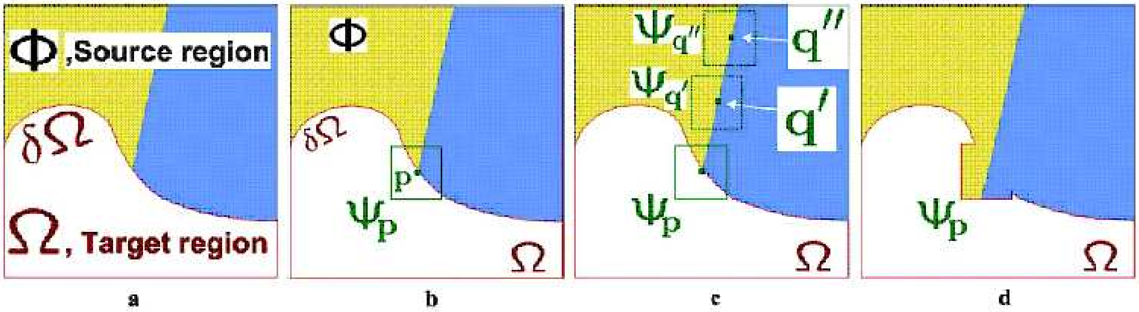


Figure 2.1: *Exemplar based inpainting: Adopted from [13] a) Original image with target, its contour and source region. b) The filled patch marked by highest priority pixel. c) Most likely candidates for filling the patch. d) Pixel with highest priority is filled with the matching patch.*

The neighborhood or filled patch surrounding the highest priority pixel is then filled by finding the best matching patch in the known regions as explained in Figures 2.1(c) and (d). The patch size can be varied depending on the underlying characteristics of the image. This exemplar-based removal technique performs well for a wide range of images with good texture and structure replication but has some difficulty in handling curved structures. A major drawback of this exemplar method is the bias caused by selection of few incorrect patches in the priority based filling mechanism. These incorrect patches have a tendency to hijack the entire inpainting process by building upon the initial incorrect completions and have a spiralling effect that undermines the stability of the inpainting process.

Drori et.al [34] present an algorithm which iteratively approximates the unknown regions and composites adaptive image fragments into the image. The visible parts of the image serve as a training set to infer the known parts and it performs image completion in a multi-resolution fashion by searching and filling patches at different resolutions and orientation. The computation time to fill the hole is quadratically related to the number of pixels and hence this method suffers from inordinate amount of delay as we increase the size of the hole region. By combining the texture synthesis based technique in [30] along with a directional measure, Fang et.al proposed an exemplar based method in [35]. The main advantage of this method is faster completion and use of multi-resolution based method to help in more natural texture synthesis.

While all exemplar based inpainting algorithms inherently use search, we explicitly denote techniques which search for matching textures and structures from images other than the given source as *search* based image completion. Until recently, image inpainting techniques operated under the assumption that the information necessary to complete the hole is to be fetched from the regions outside the hole region of the same image. While this approach has some merits, Hays et.al [1] argue that by using millions of images as the database, we are likely to have a more natural and semantically rich completion than that can be obtained from using just a single image. This work has opened a new set ideas for image inpainting and produced promising image completion results over a wide variety of images. In this method, Gist, an image descriptor which characterizes the image using information in multiple frequency bands and orientation, is computed for every image in a database of over millions [36]. The nearest semantic match for the image is obtained by searching through the entire database.

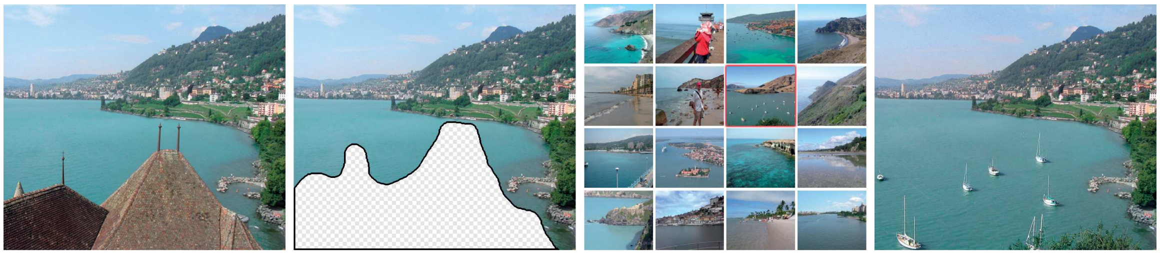


Figure 2.2: *Search based inpainting: Adopted from [1] a) Original image b) Image with a hole area marked by the user c) Results for the nearest semantic match using Gist of the image d) Completed image using Poisson blending*

Once the match is obtained, the region inside the matching image is seamlessly blended into the source image using a poisson blending process [37]. The result of applying this algorithm on a sample image with a large fill area is presented in Figure 2.2.

2.1.4 Hybrid digital inpainting

The hybrid approaches combine both texture synthesis and PDE based inpainting for completing the holes. The main idea behind these approaches is to decompose the image into separate structure and texture regions. The corresponding decomposed regions are filled by edge propagating algorithms and texture synthesis techniques respectively [38–40]. These algorithms are computationally intensive unless the fill region is small.

One important direction we believe is more natural to the inpainting process is by structure completion through segmentation. An example of such an approach can be found in [41]. It uses a two-step approach: the first stage is structure completion followed by texture synthesis. In the structure completion stage, a segmentation, using the algorithm of [42], is performed based on the insufficient geometry, color and texture information on the input and the partitioning boundaries are then extrapolated to generate a complete segmentation for the input using tensor voting [43]. The second step consists of synthesizing texture and color information in each segment, again using tensor voting.

2.1.5 Semi-automatic and fast digital Inpainting

Semi-automatic image inpainting with user assistance, in the form of guide lines to help in structure completion has found favor with researchers. The method by Jian et.al termed as inpainting with Structure propagation follows a two step process [44]. In the first step a user manually specifies important missing information in the hole by sketching object boundaries from the known to the unknown region and then a patch based texture synthesis is used to generate the texture. The missing image patches are synthesized along the user specified curves by formulating the problem as a global optimization problem under various structural and consistency constraints. Simple dynamic programming can be used to derive the optimal answer if only a single curve is present. For multiple objects, the optimization is great deal more difficult and the proposers approximated the answer by using belief propagation. Depending on the size of the inpainting area, all the methods discussed above take minutes to hours to complete and hence making it unacceptable for interactive user applications. To speed up the conventional image inpainting algorithms, a new class of fast inpainting techniques are being developed. Oliviera et.al proposed a fast digital Inpainting technique based on an isotropic diffusion model which performs inpainting by repeatedly convolving the inpainting region with a diffusion kernel [45]. A new method which treats the missing regions as level sets and uses Fast Marching Method (FMM) to propagate image information has been proposed by Telea [46]. These fast techniques are not suitable in filling large hole regions as they lack explicit methods to inpaint edge regions. As a result, they introduce blur artifacts in these areas making the inpainting unsatisfactory.

2.2 Digital Video Inpainting

Video inpainting started off as a natural extension of image inpainting algorithms and it has garnered a great deal of attention due to its potential applications in video error concealment in video transmission [47], multimedia editing and visualization [16], video stabilization [48] and new applications such as video modification for privacy protection [8, 9, 18]. A straightforward extension of image inpainting algorithms to video inpainting is to treat the underlying video data as a set of distinct images and apply image inpainting algorithms to them individually. This mode of operation does not take full advantage of the high temporal correlation that exists in video sequences and hence the quality of video inpainting across the frames are usually unsatisfactory. For example, one of the earliest efforts in extending the Partial Differential Equation (PDE) based image inpainting [3] to video was performed by Bertalmio et.al [15]. The focus of this method is to fill in the hole spatially by extending the edges and filling the hole with smoothed color information by a diffusion process using Navier-Stokes equation. It does not take into effect the temporal information available in the video and treats the video as individual images. Due to extensive smoothing, it does not reproduce the texture information and suffers from severe blurring artifacts. Consequently this method is effective only in restoring small scratches or spots occurring in archival footage. Cheung et.al introduced a space-time patch model based on probabilistic learning with applications to inpainting [49]. These condensed models called *epitomes* are learned by compiling large number of space-time patches drawn from input videos. Inpainting is treated as a reconstruction problem and the epitomes in this case are learned from the observed pixels. Inferring the missing pixels from the condensed epitomes leads to severe over-smoothing of the reconstructed pixels

A priority based exemplar approach proposed for image inpainting by Criminisi et.al in [13] was modified by Patwardhan et al. to video inpainting in [16, 50]. This method is capable of inpainting videos under a set of constrained camera motion. Initially, the input video is separated into background layer and foreground object layer utilizing the optical flow. Hole regions identified in the foreground layer are first inpainted by a priority-based exemplar process before proceeding to complete the damaged regions in the background layer. In this patch based exemplar method, damaged patches around the boundary of the hole are filled by a priority based mechanism. The appropriate candidates for filling the damaged areas are selected by minimizing a 5-dimensional distance metric based based on the the pixel color values and the optical flow vectors. While being effective in completing regions with sparsely distributed structural cues, this method, like its image inpainting counterpart, is susceptible to providing unstable inpainting due to few incorrect patches as a result of spurious local variations. This tendency to rapidly build upon few incorrect regions, especially, when completing regions with high structural variations makes the process unstable. More importantly, this technique cannot handle the case when a significant portion of the object is missing, and it has difficulty in inpainting curved structures.

A video completion scheme based on motion layer estimation followed by motion compensation and texture completion has been proposed [51]. After removing a particular motion layer, motion compensation is used to complete moving objects and non-parametric texture synthesis is used to complete the static background regions.The inpainted layers are then warped into every video frame to complete the holes. Video completion by motion field transfer – transfer of spatio-temporal patches of motion field instead of direct color sampling has been introduced recently [52]. This technique is extremely sensitive to noise as they involve local motion estimates by a derivative-based process. It has difficulty inpainting large motion as their motion estimation techniques focus solely on measuring small local movement. In addition, as the scheme transfers only motion information, it suffers from blurring artifact due to the use of a re-sampling process to estimate color information. A video completion algorithm for perspective camera under constrained motion has been proposed recently [53]. The foreground and background layers are separated and objects in foreground volume are rectified to compensate for perspective projection. The pixels in the foreground are completed by modeling it as a graph labeling problem as described in [44] and a dynamic programming is used to solve it.

Deviating from the patch-based methods discussed above, Jiaya et.al introduced an object-based inpainting system which utilizes a user-assisted segmentation to inpaint holes in foreground regions that exhibit cyclic motions [54]. To complete the missing foreground regions they explicitly estimate the periodicity of the moving foreground object and align them with the partially damaged pixels in the hole boundary to complete missing regions. Temporal consistency is achieved by a movel (moving pixel) wrapping and regularization process using tensor voting. A similar technique that utilizes mean shift tracking to limit the search space and nonparametric texture synthesis coupled with graph cuts has been proposed [55]. This method currently does not have a mechanism to handle moving cameras and also reports artifacts at the boundaries of the hole region.

2.3 Perceptual Evaluation of Image Inpainting Quality

While most of the research in digital inpainting has been focused on introducing new algorithms for specific applications, there exists relatively little research in evaluating what constitutes a perceptually good inpainting. A majority of the image inpainting techniques discussed in Section 2.1 have compared the performance of their algorithm with existing methods by highlighting their efficacy in terms of their capability to handle large fill areas, ability to inpaint curved structures and regions with high edge (structural) content, effectiveness of their texture replication capacity and the time taken for completion, in a non-descriptive manner with a limited set of images that appeared in initial work [3]. In these techniques, the objective of inpainting was to make the damaged image appear as close to the original image. Hence, most of them utilized the image information surrounding the hole regions from the *same image* to fill in the holes. In such restoration type applications, this form of comparison though limited in scope was reasonable. Also to make the inpainting appear imperceptible to the human observers, they also converted the RGB images into CIE LAB color space to exploit some aspects of perceptual uniformity [3,13]. In spite of the inherently subjective nature of the inpainting process, these methods measured their inpainting quality by using simple objective metrics such as Peak Signal to Noise Ratio (PSNR) and Mean Squared Error (MSE) of the inpainted images to the original image or by just visually comparing them [33]. Oliviera et.al [45] evaluate their performance by computing the MSE of the reconstructed region in RGB space with the original image. It has been widely reported that these objective metrics like MSE and Peak Signal to Noise Ratio (PSNR) do not perform a well in characterizing perceptual image quality [56].

One of the earlier work in analyzing the error in image inpainting with the shape of the domain of inpainting was performed by Chan. et.al [57]. In this important work, they show that the quality of digital image inpainting depends more on the shape of the image inpainting domain than the size or total area of the inpainting domain. In effect, they show that inpainting techniques that use PDE based inpainting techniques are effective in inpainting small or narrow smooth regions as they are based on smoothing. While this work raised an important issue on impact of the shape of the inpainting domain on the inpainting quality, it does not address the issue of what constitutes a perceptually good inpainting technique. For example, from the figure in 2.3, humans can readily observe that the first figure in the bottom row has a better perceptual quality than the second. This inference on inpainting quality is made by the HVS based on a combination of multiple hierarchical process based on low, mid and higher level vision and cognition process. It would be great advantage if we could identify those features that impact the inpainting so that we may be able to develop an objective model to estimate the perceptual quality.

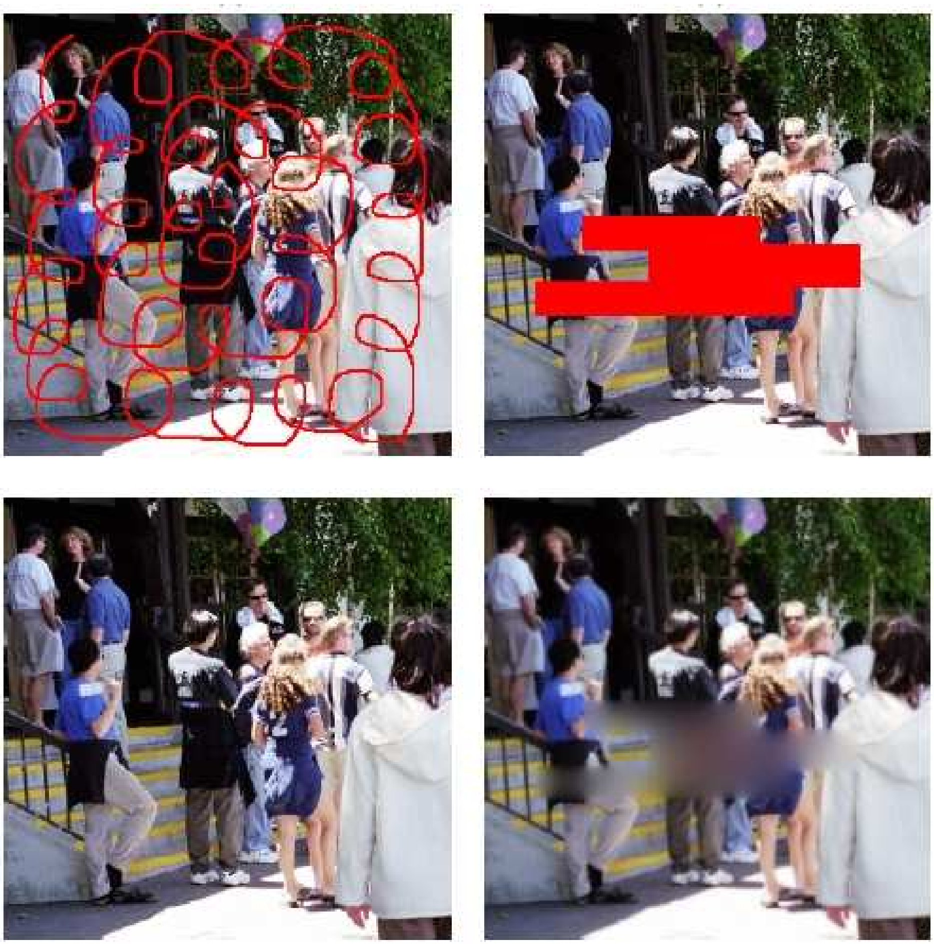


Figure 2.3: *Inpainting quality based on domain shape: Adopted from [57] Top Row) Same images with different inpainting domains Bottom Row) Corresponding inpainted images where the first one has a better quality than the second*

As mentioned previously, the domain of image inpainting has expanded to include advanced editing of multimedia objects which require larger regions to be inpainted. To tackle these challenges, alternate paradigms of image inpainting algorithms have been proposed in which *millions of images* from a source database, as opposed to a single image, are used to find appropriate image information to complete the missing information [1]. This recent approach has already shown considerable promise by its ability to address difficult inpainting conditions via a more semantic completion. Subsequently, the objective of the new approach has expanded from making the inpainted image to look as close as original to making the inpainted modifications difficult to discern by human observers. This implicit difference brings in significant change in inpainting capabilities and demands more advanced techniques to evaluate their perceptual efficiency by taking into account the subjective nature of HVS. In their work, Hays et.al conduct a human subjective experiment to rate the quality of inpainting and based on that rating they conclude that their method is perceptually more efficient when compared to exemplar based technique introduced by Criminisi et.al [13]. To quantitatively evaluate the performance of inpainting quality, they conducted an experiment in which 20 naive observers were instructed to identify fake images distributed among unmodified real images, images inpainted by both [1] and [13] by presenting them in a random order. In that experiment, after ten seconds of visual examination, 34% of the images modified by their algorithm, 69% of images modified by Criminisi et al. and 3% of real photographs respectively were marked as fake by the participants. Under unlimited time constraints they also obtained similar performance. Based on that, they conclude that the performance of their inpainting algorithm is better when compared with method in [13]. While this experiment provides interesting outcome, the scope is limited to evaluating which algorithm is better and does not provide useful information on the underlying reason behind the choice of the testers.

Very recently, Ardis et.al [19] analyze the visual salience map generated by a computational vision model [58] and relate it to the perceptual quality of image inpainting. In analyzing smaller inpainted regions, they show that the visual salience maps can be used to relate the perceptual quality to the human subjective preference. Further experiments are needed to confirm the findings of this study as it was conducted with a limited set of 4 images and a total of 5 human subjects. Moreover, detailed studies are required to understand the capability of these generalized human computational visual models to estimate the quality as related to inpainting.

Chapter 3

Symmetry Completion and Global Image Inpainting

Symmetry is one of the most important pervasive cues that can be observed in most of natural as well as man made environments. Almost all of the objects created by human beings exhibit some form of symmetry, few examples include periodic texture patterns on a carpet and modern architectural buildings. Even naturally occurring phenomenon such as the gait of the human beings and the appearance of human face possess an inherent symmetry and has been utilized as a biometric in human identification, classification and face recognition applications. Thus the concept of symmetry due to its widespread prevalence and its invariant nature has been extensively studied and thoroughly analyzed. Completion of the missing structure by utilizing the mid-level image attributes to obtain a global image inpainting was identified as one of the major research goals in Section 1.2.1. Building on that idea, we propose a novel structure completion algorithm to complete the missing boundaries of rotationally symmetric object under severe occlusion by utilizing the inherent symmetry. The intuitive idea is to use the existing contour, under a carefully estimated similarity transform, to fill the missing portion of a symmetric object under occlusion. This algorithm exploits the invariant nature of the curvature under similarity transform and the periodicity of the curvature of symmetric object contour. To arrive at the appropriate transform, we first estimate the fundamental period in the curvature. We use the fundamental period and the harmonic components to estimate the fundamental angle of rotation and the centroid of the unoccluded shape, which establishes different modes of symmetry. By following each mode of symmetry we compute the corresponding transform and select the ones that best complete the missing portion of the contour. We begin by introducing mathematical aspects of symmetry, define the problem setting and elaborate the challenges posed by occluded symmetry and the approach taken to solve the problem in Section 3.1. Following this, in Section 3.2, we discuss the computational details and the formulations involved in the estimation of the fundamental angle of rotation and centroid of occluded symmetrical objects. Section 3.3 describes a cost function which is used to select an appropriate candidate from among several options. In the final section, with suitable examples, we demonstrate that this technique can serve as an effective global structure completion strategy in the context of image inpainting.

3.1 Introduction to symmetry

Due to its widespread prevalence, the concept of symmetry has attracted considerable attention and much research efforts have been devoted to analyze and quantify the properties of symmetric structures [59]. For example, many diverse applications such as object identification and recognition in machine vision [60], detection, classification and recognition of human beings in video surveillance applications [61], segmentation in images [62,63], and digital inpainting have utilized various computational aspects of symmetry. Despite the diverse nature of the many applications listed above, the use of computational aspects of symmetry remain as the common thread linking those diverse applications. In the next section, we provide an introductory mathematical background on symmetry pertinent to this study. Detailed treatment on the formal aspects of symmetry based on group theory, various aspects of symmetry groups, different symmetries and different types of symmetries groups on a plane and their properties can be found in [64–69].

3.1.1 Mathematical Introduction to Symmetry

Let us denote R^2 to be the two dimensional real Euclidean space and associate with each point in the space an ordered pair $\{(x_i, y_i)\}$ of real numbers specifying its position in the coordinate axes. An object or an element E in this space is represented by a set of ordered pairs $\{(x_1, y_1), (x_1, y_1), \ldots (x_n, y_n)\}$ denoting the coordinates of the object in the space. A transformation or homomorphism ϕ is a one-to-one correspondence function that maps every point on the space into another point. An object is said to possess symmetry if it remains invariant to certain class of transformations. A symmetry group is a collection of transformations defined on the set G, that satisfy the following properties of the abstract group under the composition of transformations. They are,

1. **Closure:** If $\phi, \varphi \in G$, then the product $\phi\varphi$ is also in group G.

2. **Associativity:** The identity $(\phi\varphi)\omega = \phi(\varphi\omega)$ is satisfied for all elements of G.

3. **Identity:** There exists an element represented as I such that $I\phi = \phi I$ for all $\phi \in G$.

4. **Inverse:** There exists an element $\psi = \phi^{-1}$ for every element in G such that $\phi\phi^{-1} = \phi^{-1}\phi = I$.

In planar Euclidean geometry, the fundamental transformations that preserve symmetry are the distance preserving transforms called as *isometries* or *congruence transformation* which form a group. A pattern $P \in R^2$ can be transformed into a regular periodic structure having symmetry by using only four basic planar isometries. These four fundamental isometries that characterize the entire symmetry groups in the planar Euclidean space are translation, rotation, reflection and glide reflection. For a given pattern P, we represent the set $S(P)$ to denote the set of symmetries possessed by P and they constitute an algebraic group G. The number of symmetries in $S(P)$ is defined as the *order* of the group. Two dimensional symmetric groups are classified into three different categories known as point, line and plane groups. The pattern or the motif belonging to a particular group exhibits symmetry around a point, line or a plane respectively. Under the point groups, one major type of symmetry group is the Rotational symmetry.

Rotational Symmetry: We define the *Rotational* symmetry of order n by C_n with $n \geq 2$ if it is invariant to rotation of $\frac{2\pi}{n}$ radians about a fixed point on a plane. A rotationally symmetric object of order 8 denoted by C_8 is shown in Figure 3.1.

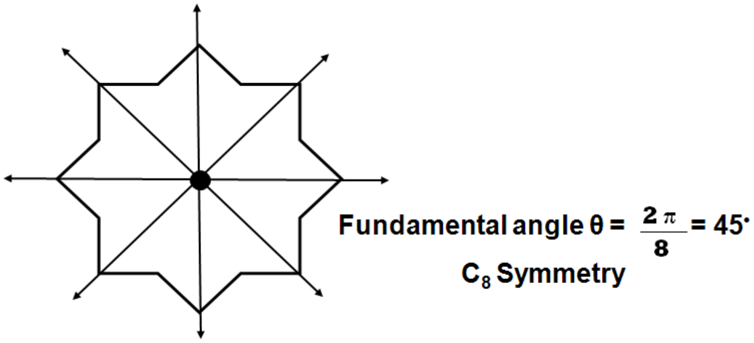


Figure 3.1: An object with C_8 symmetry and its centroid

The rotational symmetry can be completely characterized two quantities namely

1. **Fundamental angle of Rotation**: The angle by which, when the object undergoes rotation about its centroid, its shape remains unchanged.
2. **Centroid**: The center of mass of the object.

3.1.2 Challenges in Occluded Symmetry: Problem Definition and Approach

Consider the partially occluded equilateral hexagon. An intuitive way to complete the missing region is to rotate and translate the original contour around the centroid of the unoccluded shape so as to match the missing portion and form a symmetric hexagon. There are two things that need to be done before we can do this process. One is to estimate the centroid of the object and the other is to estimate the angle by which we rotate the object. Estimating both these attributes is non-trivial because of the fact that under severe occlusions, the centroid of the occluded object can be far away from that of the unoccluded object [70] as shown in Figure 3.2 and we do not know the a priori fundamental angle of rotation.

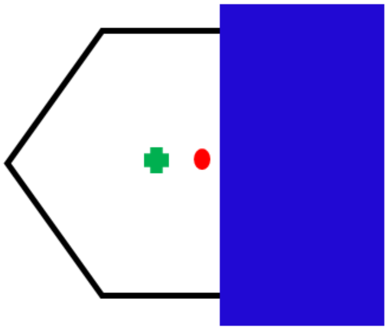


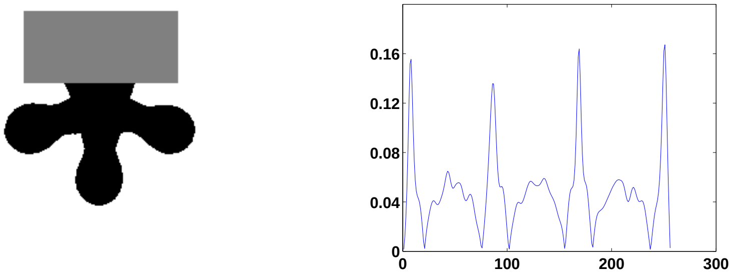
Figure 3.2: Error in centroid estimation: Centroid of occluded hexagon denoted in green is very far away from the original unoccluded hexagon in Red.

To define our problem in a concise manner, we aim to complete the missing regions of a rotationally symmetric object under severe occlusions. At this juncture, we can readily observe that one of the important applications of contour completion is in structure completion aspect of image inpainting. Using object symmetry to complete occluded or missing object contour is a relatively unexplored area in computer vision. In [70], Zabrodsky et al. describe various symmetry structures and define a continuous symmetry measure referred to as "symmetry distance" for evaluating different types of symmetry. They use this distance measure to reconstruct the symmetric shape similar to the original occluded contour. Nonetheless, their approach requires an a-priori determined order of rotational symmetry for completing the missing structure. We, on the other hand, follow a different path and exploit the invariant nature of the curvature to complete the missing structure. The intuition behind our approach in solving this problem can be explained by first observing the following facts,

1. It can be observed from Figure 3.3 that the curvature of the rotationally symmetric object is periodic in nature and that the curvature is invariant to rotations.

2. Rotation of the object by the fundamental angle of rotation θ about the centroid is equivalent to translation of the curvature by its period T.

Taking note of these facts, we follow a two stage approach to solve the problem. In the first stage we compute the curvature and estimate the period of the curvature. The estimated period of the curvature helps us to establish correspondence between similar points on the contour. In the next stage, by using the correspondence established, we estimate the fundamental angle of rotation and centroid. These two attributes completely characterize the rotationally symmetric object and helps us to complete the missing regions of the occluded object.



(a) (b)

Figure 3.3: Invariant nature of curvature of rotationally symmetric object: (a) An occluded rotationally symmetric object (b) Periodic nature of curvature of the object.

3.2 Methodology

In the following sections, we describe the mathematical aspects of curvature computation and the estimation of its periodicity. By utilizing the invariant nature of the curvature we establish correspondence between points in the contour. We explicitly use these correspondences to formulate a system of equations to estimate both the fundamental angle of ration and the centroid of the unoccluded shape.

3.2.1 Curvature Computation and Periodicity Estimation

We treat the input curve as an open contour which is represented as a sequence of n points (x_1, y_1), (x_2, y_2), $\ldots$, (x_n, y_n) following a particular orientation. The x and y coordinates of the pixels are parameterized by the curve arc-length parameter u, and u is normalized to take values from the interval $[0, 1]$. The functions $x(u)$ and $y(u)$ are then resampled to N equidistant points using a cubic spline interpolation. We use $N = 256$ which is found to be reasonable for typical image processing applications. The resampled function $x(t)$ and $y(t)$ is low-pass filtered using a normative Gaussian filter to obtain a smoothed contour. We then compute the curvature of the contour as follows:

$$\kappa(t) = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{3/2}} \tag{3.1}$$

where the dots indicate differentiation with respect to t and the discrete parametrization of the contour is $\{(x(t), y(t))\}$ where $t = 0, 1, \ldots, N-1$. The computed curvature curve is shown in Figure 3.4(b). We also compute the normal vector $\mathbf{n}(t)$ at each point on the curve as follows:

$$\mathbf{n}(t) = \frac{\mathbf{e}(t)}{||\mathbf{e}(t)||} \text{ where } \mathbf{e}(t) = (\ddot{x}, \ddot{y}) - \frac{(\ddot{x}\dot{x} + \ddot{y}\dot{y})}{(\dot{x}^2 + \dot{y}^2)}(\dot{x}, \dot{y}) \tag{3.2}$$

The normal vectors will later be used to compute the *fundamental angle of rotation* – the smallest angle of rotation of the unoccluded object about its centroid so that it returns to its original position. Using the arc-length parametrization, it can be easily shown that *a rotation about the centroid of the unoccluded object* manifests as a translation of the curvature curve [71]. Since the contour realigns itself after rotating an integral number of the fundamental angle, the curvature curve of a rotationally symmetric contour must be periodic. We also note that, the periodicity of the curvature is more apparent in the autocorrelation as observed in Figure 3.4(c). The distance between the peaks in the autocorrelation function of curvature correspond to the translations needed by the existing contour points to maintain the curve structure. Estimating this periodicity provides us with correspondence between among the existing contour points.

To estimate the periodicity of the autocorrelation function of the curvature, we assume that the visible contour contain at least two periods, otherwise the period cannot be estimated. To robustly estimate this period T of the curvature curve, we employ a sliding-window based technique. First, we select a $N/2$-point *search segment* from the curvature curve at a random starting point $t = q$ to $t = N/2+q-1$. Second, among all the $N/2$-point segments from the curvature curve, we identify the segment $\hat{\tau}$ points away from the search segment that maximizes the autocorrelation:

$$\hat{\tau} = \max_{\tau \in S} \sum_{t=q}^{N/2+q-1} \kappa(t)\kappa(t+\tau) \tag{3.3}$$

where $S = \{-q, \dots, q-1, q+1, \dots, N/2-q\}$. $\hat{\tau}$ must be in the form of kT where k is a positive integer. As neighboring structures tend to be more correlated than their distant counterparts, k is typically 1. To ensure a robust estimate, we randomly select multiple search segments and estimate T based on the smallest computed $\hat{\tau}$. We then identify all pairs of curvature points that are an integral number of T from each other. Let the number of correspondence be M. Each correspondence $(x(t_i), y(t_i)) \leftrightarrow (x(t_i + k_iT), y(t_i + k_iT))$ for $i = 0, 1, \dots, M-1$ is parameterized by the index t_i of the first point and the number of period k_i the second point from the first.

3.2.2 Estimation of Fundamental Angle of Rotation

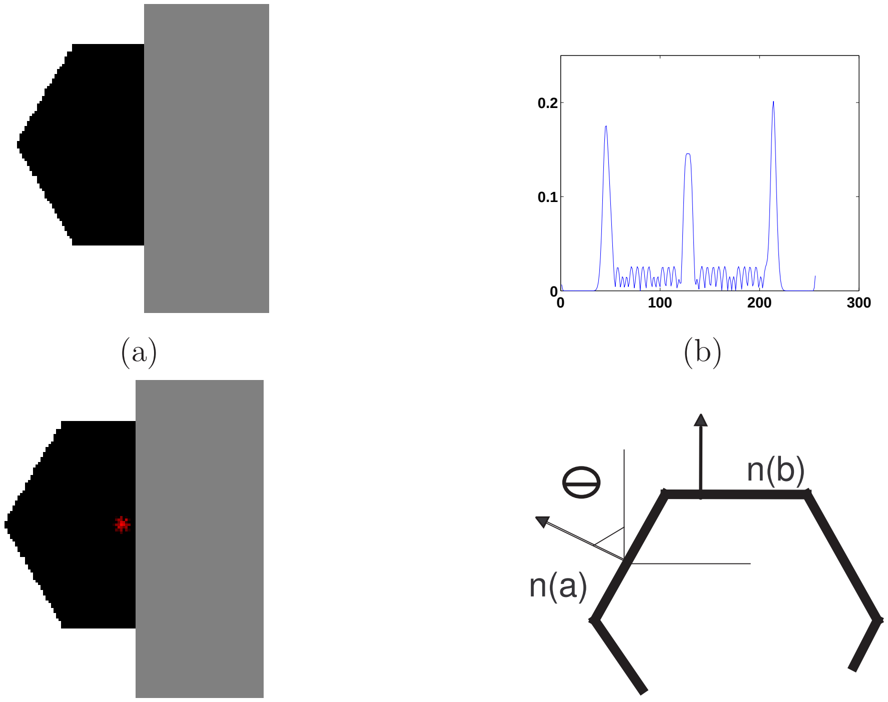
After the correspondences are established, we can estimate the fundamental angle of rotation θ, which is needed to perform the rotation for the Euclidean transform. It is calculated by computing the angle between the normal vectors of the corresponding points in the original contour. We compute the angle between the normal vectors of all the corresponding points and take the average value as the estimate:

$$\theta = \frac{1}{M} \sum_{i=0}^{M-1} \frac{1}{k_i} \cos^{-1}\left(\frac{\mathbf{n}(t_i) \cdot \mathbf{n}(t_i + k_iT)}{|\mathbf{n}(t_i)||\mathbf{n}(t_i + k_iT)|}\right) \tag{3.4}$$

The above process is explained in Figure 3.4(d) where we show normal vectors of the two corresponding points $\mathbf{n}(a)$ and $\mathbf{n}(b)$ separated by the fundamental period. Due to the constraint of the rotational symmetry imposed by the symmetry group, the smallest angle of rotation θ must be of the form

$$\theta = \frac{360}{n}, \in 2, 3...K \tag{3.5}$$

We use this constraint to further refine our estimation of the angle of rotation.



(a) (b)

(c) (d)

Figure 3.4: Occluded symmetric object, curvature and surface normals: (a) Symmetric hexagon with occlusion; (b) Curvature of the contour of the occluded hexagon; (c) Hexagon with the estimated centroid; (d) Normal vectors of corresponding points $n_x(a)$ and $n_x(b)$ on the contour.

3.2.3 Robust Centroid Estimation

We now have the fundamental rotation angle θ, and a set of point correspondences $(x(t_i), y(t_i)) \leftrightarrow (x(t_i + k_iT), y(t_i + k_iT))$ for $i = 0, 1, \ldots, M-1$. For each correspondence, there exists a rotation transformation matrix M_{k_i} such that

$$M_{k_i} \begin{pmatrix} x(t_i) \\ y(t_i) \\ 1 \end{pmatrix} = \begin{pmatrix} x(t_i + k_iT) \\ y(t_i + k_iT) \end{pmatrix} \tag{3.6}$$

M_{k_i} is given by

$$M_{k_i} = \begin{pmatrix} \cos(k_i\theta) & \sin(k_i\theta) & T_x^{k_i} \\ -\sin(k_i\theta) & \cos(k_i\theta) & T_y^{k_i} \end{pmatrix} \tag{3.7}$$

where $T_x^{k_i}$ and $T_y^{k_i}$ are translations in x and y directions.

We also note that the centroid (C_x, C_y) is the only point inside the contour which does not rotate or translate under different angles of rotation. In other words, it is the pivot around which the rotation and translation of the existing contour structure takes place and it in itself remains fixed. Since the centroid of the unoccluded shape is the center of rotation, it is a fixed point of M_{k_i} for any integer k_i. If the coordinates of the centroid is (C_x, C_y), we must have

$$M_{k_i} \begin{pmatrix} C_x \\ C_y \\ 1 \end{pmatrix} = \begin{pmatrix} C_x \\ C_y \end{pmatrix} \tag{3.8}$$

Combining equations (3.7) and (3.8), we can eliminate the translation parameters and rewrite Equation (3.6) as follows:

$$\begin{pmatrix} (1 - \cos(k_i\theta)) & -\sin(k_i\theta) \\ (\cos(k_i\theta)) & 1 - \sin(k_i\theta) \end{pmatrix} \begin{pmatrix} C_x \\ C_y \end{pmatrix} = \\ \begin{pmatrix} x(t_i + k_iT) - x(t_i)\cos(k_i\theta) - y(t_i)\sin(k_i\theta) \\ y(t_i + k_iT) + x(t_i)\sin(k_i\theta) - y(t_i)\cos(k_i\theta) \end{pmatrix} \tag{3.9}$$

As we have estimated θ in the previous section, we can formulate a system of equations based on (3.9) for all M correspondences established, $(X_i \rightarrow X_i^{'}, Y_i \rightarrow Y^{'}, k)$, where $i = 1...M$, we have

$$\begin{pmatrix} (1-\cos(k_1\theta)) & -\sin(k_1\theta) \\ (1+\sin(k_1\theta)) & -\cos(k_1\theta) \\ (1-\cos(k_2\theta)) & -\sin(k_2\theta) \\ (1+\sin(k_2\theta)) & -\cos(k_2\theta) \\ (1-\cos(k_M\theta)) & -\sin(k_3\theta) \\ (1+\sin(k_M\theta)) & -\cos(k_3\theta) \end{pmatrix} \begin{pmatrix} C_x \\ C_y \end{pmatrix} = \begin{pmatrix} X_1^{'} - X_1\cos(k_1\theta) - Y_1\sin(k_1\theta) \\ Y_1^{'} + X_1\sin(k_1\theta) - Y_1\cos(k_1\theta) \\ X_2^{'} - X_2\cos(k_2\theta) - Y_2\sin(k_1\theta) \\ Y_2^{'} + X_2\sin(k_2\theta) - Y_2\cos(k_1\theta) \\ X_M^{'} - X_M\cos(k_2\theta) - Y_M\sin(k_1\theta) \\ X_M^{'} + X_M\sin(k_2\theta) - Y_M\cos(k_1\theta) \end{pmatrix} \tag{3.10}$$

Solving this system of equations by taking pseudo inverse, provides us with a least squares estimate on the centroid of the partially visible contour. We then proceed to rotate the existing contour structure using the fundamental and integer multiples of the rotation angle to complete the missing regions of the symmetric structure.

3.3 Cost Function and Candidate Selection

The final stage of this algorithm involves selecting a suitable candidate from the set of rotations about the centroid to extrapolate the missing contour. Let $\{(\tilde{x}(t), \tilde{y}(t)), t = 0, 1, \ldots, N-1\}$ be a rotated contour. One end of the rotated contour will align with the original one, while the other end will extrapolate into the missing region and possibly connect back to the opposite end of the original contour. Assume the indices of the extrapolated portion, in reverse order, are $N-1, N-2, \ldots$ and so forth. We use the following cost function to measure how well the extrapolated contour aligns with the unmatched end of the original contour:

$$C(\{\tilde{x}, \tilde{y}\}) = \min_{0 \leq k \leq N-1} \frac{1}{k+1} \cdot \sum_{t=0}^{k} [\tilde{x}(N-1-t) - x(k-t)]^2 + [\tilde{y}(N-1-t) - y(k-t)]^2 \tag{3.11}$$

Intuitively, this cost function measures the distance between the extrapolated re-

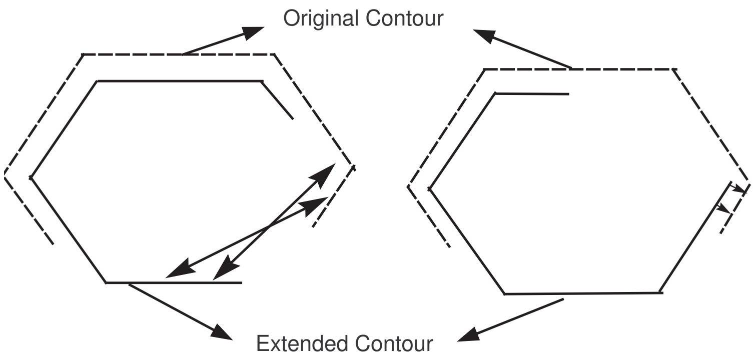


Figure 3.5: Illustration of cost function: The two figures show the alignment of two candidate contours with the original one.

gion of the transformed contour with the opposite end of the original contour segment. The search process is illustrated in the Figure 3.5.

In the case of the occluded hexagon, the cost function is computed for all valid candidates, and the suitable candidate is chosen to be the one with the minimal cost. Figure 3.6 (a)-(c) shows the completion of the occluded hexagon using the first three harmonics. The associated cost for them are 6823.1, 1606.3 and 50.78. Thus, the third harmonic provides the best transformed contour for the completion and the associated cost function is shown in Figure3.6d . If the missing region is too large, it is straightforward to repeat the above process to complete the entire region in a piecemeal fashion.

We illustrate the robustness of our algorithm by showing another example for occluded object contour completion shown in Figure 3.7(a). We can find that the final completion of the occluded contour is arrived by using the second harmonic and the minimum cost as explained previously.

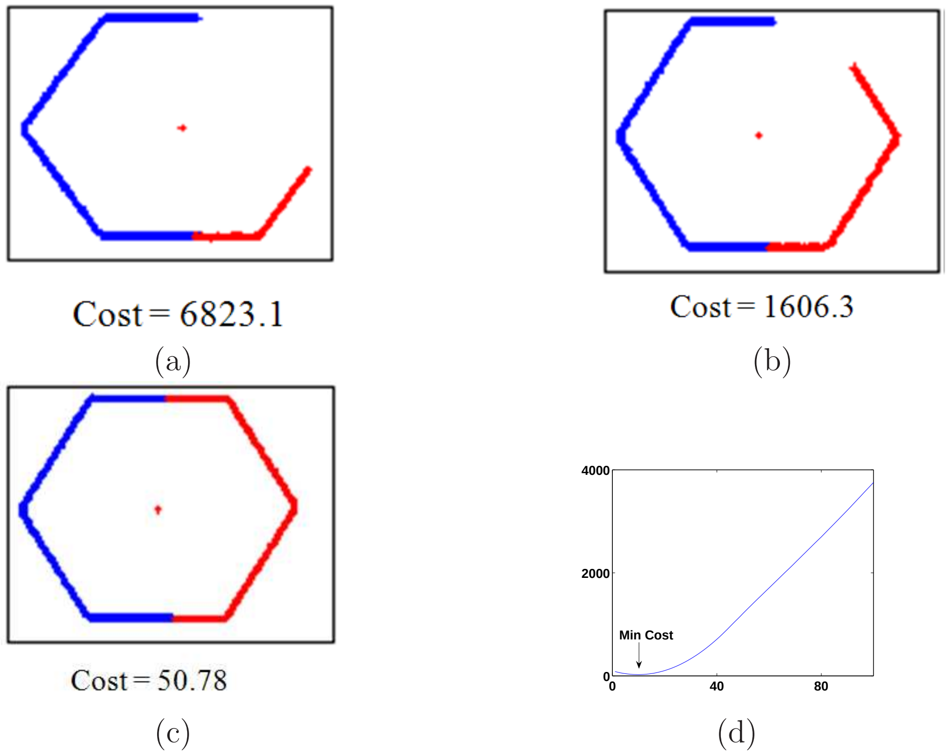
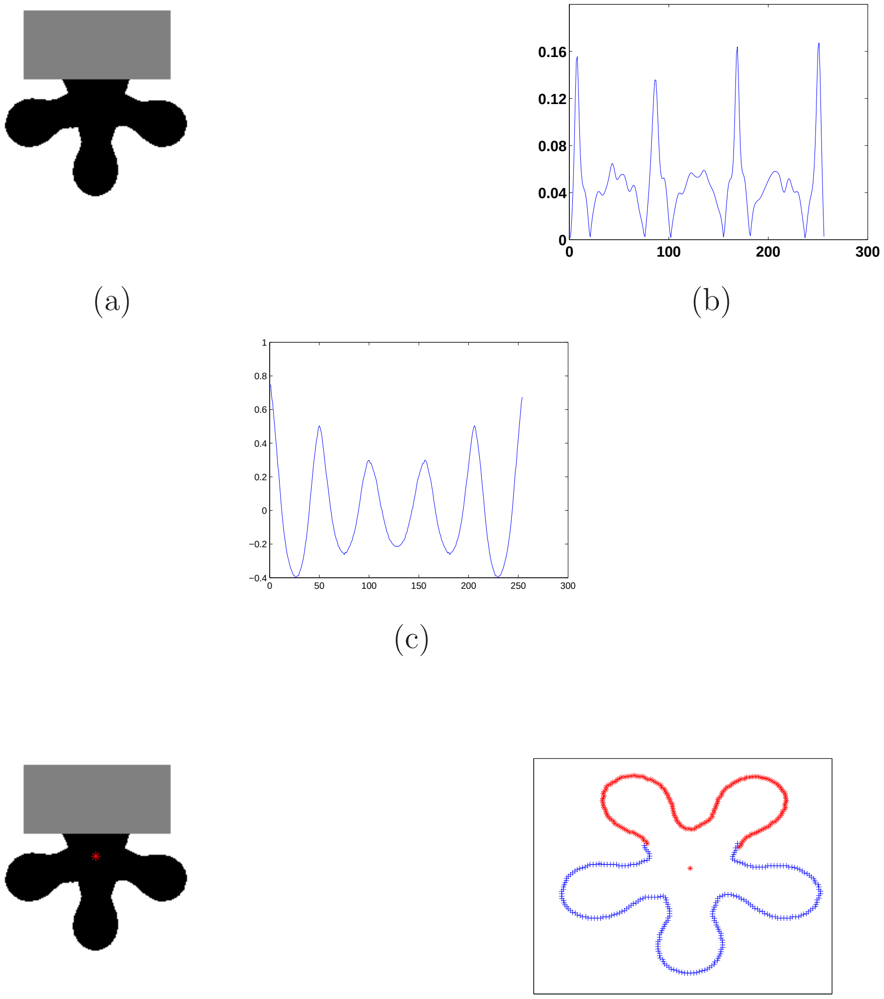


Figure 3.6: Appropriate candidate selection using cost function: (a) Completion (red) due to rotating the contour by one fundamental period (first harmonic); (b) Completion due to the second harmonic; (c) Completion due to the third harmonic that results in the lowest cost.(d) Cost function corresponding to the third harmonic.

3.4 Application to Global Image Inpainting

Earlier we described how missing contour completion could be an effective tool in performing structural completion task of image inpainting. Here we present a couple of real world examples which illustrate how occluded symmetry completion algorithm proposed in this section can be effectively utilized in structure completion aspect. Figure 3.8(a) shows an image with a hole. Firstly, we segment the image, extract the outer contour and we compute the curvature which is shown in Figure 3.8(b). We then proceed to estimate the period of the curvature and estimate the centroid shown in Figure 3.8(c). Finally we select a suitable candidate from a finite set of available candidates by minimizing the cost function defined in Equation 4.9. The final result



(a)

(b)

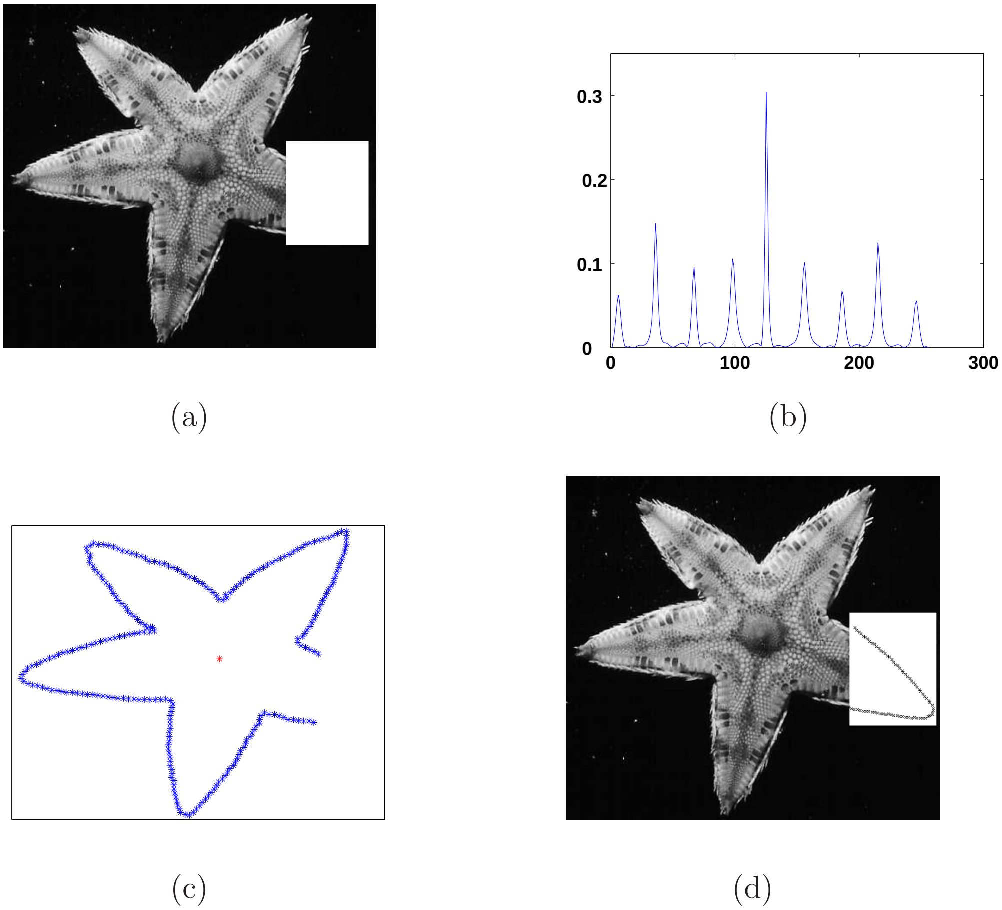
(c)

(c)

(d)

Figure 3.7: Occluded symmetric shape completion: (a) Symmetric object under occlusion; (b) Curvature of the contour of the object; (c) Autocorrelation of the Curvature of the object; (d) Symmetric object with estimated center; (e) Completed symmetry corresponding to minimal cost using 2nd harmonic.

of the occlusion completion is shown in Figure 3.8(d). It is clear that the structure of the occluded region is reconstructed in a perceptually consistent manner. Following the same vein, we present another example in which the occlusion is more than 40% of the entire structure.



(a) (b)

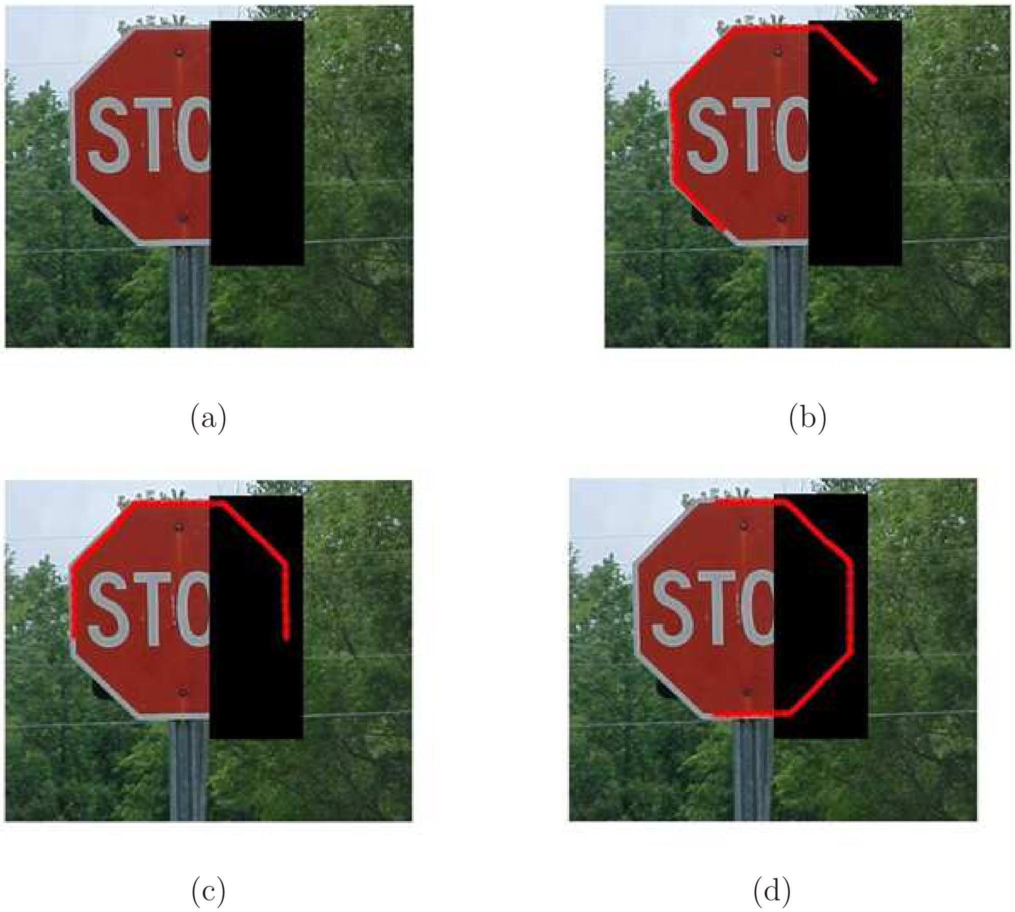
(c) (d)

Figure 3.8: Structure completion and global image inpainting: (a) An image with the hole; (b) Curvature of the contour of the segmented object; (c) Contour with estimated center; (d) Structure completion result corresponding to first period and minimal cost

From these examples we can see that once the underlying structure is completed, we can then utilize effective texture synthesis techniques to fill in the texture details inside the closed boundary to complete the inpainting process.

3.5 Summary

In this chapter we have proposed a shape completion algorithm for rotationally symmetrical objects under the presence of severe occlusions. This algorithm does not make any assumption about the nature of the circular symmetries to perform



(a) (b)

(c) (d)

Figure 3.9: Structure completion and global image inpainting: (a) An image with the hole; (b) Completion using First harmonic; (c) Completion by second harmonic; (d) Final completion due to third harmonic and corresponding to minimal cost

the completion and can potentially be extended to handle various symmetries. The usefulness of this contour completion algorithm is demonstrated in a global inpainting technique by using it for structure completion process. This technique can also be used to complete periodic structures of arbitrary lengths by repeatedly extending the matching segments obtained from the correlation process until it satisfies a minimum cost criterion.

Chapter 4

Efficient Object Based Video Inpainting

We presented an overview of existing video inpainting techniques in Section 2.2 and in section 1.2.2 we identified the key challenges in video inpainting area which would expand the usefulness of these algorithms in practical applications. In this chapter, we describe our proposed fast and efficient object based video inpainting technique which addresses those issues. Our modular approach utilizes a two stage process in which the input video is segmented into constituent object followed by inpainting of identified regions. The static regions to be inpainted are completed by a combination of adaptive background replacement and image inpainting technique. Our key contribution is the introduction of a computationally efficient object based inpainting method for inpainting moving foreground regions. By grouping consecutive object templates in a sliding window fashion and by defining a novel dissimilarity metric, we propose to solve the video inpainting as a minimization problem under a dynamic programming framework. This technique can effectively inpaint large regions of occlusions, inpaint objects that are completely missing for several frames, change in size and pose and has minimal blurring and motion artifacts. Furthermore, our object-based approach is significantly faster than the patch-based scheme and takes minutes, rather than hours as required by many existing schemes, to obtain comparable inpainting results. Our proposed scheme also offers a unified framework to address video inpainting under both static and limited moving camera conditions.

The rest of this chapter is organized as follows: In the opening Section 4.1 we briefly present our modular approach and summarize the key contributions of our approach. We present an overview of our system in Section 4.2, and discuss the technical details in segmenting the video into constituent objects Section 4.3. After segmentation, we briefly explain static background inpainting in Section 4.4 followed by a detailed discussion of our moving foreground inpainting framework in Section 4.5. We demonstrate the results of our method under different operating conditions in Section 4.6 and analyze the impact of segmentation on the inpainting process in Section 4.7, present a complexity analysis with existing techniques in Section 4.8 and summarize our contribution in Section 4.9.

4.1 Approach and Key Contributions

A visible trend emerging from recent works is the use of a two-stage process: segmentation of the video into different regions followed by separate completion of the respective regions.Existing algorithms differ by the method they use to complete hole regions in the moving foreground. It is often the most challenging and time consuming step due to the extensive search involved. Algorithms following this approach often have better performances because of two main reasons: First, segmenting into different layers not only provides better matching results, but also significantly reduces the search space for finding appropriate matches used for inpainting. Second, the background completion process can be made much faster by using an adaptive background replacement scheme or an efficient image inpainting technique.

While following this major trend of modular approach, our proposed technique differs from existing techniques by inpainting the entire objects instead of small spatio-temporal patches using available templates. The use of the entire object rather than individual patches provides a significant computational reduction which leads to a close-to-real-time implementation. The alignment and motion continuity are accomplished by using a sliding-window similarity metric in a dynamic programming setting. A preliminary version of our work has appeared in [18]. This earlier version lacked the ability to handle partial occlusions occurring when a moving object enters and exits a hole region. Partial occlusions were treated as complete occlusions by discarding the available information. As a result the earlier scheme had difficulties in establishing smooth transitions at the boundaries. We improve our previous approach by grouping consecutive, possibly incomplete object into a single entity under a sliding window mechanism. The process of grouping consecutive templates allows us to implicitly lock onto the global movement of the constituent objects and allows us to handle cases where the entire object is completely occluded. Succinctly put, our key contribution is a computationally-efficient object-based inpainting algorithm capable of inpainting partially- and completely-occluded objects and providing global motion consistency by using sliding-window registration and dynamic programming. We will demonstrate the effectiveness of our algorithm by inpainting different human subjects of varying pose, size and motion in video sequences captured through both static and moving cameras.

4.2 Overview of the Inpainting System

We first provide an overview of the entire system and describe the motivations behind the design. We begin by specifying the assumptions under which our system operates and note that these assumptions are in line with those made by the existing state-of-the-art systems. We assume that the scene consists of stationary background and moving foreground regions.The foreground object may change in pose significantly but it still retains the repetitive motion pattern. To inpaint the foreground objects that are occluded in the absence of model based or learning based methods, digital inpainting techniques require that the necessary information to fill in the hole be available in the video. For our focus on inpainting human motion in a short time window, this assumption is usually satisfied due to the fact that common human movements like walking or running are repetitive. This also ensures that the necessary information to fill in the hole is available in the video.We also limit that any camera movement present to be parallel to the image projection plane.

Figure 4.1 shows the schematic diagram of our proposed system. The input video is first fed into a background subtraction module. The output of this stage is a set of moving foreground blobs. In case of moving cameras, the camera motion is first estimated by a block matching scheme. The camera motion is then compensated by warping back all the video frames into the same coordinate system and estimate a background panorama. We then perform moving object segmentation that serves two purposes - it segments foreground blobs into constituent objects and links objects from frame to frame. The segmented objects, or object templates, are stored in a database. Target objects specified by external means are removed from each image forming the hole region to be inpainted.

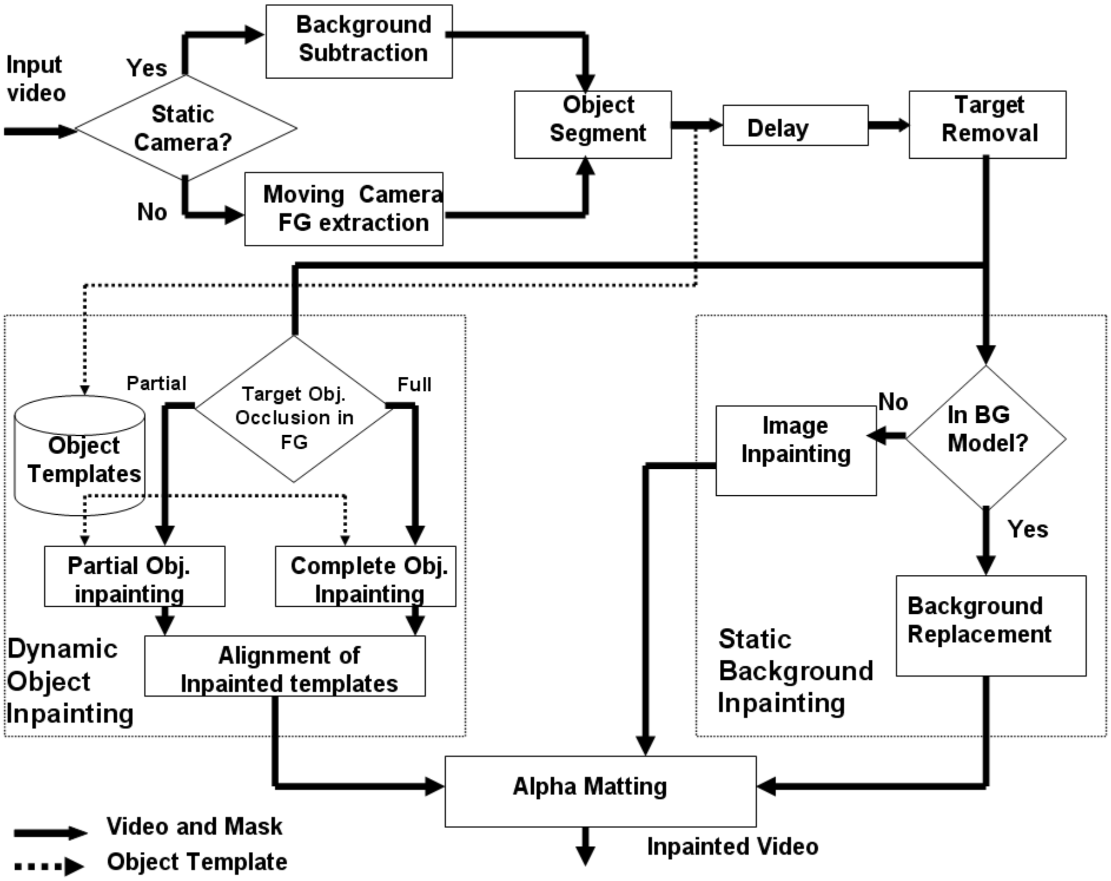


Figure 4.1: Schematic diagram of the proposed object removal and inpainting system.

The static portion of the hole can be filled by an adaptively updated background image if background information is available. Otherwise, image inpainting is performed based on the surrounding image statistics. The occluded moving foreground objects are inpainted by a two-stage process using the stored object templates. We first classify the frames in the hole as either partially-occluded or completely-occluded as shown in Figure 4.2. This is accomplished by comparing the size of the templates in the hole with the median size of templates in the database. The reason of handling these two cases separately is that the availability of partially-occluded objects allow direct spatial registration with the stored templates, while completely-occluded objects must rely on registration done before entering and after exiting the hole. The partial objects are first completed with the appropriate object templates selected by minimizing a window-based dissimilarity measure. Between a window of partially-occluded objects and a window of object templates from the database, we define the dissimilarity measure as the Sum of the Squared Differences (SSD) in their overlapping region plus a penalty based on the area of the non-overlapping region. The partially-occluded frame is then inpainted by alpha-matting with the object template that minimizes the window-based dissimilarity measure. Once the partially-occluded objects are inpainted, we are left with completely-occluded ones. They are inpainted by a Dynamic Programming based dissimilarity minimization process, but the matching cost is given by the dissimilarity between the available candidates in the database and the previously completed objects before and after the hole. The completed foreground and background regions are fused together using simple alpha matting. We now proceed to now describe each component in detailed manner.

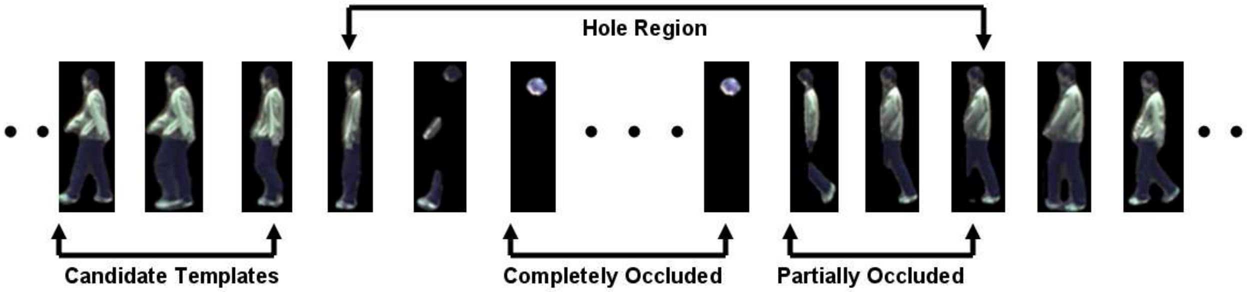


Figure 4.2: Classification of the input frames into partially and completely occluded frames.

4.3 Foreground Extraction and Object Segmentation

To extract the foreground moving blobs and the shadows, we employ a background subtraction algorithm proposed in [72] which needs to be trained by a sequence with just a static background. In the training phase, each pixel in the background is modeled by a 4-tuple (E_i, σ_i, a_i, b_i),where E_i is the expected color value, σ_i is the standard deviation of color value, a_i is the variation of the brightness distortion, and b_i is the variation of the chromaticity distortion. In the classification phase, every pixel in the incoming frame is classified either as a foreground pixel if the chromaticity exceeds a color threshold, or as a shadow pixel if they have similar chromaticity but lower brightness than those of the same pixel in the background image. The threshold values for the classification are computed in the training phase by setting the detection rate at 99.9%. Shadow separation is important as objects extracted in this step may be used for object interpolation in other locations where the old shadow is unlikely to be correct. We also apply a two-threshold hysteresis to prevent over-segmentation of foreground blobs.

In case of moving cameras, we extract the moving object in each frame by comparing it with the following frame using a block matching technique. Each frame is partitioned into non-overlapping blocks and the motion vector is estimated by searching in the neighborhood. Assuming that the camera moves parallel to the image projection plane, the median shift of all the blocks gives us a reliable estimate of the camera motion. The estimated camera motion is used to align all the available frames and the median value of each pixel position over time is chosen to create a background panorama. All the frames are then registered within this panorama by using the estimated motion vectors. We then segment the moving foreground from the background by modeling the distribution of the background pixels as a Gaussian Mixture and identifying the foreground pixels as those that significantly deviates from the background model.

The background subtraction process has a direct bearing on the quality of the captured object templates which are later used to perform foreground inpainting. The classification threshold is conservatively chosen based on confidence level so that static background is rarely mistaken as moving objects. Such an approach is usually adequate for indoor environment. There are other more sophisticated background subtraction schemes such as [73] that can handle outdoor dynamic background. The main focus of our work, however, is to demonstrate the robust and efficient performance of our inpainting algorithm to complete missing foreground regions when compared with the existing techniques. As such, we have not incorporated the most advanced background subtraction scheme in our experiments.

Following this step, it is imperative to segment foreground blobs into constituent objects during occlusion. Due to the large variation of object pose and movement, moving object segmentation is a very challenging problem. However, we would like to emphasize that the performance of our inpainting algorithm does *not* depend heavily on this process. By checking the variation in the object bounding boxes, we can usually detect whether the resulting segmentation is reasonable. If we cannot accurately segment the objects during occlusion, we can consider them as complete occlusions and proceed to inpaint them using our complete object inpainting process. Furthermore, in creating the hole for inpainting, we use a more relaxed classification threshold in our background subtraction to remove all traces of the foreground objects. By using our background replacement scheme, we can recreate the background texture that is erroneously removed by the background subtraction process.

As such, we have implemented a relatively simple tracking-based segmentation algorithm that is computationally very efficient. The object with ID i observed at time t, denoted as O_t^i, is represented by a normalized spatial color histogram $h_t^i(\mathbf{p}, c)$ at each pixel position $\mathbf{p}$ and color c. This histogram is adaptively updated throughout the lifetime of this object. Different quantities can be computed based on this histogram. For example, we can define the bounding region B_t^i to include all pixel location $\mathbf{p}$ with $h_t^i(\mathbf{p}) = \sum_c h_t^i(\mathbf{p}, c) > \tau$ for some small $\tau > 0$. We can also define the location of the object centroid $\mathbf{c}_t^i = \sum_{\mathbf{p}} \mathbf{p} \cdot h_t^i(\mathbf{p})$. To track an object, we adaptively maintain its velocity vector $\mathbf{v}_t^i$ using the following recursive update:

$$\mathbf{v}_t^i = \alpha \mathbf{v}_{t-1}^i + (1 - \alpha)(\mathbf{c}_t^i - \mathbf{c}_{t-1}^i) \tag{4.1}$$

where $\alpha > 0$ is an empirical parameter that dictates the rate of adaptation. With the velocity vector, we predict the spatial color histogram at time $t + 1$ as $\widetilde{h}_{t+1}^i(\mathbf{p}, c) = h_t^i(\mathbf{p} - \mathbf{v}_t^i, c)$. To determine the object label of foreground pixel $I_{t+1}(\mathbf{p})$, we adopt the following decision rule:

$$I_{t+1}(\mathbf{p}) = c \in O_{t+1}^i \text{ if } \frac{\widetilde{h}_{t+1}^i(\mathbf{p}, c)}{\widetilde{h}_{t+1}^j(\mathbf{p}, c)} > \frac{f_t^j}{f_t^i} \;\; \forall j \tag{4.2}$$

f_t^i is the "depth score" for object O_t^i. For all our test sequences, we take advantage of the simple scene geometry by assuming that object O_t^i is closer to the camera than object O_t^j if the baseline of the bounding region B_t^i is lower than that of B_t^j. The depth scores essentially act as priors for different objects in our moving object segmentation algorithm. The idea is to give bias towards objects that are closer to the camera as indicated by the baseline of their corresponding bounding boxes. The true prior would base on the part of the object (more likely to observe the occluded objects through the legs than the body of the occluding objects) and the amount of overlap between the occluding and occluded objects. As it is beyond the scope of this paper to precisely model all the different aspects of occlusion, we adopt the heuristic of assigning one as the depth score to the object farthest away from the camera, and doubling the object depth score as we move closer to the camera. After segmentation, the spatial color histogram of the *foreground* object is updated as follows:

$$h^i_{t+1}(\mathbf{p}, c) = \begin{cases} \alpha h^i_t(\mathbf{p} - \mathbf{v}^i_t, c) + (1 - \alpha) & I_{t+1}(\mathbf{p}) = c \in O^i_{t+1} \\ \alpha h^i_t(\mathbf{p} - \mathbf{v}^i_t, c) & \text{otherwise} \end{cases} \tag{4.3}$$

For all occluded objects, we ignore any partial observation and use the update rule $h^i_{t+1}(\mathbf{p}, c) = h^t_t(\mathbf{p} - \mathbf{v}, c)$.

4.4 Static Background Inpainting

To perform background replacement, we cannot directly use the background image from Section 4.3, which represents the long-term average of the time-varying background. Instead, we need to use background pixels that are most compatible with the current frame to fill the hole. As a result, we maintain a separate, fast-adaptive background image based on Kalman filter [12] and use it for background replacement. We further apply an edge-sensitive filter similar to the de-blocking filter used in H.263 to smooth the boundaries of the region [74]. In our implementation, we use a 5 × 5 filter with its strength determined by the dynamic range of the original pixels under the filter mask.

Occlusions cause by stationary objects cannot be completed by adaptive background replacement scheme as they lack background information in the occluding regions. For completing these regions, we use the exemplar-based image inpainting algorithm [13]. This patch based method first computes a priority for each pixel in the damaged region by assuming a patch size, which is set to 9×9 in our implementation. The priority is determined by two factors, extent of the support of undamaged pixels and edge strength. Consequently narrow or edge regions are assigned with higher priority and are completed before attempting to fill smooth and homogenous regions. This method is capable of filling large areas by first propagating linear structure and then synthesizing the image textures. It performs well for a wide range of images with good texture and structure replication but has difficulties in handling curved structures.

4.5 Dynamic Object Inpainting

In this section, we describe the sliding-window based technique for foreground object inpainting. Stationary or moving target objects that occlude the object of interest in the foreground are removed from the video. Removal of these target objects creates a hole which could contain partially or completely-occluded object of interest. We first identify the partially-occluded frames and inpaint them before proceeding to the completely-occluded ones. The partially-occluded frames are not used to inpaint the completely-occluded ones. They are only used in the registration process and at the final stage as the alpha matting to composite with the inpainted objects to synthesize natural object movements.

To simplify the explanation of the process, we assume that there is only one moving object that needs to be inpainted. If there are multiple objects, one can apply the same technique to each object in the order of their depth. For our test sequences, this order is computed based on the baselines of the objects. We denote the object template in the database as O_{t_i} where t_i indicates the time of the frame when the template is captured, and i indicates the time of *the frame inside the hole where this template is a candidate for inpainting.* We use $\widetilde{O}_i$ to denote the actual object inside or near the boundary of the hole at frame index i.

4.5.1 Sliding Window Based Dissimilarity Measure

The basic idea is to use a combined set of successive object templates extracted at other time instances as candidates for interpolation. In our previous work, we used individual object candidates in a sliding-window framework to complete the hole [18]. This process is further improved by grouping successive object templates and treating them as a single entity under a sliding-window mechanism. Let w be the number of consecutive templates in a group. Our algorithm works for any odd numbered group size w. We first define an appropriate distance measure which computes the dissimilarity in shape, color, texture between the available candidates in the database and the partial or completely-occluded foreground objects present in the hole.

Let O_{t_i} be an object template in the database and $\widetilde{O}_i$ be the same foreground object near the boundary of or inside the hole. All object templates are modeled as functions that map the 2D coordinates $\mathbf{p}$ to its gray-scale pixel value in the range of [0,255]. A dissimilarity measure $d(\widetilde{O}_i, O_{t_i})$ must take into account of the proper alignment and the difference in shapes. Let $\widetilde{M}_i$ and M_{t_i} be the binary masks that define the shape of $\widetilde{O}_i$ and O_{t_i} respectively. We define the distance $d(\widetilde{O}_i, O_{t_i})$ as follows:

$$d(\widetilde{O}_i, O_{t_i}) \triangleq \min_{\mathbf{m}} E_O(\widetilde{O}_i, O_{t_i}; \mathbf{m}) + E_N(\widetilde{O}_i, O_{t_i}; \mathbf{m}) \tag{4.4}$$

where the score $E_O(\widetilde{O}_i, O_{t_i}; \mathbf{m})$ between the overlapping areas and the score $E_N(\widetilde{O}_i, O_{t_i}; \mathbf{m})$ between the non-overlapping areas are defined as follows:

$$E_O(\widetilde{O}_i, O_{t_i}; \mathbf{m}) \triangleq \sum_{\mathbf{p} \in \widetilde{M}_i} \left[\widetilde{O}_i(\mathbf{p}) - O_{t_i}(\mathbf{p} + \mathbf{m})\right]^2 \widetilde{M}_i(\mathbf{p}) M_{t_i}(\mathbf{p} + \mathbf{m}) \tag{4.5}$$

and

$$E_N(\widetilde{O}_i, O_{t_i}; \mathbf{m}) \triangleq 255^2 \sum_{\mathbf{p} \in \widetilde{M}_i} \widetilde{M}_i(\mathbf{p}) \left[1 - M_{t_i}(\mathbf{p} + \mathbf{m})\right] \tag{4.6}$$

$E_O(\widetilde{O}_i, O_{t_i}; \mathbf{m})$ measures the SSD between the overlapping region of $\widetilde{O}_i$ and the candidate template O_{t_i} shifted by vector $\mathbf{m}$. The penalizing score $E_N(\widetilde{O}_i, O_{t_i}; \mathbf{m})$ counts the number of pixels, weighted by 255^2, within the object $\widetilde{O}_i$ not covered by O_{t_i} shifted by vector $\mathbf{m}$. This score acts as a balancing factor in helping to choose a candidate which not only has good similarity in overlapping areas but also is structurally similar to the partial template in the hole. For example, a candidate template similar in color and texture as well as structure (by the way of alignment with the partial template in the hole) would be chosen over a template which is similar in color and texture but does not align properly with the original partial template in the hole. The distance $d(\widetilde{O}_i, O_{t_i})$ is the minimum combined score over all possible alignment vectors. We denote the optimal alignment vector as $\mathbf{m}^*$.

If partial templates are available in a window of consecutive frames, we can increase the robustness of the matching by simultaneously matching the entire window. Let G_{t_i} be a group of w consecutive object templates O_{t_i}, …, O_{t_i+w-1} of the same foreground object from the candidate database. Let $\widetilde{G}_i$ be a group of partial objects $\widetilde{O}_i$, …, $\widetilde{O}_{i+w-1}$ of the same object in the hole. We now define a window-based distance $d_w(\widetilde{G}_i, G_{t_i})$ as follows:

$$d_w(\widetilde{G}_i, G_{t_i}) \triangleq \min_{\mathbf{m}} \sum_{f=0}^{w-1} E_O(\widetilde{O}_{i+f}, O_{t_i+f}; \mathbf{m}) + E_N(\widetilde{O}_{i+f}, O_{t_i+f}; \mathbf{m}) \tag{4.7}$$

Note that a single alignment vector $\mathbf{m}^*$ is used to minimize the distance between all the objects in the window. As such, $\mathbf{m}^*$ is robust enough to be directly used in the registration and inpainting of partial objects in the hole. As for complete occlusion, no partial templates are available to anchor the entire window and we have to resort to a dynamic programming framework to derive the proper registration throughout the entire duration of the hole. This process is explained in the following section.

4.5.2 Dynamic Programming Based Optimization

Let h be the number of frames we need to interpolate and w be the size of the object template window. We define our time index i ranged from $i = 0$, corresponding to the time instance $w - 1$ frame before the hole, to $i = h + 2w - 3$ or $w - 1$ frames after the hole. Thus,the hole ranges from frame $i = w - 1$ to $i = h + w - 2$.

Partial templates are usually available at the beginning and towards the end of the hole region. If there are more than w consecutive partial templates, they can be directly used to identify the proper object templates in the database for inpainting. For example, assume that we have half a window of partial objects present starting from the first frame (frame index w) of the hole. As the partial object $\widetilde{O}_w$ is in the middle of the window $\widetilde{G}_0$,$\widetilde{O}_w$ can be inpainted by the middle frame of the template window $G_{t_0^*}$ that minimizes the window distance with $\widetilde{G}_0$ as defined by Equation (4.7). Specifically, $G_{t_0^*}$ is defined as

$$G_{t_0^*} = \arg\min_{t_0} d_w(\widetilde{G}_0, G_{t_0}) \tag{4.8}$$

On the other hand, if there are completely-occluded objects in the hole, we cannot directly compute the window based dissimilarity measure. This is due to the lack of partial objects over the span of the window length. Furthermore, in this condition, the registration must be done using the available object templates before and after the hole. Figure 4.3 shows an example of such a sequence in which $h = 3$ and $w = 3$. To define the matching cost function for the completely-occluded objects, we consider all possible w-frame windows of object candidates in our database such that when we slide them across the hole, the total distances between overlapping object templates from successive windows are minimized. This total distance is recursively computed as shown in the following equation:

$$\begin{aligned} C_{t_i} \triangleq & \sum_{j=0}^{w-1} d(\widetilde{O}_{i+j}, O_{t_i+j}) \cdot 1_{\widetilde{O}_{i+j}} + \\ & \min_{t_{i-1}} \left(C_{t_{i-1}} + 1_{\{i>0\}} \cdot \sum_{j=0}^{w-2} d(O_{t_i+j}, O_{t_{i-1}+j-1}) \right) \end{aligned} \tag{4.9}$$

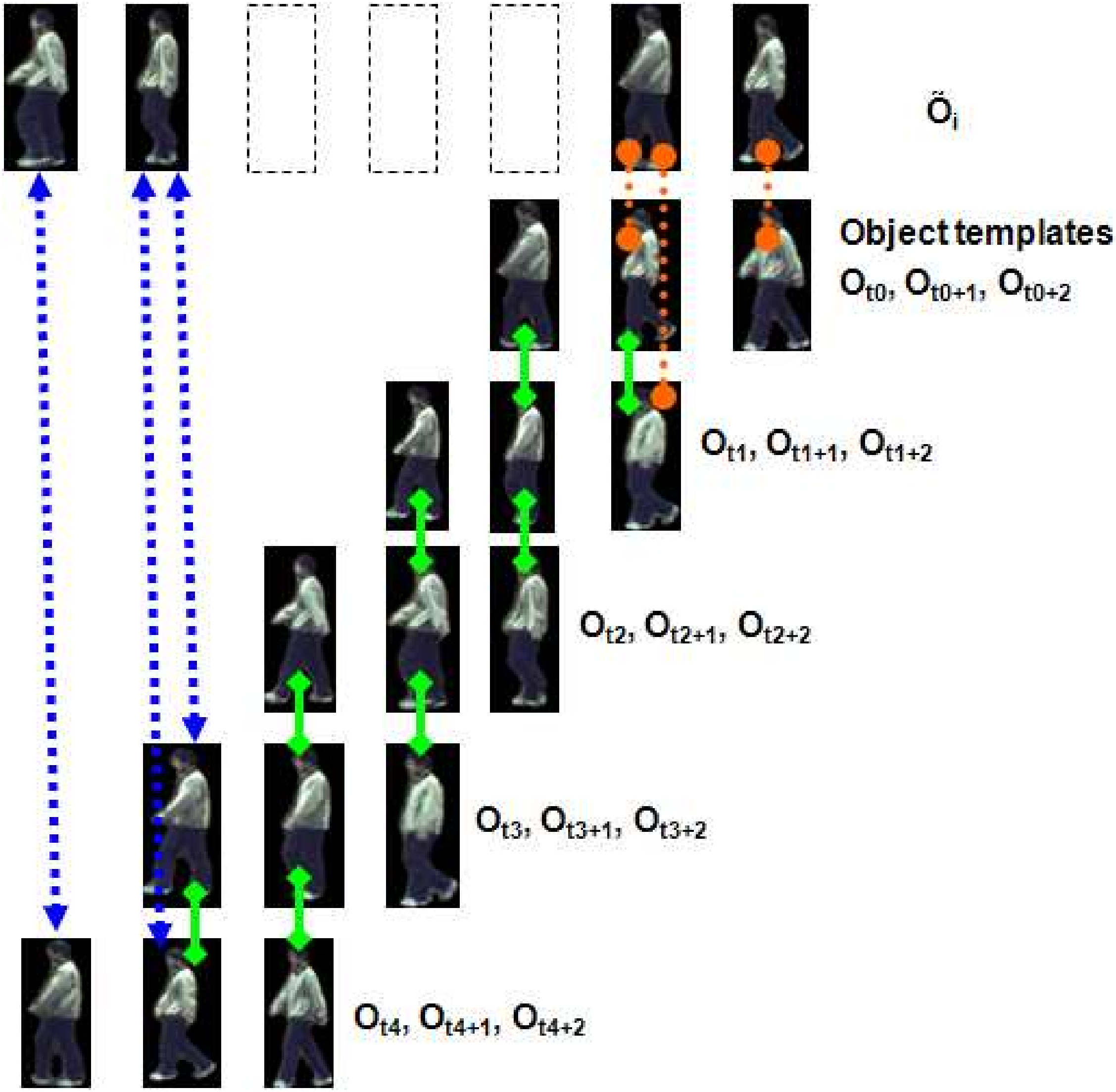

$\tilde{O}_i$
Object templates
$O_{t0}, O_{t0+1}, O_{t0+2}$
$O_{t1}, O_{t1+1}, O_{t1+2}$
$O_{t2}, O_{t2+1}, O_{t2+2}$
$O_{t3}, O_{t3+1}, O_{t3+2}$
$O_{t4}, O_{t4+1}, O_{t4+2}$

Figure 4.3: Illustration of object interpolation under complete occlusion: the top row shows the original sequence with completely occluded templates in the hole. The subsequent rows show the best 3-frame sliding windows of object templates that minimize the recursive cost function defined in (4.9). Different color lines indicate different components in (4.9).

for $i = 0$ to $i = h + 2w - 3$, spanning the entire duration of the hole. We set the boundary condition $C_{t_{-1}} = 0$. The first term of Equation (4.9) is the cost between the candidate templates O_{t_i+j} and the partial objects $\widetilde{O}_{i+j}$ near the boundary of the hole, provided that they are present and hence the use of an indicator function $1_{\widetilde{O}_{i+j}}$. These distance measurements are indicated by the red and blue arrows in Figure 4.3. The second term in (4.9) represents the cumulative cost across the complete occlusion where we maintain the motion continuity by computing the overlap between successive candidate windows. These measurements are indicated by the green arrows in Figure 4.3. By using a dynamic program to minimize the ending cost function $C_{t_{h+2w-3}}$, we obtain a series optimal template windows $G_{t_i^*} = \{O_{t_i^*}, \ldots, O_{t_i^*+w-1}\}$ for $i = 0$ to $i = h + 2w - 3$. To inpaint frame i which is in the middle of the window $G_{t^*_{i-(w-1)/2}}$, the middle object template $O_{t^*_{i-(w-1)/2}+(w-1)/2}$ is used.

In the final step, we need to put the optimal object candidate at the correct location within each interpolated frame to achieve smooth object movement. To accomplish that, we utilize the alignment vectors from the object distances defined in (4.4) and (4.7). When partially-occluded objects are present in the entire window, solving the minimization in (4.8) provides the optimal template window $G_{t_i^*}$ with corresponding alignment vector $\mathbf{m}_{t_i^*}$. As we use the frame in the middle of the window for inpainting, we denote the centroid of that object template as $\mathbf{c}_{t_i^*}$. The centroid of the inpainted object $\widehat{\mathbf{c}}_i$ is simply:

$$\widehat{\mathbf{c}}_i = \mathbf{c}_{t_i^*} + \mathbf{m}_{t_i^*} \tag{4.10}$$

For the duration of the hole where complete occlusion occurs, we do not have a direct measurement of the actual alignment vector for each optimal inpainting template. The alignment vector $\mathbf{m}_{t_i^*}$ is measured with respect to the optimal inpainting template of the previous frame rather than the actual object in the scene. As such, to compute $\widehat{\mathbf{c}}_i$, we need to sum all the alignment vectors starting from the first frame when complete occlusion occurs:

$$\widehat{\mathbf{c}}_i = \mathbf{c}_{t_i^*} + \sum_{j=w-1}^{i} \mathbf{m}_{t_j^*} \tag{4.11}$$

As error can accumulate throughout the hole, we need to introduce an adjustment vector $\mathbf{d}$ to each frame and then compute the actual centroid location $\widehat{\widehat{\mathbf{c}}}_i$ by adding this adjustment vector:

$$\mathbf{d} = \frac{1}{h}\left(\mathbf{c}_{h+w-1} - \widehat{\mathbf{c}}_{h+w-1}\right) \quad \text{and} \quad \widehat{\widehat{\mathbf{c}}}_i = \widehat{\mathbf{c}}_i + d \tag{4.12}$$

4.6 Experimental Results and Discussion

The algorithm presented in the previous sections are tested on six video sequences and the results of results of our algorithm are presented in this section. Some of the tested video sequences are form previously existing methods. To compare the effectiveness of our algorithm and to do it in an equal footing, we our we apply our technique on the data provided by [16, 75] in Section 7.3 and 7.5. The performance of our algorithm is compared with the existing methods in terms of the inpainting quality and complexity. Each of the sequences presented here highlights the flexibility of our technique in addressing challenging scenarios including inpainting of partial or completely occluded objects, inpainting under moving camera, and inpainting of moving objects with complex motion, changing pose and perspective. Table 4.1 summarizes the details of the video sequences used in this section. Our results and the original video sequences are available in our project website at `http://vis.uky.edu/mialab/Video Inpainting.html`.

Table 4.1: Video sequences and their attributes (POF=partially occluded frames, FO=fully occluded frames)

Fig	Name	Resolution	length	POF	FOF
4.4	Three Person	320 × 240	75	15	0
4.5	One Board	320 × 240	100	14	14
4.6	Moving Camera	320 × 240	40	9	0
4.7	Spinning Person	320 × 240	140	23	0
4.9	Perspective change	320 × 240	64	12	0
4.10	Jumping Girl	300 × 100	240	25	21

4.6.1 Inpainting Under Multiple Occlusions Using Tracking

We first show our results on a sequence which has multiple occlusions. The "three person" sequence listed in Table 4.1 is a typical indoor surveillance sequence captured by a stationary camera. As shown in Figure 4.4, the object to be inpainted is being occluded by two other moving foreground objects at different time instances. Also notice that the object to be inpainted is occluded by a stationary object at the start of the sequence. As a result, there are incomplete templates of the object of interest in the database. We do not assume any a priori information about the presence of incomplete templates due to this occlusion. As discussed in Section 4.3, we first extract the moving foreground regions by background subtraction and employ tracking and object segmentation to identify individual objects. In Figure 4.4, the first column shows four frames from the input sequence and the second column shows the results of object segmentation. The cyan and yellow colored regions represent the hole region. As the object of interest is only partially occluded, we use the partial object templates for registration and inpainting as described in Section 4.5. The window size used in the dissimilarity measurement is set to five. The last column in Figure 4.4 shows the inpainted results. As we can observe, the system is able to fill the holes in a seamless manner without using a priori knowledge about the incomplete templates.

4.6.2 Inpainting Under Complete Occlusions

In this section we highlight the efficacy of our inpainting algorithm in handling complete occlusion over a number of frames. The top row of Figure 4.5 shows four input frames from the "One Board" sequence in which the moving foreground is occluded by a stationary board in the front. This sequence presents an extremely challenging task as it suffers near complete occlusion for 14 frames when the person is walking behind the board. Our system first identifies the hole subsequence that contains partial templates and employs the partially-occluded object inpainting scheme. The system then proceeds to complete the remainder of the holes using fully-occluded object inpainting. The window size is again set at five. The inpainted video frames are shown in the second row of Figure 4.5. The static regions in the background are inpainted by an image inpainting scheme [13]. On close inspection, we can observe some artifacts on the ground below the inpainted person. This is caused by transferring part of the moving shadows due to imprecise segmentation.

4.6.3 Inpainting Under Moving Camera Conditions

Figure 4.6 shows the video sequence "Moving Camera" taken from a moving camera originally used in [16]. The goal is to remove the foreground person and inpaint

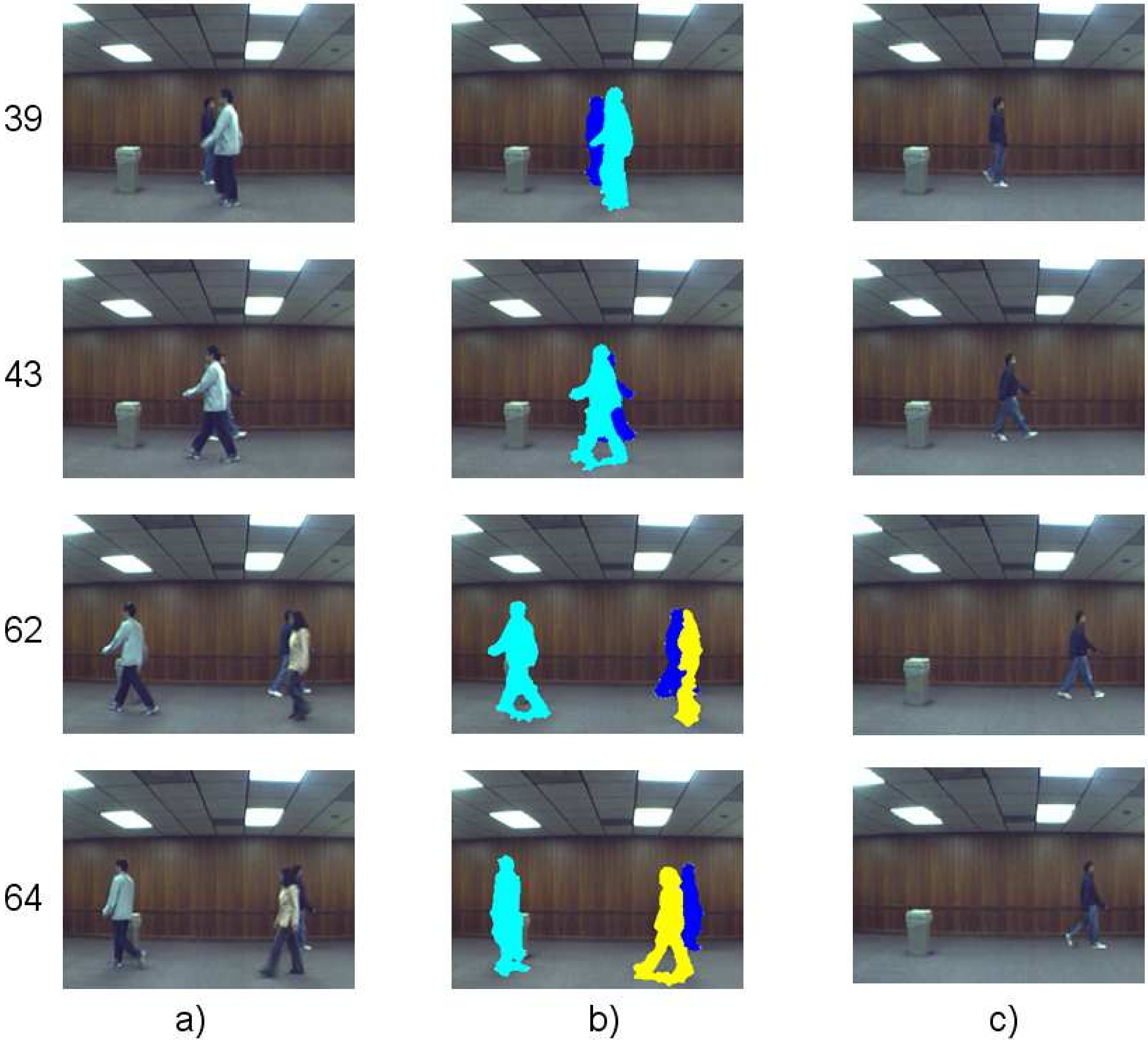


Figure 4.4: Inpainting under multiple occlusions using tracking : a) The first column shows the original input sequence along with the frame number. b) The second column shows results of the tracking and foreground segmentation stage in which objects are classified and segmented. c)The third column shows the inpainted result in which the moving foreground target objects are eliminated and the object of interest located at the back is inpainted. Also notice that the stationary object which is occluded by moving foreground in frame 62 and 64 has been inpainted back by adaptive background substitution.

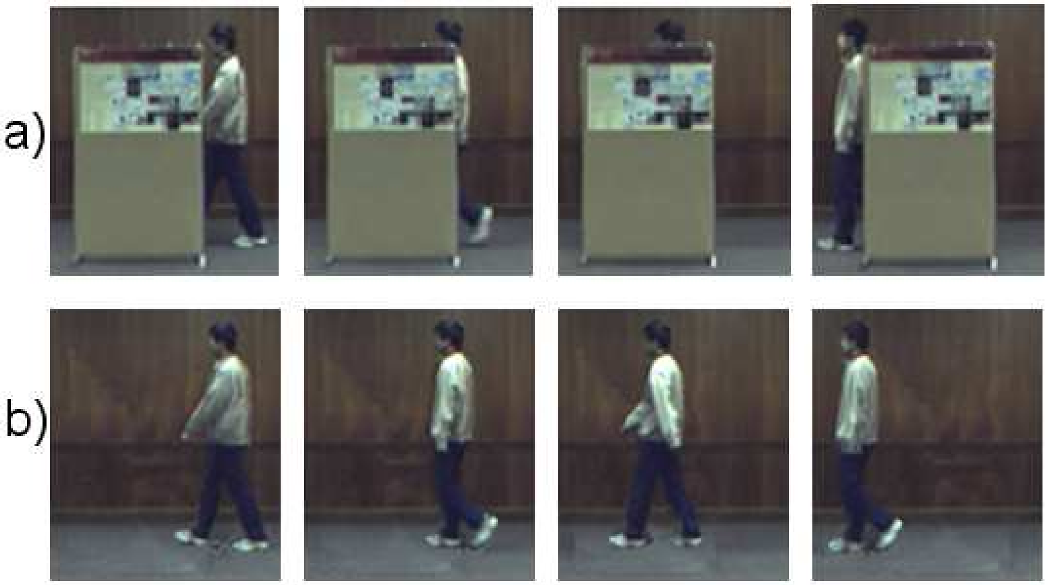


Figure 4.5: Inpainting under complete occlusions: a) First row shows the unmodified input sequence with a stationary occlusion. Note the heavy occlusion and also the fact that very little foreground information is available in third frame. b) The second row shows the inpainted sequence. Notice that the completely occluded object has been effectively inpainted with smooth transitions in the entry and exit. The stationary background is inpainted using the image inpainting the technique in [13].

the background person during the brief occlusion. We use this sequence to demonstrate the capability of our system in addressing inpainting under restricted camera motion. We use the ground truth of the mask of the target object to be removed, which lies in front of the object of interest, which were provided by the authors of [16]. It should be noted that our appearance based segmentation technique would not yield better results in this case as the interfering objects share similar appearances. The hole region extends over nine frames and has near complete occlusions when the objects cross each other. We employ our moving camera foreground extraction scheme explained in Section 4.3 to extract the moving foreground regions. The occluded regions are inpainted using our dynamic object inpainting scheme. The background regions are inpainted using a background replacement scheme based on the estimated background panorama. A closeup look at the inpainted frames using our method and that of [16] are shown in the second and third columns of Figure 4.6 respectively. Our approach clearly provides better spatial details due to the use of actual texture rather than texture synthesis. A potential shortcoming of our object-based approach over patch-based approaches is that it might occasionally select a template with a wrong pose due to the similarity of the visible regions of the partial template and the lack of depth information. An example can be found in the top image of Figure 4.6b which shows the wrong leg moving forward as it becomes apparent in the subsequent images.

4.6.4 Inpainting Under Change in Pose

In previous sections, we compared our inpainting scheme with other methods by using video sequences that can operated by all the techniques. Here we illustrate the versatility of our scheme by inpainting a foreground object that undergoes a significant change in pose in the hole region. To the best of our knowledge, currently existing techniques are not capable of processing video sequences containing complex motion such as change in pose. The work by Shiratori et.al address a very restricted case of completing complex motion, for instance, by recreating small transient movements over a period of two or three frames [52]. While doing so, they do suffer from severe blurring artifacts while our scheme provides a much clearer rendering of the missing motion through the use of compatible object templates as explained below.

The “Spinning Person” sequence shows a person spinning from the right of the scene to the left. The first row of Figure 4.7 shows the closeup view of the partially

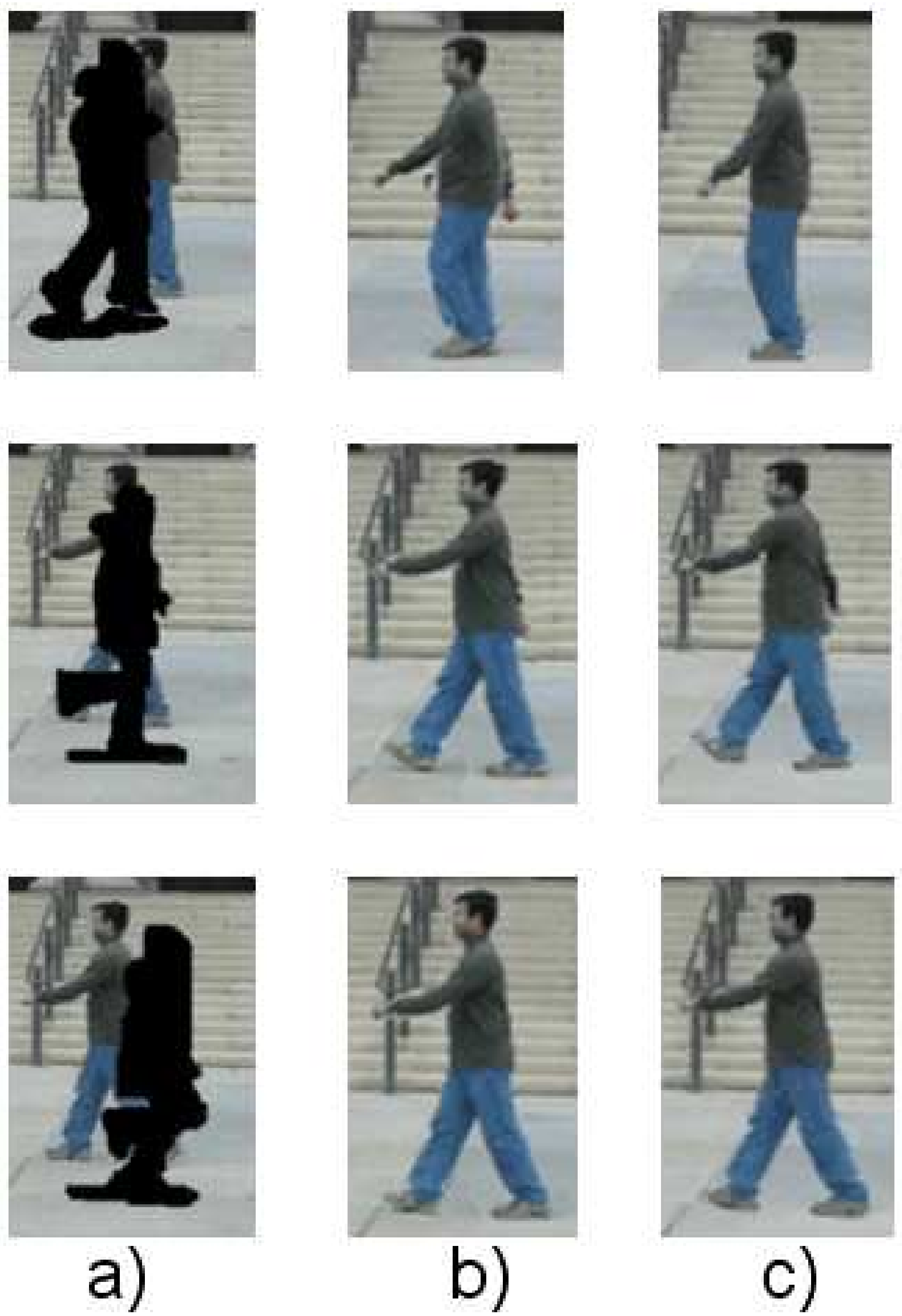


Figure 4.6: Inpainting under moving camera conditions: a) First column shows the magnified version of the input sequence with the hole. b) The second column shows a close-up look of the final inpainting results using our algorithm. c) The third column shows the final inpainting result using the algorithm in. [16].

occluded foreground object inside a synthetic hole. Figure 4.7b shows the results of our inpainting algorithm. It is clear that the change of pose of the head relative to the body inside the hole has been been inpainted realistically. By using the partial templates inside the hole region in a sliding window based measure, we are able to lock onto the global change in pose occurring at a different time instance to inpaint the hole in a holistic manner. Note that the assumption of repetitive movement is not strictly observed as the pose of the spinning person varies greatly throughout the sequence. Existing video inpainting sequences including patch based methods may encounter difficulty in inpainting such a sequence because of the greedy approach they pursue in filling the patches. The idea of combining sets of consecutive object templates for dynamic object inpainting gives us a significant advantage over existing techniques.

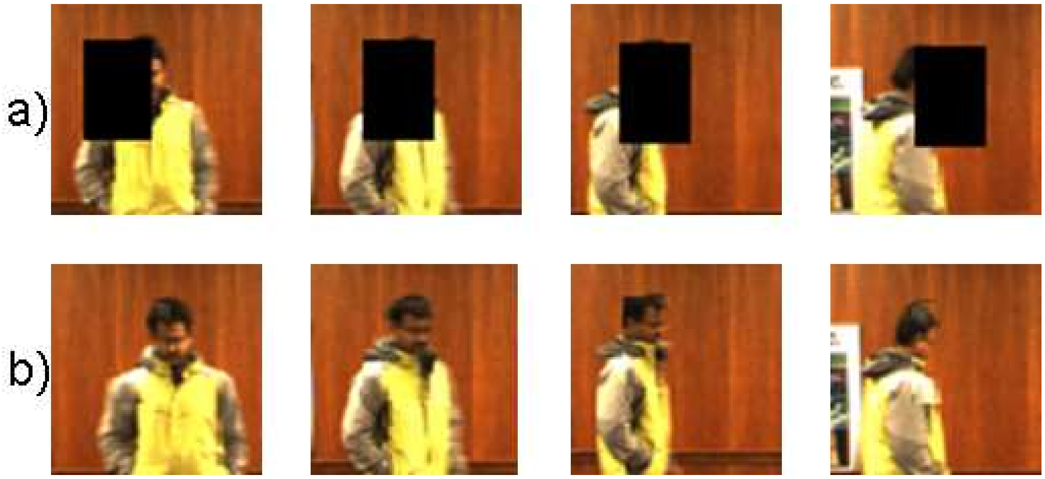


Figure 4.7: Inpainting of foreground object with changing pose: a) The first row shows the magnified version of the input sequence with the hole. b) The second row shows a close-up look of the final inpainting results using our algorithm.

4.6.5 Inpainting Under Perspective Change

The motivation behind this section is to illustrate that our video inpainting scheme can be extended to moving objects under perspective changes. Here we consider a case in which the person is walking at a constant velocity but the trajectory of the motion is not parallel to the camera plane. Due to this condition, the foreground object undergoes a perspective change. Figure 4.8a shows a montage of the foreground objects increasing in size as it moves closer towards the camera. We perform a normalization procedure to rectify the foreground templates so that the motion trajectory is parallel to the camera plane. Two points, at the top of the head and at the bottom of the feet, are extracted from the foreground object at different time instances. We obtain a series of vertical lines by connecting the head and feet points at different frames, and two horizontal lines by connecting all the head points as well as all the feet points. The vertical vanishing point is then computed as the intersection of all the vertical lines while the horizontal vanishing point is formed by intersecting the two horizontal lines. A metric rectification is then performed with the help of these vanishing points and the resulting rectified foreground volume is shown in Figure 4.8b [76].

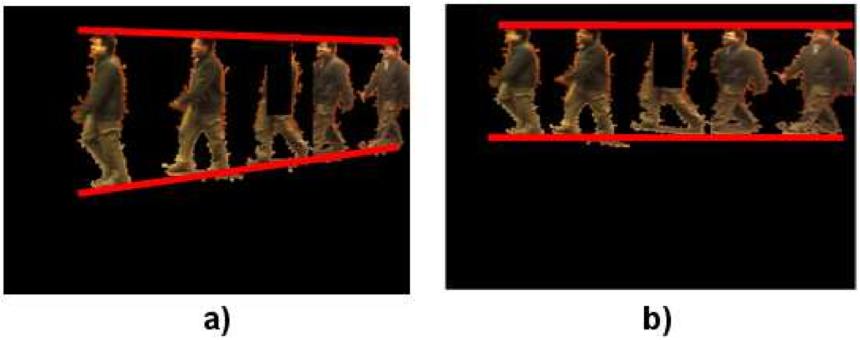


Figure 4.8: Rectification of foreground objects: a) Foreground object under perspective change; b) Foreground object normalized by metric rectification.

We then apply our inpainting algorithm on the rectified foreground volume. Figure 4.9a and 4.9b show the damaged and the inpainted foreground volume. The blocking artifact is more pronounced in this case due to the brightness variations of the objects templates in the database and the blurring effect is a main consequence of the scaling involved during the interpolation of candidates in the metric rectification process.

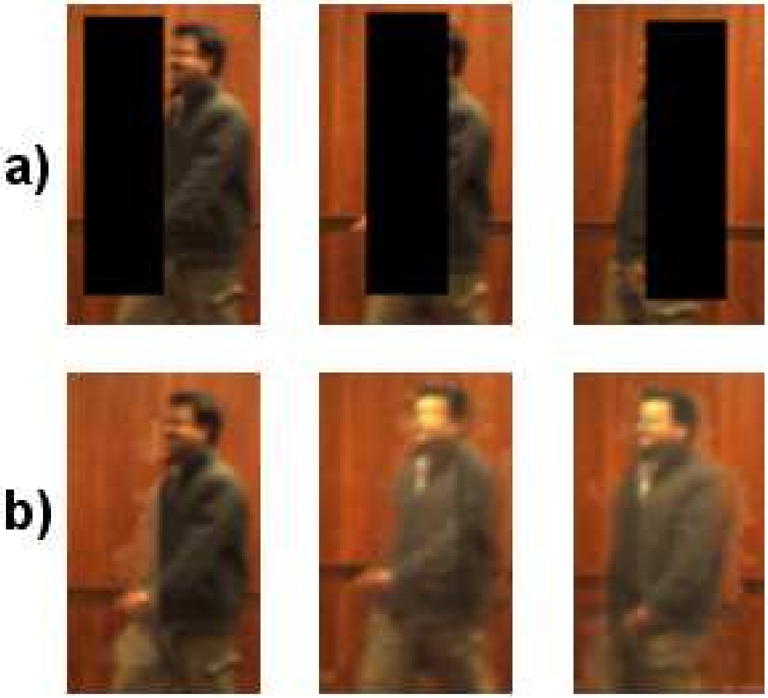


Figure 4.9: Inpainting foreground object under perspective change: a) A closeup of the foreground objects with the hole; b) Inpainted foreground objects using our algorithm.

4.6.6 Inpainting Under Prolonged Partial and Complete Occlusions

This “Jumping Girl” sequence used in [75] shows the moving object of interest exhibiting a repetitive jumping motion throughout the video. The object of interest is occluded by a stationary object which is cropped out using a user-supplied mask. Close-ups of the object of interest near the hole are shown in the first column of Figure

4.10. Using background subtraction algorithm, our system first extracts the moving foreground objects. Then, the frames with partial occlusion are identified and filled using the partial template inpainting. The remaining frames of the hole are filled by fully-occluded object inpainting scheme. Since we lack a background model in the hole regions, the background image is filled using an image inpainting scheme. The second column in Figure 4.10 shows our inpainting results. The corresponding results from [16] and [75] are shown in the third and fourth column of Figure 4.10 respectively. The results from [16] are noisy and of poor resolution – this could be the limitation of their method in dealing with near complete occlusions as there is very little structural information present to do a reasonable inpainting. The foreground inpainting results from [75] are similar to ours but their results suffers from oversmoothing in the background regions.

4.7 Impact of Segmentation on Video Inpainting Quality

Our system while being modular and intuitive can be construed to be heavily dependent upon the performance of the background subtraction and moving object segmentation algorithms. While there is some merit in this argument, it is equally important to carefully analyze the impact of these processes on our inpainting algorithm. The discussion presented in this section will help understand the aspects of segmentation process that has an impact on the proper working of the inpainting process.

There are two steps of segmentation that can affect the performance of our inpainting algorithm. They are:

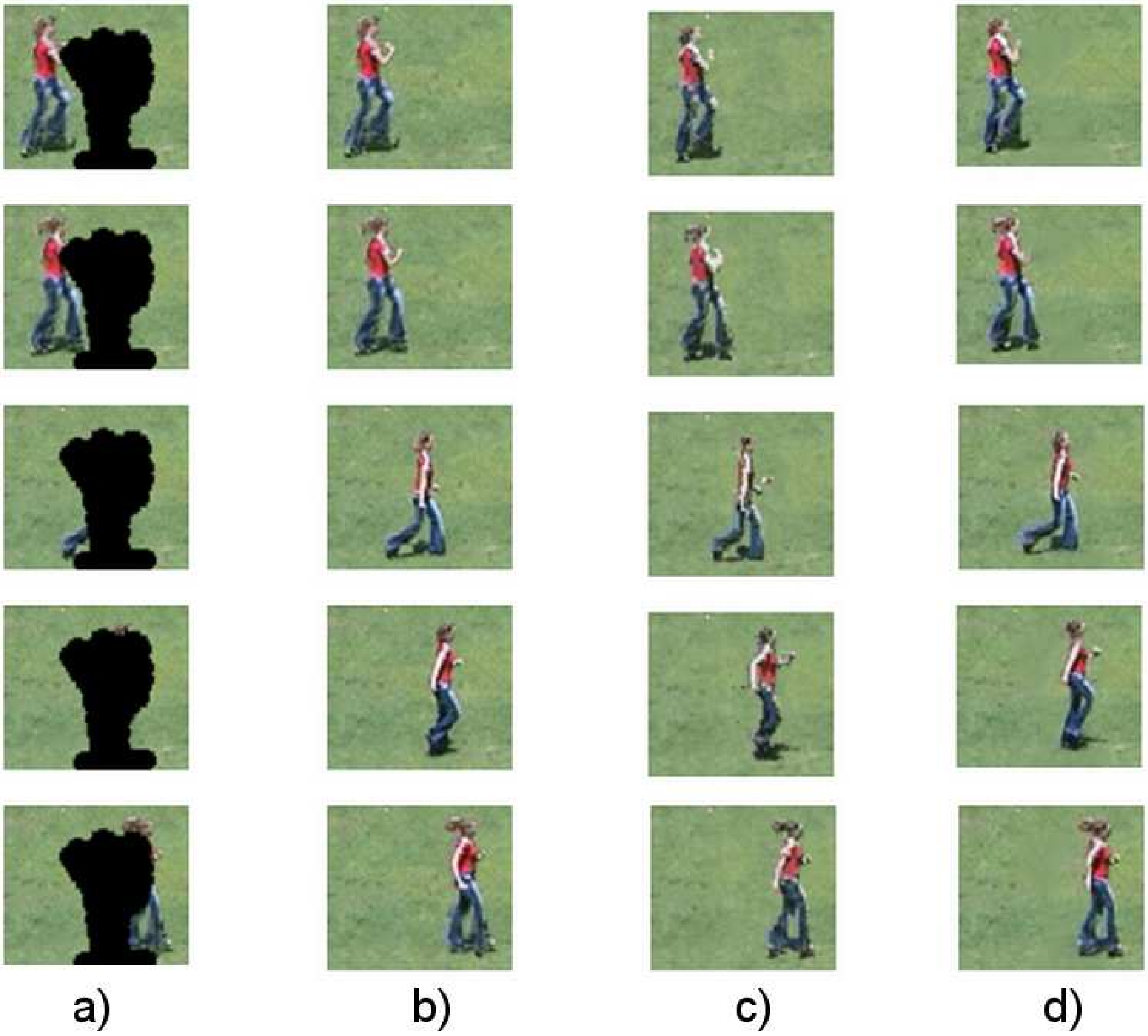


Figure 4.10: Inpainting under prolonged partial and complete occlusions: a) First column shows a magnified view of the input sequences adjoining the hole boundaries along with the corresponding frame number. b) Second column shows the inpainted result using our algorithm. Notice the visually consistent foreground inpainting and no blurring of the background. c) Third column shows the result of inpainting using the algorithm by Patwardhan et al. [16]. The inpainted foreground inpainting appears coarse and is not consistent throughout. d) Fourth column shows the result of inpainting using the algorithm by Wexler et al. [75]. The background appears to suffer from oversmoothing.

1. Background subtraction on the given input video to separate the foreground and background layers to collect object templates in the database for inpainting.
2. Moving object segmentation of occluding foreground objects into respective constituent objects.

The first segmentation process has a direct bearing on the quality of the templates in the database, which are later used to perform foreground inpainting. To that effect, we have employed statistical color based background subtraction method based on [72] for stationary cameras and moving foreground extraction method for moving cameras. The classification threshold is conservatively chosen based on confidence level so that static background is rarely mistaken as moving objects. Such an approach is usually adequate for indoor surveillance environment. There are other more sophisticated background subtraction schemes such as [73] that can handle outdoor dynamic background. The main focus of our work, however, is to demonstrate the robust and efficient performance of our inpainting algorithm to complete missing foreground regions when compared with the existing techniques. As such, we have not incorporated the most advanced background subtraction scheme in our experiments.

The second segmentation process aims at separating two or more foreground objects during occlusion. Due to the large variation of object pose and movement, moving object segmentation is technically more difficult than background subtraction. However, we would like to emphasize that the performance of our inpainting algorithm does *not* depend heavily on this process. By checking the variation in the object bounding boxes, we can usually detect whether the resulting segmentation is reasonable. If we cannot accurately segment the objects during occlusion, we can consider them as complete occlusions and proceed to inpaint them using our complete-occlusion based foreground inpainting process. Furthermore, in creating the hole for inpainting, we use a more relaxed classification threshold in our background subtraction to remove all traces of the foreground objects. By using our background replacement scheme, we can recreate the background texture that is erroneously removed by the background subtraction process.

4.8 Complexity Analysis

In this section, we analyze the asymptotic complexity of our algorithm and compare it with those of the methods described in [16, 75]. Let H be the number of frames in the hole, and within each frame, the hole area can be separated into N_f foreground pixels and N_b background pixels. Assume that the information needed to inpaint this hole can be found in a spatio-temporal volume of K frames, and each frame contains roughly $N \approx N_f + N_b$ useful pixels. We further assume that all the foreground pixels belong to the same object. Denote the window size and the search size of the alignment vectors in our algorithm as W and S respectively. Ignoring all the preprocessing tasks and focusing solely on the inpainting process, the following table compares the asymptotic complexities of three different variants of our scheme and those in [16, 75]:

The first row of Table 4.2 shows the complexity of the partially-occluded object inpainting as described in Section 4.5. The complexity of the dissimilarity measurement in Equation (4.7) is $O(N_f WS)$. The dynamic programming is linear in terms of the number of frames H in the hole and thus the overall complexity for inpainting the

Table 4.2: Asymptotic Complexities of Three Schemes (PO=Partially-Occluded, FO=Fully-Occluded)

Schemes	Complexity
PO objects	$O(N_f HKWS + N_b H)$
FO objects with insufficient templates	$O(N_f HK^2 WS + N_b H)$
FO objects with sufficient templates	$O(N_f KWS + N_b H)$
Algorithm in [16]	$O(N_f^2 HK + N_b H)$
Algorithm in [75]	$O((N_f + N_b)^2 HK)$

foreground is $O(N_f HKWS)$. The complexity of background inpainting, regardless of whether background replacement or image inpainting is used, is simply $O(N_b H)$. The second row shows the complexity of the fully-occluded object inpainting based on Equation (4.9). Since there is no existing object template to compare with, this scheme requires each template to be compared with all other possible templates in order to deduce the optimal motion continuity. This results in a quadratic increase from K to K^2, where K denotes the number of available templates. We notice that when the number of consecutive candidates available in the database is significantly greater than the number of the completely occluded templates in the hole, our algorithm always uses consecutive object templates to fill the hole. In other words, it is possible to have a significant speedup by heuristically testing only the strings of consecutive object templates slightly longer than the hole, and choosing the one that minimizes the dissimilarity at the boundaries. This scheme does not depend on the length of the hole and results in the complexity depicted in the third row of Table 4.2. The exemplar-based scheme in [16] is similar to ours in terms of the separation of foreground and background inpainting. However, their scheme requires an extra factor of N_f as it needs to search the entire available foreground for filling each patch. The space-time completion scheme in [75] does not separate foreground and background and as a result, has the highest computational complexity.

Our algorithm was implemented using MATLAB version 7.0 on a Xeon 2.1Ghz machine with 4 Gigabyte of memory. The actual running time to perform the inpainting for the sequences used in this work are given in Table 4.3. Optimizing the code and porting it to C or C++ should significantly improve the performance. In addition, we do not exploit the preliminary alignment results from the tracking module and end up using a search size S close to the size of the entire foreground object. The comparison with the available templates can easily be parallelized which can further improve the performance.

Table 4.3: Actual Execution Time for the video sequences

Name	Inpainting $W = 3$	Inpainting $W = 5$	Pre-Process
Three Person	7.4 mins	10.2 mins	30 secs
One Board	3.5 mins	8.3 mins	40 secs
Moving Camera	2.6 mins	4.8 mins	3 mins
Spinning Person	12.6 mins	18.2 mins	35 secs
Perspective	3.6 mins	7.1 mins	35 secs
Jumping Girl	6.4 mins	11.4 mins	30 secs

4.9 Summary and Conclusions

The elaborate experiments, results and discussion presented in the sections above demonstrate the versatility of our proposed system in addressing a variety of video sequences in our daily lives. It is also worthwhile to point out that most state-of-the-art inpainting schemes are tested on simple walking and jumping sequences [16,54,75].

We have presented a complete framework for an efficient video inpainting algorithm capable of addressing inpainting under stationary camera and moving cameras. The stationary background region is filled by a combination of adaptive background replacement and image inpainting technique. Unlike other patch-based inpainting schemes, we inpaint the foreground object as a whole by formulating the problem as energy minimization. A new window-based dissimilarity measure is introduced to provide improved motion continuity within and at the boundaries of the hole. The final optimal candidates are selected by solving the minimization problem with dynamic programming. Our system offers several advantages over existing state-of-the-art methods in the following aspects:

1. Ability to handle large holes including cases where the occluded object is completely missing for several frames

2. Robust candidate selection from a set object templates provides significant speed up over existing patch-based schemes.

3. Object alignment process by the window-based scheme generates natural object movements inside the hole and provide smooth transitions at hole boundaries without resorting to any a prior motion model.

4. Instead of explicit use of motion information in the form of optic flow, we introduce a novel similarity measure which provides better spatial and temporal continuity in inpainting by grouping of successive templates in a dynamic programming framework.

5. The proposed scheme also provides a unified framework to address videos from both static and moving cameras and to handle moving objects with varying pose and changing perspective.

Chapter 5

Perceptual Evaluation of Image Inpainting

The objective of the image inpainting algorithm is to perform a perceptually seamless inpainting, making it difficult for the HVS to detect that the image has been modified. Hence, evaluating the perceptual quality of the inpainting algorithm we must rely on the features of HVS. Through a combination of detailed eye-tracking experiment and a novel subjective ranking mechanism we establish a direct relation between the perceptual quality of image inpainting and human attention. By comparing the gaze density within and outside the hole regions of the inpainted images, we show that HVS is attracted to the discernible artifacts introduced by the inpainting process. Defining a normalized gaze density measure using the unmodified original images as the reference, we show that the change in gaze distribution of unbiased observers is related to its perceptual quality. The subjective rating of images obtained from our experiment establishes a clear order of preference among different inpainted images and corroborates the results of the eye-tracking experiment. The strong coherence exhibited by the subjective ratings and the normalized gaze density give further credence to the mutual relation between the gaze and the perceptual image inpainting quality.

The rest of the chapter is organized as follows: in Section 5.2, we explain the experimental conditions, the stimuli followed by the setup of eye-tracker. In Section 5.3, we present the analysis of results of the gaze data along with the discussion of subjective ranking of inpainted images. We indeed show that low level artifacts created by imperfect inpainting process do impact the perceptual inpainting quality, even though this perceptual process is partly governed by higher level cognitive process. We obtain independent results that corroborate the findings of a previous study in analyzing the image inpainting quality and the viewing time in Section 5.5. Finally in Section 5.6 we provide the conclusions of this study and point future research directions.

5.1 Overview and Motivation

In our work we study the reason behind perceptual quality of image inpainting by a combination of eye-tracking experiments and a novel subjective ranking mechanism. An inpainting technique that cannot fill the hole in a *natural* way will introduce artifacts on the inpainted image. Extensive studies in cognitive visual attention have shown that interesting objects are visually salient and these regions provide *surprise* which attract human visual attention [20, 21]. Even though perceptual processes are partly driven by higher level cognitive process, low level vision plays an important role. In the context of image inpainting, the artifacts introduced could capture the attention which may cause a change in normal behavior of the HVS. This deviation from normalcy, can be quantified by measuring the changes in gaze pattern using an eye-tracker and can potentially relate to subjective inpainting quality. However, when the regions to be inpainted are ”interesting” areas of image, comparing it with original gaze might not be appropriate. In such case, if the inpainting algorithm were to leave an artifact it could potentially attract attention of HVS and the resulting gaze pattern would be similar to gaze obtained for unmodified image. Hence evaluating true perceptual inpainting quality should be carried out through a No-reference method.

5.2 Experimental Conditions, Stimuli and Eye-Tracker Setup

The data set used in this experiment, approved by non-medical IRB review, consisted of 45 unmodified images that were acquired from the data set made freely available in [1]. Along with the 45 unmodified images, two more versions for each image, one inpainted using [1] and the other using [13] were also used. Totally there was three distinct sets - Set A, B and C - each consisting of 45 images, equally distributed among the three versions of the images was created. A total of 24 nave observers, divided into 3 groups of 8 people, with normal or corrected vision participated in this study. Each subject from a group was required to view a total 45 images from one of Set A, B or C. It was ensured that there were no overlap among the three sets so that all the 45 images observed by each individual observer were distinct from each other. The subject had no a priori knowledge on the type of the image displayed (original, inpainted by algorithm [1] or [13]) and the ratio of unmodified versus inpainted images was unknown to them. The experiment was conducted under ambient lighting conditions and the images were displayed on a 19 inch, gamma corrected display with the monitor resolution set to 1280X960 pixels. The viewing distance was maintained at 65 cm's and each image was displayed in a random order for a duration of 6 seconds followed by a two second interval in which a blank screen was displayed. Studies have shown that while viewing a visual stimuli, the first 2-3 seconds of visual attention is driven mostly by low level vision processes and the six second viewing period in our experiment would be sufficient to evaluate the quality task [77].

The head position of each subject was stabilized using a chin rest to reduce the error in eye-tracking. Before the commencement of the actual experiment a calibration process is performed which enables the eye tracker to lock onto the position of the eye [78]. In the calibration process, a linear transformation that maps the direction of the eye gaze to the position of the gaze on computer monitor is estimated using a 3x3 calibration grid. During the course of the experiment, calibration patterns were displayed at random intervals for each observer to estimate calibration error and this data was used to offset for errors during the data analysis process. The gaze of the observers, in the form of the fixation position of the eye and the time during the viewing task, was recorded using the Seeing Machines faceLAB eye tracker. The fixation data in our experiment is indicative of the greater interest of the HVS in the particular area of the displayed image [79]. A gaze distribution image, as defined in Eqn(5.1) where δ is kronecker delta function and N is the number of samples, was obtained for every image viewed by the user by sampling the gaze data at 60 samples per second.

$$Gaze_{user}^{img}(x,y) = \sum_{i=1}^{N} \delta(x - x_i, y - y_i) \tag{5.1}$$

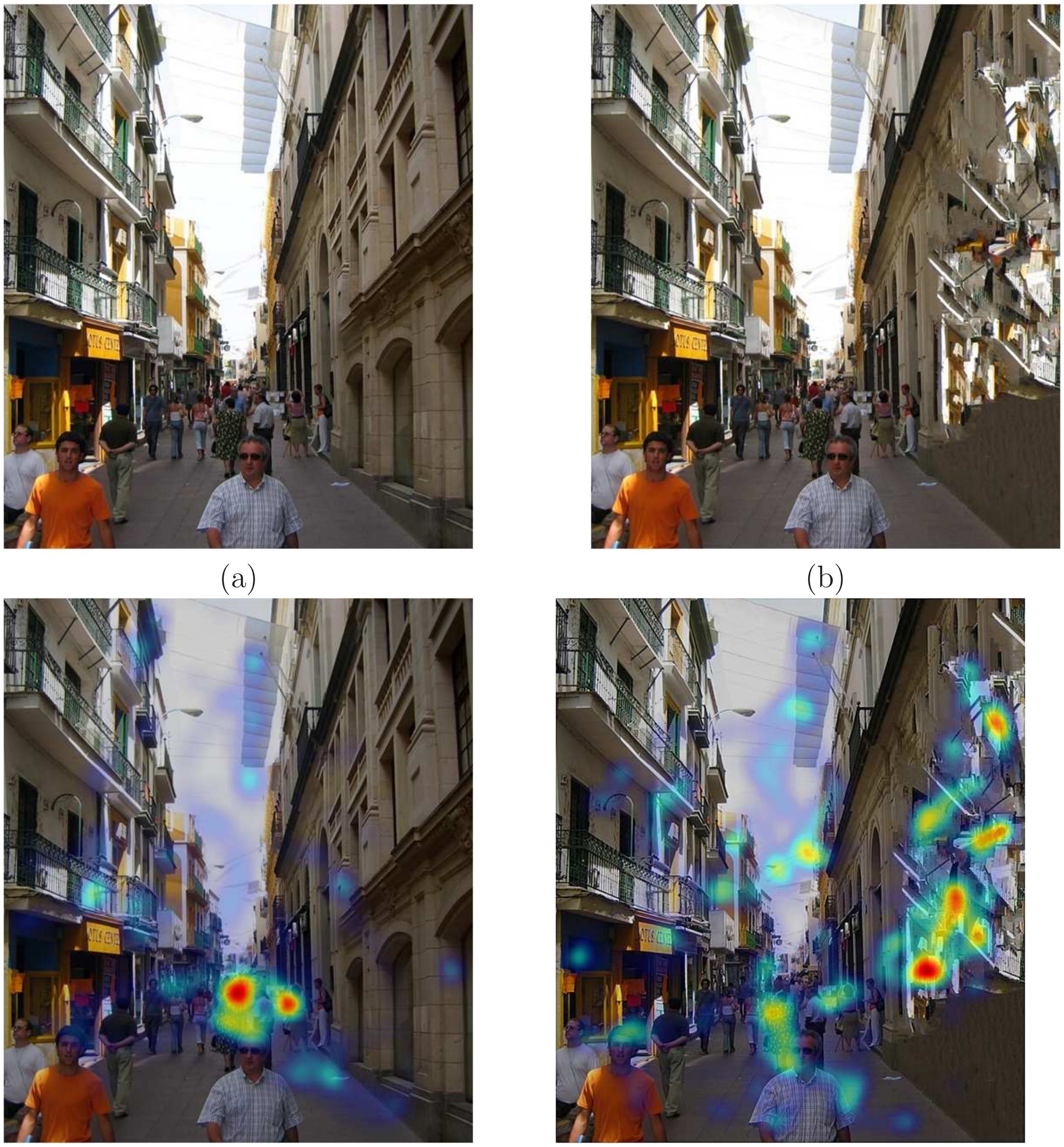
Each subject was advised to view the image freely and was instructed to use a mouse click event, as a binary indicator, if they think that the image displayed has been modified in any manner. Since there are eight subjects in a group, each image from Set A, B, or C can receive a maximal score of 8, reflecting the perceived image quality. An image with the score 0 corresponds to highest perceived quality, meaning that all eight observers concurring that the image was unmodified. Similarly an image with a subjective rating of 8 implies that all the observers deemed that the image was modified and a score of 4 corresponds to a 50% decision threshold. To summarize succinctly, for every image in Set A, B and C, we obtain the gaze distribution from eight subjects and a subjective rating in the range zero to eight, reflecting the perceived image inpainting quality of the image.

5.3 Gaze Data Analysis

We compute the average gaze distribution for every image from the gaze data obtained from eye-tracking experiments involving 8 subjects. Figure 5.1 shows two example inpainted images with their average gaze distribution superimposed on them. As expected, we can observe that the attention of the HVS is attracted to regions where inpainting artifacts are discernible indicating that the image has been subjected to some modification.

From the gaze distribution obtained from every subject for ever image, we compute the gaze density of any given image. Let Ω be the area denoted by the hole region and $\overline{\Omega}$ be the region outside the hole of an inpainted image. By comparing the gaze density within and outside the hole region of images inpainted by both the techniques we can quantitatively analyze the perceptual image inpainting quality. The gaze density of an image, inside and outside the hole region is given in (5.2) as

$$\begin{aligned} GD_{in} &= \sum_{user=1}^{8} \sum_{\Omega} gaze(\Omega) \\ GD_{out} &= \sum_{user=1}^{8} \sum_{\overline{\Omega}} gaze(\overline{\Omega}) \end{aligned} \tag{5.2}$$



(a)

(b)

(c)

(d)

Figure 5.1: Inpainted Images with gaze distribution superimposed: a) Image inpainted by technique in [1] b)Same image inpainted by [13] c) Average gaze distribution superimposed d) Corresponding avg. gaze distribution superimposed. Note the concentration of gaze distribution on the right side of image where the artifact in hole region is quite apparent.

Figure 5.2 presents the gaze density inside the and outside the hole region of inpainted images of both methods. From the results presented, we can see that the gaze density inside the hole region for images modified by algorithm in [13] is consistently greater than for images modified by [1]. To compare it on an equal footing with reference

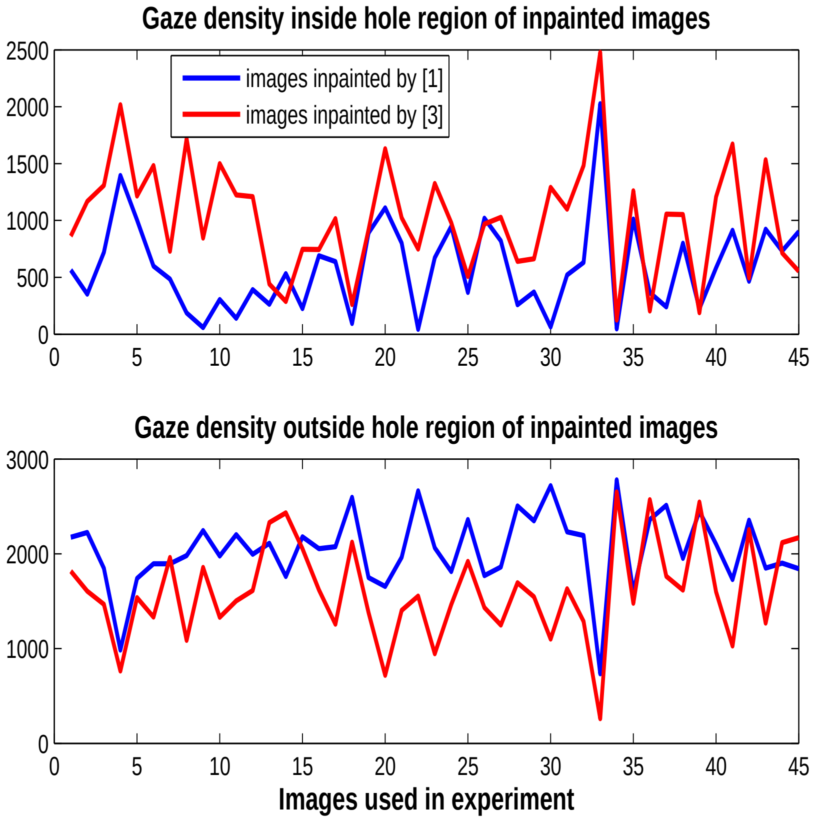


Figure 5.2: Gaze density inside and outside hole regions of inpainted images.

to the unmodified image, we compute the normalized gaze density, defined in (5.3), by calculating the ratio of gaze density of unmodified image (corresponding to the hole region Ω) to the gaze density in the hole regions of the images inpainted by [1] and [13]. Evaluating the gaze of the inpainted images to unmodified images through KL divergence would not be suitable in this task because, as mentioned previously, if there were inpainting artifacts appearing in perceptually interesting regions of image, gaze of both inpainted and unmodified image would indicate a coherence which is misleading.

$$\overline{GD_{in}} = \frac{GD_{in}^{inp}}{GD_{in}^{unmod}} \tag{5.3}$$

This normalized ratio for both the versions of the inpainted image inside the hole region is shown in Figure 5.3. This ratio gives us more insight relating attention of HVS and the perceptual quality of inpainting. It can be inferred from Fig 5.3, that the attention inside the hole region of images inpainted using [13] is consistently greater than images inpainted by algorithm in [1]. Furthermore, we can also observe that this ratio between unmodified image and images modified by [1] is closer to unity. This implies us that the gaze of the HVS when viewing images modified by [1] does not significantly deviate from normal viewing behavior when compared with images modified with [13]. This ratio can serve as a subjective discrimination measure and can be useful to predict the perceptual quality of image inpainting.

5.4 Subjective Rating and its Relation to Gaze

Based on the mouse click event, the subjective rating ranging between [0-8] received by each image is indicative of the true perceptual image inpainting quality. The ratings received by all three types of images, are presented in Figure 5.4. We can see that the lowest ratings were received by unmodified images followed by images modified by algorithm proposed [1] and [13] respectively. The median and mean rating of the images in the three different categories (real unmodified, inpainted by algorithm proposed in [1] and [13]) are [1, 3, 6] and [1.2, 3.11, 5.67] respectively. The subjective rating thus obtained clearly establishes an order of preference and is

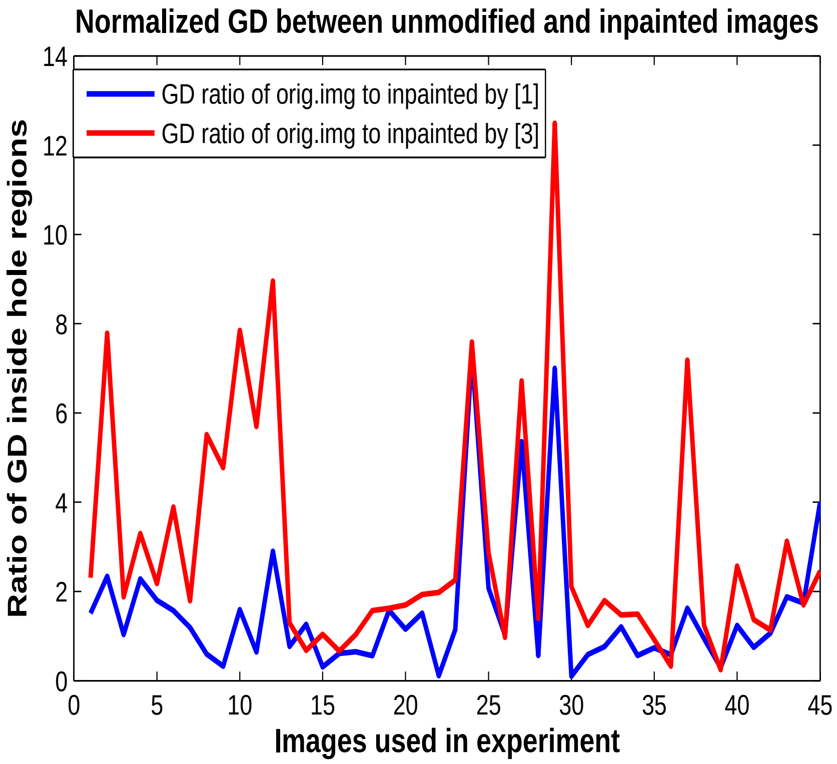


Figure 5.3: Gaze density ratio between original unmodified image and inpainted images in the hole region.

along the expected lines. It should be noted that some real unmodified images that received higher rating could possibly be due to unusual image content which may have convinced the observer to conclude that it may have been modified or even in some cases due to finger error. This behavior of observers confusing themselves has also been noted in [1].

Table 5.1 is the contingency table of comparing the normalized gaze density ratios $\overline{GD_{in}}$ and comparing the subjective ranking between [1] and [13] among all the 45 images tested. Out of the total of 45 images used in the experiment, 39 images modified by method in [13] received an equal or higher rating when compared with images modified by [1]. Among the other 6 images, which were perceived to be better inpainted, 75% (4 out of 6) of the images had the gaze density ratio lesser than

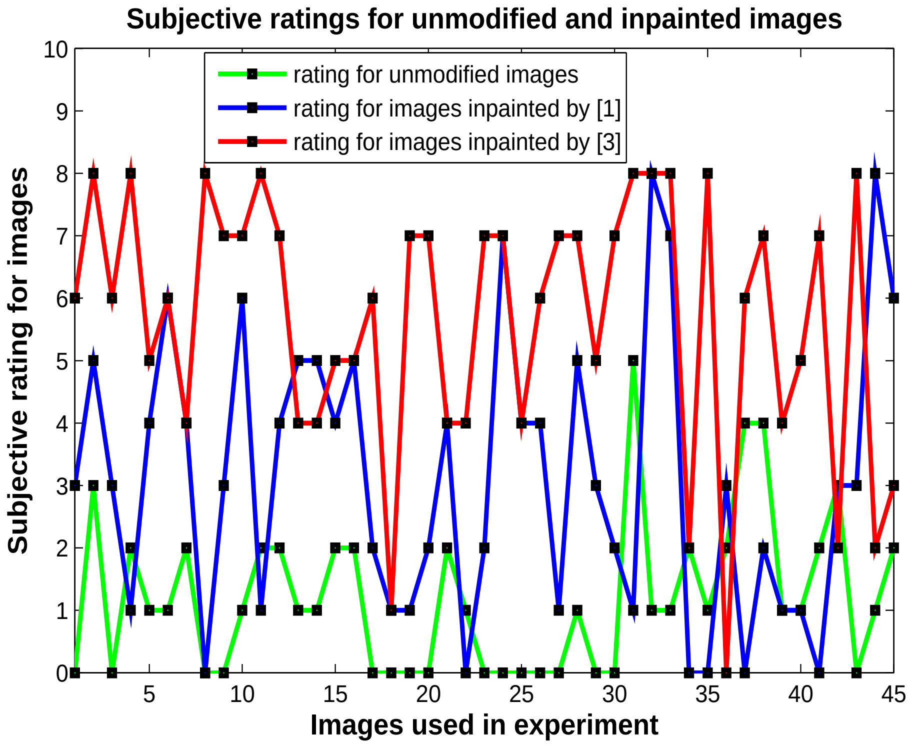


Figure 5.4: Subjective rating for real and inpainted images (lower rating indicate better perceptual quality).

that of images inpainted using [1]. The strong correlation between the subjective rankings and the gaze density data confirms the relation between the perceived image inpainting quality and the attention of the HVS.

Table 5.1: Contingency table showing the number of images in each category using the combination of the normalized gaze density ratios and subjective rankings. Numbers in the diagonal cells indicate agreements between the two measures.

Ranking	$\overline{GD_{in}}$	
	[1] $\geq$ [13]	[1] $<$ [13]
[1] $\geq$ [13]	35	4
[1] $<$ [13]	2	4

While it is cumbersome to measure eye gaze density, the significance of our results is that, with a computational visual saliency model to predict gaze patterns, our study can be used to provide a direct objective measurement of inpainting quality.

5.5 Inpainting Quality vs Viewing Time

In [1], the quality of the inpainted images were adjudged through subjective experiments in which 20 observers were instructed to identify fake images among a collection of inpainted and unmodified images. The results of those experiments which mark the proportion of images identified as fake versus the viewing time in the experiment is shown in Figure 5.5.

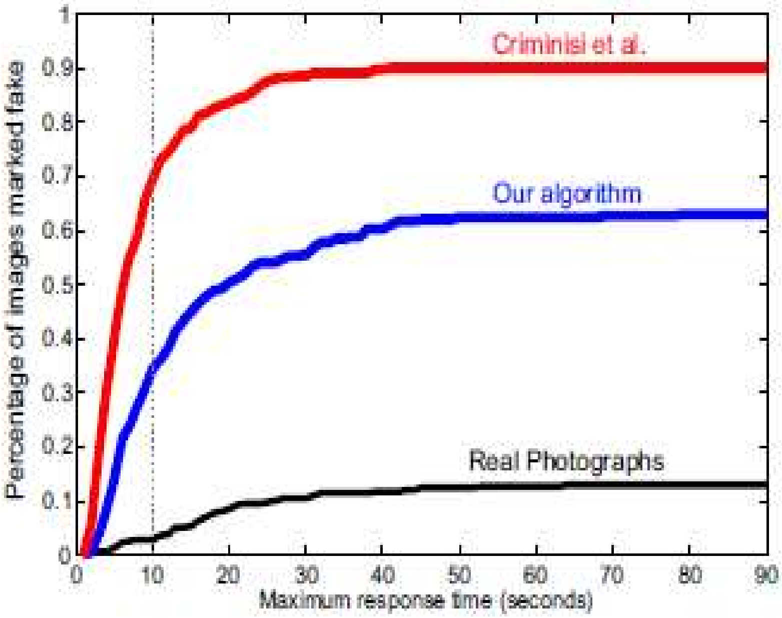


Figure 5.5: Percent image identified as fake vs Viewing Time in [1]

After 10 seconds of examination, participants have marked 34% of the images modified by their inpainting as Fake. For images modified by Criminisi et al. and unmodified images they are 69% and 3% respectively. They claim that unbiased observers are likely to make a decision regarding the authenticity of the images within 10 seconds and it is unlikely that they will find anything wrong with the image after that. Also, they note that under unlimited viewing time the participants classified their results as real 37% of the time compared with 10% for Criminisi et al. and interestingly the participants identified real images as such only 87% of the time. This is because participants scrutinized the images so carefully that they frequently convinced themselves that the real images were fake.

In our experiments with eye-tracking, corroborating the study of [1], we also arrive at a similar conclusion under our experimental settings. Here, each subject was instructed to view 45 images, equally distributed among the three sets as described in Section 5.2. The results of our experiment are presented in Figure 5.6. We observe

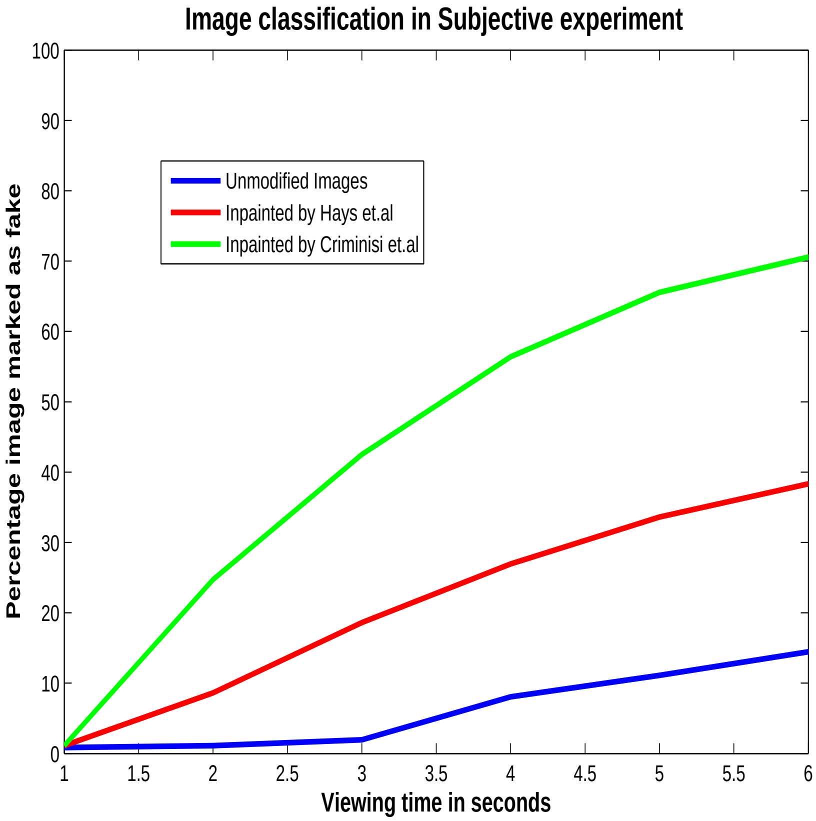


Figure 5.6: Percent image identified as Fake vs Viewing Time based on our experiments

that 71%, 39% and 12% images modified by Criminisi et.al, Hays et.al and unmodified images were classified as fake by a total of 24 observers. These numbers also relate well with the median and mean rating of the images obtained through mouse click event in the three different categories (real unmodified, inpainted by algorithm proposed in [1] and [13]) are [1, 3, 6] and [1.2, 3.11, 5.67] respectively. The higher percentage of real unmodified images being classified as fake when compared with [1] could be due to user error in the experiment because of limited time duration and also due to the unusual visual content.

5.6 Summary and Conclusions

In this work we analyzed the impact of perceptual image inpainting quality on the lower level vision process of HVS by using gaze information and a subjective evaluation score. By analyzing the gaze density inside and outside the hole region of the inpainted images and comparing it with the similar regions of the unmodified images, we observe that the image inpainting artifact do have an impact on the perceptual inpainting quality. From the analysis of the gaze patterns we can establish that when the artifacts are discernible, the distribution of gaze for the unmodified image is different from the inpainted image. It should also be mentioned that when the hole regions happen to be in the interesting regions of the image, where there is a natural attention process, the impact will be even more pronounced and further analysis is required to separately study those effects. This study provides valuable contextual information of gaze in inpainting setting which could be compared with available human visual salience models. This can potentially lead to evaluation of perceptual image inpainting quality through objective measures, reflecting the subjectivity of human observers.

Chapter 6

Conclusion and Future Work

This dissertation addressed three significant challenges associated with the existing and emerging inpainting algorithms and applications. They are 1) Structure completion for image inpainting algorithms, 2) Fast and efficient object based video inpainting framework and 3) Perceptual evaluation of large area image inpainting algorithms.

Our main contribution to structure completion aspect in image inpainting was the introduction of a boundary completion algorithm for rotationally symmetric objects under severe occlusions. We utilize the invariant nature of the curvature under similarity transform to estimate two quantities that characterize the nature of rotational symmetry, the fundamental angle of rotation and centroid of the occluded object. By using these estimated quantities and using a cost function, we arrive at an appropriate completion result. This algorithm does not make any assumption about the nature of the circular symmetries to perform the completion and can potentially be extended to handle various symmetries. The usefulness of this contour completion algorithm for structure completion in global image inpainting application was demonstrated.

We introduced a fast and computationally efficient object based video inpainting technique that differs from existing techniques by inpainting the entire objects instead of small spatio-temporal patches using available templates. The static regions to be inpainted are completed by a combination of adaptive background replacement and image inpainting technique. Our key contribution is the introduction of a computationally efficient object based inpainting method for inpainting moving foreground regions. By grouping consecutive object templates in a sliding window fashion and by defining a novel dissimilarity metric, we formulate the video inpainting process as a energy minimization problem under a dynamic programming framework. The process of grouping consecutive templates implicitly locks onto the global movement of the constituent objects and allows us to handle cases where the entire object is completely occluded. This technique can efficiently inpaint large regions of occlusions, inpaint objects that are completely missing for several frames, change in size and pose and has minimal blurring and motion artifacts. Furthermore, our object-based approach is significantly faster than the patch-based scheme and takes minutes, rather than hours as required by many existing schemes, to obtain comparable inpainting results. Our proposed scheme also offers a unified framework to address video inpainting under both stationary and limited moving camera conditions.

The perceptual quality of large area inpainting technique is inherently a subjective process and yet no previous research has been carried out by taking the subjective nature of the Human Visual System (HVS). We perform subjective experiments using eye-tracking device involving 24 subjects to analyze the effect of inpainting on human gaze. We experimentally show that the presence of inpainting artifacts directly impacts the gaze of an unbiased observer and this in effect has a direct bearing on the subjective rating of the observer. Specifically, we show that the gaze energy in the hole regions of an inpainted image show marked deviations from normal behavior when the inpainting artifacts are readily apparent. The subjective rating of images obtained from our experiment establishes a clear order of preference among different inpainted images and corroborates the results of the eye-tracking experiment. The strong coherence exhibited by the subjective ratings and the normalized gaze density give further credence to the mutual relation between the gaze and the perceptual image inpainting quality.

While we have described our efforts in tackling these three major challenges in this dissertation, nonetheless, there are many exciting challenges that lay ahead in making inpainting practical and robust in day to day applications. We highlight few possible research directions and identify challenges which merit further investigation to extend the state of art. In real images, symmetrical objects may not appear to be symmetric due to the projection from three-dimensional world to the two-dimensional image plane. We can extend our framework by modeling the projection and estimating the appropriate transform that would result in a proper completion by optimizing the cost function over the space of all projective transforms. In case of video inpainting, while we have made some progress in reducing the complexity, it nevertheless remains a very challenging task. For instance, our scheme entirely ignores the presence of shadows which results in unnatural appearance of objects. When the object to be inpainted exhibit complex, non-repetitive motion, the inpainting becomes significantly more challenging without assuming any a prior model. The variability in illumination conditions and arbitrary camera motion can also complicate the inpainting performance. Also our inpainting technique is not extremely amenable to inpaint small hole regions. By using a multi-resolution based analysis and combining it with a patch based process to handle such cases we can further improve the efficiency and applicability of video inpainting algorithms. Perceptual evaluation of image inpainting is a formidable challenge due to effects of high level cognitive process involved. The findings of our perceptual evaluation of inpainting techniques yielded valuable information related to the quality of the inpainting. Combining this information with low level and mid level image features we are currently involved in introducing a objective computational model for automatically predicting the perceptual image inpainting quality.

Bibliography

[1] James Hays and Alexei A Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), 2007.

[2] G. Emile-Male. *The Restorer's Handbook of Easel Painting*. Van Nostrand Reinhold, 1976.

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of ACM Conf. Comp. Graphics (SIGGRAPH)*, pages 417–424, New Orleans, USA, July 2000.

[4] A. Senior et.al. Blinlering surveillance: Enable video privacy through computer vision. Technical report, IBM, Research report, August 2003.

[5] A. M. Berger. *Privacy mode for acquisition cameras and camcorders.* Sony Corporation, us patent 6,067,399 edition, May 23 2000.

[6] J. Wada, K. Kaiyama, K. Ikoma, and H. Kogane. *Monitor camera system and method of displaying picture from monitor camera thereof.* Matsushita Electric Industrial Co. Ltd., european patent, ep 1 081 955 a2 edition, April 2001.

[7] J. Schiff, M. Meingast, D. Mulligan, S. Sastry, and K. Goldberg. Respectful cameras: Detecting visual markers in real-time to address privacy concerns. In *International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[8] W. Zhang, S.-C. Cheung, and M. Chen. Hiding privacy information in video surveillance system. In *Proceedings of the 12th IEEE International Conference*

on Image Processing (ICIP), volume 3, pages 868–871, Genova, Italy, September 2005.

[9] J. Wickramasuriya, M. Datt, S. Mehrotra, and N. Venkatasubramanian. Privacy protecting data collection in media spaces. In *ACM International Conference on Multimedia*, pages 48–55, New York, NY, October 2004.

[10] E. N. Newton, Latanya Sweeney, and B. Main. Preserving privacy by de-identifying face images. *IEEE transactions on Knowledge and Data Engineering*, 17(2):232–243, February 2005.

[11] H. Wactlar, S. Stevens, and T. Ng. *Enabling Personal Privacy Protection Preferences in Collaborative Video Observation.* NSF Award Abstract 0534625, http://www.nsf.gov/awardsearch/showAward.do?awardNumber=0534625.

[12] K. P. Karmann and A. Brandt. Moving object recognition using an adaptive memory background. In V. Cappellini, editor, *Time-Varying Image Processing and Moving Object Recognition*, pages 289–307. Elsevier Science Publishers, 2nd edition, 1990.

[13] A. Criminisi, Patrick Perez, and Kentaro Toyama. Region filling and object removal by exemplar-based inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, September 2004.

[14] M. Vijay Venkatesh and Sen ching S. Cheung. Symmetric shape completion under severe occlusions. *International conference on image processing (ICIP)*, pages 7009–712, October 2006.

[15] M. Bertalmio, A.L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume I, pages 355–362, Hawaii, 2001.

[16] K. A. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting under constrained camera motion. *IEEE Transactions On Image Processing*, 16(2):545–553, Feb 2007.

[17] M. Vijay Venkatesh, S.-C. Cheung, and J. Zhao. Efficient object based video inpainting. *Pattern Recognition Letters*, 30(2):168–179, 2009.

[18] Sen-Ching S. Cheung, Jian Zhao, and M. Vijay Venkatesh. Efficient object-based video inpainting. In *Proceedings of International Conference on Image Processing (ICIP)*, pages 705–708. IEEE, 2006.

[19] Paul Ardis and Amit Singhal. Visual salience metrics for image inpainting. *Proceedings of the SPIE*, 7257, 2009.

[20] Lior Elazary and Laurent Itti. Interesting objects are visually salient. *Journal of Vision.*, 8(3):1–15, 2008.

[21] L. Itti and P. F. Baldi. Bayesian surprise attracts human attention. In *Advances in Neural Information Processing Systems, Vol. 19 (NIPS*2005)*, pages 547–554, 2006.

[22] M. Vijay Venkatesh and Sen ching S. Cheung. Eye tracking based perceptual image inpainting quality analysis. *International conference on image processing (ICIP)*, October 2010.

[23] A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International conference on Computer Vision (ICCV)*, pages 1033–1038, Corfu, Greece, September 1999.

[24] A. Efros and W.T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of ACM Conf. Comp. Graphics (SIGGRAPH)*, pages 341–346, August 2001.

[25] P. Harrison. A non-hierarchical procedure for re-synthesis of complex texture. In *WSCG Winter School of Computer Graphics Conf. Proc. (WSCG)*, pages 190–197. Univ. of West Bohemia, 2001.

[26] D.J. Heeger and J.R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of ACM Conf. Comp. Graphics (SIGGRAPH)*, volume 29, pages 229–233, Los Angeles, CA, 1995.

[27] M. Ashikhmin. Synthesizing natural textures. In *Proceedings of ACM Symposium on Interactive 3D Graphics*, pages 217–226, Research Triangle Park, NC, March 2001.

[28] H. Igehy and L. Pereira. Image replacement through texture synthesis. In *Proceedings of International Conference on Image Processing (ICIP)*, volume III, pages 186–190, 1997.

[29] Hitoshi Yamauchi, Jörg Haber, and Hans-Peter Seidel. Image restoration using multiresolution texture synthesis and image inpainting. In *Computer Graphics International (CGI 2003)*, pages 120–125, Tokyo, Japan, July 2003. IEEE.

[30] C.W. Fang and J.J.J. Lien. Fast image replacement using multi-resolution approach.

[31] T. Chan and J. Shen. Mathematical models for local deterministic in-paintings. Technical Report CAM TR 00-11, UCLA, March 2000.

[32] T. Chan and J. Shen. Non-texture in-painting by curvature-driven diffusions (cdd). Technical Report CAM TR 00-35, UCLA, March 2000.

[33] David Tschumperl and Richard Deriche). Vector-valued image regularization with pde's : A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):506–517, 2005.

[34] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. In *Proceedings of ACM Conf. Comp. Graphics (SIGGRAPH)*, volume 22, pages 303–312, July 2003.

[35] Chih-Wei Fang and Jenn-Jier James Lien. Rapid image completion system using multi-resolution patch-based directional and non-directional approaches. *IEEE Transactions on Image Processing*, 18(11), 2009.

[36] Aude Oliva and Antonio Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research: Visual perception*, pages 155: 23–36, 2006.

[37] P. Prez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics (SIGGRAPH'03)*, 22(3):313–318, 2003.

[38] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, August 2003.

[39] J.-L. Starck, M. Elad, and D.L. Donoho. Image decomposition via the combination of sparse representation and a variational approach. *IEEE Transaction on Image Processing*, 2005. To appear.

[40] M. Elad, J.-L Starck, D. Donoho, and P. Querre. Simultaneous cartoon and texture image in-painting using morphological component analysis (mca). *Applied and Computational Harmonic Analysis*, 2005. To appear.

[41] Jiaya Jia and Chi keung Tang. Inference of segmented color and texture description by tensor voting. *IEEE Transactions Pattern Analysis and Machine Intelligence (PAMI)*, 26(6):771–786, June 2004.

[42] Yining Deng and b. s. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 23(8):800–810, 2001.

[43] G. Medioni, Mi-Suen Lee, and Chi-Keung Tang. *A Computational Framework For Segmentation And Grouping*. Elsevier, 2000.

[44] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. In *Proceedings of ACM Conf. Comp. Graphics (SIGGRAPH)*, volume 24, pages 861–868, July 2005.

[45] M. Oliviera, B. Bowen, R. McKenna, and Y.-S. Chang. Fast digital image inpainting. In *Proc. of Intl. Conf. on Visualization, Imaging and Image Processing (VIIP)*, page 261266, 2001.

[46] A. Telea. An image in-painting technique based on the fast marching method. *Journal of Graphics Tools*, 9, 2004.

[47] Shantanu Rane, Guillermo Sapiro, and Marcelo Bertalmío. Structure and texture filling-in of missing image blocks in wireless transmission and compression applications. In *IEEE Transactions on Image Processing*, volume 12, pages 296–303, 2003.

[48] Yasuyuki Matsushita, Eyal Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1150–1163, 2006.

[49] V.Cheung, Brendan. J. Frey, and Nebojsa Jojic. Video epitomes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 42–49, 2005.

[50] K. Patwardhan, G. Sapiro, and M. Bertalmio. Video in-painting of occluded and occluding objects. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 69–72, 2005.

[51] Yunjun Zhang, Jiangjian Xiao, and Mubarak Shah. Motion layer based object removal in videos. In *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision*, volume 1, pages 516–521, 2005.

[52] Takaaki Shiratori, Yasuyuki Matsushita, Sing Bing Kang, and Xiaoou Tang. Video completion by motion field transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 411–418, June 2006.

[53] Yuping Shen, Fei Lu, Xiaochun Cao, and Hassan Foroosh. Video completion for perspective camera under constrained motion. In *Proceeding of the International Conference on Pattern Recognition (ICPR)*, volume 3, pages 63–66, 2006.

[54] Jiaya Jia, Yu-Wing Tai, Tai-Pang Wu, and Chi-Keung Tang. Video repairing under variable illumination using cyclic motions. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 28, pages 832–839, May 2006.

[55] Y. T. Jia, S. M. Hu, and R. R. Martin. Video completion using tracking and fragment merging. In *Proceedings of Pacific Graphics*, volume 21, pages 601–610, 2005.

[56] B. Girod. What's wrong with mean-squared error. In A.B. Watson, editor, *Digital Images and Human Vision*, chapter 15, pages 207–220. MIT Press, 1993.

[57] T.F. Chan and S.H. Kang. Error analysis for image inpainting. *Journal of Mathematical imaging and Vision*, 26(1-2):85–103, November 2006.

[58] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, November 1998.

[59] Yanxi Liu, Robert Collins, and Yanghai Tsin. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Trans. on Pat. Anal. and Mac. Int.*, 26(3):354 – 371, March 2004.

[60] Marco Zuliani, S. Bhagavathy, B.S. Manjunath, and C.S. Kenney. Affine invariant curve matching. In *International conference on Image Processing, ICIP 2004*, pages 3041– 3044, 2004.

[61] Ross Cutler and Larry S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):781–796, 2000.

[62] Thomas K. Leung and Jitendra Malik. Detecting, localizing and grouping repeated scene elements from an image. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume I*, pages 546–555, London, UK, 1996. Springer-Verlag.

[63] Frederik Schaffalitzky and Andrew Zisserman. Geometric grouping of repeated elements within images. In *Shape, Contour and Grouping in Computer Vision*, pages 165–181, London, UK, 1999. Springer-Verlag.

[64] Branko Grunbaum and C. Shephard. *Tilings and Patterns.* W. H. Freeman Co. New York, NY, USA, 1986.

[65] Hugo F. Verheyen. *Symmetry Orbits*. Birkhauser, 1996.

[66] Peter S. Stevens. *Handbook of Regular Patterns: An Introduction to Symmetry in Two Dimensions*. The MIT Press, Boston, 1981.

[67] Willard Miller. *Symmetry Groups and their applications*. Academic Press, New York, 1972.

[68] George E. Martin. *Transformation Geometry: An introduction to Symmetry*. Springer Verlag, New York, NY, USA, 1982.

[69] M. A. Armstrong. *Groups and Symmetry*. Springer Verlag, New York, NY, USA, 1997.

[70] H. Zabrodsky, S. Peleg, and D. Avnir. Symmetry as a continuous feature. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 17(12):1154–1166, 1995.

[71] M. Vijay Venkatesh and Sen ching S. Cheung. Symmetric shape completion algorithms. Technical report, University of Kentucky, 2006.

[72] T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proc. IEEE Frame-Rate Applications Workshop*, 1999.

[73] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using non-parametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90:1151–1163, 2002.

[74] H.263. *Video Coding for Low Bitrate Communication.* ITU-T Recommendation H.263 Version 2, 1998.

[75] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):463–476, 2007.

[76] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2004.

[77] Matei Mancas. *Computational Attention: modelisation and application to audio and image processing.* PhD thesis, Faculté Polytechnique de Mons, 2007.

[78] Andrew T. Duchowski. *Eye Tracking Methodology: Theory and Practice.* Cambridge University Press, first edition, 2003.

[79] Alex Poole, Linden. J. Ball, and Peter Phillips. In search of salience: A response time and eye movement analysis of bookmark recognition. In David Moore Sally Fincher, Panos Markopoulos and Roy Ruddle, editors, *People and Computers XVIII Design for Life*, pages 363–378. Springer London, 2007.

VITA

Vijay Venkatesh Mahalingam was born on Oct 4th 1977 in Madurai, India. He completed his under graduation with distinction in Electronics and Communication Engineering from Thiagarajar College of Engineering, Madurai. After gaining work experience in industry for a couple of years, he ventured to complete his graduate program in Electrical Engineering at the University of Kentucky and his graduate research was in Image quality analysis which was funded by Lexmark International. He continued his studies further and his research has been published in major conferences and journals. His research interests are in Signal and Image processing, Computer Vision and Pattern recognition areas and he is a member of IEEE.