Layer Extraction, Removal and Completion of Indoor Videos: A Tracking Based Approach

A Thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science (by Research)
in
Computer Science

by

Vardhman Jain
200507020
vardhman@students.iiit.ac.in

International Institute of Information Technology
Hyderabad, INDIA
August, 2006

Copyright © Vardhman Jain, 2006
All Rights Reserved

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Layer Extraction, Removal and Completion of Indoor Videos: A Tracking Based Approach” by Vardhman Jain, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Advisor: Prof. P. J. Narayanan

To my Parents, Didi and Rishabh

Acknowledgements

It is a pleasure to thank the people who made this thesis possible. This includes all my well wishers, by which I mean family, friends, professors and others who directly or indirectly have taught me something throughout my life. I am grateful to all of them.

I am thankful to Dr. P. J. Narayanan for his support and guidance throughout. His great knowledge and enthusiasm has always been a great source of inspiration and motivation. His willingness to allow me explore various fields and areas and have fruitful discussions on various aspects of research has been very helpful. Without his guidance the thesis would have never been possible.

I am specially thankful to Dr. P. J. Narayanan and Dr. C. V. Jawahar for kindling my interest in the field of Computer Vision, Pattern Recognition and Image Processing and for encouraging me to pursue higher education by providing a motivating and learning based research environment in form of our lab, CVIT. I am thankful for the support from the lab during my Master's degree. I thank all my peers and seniors for the fruitful discussions and suggestions throughout my stay at the institute.

I am grateful to my institute IIIT and its faculty for providing an excellent place, environment that enabled me to build a strong foundation in Computer Science during my Bachelor's degree and also inculcated many values in me.

Lastly and most importantly, I would like to thank my family, parents and sister, from whom I have learned the values of hard work, high standards and to focus on what is important in life. My sister has been my constant guide and motivator throughout. My achievements are their achievements, and I am grateful for the education and support they gave me, which is their most valuable and most enduring gift. I dedicate this thesis to them.

Abstract

Image segmentation and layer extraction in video refer to the process of segmenting the image or video frames into various constituent objects. Automatic techniques for these are not always suitable, as the objective is often difficult to describe. With the advent of interactive techniques in the field, these algorithms are now usable for selecting an object of interest in an image or video precisely with less efforts. Object segmentation brings up various other possibilities like cut and paste of objects from one image or video to another.

Object removal in image and videos is another application of interest. As the name suggest the task is to eliminate an object from the image or video. This involves recovering the information of the background previously occluded by the object. Object removal in both image and videos have found interesting applications especially in the entertainment industry. The concept of filling-in of information from the surrounding region for images and surrounding frames for videos has been applied for recovering damaged images or clips.

This thesis presents two new approaches. The first is for object segmentation or layer extraction from a video. This method allows segmenting complex objects in videos, which can have difficult motion model. The algorithm integrates a robust points tracking algorithm to a 3D graph cuts formulation. Tracking is used for propagating the user given seeds in key frames to the intermediate frames which helps to provide better initialization to the graph cuts optimization. The second is an approach for video completion in indoor scenes. We propose a novel method for video completion using multiview information without applying a full frame or complete motion segmentation. The heart of the algorithm is a method to partition the scenes into regions supporting multiple homographies based on a geometric formulation and thereby providing precise segmentation even at the points where the actual scene information is missing due to the removal of the object. We demonstrate our algorithms on a number of representative videos. We also present a few directions for future work that extends the work presented here.

Contents

List of Figures

Chapter 1

Introduction

With the increase in the storage capabilities, images and videos are increasingly becoming very popular form of stored data. A large number of handhelds available in the market have ability to capture images and videos. The increasing availability of such form of data increases the need for ways of manipulating the captured information to suit the user demands. As a result, image and video manipulation has been an active topic of research in the recent past. Many techniques which were traditionally used in motion pictures for creating special effects mainly by experienced artists on specialized hardware are now becoming more and more automated and easy to use. The special setup and hardware is slowly being replaced by more advanced algorithms in software. Many interesting video editing applications have been demonstrated recently including video sprites [64], video textures [65], video matching [63], motion magnification [48], video synthesis and editing [8] and automatic photo pop-up [30].

As computing power has increased over the years the algorithms have become faster and real time. The improved speed of the algorithm makes user interaction possible while the algorithm is in action. Most image manipulation tools like Adobe Photoshop 7 [1] or GIMP [60] provide users with brushes to perform various actions interactively rather than performing them automatically. The trend is to find methods to improve the output with some user interaction rather than finding automatic methods with less than optimal output. This has been illustrated by various recently proposed techniques like [3, 42, 46, 47, 57, 71, 72, 77].

One of the interesting application of the image and video manipulation techniques is object segmentation from an image or video. The segmented object can be pasted over a new background. The problems of image segmentation and grouping remain great challenges to computer vision. Despite a lot of work in the area the algorithms for image segmentation are still not as successful and versatile as that of edge detection or other low-level vision problems. It is even believed that the problem of image segmentation is an ill-defined one, as the expected output is not well defined. The technique of extracting an object with precise boundary information is known as *Matting* while that of pasting it over a new background is termed as *Compositing*. These techniques find interesting uses in creating both natural and unrealistic scenes. The area has have come a long way from traditional blue screen matting with specialized setup to current natural image matting where no specialized setup is needed. In recent years,

these techniques have been extended to videos. An object's trajectory over the frames is termed as a *layer*. The process of extracting a layer from a video is therefore called *layer extraction*. An object can be extracted from a video as a layer and directly composited on a different video. These techniques find use in the entertainment industry where the actors commonly perform in front of a studio background and are later composited into complex environments.

The extraction of objects from images and videos has another interesting application, *viz* object removal. An unwanted object in the image or video is removed and background is recovered so as to make the image or video look natural. Object removal is very regularly used in film post production to improve the composition of the scene. Object removal is also used to repair bad films in which case the destroyed part of the frame is removed. Interactive techniques to remove objects can also be beneficial to individual users to clean up their amateur videos offline.

1.1 Layer Extraction

Layer extraction has been a topic of research in recent years. Many techniques have been proposed for automatic segmentation of layers [39, 66, 78, 81]. Automatic segmentation of video is useful in many application like compression, coding, recognition [81]. Interactive segmentation of images [47, 61] and videos [46, 77] has developed recently. The superior quality they achieve with minimal user interaction makes them very attractive. These segmentation methods have objectives similar to those of layer extraction. The extracted layers can be used in many applications of advanced video editing including matting and compositing. The problem is closely related to the object tracking problem which also has received lot of attention over the years.

The experience from the domain of image segmentation has been used in video layer extraction approaches to a large extent. Many techniques proposed for the task [39, 81, 82] directly or indirectly depend on clustering of motion vectors across frames similar to use the color values used in case of image segmentation techniques. Graph cuts have also emerged recently as a popular method for segmentation of images [11]. The success of image segmentation techniques have motivated their application to videos [46, 77].

One way to extract layers in a video is to segment each frame independently. There are certain issues which discourage the use of such techniques:

1. The object's segmentation over individual frames may not provide temporal continuity.

2. The segmentation information obtained in earlier frames is not used.

3. The technique will be slow due to the huge amount of re-computation at every frame.

We try to address these problems in this thesis. We propose a method in Chapter 3 based on the assumption that objects in videos usually exhibit small motions over frames and also the frames are temporally highly related. We use a multi-frame graph which helps maintain temporal continuity and

leverage the segmentation obtained in one frame for later frames. We also prune a large part of the frame from being a part of the minimization process, making the graph smaller in terms of number of nodes and edges. Robust tracking provides hard constraints in the target frame which act as good seed points for the graph cuts minimization of the next frame. We also use iterative graph cuts algorithm during the interactive correction to make the interaction fast and real time. Together, our algorithm provides mostly automatic and accurate layers.

The layer obtained by our approach can then be used for variety of other applications like video cutout, matting, compositing and object removal etc. The object's mask found using our method can be used to produce a trimap input to Bayesian matting [15] technique to find the precise alpha values for the boundaries of the object. Our approach produces the output similar in quality to that of other video matting techniques [14, 53].

1.2 Object Removal and Video Completion

Segmenting and removing objects from images or videos is of much current interest. Object removal leaves the image or video with unknown information where the object was earlier placed. Missing information recovery in images is called *inpainting*. This is accomplished by inferring or guessing the missing information from the surrounding regions. For videos, the process is termed as *completion*. Video completion uses the information from the past and the future frames to fill the pixels in the missing region. When no information is available for some pixels, inpainting algorithms are used to fill them. Video completion has many applications. Post-production editing of professional videos in creative ways is possible with effective video completion techniques. Video completion is perhaps most useful for with home videos. Video can be cleaned up by removing unnecessary parts of the scene and filling the gaps correctly. Inpainting and video completion is often interactive and involve the users as the objective is to provide desirable and appealing output.

Image inpainting inevitably requires approximation as there is no way of obtaining the missing information. For videos, the missing information in the current frame may be available from nearby frames. Significant work has been done on inpainting and professional image manipulation applications and tools exist to accomplish the task to various degrees. The solution to the problem of object removal in video depends also on the scene complexity. Most video completion work has focused on scenes in which a single background motion is present such as an outdoor scene. In scenes with multiple large motion, motion layer segmentation methods are used to obtain different motions layers. A particular layer can be removed by filling the information with the background layers. Another common approach is the interpolation of the motion flow vectors of the unknown region from the surrounding regions. Scenes with multiple motion, such as indoor scenes, are challenging to these algorithms. For scenes with many planes, motion model fitting may not be suitable as the boundaries between the layers are not exact. This is especially problematic for video completion as the region being filled could straddle

these boundaries. Periodicity of motion is also often used by techniques which fill the holes by patching from some other part of the video.

In Chapter 4, we present a technique for video completion for indoor scenes. We concentrate on scenes where the background motion consists of two or three planes in the neighborhood of the object to be removed. Our main contribution is the use of the geometry of intersecting planes in multiple views for motion segmentation, without applying a dense motion segmentation in the image. We also show that segmentation of only the nearby background around the missing region is sufficient for the task of video completion. Full-frame motion segmentation can thus be avoided. The geometric nature of the method ensures accurate and unique background assignment to the pixels in the unknown region, which to the best of our knowledge is not possible with other video completion methods. We particularly concentrate on scenes where the neighborhood around the object to be removed is planar in nature.

1.3 Contributions of the Thesis

This contributions of the thesis are in presenting two new approaches. The first is an object segmentation or layer extraction technique for a video. This method allows segmenting complex objects in videos, which can have difficult motion models. The algorithm integrates a robust point tracking algorithm and a 3D graph cuts formulation. Tracking is used for propagating the user-given seeds in key frames to the intermediate frames which helps to provide better initialization to the graph cuts process. The second contribution is an approach for video completion in indoor scenes. We propose a novel method for video completion using multiview information without applying a full frame or complete motion segmentation. The heart of the algorithm is a method to partition the scenes into regions supporting multiple homographies based on a geometric formulation and thereby providing precise segmentation even at points where the actual scene information is missing due to the removal of the object. We demonstrate our algorithm on a number of representative videos. We also present a few directions for future work that extends the work presented here.

1.4 Organization of the Thesis

This chapter describes a general introduction to the problems we attempt to solve in the thesis. The importance of the two problems, namely object segmentation and removal in videos is described.

Chapter 2 discusses the related and previous work in the field of layer extraction and object removal. We review the related topics like image segmentation, semi-interactive image segmentation, interactive image segmentation, matting, layer extraction, motion segmentation, image registration, image inpainting, texture synthesis, image completion and video completion or object removal. The chapter provides a detailed review of various techniques which have been proposed over the years. The discussion provides a background into understanding the general problems and issues faced in solving the problem and the techniques which evolved to overcome them.

In Chapter 3, we discusses the details of our algorithm for layer extraction or object segmentation. The problem of layer extraction is introduced in details there. The focus of the chapter is on our approach and the advantages it provides over the current method for object segmentation in videos.

In Chapter 4, we describe the details of our algorithm for object removal in videos. We demonstrate our novel approach of video completion which uses purely geometrical method and doesn't involve any approximation or interpolation.

We derive some conclusions and discuss areas for future work in Chapter 5.

Chapter 2

Related work

The work presented in this thesis spans two related problems. First is the object segmentation or layer extraction in videos which deals with extracting out the set of pixels satisfying certain homogeneity criteria such as color or motion from all the frames. The second problem is that of object removal from the video where we remove the object from the video and fill the pixels belonging to the object by the background information such that the video looks plausible.

Layer extraction problem is closely related to problems like image segmentation, object segmentation in videos, image and video matting, interactive image editing, video editing and object removal [83]. Our video completion is closely related to a few well studied problems these include Image registration, Inpainting and texture synthesis. In many cases the algorithm for completion of videos is considered as an extension to an image completion algorithm. We provide a brief review of the related work in these domains before discussing about the work in the field of video completion and object removal.

2.1 Image Segmentation

The problem of image segmentation has been studied for a long time. The automated techniques are based on clustering the image pixels based on a similarity criterion, which includes intensity or color similarity and spatial coherence. Methods in this category were the earliest to be proposed. These include Watershed segmentation [76] and Mean Shift segmentation [17]. Watershed segmentation visualizes the image as a surface with the gray values or intensities at any particular pixel representing the height of the surface at that point. The algorithm involves finding the points on the surface which are local minima in the regions.

Mean shift segmentation models the problem of segmentation as clustering in the feature space while giving importance to the image domain information also. The significant features in images correspond to regions with high density. First a radius and an initial location is chosen for the search window. The algorithm then computes the mean shift vector and translates the search window by that amount in each iteration until the mean shift vector is close to zero which represents a mode for the cluster. This algorithm has the advantage of not requiring to know the number of final clusters. However, the

clustering decision is highly affected by color similarity which is used as the homogeneity criteria for clustering.

More recently, Felzenszwalb and Huttenlocher [21] proposed an efficient graph-theoretical method for segmentation, where the image is represented by a graph in which each pixel is a node and the edges connect the neighboring pixels with weights proportional to their dissimilarity such as difference in intensity, color, motion, location or some other local attribute. Their predicate for evaluating the existence of boundary between two region is based on measuring inter-region and intra-region dissimilarity between pixels similart to the one proposed by Shi and Malik in *normalized cuts* [67]. The basic problem with the automated methods is setting of some parameter for thresholding or weighting various terms which in general is non intuitive and very specific to the image under consideration.

All the algorithms discussed above belong to the category of bottom-up approaches, owing to the generally output of these algorithms a new set of algorithms called the top bottom approaches were proposed. Top-down approaches try to solve the image segmentation problem in a class specific sense. These algorithms fit a deformable model of a known object for e.g a horse to the image. The shape of the deformed model gives an estimate of the desired segmentation. Our method is itself a bottom-up approach. We refer the reader to [44, 58] for a details of some top-down approaches.

2.2 Semi-interactive Image Segmentation

Methods like image snapping [25] and intelligent scissors [54] in Adobe Photoshop [1] allow users to obtain a contour around the object boundary by roughly tracking the object's boundary with the mouse. As the mouse is moved across the contour the plausible boundary is calculated. If the boundary is not satisfactory new seed points are added by the user. These methods rely on local features like gradient information and Laplacian zero-crossing measures.

2.3 Interactive Image Segmentation

Recently techniques like Interactive image segmentation [11], Lazy Snapping [47], and GrabCut [61] have demonstrated that with small user input the segmentation of an image can be driven according to higher level context rather than the automatic color based segmentation techniques. The interactive segmentation methods provide an easy way of segmenting complex objects in an image, which would otherwise require tedious boundary selection.

Most interactive techniques are based on graph cuts [10, 11]. In graph cuts based techniques, a graph $G = (V, E)$ is constructed such that the set V includes all the pixels in the image whereas E is the set of edges connecting these pixels, similar to [21]. The objective is expressed in terms of minimization of the energy which is defined as the sum of a data term and an smoothness term. Boykov and Jolly [11] modeled the data term by pixel similarity to background or foreground using gray scale histogram. The smoothness term is defined as the dissimilarity between two connected pixels. User interaction is needed

to first provide the seed points for foreground and background regions from which the foreground and background histograms are evaluated. User can also interactively improve the output of the optimization by providing extra strokes and running an iterative optimization on the same graph again.

GrabCut [61] improves the interactive segmentation technique [11] first by making use of Gaussian Mixture Model(GMM) to allow segmentation process in color domain instead of intensity histograms. Secondly, the one shot graph cut minimization is replaced by a more powerful iterative procedure, which iterates between estimation and parameter learning. Finally the user interaction requirements are relaxed as user only needs to provide the background seeds.

GMMs are used for modeling the foreground and background regions in color space in [15,29,46,62]. Color space is more discriminative compared to gray scale and GMMs provide a compact representation the of color values compared to color histograms. Color histograms have a large number of bins with small frequency each bin.

2.4 Matting and Compositing

Matting is the process of obtaining accurate alpha values at the boundaries of the object, called the the alpha matte. The problem is solving the equation :

$$C = \alpha * F + (1 - \alpha) * B \tag{2.1}$$

where F and B represent the foreground and the background color the pixel whose composited color is C and α represents the alpha value. F, B and α constitute the seven unknowns. The problem is under-constrained the number of equations is only three.

In blue screen matting technique [70], the desired foreground image is separated from a constant or almost constant backing color, which has mostly been blue, thus giving the method the name. The knowledge of the background color helps reducing the number of variables from the original seven to four.

Natural matting involves solving Equation 2.1 in a general case. In most natural matting systems, the user specifies a trimap to the system specifying pixels which are (i) 100% foreground ($\alpha = 1$) (ii) 100% background ($\alpha = 0$) or (iii) unknown, i.e., for which the alpha is to be determined. The system then estimates the α values for the unknown region. Ruzon and Tomasi [62] model the foreground and background colors as a mixture of Gaussians for which the distribution P(F) and P(B) is learnt using surrounding samples for an unknown point, the α value is then estimated as coming from an intermediate distribution P(C), somewhere between foreground and background distributions. The intermediate distribution is also defined as a sum of Gaussians, each Gaussian is centered at a distinct mean value $\bar{C}$ located fractionally (according to a given alpha) along a line between the mean of each foreground and background cluster pair with fractionally interpolated covariance Σ_C. The optimal alpha is the one that yields an optimal alpha is the one that yields an intermediate distribution for which the observed color has the maximum probability. Chuang *et al* [15] model color distributions probabilistically and alpha is

obtained by finding a maximum a posteriori(MAP) estimate of the F, B and α values at a pixel given the value of C. This requires modelling the probability distribution of the foreground and background color from the nearby known foreground and background regions.

Poisson Matting [71] models the matting problem as a combination of automated global Poisson matting and interactive local Poisson matting. Global Poisson matting is based on the idea that the gradient of the foreground and background is very small compared to gradient of the alpha matte of the image. This assumption doesn't give very precise α-values at points where F and B have strong gradients and the method requires substantial application of the manual brushing tool or a local Poisson matting step. The main requirement for most matting systems is the specification of proper trimap input. Matting techniques can be applied in cascade to our layer extraction method to obtain fine mattes after the layers are extracted.

Compositing is easy once the precise alpha values are available at the boundary but methods like and do not perform very well on highly textured regions where they can easily choose the wrong directions [61]. Poisson editing [59] method allows seamless cloning of two images. An object can be selected imprecisely in source image and then transfered to the destination image so it merges with the background seamlessly. This method avoids the use of matting of the objects to be moved to the destination image. The smooth mixing is implicitly determined by the method.

Advances in the methods for image segmentation and matting [15, 71] have motivated the researchers to provide similar techniques for videos. Chaung *et al*. proposed video matting [14], where they propagate the user given trimaps for the key frames to the intermediate frames and apply image matting technique on each frame. As discussed by Li *et al* [46], the dense optical flow can not be accurately determined for all pixels and errors creep in. Other techniques like Interactive Video Cutout [77] and Video Object Cut and Paste [46] allow extraction of the object from a video. Wang *et al* [77] proposed taking user inputs for seeding across the set of frames via a special user interface. A 3D graph is constructed by using pixel, region and volume level nodes instead of only pixel nodes using a hierarchical mean shift clustering [16] based on color similarity criteria. The graph cut minimization on this 3D graph provides the segmentation for the video. The method provides for real-time correction via interactive graph cuts. Li *et al*'s approach is similar to Wang *et al* except for the use of only 2D regions as the nodes and a method to improve the segmentation obtained by region level 3D graph cuts on video by a pixel level 2D graph cuts on selective sub window of each frame.

2.5 Layer Extraction

Layer extraction methods usually rely on motion model estimation for a set of regions followed by a clustering technique to cluster regions with similar motion models. In one of the earliest work on layer extraction, Adelson and Wang [78] proposed the patch-wise motion model estimation followed by clustering of patches with similar motion model. Ke and Kanade [39] formulated the problem of layer extraction by first expanding the seed region into initial layers and then clustering them in a lower

dimensional subspace. Xiao *et al*. [81] proposed a technique for layer extraction by first obtaining regions of seed correspondence and then growing them to arbitrary shapes using the graph cuts approach integrated with level sets based formulation. The reader is suggested to refer to [81] for a more detailed survey of layer extraction work.

Most of these techniques [39, 78, 81] target at automated layer extraction and in theory assume the existence of a prominent single motion model for a layer. In practice, the object that we want to segment from the video may not show consistency in motion model across its spread. Human motion is a typical example. Layer extraction is closely related to motion segmentation, which we discuss in Section 2.6.

Interactive methods are sometimes more suitable because user can guide the output to the desired. For instance, the shadow of the object may possess the same motion model as the object but the user might like to exclude it from the foreground layer. Purely automatic techniques find this case difficult to handle as shown in Figure 3.1. The method we propose is suitable for handling reasonably fast inter-frame motion for an object. The point based tracking ensures that the seeds are available over frames even if the layer's shape is changing quite often. This setup would require large number of key frames in the usual 3D graph cuts setting.

2.6 Motion Segmentation

Traditional motion-based segmentation methods employ only motion information which allows the handling of only rigid motion or piecewise rigid motion. Recently, techniques employing spatio-temporal segmentation techniques have been proposed. These techniques employ both motion and spatial information for segmentation. The advantage of these methods is that application of both avoids over-segmentation, which is typical to segmentation techniques and overcomes the noise-sensitivity and inaccuracy problems of purely motion-based segmentation [82]. The spatio-temporal segmentation techniques also adapt easily to more generic non-rigid motion and therefore to more generic scenes. Two major categories of 2D motion-based segmentation are the optical flow discontinuity based and the change detection based. Computation of motion and detection of motion boundaries present a chicken and egg dilemma as noted by [82]. Local flow discontinuity tests are used to find optimal boundaries in many techniques. In essence, the optical flow field has the same statistical characteristics as that of intensity or color in an image. Image segmentation experience suggests that only optical flow field is not sufficient for motion segmentation and high level information and rules are helpful in analysis.

Wills *et al* [80] proposed a graph cuts formulation for motion segmentation. First a set of dominant motions in the two views is obtained. The energy terms in the graph are based on the re-projection error due to each motion model and the smoothness term is defined based on color similarity between the pixels. The graph cuts minimization is then performed to assign to each pixel one of the dominant motions. Only planar motion is considered and all motion models are represented by a 3×3 homography matrix. Bhat *et al* [9] proposed a similar method for dense optical flow estimation in scenes with multiple large rigid motions. They extend the method proposed by wills *et al* [80] by also taking care of

non planar motion layer, using the Fundamental matrix and disparity combination as a label for pixels not satisfying any homography.

2.7 Image Registration

Image registration refers to the problem of finding the transformation that needs to be applied to one of the image to align it with the second image. Registration has been a topic of research for many years [12]. The great deal of work in this field is driven by the importance of registration in various problems, including medical imaging and satellite imaging where the images taken from two or more different sensors needs to be registered for purpose of study and analysis. The same concept is also responsible for the panoramic mosaic generation.

It is a established that two views of a planar scene are related by a projective transformation. This projective transformation can be represented up to scale by a 3×3 matrix, called the Homography (or H) matrix [23]. In usual scenes the number of planes in an image is very large and registration is applied more at region level rather than the image level. Numerous models and methods have been suggested for estimation of the parameters of H Matrix. The reader is suggested to refer to [2, 5, 12, 82] for a detailed review. In our work we mainly use the 4 point algorithm [26] in coordination with robust estimation methods. This matrix is used for transforming the pixels in one frames' coordinate system to anothers. The pixel's which are missing in a particular frame are looked-up in the neighboring frame using the homography information. This is the idea used in creation of mosaics [73].

2.8 Image Inpainting, Texture Synthesis, and Image Completion

Image inpainting fills-in the unknown regions (or holes) in an image based on the surrounding pixels. Structure propagation and texture synthesis are the two basic approaches for image inpainting. Structure propagation methods propagate the structure around the unknown region progressively to inside it. Bertalmio *et al* [6] proposed a method for filling-in of the image holes by automatic propagation of images the isophotes (lines of similar intensity) in the image by means of Laplacian smoothness operator. Their method belongs to a class of methods called PDE based methods [4, 13]. Jia *et al* [34] proposed *image repairing* where the damaged image is first segmented using texture information and the segmentation is extended to the missing region via *tensor voting*. The color value for a missing pixel is then synthesized using only known pixels from the same region again using ND tensor voting.

Texture synthesis methods [19, 20, 28] assume the existence of a pattern in the image and fill the pixels in the missing region by finding a patch matching the neighboring texture in the whole image. In [20] texture synthesis was demonstrated at pixel level, i.e. an unknown pixel's value is synthesized by matching the known neighboring pixels in the source region. Texture synthesis at block level is proposed in [19].

Structure propagation methods work well only on small holes, whereas texture synthesis methods require texture in the image. Methods combining both structure propagation and texture synthesis have also been proposed [7, 18]. Bertalmio *et al* [7] decomposed the original image into a texture image and a structure image. The texture image is completed by texture synthesis method while structure image is completed using a structure propagation algorithm.

Block level synthesis methods are also termed as exemplar-based methods. Exemplar-based approaches are observed to give the best inpainting results. Criminisi *et al* [18] were the first demonstrated the application of exemplar-based approaches on natural images rather than texture-only images. It was demonstrated that the filling order is crucial and priority of filling was biased towards patches which were on the continuation of strong edges and were surrounded by high-confidence pixels. The technique was improved and generalized in recent works [41, 45, 72]. Sun *et al* [72] require the user to specify the curves on which the most salient missing structure reside. Its an exemplar based approach where the target patches are selected by use of belief propagation based energy minimization. Komodakis and Tziritas [41] proposed a new image completion algorithm based on an improved belief propagation algorithm called Priority Belief Propagation (PBP). The method is exemplar based and the PBP is used to find the appropriate patch for a given hole pixel efficiently. The proposed method has the advantage of yielding a globally optimal filling rather than making greedy decision like [18] and being applicable to texture synthesis as well as image inpainting .It doesn't use any user intervention.

Kang *et al* [38] proposed a technique for inpainting or region filling using multiple views of a scene. Their technique is based on finding the appropriate region in the second view and then mapping the pixels back to the first view using the affine projection calculated using the correspondence in the two views. Similar methods are used in video completion as discussed below. In Interactive Digital Photomontage [3], Agarwala *et al* demonstrated that for a sequence of images taken from same view point, i.e. fixed camera positions, the transient/moving foreground object removal can be done by applying a maximum likelihood filter at each pixel. This technique is theoretically very similar to that of registering the sequence of images to a reference image and then filling the unknown pixels in the reference image from other images.

2.9 Video Completion or Object Removal

Object removal in videos has received attention in recent years. Two types of techniques have been proposed. The first type finds out the missing data by searching for a patch matching the neighborhood of the hole in the video similar to the exemplar based methods in case of images. The match is defined in terms of spatial and temporal feature similarity. Periodicity in motion is a common assumption for these techniques. Space time video completion [79] uses a five dimensional sum of squared differences to find the appropriate patch for filling the holes where the matrices include the three color values and velocity along x and y direction. *Video Repairing* proposed by Jia *et al* [35] recovers the missing part of foreground objects by *movel* sampling and alignment using tensor voting to generate loops of motion by

connecting the last frame to the first frame. Jia *et al* [36]'s method is another exemplar based approach for video completion. They use the motion information at a pixel as a criteria for determining the priority of filling as well as for determining the appropriate source patch. The patches are merged using a graph cuts based method for minimizing seams.

Motion field interpolation based methods have also been developed recently. Kokaram *et al* [40] perform object removal by using the motion information to reconstruct the missing data by recursively propagating data from the surrounding regions. Matsushita *et al* [52] proposed *motion inpainting* where the inference of the unknown pixels information is based on the optical flow vectors which are in turn interpolated based on the flow of the surrounding pixels. In the second scenario, explicit use of the geometry of multiple views is made to infer the information missing in the current frame from the nearby frames. This is directly related to the problem of disocclusion in computer vision. The fact that two views of a plane are related by a perspective transformation defined using a Homography matrix, forms the basis of most such approaches. Jia *et al* [35] proposed the repairing of the static background by the use of planar layered mosaics. The layers are assumed to be available from initial manual segmentation followed by tracking using the mean shift algorithm. Similar approach has been demonstrated by Zhang *et al* [83]. They use an automatic layer extraction approach followed by layered mosaicing. If some holes still remain, an image inpainting approach is used in frame-wise manner based on a graph cuts formulation.

When the camera is far from the background, the nearby frames of the background can be approximated to be related by an affine or projective transformation. This approximation is used by some methods [35]. Such methods will fail for indoor scenes where multiple background motion exists. In general, it would be impossible to identify every single plane in the scene and apply layer mosaicing on each of them individually, automatically and accurately.

Structure from motion problems employ some techniques that are relevant to this problem. Vincent and Laganiłre [75] discuss the problem of dividing the image into planes. They start with a set of point correspondence and apply the RANSAC [22] algorithm with an optimal selection of the four initial points to maximize the chance that the points are on same plane. All the other points in the image are declared to belong to the plane whose homography gives least re-projection error. Friedrich *et al* [24] finds the interest regions [56] in the two views on which affine region matching is performed. The affine matching is helpful in removing the non-planar regions from considerations. On the matched region the homography is determined and a region growing is performed around the region to include regions which match the homography well. During the region growing step, the homography is updated to include the new interest points inside the region for the estimation. At the termination of the region growing, the scene is segmented into a set of planar regions.

2.10 Summary

In this thesis, we propose two novel approaches. The first approach is for object segmentation in videos. The user segments the first frame using the interactive image segmentation technique. A 3D graph is constructed in which the segmented image is used as a plane and an unsegmented image becomes the second plane. The pixels in the first plane are seeded using earlier segmentation and those in the second plane are seeded by tracking pixels from the first frame. Our 3D graph cut optimization thus runs on two frames at once as against [46] where multiple frames are used to build a single graph. Further we use tracking of pixels to seed nodes in the intermediate frame automatically which is a novel contribution compared to earlier works where it was either not used [46] or was accepted from the user via a special 3D interface [77]. The special advantage of the approach against the automatic approach for object segmentation in video is a high level of flexibility in determine the object or the layer extent, which is not possible in automatic motion segmentation techniques [81].

The method for video completion we propose here is different from earlier approaches as we use a geometric approach for segmentation of only a small region around the missing region instead of going for a complete image segmentation [9, 83]. Our method has the special advantage of not being entirely dependent on motion segmentation using optical flow which is still an unsolved problem in general case. The method handles the case of the background containing two or three different planes and doesn't assume an affine relation between two frames of the sequence [52]. We only use the motion flow vector for a weak clustering of pixels to determine the homographies of the various planes. Final segmentation between the planes is obtained using generalized eigen vectors of the two planes. Our method doesn't involve any interpolation as in the case of [52, 69]. Each pixel in the unknown region gets a definitely source plane label.

Chapter 3

Layer Extraction Using Graph Cuts and Tracking

We present a new method for layer extraction by tracking a non-rigid body with no fixed motion model in a video. The method integrates a graph cuts approach with robust point based tracking to achieve good tracking of the whole object over frames of a video. With minimal user interaction, our method can perform fine layer extraction over irregular motion and difficult object boundaries. To achieve this, we apply 3D graph cuts on a pair of frames and propagate the labels obtained in the earlier frame to a new frame by use of robust tracking method. The user can interactively improve the automatically extracted layers using a few extra strokes if necessary.

As described in literature [10, 11, 46, 61, 77] graph cuts optimization for more than 2 labels is comparatively slower compared to a 2-label case. As is the common approach we also solve the multiple layer extraction problem by solving a cascade of 2-layer segmentation at a time.

3.1 Layer Extraction Using Graph Cuts and Tracking

An overview of our system is shown in Figure 3.2. The steps involved are the following. The user first selects one or more key frames from the video and segments them using an interactive image segmentation technique into foreground and background regions (We use the term foreground to mean

(a) (b) (c)

Figure 3.1 Advantage of interactive segmentation: shadow of train on calendar (a) can be regarded as part of the background layer in our case (c), unlike the automatic case in [81] (b). (marked by red circle)

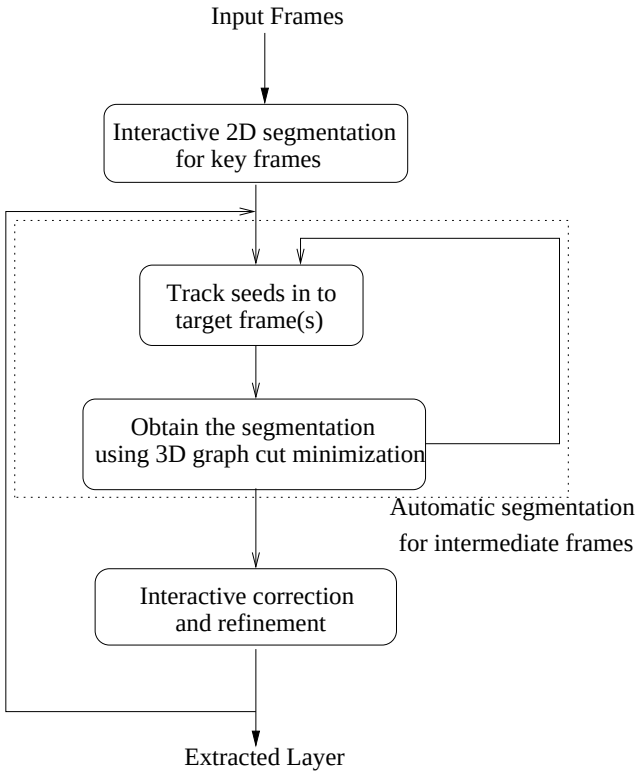


Figure 3.2 Overview of the different steps of our system.

the layer to be extracted and background to refer to all the pixels in the image which are not part of this layer.)

Using the segmentation given in the key frame(s), robust tracking provides the seed points for the intermediate frames. Our algorithm can proceed with just one key frame, namely, the first frame. We build a 3D graph for each pair of frames using individual pixels as nodes of the graph. The 3D graph cuts technique [11] is then applied and the segmentation is achieved for the new frame. This is continued for all the frames in the video.

The user can manually inspect the segmentation results and provide extra strokes to improve the results of the automatic segmentation. In the following subsections we provide the details of each of the above steps.

3.1.1 Interactive Segmentation for Key Frames

Interactive segmentation is done for one or more key frames in the video. This step is based on the interactive segmentation method by Boykov and Jolly [11]. The user gives a few strokes to mark the foreground and background regions in the image in each key frame. Since we move only forward in frames, it is sufficient to start with the first frame as the key frame. During the process, any frame can be segmented from scratch and can effectively become a key frame, if the user desires.

3.1.2 Automatic Propagation of Segmentation

Various approaches [11, 46, 77] discuss the use of the min-cut on more than two dimensional data. A 3D graph can be obtained by treating a set of frames as planes and connecting the pixels in these images to the pixel of neighboring frames in addition to the neighboring pixels in the same frame. Some of these approaches do not give hard constraints in the intermediate frames [11, 46] while others take them from the user [77]. We propose a novel approach to obtain the hard constraints automatically. Based on the segmentation of the previous frame, we obtain good features points inside both foreground and background regions [68]. These features are then tracked over to the next frame where they are used for setting the hard constraints for further segmentation.

3.1.2.1 Propagation Step

We use robust tracking to propagate the seed points from one frame to another. In our approach we use the KLT tracker [68, 74] which tracks given feature points from one frame to another. We track two kinds of points: one set is obtained as a set of pixels which are good features to track [68] and second is a set of pixels spread evenly in the image[1]. The KLT algorithm tracks these points in the next frame. Points not tracked confidently are ignored. Confidence is measured in terms of residual error per pixel. We use a value of 10 as threshold in our experiments. In practice any good tracking algorithm can be used to propagate pixels from foreground and background regions across frames. Algorithm 1 describes this step in pseudo code.

As shown in Figure 3.4, we label the points tracked from the background region in the source frame as background in the target frame. The same holds for the pixels of the foreground region.

Input: The frames F_1 and F_2 with segmentation label L_1 for F_1
Output: Some seed labels for F_2
begin
  /*obtain the feature list using KLT tracking */
  feature_list=**trackKLTFeatures**(F_1, F_2);
  /*propagate the labels to the next frame */
  foreach $(l_1, l_2) \in feature\_list$ **do**
    $L_2(l_2) = L_1(l_1)$
  end
end

Algorithm 1: Algorithm for propagation step.

3.1.2.2 3D Graph Construction

We build graphs using two frames at a time. The first frame is one which has been segmented previously. The next frame is the frame which has to be segmented. Each pixel in the image is connected

[1] precisely, only on the region of interest

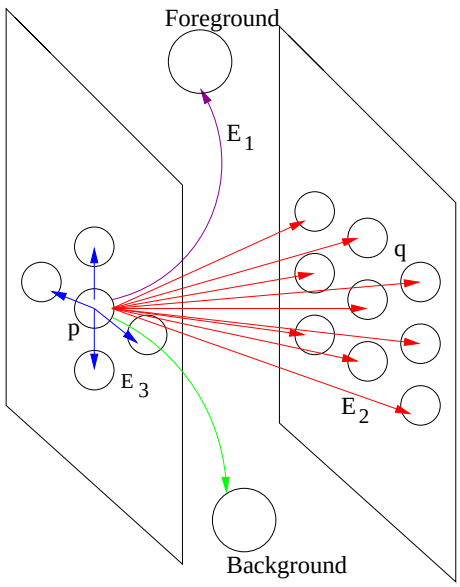


Figure 3.3 The 3D graph construction. Every pixel p is connected to 8 neighbors in same frame (only 4 shown, marked by blue edges), and 9 pixels in the neighboring frame, marked by red edges, and to the two terminal nodes namely the source (foreground) and sink (background) marked in cyan and green colors respectively. The energy for the three types of connections are E_3, E_2 and E_1 respectively.

to its 8 neighbors in the same frame and to the 9 neighbors in the next frame, as shown in Figure 3.3. We can use a more densely connected graph in theory but our experiments show this gives good results.

Now we define the energy terms for the min-cut algorithm. The energy that needs to be minimized can be seen as the sum of three terms as [46]

$$E = \sum E_1(p, f_p) + \lambda_1 \sum_{(p,q)\in V_I)} E_2(p, q, f_p, f_q) + \lambda_2 \sum_{(p,q)\in V_c)} E_3(p, q, f_p, f_q), \tag{3.1}$$

where f_p is the foreground/background label for the pixel p. λ_i denote the relative importance of the terms. We use values $\lambda_1 = 10$ and $\lambda_2 = 1$ in our experiments.

The term $E_1(p, f_p)$ denotes the data energy term [61]. It is the penalty of labeling the pixel p as f_p. This term is defined as the similarity of the pixel color to that of the foreground or background samples. Boykov and Jolly [11] defined this similarity using gray scale histogram.

We use the Gaussian Mixture Models (GMMs), which are commonly used to represent the foreground and background pixels, in place of intensity histogram. We use the method originally proposed by Orchard and Bouman [55] for obtaining the approximate GMM from the user segmented images. Let us denote the components of the foreground GMMs by (μ_m, Σ_m, w_m) for $m \in [1, M]$, where M is number of Gaussians in the model. We use a value of $M = 6$ in our experiments. For a pixel color c, the distance to the foreground GMMs is defined as [46, 61]

$$d^f = \min_{m\in[1,M]} [D(w_m^f, \Sigma_m^f) + D(c, \mu_m^f, \Sigma_m^f)], \tag{3.2}$$

where

$$D(w, \Sigma) = -\log w + \frac{1}{2} \log |\Sigma|, \tag{3.3}$$

and

$$D(c, \mu, \Sigma) = \frac{1}{2}(c - \mu)^T \Sigma^{-1}(c - \mu). \tag{3.4}$$

Our definition of E_1 is similar to one proposed by Boykov and Jolly [11]. The term's value for seed points is set to a very high value K to the seed's label node (source or sink) and very small (0) to the opposite label. The value for a non-seed point is set to be the distance d^f and d^b for the edge to the background and foreground respectively. The values are depicted in Table 3.1. The value K is defined as:

$$K = 1 + \max_{p \in P} \sum_{q: \in N_p} V_{\{p,q\}}.$$

edge	weight(cost)	for
{p,S}	$\lambda.d^b$	$p \in P, p \notin O \bigcup B$
	K	$p \in O$
	0	$p \in B$
{p,T}	λd^f	$p \in P, p \notin O \bigcup B$
	0	$p \in O$
	K	$p \in B$

Table 3.1 Weights assigned to the various edges in the graph. p is any pixel in graph, S and T are the two virtual nodes representing the source and sink respectively.

The terms E_2 and E_3 denote the interaction penalty for intra-frame neighboring pixels and the pixels in the neighboring frame. We define these values using the well known interaction penalty measure [11]:

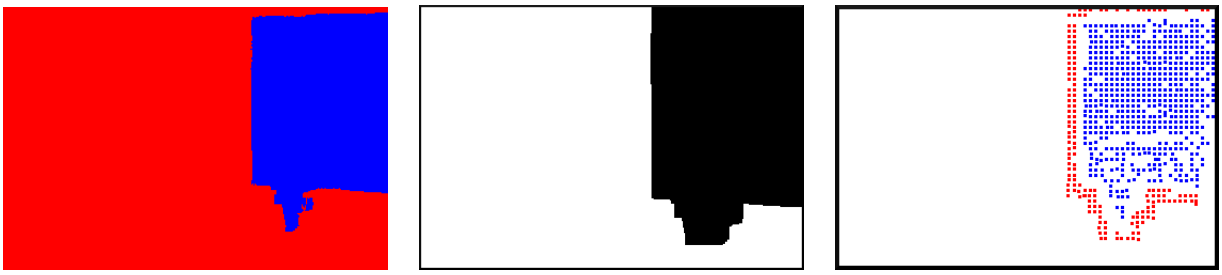
$$E(p, q, f_p, f_q) = |f_p - f_q|. \exp\{-\frac{||c_p - c_q||^2}{2 * \sigma^2}\}. \frac{1}{dist(p, q)}, \tag{3.5}$$

where $||c_p - c_q||^2$ is the Euclidean distance of the color values of pixel p and q. The term σ can be described as a parameter weighing the contrast. A high value of σ puts a low penalty on high color difference and vice versa. The term $|f_p - f_q|$ ensures that the penalty is taken only for the boundary values [11]. We use a value of $\sigma = 50$ for our experiments.

The algorithm for extracting a single layer in the sequence is listed in Algorithm 2. The previously segmented frame is loaded as the first plane on graph, with the pixels labeled either background or foreground. The labeling of these pixels is not changed during the minimization. The frame to be segmented is loaded as the second plane of the graph. Pixels are tracked from the first frame to second

Input: The frames F_i.
Output: The segmentation labels on $L[F_i]$
begin
  if F_i *is key frame* **then**
    goto nextFrame;
  end
  /*load the previous(segmented) frame as the first plane on the graph */
  $graphPlane_1$=**loadFrame**(F_{i-1});
  /*load the current frame as the second plane on the graph */
  $graphPlane_2$=**loadFrame**(F_i);
  /*Track feature points from previous frame to current using Algorithm 1. */
  featuresList = **propagateSeeds**(F_{i-1}, F_i);
  /*Calculate the edge weights for all the edges in the graph. */
  calculateEdgeWeights(for each pixel $(p, q) \in graphPlane_{i=1,2}$);
  /*Set the terminal weights for each pixel on plane 1 from prior segmentation knowledge */
  setTerminalWeights(for each pixel p in $graphPlane_1$);
  /*Set the terminal weights for each tracked pixel on plane 2 */
  setTerminalWeights(for each pixel q, s.t. $(p, q) \in featureList$);
  /*Obtain the segmentation label by running the 3D graph cuts minimization on the constructed graph */
  $L[F_i]$=**graphCutMinimisation**();
end

Algorithm 2: Algorithm for layer extraction.



(a) (b) (c)

Figure 3.4 The tracking process: (a) The calendar layer is shown segmented in source frame, (b) The estimated region mask to decide which pixels (shown in black) in the image will be included in graph cuts minimization for segmentation of next frame, (c) The seed points or hard constraints obtained using reliable tracking of points from the source frame (red indicates background and blue indicates foreground).

and these pixels are set as hard constraint. All the pixels in the previously segmented frame act as hard constraint too. The segmentation is obtained for the new frame by graph cut minimization on the constructed graph.

3.1.3 Interactive Refinement

User interaction is needed to manually refine some of the labellings obtained in the intermediate frames during the process. In our system, the user gives the corrective strokes in one frame and chooses the number of frames for which the automatic segmentation step has to be re-done. Once the segmentation is obtained for a particular frame, user can interactively modify the segmentation using the iterative max-flow algorithm on the original 3D graph. Iterative graph cuts optimization on a already saturated graph are applied by changing the weights of the pixels marked by the user and running the optimization on the modified graph. If pixel p which was earlier not a seed pixel is now declared a foreground seed, the weights of the edges are updated as described in Table 3.2.

t-link	**initial cost**	**add**	**new cost**
{p,S}	$\lambda.d^b$	K+$\lambda.d^f$	$K + c_p$
{p,T}	$\lambda.d^f$	$\lambda.d^b$	c_p

Table 3.2 Iterative graph cuts weight updates, p is the added seed foreground pixel. c_p represents constant which is actually sum of the d^b and d^f.

Unlike other approaches which have a final stage where user interaction can be applied, in our technique user can interact and improve the labellings (segmentation) at any intermediate frame. Interaction step is fast as we will see in Section 3.2.

3.1.4 Speeding Up the Segmentation

A typical graph cut on the whole video could be slow due to the large number of pixels over which optimization is to be applied. As pointed out in Section 1.1, one of the main emphasis of our approach is to make the 3D graph cuts more efficient using the temporal and spatial continuity. We increase the efficiency using several steps.

We first limit the object position in the the second frame to a neighborhood of its previous position, called the estimated region mask. We can prune all the pixels which are not in the union of the original mask and estimated region mask (Figure 3.4). The estimated region mask can be computed based on the estimated motion of the object and any knowledge of motion model. In our experiments, we use a radial disc around the previous position as the estimated region mask. This prunes out large parts of the image from the graph and boosts the efficiency by both avoiding the calculation of the energy terms and the actual running of the minimization algorithm. We also get many hard constraints using tracking

and avoid calculating the computation intensive energy functions for these pixel positions. Finally we use an iterative graph cuts algorithm and avoid the expensive from scratch optimization during the user interaction.

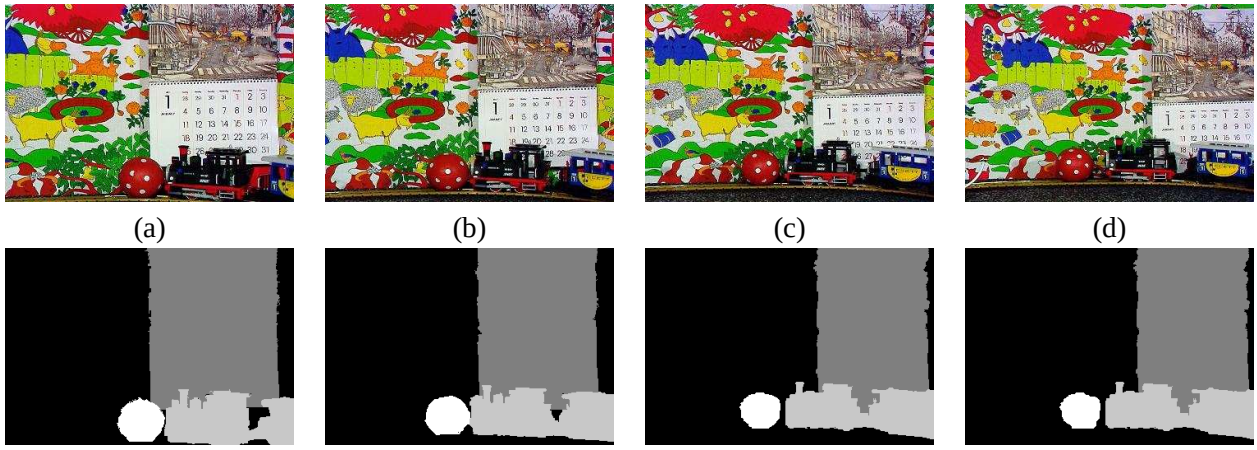


Figure 3.5 Layers obtained by application of our method on the *mobile & calendar* sequence. (a-d) show four input frames from the sequence. The extracted layers are shown in (e-h). Note the separation of shadow in (e-f) as discussed in Figure 3.1.

3.2 Results

We show the layers extracted from the *mobile & calendar* sequence and the *flower garden* sequence in Figure 3.5 and Figure 3.6 respectively. The figure shows that the algorithm extracts the ball from the surrounding objects, many of which have similar colors, quite well. It should be noted that the ball's motion doesn't follow any specific motion model. The train's shadow was also declared as part of the background layer as can be seen in Figure 3.1.

In case of the *flower garden* sequence the tree matches in color with some of the background regions. In this case more user interactions were required to un-mark the spilling-in of the background in foreground regions and vice versa. The average interactive processing time was less than a second per frame. The time required for interactive correction depends on boundary smoothness. The garden and house layer separation for example required just 3-5 strokes after the first key frame. Figure 3.7 shows another example where we segment the football and player as a single layer from the video. This example demonstrates that the layer extraction in our approach is highly driven by the user's choice.

The time required for the segmentation depends on the object size as the graph size is dependent on it. For a small object like ball in the *mobile & calendar* sequence, time taken on each iteration of 3D graph cuts is around 1 second, while for the calendar it is around 2 seconds. Iterative improvements on

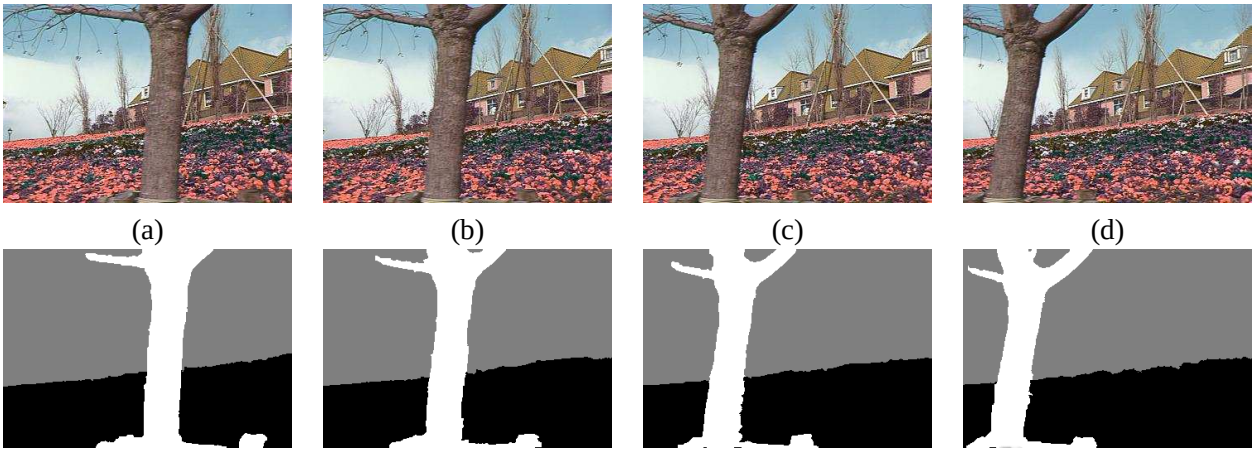


Figure 3.6 Layers obtained by application of our method on the *flower garden* sequence. (a-d) show four input frames from the sequence. The extracted layers are shown in (e-h).

the graph are fast and take less that 0.1 second per optimization. All the experiments are performed on an Athlon 2600+ Machine, with 256MB RAM. The sequence had a frame size of 320×240. The overall processing time for one layer comes to around 2.5-4 seconds including the interaction. Therefore a 50-frame video can be processed in 3-4 minutes. Our approach has the advantage of allowing precise user inputs while performing 3D graph cuts on individual pair of frames if necessary.

Figure 3.7 The football and player can be extracted as a single layer by our algorithm even though their motions do not have any common motion model.

3.3 Application to video matting

The layers obtained by the method represent optimal boundaries at pixel level for the foreground and background. We can further improve the layers by frame-wise application of matting. We use the matting technique proposed by Levin *et al* [43]. The mask obtained in the layer extraction process is

eroded to obtain the sure foreground region, a negation of dilated mask from the whole image gives the sure background region. All the other pixels are labeled as unknowns. This trimap is passed to the matting algorithm, which produces an alpha matte for the image. This process is illustrated in Figure 3.8. The individual alpha mattes in each frames are then combined to give the video matte for the object across the video. Figure 3.9 compares the layer obtained by initial layer segmentation with those obtained after applying matting. Matting produces much more smooth layer transition boundaries and removes the extra non-foreground pixels from the layer, producing a cleaner foreground layer. The application of matting provides as alpha matte of the object which can be used to compose the layer on a different foreground. This is specially useful for objects with fine boundaries like hair. Figure 3.10 shows the application of the alpha mattes obtained using the process to cut paste the actor from the original video onto a video with a new background.

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

Figure 3.8 Matting: The input frame (a) is first segmented in to foreground and background layers (b). (c) shows the trimap obtained by simple morphological operation on the binary segmentation map (b). (d) shows the alpha matte obtained using the trimap (c) by application of a matting technique.

3.4 Summary

We proposed a method that integrates robust feature tracking to seed the hard constraints on a 3D graph cuts minimization is proposed. This method can be used for a variety of purposes where layer extraction is useful. Combined with a matting approach the layer obtained can be refined to have precise alpha values at the borders. The method has the advantage of handling non-rigid object segmentation. The method clearly falls in the category of spatio-temporal motion segmentation methods. Our method is currently limited to binary labeling. We propose to investigate the feasibility of multi-label segmentation. Significant improvement to the algorithm's efficiency is terms of processing time is expected when graph cuts is performed for pixel clusters in 2D and 3D as in [77] instead of individual pixels. The 3D graph cuts optimization can be further quickened up by use of GPU for calculation of the terminal links(*t-links*) which is independent for each pixel and can be calculated faster due to parallelization in hardware.

A limitation of our technique is the requirement of texture on the foreground and background regions, which is mainly required for the tracking to work. Performance of pixel correspondence methods like

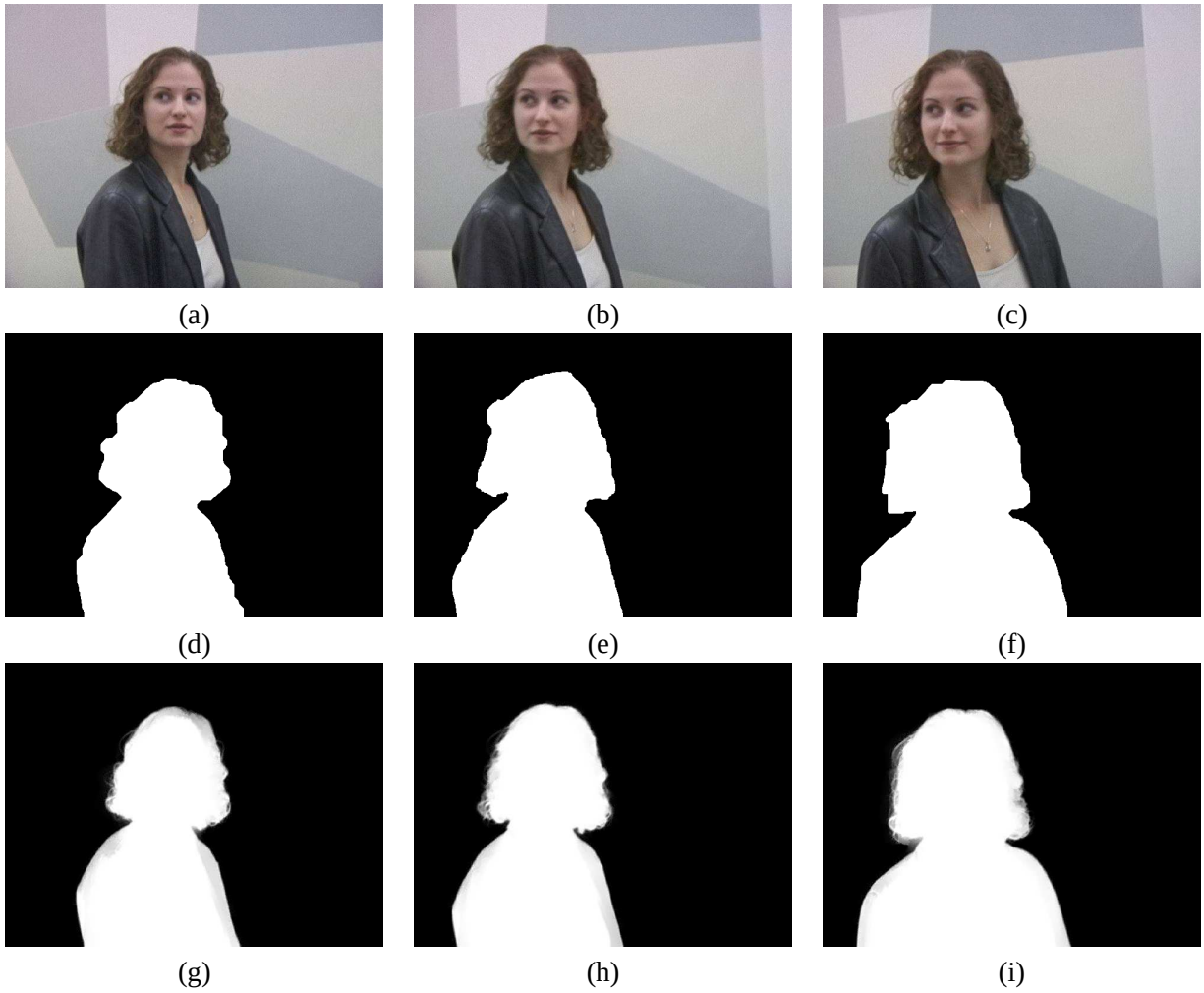


Figure 3.9 Alpha Matte extracted from the frames of the actor-sequence, (a,b,c) show the input frames, (d,e,f) show the segmentation obtained by layer extraction. The final alpha mattes obtained by applying matting are shown in the (g,h,i).

Figure 3.10 The actor is cut from the input frames (a,d,g,j) and is pasted on a new background (b,e,g,k).

KLT is highly dependent on the textured-ness of the region. In other words the image should have good corners. However as new tracking measures like SIFT based tracking or region level tracking evolve the limitation can be overcome.

Figure 3.11 The User Interface of the software for layer extraction in videos.

Chapter 4

Object Removal and Video Completion for Indoor Scenes

In this Chapter, we present a new approach for object removal and video completion of indoor scenes. In indoor images, the frames are not affine related. The region near the object to be removed can have multiple planes with sharply different motions. Dense motion estimation may fail for such scenes due to missing pixels. We use feature tracking to find dominant motion between two frames. The geometry of the motion of multiple planes is used to segment the motion layers into component planes. The homography corresponding to each hole pixel is used to warp a frame in the future or past for filling it. We show the application of our technique on some typical indoor videos.

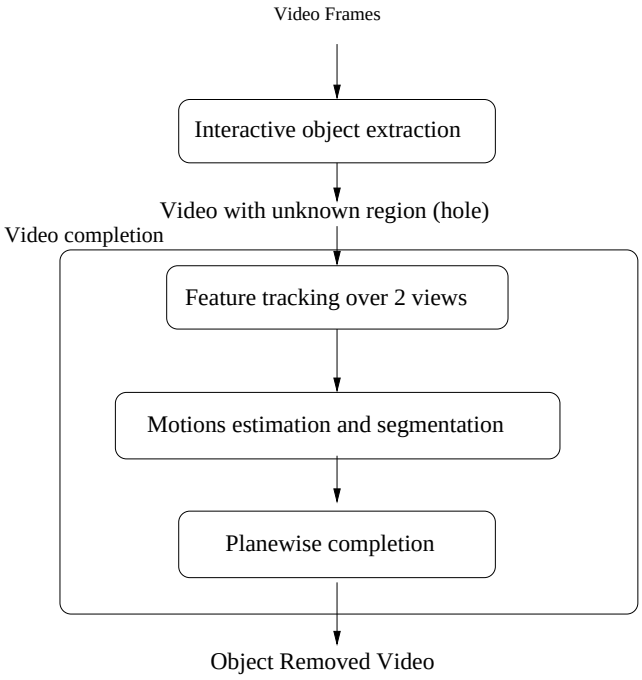


Figure 4.1 The overview of the various steps of our system.

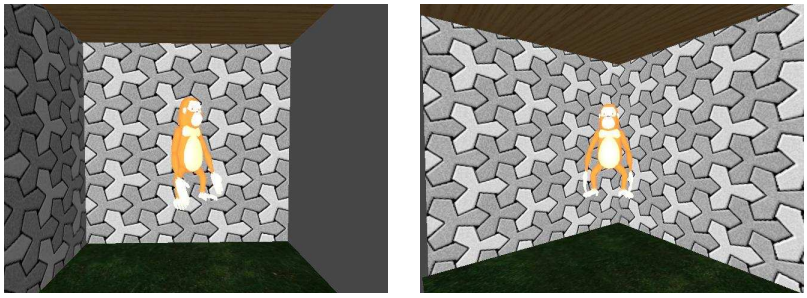
4.1 Video Completion for Indoor Scenes

In this paper, we address the problem of object removal and video completion for indoor scenes where the transformation of the background is non trivial and variable. An overview of the process is shown in Figure 4.1. We track the foreground (the object to be removed) interactively using the method described in the previous chapter [33] to track the objects across the video. We assume that the background has a maximum of 2 planes around the object to be removed in two adjacent views. The region around the object is segmented into one or two planes, using dominant motion model estimation followed by an optimal boundary detection algorithm. We then apply the respective homography [26] to recover the unknown pixels from the neighboring frames. These steps are explained below.

4.1.1 Object Segmentation

The segmentation step provides the masks of the object to be removed across the video frames. Unlike image inpainting techniques, getting this mask from the user in each frame is not feasible. We use an interactive method of object extraction using graph cuts and feature tracking to generate the mask across the video sequence as described in the previous chapter. The user gives a binary segmentation of the first frame, marking the foreground and the background. We track features points in the segmented frame to the current frame (unsegmented) and set them as seed points in the 3D graph constructed with the two frames. A graph cuts optimization on the graph gives the segmentation for the current frame. The user can mark extra stroke and run the iterative graph cut to improve the segmentation before proceeding to next frame.

After running through the frames of video, we get the object mask in each frame. This mask defines the region to be filled in using the video completion algorithm.



(a) (b)

Figure 4.2 Two different cases of object removal (a) The local background around the object is a single plane (b) The local background around the object is spread over more than 1 plane. Due to the local nature of the plane segmentation technique the first case (a) doesn't need any motion segmentation. Motion segmentation in the second case (b) is also local in nature and even though there are more planes in the image only the two planes which constitute the object's background would be segmented.

4.1.2 Video Completion

Our algorithm's basic assumption is the existence of a piecewise planar background in local neighborhood of the object to be removed. Our video completion algorithm can be divided into following major sub-steps.

4.1.2.1 Feature Tracking in Two Views

The first step is finding the corresponding feature points in the two frames of the video. We use the KLT tracking for tracking point features across the frames. The method involves finding trackable features in the first image, which are then matched in the second image. We find the features selectively in only local neighborhood of the hole, this is to ensure that we only consider useful correspondences for our motion estimation and completion steps. We call the region around the hole where we do the selective matching as the Region of Interest (ROI). Figure 4.3 (b) shows the optical flow vectors calculated in the ROI. The ROI can be obtained by dilating the object mask with an appropriate thickness. The algorithm is described in Algorithm 3. We use simple morphological operations [31] to obtain a region around the object, in which KLT features(points) are extracted. These features are then tracked into the next frame and the features which are not matched in objects neighborhood are removed(pruned) from the list.

Input: The two adjacent frames F_1 and F_2, with the object masks M_1, M_2
Output: The valid features list feature_list
begin
  /*Dilate the mask M_i with a specific structure element */
  D_i = **dilateImage**(M_i);
  /*find the features in the first image within valid region, M_1 */
  features= **findFeatures**(F_1 where $M_1 = 255$);
  /*track the features in to the second image */
  feature_list = **trackFeatures**(F_2,features);
  /*prune those features which are outside M_2 */
  pruneFeatures(feature_list,M_2);
end

Algorithm 3: Algorithm for finding selective correspondences.

4.1.2.2 Motion Segmentation

Given point correspondences in the two images(frames), our aim is to find the planar segmentation of the ROIs. Figure 4.2 shows the two possible scenarios. In Figure 4.2(a) the ROI around the object is a single plane, while in Figure 4.2 (b) the ROI includes two different planes. We use a combination of two approaches to robustly estimate the segmentation of the points inside the ROI into multiple planes. The algorithm proceeds by finding the dominant motions in the ROI using the set of correspondences.

We use the RANSAC [22] algorithm to determine the dominant motion. RANSAC algorithm has the advantage of being robust to outliers, which are indeed present in our correspondence pairs due to the existence of multiple planes.

To begin with, we use all the correspondence pairs to determine the dominant motion. The features which are inliers for the current dominant motion are then removed from the set and the step is repeated to find the next dominant motion. To avoid RANSAC algorithm from choosing wrong set of initial four points, we modify the selection phase to accept the set of points only if they are within a set threshold distance. The points which are declared inliers to the RANSAC algorithm are then used for a least square error fitting estimate of the homography using the normalized DLT algorithm [27]. This fitting gives us the final homography for the set of points. Figure 4.3 (c,d) shows the automatically determined first and second dominant motions as cluster of optical flow vectors which are their inliers. The algorithm for finding dominant motion is listed in Algorithm 4. We cluster the motion vectors to determine the underlying motion model (Homography), until the number of unassigned motion vectors is below a threshold, when we declare then as outliers or false correspondences.

Input: The Set of valid correspondence pairs in frames F_1 and F_2
Output: The set of dominant motions H_i
begin
  /*Obtain the set of correspondence from the Algorithm 3 */
  S_0=set of all correspondence pairs obtained;
  $i = 1$;
  /*While the number of elements in the set is greater than set threshold */
  while $|S_{i-1}|\ ge\ \tau$ **do**
    /*Find the dominant motion model using RANSAC */
    H_i =**fitRANSAC**(S_{i-1}) ;
    /*Get the inliers satisfying the homography */
    I_i =**getInliers**(S_{i-1},H_i) ;
    /*Update the set of correspondence pair, yet to be assigned to a motion model */
    $S_i = S_i - I_i$;
  end
end

Algorithm 4: Algorithm for finding dominant motion models.

4.1.3 Optimal Boundary Estimation

Optimal boundary estimation is needed to actually separate the ROI into two different planes. This information is later used during the filling-in process. Note that we cannot depend on the region growing method to give us the boundaries of the planes unlike other methods [24,75] because we cannot estimate these boundaries in the unknown region. We assume the intersection of the two planar regions to be a line. Let H_1 and H_2 be the homography due to π_1 and π_2 between the two views. We find the

generalized eigenvectors of the pair (H_1, H_2) by solving the equation,

$$H_1 v = \lambda H_2 v.$$

The eigenvectors obtained have the property that two of them are the projections of two points on the line of intersection of the two planes π_1, π_2 on to the image plane I_1 and third one is the epipole in the image I_1. The two eigenvectors corresponding to the points on the plane can be identified due to the equality of their corresponding eigenvalues. The reader is referred to Johansson [37] for a proof of this fact.

Input: The set of dominant motion models H_i
Output: The region segmented in to different dominant motion models
begin
  /*for each pair of homographies from H_i */
  foreach (H_i, H_j) *pair from* $H_i's$ **do**
    /*Find the eigen vectors corresponding to equal eigen values, e_1, e_2 */
    e_i=**eigenVectors**(H_i,H_j) ;
    /*determine the line(s) partitioning the planes pair */
    L_{ij} =**lineFromPoints**(e_1, e_2)
  end
end
Partition the region using L_{ij}

Algorithm 5: Algorithm for partitioning regions into various planes.

Using the homogeneous coordinates of the two points on the image plane, we can obtain the exact line of intersection in the image. In fact we need this line only over the ROI. Thus, we have the planar layers for the ROI. We warp these layers in the neighboring frame to the frame to be fill-in the unknown region. The correspondence between layers obtained in two views is established by measuring the percentage of the tracked points that are part of the layer in previous frame. In the ongoing discussion we use the word *label* of a pixel to refer to the layer assigned it. Figure 4.3 (f) shows a line obtained by this method, (g) shows the plane segmentation in the ROI which is defined by the line. The algorithm is listed in Algorithm 5.

The correctness of the line determined using the method needs to be ensured as small errors in homography calculation can lead to high errors in line determination. In fact the homography pair may have complex generalized eigenvalues and eigenvectors and may not yield a valid pair of points to obtain the line. We validate the correctness of the boundary line by ensuring that it partitions the correspondence pairs into different clusters depending on the homography to which they belong. In case the line is not determinable or validation fails we obtain the line from a neighboring frame where it was detected and verified by applying the underlying homography.

It should be noted that the methods which give good results for dense motion segmentation from multiple views are not suitable for segmentation of the frames with the missing region. Graph cuts based motion segmentation techniques [9, 80] determine the dominant motion models in the scene and assign

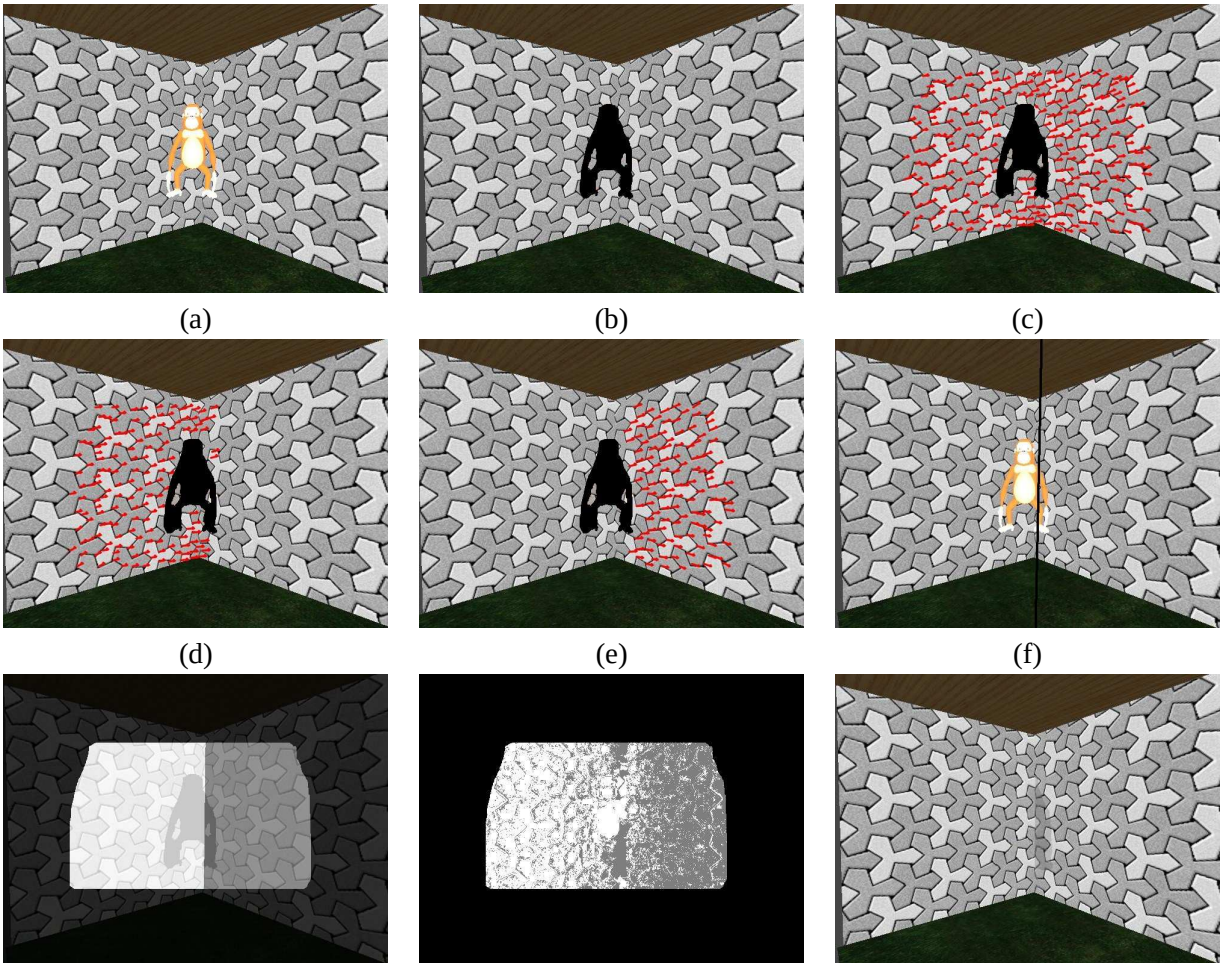


Figure 4.3 Intermediate outputs at the various stages of the algorithm (a) Input image (second frame is not shown) (b) The object to be removed is masked out and region is shown in black (c) Sparse optical flow vectors on the image (shown in red, in twice the original size to make them visible) (d,e) First and second dominant motion vectors clustered respectively (f) Line of intersection of the two planes calculated as detailed in Section 4.1.3. (g) The surrounding background of the region is segmented into two planes (h) Output of graph cuts based binary partitioning of the segments, shown for comparison (i) The results of the completion on this frame.

each pixel to one of the motion model based on an optimal graph cuts segmentation. The unknown pixel can never be accurately assigned to any particular label in these approaches due to lack of both color and motion information, which are used for determining the weights in the graph. We show the result of applying binary graph cuts partitioning in Figure 4.3(f), to illustrate this fact. We only apply a binary labeling in the graph, the white region shows points supporting first dominant motion and gray region shows points supporting second dominant region. Grey region of the image was not considered for the segmentation stage. Similarly methods like [24, 75] which assign the pixels to the motion model or planes based on re-projection error measure can not assign the unknown pixels to any particular layer accurately.

4.1.4 Layer-wise Video Completion

The line dividing the two planes gives a single confident label to each pixel in the ROI. Once the label is determined we can fill the hole by warping the nearby frames according to the homography related to the label. We build the mosaic of each plane using the neighboring frames. The missing pixels are assigned the color from the mosaic of the plane correspondence to their label. This method is in principle similar to the layered mosaic approaches [35, 83]. The difference is that we have exact knowledge of which plane an unknown pixel belongs to and use only that corresponding plane (layer). The blending of homographies of multiple layers is not needed. As in case of layered mosaic approaches the intensity mismatch might occur due to combination of various frames, simple blending methods could be applied to circumvent the error due to this. Algorithm 6 lists the algorithm for layer completion. A planar mosaic is build for each plane in the scene. The missing pixels in an frame are then obtained from the mosaic corresponding to the plane they belong to.

Input: The partitioned region R with label L for each pixel
Output: The completed region
begin
  /*Obtain mosaic for each plane present in image */
  foreach *plane* $p \in$ *Image* I **do**
    Mosaic[p]=**mosaicPlane**(p);
  end
  /*for each pixel in the unknown region R */
  foreach $q \in R$ **do**
    /* assign the value to the pixel q from the mosaic */
    I(q) = Mosaic[p](L[q]);
  end
end

Algorithm 6: Algorithm for layer-wise video completion.

4.1.5 Inpainting

Some pixels may remain unknown after the layer-wise video completion due to absence of the information in the video. Pixels which are always covered by the object to removed belong to this set. As in case of image inpainting techniques we can only approximate the values of these pixels based on the surrounding information. The extra information however is the knowledge of which plane the pixel belongs to. We can restrict the filling algorithm to use values only from the corresponding plane.

4.2 Results

We demonstrate the application of our approach on two sequences. Figure 4.4 shows the results of our algorithm on a synthetic sequence. The sequence is set in a room with two wall, a roof and a ceiling with four planes. Our approach removes the monkey as shown in the figure. Due to intensity difference on the wall during the motion the mosaicing of the wall over the views generate some intensity seams. Simple blending applied during the mosaic construction gives much better results. No application of inpainting was needed in this sequence.

Figure 4.5 demonstrate the result of the technique applied to a real sequence. Some black holes are present in the output due to unavailability of data. Inpainting is not being applied on the sequence as it is neither structure rich nor texture rich. Seams which are visible in the results can be removed by applying some blending approach.

The algorithm takes around two seconds per frame for the motion segmentation and plane matching. The completion step depends on number of neighboring frames used for creating the mosaic and takes around 1-2 seconds when 12 (6 forward and 6 backward) frames are used.

Our method can also be used for object removal in pairs of images. We demonstrate a simple example of this in Figure 4.6. The background of the flag object has three planes. Motion estimation gives us three different motion models. The intersection line is obtained for each pair of planes and used in same way as described as for videos for layer-wise completion of the unknown region. We used an affine region detection and matching, based on scaling invariant feature transformation (SIFT) [50, 51], an implementation of which is available from [49], to determine the point correspondences as the inter-frame motion was large in this case. There is also significant change in illumination between the views, which is apparent after the flag is removed and the image is completed. Both images didn't see table in the region near the flag and in the region containing the flag's shadow. Thus, that information could not be filled in.

4.3 Summary

In this chapter, we addressed the problem of video object removal and completion for indoor scenes. Our method involves user interaction only for object selection and performs the rest of the operations

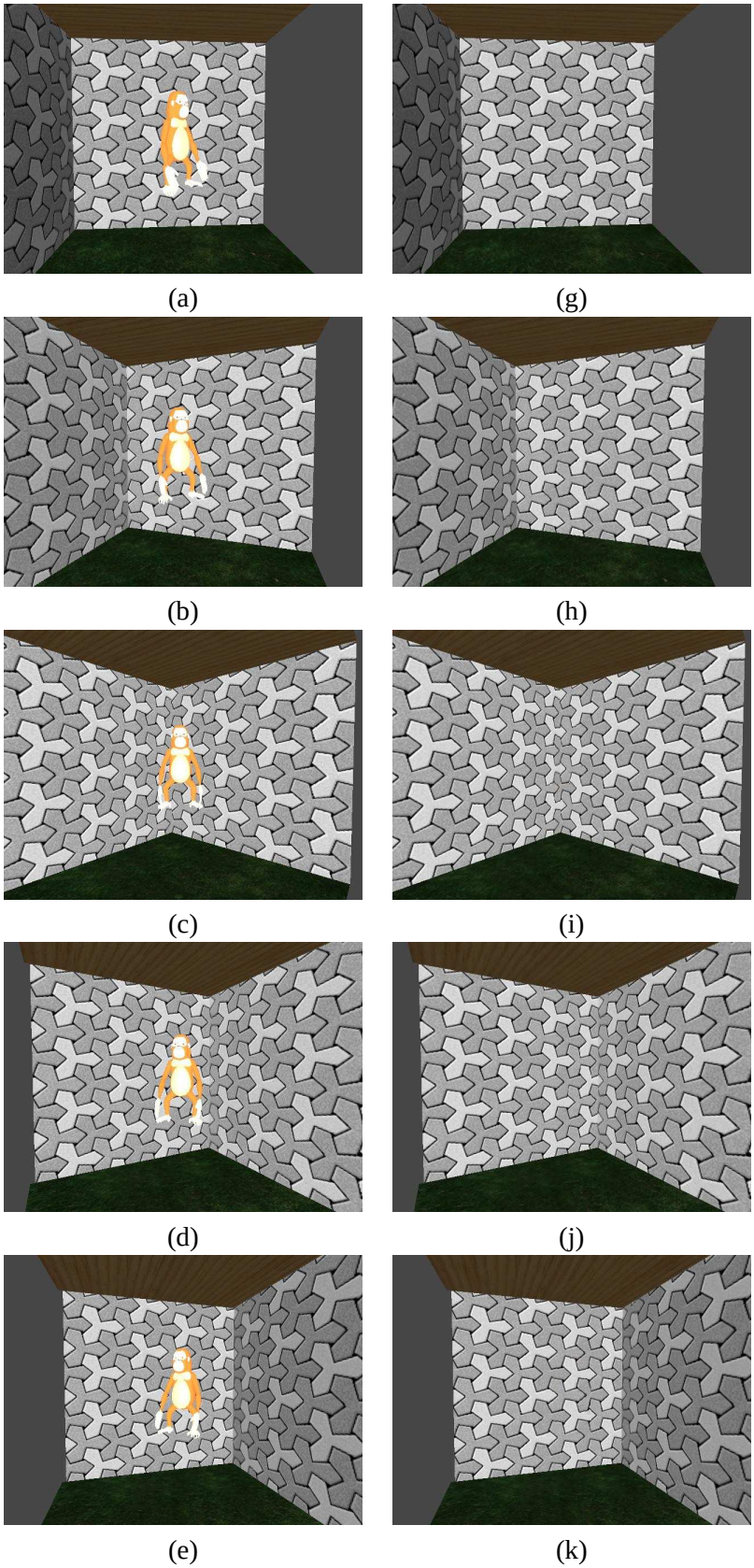
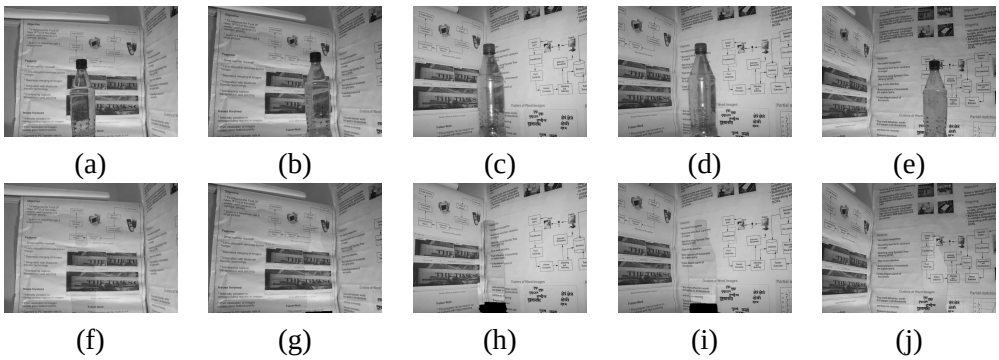


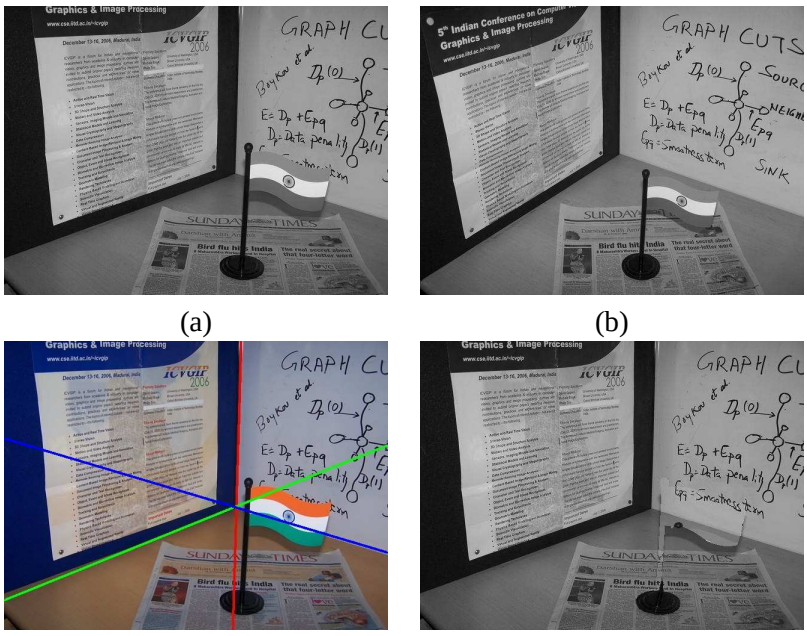
Figure 4.4 The process applied on a synthetic sequence. (a-d) show the five frames of the sequence. (e-h) show the frames after completion. The monkey is removed from the original video. (a,e) have only one background plane, while in (b,c,d) two planes are present in the background.



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

(f) (g) (h) (i) (j)

Figure 4.5 The process applied on a real sequence, we remove the bottle from the video (a-e) shows five frames of the sequence. (f-j) shows the results of video completion algorithm on each input frame. Initial and final frames have only one background while frames in the middle have two background planes. The output has visible seams at the junction of the removed object due to very high intensity change in the scene.



(a) (b)

(c) (d)

Figure 4.6 Application of our approach to images. (a,b) two views of the scene containing 3 different background planes. (c) shows the lines partitioning the planes. (d) Image (a) is filled-in using information from image (b) to remove the hole created due to the removed flag. Note that the shadow of the flag is present in the completed image as shadow region was not selected for removal.

without any user interaction. Ours is an attempt to use multiview information for scene inference and video completion. We showed results on scenes with piecewise planar background near the object to be removed. The technique can be easily extended to more planes as long as the dominant motion segmentation can be achieved. We also demonstrated the application of the technique for image completion for images taken with widely distant viewpoints.

The geometric information we used give better segmentation of multiple motions. The motions are segmented at the pixel level without region growing or interpolation, unlike the motion segmentation performed in the image space. Motion inpainting methods can work well for scenes with a multiple planes or non-textured surfaces. Combining the geometric information with motion inpainting will be the most promising one for scenes with multiple planes. The advantage of the motion inpainting techniques lies in its applicability to large number of scenes. We propose to investigate the problem further in that direction. The use of user interaction to understand the scene better is also an interesting problem.

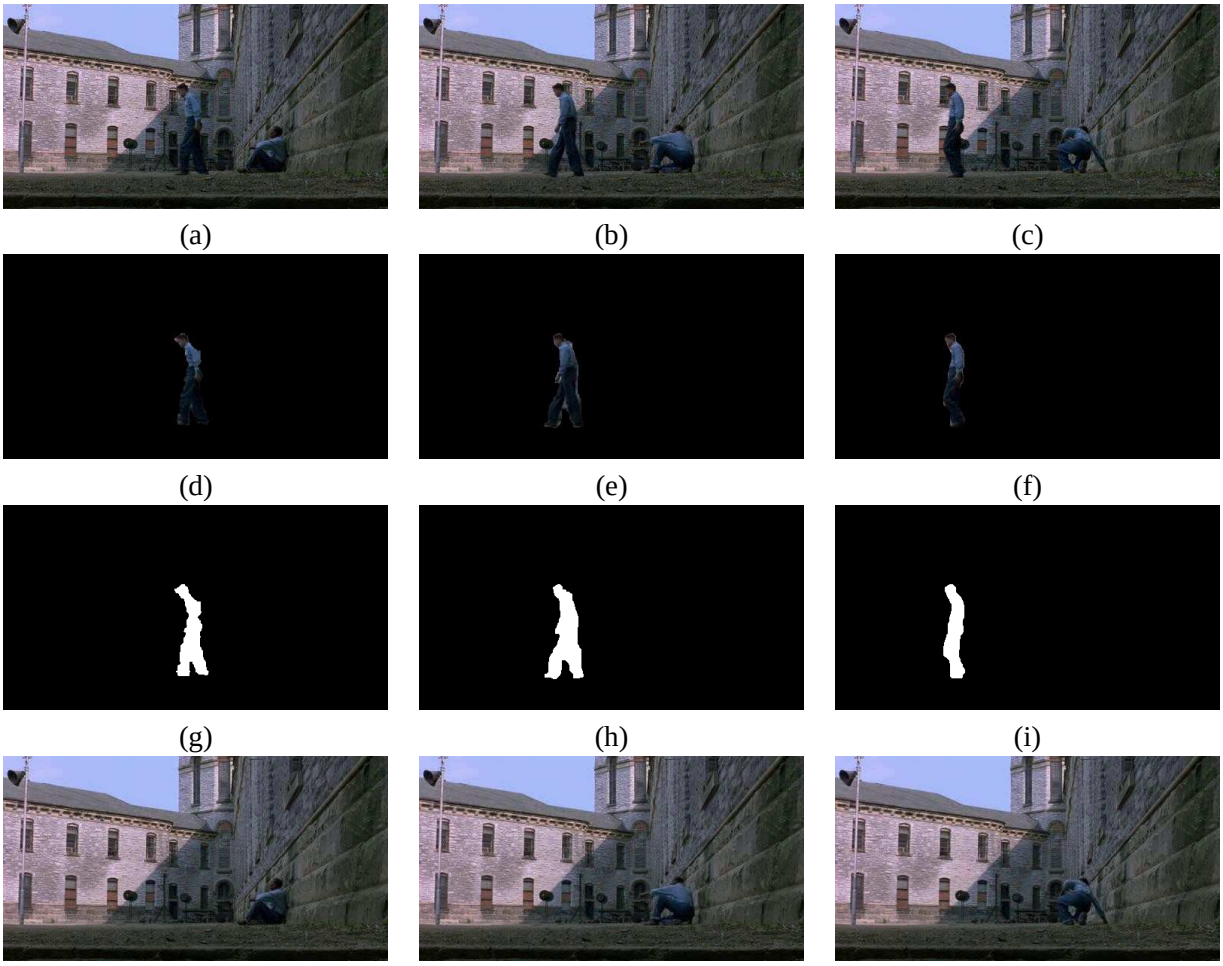


Figure 4.7 The video object removal technique applied for removing an actor from a clip from the movie *Shawshank Redemption.* (a,b,c) are the input frames. (d,e,f) and (g,h,i) show the extracted masks and the layers using the layer extraction technique presented in Chapter 3. (j,k,l) show the output frames where the actor walking across the scene is removed from the video.

Chapter 5

Discussions, Conclusions and Future Work

In this thesis, we presented two new algorithms. The first one is for object segmentation in videos. Our approach allows for segmenting objects in a spatio-temporal way by combining 3D graph cuts based segmentation with a tracking approach. We successfully handle segmentation of objects with complex motion models, as the method doesn't depend on motion model estimation. Thus our approach handles a larger set of objects as compared to non-rigid objects. Our method, to the best of our knowledge, is the first attempt at combining tracking to the graph cuts algorithm for obtaining seeds in the intermediate frames of a video. Robust tracking provides the seeds across the intermediate frames of the graph which results in optimal boundary via graph cuts minimization.

Our approach was developed with the objective of layer extraction, where precise alpha values at the boundaries are not needed. However, the layers obtained by this method can be easily used to generate the trimap, which is required as input by most matting algorithm. We have demonstrated the result of applying matting using the trimap obtained by our algorithm and the results are encouraging.

Some recent approaches have proposed graph cuts application with regions as primitives. Wang *et al* [77] demonstrated the application of graph cuts minimization on such a graph with pixel clustering at inter-frame and intra-frame level. Though the technique overall takes significant amount of time due to the clustering step, the interaction during graph cuts and iterations of graph cuts are much faster. Assuming regions to be of size of around 100 pixels each. The number of nodes can easily be brought down by a factor of 100. This produces a significant time improvement. Region level tracking has been demonstrated recently via use of affine invariant feature detectors [49, 50] and methods based on geometric hashing. Region level tracking and graph cuts can therefore provide two advantage to us. Firstly, the tracking at region level can be used to seed much larger part of the graph in the target image. Secondly, efficiency can be improved by working on a region level primitive instead of pixels. As has been demonstrated in many applications of multiresolution techniques like image registration, texture synthesis or more recently in matting [43], multiresolution application of the 3D graph cuts algorithm holds good promises of performance gain.

As future work in this direction, we would like to explore the use of region-level primitives as nodes for the 3D graph. One interesting direction with respect to object segmentation from videos is the

use of motion clusters along with color based approach. Though such approaches have been tried in automatic cases [39, 81], interactive correction-based methods for segmentation are yet to be explored. Another interesting direction for investigation is in use of class specific image information, which has been shown to be more effective in object segmentation in images [44, 58]. The efficiency of these algorithms can easily be improved for videos due to the temporal continuity assumption which brings down the search space drastically.

Our video completion method is a geometry based approach. The use of multiview information for determining the occluded region in one frame from the neighboring frames is promising due to the precise (non-approximate) calculation of a pixel's value. Compared to the motion flow based methods which involve approximation or estimation of motion flow for the particles inside the hole, our method uses exact information for the pixel available from the other frames.

Though motion segmentation using 2D motion vectors is popular, finding motion flow in smooth regions accurately is a difficult problem. Our approach, however, doesn't depend completely on the motion vectors, in fact we use motion vectors clustering only to estimate the homography of the planes between the two frames. We can therefore replace the dependence on motion flow estimation using other techniques like the one proposed by Jain and Jawahar [32] where the homography for two images is found using contours instead of point correspondences. The determination of the exact partitioning boundary can still be performed using the geometric formulation used in our approach.

The drawback of the technique is its limited applicability. The approach inherently involves motion segmentation of scenes into planar layers, which is not easy for complex scenes. For motion segmentation we use the optical flow information between two frames. The success of motion segmentation depends on the separability of the motion vectors. In practice, it is observed that the thresholding parameters for most clustering problems cannot be determined easily. For instance, the parameter for thresholding inliers in RANSAC algorithm may vary across scenes. We believe that the use of *motion history* in initializing the point set for RANSAC will improve the segmentation results.

Exemplar methods based on SSD [35, 36, 79] are very restrictive in the domain of videos. The assumption for periodic repetition of a patch doesn't fit well with perspective distortions, though an extension of approach followed in [57] may provide better results on more generic scene. The approach of Pavić *et al* clearly demonstrates that with the help of little user interaction the technique can be extended to a very large set of images [57]. The interface effectively allows the user to select the target patch's position and size and a real-time optimal source patch search is shown to the user. The results demonstrated bring up possibilities for further research in how user interaction can help to get more plausible solutions. The problem of video completion can also benefit from user interaction. The 3D scene information, if available, can be used in an interesting way. Once a correspondence between image frame and the 3D model can be established removing objects from the video would be essentially equivalent to rendering the scene without the pixels belonging to the object. Our initial investigation on structure from motion techniques suggests that a full 3D reconstruction from a video is still an unsolved problem. However, with the use of some information or interaction, reconstruction can be

achieved. We would like to explore ways in which simple user interaction can provide good deal of 3D information about the scene which can be used in the completion. A combination of geometry and optical flow interpolation would have the advantage of being generic and more accurate.The recent work of Shiratori *et al* [69], where they demonstrate the combination of exemplar based method with motion vector interpolation techniques, further motivates investigation in this direction.

An area still left unexplored by the research community in field of video completion is the use of custom setup to make the problem easier. Most of the current work has been focusing on making the video completion work on more and more general videos. Experience from the work in the field of video matting suggests that much better mattes are obtained if the environment can be setup in a particular way. Defocus matting [53] proposed use of three camera placed at different view points. Video completion problem can similarly be benefited by an appropriate setup. The use of multiple cameras to capture same scene from different view points provide equivalent information as that of view registration across multiple frames. The special application of this technique comes from its use in creating special effects. The technique will also be usable in the case where the background of the scene is difficult to customize according to the needs.

Related Publications

1. Vardhman Jain and P. J. Narayanan, ”Layer Extraction using Graph Cuts and Feature Tracking”, In *Proceedings of Third International Conference on Visual Information Engineering (VIE) 2006*, Bangalore 2006 *to appear*

2. Vardhman Jain and P. J. Narayanan, ”Video Completion for Indoor Scenes”, Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP 2006 .*UNDER REVIEW*.

3. Vardhman Jain and P. J. Narayanan, ”Tracking-Based Object Segmentation and Video Completion for Indoor Scenes”, Image and Vision Computing. *UNDER PREPARATION*.

Bibliography

[1] ADOBE Systems Incorp. 2002. Adobe photoshop user guide.

[2] Anubhav Agarwal, C.V. Jawahar, and P.J. Narayanan. A survey of planar homography estimation techniques. Technical Report IIIT TR-2005-14, International Institute of Information Technology, Hyderabad, June 2005.

[3] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. *ACM Transactions of Graphics*, 23(3):294–302, 2004.

[4] C. Ballester, V. Caselles, J. Verdera, M. Bertalmio, and G. Sapiro. A variational model for filling-in gray level and color images. In *International Conference on Computer Vision*, pages I: 10–16, 2001.

[5] J.R Bergen, P. Anandan, K.J Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, pages 237–252, London, UK, 1992. Springer-Verlag.

[6] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH, ACM Transaction on Graphics, Conference on Computer graphics and interactive techniques*, pages 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[7] M. Bertalmio, L.A. Vese, G. Sapiro, and S.J. Osher. Simultaneous structure and texture image inpainting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages II: 707–712, 2003.

[8] Kiran S. Bhat, Steven M. Seitz, Jessica K. Hodgins, and Pradeep K. Khosla. Flow-based video synthesis and editing. *ACM Trans. Graph.*, 23(3):360–363, 2004.

[9] P. Bhat, K.C. Zheng, N. Snavely, A. Agarwala, M. Agrawala, M.F. Cohen, and B. Curless. Piecewise image registration in the presence of multiple large motions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages II: 2491–2497, 2006.

[10] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. In *International Conference on Computer Vision*, pages 377–384, 1999.

[11] Y.Y. Boykov and M.P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *International Conference on Computer Vision*, pages I: 105–112, 2001.

[12] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[13] T. Chan, S. Jianhong(Jackie), and L. Vese. Variational PDE models in image processing. Technical report, UCLA Math CAM, 2003.

[14] Y.Y Chuang, A. Agarwala, B. Curless, D.H Salesin, and R. Szeliski. Video matting of complex scenes. In *SIGGRAPH, ACM Transaction on Graphics, Conference on Computer graphics and interactive techniques*, pages 243–248, New York, NY, USA, 2002. ACM Press.

[15] Y.Y Chuang, B. Curless, D.H Salesin, and R. Szeliski. A bayesian approach to digital matting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271. IEEE Computer Society, December 2001.

[16] Dorin Comaniciu and Peter Meer. Mean shift analysis and applications. In *International Conference on Computer Vision*, volume 2, page 1197, Washington, DC, USA, 1999. IEEE Computer Society.

[17] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

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

[19] A.A Efros and W.T Freeman. Image quilting for texture synthesis and transfer. *SIGGRAPH, ACM Transaction on Graphics, Conference on Computer graphics and interactive techniques*, pages 341–346, August 2001.

[20] A.A Efros and T.K Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, pages 1033–1038, Corfu, Greece, September 1999.

[21] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal on Computer Vision*, 59(2):167–181, 2004.

[22] M.A Fischler and R.C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communication of the ACM*, 24(6):381–395, 1981.

[23] David Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, May 2003.

[24] Friedrich Fraundorfer, Konrad Schindler, and Horst Bischof. Piecewise planar scene reconstruction from sparse correspondences. *Image and Vision Computing*, 24(4):395–406, April 2006.

[25] M. Gleicher. Image snapping. In *SIGGRAPH, ACM Transaction on Graphics, Conference on Computer graphics and interactive techniques*, pages 183–190, New York, NY, USA, 1995. ACM Press.

[26] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[27] R.I Hartley. In defence of the 8-point algorithm. In *International Conference on Computer Vision*, page 1064, Washington, DC, USA, 1995. IEEE Computer Society.

[28] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH, ACM Transaction on Graphics, Conference on Computer graphics and interactive techniques*, pages 229–238, 1995.

[29] P. Hillman, J. Hannah, and D. Renshaw. Alpha channel estimation in high resolution images and image sequences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages I:1063–1068, 2001.

[30] D. Hoiem, A. Efros, and M. Hebert. Automatic photo pop-up. In *SIGGRAPH, ACM Transaction on Graphics, Conference on Computer graphics and interactive techniques*, 2005.

[31] Intel. Open computer vision library. http://www.intel.com/technology/computing/opencv/index.htm.

[32] Paresh Jain and C. V. Jawahar. Homography Estimation from Planar Contours. In *Proc. of Third International Symposium on 3D Data Processing, Visualization and Transmission*, 2006.

[33] Vardhman Jain and P.J Narayanan. Layer extraction using graph cuts and feature tracking. In *Proceedings of the Third International Symposium on Visual Information Engineering (VIE) 2006*, 2006.

[34] J. Jia and C.K. Tang. Image repairing: robust image synthesis by adaptive nd tensor voting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages I: 643–650, 2003.

[35] J. Jia, T.P Wu, Y.W Tai, and C.K Tang. Video repairing: Inference of foreground and background under severe occlusion. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, 2004.

[36] Yun Tao Jia, Shi-Min Hu, and Ralph R. Martin. Video completion using tracking and fragment merging. *The Visual Computer*, 21(8-10):601–610, 2005.

[37] B. Johansson. View synthesis and 3D reconstruction of piecewise planar scenes using intersection lines between the planes. *International Conference on Computer Vision*, 1:54–59, September 1999.

[38] S. Kang, T. Chan, and S. Soatto. Landmark based inpainting from multiple views. Technical Report UCLA CAM Report 02-11, UCLA, 2002.

[39] Qifa Ke and Takeo Kanade. A subspace approach to layer extraction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

[40] A. Kokaram, B. Collis, and S. Robinson. A bayesian framework for recursive object removal in movie post-production. In *International Conference on Image Processing*, pages I: 937–940, 2003.

[41] Nikos Komodakis and Georgios Tziritas. Image completion using global optimization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[42] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, 2004.

[43] Anat Levin, Dani Lischinski, and Yair Weiss. A closed form solution to natural image matting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, June 2006.

[44] Anat Levin and Yair Weiss. Learning to combine bottom-up and top-down segmentation. In *European Conference on Computer Vision*. IEEE Computer Society, June 2006.

[45] Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *International Conference on Computer Vision*, page 305, Washington, DC, USA, 2003. IEEE Computer Society.

[46] Y. Li, J. Sun, and H.Y Shum. Video object cut and paste. *ACM Trans. Graph.*, 24(3):595–600, 2005.

[47] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, 2004.

[48] Ce Liu, Antonio Torralba, William T. Freeman, Frédo Durand, and Edward H. Adelson. Motion magnification. *ACM Trans. Graph.*, 24(3):519–526, 2005.

[49] David Lowe. SIFT keypoint detector. http://www.cs.ubc.ca/ lowe/keypoints/.

[50] David Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal on Computer Vision*, 2003.

[51] David G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, Corfu, 1999.

[52] Y. Matsushita, E. Ofek, Tang.X., and H.Y Shum. Full frame video stabilization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[53] Morgan McGuire, Wojciech Matusik, Hanspeter Pfister, John F. Hughes, and Frédo Durand. Defocus video matting. *ACM Trans. Graph.*, 24(3):567–576, 2005.

[54] E.N. Mortensen and W.A. Barrett. Interactive segmentation with intelligent scissors. *GMIP*, 60(5):349–384, September 1998.

[55] M. T. Orchard and C. A. Bouman. Color Quantization of Images. *IEEE Transactions on Signal Processing*, 39(12):2677–2690, 1991.

[56] The Visual Geometry Group Oxford. Affine covariant feature detectors. http://www.robots.ox.ac.uk/ vgg/research/affine/.

[57] D. Pavi, V. Schnefeld, and L. Kobbelt. Interactive image completion with perspective correction. In *Pacific Graphics*, 2006.

[58] M. Pawan Kumar, Philip H. S. Torr, and Andrew Zisserman. Obj cut. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume I, pages 18–25, Washington, DC, USA, 2005. IEEE Computer Society.

[59] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2003.

[60] The GNU Image Manipulation Program. http://gimp.sourceforge.net.

[61] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[62] M.A Ruzon and C. Tomasi. Alpha estimation in natural images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1018–1025, 2000.

[63] Peter Sand and Seth Teller. Video matching. *ACM Trans. Graph.*, 23(3):592–599, 2004.

[64] Arno Schödl and Irfan A. Essa. Controlled animation of video sprites. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 121–127, New York, NY, USA, 2002. ACM Press.

[65] Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. Video textures. In Kurt Akeley, editor, *SIGGRAPH, ACM Transaction on Graphics, Conference on Computer graphics and interactive techniques*, pages 489–498. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[66] Jianbo Shi and Jitendra Malik. Motion segmentation and tracking using normalized cuts. In *International Conference on Computer Vision*, pages 1154–1160, 1998.

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

[68] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seattle, June 1994.

[69] T. Shiratori, Y. Matsushita, S.B Kang, and X. Tang. Video completion by motion field transfer. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[70] Alvy Ray Smith and James F. Blinn. Blue screen matting. In *SIGGRAPH, ACM Transaction on Graphics, Conference on Computer graphics and interactive techniques*, pages 259–268, New York, NY, USA, 1996. ACM Press.

[71] J. Sun, J. Jia, C.K. Tang, and H.Y Shum. Poisson matting. *ACM Trans. Graph.*, 23(3):315–321, 2004.

[72] Jian Sun, Lu Yaan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. *ACM Trans. Graph.*, 24(3), 2005.

[73] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. *Computer Graphics*, 31(Annual Conference Series):251–258, 1997.

[74] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[75] E. Vincent and R. Laganiere. Detecting planar homographies in an image pair. In *Symposium on Image and Signal Processing and Analysis (ISPA01)*, 2001.

[76] Lee Vincent and Pierre Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.

[77] J. Wang, P. Bhat, A. Colburn, M. Agrawala, and M.F Cohen. Interactive video cutout. *ACM Trans. Graph.*, 24(3):585–594, 2005.

[78] J.Y.A. Wang and E.H. Adelson. Layered representation for motion analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 361–366, 1993.

[79] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages I: 120–127, 2004.

[80] J. Wills, S. Agarwal, and S. Belongie. What went where. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 37–44, 2003.

[81] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages II: 972–979, 2004.

[82] DengSheng Zhang and Guojun Lu. Segmentation of moving objects in image sequence: A review. In *CSSP*, 2000.

[83] Y. Zhang, J. Xiao, and M. Shah. Motion layer based object removal in videos. *WACV/Motion*, 01:516–521, 2005.