Computer Vision for the Structured Representation and Stylisation of Visual Media Collections

Tinghuai Wang

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey

Centre for Vision, Speech and Signal Processing
Faculty of Engineering and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

July 2012

Summary

The proliferation of digital cameras in commodity consumer devices, and the social trend for casually capturing and sharing media, has led to an explosive growth in personal visual media collections. However this wealth of digital material is infrequently accessed beyond the point of initial capture or sharing, and often lies dormant gathering digital dust in the media repository. This thesis proposes novel Computer Vision and Computer Graphics techniques to release the value in personal media collections - investigating new ways to stylise and present images and video in such collections.

First, personal visual media tends to be shot casually in varied and challenging capture conditions by amateur operators. This necessitates some interactive manipulation prior to presentation. This thesis contributes a novel solution for editing such amateur home video into succinct clips of this nature, using a parse-tree representation of the video editing process. We also enable interactive manipulation of still images through a novel object segmentation algorithm dubbed TouchCut which enables object selection with a single touch and is intended for direct media manipulation on commodity media capture devices such as touch-screen digital cameras.

Second, there is an interaction barrier to digital media. The casual nature of personal media encourages the capture of significantly larger collections than traditional media. This thesis explores the application of artistic stylisation to create digital ambient displays (DADs) of personal media in the style of cartoons, paintings and paper-cut out. Underpinning this contribution are two new algorithms for video segmentation that enforce temporal coherences within the stylised video. Furthermore, we explore how structuring the media collection hierarchically within the DAD can promote interest and engagement with the collection.

Third, personal media collections often contains images of friends or family members. The artistic stylisation of such content using existing approaches rarely results in acceptable output. We propose a novel example-based approach to portrait stylisation driven by a high level model of facial structure that gives rise to improved aesthetics and enables the example-based rendering of a diverse range of portrait styles.

This work was supported by the Hewlett Packard Laboratories Innovation Research Program.

Key words: video segmentation, interactive image segmentation, video composition, video editing, visual media browsing, non-photorealistic rendering.

Email: tinghuai.wang@surrey.ac.uk

WWW: http://personal.ee.surrey.ac.uk/Personal/Tinghuai.Wang/

Acknowledgements

My sincerest gratitude goes first to my supervisor John Collomosse who took me into this incredible journey of PhD. I am grateful for his constant guidance, insightful advice, patience and encouragement. He not only teaches me how to carry out rigorous and innovative scientific research, but also is willing to spend considerable time and effort to dig into technical details. His expertise and insight in computer vision and graphics greatly contributed to shaping this thesis.

I would like to thank my second supervisor Adrian Hilton and Jean-Yves Guillemaut (who got me started with Graph Cut). I would like to thank my colleagues for volunteering with the user studies and data capture for project. Thanks also to other members in CVSSP at the University of Surrey, for making me feel at home.

My sincere gratitude to David Slatter, Phil Cheatle, Darryl Greig and Andy Hunter from HP Labs Bristol for their constant and valuable input and feedback on the IRP project during the three years. I am also very grateful to Bo Han who provided me the opportunity to work on an exciting project in Sony China Research Lab. I would also like to thank Jan Eric Kyprianidis for providing his code for stylisation and structure tensor, with whom we also shared many conversations and opinions on the survey paper for artistic rendering. I would like to thank my PhD examiners Paul Rosin and Janko Calic.

I am grateful to HP Labs Bristol for funding my PhD research under HP's IRP programme.

Last but first in my heart, my eternal gratitude goes to my family for their perpetual love, encouragement and support.

Contents

PUBLICATIONS

This thesis presents research that has appeared in the following papers:

Chapter 2

[116] J.-E. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg. State of the 'art': A taxonomy of artistic stylization techniques for images and video. IEEE Trans. Vis. Comput. Graph., 19(5):866-885, 2012.

Chapter 3

[223] T. Wang, A. Mansfield, R. Hu, and J. Collomosse. An evolutionary approach to automatic video editing. In Proc. 6th European Conf. on Visual Media Production (CVMP), pp. 127-134, 2009.

Chapter 4

[222] T. Wang, B. Han, and J. Collomosse. Touchcut: Single-touch object segmentation driven by level set methods. In Proc. ICASSP, 2012, to appear.

[221] T.Wang, B. Han, and J. Collomosse. Touchcut: Fast image and video segmentation using single-touch interaction. submitted to Computer Vision and Image Understanding (CVIU), 2012.

Chapter 5

[220] T. Wang, J.-Y. Guillemaut, and J. P. Collomosse. Multi-label propagation for coherent video segmentation and artistic stylization. In Proc. ICIP, pp. 3005-3008, 2010.

[219] T. Wang, J. P. Collomosse, D. Slatter, P. Cheatle, and D. Greig. Video stylization for digital ambient displays of home movies. In Proc. NPAR, pp. 137-146, 2010.

[218] T. Wang, J. P. Collomosse, R. Hu, D. Slatter, D. Greig, and P. Cheatle. Stylized ambient displays of digital media collections. Computers & Graphics, 35(1):54-66, 2011.

Chapter 6

[216] T. Wang and J. Collomosse. Progressive motion diffusion of labeling priors for coherent video segmentation. IEEE Trans. Multimedia, 14(2):389-00, April 2012.

Chapter 7

[217] T. Wang, J. Collomosse, A. Hunter, and D. Greig. Learnable Stroke Models for Example-based Portrait Painting. In Proc. BMVC, 2013.

The following publication has resulted from other work and is not presented here:

[98] R. Hu, T. Wang, and J. Collomosse. A bag-of-regions approach to sketch-based image retrieval. In Proc. ICIP, pp. 3661-3664, 2011.

Part I

Introduction

Chapter 1

Introduction

In recent years we have witnessed the explosive growth of consumer imaging devices. Vast personal archives of images and video are being amassed through the casual capture of imagery on cameras, phones and other commodity devices. These digital media collections are intrinsically more accessible than physical archives - for example, a box of old photos in the attic. Yet, beyond the time of capture or initial sharing, content often lies infrequently accessed gathering digital dust in media repositories. This is unfortunate as such media collections often capture memories of friends, family and past experiences that hold significant value to us, but from which we are isolated by technological barriers. The proliferation of this wealth of under-used visual content motivates new techniques to browse and visualise large personal media collections.

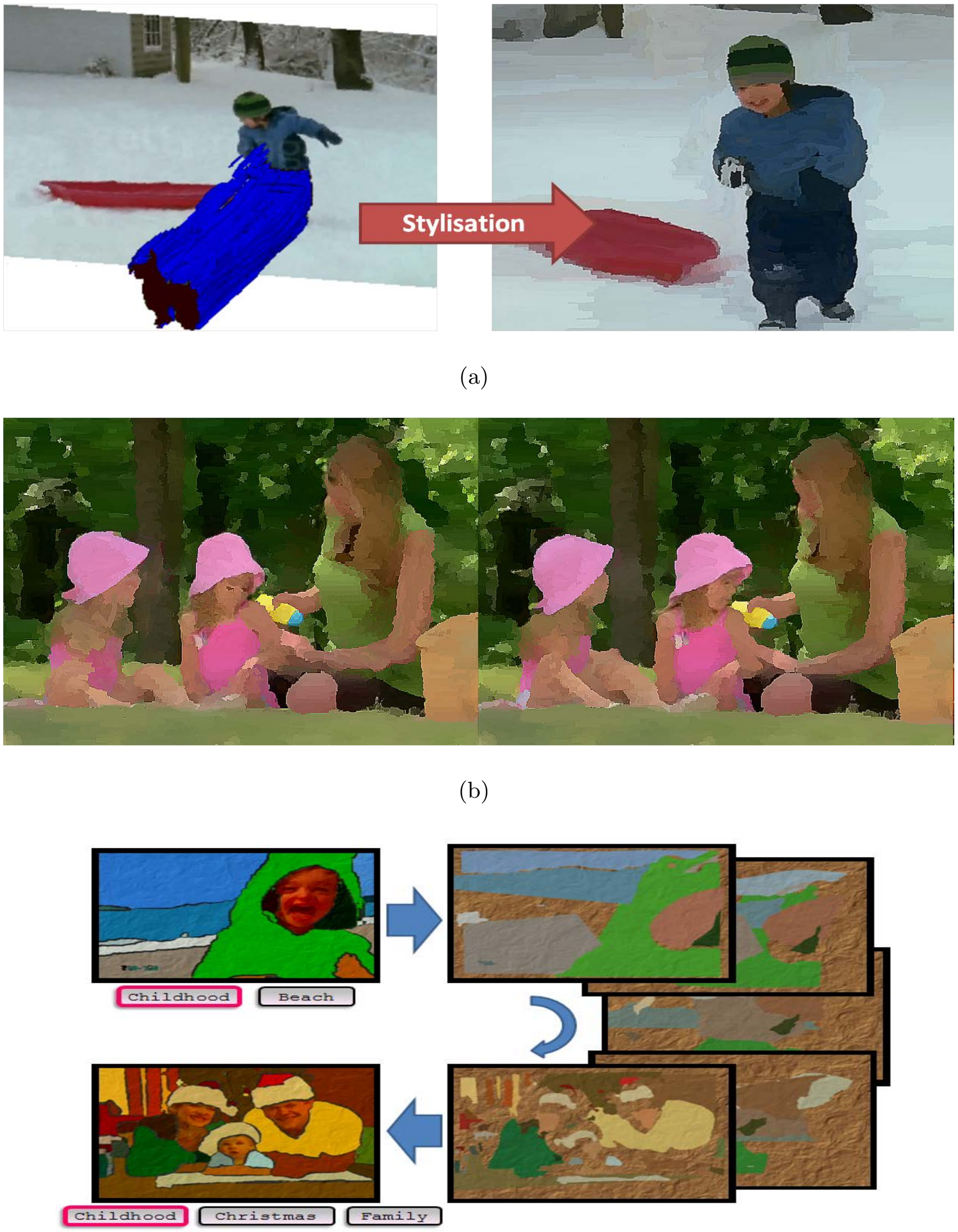
A significant step to increasing the impact of personal visual media collections is to ensure they contain interesting and aesthetically compelling content. Personal media is typically captured by amateurs possessing varying levels of aesthetic skill, requiring substantial manipulation to be comparable - in terms of aesthetics and succinctness - with professional work. There is a need to automatically structure the presentation of these media collections, as well as enhance the composition and appearance of individual collections items (images and videos) to enhance their aesthetic value. Recently, new forms of ambient display have begun to emerge around shared living spaces in the home. From digital photo frames to wall-scale large displays, these displays offer the opportunity to present media in an unobtrusive yet engaging manner analogous to hanging a physical photograph or painting on a wall. Such displays present both an opportunity and a challenge to devise intelligent ways to present digital media collections. This thesis thus proposes a solution for large personal media collections in the form of *Digital Ambient Displays* (DADs); always-on displays for living spaces that enable users to effortlessly visualize and rediscover their media collections.

A central theme of this thesis is the enhancement of the presentation of personal media collections - composition and appearance of individual media items and the structuring of whole media collections. Automatic video editing transforms raw video footage into salient and aesthetically pleasing video clips; these short, visually interesting clips form the atomic unit of stylisation and composition; robust video segmentation algorithm stably segments the visual structure in a scene which enables the coherent video stylisation and composition; portrait rendering enhances the aesthetics and impact of the stylisation of visual content containing faces.

1.1 Contributions of this Thesis

This thesis contributes several new Computer Vision (CV) and Computer Graphics (CG) techniques to enhance the presentation of personal visual media collections. Specifically, we develop new algorithms to parse representations of visual structure from users' images and videos (contributing to CV) and render i.e. present that content in stylised or more succinct forms (contributing to CG). Several representations are proposed at various levels of abstraction tailored to the requirements of the rendering technique developed.

The rendering techniques focused upon in this thesis are Non-Photorealistic, and aim to transform 2D content into synthetic artwork. Non-Photorealistic Rendering (NPR) is a mature sub-discipline within CG, established in the early nineties [79]; a discipline that has diversified to the extent that expressive rendering for the purposes of art and aesthetics is now often referred to independently as Artistic Rendering (AR) to distinguish from the broader definition of NPR. AR offers many advantages over photorealistic rendering, including its ability to manipulate media to stylise presentation, clarify shape, abstract away detail and focus attention. Over the years AR algorithms have begun to evolve from simple processes relying on low-level analysis, typically



(a)

(b)

(c)

Figure 1.1: Previewing some of the results of our algorithms: (a) Our video segmentation algorithm produces spatio-temporal coherent representation of sequence which facilitates production of (b) high quality video stylisation. (c) This parsed visual structure from sequence also facilitates video temporal composition adding aesthetic to personal media collections.

driven by image filtering, towards sophisticated processes measuring relative scene importance and analysing scene structure. A core contribution of this thesis is to develop novel algorithms for extracting and representing visual structure at different levels of abstraction to drive AR. Centred upon this theme the thesis explores three core hypotheses (H1-3), namely that:

- **H1.** Improving the stability of the structure extracted from video sequences beyond the state of the art enhances the temporal coherence of artistic renderings.
- **H2.** Structured presentation and visual stylisation of content in personal media collections enhances user engagement with that content.
- **H3.** New approaches for parsing visual structure can unlock new forms of stylisation so diversifying AR.

In investigating these hypothesis the thesis focuses on two currently under-researched topics within AR.

First, the stylisation and abstraction of video; arguably the most under-exploited form of personal media since it cannot be printed or displayed effectively in a static form. Artistic stylisation of video remains a challenging and open problem within AR. Many video AR approaches adopt a stroke based rendering (SBR) paradigm [92]; placing a multitude of rendering marks (i.e. strokes) upon the video canvas that must move coherently over time. The rendering marks are said to be temporally coherent when their movement matches that of the underlying video content and any flickering is absent. Unfortunately temporal coherence remains elusive since moving these marks over time coherently requires robust estimation of the dynamics of the underlying scene structure. This argument motivates the extraction of our structure representations from video. In addition to the stylisation of video we also explore the problem of temporally manipulating video for interest, through salience-driven editing and the temporal composition of video clips into sequences.

Second, the thesis addresses the stylisation of people, in particular portraits, which are frequently encountered in personal media collections yet which general AR algorithms perform particularly poorly on. It has long been suspected that the human ability to

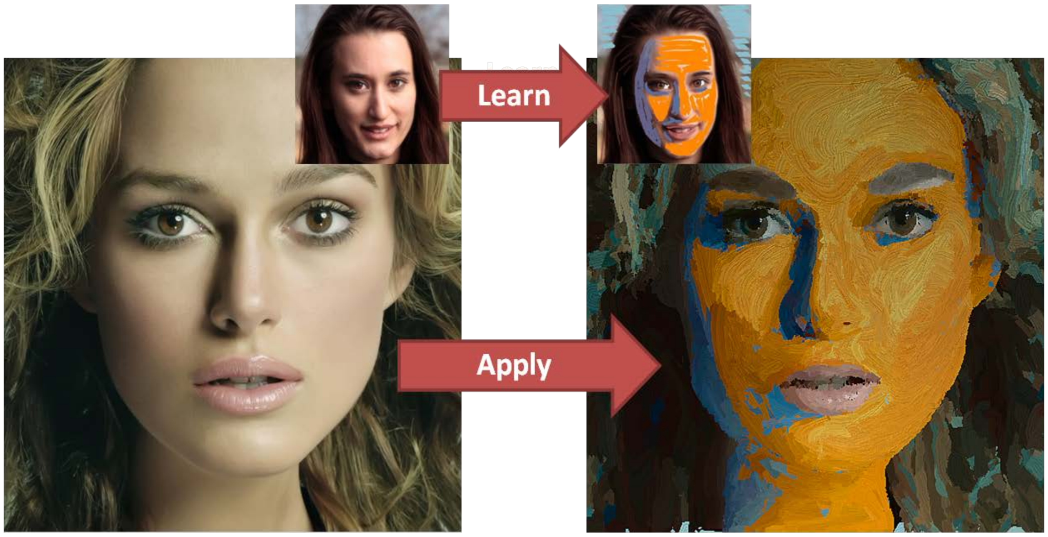


Figure 1.2: Based on a novel composition of higher-level features, our portrait painting algorithm is able to learn the brush and shading styles given a training pair.

recognise faces is hard-wired into our physiology, and a significant portion of visual brain function is dedicated to this task [195]. This may go some way to explain our sensitivity to violations in the natural structure of faces that often occurs when general AR algorithms are applied to portraits (often leading to blurring or distortion of facial features).

To support our general aim of AR driven by representations of visual structure, and to verify hypotheses H1-3, we have developed several novel algorithms which operate at different levels of abstraction and render a wide range of expressive styles on images and video, namely:

1. an algorithm to edit raw home movie footage into salient, aesthetically pleasing video clips. (Ch. 3, published as [223])

2. a fast interactive object segmentation algorithm driven by single finger touch for image and video. (Ch. 4, published as [222])

3. two algorithms based on multi-label graph cut for segmenting video into temporally coherent region maps. (Ch. 5 and 6, published as [220] and [216] respectively)

4. an algorithm to both stylise video into cartoons and paintings. (Ch. 5, published as [219] and [218])
5. an algorithm to automatically structure the media collection into a hierarchical representation based on visual content and semantics, facilitating intelligent media browsing in a coarse-to-fine manner, driven by user attention level. (Ch. 5, published as [218])
6. an algorithm to interpret human facial features which drives a user trainable algorithm for stylising photographs into portrait paintings. (Ch. 7)

The representations of visual structure parsed by our algorithms vary in their level of abstraction according to the rendering application, and are arranged in this thesis from low-level (Part II) to high-level (Part IV). We first describe our salience-driven algorithm for automatic video editing in Ch. 3 using heuristics driven by low-level measures derived from video content. The TouchCut image segmentation system integrates both low- and mid-level models of colour, texture and geometry in order to perform single-touch image segmentation in Ch. 4. Ch. 5 and 6 drive video stylisation algorithms using mid-level representations of video parsed from footage via our novel video segmentation algorithms. Ch. 7 for the first time introduces a stroke-based rendering algorithm for portraits using a high level structural model of the face fitted to photographs using an Active Shape Model (ASM) [48].

1.2 Application Domain

Our motivation is to produce a series of encompassing frameworks, capable of manipulating, visualising and stylising images and video sequences with a high degree of automation - whilst retaining user creativity in the process through high level control and parameterisation. Applications of this work lie most clearly within the creative industries, e.g. film special effects, animation and games, and in domestic software for the digital manipulation of media. For example, users might wish to create portraiture in a particular style from their own photographs (Ch. 7) or create an animated painting from their video (Chapters 5-6), yet lack the artistic training or the spare time to be able to perform this task. The systems we propose enable users to express their artistic requirements at a high level whilst providing the low-level automation that enables experimentation with different settings and eliminates the requirement for specialist training to produce artwork.

The algorithmic contributions presented in this thesis span both CG and CV, echoing the multi-disciplinary nature of AR and suggesting further applications for our algorithms designed to parse visual structure from images and video.

For example, consider the TouchCut segmentation algorithm developed in Ch. 4. This new approach to visual object cut-out using a single ‘touch’(2D coordinate) interaction provides a practical solution for image and video segmentation on emerging tablet and touch-screen devices. The TouchCut object selection mechanism has the potential to enable on-device editing of content for tracking, object replacement, or as we show in Ch. 4, the application of visual effects through stylisation. The video segmentation algorithms of Chapters 5 and 6 suggest a similar potential for broader impact. Our approach to video segmentation is sufficiently robust to deal with challenging motion and occlusion conditions. These algorithms not only facilitate the production of stylised animation from home movies, but also serve as an enabling tool for other computer vision applications such as video object retrieval and tracking. The ability to create robust and coherent video mattes could also benefit content post-production and rotoscoping in the creative industries, where such operations are still performed with a high degree of manual interaction.

1.3 Measuring Success

As any field matures, benchmarks and best practice methodologies emerge for the comparative evaluation of novel contributions that seek to improve upon existing techniques. In this thesis we use a variety of qualitative and quantitative methods to assess the efficacy of our algorithms. Image and video segmentation are two firmly established CV topics with several methodologies for evaluating accuracy in the literature. Over recent years the Berkeley Methodology has emerged as one of the most commonly cited benchmark for the evaluation of image segmentation algorithms, presenting both a dataset and quantitative metric (Berkeley F-measure) for this purpose. We adopt this metric both for image segmentation, and also to evaluate the accuracy of video segmentation on a per frame basis. For the latter we have developed our own dataset [1] and ground truth segmentation, since video with ground truth mark-up is largely absent in the literature and in any case commonly used videos (the garden, Foreman, etc.) are not representative of the personal video footage we seek to process.

A good video segmentation should not only exhibit quantifiable accuracy but also produce regions whose shape and neighbourhood topology evolve smoothly over time (i.e. with temporal coherence) whilst tracking the underlying video content. This is especially important as we use video segmentation to drive artistic stylisation of video where flicker can cause distraction and reduce the quality of animation. Assessing the temporal coherence of a segmentation (or a resulting rendering) remains a subjective task within the literature. Although the measurement of flicker might be quantifiable by a video-derived measure, such measures raise similar objections to the use of, say, PSNR for image comparison; they can often contradict human perception. Quantifying flicker is not an accepted practise in video segmentation or AR evaluation, rather we adopt a more qualitative and subjective side-by-side comparison of approaches to assess the relative presence of flicker in two videos.

AR is now well established, with many techniques proposed in the early 1990s and 2000s that broke ground implementing a wide range of artistic styles. As new AR techniques are proposed it becomes incumbent upon researchers to demonstrate improvement over the state of the art. Yet, objectively assessing the relative aesthetic merit of two renderings seems impossible for humans to agree on let alone encode within an algorithmic measure. Rather, a subjective comparison is typically used in AR to determine the relative aesthetic quality of two renderings, and we follow this process in our comparative evaluation on the basis that this is established practice within the field.

By contrast, it is easier to demonstrate breadth and capability of style; for example a system such as that of Ch. 7 where we demonstrate a trainable rendering system that

[1] `http://personal.ee.surrey.ac.uk/Personal/Tinghuai.Wang/TMM2011.html`

encompasses several styles. In such cases the variation in style within the generated renderings is prima facie evidence of the versatility of the system.

1.4 Structure of the Thesis

We now outline the structure of the thesis, summarising the principal contributions made in the subsequent chapters, and how they contribute to our core argument for structured representation and stylisation of visual media.

Part I — Introduction

Chapter 2 — Literature Review

A comprehensive literature survey of related work is presented in this chapter, forming observations on trends and identifying gaps in the literature.

Part II — Image and Video Manipulation

Chapter 3 — An Evolutionary Approach to Automatic Video Editing

We propose a novel algorithm for transforming raw home movie footage into concise, temporally salient video. We interpret the sequence of editing operations applied to footage as a novel structured representation, i.e. ‘program’, comprising cutting, panning and zooming constructs. We develop a Genetic Programming (GP) framework for representing and evolving such programs. Under this framework, the search for an aesthetically pleasing video edit becomes a search for the optimal genetic program. Our aesthetic criterion promotes the inclusion of people in shots, whilst penalising rapid shot changes or shot changes in the presence of camera motion. We demonstrate that our structured representation of editing operations driven by salience measure bridges the gap between low-level visual feature and high-level video editing grammar.

Chapter 4 — TouchCut: Fast Object Segmentation using Single-Touch Interaction

We present TouchCut; a robust and efficient algorithm for segmenting image and video sequences with minimal user interaction. Our algorithm requires only a single finger touch to identify the object of interest in the image or first frame of video. Our approach is based on a level set framework, with an appearance model fusing edge, region texture and geometric information sampled local to the touched point. We first present our image segmentation solution, then extend this framework to progressive (per-frame) video segmentation, encouraging temporal coherence by incorporating motion estimation and a shape prior learned from previous frames. This new approach to visual object cut-out provides a practical solution for image and video segmentation on compact touch screen devices, facilitating spatially localized media manipulation. TouchCut extracts a stable structure from video sequence which facilitates a wide range of higher-level applications. We describe such a case study, enabling users to selectively stylise video objects to create a hand-painted effect. TouchCut serves our argument that extracting a stable representation of visual structure from video sequences enhances the temporal coherence of the resulting artistic renderings.

Part III — Video Stylisation

Chapter 5 — Stylised Ambient Displays of Visual Media Collections

We develop a system to breathe life into home digital media collections, drawing upon artistic stylisation to create a “Digital Ambient Display” that automatically selects, stylises and transitions between digital contents in a semantically meaningful sequence. We present a novel algorithm based on multi-label graph cut for segmenting video into temporally coherent region maps. These maps are used to both stylise video into cartoons and paintings, and measure visual similarity between frames for smooth sequence transitions. The system automatically structures the media collection into a hierarchical representation based on visual content and semantics. Graph optimization is applied to adaptively sequence content for display in a coarse-to-fine manner, driven by user attention level (detected in real-time by a webcam). Our system is deployed on embedded hardware in the form of a compact digital photo frame. We demonstrate that the improved stability of the structure extracted from video sequences enhances the temporal coherence of artistic renderings. We evaluate our media sequencing algorithm via a small-scale user study, indicating that our structured presentations and artistic stylisation convey a more compelling media consumption experience than simple linear 'slide-shows' and significantly improved user engagement.

Chapter 6 — Probabilistic Motion Diffusion of Labeling Priors for Coherent Video Segmentation

A robust algorithm for temporally coherent video segmentation is proposed in this chapter. Our approach is driven by multi-label graph cut applied to successive frames, fusing information from the current frame with an appearance model and labelling priors propagated forwarded from past frames. We propagate using a novel motion diffusion model, producing a per-pixel motion distribution that mitigates against cumulative estimation errors inherent in systems adopting 'hard' decisions on pixel motion at each frame. Further, we encourage spatial coherence by imposing label consistency constraints within image regions (super-pixels) obtained via a bank of unsupervised frame segmentations, such as mean-shift. We demonstrate quantitative improvements in accuracy over state-of-the-art methods on a variety of sequences exhibiting clutter and agile motion, adopting the Berkeley methodology for our comparative evaluation. The improved stability of the parsed visual structure from video sequences potentially enhances the temporal coherence of the resulting artistic renderings.

Part IV — Portrait Stylisation

Chapter 7 — Digital Raphael: Learnable Stroke Models for Example-based Portrait Painting

We introduce the Digital Raphael; a novel algorithm for stylising photographs into portrait paintings comprised of curved brush strokes. Rather than drawing upon a prescribed set of heuristics to place strokes, our system learns a flexible model of artistic style by analyzing training data from a human artist. Given a training pair — a source image and painting of that image — a model of style is learned by observing the geometry and tone of brush strokes local to image features. The feature composition process is driven by a Markov Random Field model to force the spatial coherence and structural context of the feature set. Style models local to facial features are learned using a semantic segmentation of the input face image, driven by a combination of an Active Shape Model and Graph-cut. We evaluate style transfer between a variety of training and test images, demonstrating a wide gamut of learned brush and shading styles using minimal training data. This work shows that a high-level domain specific structure enables highly aesthetic quality renderings, and ability to learn and reproduce a wide gamut of styles.

Part V — Conclusions

Chapter 8 — Conclusions and Further Work

We summarise the contributions of the thesis in this chapter, and discuss how the results of the algorithms we have developed support our central argument for structured representation and stylisation in visual media. We suggest possible avenues for the future development.

Chapter 2

Literature Review

The structured representation and stylisation of visual media have been extensively studied in computer vision and graphics community. In this chapter we present a comprehensive literature survey of related work, forming observations on trends and identifying gaps in the literature. We explain the relevance of our research within the context of the reviewed literature.

2.1 Introduction

The work present in this thesis innovates in a range of topics within CV and CG. This literature review does not make attempt to give a complete and thorough survey to each area, but provides the technical background for the subsequent chapters by introducing notation and reviewing key techniques.

Sec. 2.2 introduces previous work in video editing and composition, which has gained momentum recently to add aesthetic to personal media collections. Sec. 2.3 aims to give a brief overview of image and video segmentation techniques in the literature. We broadly divide image segmentation into unsupervised and interactive approaches based on whether human intervention is involved. Video segmentation is discussed according to whether video data is processed as 3D volume or on a per frame basis. Recent systems involving human interaction are also reviewed. We introduce previous approaches to hierarchical structuring of media collection which is the key to interactively navigating collections in Sec. 2.4. Sec. 2.5 gives a broad overview of artistic rendering techniques to abstract and stylise visual data for the purposes of art and aesthetics. We focus our review upon the problem of stroke placement and anisotropic filtering, i.e. the areas this thesis innovate in, which can further be divided into the following categories based on the driving mechanisms: stroke-based techniques, region-based techniques, example-based techniques, image processing and filtering, video stylisation, and portrait painting.

2.2 Video Editing and Composition

Automated video editing is closely related to research on video summarisation, which has gained momentum in recent years. Many such algorithms rely on shot detection to extract representative key-frames from video [153]. Such techniques are well suited to movies exhibiting frequent cuts between shots, but are ill-suited to home videos (typically captured as a single lengthy shot). An alternative is [54] who model video as a trajectory through a high-dimensional appearance space, cutting key frames at points of high curvature.

Techniques that summarise video into shorter videos by ‘cutting’ frames have been proposed. Lienhart defines a visual quality metric, creating an automatic digest of home videos by selecting portions of video with good quality and inserting transition effects [133]. Girgensohn *et al.*’s semi-automatic “Hitchcock” system [67, 66] is similar to [133], but defines quality in terms of camera stability; we incorporate a similar cue in our work in Ch. 3 to incorporate ‘cutting’ operation in order to enhance the interest or aesthetic appeal of video. Simakov *et al.* [193] propose a bi-directional similarity measure to summarise images or video. Hua *et al.* propose an automatic video editing system that seeks to cut video to synchronise motion in selected sub-shots with music tempos [99]. Attention models for video summarization were studied in [142, 148], integrating visual, auditory, and linguistic cues. However, the gap between high-level video editing operations and low-level visual feature has not been investigated in these approaches which makes them impractical for personal media archives.

Most recently researchers have looked beyond cutting, to the framing of video content (e.g. zooming/cropping). Al-Hames *et al.* controlled multiple cameras to select and zoom-in on meeting participants to "direct" a live video stream of a meeting [4]. Hospedales and Williams recently explored Bayesian networks to learn director preferences for similar real-time editing of streamed video [97]. Such techniques necessarily make temporally local editing decisions. In Ch. 3 we present a Genetic Programming (GP) approach which performs global optimization over all frames of a pre-captured video.

Video temporal composition concerns the temporal sequencing and transitioning of clips in order to synthesize a new video. Video temporal composition was first proposed by Schodl *et al.* [185], within the scope of a single video and based upon visual similarity only. Much as motion graphs [111] construct a directed graph that encapsulates connections among motion capture fragments, so the Video Textures of Schodl *et al.* create a graph of video fragments that may be walked in perpetuity to create a temporal composition. Our proposed approach in Ch. 5 to composition borrows from the graph representations of [185, 111], but using a *hierarchical* representation, comprising multiple videos, and measuring similarity both visually and semantically.

2.3 Image and Video Segmentation

In computer vision, segmentation refers to the process of partitioning a digital image into multiple segments, i.e. superpixels, such that pixels in the same segment share certain visual characteristics. Image segmentation is typically used to locate objects and boundaries in images which naturally parses the image into a structured representation of the scene. Video segmentation aims to partition pixels into spatio-temporal groups exhibiting coherence and consistency in both appearance and motion.

Segmentation remains a long-standing fundamental and inherently challenging computer vision problem. This section is aimed to capture general trend in this field via a comprehensive review of previous work.

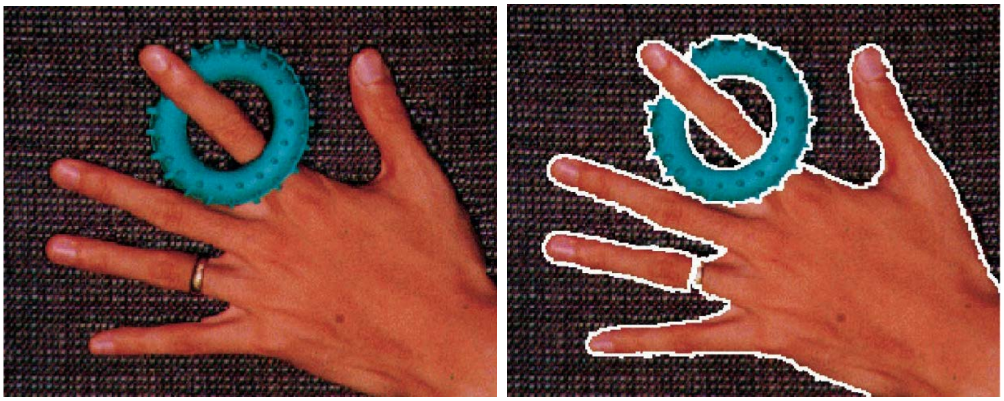


Figure 2.1: Mean shift image segmentation (right) on hand image (left) [45]. This algorithm performs density estimation in Luv space of colour image.

2.3.1 Image Segmentation

There has been a tremendous research effort for the past four decades dedicated to improving the robustness and efficiency of image segmentation, which can not fully be covered by this review. Image segmentation can further be partitioned into two categories: unsupervised, or automatic segmentation; and interactive, or semi-automated segmentation. Though this thesis contributes primarily to the former category, unsupervised segmentation is still strongly relevant and is briefly reviewed in this section. Refer to [157, 63] for some comprehensive surveys of unsupervised segmentation.

Unsupervised Approaches to Image Segmentation

Various methods [15, 52, 122, 158, 214] have been proposed adopting stochastic models to solving the unsupervised segmentation problem. For example, Belongie *et al.* [15] present a "blobworld" representation which provides a transformation from the raw pixel data to a small set of image regions using the Expectation-Maximization (EM) algorithm on combined color and texture features. Delignon *et al.* [52] address the generalized mixture estimation problem by defining a mixture and proposing its estimation based on Stochastic EM (SEM); it is then applied to the problem of unsupervised Bayesian image segmentation in a "local" and "global" way. Langan *et al.* [122] model an image as a doubly stochastic field in which the state or region map and the intensity data are each modeled as random fields. As a result of using a stochastic model based approach to image segmentation, the log-likelihood of the observed intensity image and estimated state map may be calculated. Panjwani and Healey [158] introduce an unsupervised segmentation algorithm using a Markov random field model with an efficient maximum pseudo likelihood scheme for estimating model parameters from image regions. Wang [214] proposes a hierarchical approach which at each step minimizes a cost function on the space of partitions with connected components of a graph.

Nonparametric density estimation approaches normally begin at each pixel and estimate the local density of similar pixels. As a general nonparametric density estimator, mean shift is a classical pattern recognition procedure proposed by Fukunage and Hostetler [65], and its efficacy on low-level vision tasks such as segmentation has been extensively exploited. Comaniciu and Meer [45] utilise it for continuity preserving filtering and image segmentation (shown in Fig. 2.1). Its properties were reviewed and its convergence on lattices was proven. Wang *et al.* [212] present an anisotropic kernel mean shift in which the shape, scale, and orientation of the kernels adapt to the local structure of the image or video. Paris and Durand [162] introduce the use of Morse theory to interpret mean shift as a topological decomposition of the feature space into density modes, and design an algorithm to compute mean-shift segmentations of images and videos based on the watershed technique.

Region growing approaches have also attracted the attention of segmentation research [186, 56, 95]. Shafarenko *et al.* [186] adapt the watershed transform to the LUV gradient of images with small color saliency, proposing a bottom-up segmentation algorithm that takes into consideration both color and texture properties of the image. Deng and Manjunath [56] propose to use the quantised image to compute an edge indicator and then apply region-growing method to segment the image. Hill *et al.* [95] introduce the concept of texture gradient and have used it to produce an effective watershed segmentation technique for natural images based on intensity and texture boundaries. They also implement a marker selection algorithm to counteract the problem of over-segmentation. Arbelaez *et al.* [7] propose a contour detector which combines multiple local cues into a optimisation framework based on spectral clustering and use the

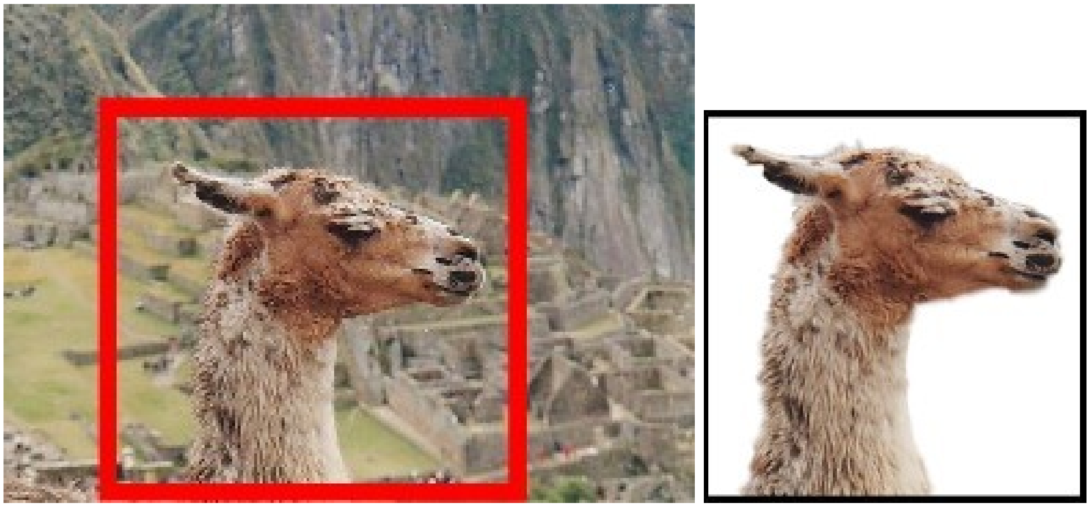


Figure 2.2: Interactive image segmentation driven by a bounding box using GrabCut [178]. GrabCut performs iterative graph-cut optimisation which reduces considerably the amount of user interaction needed to complete a segmentation task.

Oriented Watershed Transform (OWT) to producing a set of initial regions from contour detector output.

Rather than focusing on local features and their consistencies in the image data, graph partitioning approaches [188, 143, 60] aim to extract a global impression of an image. Shi and Malik [188] propose a novel global criterion, the normalized cut, for segmenting the graph. The normalized cut criterion measures both the total dissimilarity between the different groups as well as the total similarity within the groups. Malik *et al.* [143] propose an algorithm for partitioning grayscale images into disjoint regions of coherent brightness and texture, using the spectral graph theoretic framework of normalized cuts. Felzenszwalb and Huttenlocher [60] develop an efficient segmentation algorithm based on a predicate which measures the evidence for a boundary between two regions using a graph-based representation of the image, which produces segmentations that satisfy global properties. Tolliver and Miller [202] introduce a family of spectral partitioning methods in which edge separators of a graph are produced by iteratively reweighting the edges until the graph disconnects into the prescribed number of components.

Interactive Approaches to Image Segmentation

Over the past decade a number of successful interactive approaches have emerged, enabling the user to *seed* or '*scribble*' on part of the desired object and background to initialize the segmentation [22, 139, 74, 11, 126, 194, 107, 169, 179]. This approach is intuitive and generally tolerant of low accuracy user input, though requires the user to trace a contour contacting multiple points in the image. Drawing a bounding box [178, 126] to constrain the spatial extent of object is simpler in many cases, taking two mouse clicks to specify the box (shown in Fig. 2.2). Yet scribble-based corrections are often needed to refine the results as the bounding box may not provide sufficiently tight capture for some object shapes.

An important class of seeded segmentation algorithm are those performing a graph-cut optimization, after Boykov and Jolly [22] who address object segmentation in images via max-flow/min-cut energy minimization. Typical energy functionals balance the probability of pixels belonging to the foreground/background with spatial coherence constraints expressed via edge contrast. The user-specified scribbles serve as hard constraints and also provide statistical information. This region-edge combination is very effective in improving segmentations based on edge or region alone. However, there is an inherent shrinking bias of graph cut towards shorter paths, i.e. small segments as the optimization sums over the boundaries of segmented regions. By contrast, level set based methods include a length-based "ballooning" term which encourages a larger object segment. Most graph-cut based approaches [109] for otherwise avoiding the shrinking bias in graph-cut and similar approaches involve variations on normalizing the cost of the cut by the size of the resulting object(s). Alternatively, a subspace of solutions may be explored by varying the relative weighting of the boundary and region terms [110]. Beyond [178], Rother *et al.* [179] make modifications to GrabCut to deal with large images and semi-transparency for practical applications.

Many methods [11, 171, 78] driven by scribbles selectively fill the desired region by expanding from the interior of the selected object outwards and do not explicitly consider the object boundary. This makes them advantageous for segmenting objects with complex topologies, whilst they may suffer from a bias that favors shorter paths from the seeds. Another drawback of these methods is that they may fail to accurately identify the real object boundaries due to the lack of an explicit presentation of edge contrast. Grady [74] proposes to use random walks for soft image segmentation with explicit edge weights, where each pixel is assigned the label with maximal probability that a random walker reaches it when starting from the corresponding scribbles. This method may be mathematically considered as a relaxation of the binary values of the potential function in graph cuts to avoid the shrinking bias or "small cut".

Efforts to combine the complementary strengths and weaknesses of seed-expansion and graph-cut approaches have been made. Sinop and Grady [196] show that graph-cuts, random-walkers [74] and a method similar in principle to geodesic segmentation [11] can be placed in a common framework. Price *et al.* [169] combines geodesic-distance region information with explicit edge information in a graph-cut optimization framework, which has the ability of seed-expansion approaches to fill contiguous, coherent regions without regard to boundary length with the ability of edge-based segmentation to accurately localize boundaries.

Our work is most closely related to prior image segmentation approaches using level set methods [230], which neatly enable the minimization of energy functionals such as those proposed by Mumford-Shah [152] or Zhu-Yuille [233]. One application of level set methods to image segmentation has been the edge-based active contour model [31, 105], which depends on image gradient and therefore is a rather local approach sensitive to noise. More robust approaches that encode region information have been proposed later in [159, 32]. Chan and Vese [32] propose the approach of active contours without edges by assuming the intensity distributions in different partitions to be Gaussian distributions with different variances. Heiler *et al.* [87] adopt Laplace distributions for natural image segmentation. Non-parametric methods have also been proposed to model the intensity distributions [88, 106, 28]. These Non-parametric methods all assume that pixels belonging to one region all share the same probability distribution and thus can not handle the inhomogeneity of the sought regions. Recent works [27, 123, 129] have been proposed to incorporate local intensity statistics instead of modeling the intensity distribution globally for each region. Higher level prior knowledge such as geometric shape priors has been introduced to level set framework [180, 26, 49].

One of the appealing advantages of level set methods is that they can neatly enable flexible forms of energy functionals. However, they are prone to getting stuck in a local minimum frequently caused by the sensitive edge-based term. Early approaches to edge detection aim at identifying the presence of a boundary through local measurements, such as Sobel operator [59] and Canny detector [30]. Recent local approaches incorporate color and texture information, either taking advantage of learning techniques for cue combination [145, 58] or observing the local distribution of quantized color class labels without estimating a specific model for a texture region [56]. In Ch. 4 we propose a robust and efficient algorithm for segmenting image and video sequences with minimal user interaction, adopting the seed-expansion approach driven by a level set framework.

2.3.2 Video Segmentation

Video segmentation has received considerable attention in recent years, with the majority of research effort categorized into automatic methods following two fundamental strategies; spatio-temporal ($3D$) analysis and frame-to-frame segmentation ($2D + t$). Recently, interactive approaches to video segmentation have also been investigated which is briefly reviewed in this section.

Spatio-temporal Approaches ($3D$) to Video Segmentation

Methods in the first category tackle video segmentation as a spatio-temporal (x,y,t) clustering problem. For example, Dementhon [53] proposes a spatio-temporal approach in which hierarchical mean shift clustering is applied to pixels of 3D space-time video stack, which are mapped to 7-dimensional feature points, i.e., three colour components and 4 motion components derived from inter-frame flow estimates. However, the differences in the spatial and temporal resolution of video and the isotropy of mean shift kernels can result in spurious regions manifesting local movement in the footage.

Anisotropic [212] and causal spatio-temporal kernels [161] have also been explored to refine mean-shift approaches to space-time segmentation. We compare our video segmentation algorithms proposed in Ch. 5 and 6 against [161] as a state-of-the-art benchmark. Shi and Malik [187] propose a pairwise graph based model to describe the spatio-temporal relations in the 3D video data and have employed the spectral clustering analysis to solve the video segmentation problem. Ristivojevic and Konrad [175] derive active surfaces through the space-time volume, which compete iteratively to delineate object boundaries. Greenspan *et al.* [75] present an approach to extracting coherent space-time regions in feature space via GMM unsupervised clustering. Grundmann *et al.* [76] present an efficient and scalable approach to spatio-temporal segmentation of long video sequences using a hierarchical graph-based algorithm, combining a volumetric over-segmentation with a hierarchical re-segmentation. However, these approaches usually become computationally infeasible for pixel counts in even moderate size videos, and often under-segment small or fast moving objects that form disconnected space-time volumes.

Frame-to-frame Approaches ($2D + t$) to Video Segmentation

The second category of approach segments 2D frames independently, and then creates associations between regions over time to identify and prune sporadic regions [151, 44] [25]. Moscheni *et al.* [151] process two consecutive frames at a time by iteratively merging over-segmented regions together based on their mutual spatio-temporal similarity. Collomosse *et al.* [44] create spatio-temporal volumes from video by associating 2D segmentations over time and fitting *stroke surfaces* to voxel objects. Brendel and Todorovic [25] adopt a region-tracking approach in which similar regions are transitively matched and clustered across the video and temporal coherence is forced by incorporating contour cues to allow splitting and merging of regions. These methods are inspired from the observation that pixels constituting a particular segment often belong to the same object or may share common appearance properties. Furthermore, it becomes much more efficient as inference only needs to be performed over a small number of segments rather than all the pixels. Although the stability is improved in these methods, lack of temporal information from adjacent frames during over-segmentation may cause jitter across frames and the temporal coherence is not ensured; the poor repeatability of 2D segmentation algorithms between similar frames, causing variations in the shape and photometric properties of regions.

Falling in the second category, this thesis proposes two video segmentation algorithms to apply multi-label graph cut on successive frames, in which the segmentation of each frame is driven by motion flow propagated labeling priors and incrementally updated data model estimated from the past frames to improve the temporal coherence. The flow-propagated labels in the first algorithm are assumed to be hard constraints i.e. perfect estimates (Ch. 5). The second algorithm follows a flow-propagation strategy, but adopts 'soft' constraints on motion propagated priors (Ch. 6).

In addition to motion propagation, the algorithm of Ch. 6 utilizes conceptually higher level soft constraints defined via multiple unsupervised over-segmentations of the video frame. This approach has also been widely adopted for image segmentation [84, 172, 6] using a single over-segmentation. In contrast to these that use multiple super-pixels as a hard constraint (i.e. assuming that all pixels constituting a particular region belong to the same label), more recent work integrates a higher-order region consistency potential with conventional unary and pairwise constraints by using CRFs in a soft framework [108, 107]). We adapt the latter approach in our video framework, but differentiate ourselves in several ways. First, we adopt over-segmented super-pixels from multiple unsupervised segmentation algorithms rather than a single segmentation algorithm — after [96, 181, 125] but using the soft framework of [108, 107]. Second, rather than computing a penalty via the number of pixels in the super-pixel not taking the dominant label, our method considers the region consistency potential as an even *softer* constraint which is similar to the data prior present in pairwise CRFs [22, 163], and thus can be solved efficiently. Third, to the best of our knowledge, we are the first to apply higher-order spatial constraints to address the video segmentation problem. This is interesting because the temporal incoherence of the per-frame segmentations is nevertheless shown to improve the spatial and temporal coherence of our video segmentation.

Interactive Approaches to Video Segmentation

Interactive video object segmentation systems have also been proposed in recent years. Various directions have been investigated such as tracking region boundaries over time [2, 167], extending 2D segmentation to 3D video volumes [132, 211, 10, 11], and applying graph cut segmentation on successive frames driven by motion flow [13, 12]. Agarwala *et al.* [2] propose a rotoscoping system combining computer-vision-based tracking with user interaction, by solving a spacetime optimisation problem for time-varying curve shapes. Price *et al.* [167] present a method for interactively segmenting video sequences by propagating multiple cues from one frame to another, which are automatically weighted according to their predicted importance on the specific video sequence being segmented, and are further weighted based on learning from user corrections. Li *et al.* [132] apply a 3D graph cut based segmentation approach on the spatialtemporal video volume. Wang *et al.* [211] extend the mean shift algorithm in [45] (see Sec. 2.3.1) to 3D to address the spatio-temporal video segmentation. Armstrong *et al.* [10] treat interactive video cutout as a global optimization problem, segmenting and rendering complex surfaces from 3D image volumes at interactive (sub-second) rates using a cascading graph cut (CGC). Bai and Sapiro [11] propose an interactive framework for soft segmentation and matting of natural images and videos, based on the optimal, linear time, computation of weighted geodesic distances to the user-provided scribbles, from which the whole data is automatically segmented. Bai *et al.* [13] use a set of propagated local classifiers along the boundary to drive the local graph-cut segmentation. This method is improved in Bai *et al.* [12] by introducing a new color model, which incorporates motion estimation into color modeling in a probabilistic framework, and adaptively changes model parameters to match the local properties of the motion.

2.4 Hierarchical Structuring of Media Collections

Hierarchical structuring of media collection is common to many contemporary approaches for interactively navigating collections. For example, Krishnamachari *et al.* form tree structures from an image collection, imposing a coarse to fine representation of image content within clusters and enabling the users to navigate up and down the tree levels via representative images from each cluster. This approach was later adopted in [35, 70]. In [35], a fast search algorithm and a fast-sparse clustering method are proposed for building hierarchical tree structures from large image collections. Goldberger *et al.* [70] combine discrete and continuous image models with information-theoretic based criteria for unsupervised hierarchical clustering. Images are clustered such that the mutual information between the clusters and the image content is maximally preserved. Our approach to structuring collections combines a hierarchical clustering similar to [114, 35, 70] with graph optimization approach [111] to navigate and visualize large media collections. The resulting system differs from existing hierarchical clustering approaches in several ways.

We introduce a novel approach to the hierarchical structuring of media collections in Ch. 5. Rather than exploiting low level visual features in the clustering, we incorporate both high-level semantic similarity when constructing top levels of the tree, and global image feature descriptors via a Bag of visual words (BoW) framework [197] for constructing lower levels of the tree. Consequently, this tree structure not only enables a global semantic summary of the collection, but also encodes visual similarities at various levels. Furthermore, in our system each node in the hierarchy encodes a directed graph that encapsulates connections among the digital items assigned to that node, rather than an unstructured subset of media as typified by previous work.

2.5 Artistic Rendering of Images and Video

The field of non-photorealistic rendering (NPR) has expanded into a vibrant area of research covering a wide range of expressive rendering styles for the visual communication: exploded diagrams [131], false color [166, 173], and artistic styles such as painterly rendering [20, 228]. This section presents a comprehensive review of the latter category of *artistic rendering* (AR); specifically techniques focusing on artistic stylisation of two-dimensional visual content, i.e. image and video.

The vast majority of early AR techniques addressed the digital simulation of physical materials used by artists, from simple simulations of hairy brushes [200] to full multi-layered models of pigment diffusion and bi-directional transfer between brush and canvas [50]. In this thesis we focus only on the problem of brush stroke placement and anisotropic filtering that conveys the impression of stroke placement.

2.5.1 Stroke-based Renderings

A significant number of AR algorithms operate by placing strokes on a virtual canvas, where they are composited to form a rendering. These approaches are referred to as stroke-based rendering (SBR). Within SBR, we partition algorithms into semi-automatic, i.e. user-assisted, and automatic processes. The former typically pre-dates the latter, pointing to a trend toward automation.

Semi-automatic Approaches

Haeberli [79] proposed a semi-automatic system which permits a user to rapidly generate impressionist style paintings by creating brush strokes. The system automates the selection of stroke color, and for non-circular brush strokes can also decide stroke orientation by painting strokes orthogonal to the intensity gradient in the source image. The system relies upon the user to determine the order and scale of strokes. The size and sequencing of stroke overpainting is crucial to producing results with an acceptable aesthetic and without the loss of salient detail. Haeberli formalises the concept of a painting as an ordered list of brush strokes - each with associated attributes: location, size, colour, orientation, shape. Virtually all modern AR techniques make use of this paradigm in their generation and representation of paintings. For example, Curtis et al. [50] use a similar approach to create water colour effects.

Later, semi-automatic systems adopt an image segmentation approach to painting [183, 14]. Both algorithms operate by segmenting the source image at various scales (coarse to fine), to form a similar hierarchical representation of regions in the image to that of a low-pass pyramid. An image region in the output may then be rendered at scales, proportional to the depth to which the scale-space hierarchy is traversed. This tree depth is specified interactively by the user. Santella and DeCarlo [183] use gaze trackers to directly harness the perceptual measures inherent in the human visual system to control the scale, whilst Bangham et al. [14] place a cross shaped mask at a user specified location in the image and the hierarchy depth is proportional to distance from this mask.

Fully Automatic Approach

The earliest fully automatic AR algorithm is described by Haggerty [80], which attempts to automate Haeblerli's pipeline using pseudo-random stroke size and painting order. However this method often over-paints salient features with nearby large strokes generated in non-salient regions.

Litwinowicz [136] proposes the first automatic painting algorithm which places rectangular brush strokes at regular intervals on the canvas while retaining a pseudo-random fashion. Strokes are oriented using Sobel gradients after [79, 80]. Strokes crossing strong edges in the source image are clipped to preserve the edge details. This algorithm applies thin-plate splines to interpolate stroke orientation within flat, near textureless regions. Hays and Essa [82] adopted similar approaches for interpolation in their video painting algorithm. Intensity variance [205] and chromatic variance [189] have also been used to drive the stroke placement.

Hertzmann [89] (shown in Fig. 2.3) was the first to use curved brush strokes of multiple sizes rather than constant-sized rectangular strokes to increase the aesthetic of painting. The algorithms starts by generating a Gaussian pyramid of the source image corresponding to a series of layers in the painting. Starting with a rough sketch drawn with a large brush, the canvas is painted over with progressively smaller brushes, but only in areas where the sketch differs from the blurred source image. Thus, visual emphasis in the painting corresponds roughly to the spatial energy present in the source image. The algorithm can produce painting with long, curved brush strokes, aligned to normals of image gradients.

There has been a tremendous effort to develop AR algorithms [39, 41, 191] using fully automated measure of salience beyond the semi-automatic approach proposed by Decarlo and Santella [183]. Collomosse and Hall [39] were the first to adopt such an approach, using statistical analysis to determine the importance or salience of pixels within the original image. Stroke attributes are derived from salience and gradient information in the image, producing an aesthetically pleasing painting whilst mitigating against loss of detail. Collomosse and Hall [41] further employed a genetic algorithm (GA) to search the space of possible paintings for a given image, so approaching an 'optimal' artwork

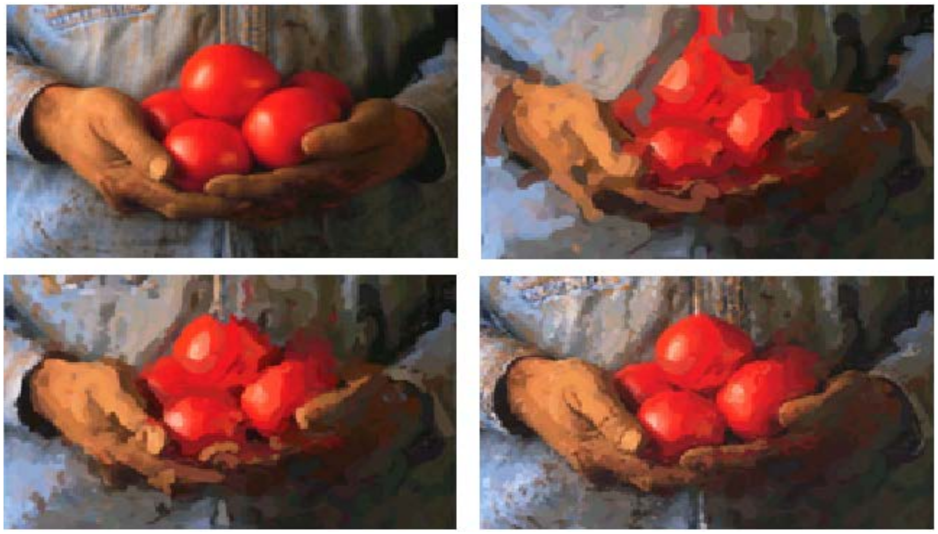


Figure 2.3: Incremental painting in Hertzmann's coarse to fine approach, images reproduced from [89]. Observe that large coarse strokes often remain visible in flatter areas, e.g. non-salient texture on the shirt, whereas fine strokes appear around edge detail, e.g. salient detail on the hands.

in which salient detail is conserved and non-salient detail is attenuated. Shugrina et al. [191] described an interactive system, in which the stroke placement is influenced by even higher-level contextual parameters, i.e. emotion.

2.5.2 Region-based Techniques

Santella and DeCarlo [183] were among the first to apply image segmentation in AR to form a hierarchical representation of an image using a variant of mean-shift [46]. Driven by eye-tracking data this approach can perform highly abstract renderings by descending the hierarchy.

Gooch et al. [72] propose an image-space painterly technique using curved brush strokes. The algorithm initially segments the input image with flood filling to compute brush stroke paths. Curved strokes are then fitted to the medial axis of each homogeneous region. However, we observe that the formation of intensity homogeneous regions is sensitive to image noise or textures, and so causes the system to tend toward photorealism in most real images.

Collomosse and Hall [40] use higher-level salient features identified within a set of two-dimensional images as compositional elements to produce a Cubist style painting with minimal user interaction. Song et al. [198] classify regions into one of several canonical shapes and fit regions with those shapes to create a simplified shape rendering. Region deformation was also employed to warp regions into superquadric shapes reminiscent of Cubist renderings [40].

Zeng et al. [228] propose region-based painterly rendering system which classifies texture within regions to drive the type of stroke used. Using the same brush stroke model from [228], Zhao and Zhu [231] propose a region-based painterly rendering system, which decomposes an image into a hierarchy of its constituent regions and augments painterly rendering with perceptual ambiguity computation and control for the simulation of abstract paintings.

2.5.3 Example-based Techniques

In contrast to heuristic approaches, example-based rendering (EBR) algorithms harness machine learning to model the stylisation process — typically by densely sampling corresponding patches in a source and stylised training image pair (A and A' respectively). Stylization of a target image B proceeds by matching patches on an approximate nearest neighbor (ANN) basis with those sampled from A during training [93]. The corresponding patch from A' is then composited into stylised version of B, to create mapping $B \mapsto B'$ said to be *analogous* to $A \mapsto A'$.

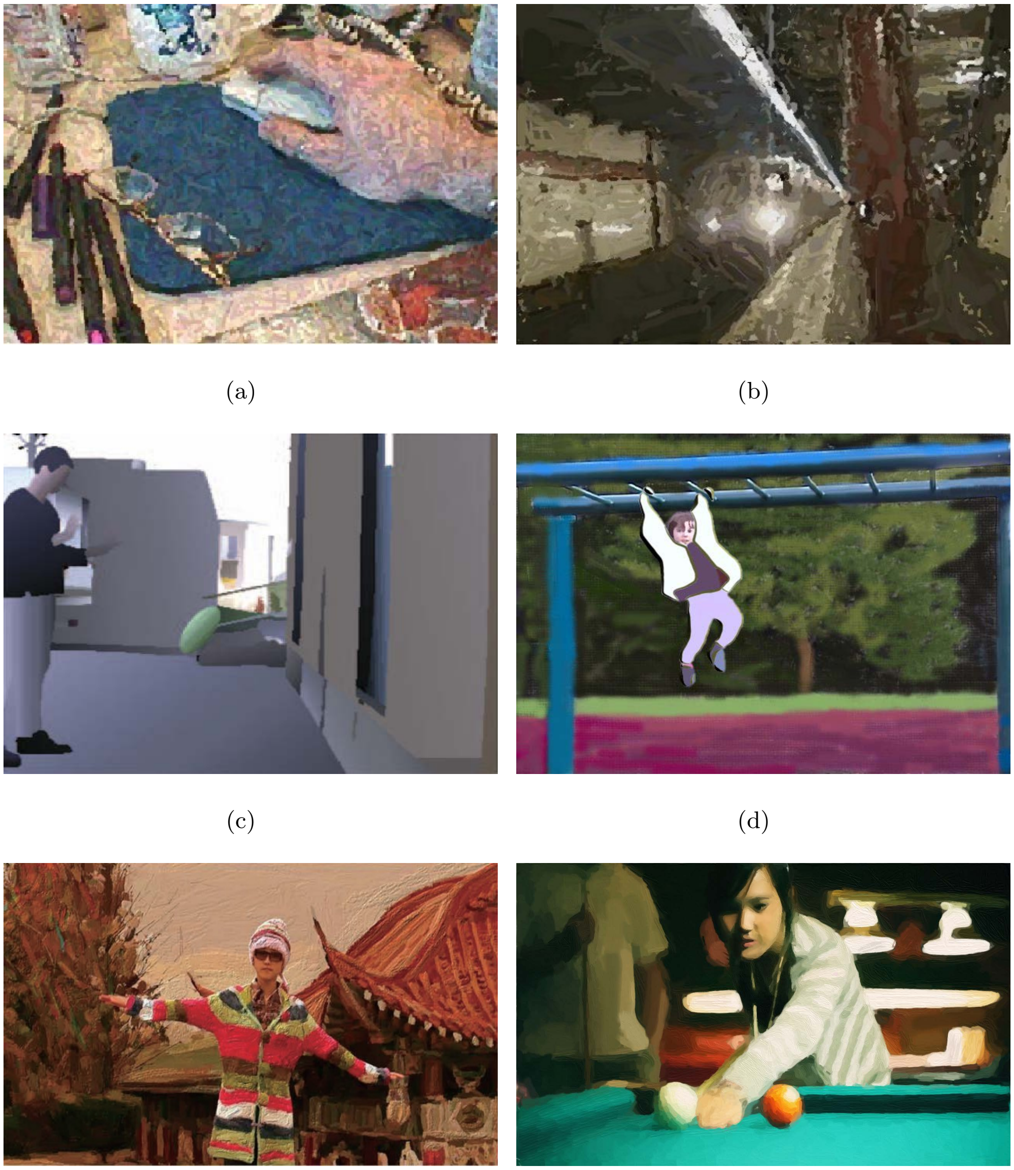
An extension of image analogies incorporates edge orientation to influence patch choice [124], proposing a texture transfer algorithm which modifies the target image replacing the high frequency information with the example source image. To ensure that the texture directions conform to the target image, they propose to evaluate of the neighborhood similarity that takes the gradient direction into account. Freeman et al. [62] propose an example-based method for translating line drawings into different styles by fitting each line as a linear combination of similar lines in a training set, and interpolate between the corresponding training examples in the output style. Kalogerakis et al. [102] propose to generate predictive models for synthesizing detailed line illustrations from examples.

2.5.4 Image Processing and Filtering

Image processing filters have recently shown their value in AR. Winnemöller et al. [226] propose a method to abstract image using a bilateral filter and *difference of Gaussians* (DoG) filter, attenuating detail in low-contrast regions while preserving sharp edges. It applies smooth quantisation of the luminance channel in CIELab space to achieve a strong cartoon-like effect. Rosin and Lai [177] use a combination of refined lines and blocks, as well as a small number of tones, to produce AR with sufficient elements from the original image. Kyprianidis et al. [115, 120] propose the *anisotropic Kuwahara filter* to abstract image resulting in sharper edges and the enhancement of anisotropic image features such as hair or fur. Kang and Lee [103] were the first to apply shock filtering for AR. Kyprianidis and Kang [119] present an approach based on adaptive line integral convolution in combination with directional shock filtering. The smoothing process regularizes directional image features while the shock filter provides a sharpening effect, both of which are guided by a flow field derived from the structure tensor. In Ch. 7 we compute the orientation field of non-facial area in portrait rendering based on the eigenvalues of the structure tensor inspired by [117, 119].

2.5.5 Video Stylisation

Video stylisation was first addressed by Litwinowicz [136], who produces painterly video by pushing brush strokes from frame to frame in the direction of optical flow motion vectors. This approach was later extended by Hayes and Essa [83] who similarly move strokes but within independent motion layers. Complementary work by Hertzmann [94] use differences between consecutive frames of video, painting over areas of the new frame that differ significantly from the previous frame. While these methods can produce impressive painterly video, the errors in the estimated per-pixel motion field can quickly accumulate and propagate to subsequent frames, resulting in increasing temporal incoherence. This can lead to a distraction scintillation or “flicker” when strokes of the stylised output no longer match object motion [149].



(a) (b)

(c) (d)

(e) (f)

Figure 2.4: Key video stylisation techniques in the literature: (a) Litwinowicz [136] and (b) Hertzmann [94] produce painterly video by pushing brush strokes from frame to frame in the direction of optical flow motion vectors; (c) Collomosse *et al.* [44] and (d) Wang *et al.* [213] create spatio-temporal volumes to improve coherence; (e) Lin et al. [134] and, recently, (f) O'Donovan and Hertzmann [156] interactively segment video into layers, each of which is populated with strokes.

More recently, image segmentation techniques have been applied to yield *mid-level* models of scene structure [213, 38] that can be rendered in artistic styles (Figure 2.4). By extending the mean-shift based stylisation approach of [51] on images, Collomosse *et al.* [44] create spatio-temporal volumes from video by associating 2D segmentations over time and fitting *stroke surfaces* to voxel objects. Although this geometric smoothing improves stability, temporal coherence is not ensured because the region map for each frame is formed independently without knowledge of the adjacent frames. Furthermore, association is confounded by the poor repeatability of 2D segmentation algorithms between similar frames, causing variations in the shape and photometric properties of regions that require manual correction. Wang *et al.* [213] also transform video into spatio-temporal volumes by clustering space-time pixels using a mean-shift operator. However, this approach becomes computationally infeasible for pixel counts in even moderate size videos, and often under-segments small or fast moving objects that form disconnected volumes. This also requires manual correction and frequent grouping of space-time volumes.

Semi-automated video painting systems have recently been developed and can be considered to be advanced rotoscoping tools, permitting both high-level control over groupings of strokes whilst also allowing fine-grain modification of stroke detail. Lin et al. [134] and, recently, O'Donovan and Hertzmann [156] developed systems that enable video to be interactively segmented into layers, each of which is populated with strokes. As with Agarwala et al. [2], stroke positions deform with the supporting layer and may be dampened to reduce flicker. In the system of Kagaya et al. [101], the video is first segmented into spatio-temporal coherent regions. Users can assign style and orientation parameters as key frames to these regions to be then interpolated over space-time.

Image processing and filtering approach has also been applied to create painterly animation from video. Winnemoeller *et al.* [226] present a method to abstract video using a bilateral filter, attenuating detail in low-contrast regions while preserving sharp edges. Anisotropic filtering was also proposed in [115, 120] using the Kuwahara filter. Such approaches do not seek to parse a description of scene structure, making them useful for scenes that are difficult to segment, but limited to a characteristic soft-shaded artistic style.

2.5.6 Portrait Rendering

AR for portraiture distinguishes itself from general purpose AR algorithms in that the human visual system has a strong cognitive prior for portraits, and is particularly sensitive to distortion or loss of detail around facial features [146]. Yet such artifacts are frequently observed when applying general purpose AR algorithms to photographs of faces. High quality rendering of faces is important, as many usage scenarios for artistic stylisation focus upon movie post-production effects, or consumer media collections, which predominantly contain images of people.

Prior literature dedicated to the painterly rendering of portraits is sparse. DiPaola [57] attempts to map the knowledge domain of the human portrait painter. However, this preliminary work places emphasis on methodology rather than delivering a concrete rendering system. Zhao and Zhu [232] are arguably closest to the work we present in Ch. 7, presenting an "example-based" method to paint portraits. As with our algorithm, strokes are captured during training from a human artist. However to create a new image [232], the training strokes are simply warped from the training face to the new face, using a triangular mesh established over facial features. The system does not learn a model of the painting process, and so can not generalize beyond its training data to produce new paintings. This leads to noticeable repeatability when producing several portraits, as the same captured strokes are output each time. By adopting a warping, rather than rendering, strategy the approach can also distort strokes, and cannot adapt stroke geometry and tone to image content, e.g. to emphasize shadow or highlights as we do. A further limitation is the lack of any process for rendering hair.

Portrait rendering techniques for other artistic styles have also been investigated, ranging from face sketch synthesis [33, 224, 229], paper-cut [150], caricatures [203, 204, 73, 155], to cartoon face [34, 37]. Chen et al. [33] propose an example-based portrait sketching approach which decomposes the face into blocks of semantic components and also includes a sub-system for hair rendering of specific hair styles. Wang and Tang [224] synthesize face sketch from photo by dividing the face region into overlapping patches and using a MRF model to optimise the selection of sketch patches from a training set which contains photo-sketch pairs. Zhang [229] propose a MRF based algorithm for synthesizing a face sketch from a photo taken under a different lighting condition and in a different pose than the training set, taking advantage of shape priors and patch descriptors specific to facial components. Rendering artistic paper-cut of human portraits is investigated in [150], which uses precollected representative paper-cut templates to synthesize the final paper-cut image by matching them with the bottom-up proposals. Gooch *et al.* [73] propose to create black-and-white illustrations from photographs of human faces using thresholding strategy. Chen *et al.* [34] propose an example-based system to generate cartoon face from photo based on an inhomogeneous MRF model.

The key in portrait stylisation is to protect the facial structure from distortion which consequently preserves the identity of face. People generally prefer to keep their identity recognisable in the stylised portrait as this is part of the interest in personal media collections. Measuring the quality of artistic abstraction using the recognition of face identity has been investigated in [73, 226], which assessed the recognition time of familiar faces presented as abstract images and photographs. We present a trainable portrait rendering algorithm which enables the user to choose the level of abstraction for rendering without loss of details of facial features.

2.6 Observations and Summary

From the survey, we form some observations on trends and gaps in the literature.

The majority of previous algorithms for video editing/summarisation focus on forming an attention model from low-level features. However, the gap between the low-level feature and high-level editing operations is not well addressed. Previous work primarily focused on correlating low-level features with interests or salience within video sequences without explicitly defining editing operations. Yet a structured representation of editing is desirable to account for the low-level measure of interests, which would enable domestic users to effortlessly manipulate their home videos to enhance the aesthetic value and interests. Without extracting the representation of visual structures from video, the previous video temporal composition algorithms mainly formed the sequencing and transitioning based on similarity matching of low-level features which often causes mismatch upon sequencing and limits the expressive forms of transitioning.

Early image segmentation algorithms primarily adopted stochastic models or density estimation drawn upon raw pixel data. Later, the robustness of image segmentation was improved by combining multiple local cues into a global optimisation framework. Rather than focusing on local features and their consistencies in the image data, graph partitioning approaches follow global criterion to partition images into coherent regions while measuring the evidence of boundary, which produce segmentations that satisfy global constraints. The development of video segmentation follows the similar trend of image segmentation, but has been distinguished by the utilisation of motion information in similar optimisation formulas. The extension of the 2D image segmentation [45] to a space-time video cube may be performed by adding an additional time dimension to a feature space incorporating spatial and pixel feature. However, such 2D to 3D extensions usually become computationally infeasible due to the large volume of video data. Frame-to-frame $(2D + t)$ approach is appealing in terms of efficiency to segment long videos, however the lack of temporal information during segmentation may cause jitter across frames and the temporal coherence is not ensured; the poor repeatability of 2D segmentation algorithms between similar frames causes variations in the shape and photometric properties of regions.

We have presented a comprehensive survey of artistic rendering. We observe that the early algorithms focused on the SBR paradigm with increasing levels of automation and sophistication in stroke placement and driven by low-level image processing (typically the Sobel operator). As the early convergence of computer graphics and image processing developed, AR was advanced by more sophisticated image analysis offered by contemporary computer vision algorithms (e.g. segmentation, optical flow). AR systems incorporated even higher-level contextual parameters, i.e. human vision and emotion, to parse image or drive stylisation. Later the composition of higher-level features naturally fused visual feature and human semantics to produce more perceptual interpretation of images. A consequence of the increasingly sophisticated interpretation or structured representation of the image was a divergence from SBR to alternative forms of rendering primitives, such as the use of regions, which in turn unlocked greater diversity in the gamut of styles available to AR.

In parallel with the trend toward more sophisticated scene analysis, AR benefited from the emerging popularity of edge-preserving filtering in computer graphics. Edge-preserving filtering approach is limited to painterly, cartoon and sketchy styles due to the lack of high-level image interpretation. Yet, its capability of real-time processing on GPU hardware makes it a practical solution for video stylisation and sequence (e.g. water, smoke or fur) that is otherwise challenging to parse using segmentation algorithms.

Part II

Image and Video Manipulation

Chapter 3

An Evolutionary Approach to Automatic Video Editing

In this chapter we present an algorithm for automatic video editing; bridging the gap between low-level feature and high-level video editing operations. We develop a novel parse-tree representation for automatic video editing, and an optimisation algorithm to edit raw video footage into salient, aesthetically pleasing clips by identifying the optimal sequence of edit operations. Our salience-driven heuristic approach is driven by low-level measures derived from video content and define rules of video editing derived from common practice. The goal is to enhance the aesthetic value and succinctness of raw medium items within personal media collections.

3.1 Introduction

Amateur home videos often contain lurching pans as the camera operator switches subject, and subjects often suffer from poor framing. This can lead to videos that are not enjoyable to watch, despite the periods of interest within them. We present an algorithm to transform such videos into a bridged versions, through a sequence of video editing operations.

We are concerned with three types of *editing operation*:

- **Zoom** – frames are spatially cropped to focus attention.
- **Cut** – frames are removed to shorten the video (i.e. a temporal cropping is performed)
- **Pan** – the camera view-port moves to follow a subject.

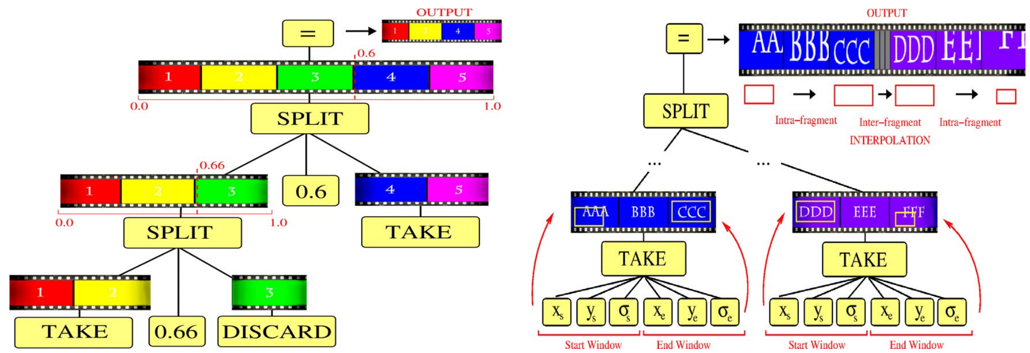
These operations may be applied to source video, with appropriate parameters and in a specific sequence, to produce an *edited video*. We interpret this sequence of operations as a *program*, and state finding the "best" program under some aesthetic criterion (Sec. 3.3) to be equivalent to finding an optimal edit sequence for a particular home video. We contribute both a novel representation for such programs, and a novel method for searching the space of programs using a Genetic Programming (GP) framework.

GP is an evolutionary optimization method [113]. Similar to the more common Genetic Algorithm (GA), GP creates a population of putative solutions (individuals) and "breeds" the best individuals together to produce successively improved generations of solutions [69]. With GP, however, the solutions are parse trees (programs) rather than points in a fixed-dimensional search space and thus GP works well in discrete spaces with discontinuities. GP is well suited to the problem of video editing, since the number and order of editing operations may vary greatly between video sequences. Furthermore, evolutionary algorithms such as GP are well suited to large search spaces in which the combination of distinct yet locally optimal solutions (e.g. partial video edits) are likely to yield globally preferable solutions. To the best of our knowledge, GP has not been previously applied to the automated editing of home videos.

Section 3.2 outlines our GP representation of an edit sequence. Our optimization process and aesthetic measure are described in Section 3.3. We present and discuss the results of applying our algorithm to representative home videos in Section 3.4, concluding in Section 3.5.

3.2 Representation of Video Edits

We represent an editing sequence as a program, specifically as a parse tree in which nodes act as operators that either manipulate or combine video fragments to form the



(a) cutting; the "split", "take" (detail omitted) and "discard" operators are used to create an edited video comprising frames 1,2,4,5.

(b) pan/zoom; the "take" operator specifies a start and end crop window for each video fragment. When fragments are concatenated, interpolation of window parameters is performed by "split".

Figure 3.1: The proposed parse-tree representation for video editing.

output clip. In this section we develop our tree representation.

3.2.1 Cutting

We begin by considering the basic cut operation, in which frames are removed from a video sequence in order to enhance its interest or aesthetic appeal. Under our tree representation, non-terminal nodes in the tree act as "*split*" operators that divide a video fragment into two sub-parts, passing the resulting fragments to their children. The point of division is governed by an operand on the node [0,1] representing the normalised length of the input video fragment. Thus *split* has three children; a child *constant node* specifying the real-valued division point, and two child operator nodes.

Video fragments may be divided recursively by further non-terminal *split* nodes. Terminal nodes may then either "*discard*" a fragment, or "*take*" it i.e. incorporate it in the output sequence.

The final edited video sequence is obtained via in-order traversal of the parse tree, appending video fragments as *take* nodes are encountered. We find linked lists of frames to be an appropriate data structure for managing fragments.

The *split*, *take* and *discard* operators form a basic editing system with cutting functionality. Fig. 3.1a provides an illustrative example of a terminal set comprising *split*, *take* and *discard* operators. It is easy to argue the sufficiency of this representation. Taking an unedited sequence of arbitrary length we can, by creating a tree comprising the right arrangement of *split* nodes, split the sequence into its individual constituent frames. We can then create any possible output sequence by applying *take* and *discard* operators.

3.2.2 Combined Panning and Zooming

In addition to cutting (temporal cropping) we enable a degree of freedom in the framing of video content through a spatial cropping mechanism. The effect of the cropping mechanism is to define a window around a portion of the frame, and then to scale that region to full frame size when outputting the edited video. When the window is appropriately positioned, this has the effect of "zooming" in on interesting content (e.g. a person) and so improving the framing of the scene.

We implement this operation by modifying the *take* terminal operator defined above. By specifying the cropping window as operand on the *take* node, we are able to specify the region of interest for cropping over each video fragment incorporated into the final edited video. Absence of cropping becomes a degenerate case; the crop window is simply positioned over the entire frame. To avoid visual artifacts we constrain the aspect ratio of the window to match the frame. The window's position is thus defined by operand $[x, y, \sigma]$; centre (x, y) and a uniform scale factor σ. Specifying the cropping window geometry in this manner also reduces our search space.

Although camera pans are technically achievable by splitting video into individual frames, and carefully specifying crop windows, this is not practically achievable by our GP optimization. Instead, we explicitly incorporate camera "panning" through an extension of the cropping mechanism. We extend the *take* operator again, to now have two operands: a crop window at the starting frame, and a crop window at the ending frame of the fragment. When outputting the final editing video, the window parameters are linearly interpolated between the start and end frames of each video fragment. Cropping thus becomes a degenerate case of panning, where the start and end cropping windows are identical. The *take* terminal node thus has six constant node operands $[x_s, y_s, \sigma_s, x_e, y_e, \sigma_e]$, where subscripts s and e indicate start and end frame respectively. As with the division point on the *split* non-terminal operator, these parameters are represented by normalised constant terminal nodes. Parameters (x, y) are normalised to frame width and height, while σ is normalised to range from half frame size (0) to full frame size (1). Figure 3.1b gives an illustrative example.

3.2.3 Concatenation of Video Fragments

Optimizations frequently result in parse trees that split video into many small fragments, with similar but slightly different cropping windows. This can result in a distracting flicker and instability in the final video. To mitigate against this, we perform some interpolation on window parameters when video fragments are concatenated by the *split* non-terminal operator.

Suppose two fragments F_1, F_2, of durations t_1, t_2, and with window parameters $\omega_1 = [x_s, y_s, \sigma_s, x_e, y_e, \sigma_e]$ and $\omega_2 = [u_s, v_s, \tau_s, u_e, v_e, \tau_e]$ are to be concatenated. A straightforward approach is to replace the end and start windows of F_1 and F_2 respectively with an interpolated window ω_I:

$$\omega_I = \frac{t_1}{t_1 + t_2}(\omega_2 - \omega_1) + \frac{t_2}{t_1 + t_2}\omega_2. \tag{3.1}$$

However, when a substantial *discard* has been made between fragments, it may be more appropriate to permit a discontinuity in the window geometry i.e. leaving ω_1 and ω_2 unmodified.

Our solution is to update the windows using a weight derived from the temporal distance d between the start and end of F_2 and F_1 respectively:

$$\begin{aligned} \omega_1 &\leftarrow \omega_1 + e^{-kd}(\omega_I - \omega_1) \\ \omega_2 &\leftarrow \omega_2 + e^{-kd}(\omega_I - \omega_2) \end{aligned} \tag{3.2}$$

where $k = 0.5$ provides interpolation over cuts up to $d \leq 10$ frames (i.e. $\sim \frac{1}{2}$ second duration).

3.3 Genetic Search for an Optimal Video Edit

We first describe the fitness function by which we measure the aesthetics of a video edit, and then provide the specifics of our GP optimization process.

3.3.1 Fitness of a Video Edit

Our fitness measure for a putative video edit seeks to estimate both the level of interest, and the aesthetics of the edited output video.

Our fitness function incorporates two terms for measuring interest; the total captured interest and the average interest captured over selected frames. The first term promotes completeness of interests selected from the raw video footage, while the second term promotes removal of "interest sparse" frames to produce feature rich video. The second term also encourages subjects of interest to be framed such that they occupy most of the scene. With respect to aesthetics, Arijon [8] notes that frequent short-term cuts within a sequence are unpleasant for the viewer. In some situations such cuts are appropriate, e.g. fast action shots, but these are too specific for general home video editing. Scene and camera motion should also be minimal at the points where shot boundaries are introduced. To incorporate these preferences, we introduce penalty terms for short cut sequences or cuts made in the presence of large-scale motion.

In line with these heuristics, we specify the following fitness function over all frames $\{E_1, E_2, ..., E_N\}$ included in the edited video:

$$\mathcal{F}(E) = \frac{P^{SC}}{N} \sum_{i=1}^{N} [CI(E_i) \cdot \left(w_1 + \frac{w_2}{N}\right) \cdot e^{-\gamma M(E_i)}] \tag{3.3}$$

where $CI(.)$ is a normalised operator evaluating the *captured interest* within a frame (subsection 3.3.1). $M(.)$ is a sum of the optical flow vector magnitudes within a frame (Figure 3.3, right). $SC(.)$ is a count of the number of short fragments within the edited sequence (below $\frac{1}{2}$ second), and constitutes a penalty term on short clips when $0 \leq P < 1$. The first term $\frac{P^{SC}}{N}$ penalises frequent cuts on the video which would lead to the rapid shot changes; it decreases significantly with the increasing number of cutting operations. The second term $CI(E_i) \cdot \left(w_1 + \frac{w_2}{N}\right)$ promotes both the completeness of

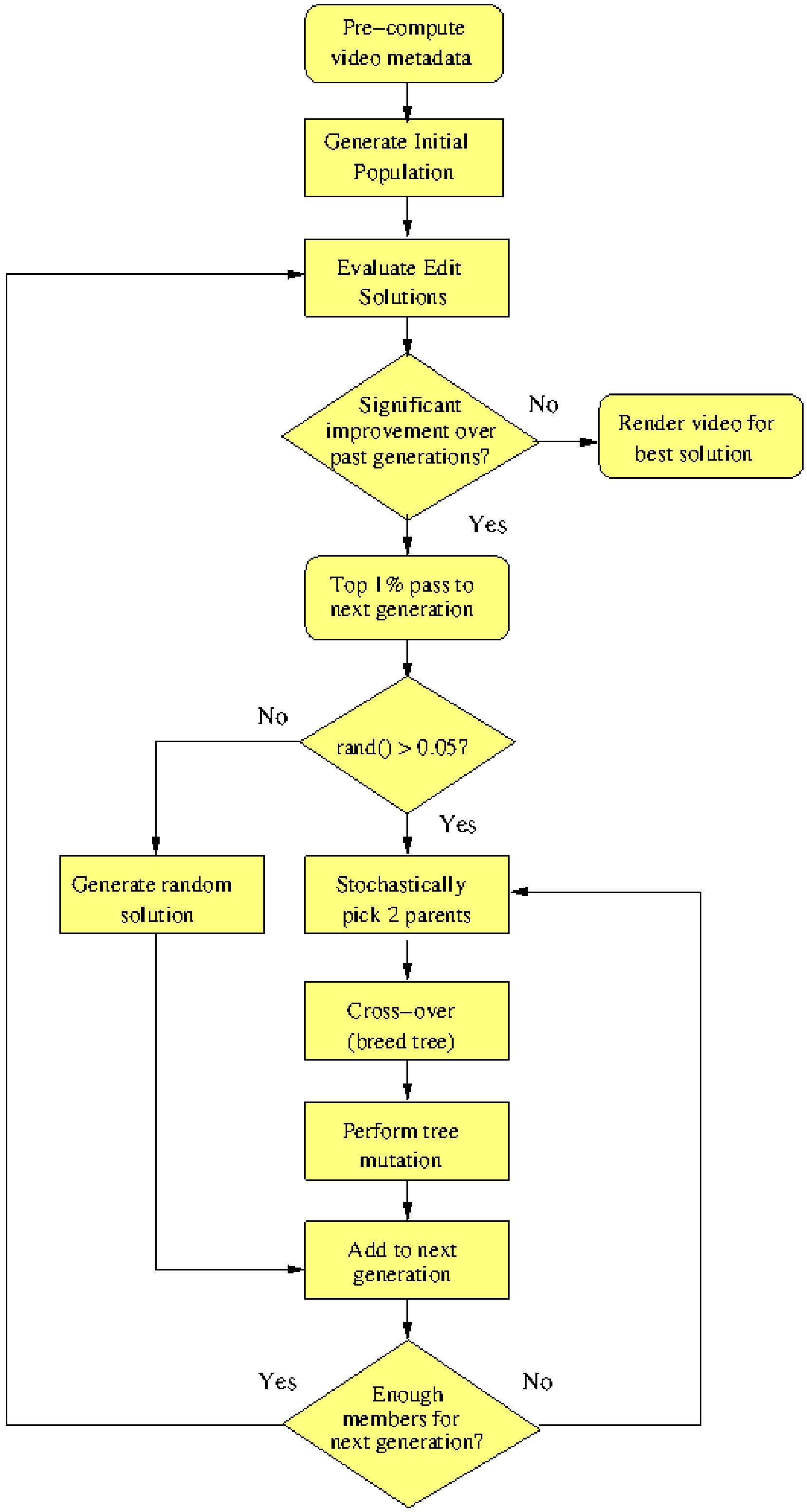

Pre–compute video metadata
Generate Initial Population
Evaluate Edit Solutions
Significant improvement over past generations?
No
Render video for best solution
Yes
Top 1% pass to next generation
rand() > 0.05?
No
Yes
Generate random solution
Stochastically pick 2 parents
Cross–over (breed tree)
Perform tree mutation
Add to next generation
Enough members for next generation?
Yes
No

Figure 3.2: Schematic of the GP optimization algorithm.

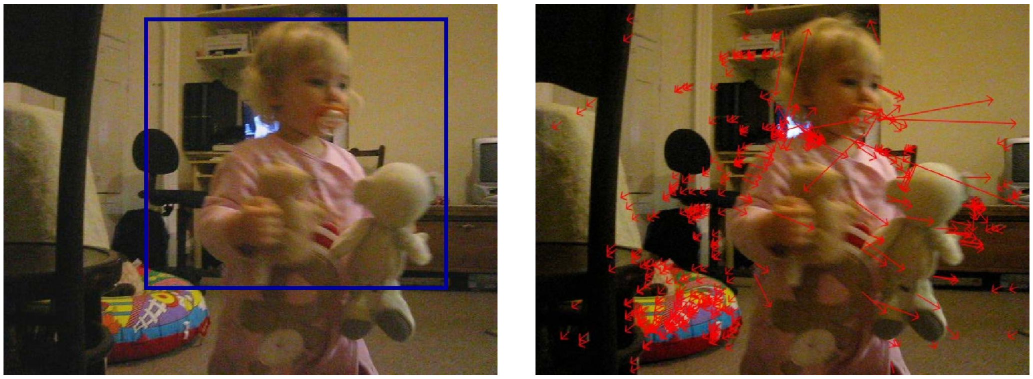


Figure 3.3: Video meta-data is extracted as a pre-process; we measure interest through detection of people (left), and inter-frame motion via optical flow (right). Result $V4$ (Sec. 3.4.2)

interests captured and the average density of interests; this linear combination form gives user the flexibility to adjust. The third term $e^{-\gamma M(E_i)}$ penalises shot changes in the presence of strong scene or camera motion. The pairs of parameters P, γ and w_1, w_2 are weights on the aesthetics and interest terms respectively, and may be adjusted to user preference. The latter weights are empirically selected to find the trade-off between the completeness and richness of captured interests. We give typical values with results in Section 3.4.

Captured Interest

Home video is predominantly used to capture life events, and people (e.g. friends and family) are frequently the objects of interest in such footage. In our system we correlate interest with the presence of people in a shot. Specifically, the greater the viewing area occupied by images of people, the more "interesting" and thus optimal our video is deemed to be. Person detection can be achieved in a number of ways, such as human face detection [209] and upper-body detection [61]. We opt for the latter, since face detection systems tend to perform poorly over the wide variations in pose, scale and lighting typical in home movies. Figure 3.3 (left) shows application of a popular cascade based person detector [61] to typical source footage. We obtain our value for $CI(.)$ by averaging the probabilities of pixels belonging to a person over the cropped window within the editing frame.

More sophisticated definitions of interest exist — for example considering temporal [154] and auditory [142] cues, or even models of linguistic semantics [148]. Although other normalised measures might be substituted, we find our measure suitable for the domain of home video. Our method also has the advantage that $CI(.)$ and $M(.)$ may be efficiently pre-computed by finding bounding regions for people in each frame of video, and intersecting those polygons with the cropping window to obtain the area of overlap during optimization. However we emphasise that our technical contribution is not in the interest measure *per se*, but rather in demonstrating the feasibility of a GP framework for identifying optimal video edits.

3.3.2 GP Optimization

Ideally a GP operator set should fulfill three criteria identified in [165]. First, any operator should return a value on any input, called evaluation safety. Second, the operator set should be sufficient; it should have enough expressive power to generate any possible solution to the problem. Third, the operators should be type consistent, i.e., return values of the same type so as they can be freely interchangeable in breeding.

Criteria one and two are satisfied (Section 3.2) however our constant terminal nodes return a different type ($\Re$) to that of the non-constant terminal and non-terminal nodes (video). This breaks the third condition of “type unity”. Koza *et al.* suggest use of a *constrained semantic structure* in such cases; effectively performing separate cross-over and mutation for constant and non-constant nodes [112]. We follow this strategy in subsection 3.3.2.

An overview of the optimization is shown as a flowchart in Fig. 3.2. We begin by randomly generating a large set of programs (or “individuals”, collectively referred to as the “population”). Each individual represents a putative solution, in the form of our edit tree representation (Section 3.2). GP is an iterative process, in which pairs of individuals are selected from each generation stochastically — with a bias to fitness — and combined via a breeding process of “cross-over” and “mutation” to create a population for the next generation. Thus at each iteration, the fitness of all individuals

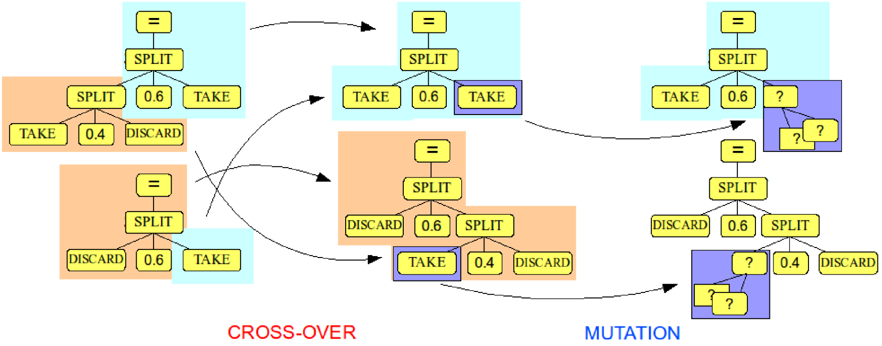


Figure 3.4: Illustrating the breeding process. GP crossover; parent trees are traversed depth-first. Corresponding nodes and their subtrees may be exchanged. Constant node operands are carried with their operators. GP mutation; non-constant nodes and their subtrees are replaced, with low probability. The value of constant nodes are subjected to mild Gaussian noise.

in the population must be evaluated using eq. (3.3) to enable fitness-proportionate selection. Optimization can be halted when maximum fitness within the population shows negligible improvement over several successive generations.

Initialization

Individuals within the first generation are initialised independently and randomly. In our experiments we use a generation size of 500. An individual's parse tree is constructed recursively by picking a node from the set of possible operators $\{take, discard, split\}$. Operators requiring constant operands will have appropriate child nodes created. In the case of a non-terminal operator being picked, further operators must be generated for the remaining child operands. The process recurses in a depth first manner until a terminal operator is generated. When choosing an operator for a non-constant node, the decision on type of node is made stochastically according to depth of recursion. Non-terminal nodes are less likely to be generated at deeper points on the tree. When generating a constant node, a value is picked uniformly at random, in range $[0, 1]$ as all operands are normalised by design (Section 3.2).

Elitism

At each iteration, the top $\sim 1\%$ fittest individuals pass through directly to the next generation. To maintain population diversity, $\sim 5\%$ of the next generation is reinitialised at random. The remainder of the next generation is bred from the current, using the processes of cross-over and mutation.

Cross-over

Cross-over is the mechanism by which elements of parent individuals are mixed to produce offsprings for the next generation. In GP this is achieved by constructing two new parse trees using portions of the parent parse trees.

Given two parents A and B we create two new individuals N_1 and N_2, initially by duplicating A and B. We then traverse N_1 in a depth-first manner, simultaneously traversing N_2 to create a one-one correspondence between nodes in N_1 and N_2. Where such a correspondence is possible (i.e. moves are possible from a parent node to a child node in both trees), we may swap the node and subtree below it in N_1 with the corresponding node and its subtree in N_2. The swap is made with probability 0.2 in our experiments. Figure 3.4 illustrates the process.

As our representation lacks type unity, evaluation problems will be encountered if constant nodes are substituted with non-constant nodes during swapping. Thus when a child node is swapped, its constant nodes are carried from the source to the destination tree in situ (as if logically part of the child node). Any non-constant operands are then recursively descended and swapped stochastically as before.

Mutation introduces diversity into the population, enabling exploration of the solution space. Again, due to the lack of type unity we must mutate constant and non-constant nodes using a separate mechanism. In the case of constant nodes, we iterate through nodes in N_1 and N_2 adding Gaussian noise to the real value assigned to each constant node encountered. The mean of the noise is the node's pre-mutation value, with a small standard deviation (0.5) in our experiments. In the case of non-constant nodes, we iterate through nodes in N_1 and N_2, and will generate an entirely new subtree for a node (using the method of subsection 3.3.2). Figure 3.4 illustrates this process. The probability of making such a mutation is 0.1 for all our experiments.

3.4 Results and Discussion

To evaluate the video editing system, we captured home videos covering a variety of events. Here we present the results of five videos ($V1 - V5$). In $V1, V2$ we disabled our zooming/panning mechanism to show the effects of the cutting operator alone. In $V3 - V5$ the full system is evaluated.

3.4.1 Cutting Only

Figure 3.5 depicts frames from our source videos, regularly sampled along a time-line running left-right. The presence of blue below the time-line indicates detection of interests (people), and red indicates portions of the source video time-line that were selected and concatenated to create the edited output.

The $V1$ and $V2$ source footage depicts family members at the park. In $V1$ the cameraman periodically becomes distracted and points the camera at the floor or at uninteresting objects. The system has identified contiguous blocks of interest in the video, and cut three sections of the source time-line for concatenation into the final edited video. Virtually all of the interest is captured in a minimal number of cuts. In $V2$ cuts have been made not only to maximise the density of interest in the clip, but also to prohibit rapid cutting in frames where detection of people is intermittent. This is frequently the case using [61] when people's backs are turned to the camera, or are of small scale. For these results, system parameters were set such that the ratio $w_1 : w_2$ was $1 : 10$, $P = 0.99$, and $\gamma = 10^{-5}$. Figures 3.6,3.7 show convergence with negligible change in population fitness or diversity after ~ 20 iterations.

3.4.2 Cutting, Zooming and Panning

For videos $V3 - V5$ we re-enabled the zooming/panning mechanism to run the system with full functionality. Figure 3.5 shows the cuts made in the source video to isolate

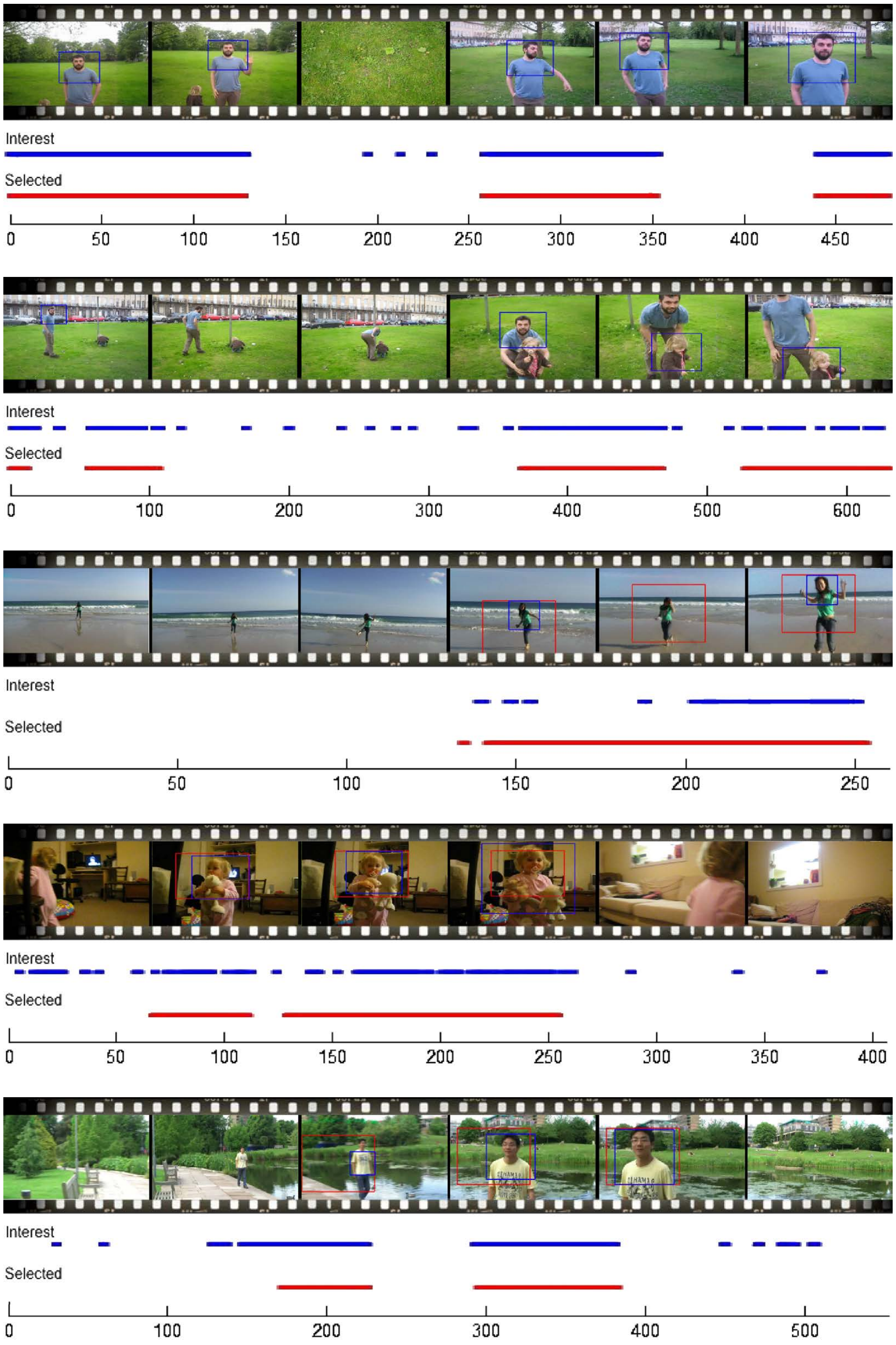


Figure 3.5: Evaluating our system over videos ($V1 - V5$), from top to bottom. For $V1$ and $V2$ we disabled zooming/panning. A time-line (in frames) runs from left to right, annotated in blue to show presence of interest, and in red to show segments of video selected for output by our editing process. Frames have been sampled from the source video at regular intervals; the blue box indicates the areas of interest detected. In the case of $V3 - V5$ the red box shows the cropping window.

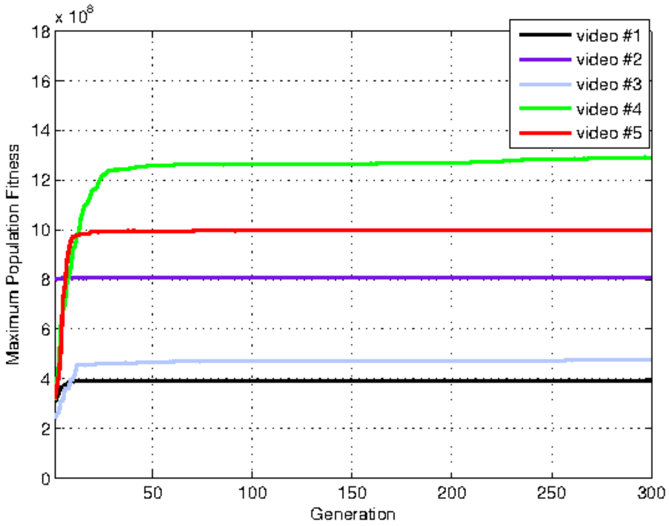


Figure 3.6: Optimization results for videos $V1 - V5$, plotting maximum fitness in each generation.

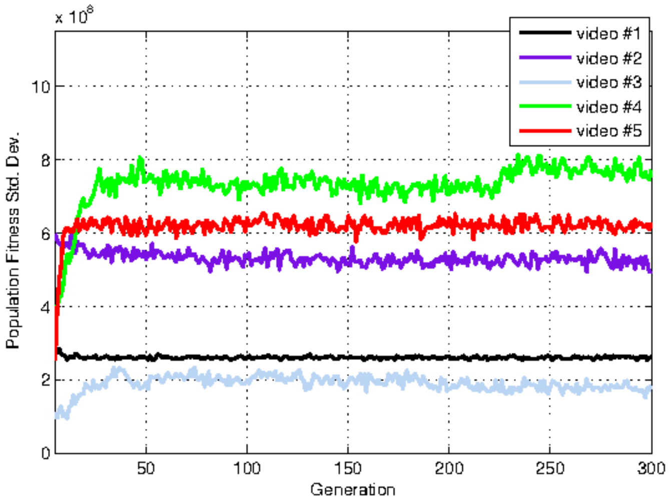


Figure 3.7: Optimization results for videos $V1 - V5$, plotting standard deviation (fitness diversity) for each generation.

"interesting" parts of the time-line. Again, source video frames exhibiting a negligible or intermittent response from the interest detector have been cut. Figure 3.5 also shows the position of the cropping window (red box) within frames. Footage within the window is scaled to create the rendered output footage shown in Figure 3.8. In the cases of $V3$ and $V4$, a crop window is created around the main subject which pans to follow the movement of the subject in the video. In the case of $V5$ a cropping window is also introduced to zoom in and improve framing of the subject; however since the camera is already panning to follow the subject, no additional panning is introduced. For these results, system parameters were set as in subsection 3.4.1, but with ratio $w_1 : w_2$ set to $1 : 100$. Figures 3.6, 3.7 again show quick convergence, with negligible change in population fitness or diversity after ~ 50 iterations. For our experiments we ran the optimizations up to 1500 generations (300 are shown).

3.5 Conclusion and Future Work

We have presented a novel tree representation for home video editing to bridge the gap between the low-level feature and high-level editing operations, suitable for use in a Genetic Programming (GP) optimization framework. Our representation incorporates cutting, zooming and panning operations. Uniquely, we search for a globally optimal video edit using GP, maximising both aesthetics and interest within the final clip. Our measures for aesthetics are grounded in common directing practice, and our measure for interest is based on the presence of people; the most common subject of interest for home videos.

We have demonstrated the efficacy of our approach over some representative examples of home video footage. Our system quickly converges to an acceptable edit sequence, requiring ~ 50 generations / minute of source video. To capture the subjectivity of video aesthetic, our fitness function is governed by user parameters weighting desire for objects of interest against frequency of cuts, and motion. The short optimization times enable user experimentation to taste.

Our algorithm has focused on GP optimization as a means for generating edit decisions. It has not explored the visual rendering of those edits. Transition effects might be

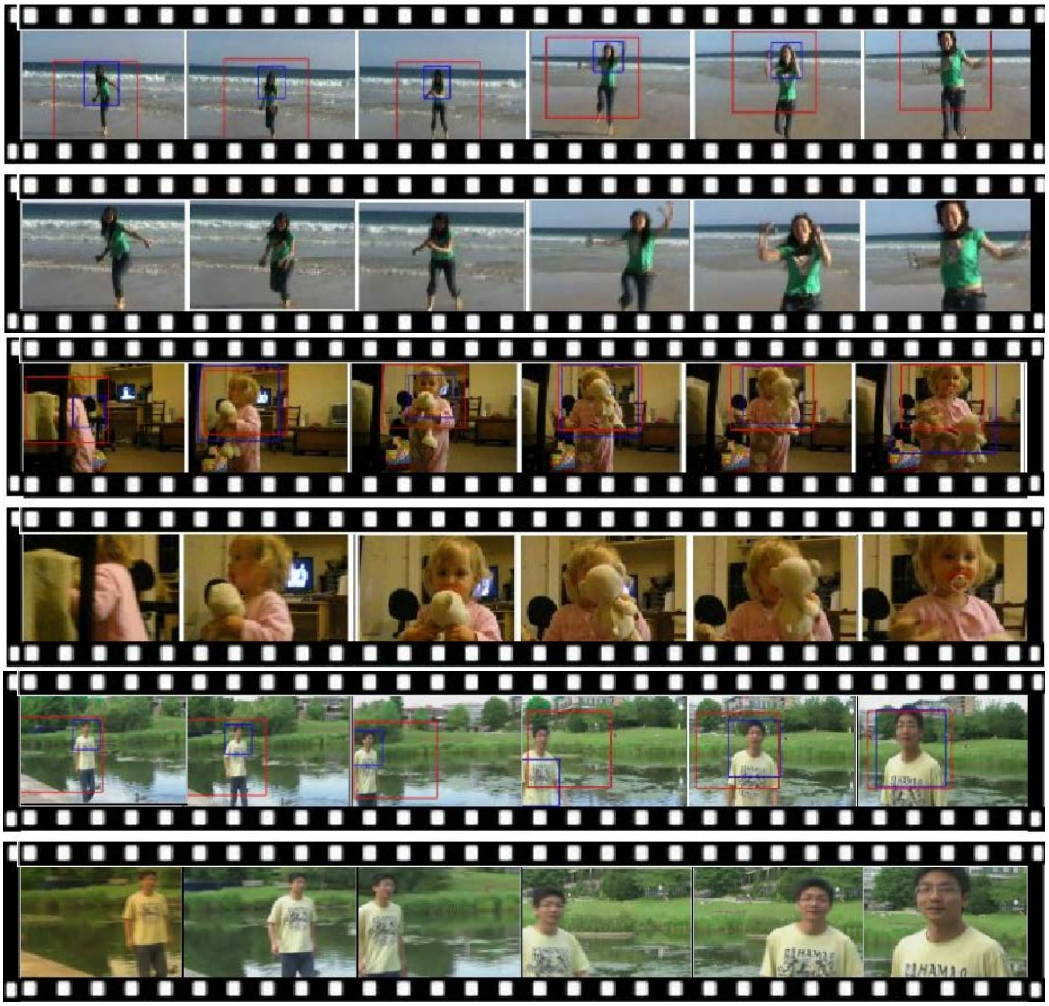


Figure 3.8: Final edited clip results from footage $V3 - V5$. Upper strip: Blue box indicates interest detection, red box indicates cropping window. Lower strip: Footage is rendered from within the red window to output the final clip.

introduced e.g. cross-fades when cutting. Future work may explore alternative cropping operators, for example seam-carving, to accommodate multiple disjoint regions of interest within a frame.

Although our fitness measure lacks the sophistication of [142, 148], we find it suitable for demonstrating value in our GP editing framework, and for the purposes of general home video editing. Extensions to this measure are a possible route for future work. A higher level temporal constraint (e.g. preferring alternating cuts between subjects during dialogue) might further enhance the aesthetic terms within fitness function.

However, within a subject domain as broad as home video, care should be taken to draw a sensible compromise between the complexity of editing heuristics and the generality of footage that may be processed.

Chapter 4

TouchCut: Fast Object Segmentation using Single-Touch Interaction

In this chapter we investigate a mid-level representation of visual media; region-based representation driven by image and video segmentation. We first propose a fast interactive object segmentation algorithm driven by minimum user intervention for image and video. We then describe a case study enabling users to selectively stylize video objects to create a hand-painted effect, using the parsed region representation. This algorithm supports our claim that stable representation of visual structure enables localised media manipulation and enhances the temporal coherence of expressive styles of visual media.

4.1 Introduction

The segmentation of objects from cluttered natural images remains a fundamental and inherently challenging Computer Vision problem. The task is generally regarded as under-constrained since, in the absence of high level scene understanding, there can be more than one interpretation of pixels comprising the desired object of interest or 'foreground' object. The past decade has seen a trend toward better constraining the segmentation task through: i) the development and increased reliance on global optimization methods; and ii) the combination of high-level prior scene understanding via *user interaction* with low-level cues such as colour and edges observed in the image.

A key challenge of interactive segmentation is to maximize the use of user provided prior information whilst minimizing user intervention. Although recent years have delivered significant advances, a considerable amount of user intervention is still required to achieve a satisfactory segmentation. Typically this involves the user indicating positive and negative class examples of pixels or regions. Often such indications requires correction either automatically to boost discrimination between by the positive or negative classes [170], or by the user iteratively working with the system to supply additional constraints [147]. Regardless of the interaction modality, the goal of any interactive image segmentation is to minimize the amount of effort to cut out a desired object while accurately selecting objects of interest.

To address this problem, we contribute an efficient algorithm for object segmentation driven by minimal user effort — a single touch. In contrast to previous interaction modes, ranging from roughly marking the desired boundary [68, 18, 210] to loosely drawing scribbles labeling the desired object and the background [22, 74, 11], to placing a bounding box around the desired object [178, 127], our system requires only a single (x, y) coordinate from the user offering an intuitive and maximally "economical" interaction method. Our method has clear applications on emerging touch-screen tablet, mobile and pervasive devices, the form factor of which devices may be inconvenient for fine-motor interaction. Further, in some use cases (e.g. deployment in high throughput, or multi-tasking situations such as driving) the high cognitive load required to trace outlines or regions may also be undesirable.

The core technical contribution of this thesis is a new model for object segmentation that fuses edge, region, and geometric cues within a *level set* [230] framework. In contrast to previous expanding contour approaches relying on intensity gradient, our proposed model incorporates a probabilistic estimate of edge location derived from a novel dominant colour extraction scheme. This scheme offers improved robustness when filling colour or texture coherent regions, leading to more accurate localization of the desired object's boundary. We also fuse this boundary information with a region-based maximum *a posteriori* (MAP) criterion designed to promote colour similarity with pixels local to the touched foreground point. The robustness of this region-based criterion is further enhanced by a consistency constraint enforcing uniformity of deviation from the foreground colour model, learned from the touched foreground region. This approach to colour consistency, combined with a novel per-pixel adaptive weighting scheme, mitigates the tendency for contour expansion to skip the real boundary when colour models of the foreground and background are indistinct. Finally, our proposed model also utilizes the geometric cue implied in single-touch input, that users typically touch an image in close vicinity to the geometric center of the desired object.

All together, our edge-region-geometry model provides a robust and flexible description of the interactive object segmentation problem, leveraging the flexibility of level set methods to promote accurate boundary placement and strong region connectivity while requiring minimum user interaction. Using an incrementally built foreground colour model, our framework also extends to address the temporally coherent video object segmentation problem, creating regions whose shape and neighborhood topology evolve smoothly over time whilst tracking the underlying video content. A motion estimation enabled shape prior is further introduced into the video adaptation to preserve temporal coherence when the foreground and background colour distributions are indistinct.

We briefly revisit level set methods in Sec. 4.2. We then describe the proposed framework for interactive object segmentation on still images (Sec. 4.3), explaining each of the energy terms comprising the proposed energy functional. We extend our system on still images to video sequences in Sec. 4.4, presenting an application of our proposed algorithm to tablet-based video manipulation. We present a comparative evaluation with previous work in Sec. 4.5 on both a qualitative and quantitative basis, concluding in Sec. 4.6.

4.2 Level Set Revisited

The basic idea of active contour models implemented via level set methods is that a contour $\mathcal{C}$ in a domain Ω can be represented by the zero level set of a higher level embedding function ϕ: $\Omega \rightarrow \Re$. Evolving the contour $\mathcal{C}$ is achieved by evolving the embedding function ϕ which is defined as the signed distance function with $\phi > 0$ inside the contour, $\phi < 0$ outside the contour and $|\nabla\phi| = 1$ almost everywhere.

The evolution of the level set function ϕ is governed by a partial differential equation (PDE). The PDE can be directly derived from a certain energy functional $E(\phi)$ on the space of level set functions. Subsequently one can derive the Euler-Lagrange equation which minimizes $E(\phi)$:

$$\frac{\partial\phi}{\partial t} = -\frac{\partial E(\phi)}{\partial\phi} \tag{4.1}$$

These methods are known as variational level set methods [230]. This formulation enables direct incorporation of statistical prior information into the design of $E(\phi)$ in the segmentation framework. Thus the segmentation boundary $\mathcal{C}$ is derived by obtaining the best ϕ at the zero level as

$$\mathcal{C} = \{x \in \Omega \mid \phi(x) = 0\} \tag{4.2}$$

4.3 Segmentation Framework of Still Images

Under the level set paradigm, we propose a new energy functional taking account of probabilistic edge map, colour distribution of foreground and background in an adaptive manner as well as the geometric cue implied by user touch:

$$E(\phi) = E_e(\phi) + E_a(\phi) + E_b(\phi) + E_u(\phi) + E_g(\phi) + E_d(\phi) \tag{4.3}$$

where $E_e(\phi)$ is the edge probability term, $E_a(\phi)$ is the ballooning term, $E_b(\phi)$ is the Bayes statistical error term based on colour distributions, $E_u(\phi)$ is the foreground consistency term, $E_g(\phi)$ is the geometric cue term, and $E_d(\phi)$ indicates the distance regularization term to ensure the stable evolution of the level set function by penalizing the deviation of the level set function from a signed distance function. These terms can be categorized as: edge based energy, statistical prior energy, geometry energy and distance regularization energy. Fig. 4.1 presents an overview of the proposed system, where the dashed lines indicate these four energy categories. Each individual energy term is detailed in the following subsections, and we also illustrate the importance

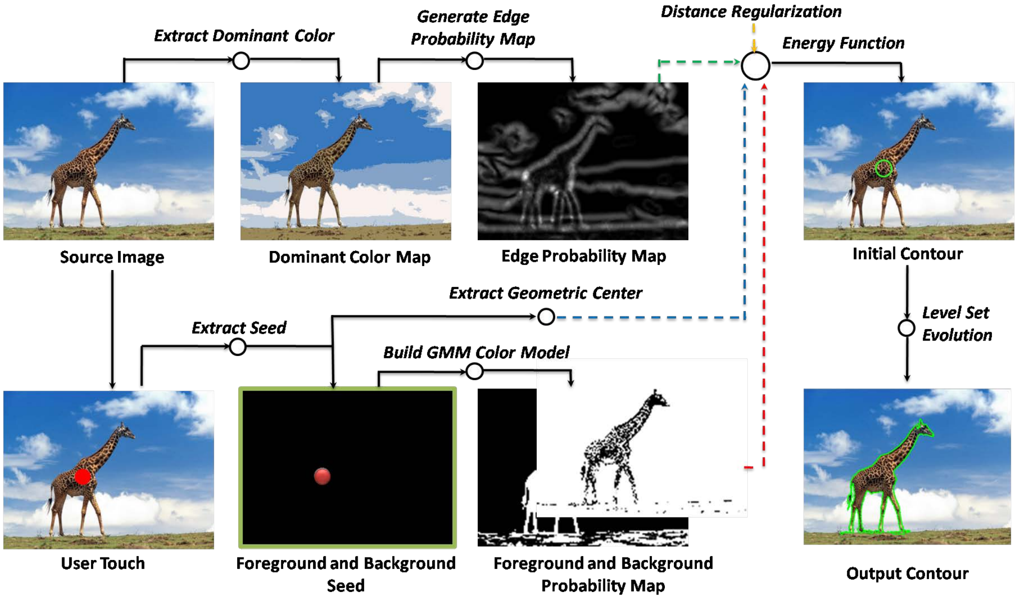


Figure 4.1: System overview. Dominant colour extraction is performed on the input image for calculating the edge probability map (first row). Foreground/background colour model is estimated based on user input and the image border respectively (second row). The energy function incorporates the various energy terms. The evolution of the embedding function ϕ is specified by the energy function (right column). The zero level contour converges to the object boundary to generate the segmentation (bottom right).

of each by disabling various terms to qualitatively demonstrate their contribution to segmentation performance.

4.3.1 Edge Based Energy

Classical snakes and active contour models [31] typically use an edge detector to halt the evolution of the curve on the boundary of the desired object. The gradient based edge detector inherently captures high frequency information but not necessarily the real boundary of the desired object. Moreover, it is also sensitive to noise. The edge-based active contour model is thus not applicable to most natural images especially texture rich or noisy data.

In order to describe the edge probability of colour-texture homogeneous region in natural images, we propose an approach inspired by JSEG [56], which calculates an edge indicator by observing the local distribution of colour class labels without estimating a specific model for a texture region. In our proposed method, the colour class labels are generated by extracting the dominant colour modes and assigning each pixel with the label of according colour mode.

Extracting Dominant Colours

A dominant colour (DC) is defined as a set of similar colours, of which the corresponding pixels occupy a relatively large proportion in (a specific region of) an image. There have been many approaches proposed to address the DC extraction problem. Splitting based colour quantisation approaches such as median cut [86] do not consider the colour distribution or the quantisation error. Lin and Zhang [135] extract coarse DCs by considering each local maximum and its neighborhood within a diameter-fixed sphere in the HSV colour space as a possible DC. Wang *et al.* [215] adopt EM algorithm to estimate the GMMs of the input colours. However, it is difficult to properly set the number of the Gaussians. The Generalized Lloyd Algorithm (GLA) is adopted to divide the input colours into clusters in [55, 56]. Because the GLA aims at minimizing the global quantization distortion, the colour ranges with high frequency are apt to be over-divided, and those with low frequency are apt to be under-divided. The Mean Shift algorithm is adopted to identify the dominant colours in [45]. However, it suffers from scale problem which makes it difficult to adaptively make a good trade-off between precision, robustness and roughness in the colour histogram. To address these problems, we propose a novel non-parametric DC extraction algorithm which considers both colour distribution and colour similarity, to better explore the inherent characteristics of DC.

The proposed algorithm [1] is performed on the histogram in a $64 \times 64 \times 64$ CIE Lab colour space. We choose CIE Lab colour space because it is designed closely matching human perception and is more perceptually uniform. The key procedure of this algorithm is shown in Fig. 4.2.

The first step finds all the local maximums in the histogram and assigns a unique label to each of them. A histogram bin x, is a local maximum if the following condition, where

[1] The code is from Sony China Research Laboratory

$H(\cdot)$ denotes the histogram value, and $N(\cdot)$ denotes the 6-connection neighborhood, is satisfied,

$$H(x) \geq H(y), \forall y \in N(x). \tag{4.4}$$

In case that neighboring peak bins have different labels, their labels are unified.

In the second step, the input colours are clustered via iteratively spreading the labels of all the peaks and regarding the bins with the same label as one cluster. The label spreading process is iteratively performed until every bin x with $H(x) > 0$ is labeled. In each iteration, bin x, which has not been labeled, may inherit the label of bin y, if $H(x) \leq H(y), y \in N(x)$. As the labels of different peaks are spread simultaneously, the label of the peak that is closer to the bin is likely to arrive first. This scheme seeks the shortest ascending route to a local maximum. If multiple labels arrive at x in the same iteration, then x is labeled as the same as the neighboring bin with the larger histogram value. Thus, this scheme seeks the shortest ascending route to a local maximum for each bin; it considers both color distribution and color similarity, whilst other local distribution based approaches only considers color similarity [135] or color distribution [45].

A bin x is defined as a joint $J_t(\cdot,\cdot)$ of two adjacent clusters Ω_i and Ω_j if the following is satisfied

$$x \in \Omega_i, \exists y \mid (y \in \Omega_j) \wedge [y \in N(x)] \wedge [H(y) \geq H(x)],$$

and thus $x \in J_t(\Omega_i, \Omega_j)$. The connection value $v_c(\cdot,\cdot)$ of these two adjacent clusters can be defined as

$$V_c(\Omega_i, \Omega_j) = \max[H(x) \mid x \in J_t(\Omega_i, \Omega_j)]$$

After the second step, all the colours are clustered. However, colour histograms, especially the high-resolution ones, are not smooth which normally leads to many local peaks. To make the algorithm robust to roughness of the histogram, some of the adjacent clusters should be properly merged. Considering the peaks as islets in a lake, some of small islets will be connected to form larger islets as the water level in the lake decreases. To this end, all the histogram values are first sorted in descending order. Then we scan the sorted values one by one to simulate the water level decreasing. We only consider merging the connected clusters during the scanning. The dominant mean colour $C_m(\cdot)$ of each cluster Ω_i is updated as the water level h decreases as

$$C_m(\Omega_i, h) = \frac{\sum x \cdot H(x)}{\sum H(x)} \mid (x \in \Omega_i) \wedge [H(x) \geq h]$$

where x indicates both the bin and the colour vector. Only the bins which are above the water level h contribute to the dominant mean colour of their cluster. When the water level h reaches the joint of two adjacent clusters, they can be merged if the following two conditions are satisfied

$$||C_m(\Omega_i, h) - C_m(\Omega_j, h)|| \leq T_d,$$

$$V_c(\Omega_i, \Omega_j) \geq T_p \cdot \min\{\max[H(x) \mid x \in \Omega_i], \max[H(y) \mid y \in \Omega_j]\},$$

where the threshold T_d indicates a colour difference that is distinctly visible to human eyes (suggested as 0.07) and $T_p \in [0.5, 0.75]$. The conditions constrain that two clusters can be merged only if their dominant mean colours are similar enough and their connection value is not too small compared to their peak values.

Compared with the agglomerative algorithm used in [55, 56], which only considers colour similarity, the proposed cluster merging scheme also considers colour distribution. When two clusters are merged, their connection relationships with other clusters will be inherited by the new cluster, so that the new cluster may be further merged with the adjacent clusters. As all the connection values are scanned, all the adjacent clusters will be considered for merging. Thus this step is finished when the iteration is over.

The average processing time on VGA (640 × 480) image is less than 60 ms (Intel Core2 CPU 2.1 GHz, single thread) which makes it a very efficient and robust algorithm for our application.

Edge Indicator and Energy

Suppose Z is the set of all N pixels in a dominant colour mode map. Let $z = (x, y), z \in Z$. Z is classified into C DC modes. The means of Z and class Z_i $(i \in C)$ in Z are

$$\begin{aligned} m &= \frac{1}{N} \sum_{z \in Z} z. \\ m_i &= \frac{1}{N_i} \sum_{z \in Z_i} z. \end{aligned}$$

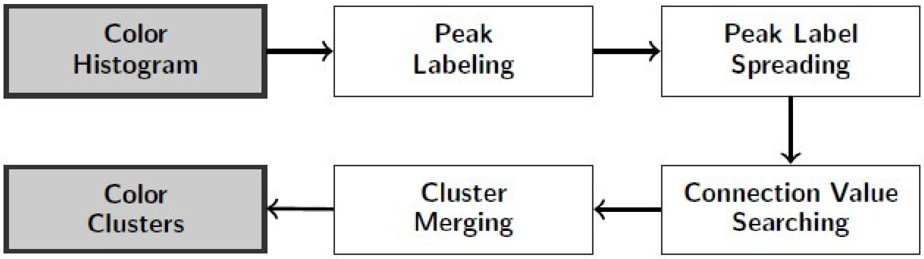


Figure 4.2: Key procedures of the proposed dominant colour extraction algorithm

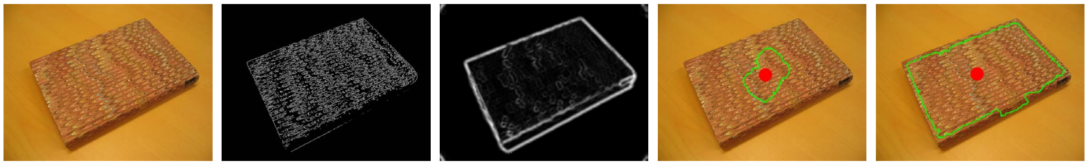


Figure 4.3: Comparisons of edge map by Sobel operator and the proposed edge probability: (1) Source image (2) Edge map by Sobel operator (3) Edge probability map by our approach (4) Segmentation with only the edge energy based on Sobel operator (5) Segmentation with only the edge energy based on proposed edge probability.

respectively. Let

$$\begin{aligned} S_T &= \sum_{z\in Z} ||z-m||^2 \\ S_W &= \sum_{i=1}^{C} S_i = \sum_{i=1}^{C} \sum_{z\in Z_i} ||z-m_i||^2 \end{aligned}$$

be the variance of pixels in Z and the total variance of pixels belonging to the same DC mode. The edge indicator is defined as

$$J = (S_T - S_W)/S_W.$$

The value of J is large near the boundaries of colour-texture homogeneous region and small in region interiors, and thus can serve as edge "probability" while suppressing high frequency information and noise as opposed to traditional edge detectors.

E_e incorporates the edge indicator J and is defined similarly as the geodesic model [31]

$$E_e = \omega_e \int_{\Omega} g\delta(\phi)|\nabla\phi|d\mathbf{x} \tag{4.5}$$

where $g = \frac{1}{1+cJ}$, ω_e is the coefficient which is specified in Table 4.1, c is a constant, H is the Heaviside function and δ is the Dirac delta function.

We define the ballooning term as

$$E_a = \omega_b \int_{\Omega} gH(\phi)d\mathbf{x} \tag{4.6}$$

which computes a weighted area of the region $\Omega_{\phi}^{+} \triangleq \{\mathbf{x} : \phi(\mathbf{x}) > 0\}$. This energy is introduced to speed up the motion of the zero level contour in the evolution process when the initial contour is not placed in the vicinity of the desired object boundary. The ballooning of the zero level contour is inhabited near the boundaries where J takes larger values.

A comparison of the Sobel edge map and the proposed edge probability is shown in Fig. 4.3. The awareness of local colour distribution avoids the converge of zero level contour stuck in a local minimum frequently caused by the traditional local edge detectors and facilitates the segmentation of natural image.

4.3.2 Statistical Prior Energy

An optimal partition $\mathcal{P}(\Omega)$ of the image plane Ω can be computed by maximizing the *a posterior* probability $p(\mathcal{P}(\Omega)|I)$ for the given image I [160]. Applying Bayes' rule, it can be expressed as

$$p(\mathcal{P}(\Omega)|I) \propto p(I|\mathcal{P}(\Omega))p(\mathcal{P}(\Omega)).$$

$p(\mathcal{P}(\Omega))$ allows to introduce prior knowledge such as geometric priors to cope with missing low-level information. Under the given prior, optimal two-region partition is achieved by maximizing

$$p(I|\mathcal{P}(\Omega)) = p(I|\Omega^{+})p(I|\Omega^{-}). \tag{4.7}$$

where Ω^{+} and Ω^{-} represent the regions inside and outside the contour respectively. Maximization of (4.7) is equivalent to minimizing its negative logarithm, we define $E_b(\phi)$ as

$$E_b(\phi) = -\omega_b[\log p(I|\Omega^{+}) + \log p(I|\Omega^{-})]. \tag{4.8}$$

We assume that the image I in each region is characterized by the individual pixel values at different locations $\mathbf{x}$ and the pixel values are i.i.d. Let $\phi(\mathbf{x}) > 0$ if $\mathbf{x} \in \Omega^+$ and $\phi(\mathbf{x}) < 0$ if $\mathbf{x} \in \Omega^-$. We reduce (4.8) to

$$E_b(\phi) = -\omega_b \int_\Omega (H(\phi) \log p(I(\mathbf{x})|\theta^+) + \\ (1 - H(\phi)) \log p(I(\mathbf{x})|\theta^-)) d\mathbf{x}. \tag{4.9}$$

where θ^+ and θ^- represent the foreground and background colour model respectively and ω_b is the coefficient specified in Table 4.1.

The foreground and background colour model are represented by Gaussian Mixture Model (GMM) as

$$p(I(\mathbf{x})|\theta_i) \quad = \quad \sum_{k=1}^{K_i} w_{ik} \mathcal{N}(I(\mathbf{x}); \mu_{ik}, \Sigma_{ik}),$$

with parameters w_{ik}, μ_{ik} and Σ_{ik} representing the weight, the mean and the covariance of the k^{th} component. The parameters of all GMMs ($\theta_i = \{w_{ik}, \mu_{ik}, \Sigma_{ik}, i = 1, \ldots, L, k = 1, \ldots, K_i\}$) are learned from observations of pixels; specifically the pixels in the user-specified area are assumed to be foreground and the border of the image is assumed to be the background. The second row in Fig. 4.1 visualizes the process of estimating the foreground and background colour model.

The user-touched area is usually a part of the desired object, and thus the foreground colour model has higher confidence than the background colour model, especially when the desired object intersects the border of the image. We propose a foreground consistency term to enforce the minimization of foreground statistical error as

$$E_u(\phi) = \frac{\omega_u \int_\Omega H(\phi)(1 - p(I(\mathbf{x})|\theta^+)) d\mathbf{x}}{\int_\Omega H(\phi) d\mathbf{x}}. \tag{4.10}$$

This energy term computes the averaged classification error $1 - p(I(\mathbf{x})|\theta^+)$ inside the zero level contour regardless of the accuracy of the background colour model. Fig. 4.5 gives examples where the background colour model is confused with the foreground whilst the foreground consistency term ensures the contour evolution proceeds as long as the foreground statistical error inside the contour is minimized.

Fig. 4.4 presents segmentation results by disabling the statistical prior energy and edge based energy respectively. Based on edge information alone, the system fails

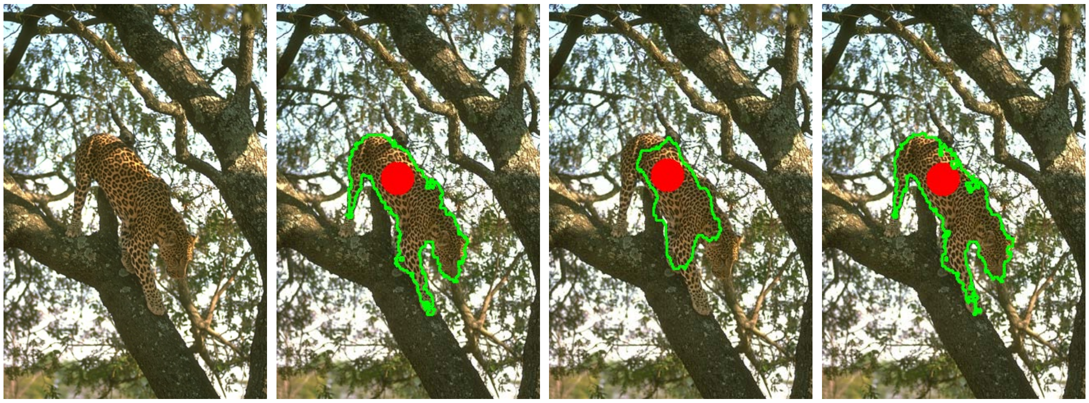


Figure 4.4: Contribution of edge based energy and statistical prior energy: (1) Source image (2) Segmentation by full system (3) Segmentation by disabling statistical prior energy (4) Segmentation by disabling edge based energy.

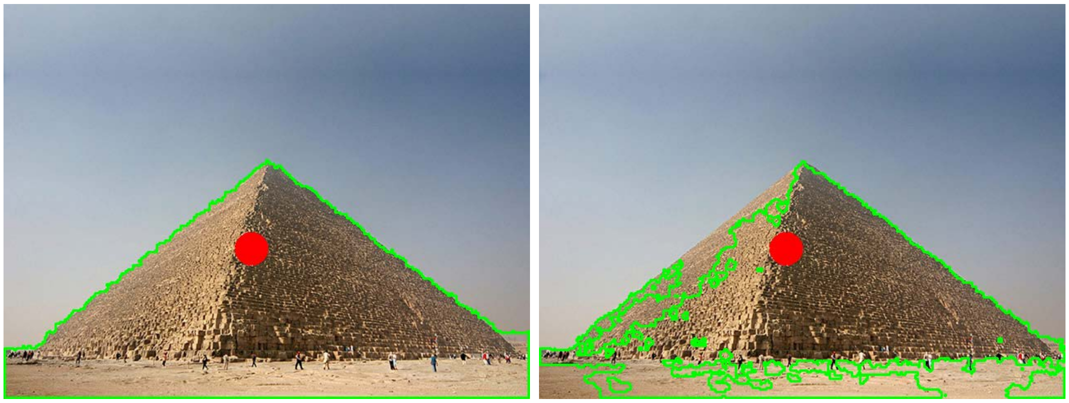


Figure 4.5: Contribution of foreground consistency term: (1) Segmentation by full system (2) Segmentation by disabling foreground consistency term.

to converge to the correct object boundary without considering the global colour distribution. Disabling the edge based energy leads to unsmooth segmentation caused by the statistical prior error. Combining these two achieves robust segmentation in the presence of inaccurate edge information or colour modeling.

4.3.3 Geometry Energy

People tend to select the geometrical center when they are indicating the object of interest. Although not a precise measurement, such a geometrical constraint provides a weak cue for the contour evolution process. We propose a central symmetry term to

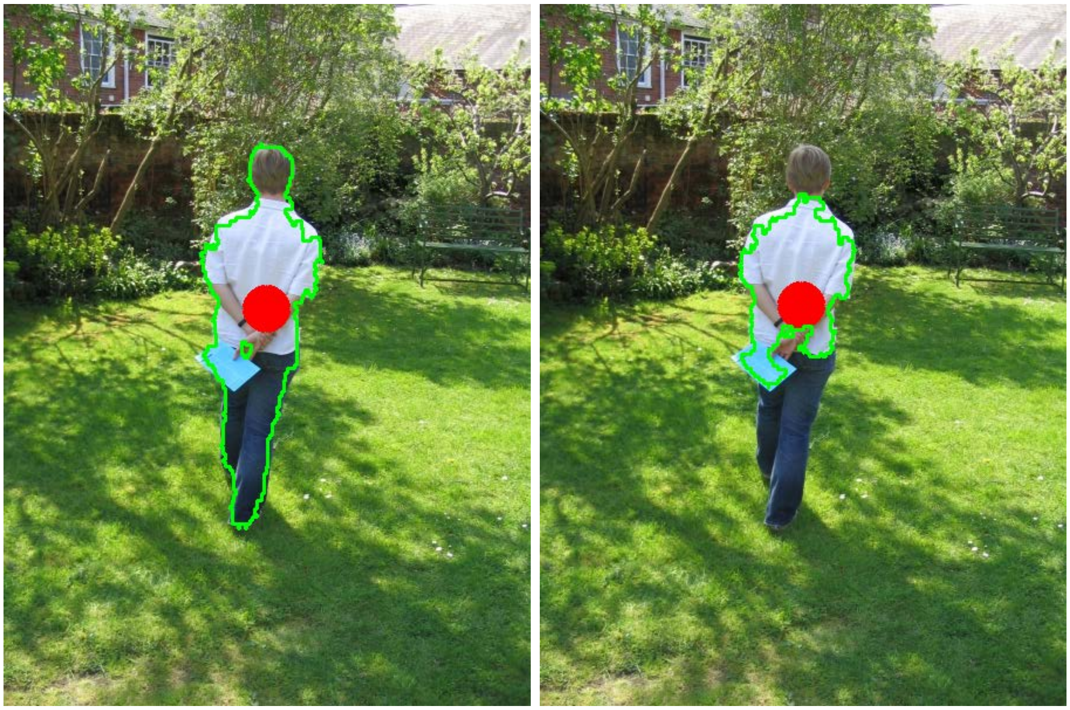


Figure 4.6: Contribution of geometry energy: (1) Segmentation by full system (2) Segmentation by disabling geometry energy.

reflect this geometrical constraint, by computing the spatial deviation of the geometrical center of zero level contour from the user-input point as

$$E_g(\phi) = \omega_g \left| \frac{\int_\Omega H(\phi)(\mathbf{x} - \overline{\mathbf{x}})d\mathbf{x}}{\int_\Omega H(\phi)d\mathbf{x}} \right| \tag{4.11}$$

where $\overline{\mathbf{x}}$ represents the user-input point. The closer the user-input point is to the geometrical center, the smaller this term would be. As the desired object could have very complex shape, this term is regarded as a relatively weak indication of the desired object's geometry. Fig. 4.6 shows an example of the situations where this higher level knowledge is explored to guide the contour evolution to segment the whole object of interest which can not be achieved by low level colour and edge information.

4.3.4 Distance Regularization

The proposed model incorporates the distance regularization term present in [130] to ensure stable evolution of the level set function, by penalizing the deviation of the level set function from a signed distance function. This deviation is characterized by the following integral

$$E_d(\phi) = \omega_d \int_\Omega \mathcal{P}(|\nabla\phi|)d\mathbf{x} \tag{4.12}$$

where $\mathcal{P}(s)$ is a double-well potential function defined as

$$\mathcal{P}(s) = \begin{cases} \frac{1}{(2\pi)^2}(1-\cos(2\pi s)), & \text{if} \quad s \leq 1 \\ \frac{1}{2}(s-1)^2, & \text{if} \quad s \geq 1. \end{cases}$$

This potential function maintains the signed distance property $|\nabla\phi| = 1$ only in a vicinity of the zero level contour to ensure accurate computation for curve evolution, while keeps the embedding function ϕ as a constant with $|\nabla\phi| = 0$ at locations far away from the zero level contour to smooth the embedding function. The distance regularization effect eliminates the need for reinitialization and thereby avoids its induced numerical errors.

4.3.5 Adaptive Weighting

Minimizing the proposed energy functional (4.3) with constant coefficients usually gives good segmentations. However, when the foreground and background distribution is not distinct, the Bayes error term would be non-discriminative and the contour evolution process would not converge to the desired object boundaries. In this case, the weight of Bayes' error term should be relatively small to increase the influence of other reliable terms. We expect it to be adaptively tuned based on the colour modeling error on a per image basis. To this end, we estimate the misclassifying error in foreground/background seeds based on the posterior probability

$$\eta = \frac{1}{|\Omega^+|}\sum_{\mathbf{x}\in\Omega^+} p(I(\mathbf{x})|\theta^-) + \frac{1}{|\Omega^-|}\sum_{\mathbf{x}\in\Omega^-} p(I(\mathbf{x})|\theta^+) \tag{4.13}$$

and define coefficient $\omega_b = \max\{\overline{\omega_b}(1-\eta), 0\}$. When the misclassifying error η is close to zero, the weight approaches $\overline{\omega_b}$. When the colour models are indistinct, ω_b approaches 0.

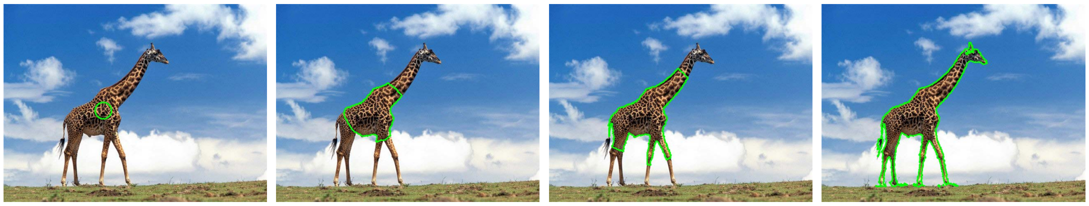


Figure 4.7: Contour evolution process: The initial contour and the zero level contours (green curve) after 100, 200, 400 iterations respectively

4.3.6 Numerical Approximation and Implementation

We use the standard gradient descent method to minimize the energy functional (4.3)

$$\begin{aligned}\frac{\partial\phi}{\partial t} &= -\frac{\partial E_e(\phi)}{\partial\phi}-\frac{\partial E_b(\phi)}{\partial\phi}-\frac{\partial E_u(\phi)}{\partial\phi}\\ &\quad -\frac{\partial E_g(\phi)}{\partial\phi}-\frac{\partial E_a(\phi)}{\partial\phi}-\frac{\partial E_d(\phi)}{\partial\phi}\end{aligned}\tag{4.14}$$

where the gradient flows are deducted as follows:

$$\begin{aligned}\frac{\partial E_e(\phi)}{\partial\phi} &= \omega_e\delta(\phi)\mathrm{div}(g\frac{\nabla\phi}{|\nabla\phi|})\\ \frac{\partial E_b(\phi)}{\partial\phi} &= \omega_b\delta(\phi)\log\frac{p(I(\mathbf{x})|\theta^+)}{p(I(\mathbf{x})|\theta^-)}\\ \frac{\partial E_u(\phi)}{\partial\phi} &= \omega_u\delta(\phi)[\frac{(1-p(I(\mathbf{x})|\theta^+))}{(\int_\Omega H(\phi)d\mathbf{x})^2}\\ &\quad -\frac{\int_\Omega(1-p(I(\mathbf{x})|\theta^+))H(\phi)d\mathbf{x}}{(\int_\Omega H(\phi)d\mathbf{x})^2}]\\ \frac{\partial E_g(\phi)}{\partial\phi} &= \omega_g\delta(\phi)\frac{|(\mathbf{x}-\overline{\mathbf{x}})-\int_\Omega(\mathbf{x}-\overline{\mathbf{x}})H(\phi)d\mathbf{x}|}{(\int_\Omega H(\phi)d\mathbf{x})^2}\\ \frac{\partial E_a(\phi)}{\partial\phi} &= \omega_a g\delta(\phi)\\ \frac{\partial E_d(\phi)}{\partial\phi} &= \omega_d\mathrm{div}(\frac{\mathcal{P}^{'}(|\nabla\phi|)}{|\nabla\phi|}\nabla\phi)\end{aligned}$$

To discretize the equations, we use a finite differences scheme. Considering the 2D case with a time dependent embedding function $\phi(x,y,t)$, the spatial derivatives $\partial\phi/\partial x$ and $\partial\phi/\partial y$ are approximated by the central difference, where the space steps are fixed as $\Omega_i x=\Omega_i y=1$. The temporal partial derivative $\partial\phi/\partial t$ is approximated by the forward difference. We discretize embedding function $\phi(x,y,t)$ as $\phi^k_{i,j}$, where (i,j) is the spatial index and k is the temporal index. The level set evolution equation (4.1) is discretized as $(\phi_{i,j}^{k+1} - \phi_{i,j}^{k})/\Omega_i t = F(\phi_{i,j}^{k})$ where $F(\phi_{i,j}^{k})$ approximates the right hand side in (4.1). The level set evolution is then expressed as an iteration process

$$\phi_{i,j}^{k+1} = \phi_{i,j}^{k} + \Omega_i t F(\phi_{i,j}^{k}), k = 0, 1, 2, \ldots \tag{4.15}$$

In the implementation, the Heaviside function H is approximated by a smooth function defined by

$$H_\epsilon(x) = \begin{cases} \frac{1}{2}(1 + \frac{x}{\epsilon} + \frac{1}{\pi}\sin(\frac{\pi x}{\epsilon}), & |x| \leq \epsilon \\ 1, & x > \epsilon \\ 0, & x < -\epsilon. \end{cases}$$

and the Dirac delta function δ is approximated by

$$\delta_\epsilon(x) = \begin{cases} \frac{1}{2\epsilon}(1 + \cos(\frac{\pi x}{\epsilon})), & |x| \leq \epsilon \\ 0, & |x| > \epsilon. \end{cases} \tag{4.16}$$

As the Dirac delta function and the Heaviside function multiply the entire image plane in (4.14), only the ϕ values in the vicinity of the zero crossing points need to be updated. This is the central idea of the narrow band methods [1]. The computational cost of a level set method can be substantially reduced by confining the computation to a narrow band around the zero level set contour. For our proposed formulation, as re-initialization is not needed due to the incorporation of distance regularization term E_d [130], the narrow band implementation is simple and straightforward. Our narrow band implementation allows the use of a large time step in the finite difference scheme to greatly reduce the iterations as long as the choice of the time step $\triangle t$ satisfies the Courant-Friedrichs-Lewy (CFL) condition $\omega_d \triangle t < (1/4)$ for numerical stability.

We adopt the narrow band method [1] to substantially reduce the computational cost of level set method by confining the computation to a narrow band around the zero level set contour. The narrow band scheme is implemented in the following major steps:

1. Compute the narrow band $B_k = \bigcup_{(i,j)\in C_0} N_{i,j}$, where C_k is the set of zero crossing points of ϕ_k and $N_{i,j}$ is a 3×3 neighborhood system centered at each point (i, j). If either $\phi_{i-1,j}\phi_{i+1,j} \leq 0$ or $\phi_{i,j-1}\phi_{i,j+1} \leq 0$, point (i, j) is regarded as a zero crossing point. $\forall (i, j) \in B_k$ and $(i, j) \notin B_{k-1}$, set $\phi_{i,j}^{k} = 2$ if $\phi_{i,j}^{k-1} > 0$, or else set $\phi_{i,j}^{k} = -2$ if $\phi_{i,j}^{k-1} < 0$.

Table 4.1: Parameters settings in the energy function

ω_e	ω_b	ω_u	ω_g	ω_a	ω_d	σ	ϵ	K
6	1	10	5	-4	0.04	-24	-1.5	400

2. Update the embedding function $\phi_{i,j}^{k+1} = \phi_{i,j}^{k} + \Omega_i tF(\phi_{i,j}^{k})$ on the narrow band B_k.

3. If k exceeds a predefined maximum number of iterations K, the evolution process is halted. Otherwise, go to (1).

In our prototype, we use a mouse click and a fixed brush size σ to simulate the finger touch of the user. The embedding function ϕ is initialized by extracting the contour of the user input brush stroke. The embedding function is assigned as 2 inside the contour and -2 outside the contour. We empirically choose the parameters in the formulation which are listed in Table 4.1. Fig. 4.7 illustrates the evolution process of the zero level contour which clearly indicates the speed of convergence.

4.4 Extension to Video Object Segmentation

The proposed TouchCut framework enables fast object segmentation with accurate boundary placement and strong region connectivity on still images. In this section, we extend TouchCut framework to video object segmentation. As one of the potential applications enabled by the proposed system, we stylize video objects or background into paintings based on the temporally coherent object/background region map.

After acquiring the object segmentation on the initial frame, TouchCut is performed on successive video frames using both photometric properties of the current frame and prior information propagated forward from previous frames. This information consists of:

i. an incrementally built GMM encoding the colour distribution of foreground/background over past frames;

ii. an initial contour for level set evolution;

iii. an estimated foreground object mask.

The image data labeled by the binary segmentation mask on previously segmented frames underpins accurate colour distribution of foreground and background region respectively when the luminance variation on successive frames is minor. In practice, to cope with variations in luminance often present in the sequence and cumulative segmentation error near boundary, the proportion of samples $S_{l,t-t_d} \in [0,1]$ $(t_d > 0), l \in l_f, l_b$ drawn from all foreground (l_f) and background (l_b) pixels from historical frame I_{t-t_d} decreases exponentially as the temporal distance t_d from the current frame I_t increases

$$S_{l,t-t_d} \propto e^{-t_d^2/\sigma_{t_d}^2}, \tag{4.17}$$

where σ_{t_d} is determined by the level of luminance variance. Smaller σ_{t_d} is selected when luminance variance is large, contributing more recent data to the GMM, otherwise the historical data contributes more to increase robustness.

We employ optical flow to create a per-pixel propagation of the foreground mask from frame I_{t-1} to create an estimated mask on frame I_t which is used as the shape prior $\tilde{\phi}$ which takes the value of the initial embedding function. We propose a shape energy term measuring the shape dissimilarity of two shapes represented by the embedding functions ϕ and $\tilde{\phi}$, which is commonly computing the area of the set symmetric difference

$$E_s(\phi) = \omega_s \int_\Omega (H(\phi) - H(\tilde{\phi}))^2 d\mathbf{x} \tag{4.18}$$

ω_s is inversely proportional to the alignment error in the scope of the foreground object Ω_f

$$\omega_s \propto 1/\sqrt{\frac{1}{|\Omega_f|}\sum_{\mathbf{x}\in\Omega_f} ||I_{t-1}(\mathbf{x}) - I_t^{'}(\mathbf{x})||^2}. \tag{4.19}$$

where $I_t^{'}$ is the warped colour image from frame I_{t-1} to I_t by the optical flow. Accurate alignment generally indicates reliable motion estimation and such shape priors thus contribute more to the contour evolution.

Applying the standard gradient descent method to minimize the shape energy term, we deduct the gradient flow of shape energy as

$$\frac{\partial E_s(\phi)}{\partial \phi} = 2\omega_s \delta(\phi)(H(\phi) - H(\tilde{\phi})). \tag{4.20}$$

The initial contour of TouchCut on current frame I_t is acquired by computing the deviation of the initial contour ϕ_0^0 on the first frame from the geometrical center g_0^0. Let the center of ϕ_0^0 be b_0, and the center b_t of the initial contour on I_t is estimated as $b_t = g_0^t + b_0 - g_0^0$. The geometrical center on frame I_t, g_0^t is estimated from the estimated foreground mask. We use a circle centered at b_t with a radius r_c as the initial contour on frame I_t. r_c is two times as large as the maximum distance from the contour to the touch point on the initial frame.

Due to the inherent error of optical flow, the new initial contour might be slightly drifting from the desired object, i.e. part of the area inside the initial contour might be the background. The robust formulation based on colour and shape priors push the contour evolution, minimizing the pixel classification error inside and outside the contour, while satisfying other criteria defined in the energy function, to achieve accurate and temporally coherent segmentation.

4.5 Experiments and Comparisons

We have applied the proposed algorithm on a dataset consisting of the combined Berkeley $BSDS300$ dataset [144], and image dataset accompanying GrabCut [178]. We also demonstrate the application of TouchCut to video sequences exhibiting clutter and agile motion. We assess segmentation performance both qualitatively through visual comparison to prior work, and quantitatively based on a manual ground truth segmentation. We indicate relative performance to the state of the art for both image and video comparisons.

4.5.1 Segmentation of Still Images

Fig. 4.8 presents a qualitative comparison of the proposed method with standard graph cut (middle) [22] and GrabCut (right) [178]. In the case of Graph cut, we adapted such the modeling of colour distributions to exactly that of the proposed approach to make a fair comparison — i.e. to solely evaluate performance of the single touch segmentation

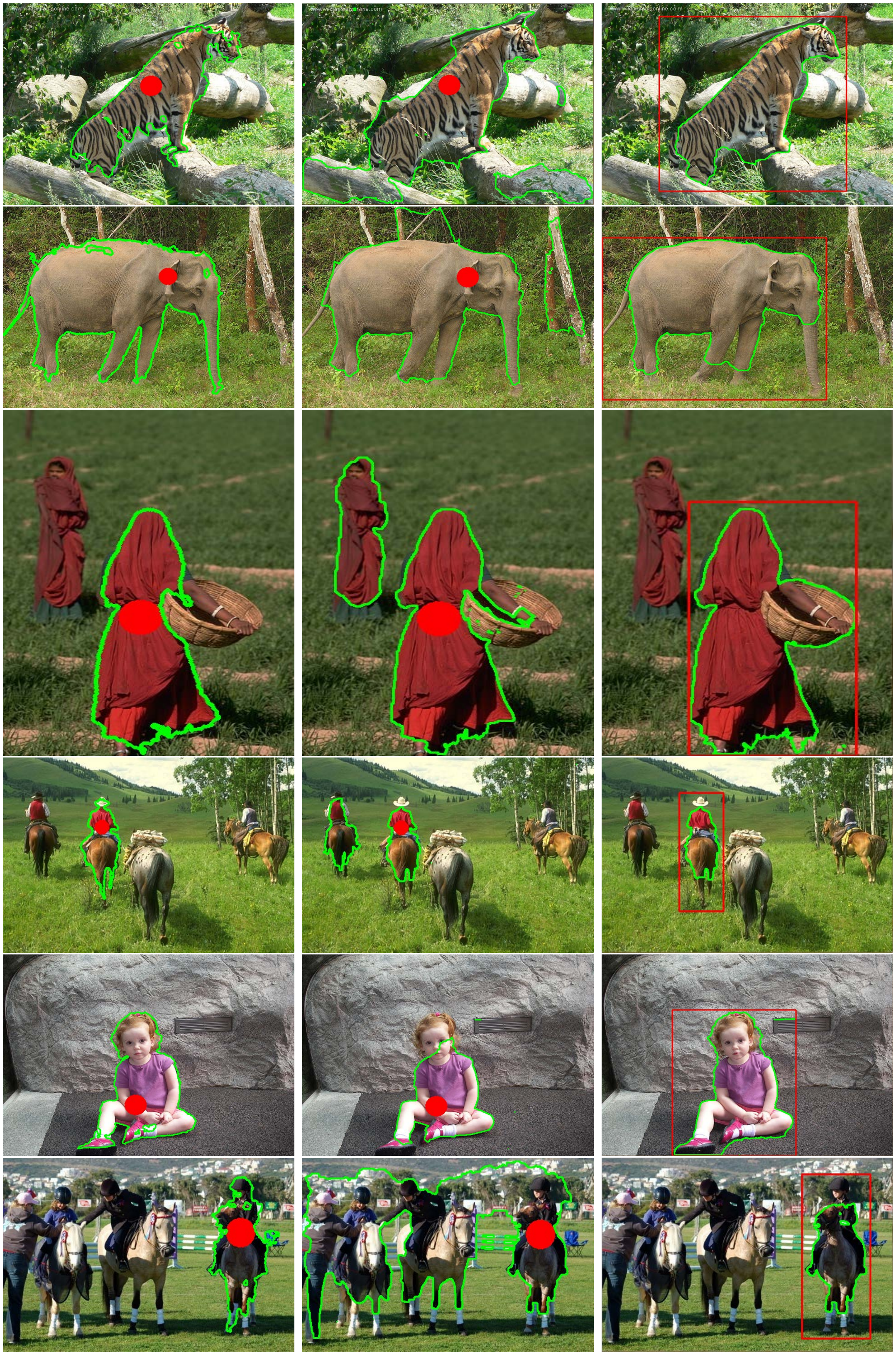


Figure 4.8: Comparison of proposed method (left) with graph cut (middle) and GrabCut (right). The contour of segmented object is shown in green.

paradigm. Specifically, the foreground colour was modeled from the pixels in the user-touch area while the background colour was modeled by taking pixels from the border of the image. With significantly less user input, our method gives satisfactory segmentation even when the foreground and background colours lack distinction (first row) or regions exhibit complex topology (second row). Graph cut fails to separate the objects exhibiting a similar colour to the desired object, whilst our approach fills the desired region by expanding from the interior of the selected object outwards and explicitly considers the object boundary and geometric properties. GrabCut presents better spatial constraints than graph cut, benefiting from the bounding box while failed to exclude noisy extraneous regions (e.g. the varying levels of luminance underneath the tiger) which do not appear outside the bounding box. The latter method also suffers from "short-cutting" regions (e.g. the elephant's legs and trunk).

For our objective comparison, we adopt the Berkeley Segmentation Benchmark [144] to quantify segmentation accuracy against a manual specified ground-truth. This benchmark considers two aspects of segmentation performance. Precision measures the fraction of true positives in the contours produced by a segmentation algorithm. Recall indicates the fraction of ground truth boundaries detected in the segmentation. The global F-measure, defined as the harmonic mean of precision and recall, provides a useful summary score for the segmentation algorithm [144]. Averaging across the dataset our proposed method receives a F-measure of 0.765 which outperforms the adapted graph cut (F-measure 0.538) and GrabCut (F-measure 0.697) despite interaction being limited to just a single touch.

Fig. 4.9 presents further segmentation results [1]. The first row shows the results on highly-textured images. The edge probability map enables the contour evolution over colour-texture homogeneous regions without being stopped at local minimum. The second row shows the segmentation results of images with indistinct foreground and background colours. In this case, the colour modeling error is large which adaptively results in a small weight on colour based term E_b. On the other hand, the foreground consistency term E_u enforces the region inside the zero level contour to be coherent in

[1] More results can be viewed online at: `http://personal.ee.surrey.ac.uk/Personal/Tinghuai.Wang/CVIU2011`

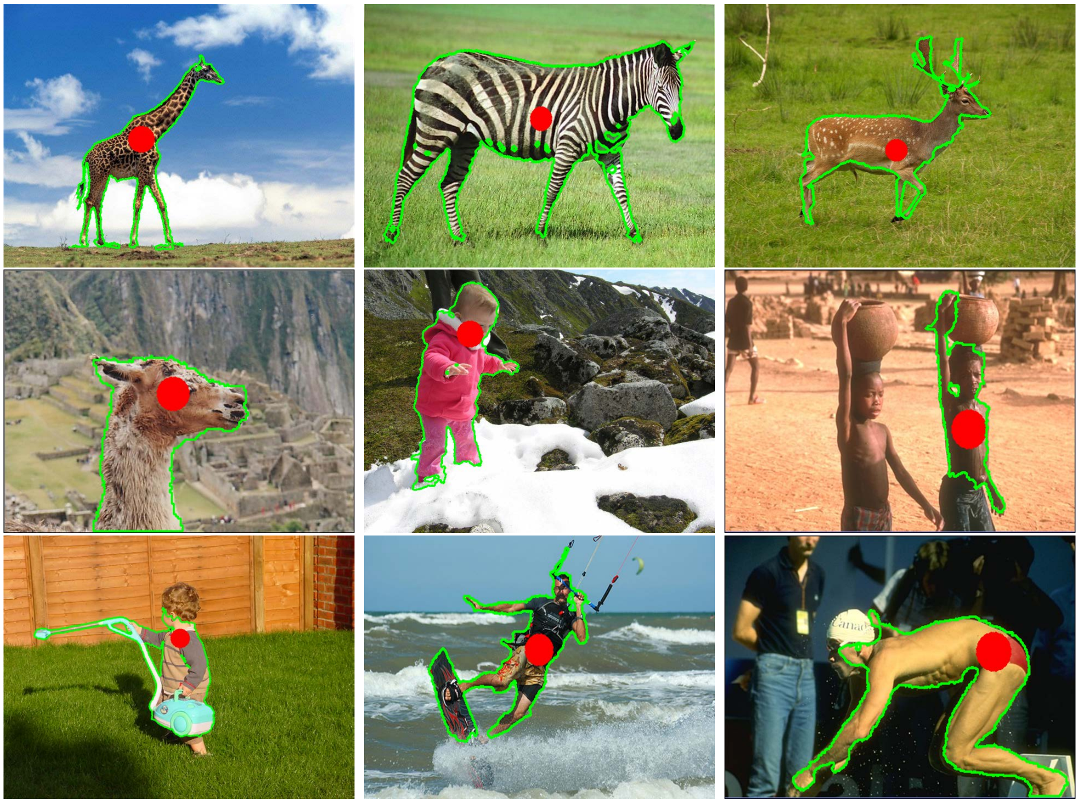


Figure 4.9: Representative segmentation results from our dataset, discussed within Subsec.4.5.1

the sense of colour distribution regardless the background colour distribution. Such a constraint significantly imposes the stability of the contour evolution process in the case of indistinct colour distributions. The third row contains segmentation results to deal with objects with complex shape. By leveraging the strength of the implicit contour representation in level set methods, our system is robust in coping with complex topologies without exhibiting short-cutting problem which is common in graph-cut based systems. The system is able to cope with weak boundaries and complex foregrounds and backgrounds, to extract meaningful objects in most cases.

For all image results the running time on a Core2 2.1 GHz PC is constant ~ 0.4 second per VGA image (640×480). More representative segmentation results are presented in Fig. 4.10.

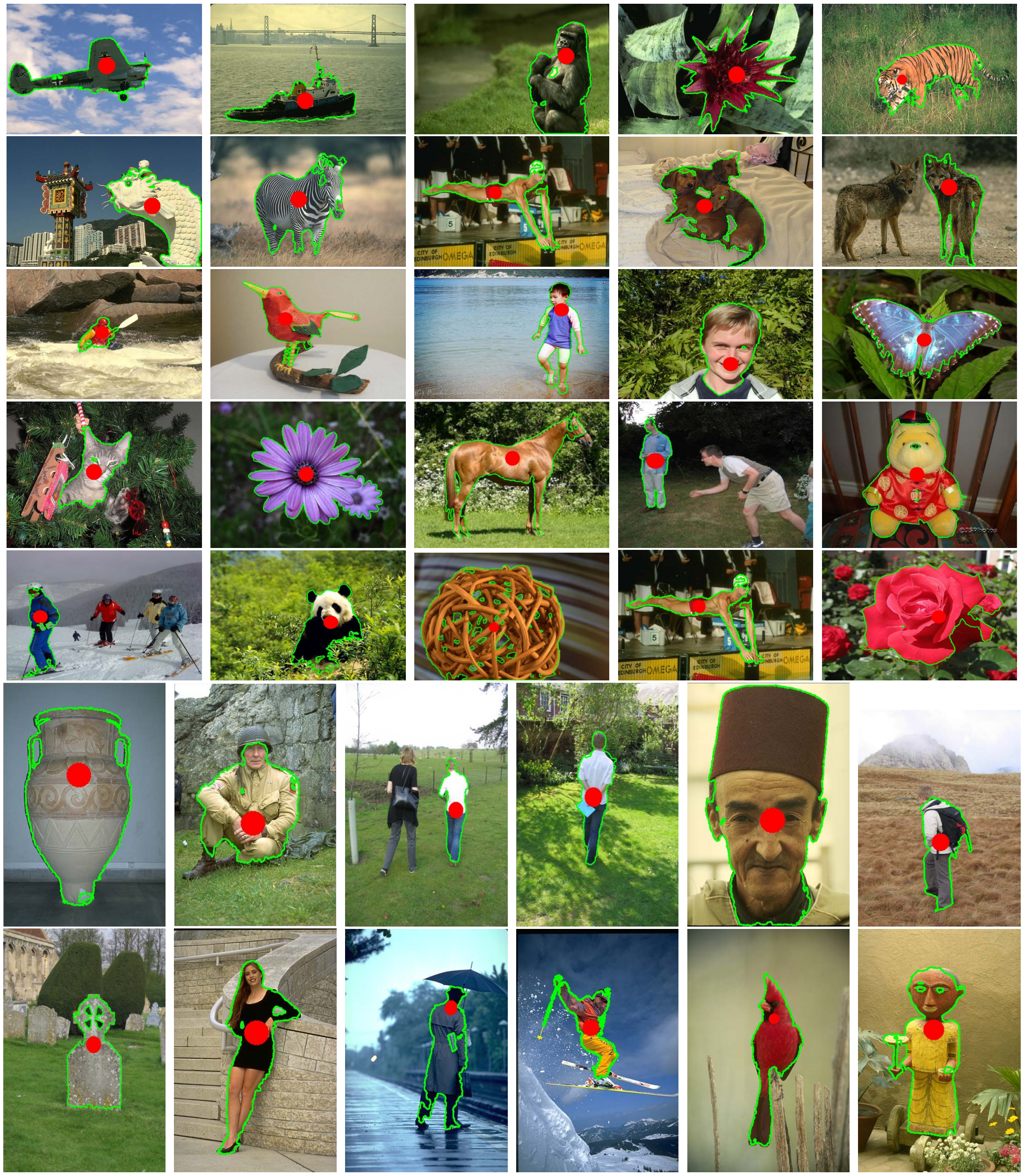


Figure 4.10: Additional representative segmentation results from our combined $BSDS300$ and GrabCut dataset.

4.5.2 Segmentation of Video Sequences

We quantitatively test TouchCut on three videos and ground-truth for the primary foreground object present in [36] and [206]. The videos tested exhibit various challenging conditions such as foreground and background colour overlap, luminance variation, shape

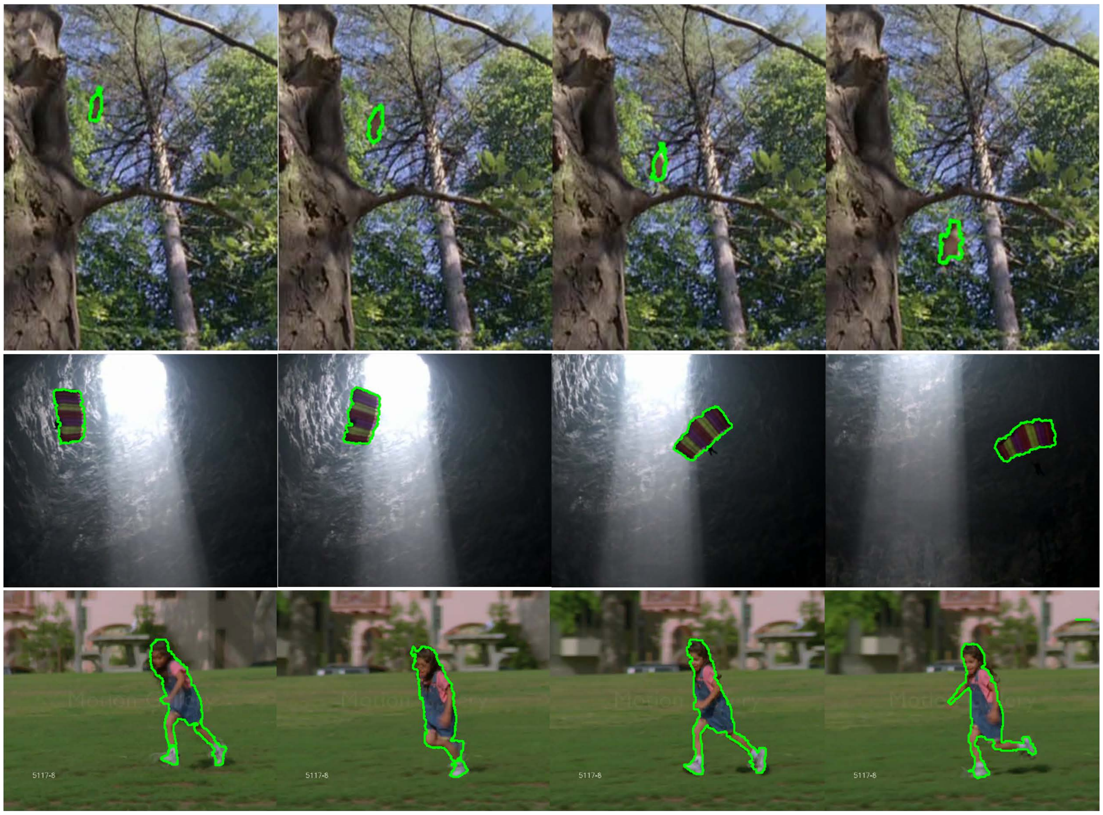


Figure 4.11: Qualitative segmentation results on *BIRDFALL* (row 1), *PARACHUTE* (row 2) and *GIRL* (row 3) respectively. 4.5.2.

deformation and camera motion. We compare against two state-of-the-art approaches: another level set based tracking approach by Chockalingham *et al.* [36] and the 'motion coherent segmentation' method of Tsai *et al.* [206]. These methods require human labeling of the object boundary (contour) in the first frame, whilst TouchCut requires minimal user intervention to guide the segmentation of whole video via a single touch on the first frame. The segmentation accuracy is quantified as the average per-frame pixel error rate, $\epsilon(S) = \frac{|XOR(S,GT)|}{F}$, where S is the segmentation, GT is the ground-truth segmentation and F is the total number of frames. As shown in Table 4.5.2, our method outperforms the approaches present in [36] and [206] on two of the three videos (*PARACHUTE, BIRDFALL*), and produces the second best result on the *GIRL* video. Our higher error rate on *GIRL* is caused by the inaccurate optical flow motion estimations and indistinct colour of foreground and background, which is reasonable since the object exhibits large appearance variation, considering TouchCut does not

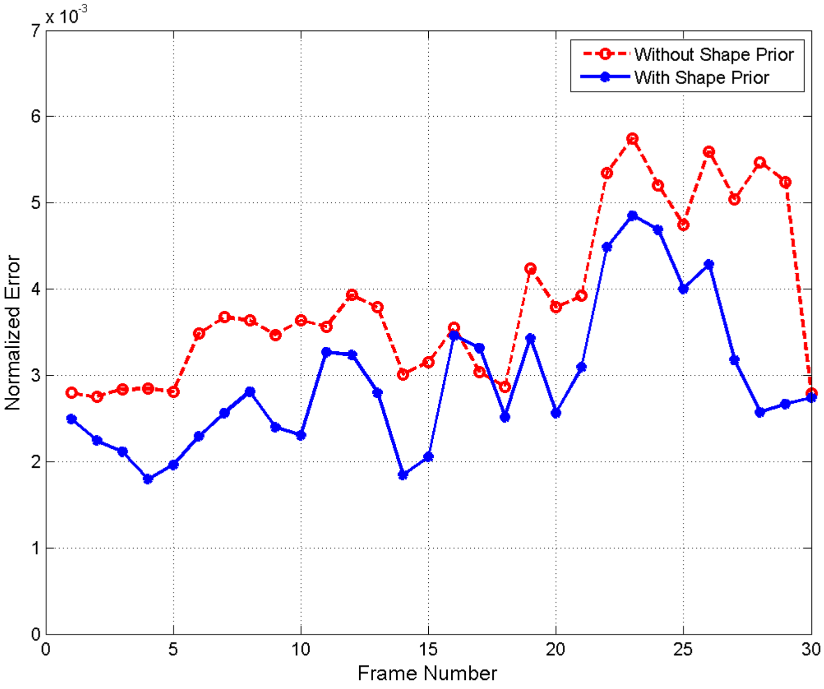


Figure 4.12: Illustrating the importance of the shape prior for Video TouchCut. Comparison of the full TouchCut with a baseline that does not incorporate a shape prior. Shape energy improves segmentation quality (normalized pixel error rate on *BIRDFALL*).

either explicitly employ global optimizations to enforce motion coherence across frames as in [206] and or manually bootstrapping the first frame. Further, TouchCut requires only a single touch to bootstrap the entire video segmentation — and we believe these comparative results to be very encouraging. Further qualitative segmentation results are shown in Fig. 4.11.

We study the impact of the proposed shape prior underpinning our video segmentation, comparing against a baseline implementation that does not incorporate the shape prior, but otherwise follows the same pipeline as our full method. Fig. 4.12 shows the results. The shape energy significantly improves segmentation accuracy, quantified against a manual ground truth.

	TouchCut	[206]	[36]
BIRDFALL	**0.003 (248)**	0.003 (252)	0.005 (454)
PARACHUTE	**0.002 (228)**	0.002 (235)	0.004 (502)
GIRL	0.012 (1691)	**0.009 (1304)**	0.012 (1755)

Table 4.2: Video segmentation error expressed as the average fraction of mis-segmented pixels (false positive plus false negative) per frame. Absolute number of mis-segmented pixels in parentheses (averaged per frame).

4.5.3 Application to Video Stylization

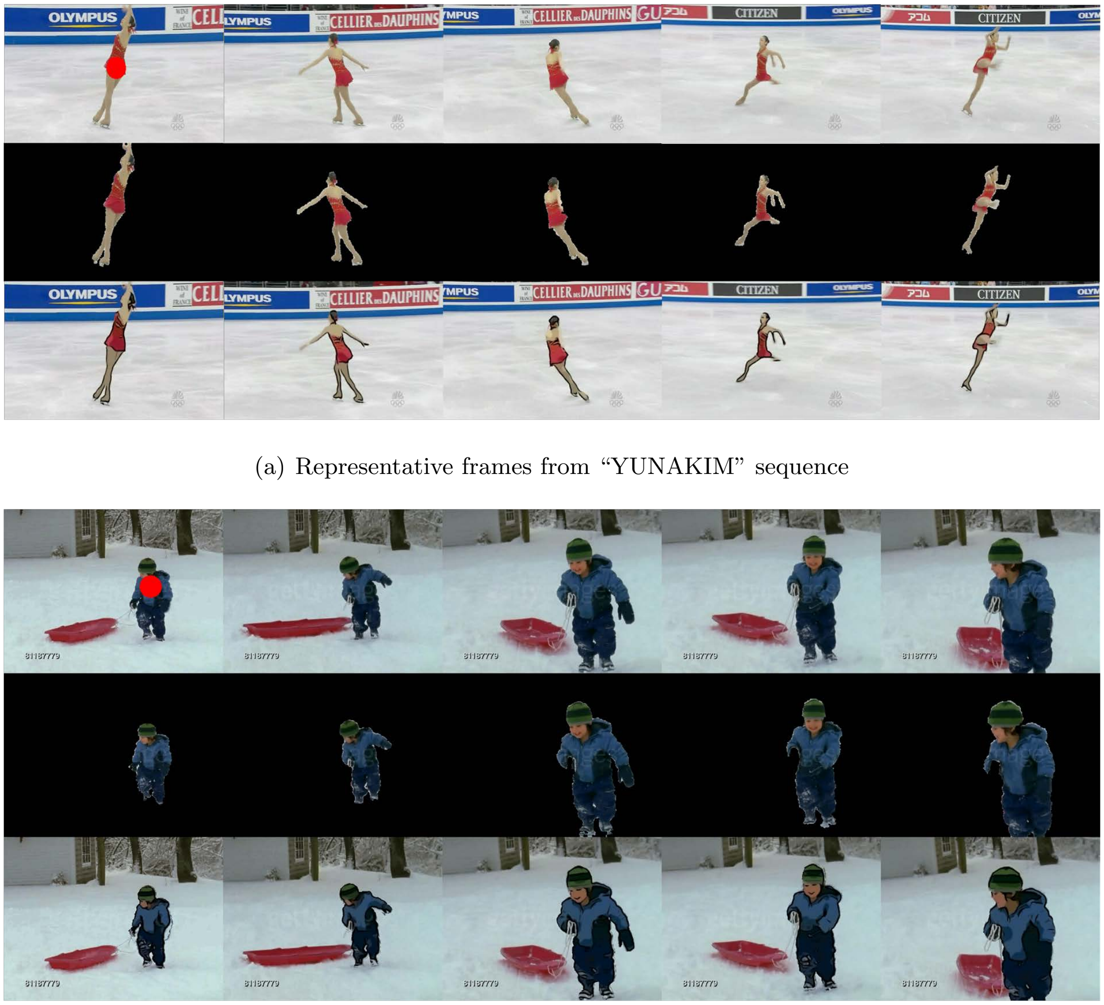
Temporally coherent segmentation of video forms a stable representation of visual structure in the scene which enables other computer vision and graphics applications. We demonstrate an application of TouchCut system on videos to create stylized region-based effects such as painterly rendering. We incorporate a framework of automatic non-photorealistic rendering by Kyprianidis and Döllner [118] to facilitate the domestic user to create artistic stylizations on either the desired object or the background scene.

As qualitative evaluation, we apply our segmentation algorithm to several video sequences exhibiting both slow moving and agile motion — summarized in Table 4.3. Fig. 4.13 presents the segmentation results applying TouchCut to these five video sequences and the foreground object or background painterly stylization effects. Our segmentation algorithm ensures the foreground and background regions deform in a coherent manner.

In Fig. 4.13(a) there is significant agile motion in "YUNAKIM" – Yuna swings and suffers frequent inter-occlusion over duration of the clip. Despite the adoption of a

Sequence	**Motion**	**# of Frames**
YUNAKIM (Fig. 4.13(a))	Agile	225
BOY (Fig. 4.13(b))	Slow	190
BEACH (Fig. 4.14(a))	Medium	300
LION (Fig. 4.14(b))	Slow	201
WALK (Fig. 4.14(c))	Medium	200

Table 4.3: Summary of video sequences used in our qualitative evaluation, annotated as to motion and number of frames present.



(a) Representative frames from “YUNAKIM” sequence

(b) Representative frames from “BOY” sequence

Figure 4.13: Segmentation results applying TouchCut to video sequences and foreground object stylization effects (source in top row, foreground object cut-out in middle row, painterly rendering on foreground object in bottom row). Please refer to **http://personal.ee.surrey.ac.uk/Personal/Tinghuai.Wang/CVIU2011** for these and further results.

forward propagation (2D+t) strategy over several hundred frames of video there is no significant degradation. With an incrementally learned GMM colour model, TouchCut is able to deal with the strong luminance variation on the boy’s face (“BEACH”) and produce stable segmentations with temporal coherence present in Fig. 4.14(a). Similar situation can be observed in “WALK” sequence, where both strong luminance variation and agile motion are present. As an application of TouchCut, the domestic user can choose either the foreground or background to create special effects, i.e. painterly stylization, with single finger touch on the first frame as shown in Fig. 4.13 (foreground object stylized) and Fig. 4.14(c) (background stylized). Drawing upon the coherent video segmentation by TouchCut, the stylized videos create either a painting object in a realistic scene or a realistic object in an unrealistic scene with painting style. Other special effects would be creating a new movie with the selected object in a totally different scene or emphasizing the desired object by blurring the background.

4.6 Conclusion

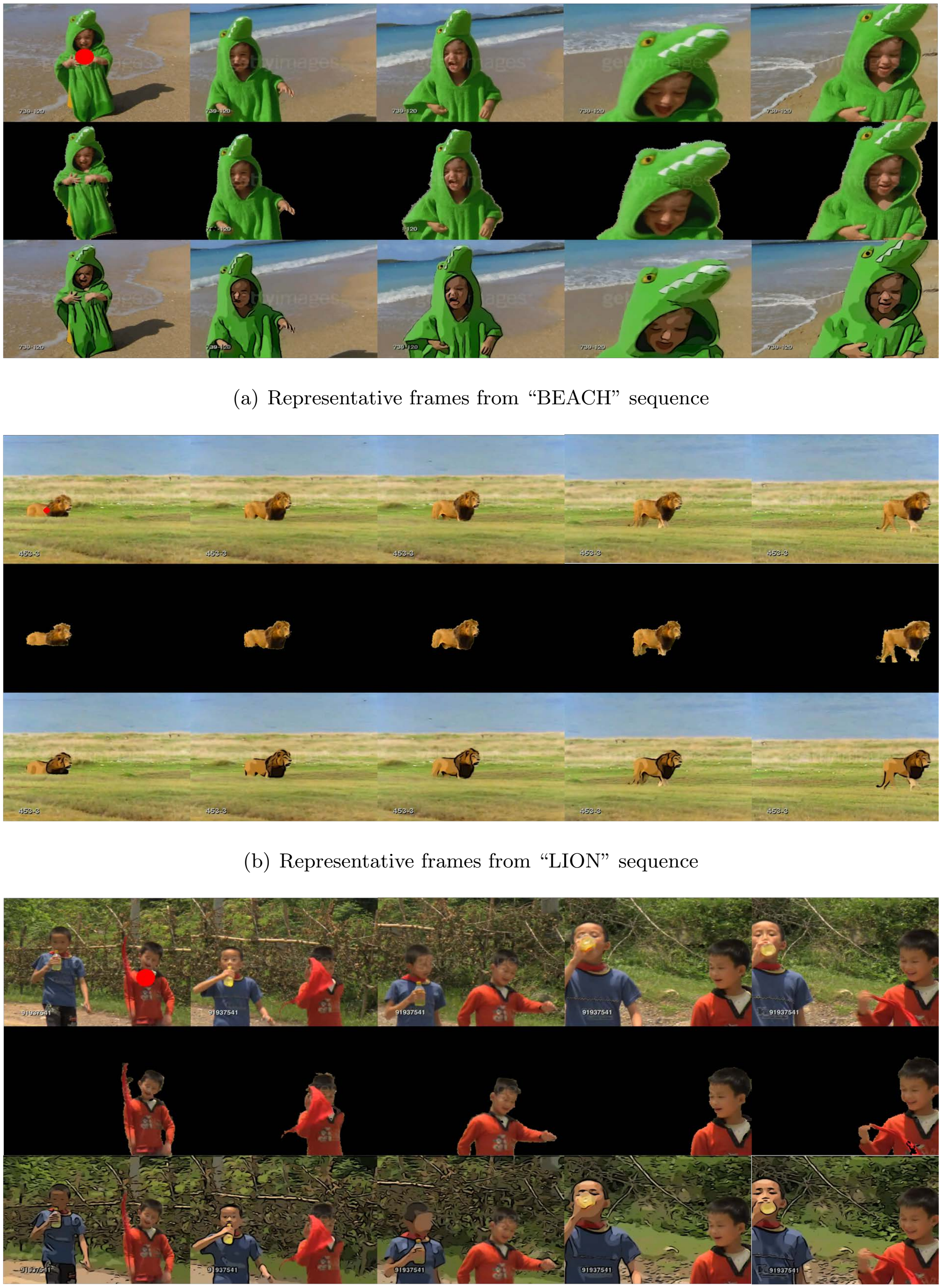
We have presented a single-touch object segmentation system using level set methods. The core contribution is an edge-region-geometry based segmentation model to robustly tackle the interactive object segmentation problem — encoding boundary probabilities of colour-texture homogeneous regions, and the statistical and geometric priors inferred from the user input. Our edge model gives a robust description of the coherent colour-texture region, which mitigates against the contour becoming stuck in local minima in the presence of noisy data. This frequently occurs in prior approaches, where traditional intensity gradient-based edge maps are used. Edge information alone only provides local information to drive contour evolution towards potential object boundary. Augmenting this model with colour information from user input introduced a global term, balancing the *a posterior* probabilities of region models inside and outside the putative object contour.

By leveraging the flexibility of level set methods in energy minimization, our system achieved promising results in various natural images with complex scenes and objects. We also demonstrated that TouchCut can be extended to segment video sequences into temporally coherent foreground and background region maps. This gives rise to potential applications to video special effects (e.g. artistic stylization) with minimal user intervention, that may be suited to consumer touch-screen video cameras. Coherence was promoted through an incrementally learned colour model, providing robustness against drift of the contour otherwise caused by motion estimation error. The introduction of a shape prior into the motion estimation framework was shown to deliver a further significant enhancement to coherence, especially when the foreground and background colour distribution became indistinct.

4.7 Future Work

TouchCut still experiences difficulties in separating the desired object from the adjacent background in the presence of highly similar colours. This remains an open question in the image segmentation community in the absence of other higher level semantic priors, e.g. shape, or other forms of global measurement. One interesting direction for future work would be to improve the background colour modeling by measuring the salience of different dominant colour modes. Another direction of future work with respect to the video extension might include detecting occlusion boundaries discovered from motion disparity in the scene, and using these to compensate for any ambiguity in appearance between the foreground and background.

Future applications of TouchCut fall within our original project motivation, to develop an image and video object segmentation algorithm with minimal user intervention suitable for emerging tablet and touch-screen devices. These applications could span embedded object extraction and tracking, intelligent focus, and video stylization [42, 219] on these devices.



(a) Representative frames from “BEACH” sequence

(b) Representative frames from “LION” sequence

(c) Representative frames from “WALK” sequence

Figure 4.14: Additional segmentation results applying TouchCut to video sequences and using the matte to create foreground (a and b) or background (c) object stylization effects (source in top row, foreground object cut-out in middle row, painterly rendering on foreground/background object in bottom row).

Part III

Video Stylisation

Chapter 5

Stylized Ambient Displays of Visual Media Collections

In this chapter we build structured representations for visual media collections, specifically from low level content parsing and understanding, to intelligent browsing and composition of visual media. We first present a novel video segmentation algorithm which performs a multi-label graph cut on successive video frames to parse the video into coherent spatial segments. This stable representation of visual structure facilitates both the coherent artistic stylisation and region correspondence between frames. The latter enables aesthetically pleasing composition of different video clips. A hierarchical representation for media collection is proposed to present a coarse-to-fine structuring of media items using graph optimisation. These representations of structure at different levels of abstraction underpin a system to automatically select, stylise and transition between digital contents. This chapter supports our claim that the improved stability of the structure extracted from video sequences enhances the temporal coherence of artistic renderings, broadening the gamut of potential expressive styles and enhancing the user engagement of visual media consumption.

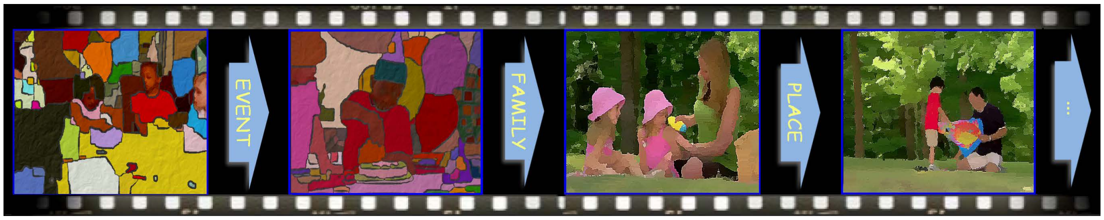


Figure 5.1: The Digital Ambient Display (DAD) selects, stylizes and transitions between home digital media items (photos and videos) according to semantic and visual similarity. The paths through the media collection are passively influenced by user interest measurement (gaze detection).

5.1 Introduction

The proliferation of video and image data in digital form creates demand for an effective means to browse large volumes of digital media in a structured, accessible and intuitive manner. This chapter proposes a novel approach to the consumption of home digital media collections, centred upon *ambient experiences*. Ambient experiences are distinguished from compelling or intense experiences in that they are able to co-exist harmoniously with other activities such as conversations, shared meals and so forth. An ambient experience does not demand the full attention of the user but is able to play out in a pleasing, unobtrusive way such that fresh and interesting content is available in the attention spaces of everyday life. We seek to emulate, for digital media, the serendipitous process of rediscovery often experienced whilst browsing physical media archives (e.g. a box of photos in the attic) that can trigger enjoyable reminiscence over past memories and events.

Although considerable research has been devoted to direct interactive approaches for browsing digital media collections, there is little previous work addressing the problem of displaying digital content in an *ambient* manner. Typical approaches to browsing small or medium scale photo sets project thumbnails onto either a planar or a spherical surface, so that images that are visually similar are located in close proximity in the visualization [47, 85, 184, 176]. Large photo sets are often handled by clustering content into subsets, sometimes arranged hierarchically for visualization and manual navigation [114, 35, 70]. With the expected proliferation of large format video displays around the home, recent work explores the specific domain of household digital media interaction [9, 227]. Yet, the *ambient* dissemination of home visual media, and the associated issues of interaction in an ambient context, remain sparsely researched.

5.1.1 Digital Ambient Display (DAD) concept

The *Digital Ambient Display* (DAD) is an always-on display for living spaces that enables users to effortlessly visualize and rediscover their personal digital media collections. DADs address the paradoxical requirement of an autonomous technology to passively disseminate media collections, that also enables minimal interaction to actively navigate routes through content that may trigger interest and user reminiscence. By transitioning between selected media items, the DAD passively presents a global summary visualizing the essential structure of the collection. This results in an evolving temporal composition of media, the sequencing of which considers both media semantics and visual appearance, as well as adaptively responding to user attention (sensed via gaze detection). Rather than simply stitching digital content together, we harness artistic stylization to depict image and video in a more abstract sense. In contrast to photorealism, which often proves distracting in the ambient setting (e.g. a television in the corner of a café), artistic stylization provides an aesthetically pleasing and unobtrusive means of disseminating content in the ambient setting; creating a flowing, temporal composition that conveys the essence of users' experiences through an artistic representation of their digital media collection (Figure 5.1).

Creating a DAD requires that the media be automatically parsed into an structured representation that enables semantically meaningful routes to be navigated through the collection. This process is dependent on meta-data tags user-assigned to each media item. Furthermore, the visual content within individual media items must be also be parsed into a mid-level *visual scene representation* that enables both:

1. Artistic rendering of media into aesthetically pleasing forms

2. Generation of appropriate transition effects and sequencing decisions, to create an appealing temporal composition.

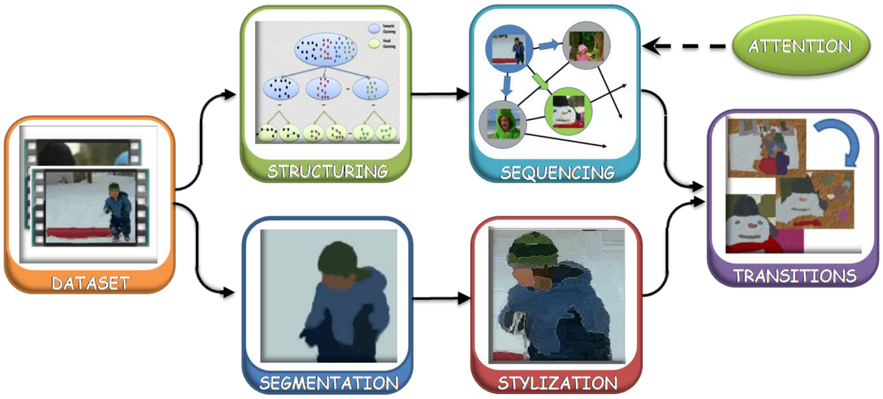


Figure 5.2: System overview. User media (photos, videos) are hierarchically clustered according to content semantics and appearance. The sequencing of displayed content is driven both by this automated clustering, and via passive measurement of user attention. Videos and photos are segmented into region maps encoding visual structure. These region maps drive the artistic stylization and transition processes on the display.

While artistic rendering of images is much explored, temporally coherent stylization of video is still a challenging task which requires a stable and consistent description of the scene structures. Following [51, 44] we identify a color region segmentation as being an appropriate "mid-level" scene abstraction, and in Section 5.4 contribute a novel algorithm for segmenting video frames into a deforming set of temporally coherent regions. We demonstrate how these regions may be stylized via either shading or stroke-based rendering, to produce coherent cartoon and painterly video styles (Section 5.4.5). We describe our hierarchical approach to structuring the media collection in Section 5.3, and describe how content is sequenced at run-time using that representation in Section 5.5. Qualitative evaluations of segmentation coherence, and a quantitative user evaluation of the DAD, are presented in Section 5.6.

5.2 System Overview

The Digital Ambient Display (DAD) visualizes home media collections comprising photos and videos. Videos are ingested as short, visually interesting clips that form the atomic unit of composition. Obtaining such clips differs from classical shot detection as raw home footage tends to consist of a few lengthy shots. The automatic editing algorithm proposed in Ch. 3 performs this pre-processing. The ingested media collection is clustered into a hierarchical representation according to semantic content (derived from keyword meta-data tags attached to the photo or video), and visual similarity (computed from the photo, or a representative video frame — typically mid-sequence). The automated clustering and related pre-processing are described in Section 5.3.

At run-time the DAD creates a temporal composition from a subset of media items from the structured collection, creating a media sequence that flows smoothly with respect to both scene appearance and semantics. For example, the DAD might select a clip or image of the family in the garden, and follow this with a family clip or image in semantically similar alternate environment such as a park. To promote user interest in the display, media choice is also governed by the level of user attention; passively measured using gaze detection. Persistent attention will guide the temporal composition toward semantically similar content to that which attracted the user's gaze. This real-time sequencing process is described in Section 5.5.

Presentation of video in the DAD is underpinned by a novel algorithm for segmenting video frames into temporally coherent colored regions (sub-secs. 5.4.1-5.4.3). These region maps form a stable representation of visual structure in the scene that is used both to drive artistic rendering algorithms for stylization (sub-Sec. 5.4.5), and to perform matching of scene elements between frames in order to generate animated clip transitions (sub-Sec. 5.5.2). Photographs are similarly segmented into colored regions, and for convenience are treated as single-frame videos within our framework. Figure 5.2 provides an overview of the complete DAD system.

5.3 Structuring the Media Collection

We represent the media collection as a hierarchy of pointers to media items. Each node in the tree represents a subset of the media collection sharing a common semantic theme or visual appearance.

5.3.1 Hierarchical Clustering

Our top-down approach recursively splits the collection, applying unsupervised clustering to each node using the Affinity Propagation (AP) algorithm [64]. In contrast to k-means clustering, which iteratively refines an initial randomly-chosen set of exemplars, AP simultaneously considers all data points within a node as potential exemplars and iteratively exchanges messages between data points until a high-quality set of exemplars and corresponding clusters gradually emerges. AP requires only a measure of similarity between items, rather than a feature vector, and does not require *prior* knowledge of the dataset such as the number of clusters present and representatives for the different clusters.

Clustering proceeds in two phases. The initial phase constructs higher levels of the tree using a measure of semantic similarity (sub-Sec. 5.3.2) that exploits user-provided tags on media items. AP is applied recursively to each node until no further division occurs (i.e. no semantic differentiation can be made between media items at a particular node). The second phase of our process then constructs lower levels of the tree from the leaf nodes of the first phase, using AP to cluster items based on a measure of visual similarity (sub-Sec. 5.3.3). Higher levels of the tree thus provide semantic summaries of the media reflecting the diversity of the visual content in the dataset. Clusters at lower levels contain predominantly visually similar images at various levels of detail (Figure 5.3).

5.3.2 Semantic Similarity

To compute the semantic similarity (S_s) of a pair of media items, we measure their tag co-occurrence. Given a vocabulary $\mathcal{V} = \{w_1, \ldots, w_K\}$ of K keywords present within all user-provided tags, the similarity of a pair of keywords is computed using asymmetric co-occurrence [192], indicating the probability of w_i appearing in a tag set given the presence of w_j:

$$p(w_i|w_j) = \frac{|w_i \cap w_j|}{|w_j|} \tag{5.1}$$

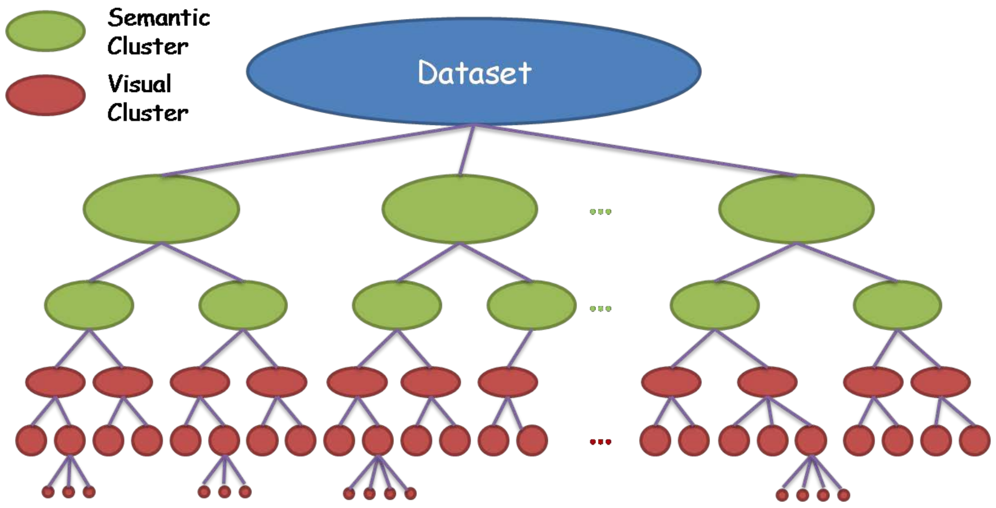


Figure 5.3: The structuring process starts with the whole dataset corresponding to the root node of the tree and continues splitting until all the leaf nodes cannot be further split by recursively applying unsupervised clustering. Higher levels of the tree are clustered based on semantic (keyword) similarity, while lower levels are constructed based on visual similarity.

where $|w_i \cap w_j|$ denotes the number of media items tagged with both keywords w_i and w_j, while $|w_j|$ is the number of items tagged with keyword w_j.

Following [192], we compute the asymmetric similarity between two sets of tags containing multiple keywords $T^1 = \{w_1^1, w_2^1, \ldots, w_N^1\}$ and $T^2 = \{w_1^2, w_2^2, \ldots, w_M^2\}$ corresponding to media items I_1 and I_2 as:

$$S_s(I_2|I_1) = \frac{\sum_{n=1}^{N} \sum_{m=1}^{M} p(w_m^2|w_n^1)}{M \cdot N}. \tag{5.2}$$

5.3.3 Visual Similarity

We adopt a Content-Based Image Retrieval (CBIR) approach to compute the visual similarity (S_v) of two media items. Given a set of media items at a particular node, we adopt a *bag of visual words* (BoW) framework to create codebook of visual words from discriminative features (descriptors) local to visual keypoints detected within each item.

Scale-invariant keypoints are obtained with the Harris-Laplace point detector [207] and then are described using the SIFT descriptor. Harris-SIFT from all images and

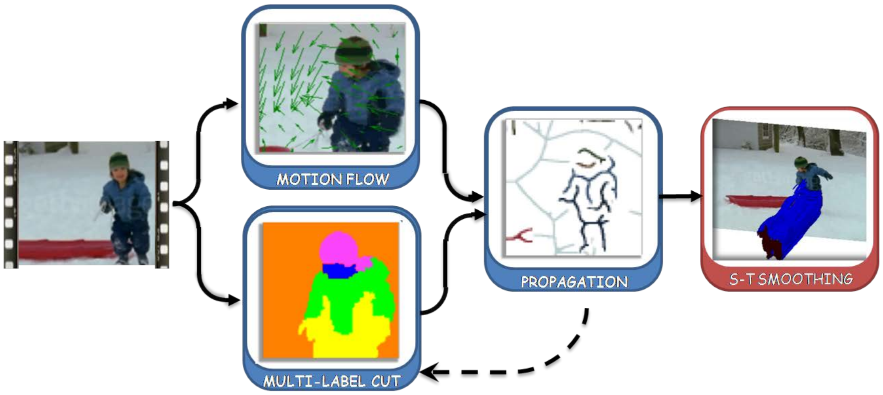


Figure 5.4: Content stylization is underpinned by a graph-cut based segmentation of photos and video frames. In the case of video, region maps from previous frames act as a prior on the graph-cut of successive frames. The skeleton of each region in the previous frame is propagated to the next frame using optical flow, along with an associated GMM colour model, to form as constraints on the graph-cut. New regions are detected via non-conformity with the propagated colour models. The resulting region maps are smoothed via spatio-temporal low-pass filtering (after [44]).

key-frames extracted from video clips are clustered to form a BoW codebook via k-means clustering. A frequency histogram H^I is constructed for each I, indicating the visual words present within that media item. Visual similarity is then computed by measuring the histogram intersections of media pairs:

$$\begin{aligned} S(H^1, H^2) &= \sum_{i=1}^{k}\sum_{j=1}^{k} \omega_{ij} \cdot \min(H^1(i), H^2(j)), \\ \omega_{ij} &= 1 - |\mathcal{H}^1(i) - \mathcal{H}^2(j)|. \end{aligned} \tag{5.3}$$

where $H(i)$ indicates the i^{th} bin of the histogram, $\mathcal{H}(i)$ the normalized visual word corresponding to the i^{th} bin. .

5.4 Video Stylization

We next describe a coherent video segmentation algorithm which performs a multi-label graph cut on successive video frames, using both photometric properties of the current frame and prior information propagated forward from previous frames. This information comprises:

1. an incrementally built Gaussian Mixture Model (GMM) encoding the color distribution of each region over past frames;
2. a subset of pixel-to-region labels from the previous frame.

We check for region under-segmentation (e.g. the appearance of new objects, or objects emerging from occlusion) by comparing the historic and updated GMM color models for each region, and introducing new labels into that region if the color model appears to be temporally inconsistent. The region map of the first frame is boot-strapped using mean-shift segmentation [45], and may *optionally* be modified by the user for aesthetics e.g. to abstract away background detail by merging regions. Figure 5.4 gives an overview of the segmentation algorithm.

We first describe our video segmentation algorithm (sub-Secs 5.4.1-5.4.3) and then describe how the coherent region maps are applied to stylize video (sub-Secs 5.4.4-5.4.5) and create the animations used to transition between successive clips in the DAD sequence.

5.4.1 Multi-label Graph Cut

We formulate segmentation as the problem of assigning region labels existing in frame I_{t-1} to each pixel $p \in \mathcal{P}$ in frame $I_t(p)$; i.e. seeking the best mapping $l : \mathcal{P} \rightarrow \mathcal{L}$ where $\mathcal{L} = (l(1), \ldots, l(p), \ldots, l(|\mathcal{P}|))$ is the set assignments of labels $l_i, i = \{1...L\}$, and $\mathcal{P}$ is an 8-connected lattice of pixels.

A subset of $\mathcal{L}$ are carried forward from the region map at $t-1$, via a propagation process described shortly (sub-Sec. 5.4.2). This *prior labeling* of pixels ($\mathcal{O} \subseteq \mathcal{P}$) forms a hard constraint on the assignments of remaining pixels in I_t, which are labeled to minimize a global energy function encouraging both temporal consistency of color distribution between frames, and spatial homogeneity of contrast within each frame. This is captured by the data and pairwise terms of the Gibbs energy function:

$$E(\mathcal{L}, \Theta, \mathcal{P}) = U(\mathcal{L}, \Theta, \mathcal{P}) + V(\mathcal{L}, \mathcal{P}). \tag{5.4}$$

The data term $U(.)$ exploits the fact that different color homogeneous regions tend to follow different color distributions. This encourages assignment of pixels to the labeled region following the most similar color model (we write the parameters of such models Θ). The data term is defined as:

$$\begin{aligned} U(\mathcal{L}, \Theta, \mathcal{P}) &= \sum_{p \in \mathcal{P}} -\log P_g(I_t(p)|l(p); \Theta). \\ P_g(I(p)|l(p) = l_i; \Theta) &= \sum_{k=1}^{K_i} w_{ik} \mathcal{N}(I(p); \mu_{ik}, \Sigma_{ik}). \end{aligned} \tag{5.5}$$

i.e. the data model of the i^{th} label l_i is represented by a mixture of Gaussians (GMM), with parameters w_{ik}, μ_{ik} and Σ_{ik} representing the weight, the mean and the covariance of the k^{th} component. The parameters of all GMMs ($\Theta = \{w_{ik}, \mu_{ik}, \Sigma_{ik}, i = 1, \ldots, L, k = 1, \ldots, K_i\}$) are learned from historical observations of each region's color distribution (sub-Sec. 5.4.2).

The contrast term $V(.)$ encourages coherence in region labeling and discontinuities to occur at high contrast locations, which is computed using RGB color distance following previous graph cut based methods:

$$V(\mathcal{L}, \mathcal{P}) = \gamma \sum_{(m,n) \in N} [l(m) \neq l(n)] e^{-\beta ||I(m) - I(n)||^2}. \tag{5.6}$$

where N is the set of pairs of 8-connected neighboring pixels in $\mathcal{P}$. β is chosen to be contrast adaptive as in [21]:

$$\beta = \frac{1}{2} \langle ||I(m) - I(n))||^2 \rangle^{-1}. \tag{5.7}$$

Constant γ is a versatile setting for a variety of images [18], and is set empirically to obtain satisfactory segmentation.

Motivated by the data term in [21] we enforce hard constraints on the motion propagated prior labels assigned to label l_i, by setting the data term of $p \in \mathcal{O}$ to be:

$$U_{p:\{p \in \mathcal{O}\}} = \begin{cases} 0 & \text{if } l(p) = l_i; \\ \infty & \text{if } l(p) \neq l_i. \end{cases} \tag{5.8}$$

Optimizing (5.4) to yield an appropriate assignment of labels to pixels is NP-hard, but an approximate solution can be computed by treating the optimization as a multi-label

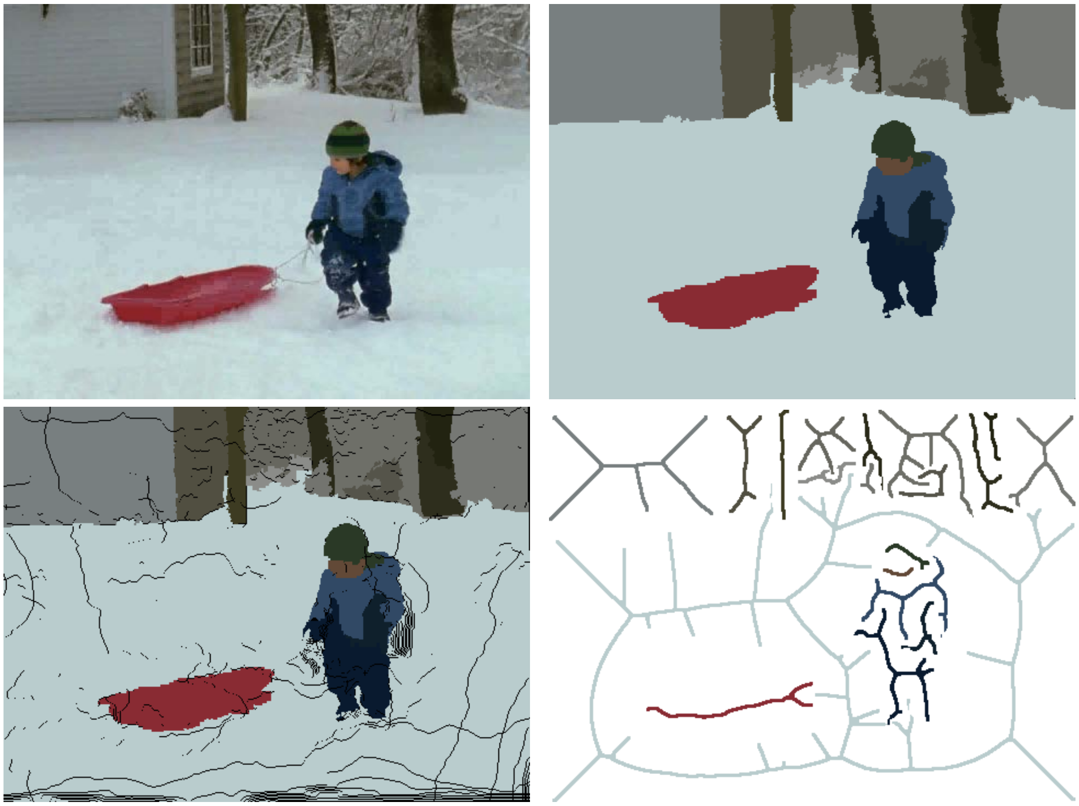


Figure 5.5: Prior propagation: (Top-left) Video frame I_{t-1}; (Top-right) region labeling of I_{t-1} following multi-label graph cut; (Bot.-left) region labels warped according to per-pixel motion flow field $I'_{t-1} \rightarrow I_t$ — for example, note the shift of the boy's left glove. (Bot.-right) Thinning yields prior labels for the segmentation of I_t.

graph cut and solving this using the expansion move algorithm of [24]. An α-expansion iteration is a change of labeling such that p either retains its current value or takes the new label l_α. The expansion move proceeds by cycling the set of labels and performing an α-expansion iteration for each label until (5.4) cannot be decreased [24]. Each α-expansion iteration can be solved exactly by performing a single graph-cut using the min-cut/max-flow as described in [23]. Convergence to a strong local optimum is usually achieved in 3-4 cycles of iterations over our label set. We improve the computation and memory efficiency of each iteration by dynamically reusing the flow at each iteration of the min-cut/max-flow algorithm (after [5]). This results in a speed-up of an order of two.

5.4.2 Region Propagation

The segmentation of I_t described in sub-Sec 5.4.1 is dependent on the information propagated from the previous frame at $t-1$; specifically: i) the color models for regions Θ; ii) the set of pixels $\mathcal{O} \subseteq \mathcal{P}$ and their corresponding label assignments at $t-1$. We now explain the propagation process in detail.

Our approach is to estimate the motion of pixels in frame I_{t-1}, and translate those pixels and their respective label assignments from the previous frame to the current frame (I_t). Motion is estimated using a model of *rigid motion plus deformation*.

We first estimate a global affine transform between successive frames I_{t-1} and I_t, using a RANSAC search based on SIFT features [141] matched between the frames. Performing an affine warp on I_t and the corresponding region map compensates for large rigid (e.g. camera) motion, resulting in a new image I'_{t-1}. Local deformations are captured by estimating smoothed optical flow [17] between I'_{t-1} and I_t, independently within each region. Note that we do not assume or require accurate motion estimation at this stage. Figure 5.5 (bot.-left) provides an example region map from the BOY sequence $t-1$ warped according to motion field $I'_{t-1} \rightarrow I_t$.

We select a subset of the motion propagated pixels (written $\mathcal{O}$), and their corresponding region assignments, as prior labels to influence the segmentation of I_t. To mitigate the impact of imprecise motion estimation, we form $\mathcal{O}$ by sampling from a morphologically thinned skeleton of the motion propagated regions (Figure 5.5, bot.-right). This approach is inspired by the "scribbles" used in the interactive Grab-Cut system of [18], but note that we perform an automatic and multi-region (as opposed to binary) labeling. The skeleton emphasizes geometrical and topological properties of the motion propagated region map, such as its connectivity, topology, length, direction, and width. To further deal with the uncertainties in positions which are closer to the estimated region boundary, we use only the skeletons whose distance to the boundary exceeds a pre-set confidence. Figure 5.5 illustrates the complete process, which we find to be tolerant to moderate misalignments caused by inaccurate motion estimation.

We build a GMM color model for each region l_i, sampling the historical colors of labeled pixels over recent frames. To cope with variations in luminance often present in the

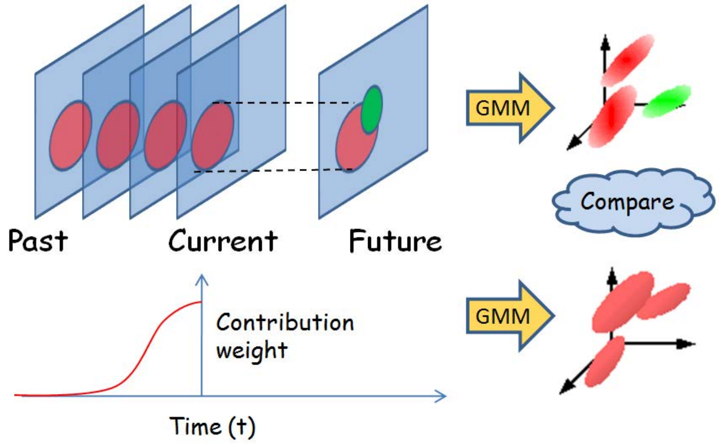


Figure 5.6: A GMM color model of each region is built incrementally over time, with contributions biased toward more recent observations. If the GMM of a region abruptly changes color distribution (χ^2 metric) then the region is re-segmented (sub-Sec 5.4.3).

sequence, the proportion of samples $S_{l_i,t-d} \in [0,1]$ $(d > 0)$ drawn from all l_i-labeled pixels from historical frame I_{t-d} decreases exponentially as the temporal distance d from the current frame I_t increases (Figure 5.6):

$$S_{l,t-d} \propto e^{-d^2/\sigma_d^2}. \tag{5.9}$$

Our system selects a smaller σ_d when luminance variance is large, contributing more recent data to the GMM, otherwise the historical data contributes more to increase robustness.

5.4.3 Refining Region Labels

The method of sub-Sec. 5.4.1 labels I_t with some or all of the region labels in use in the region map at $t-1$. However, new objects may appear in the sequence over time I_t due to occlusion effects of objects moving into shot. This is most apparent in clips such as DRAMA (Fig 5.14). These objects may warrant introduction of a new region label, should they differ in color from existing regions. In such a situation, pixels comprising the object are erroneously labeled from the existing label set by the graph cut optimization, which in turn perturbs the color distribution of the region. We can detect this by measuring the χ^2 distance (as defined in [81]) between the GMM of a region at time t and the historical GMM built over time (Figure 5.6).

For successive frames, we keep two sets of color models for each label l in frame I_t being processed: (1) Historical color models associated with each label $M^h_{l:\{l\in\mathcal{L}\}} := G_l(I_{t-4}, I_{t-3})$ and (2) an updated color model $M^u_{l:\{l\in\mathcal{L}\}} := G_l(I_t)$. We set a guard interval of two frames between those two models to detect a significant change. If the χ^2 distance between these two models exceeds a threshold, new objects are deemed present.

To build color models for the new objects we extract the dominant modes of colors within the region. We apply mean-shift to perform unsupervised clustering on the spatial-color modes (XY+RGB) of pixels in the region. This yields a localized segmentation of pixels in the region. We extend our label set to accommodate each new region arising from the mean-shift segmentation, and for each new region also compute GMM color models and region skeletons as in sub-Sec. 5.4.2. Re-applying the graph cut optimization locally within the region, using these new labels and constraints, yields an improved segmentation for I_t that is carried to successive frames.

5.4.4 Smoothing and Filtering

Our segmentation algorithm produces stable region maps, but due to visual ambiguities in poor contrast areas, the location of region boundaries tend to oscillate in position by a few pixels. We can attenuate this effect by performing spatio-temporal smoothing. Specifically, by coherently labeling regions in adjacent frames, we have formed a set of space-time volumes. Applying a fine scale (3×3×3) Gaussian filter removes boundary noise. We avoid removing detail by only filtering volumes above a certain size.

We inspect the duration $d_{l,k}$ of the disconnected video objects k ($k = 1 \ldots K_{obj_l}$) with the same label l, in a time window of 24 frames (1 second). If the duration of any of these disconnected video object within this time window is shorter than a length

$$D_{l:\{l\in\mathcal{L}\}} = \min\{\max_{k\in\{1\ldots K_{obj_l}\}} d_{l,k}, \tau_r\}. \tag{5.10}$$

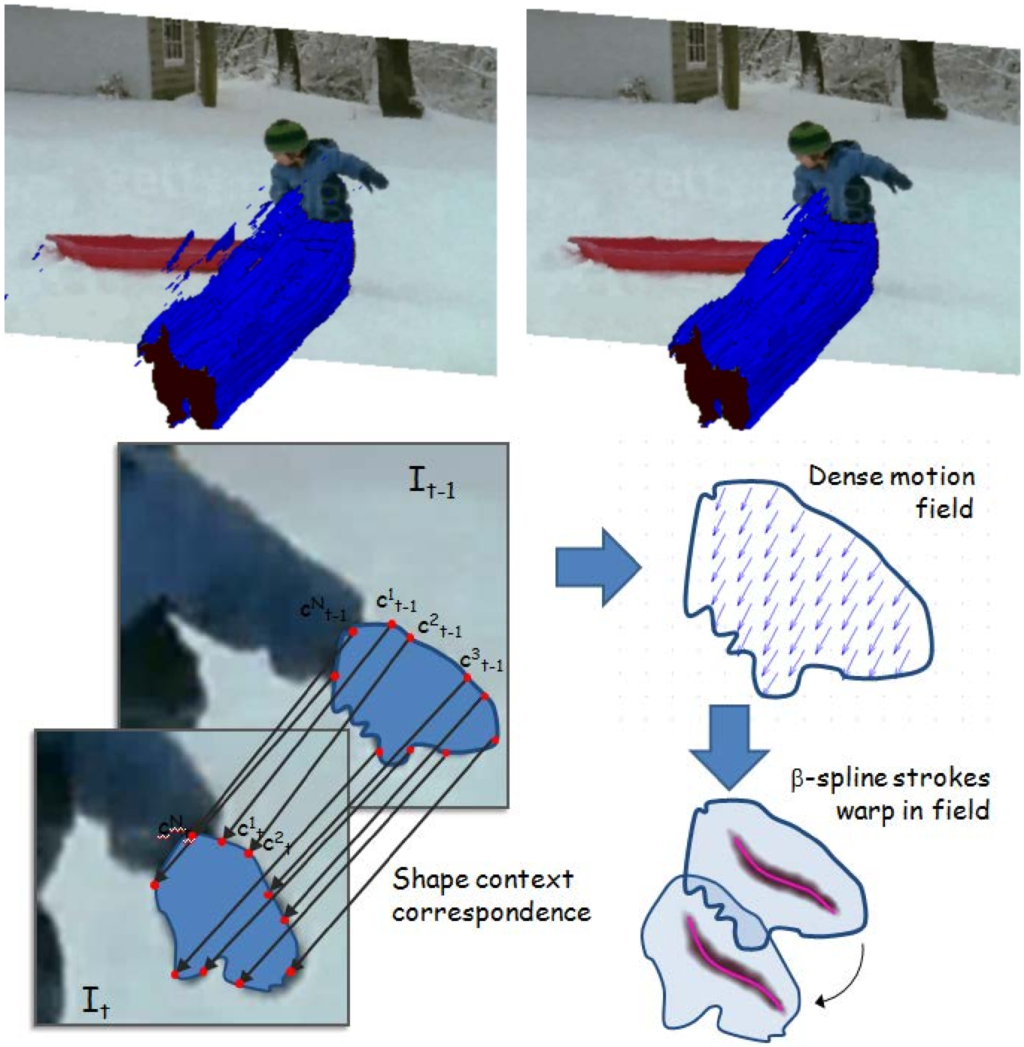


Figure 5.7: Above: co-labeled regions are smoothed in space-time to remove any spurious regions. Below: Brush strokes are painted on a stable reference frame, created by corresponding co-labeled regions in adjacent frames and interpolating a dense motion field.

this video object is removed. τ_r is set to be six frames (about 1/4 second). The effect of this process is that the spurious volumes due to false segmentation and short-lived objects are removed, as shown in Figure 5.7. The "holes" left by filtering and smoothing are filled by extrapolating region labels from immediate space-time neighbors on a nearest-neighbor basis.

5.4.5 Stroke Placement and Shading

Our video segmentation algorithm ensures regions not only deform in a coherent manner, but are also labeled consistently between frames. This space-time description of scene structure may be rendered in a variety of artistic styles; here we give an example of one shading and one stroke based style.

Cartooning

Superimposing black edges over regions shaded with their mean pixel color can produce coherent cartoon effects (Figure 5.11). In our cartoon examples, a mask of inter-region boundaries is produced for each frame. We identify "junction" points on region boundaries by identifying 3×3 pixel windows containing > 2 region labels - and remove the corresponding boundary fragments from the mask. This results in a series of connected pixel chains that we transform into β-spline strokes by sampling knots at equi-distant intervals. The strokes are rendered as dark brush strokes, with thickness proportional to stroke length (after [213]) tapering toward the stroke ends. We render frames independently without further post-processing; this is both for simplicity and to demonstrate the temporal coherence of our segmentation output.

We can also exploit the temporally corresponded region labeling to differentially render regions of interest. For instance, users are particularly sensitive to over-abstraction of detail in faces; commonly present in home video footage. We run human face detection [209] over frames to identify labeled regions likely to contain faces. Internal detail in these regions may be restored by blending in a posterized image of underlying video footage, and detail further enhanced by subtracting a Laplacian of Gaussian (LoG) filtered image from the result.

Painterly Rendering

Alternatively we can paint β-spline brush strokes inside regions, coherently deforming those splines by warping their control points to match the motion of the region boundary (similar to the manually bootstraped rotoscoping system of [2]). Boundary correspondences are computed between temporally adjacent, co-labeled regions using Shape Contexts [16]. The set of N corresponded boundary locations $\{< c^1_{t-1}, c^1_t >, < c^2_{t-1}, c^2_t >, ..., < c^N_{t-1}, c^N_t >\}$ is used to derive the motion vector for a control point p at time t as:

$$p = \frac{1}{N} \sum_{i=1}^{N} \omega(p, c^i_t) |c^i_t - c^i_{t-1}|. \tag{5.11}$$

where $\omega(.)$ is a Gaussian weighted function of the shortest distance between two points within the region (see Figure 5.7, below), and p is one of the control points of brush stroke. Our coherent segmentation promotes smooth deformation of region shape, and so flicker-free motion of brush strokes.

We paint the β-spline strokes within a region using Hertzmann's bi-directional stroke growth algorithm [90]. In the original algorithm, strokes are grown from random seed points using the orientation of an intensity gradient field computed from the underlying image. However, computing such orientation directly from video footage typically promotes incoherence. Instead, we interpolate an orientation field from the shape of the region. Orientations are locally obtained at points of correspondence on the boundary $\theta[x, y] \mapsto \text{atan}(c^{i-1}_t - c^i_t)$. We define a dense orientation field Θ_Ω over all coordinates within the region $\Omega \in \Re^2$, minimizing:

$$\underset{\Theta}{\text{argmin}} \int \int_\Omega (\bigtriangledown\Theta - \theta)^2 \quad s.t. \;\; \Theta|_{\delta\Omega} = \theta|_{\delta\Omega}. \tag{5.12}$$

i.e. $\triangle\Theta = 0$ over Ω s.t. $\Theta|_{\delta\Omega} = \theta|_{\delta\Omega}$ for which a discrete solution was presented in [164] solving Poisson's equation with Dirichlet boundary conditions. Examples of painterly output are given in Figure 5.8.

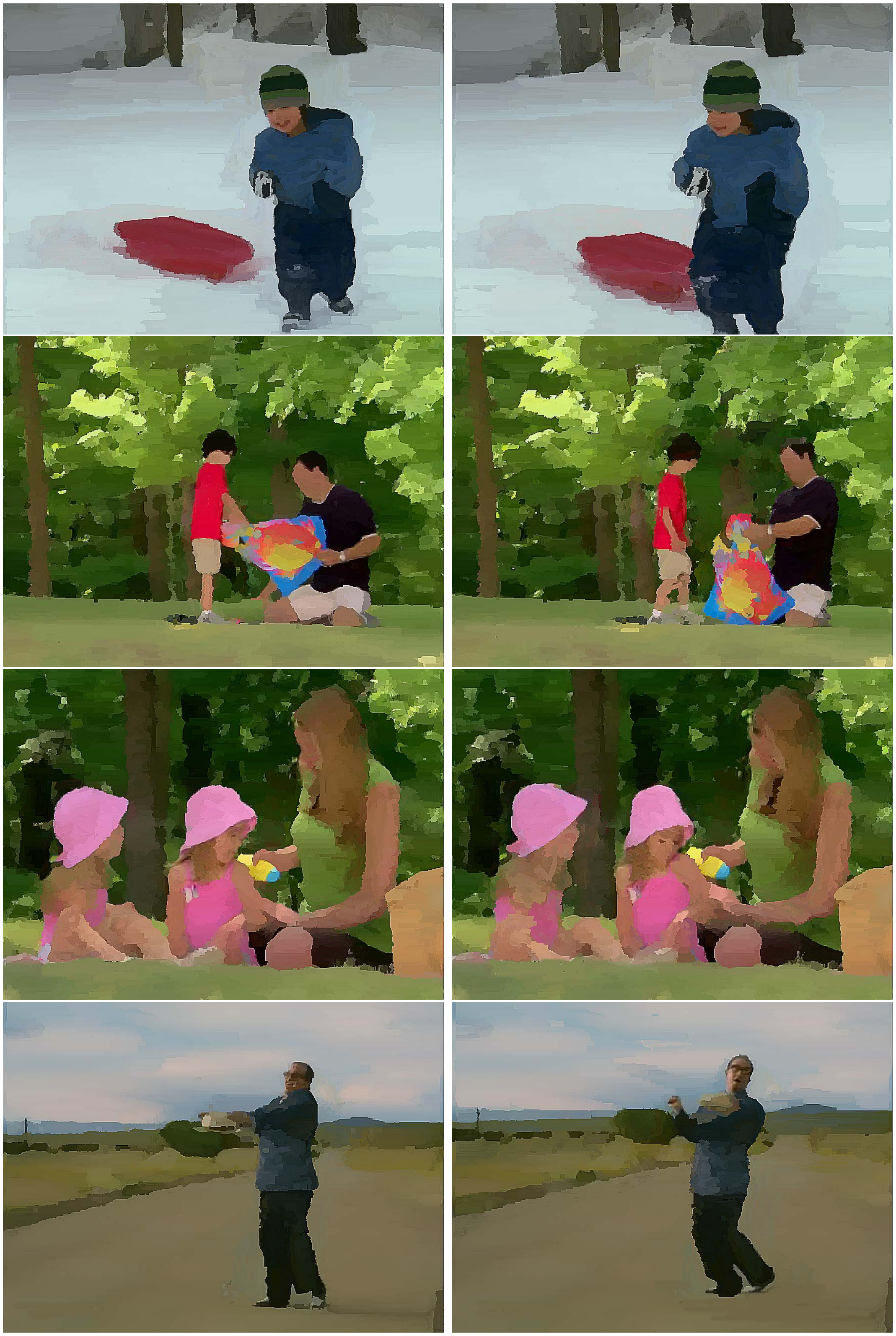


Figure 5.8: Examples of coherent painterly renderings produced from the BOY, KITE, PICNIC and DANCE videos (top to bottom).

5.5 Content Sequencing

Finally, we explain the algorithms for sequencing stylized content to create the temporal composition of media items from the user's collection, and for creating the animated transitions between displayed media items.

5.5.1 Temporal Composition

We desire the DAD to autonomously transition between a sparse yet diverse sample of user content to present a summary of the collection. This is achieved using the *hierarchical representation* of the media collection constructed during pre-processing (Section 5.3).

Recall each node in our representation encodes a *cluster* of similar media; defined using either a semantic similarity measure (toward the root) or a visual similarity measure (toward the leaves). For each node in the hierarchy, similarities between all media items within the cluster are computed (Section 5.3.1), to be form a new graph of media items — with edge weights indicating the (dis-)similarity of a pair of items. Computing the shortest Hamilton cycle within this graph creates a non-repetitive set of transitions maximizing the similarity between successive media items and thus the coherence of the sequence. Although a precise solution maps to the classical NP-hard ”traveling salesman problem” (TSP), an approximate solution can be found quickly using heuristic search methods. We adopt a Genetic Algorithm (GA) based solution [69]. In practice TSP paths for each cluster are also computed during pre-processing.

The DAD is equipped with a camera and a face detection technology [209] to detect user gaze (attention) directed towards the display (Figure 5.12). Sequencing proceeds in one of two modes, depending on whether user attention is present or not.

When attention is not present, the intention is to create a succinct and diverse summary of the collection. The intention is to speculatively display content that might catch a passing user's interest. Clusters in the first level of the tree represent coarse semantic categories across the whole collection. Displaying a sample of media from each cluster in turn yields a high-level summary of the collection. However, rather than present a random succession of media, we desire a degree of temporal coherence in our choice of media to create compelling paths through the collection — to tell a ”story” through visual media, with the aim of prompting user reminiscence. Although TSP paths within a cluster offer coherence for intra-cluster transitions, they do not offer inter-cluster coherence. We address the latter by identifying a ”semantic route” between media items (A, B) in different clusters. This is achieved by taking the union of media clusters containing A and B, and computing a (dis-)similarity graph as before. Dijsktra’s algorithm yields a shortest path between A and B, encoding the ”semantic route”; the most coherent sequence of media items to transition between A and B (Figure 5.9).

When attention is present, we do not permit jumps between siblings, but instead permit transitions to the parent or children of the current cluster. These transitions represent ‘generalization’ or ‘drilling down’ into a media topic, respectively. Suppose we have detected user interest in a media item A. We stochastically choose to either remain on the TSP path containing A in the current cluster, or to transition to a child cluster also containing A (i.e. begin transitioning along the TSP path in that cluster). For our experiments the probability of continuing on the current path, or ‘drilling down’ is even. When interest in the display abates, we transition back ‘up’ the tree in a similar manner; with even choice between continuing on the TSP path or jumping to the parent TSP path. To ensure smooth transitions when transitioning up and down the tree, the semantic route mechanism is again used to create an interim sequence between source and destination media items.

5.5.2 Rendering Transitions

Having established a sequencing mechanism during visualization, we animate the transition between stylized media items according to the scene structure (region map).

We first establish a mapping between each region R^j_{t-1} and R^i_t corresponding, respectively, to the final and initial frames of the two clips (recall that images are accommodated in our framework as single frame videos). The region mapping is created a greedy manner,

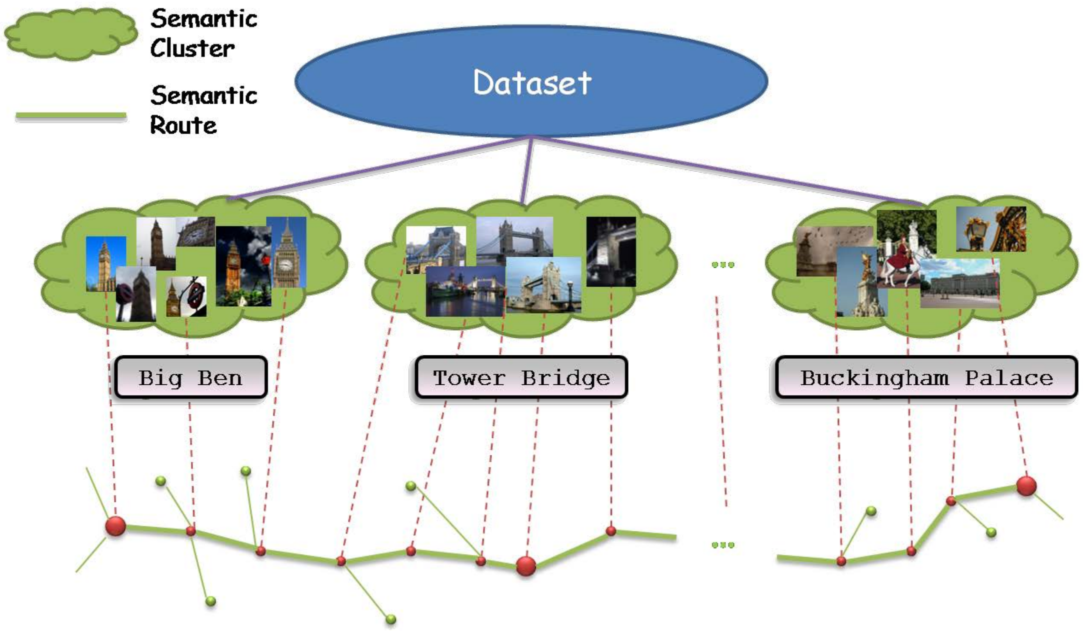


Figure 5.9: Transitions are made across top-level clusters by sequencing display of content along 'semantic route' between a source and destination media item (depicted as large red nodes). The semantic route is the shortest path computed across the graph; here nodes part of that route are shaded red, otherwise green.

iteratively pairing off regions that minimize:

$$\underset{\{i,j\}}{\operatorname{argmin}} \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} C(R^j_{t-1}, R^i_t) \\ A(R^j_{t-1}, R^i_t) \\ S(R^j_{t-1}, R^i_t) \end{bmatrix} \quad (5.13)$$

where the normalized functions: $C(.)$ indicates mean color similarity; $A(.)$ indicates relative area; $S(.)$ indicates shape similarity in terms of region compactness, which is the ratio between perimeter and area. We bias weights ω_{1-3} empirically to 0.5, 0.4, 0.1. The greedy assignment continues until (5.13) falls below a threshold. Unassigned regions in the mapping are animated to "disappear" (shrink to a point at the centroid) or "appear" (grow from the centroid); whereas regions mapped between frames are animated to morph into one another.

Regions are morphed using simple linear blending. Each region is vectorized into a polygon and a series of regularly spaced control vertices established on the boundary.

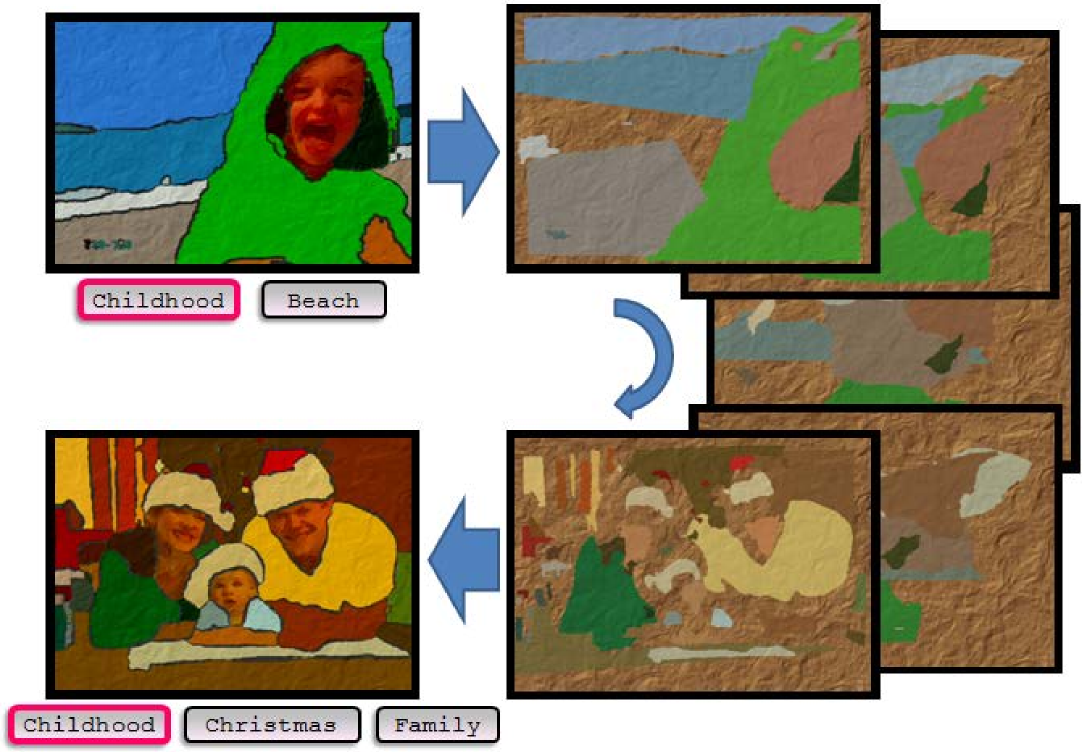


Figure 5.10: Frames from the transition animation between two clips.

A correspondence is established between vertices of R^j_{t-1} and R^i_t to minimize distance between corresponded vertices. The position of control vertices are linearly blended over time (typically $\frac{1}{4}$ second) to animate the region from one shape to another. Region color is similarly blended. Although more complex vertex correspondence approaches were investigated [16], these lacked stability when presented with moderate changes in region shape. The resulting transitions are shown in Figure 5.10.

5.6 Results and User Study

We present a qualitative comparison of the proposed video segmentation algorithm with two existing techniques [45, 161] and present a gallery of stills from videos stylized into cartoons and paintings. We also present a small-scale study exploring user engagement with the DAD.

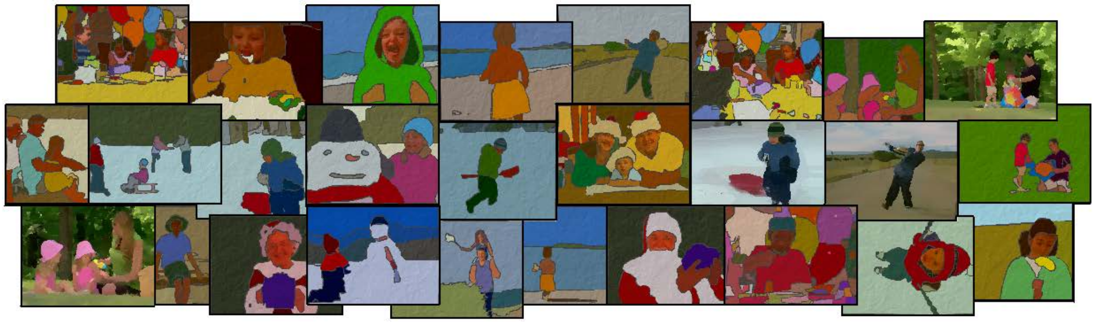


Figure 5.11: A collage of stylized frames sampled from the user video collection studied in this thesis.

5.6.1 Video Segmentation and Stylization

To demonstrate the advantages of the proposed multi-label video segmentation algorithm, we compare the approach proposed in Section 5.4 to two leading segmentation methods for per-frame [45] and spatio-temporal [161] segmentation (Figure 5.13). We observe the region boundaries in the proposed method to exhibit improved stability over time. Figure 5.14 indicates the region maps produced by the segmentation algorithm over four video sequences. We test our algorithm on fast moving footage containing small objects ("BEAR" from [44]). Unlike previous work, fine scale features (e.g. the bear's eyes and nose) are retained. Similarly, "DANCE" demonstrates the ability to cope with fast motion and partial occlusions. "DRAMA" shows correct handling of regions that disappear and appear within sequences, the latter detected by changes in the region color distribution and addressed as out-lined in sub-Sec 5.4.3. The "KITE" sequences shows the aesthetic ability to selectively abstract detail (trees) from the stylized video, when interactively removed by the user in the initial frame. In all cases our segmentations appear flicker-free; some flicker is occasionally present the bottom-left of clips due to the frame identifier which could be manually abstracted away by modifying the initial frame in a similar way.

We demonstrate the video stylization and transition animations using a collection of 23 videos. Figure 5.11 shows representative frames of the stylized footage in both cartoon and painterly styles; 6 minutes of the perpetual animated display is also included in the supplementary material. An example transition animation given in Figure 5.10.

The resulting clip transition animations match large, similarly colored regions between frames producing a pleasing smooth transition effect evident throughout the DAD sequence.

Following coherent segmentation, Figs. 5.8 shows frames of painterly renderings over four video sequences in natural scenes. The smooth deformation of regions enables stable and flicker-free motion of brush strokes, which produces an aesthetically pleasing painterly effect over the input video sequences.

5.6.2 Study of User Engagement

We evaluated the efficacy of our content sequencing algorithm (Section 5.5.1) in a small-scale user study. Our hypothesis was that the DAD media sequencing algorithm would prove more engaging for users than simple random slideshows, as typified by existing commercial digital photo frames. We tested the algorithm with and without adaptation to user interest. We measured user engagement using the DAD's gaze detection; counting the proportion of displayed media items that attracted user attention. In this evaluation we used photographic media only, rendered in a painterly style (Section 5.4.5). We eschewed video content to potential bias introduced by movement which can act as a strong attractor of attention. Transition animations were also disabled.

Experimental Setup

The media collection comprised 600 user-tagged Flickr images of eighteen landmarks in London, licensed under the Creative Commons. We de-noised the associated tags by stripping numeric tags, any punctuation and commonly used Flickr tags that do not relate to content e.g. camera model. The semantic relevance between the tags was pre-computed, and images analyzed to form a BoW code-book with 4000 visual words. The study comprised ten participants between the ages of 20-40, of mixed gender, with varying levels of technical expertise. The DAD device was positioned within proximity to the participant in their everyday working environment, e.g. on their desks. The camera and face detector were calibrated to record an attention event when the user's head is oriented towards the DAD display.

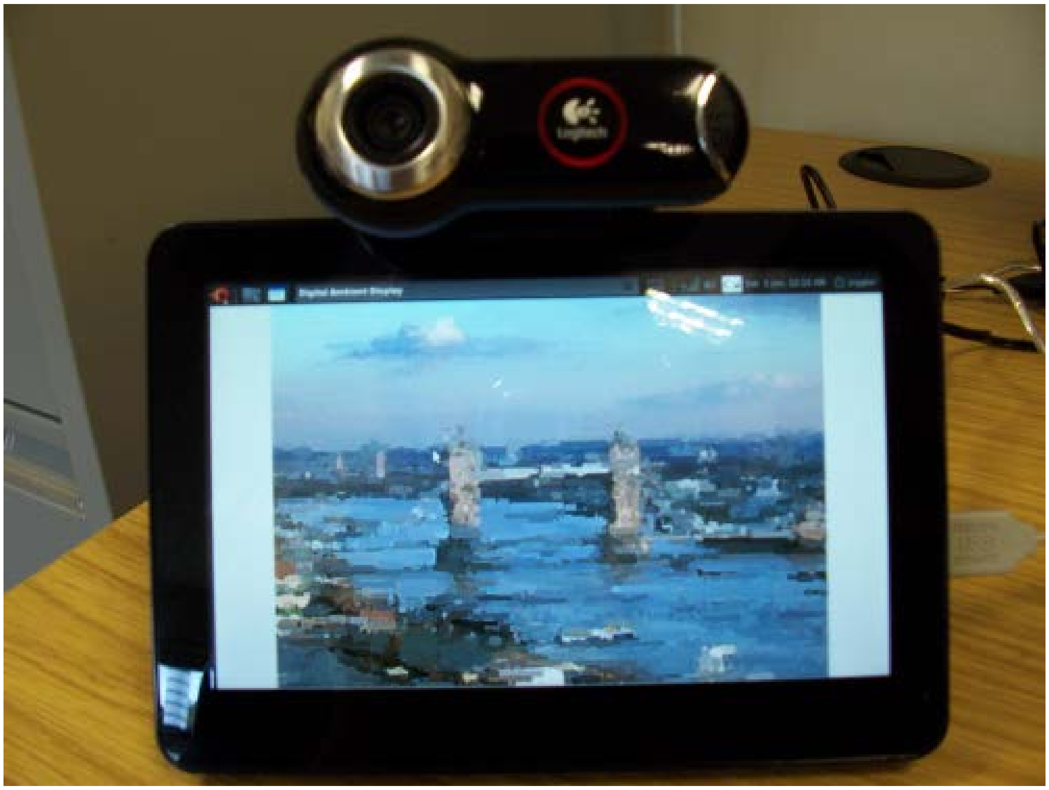


Figure 5.12: One of three identical DAD devices used in the experiment. The DAD adapts in real time to display interesting contents at various levels of detail in response to user attention level.

Protocol

The experiment comparatively evaluates three operational modes of the DAD device:

- Random and passive display (RP): the DAD randomly selects non-repetitive images from the dataset to display.
- Structured and passive display (SP): the DAD displays images in the proposed approach without responding to user attention.
- Structured and adaptive display (SA): the DAD displays images in the proposed approach.

Participants might be involved in parallel activities while seated during the experiments and only pay extended attention to the screen when attracted by the content. In order to

Table 5.1: Proportion of images attracting user attention out of total displayed images in each scheme

User	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
RP	0.047	0.113	0.140	0.105	0.075	0.135	0.085	0.037	0.030	0.143
SP	0.085	0.178	0.203	0.078	0.130	0.177	0.148	0.083	0.095	0.238
SA	0.108	0.233	0.175	0.102	0.178	0.200	0.180	0.125	0.127	0.225

mitigate the effect of short term memory of the image collection we shuffle the order in which the three modes are evaluated by each user. Furthermore, during the evaluation, the participants were unaware of the nature of the three modes or the interactive nature of mode SA. All user attention events during the three presentations for each user are recorded in the background for analysis. The experiment of each presentation lasts one hour which is empirically determined considering the size of the image collection.

Experimental Results and Feedback

Attention events recorded by the DAD for the three operational modes are given in Table 5.1. The table records the proportion of images that attracted user attention out of the total images displayed for each mode. A paired t-test between pairings of modes indicate strong statistical significance between each mode of operation (Table 5.2). This suggests a qualitative improvement in level of user engagement using our structured sequencing approach, versus a random slideshow. Improvement is also observed with user adaptive sequencing (SA) over non-adaptive (SP). Quantifying this improvement by averaging across the users the SA mode we record $\sim$ 17% more attended images then the SP scheme, which in turn recorded $\sim$55% more attended images than the RP scheme. Data on how user attention was distributed across the semantic clusters in the SA case shows large differences between the clusters, consistent with the adaption strategy adopted i.e. the participant's initial interest in a specific category is detected and draws more images from that category that serve to maintain that interest. We thus conclude that out proposed approach offers an engaging means to display the contents of a large media collection with minimal user interaction.

Table 5.2: Significance of results (paired t-test)

t-test	Means	Std. Devs.	p-value
RP/SP	0.0910 / 0.1417	0.0431 / 0.0565	0.0007
RP/SA	0.0910 / 0.1653	0.0431 / 0.0474	0.0001
SA/SP	0.1653 / 0.1417	0.0474 / 0.0565	0.0184

During a questionnaire based de-brief, user feedback on the DAD was broadly positive, both in terms of the aesthetics of the painterly rendering and the DAD system. Participants feel engaged with the system, remarking on the structured and adaptive manner of presentation. 80% of the participants find that the DAD displays more images of interest in our proposed approach (SA) than in the other two approaches and regard it as a useful means to display their own digital media collections. 70% of the participants deem that our proposed approach (SA) presents an effective global summary of the structure of the collection. 60% of the participants would consider presenting their own digital media collections in a similar painting style, with the remainder concerned about the recognition of faces in the stylized content. 60% of participants would like to be able to take over control of the presentation. Suggested controls are: (1) Ban or skip a specific category (2) Hold on the content being displayed (3) Alternative artistic rendering styles (4) Indication of user attention being detected. All participants are satisfied with the hardware specifications of the DAD device, such as the appearance, screen size, screen brightness, and speed. Participants filled in the subjective questionnaire without knowing the DAD mode they were commenting on.

5.7 Conclusion

We have presented a *Digital Ambient Display* (DAD) that harnesses artistic stylization to create an abstraction of user's experiences through their home digital media collections. The DAD automatically selects, stylizes and transitions between media contents enabling users to passively or actively consume their digital media collections and rediscover past memories.

We contributed a novel algorithm for coherent video segmentation based on multi-label graph cut, and applied this algorithm to stylized animation in the DAD. By parsing the video into coherent spatial segments, we are able to represent scene structure. This representation allows us to establish correspondence between frames, enabling the coherent stylization of video objects with both shading and painterly effects. The latter was possible by painting brush strokes on a smoothly deforming reference frame defined by the regions. We are also able to create aesthetically pleasing transition effects between different video clips using region correspondence. Video segmentation could be further enhanced by exploring the backward propagation of region labels to further improve coherence of segmentation. We would also like to improve the painterly rendering by differentiating between region motion caused by occlusion vs. object deformation, to more closely align the movement of painted strokes to the perceived structure in the scene.

A further contribution of the thesis is a novel approach to structuring and navigating visual media collections. We described an algorithm for adaptively sequencing media items using graph optimization in a coarse-to-fine manner driven by user attention. By recursively clustering media items into a hierarchy, we were able to plan routes within clusters to display content of a common theme. We were also able to plan routes between clusters to summarise media within the collection. We deployed our system on dedicated hardware and undertook a small-scale user trial to validate the our content sequencing algorithm, which was shown to be more engaging than random photo slideshows.

In future work we would like to offer more control to the user over presentation. An improved interface might enable users to ban or skip specific categories they are less interested in, and hold on interesting contents for closer inspection. In addition to global visual similarity of media items it might be interesting to harness recent developments in image cosegmentation [100, 85] to enable users to explore 'similar content' within a region of interest indicated by touching a particular area on the display. In the subsequent chapter, we refine the hard constraints on segmentation proposed in this chapter to develop a more robust segmentation scheme.

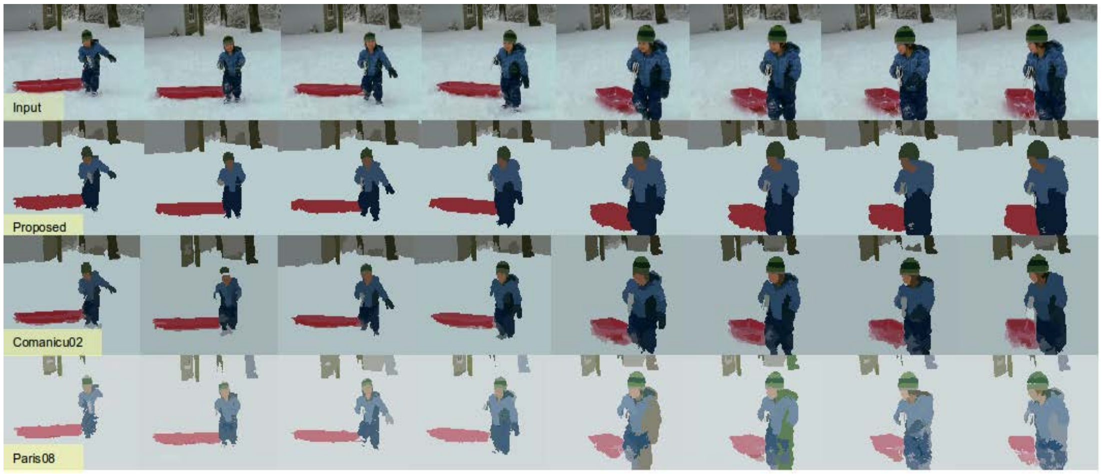


Figure 5.13: Comparing the accuracy and coherence of our segmentation algorithm on the BOY sequence, to 'synergistic' mean-shift + edge (Comaniciu, 2002) and a state of the art spatio-temporal method (Paris, 2008). Boundaries are less prone to variation in shape and topology.

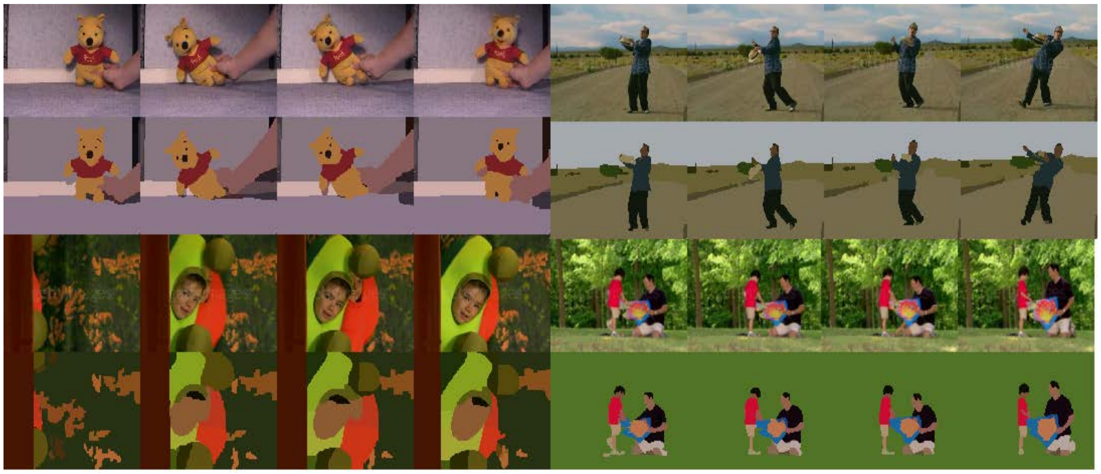


Figure 5.14: Illustrating the coherent region maps produced by our segmentation method. Top: BEAR and DANCE contain small regions moving quickly over time. Bottom: The DRAMA sequence shows correct handling of of regions appearance. The KITE sequence indicates how background detail may (optionally) be abstracted by modifying the initial frame segmentation to merge unwanted detailed regions.

Chapter 6

Probabilistic Motion Diffusion of Labelling Priors for Coherent Video Segmentation

In this chapter we advance the video segmentation algorithm proposed in Chapter 5 by improving the motion propagation model and spatial coherence; estimating the flow via a novel probabilistic motion diffusion model, and combining the per-frame estimates of super-pixel boundaries. This algorithm significantly improves the spatial-temporal coherence and robustness on sequences including these exhibiting clutter and agile motion.

6.1 Introduction

Video segmentation aims to partition pixels into spatio-temporal groups exhibiting coherence and consistency in both appearance and motion. Stable and accurate video segmentation is fundamental to many multimedia tasks, such as video summarisation [71], content based retrieval [43], matteing [3] and video stylisation [219].

A key challenge is the production of *temporally coherent* segmentations; regions whose shape and neighbourhood topology evolve smoothly over time whilst tracking the underlying video content. Although recent years have delivered significant advances, coherent segmentation remains challenging for real-world video of even moderate complexity. Changes in illumination, viewpoint, and occlusion relationships introduce ambiguities that in turn induce instability in boundaries and the potential for localized under- or over-segmentation. Temporal correlation between consecutive frames via motion estimation (e.g. optical flow) can alleviate these difficulties, however inter-frame motion estimation is often inaccurate introducing further ambiguity to the process. Given the approximate nature of boundary and motion estimation, it is natural to formulate these motion ambiguities in a probabilistic framework.

This thesis contributes a novel video segmentation algorithm, in which the segmentation of each frame is guided by motion-flow propagated label priors from previous frames, where flow is estimated via a new probabilistic motion diffusion model. Our approach builds upon the success of multi-label graph-cut approaches to image and video segmentation. The core novel contributions are our motion propagation model, and the combination of this propagated prior information with per-frame estimates of super-pixel boundaries; a growing trend in the image segmentation literature [84, 172, 6, 108, 107].

In contrast to previous techniques based on flow vectors, our diffusion model produces a new probabilistic motion estimate modelling the *distribution of motion vectors for each pixel.* This distribution guides the diffusion of information from pixel labelling in prior frames, to influence segmentation of the current frame. To decide the segmentation of a given frame, we incorporate not only motion propagated soft labelling constraints at the pixel-level but also propose a soft higher-order constraint by imposing label consistency within image regions (super-pixels [45, 174]) obtained via several unsupervised segmentations of the frame (e.g. mean-shift). These resemble the form of unary potentials commonly used in pairwise conditional random fields (CRFs) for different image labelling problems [22, 163]. This formulation enables the use of powerful graph cut based move making algorithms for performing inference in the framework. By enforcing labelling consistency in this way, we show inaccuracies in boundaries and region over-segmentation to be alleviated. We quantify this improvement through comparison to three state of the art methods; a spatio-temporal method [161] and a "hard" CRF-based motion propagation method that relies upon a single flow vector for each pixel rather than our novel “soft” motion diffusion, and recent graph based video segmentation method based on dense optical flow propagation [76].

We describe our proposed video segmentation algorithm in Section 6.2, presenting the motion diffusion model for propagation of labelling priors in Section 6.3 and describing the supporting energy terms for the CRF in Section 6.4. We evaluate our approach over several challenging video clips exhibiting clutter and agile motion, adopting the methodology of the Berkeley Segmentation Benchmark [144] to provide a quantitative comparative evaluation to state-of-the-art techniques (Section 6.5). We show our approach to be quantitatively closer to manually annotated ground-truth segmentations of our footage, and release these results at `http://personal.ee.surrey.ac.uk/Personal/Tinghuai.Wang/TMM2011`.

6.2 Preliminaries

Consider a discrete random field consisting of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ without loop edges, a finite set $\mathcal{L} = \{l_1, l_2, \ldots, l_L\}$ of labels, and a probability distribution P on the space $\mathcal{X}$ of label assignments. $\mathrm{x} \in \mathcal{X}$ is a map that assigns to each vertex v a label x_v in $\mathcal{L}$. Let N_v denote the set of neighbours $\{u \in \mathcal{V} | (u, v) \in E\}$ of vertex v. A clique c is a set of vertices in $\mathcal{G}$ in which every vertex has an edge to every other vertex. A random field is said to be Markov if and only if it satisfies the relation property: $P(\mathrm{x}) > 0 \ \forall \mathrm{x} \in \mathcal{L}$, and the Markovian property:

$$P(x_v | x_{\mathcal{V} \setminus v}) = P(x_v | x_{N_v}). \tag{6.1}$$

This property states that the assignment of a label to a vertex is conditionally dependent on the assignment to other vertices only through its neighbours.

An energy function $E: \ \mathcal{L} \rightarrow \mathbb{R}$ maps any labelling $\mathrm{x} \in \mathcal{L}$ to a real number $E(\mathrm{x})$ called its energy. Energy functions are formed as the negative logarithm of the posterior probability distribution of the label assignment. Minimising the energy function is equivalent to maximise the posterior probability. The maximum a posteriori probability (MAP) x^* of a random field is defined as

$$\mathrm{x}^* = \mathrm{argmin}_{\mathrm{x} \in \mathcal{L}} E(\mathrm{x}). \tag{6.2}$$

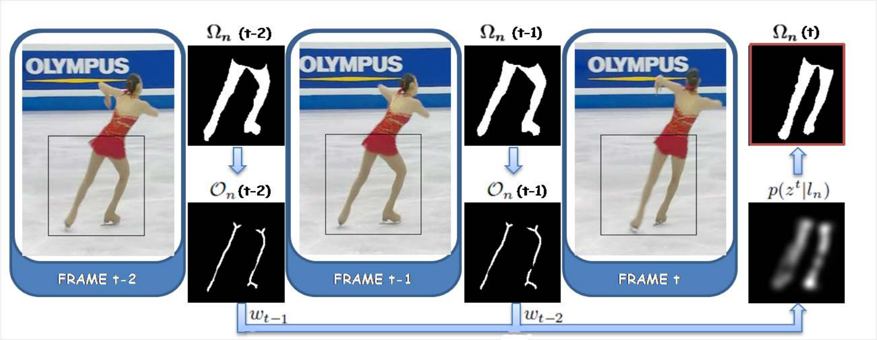


Figure 6.1: Illustration of our motion diffusion process over two frames of "YUNAKIM". A subset of pixels $\mathcal{O}_n(t-s)$ from each region $\Omega_n(t-s)$ in each frame I_{t-s} is propagated to frame I_t based on motion estimation and diffused to its close vicinity following a Gaussian distribution. Labelling prior probability $p(z^t|l_n)$ is formulated as the merged diffusion probability from previous frame I_{t-s} by weight w_{t-s}.

The posterior distribution over the labellings of the conditional random field is a Gibbs distribution and the corresponding Gibbs energy is given by

$$E(\mathrm{x}) = \sum_{c \in \mathcal{C}} \psi_c(\mathrm{x}_c). \tag{6.3}$$

where $\mathcal{C}$ is the set of all cliques [121], and $\psi_c(x_c)$ is known as the potential function of the clique c and $\mathrm{x}_c = \{x_i, i \in c\}$.

6.2.1 Segmentation Framework

We formulate video segmentation as a pixel-labelling problem of assigning each pixel $i \in \mathcal{V}$ in frame I_t with a value from the existing label set $\mathcal{L}$ in frame I_{t-1}, as in Sec. 5.4.

After a propagation process (described in Sec. 6.3) which carries forward a subset of $\mathcal{L}$ from the region map at $t-1$, each pixel in frame I_t bears a set of *prior* probabilities of observing a pixel propagated from different label regions in frame I_{t-1}. The *prior* labelling probabilities of pixels form a soft constraint on the assignments of pixels in I_t, which are labelled to minimize a global energy function. This energy function is adapted from the Gibbs energy function typically used in computer vision and consists of unary, pairwise and higher order cliques as:

$$E(x) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j) + \sum_{i \in \mathcal{V}} \sum_{c \in \mathcal{S}} \psi_c(x_i). \tag{6.4}$$

where $\mathcal{V}$ corresponds to the set of all pixels in frame I_t, $\mathcal{S}$ represents the set of superpixels from over-segmentations (sub-Sec. 6.4.3). This energy function encourages both temporal consistency of appearance between frames, and spatial homogeneity of contrast within each frame. Moreover, it incorporates a third potential partly enforcing the label consistency inside the regions generated by unsupervised image segmentation algorithms. We describe in detail how each of these potentials are defined, and the optimization of (Eq. 6.4) in Sec. 6.4, but first describe the process by which labels are propagated over time in our framework.

6.3 Label Diffusion for Coherent Segmentation

We introduce a motion diffusion model which combines motion estimates made over several time intervals (*frames*) under a probabilistic framework, and accounts for the estimation errors by adaptively refining the internal parameters of this framework. The purpose of the motion diffusion model is to propagate forward the labels of past frames — so forming a distribution of priors for segmentation of the current frame. We bootstrap the first frame of segmentation using mean-shift [45], with a bias to over-segmentation that is resolved via merging of regions due to their similar appearance in subsequent frames.

6.3.1 Single-Frame Motion Diffusion

We first compute the SIFT flow [138] from frame I_{t-1} to I_t. We choose SIFT flow as it is more robust than optical flow in case of large displacement or appearance variation between adjacent frames. The SIFT flow consists of matching densely sampled SIFT features between the two images, while preserving spatial discontinuities. The use of SIFT features allows robust matching across different scene/object appearances and the discontinuity-preserving spatial model allows matching of objects located at different parts of the scene. Although there some discontinuities in the flow field caused by matching errors, we do not assume or require accurate motion estimation at this stage. Indeed our motion diffusion framework is proposed on the assumption that there will be inaccuracies.

Let Ω_n be a region of interest in frame I_{t-1} labelled as l_n. Propagating the whole region to the successive frame I_t by SIFT flow often involves erroneous estimation, especially in positions close to boundary. We only select a subset of pixels $\mathcal{O}_n \subset \Omega_n$ for propagation (Fig. 6.1). To account for the impact from imprecise motion estimation close to boundary, we form $\mathcal{O}_n$ by sampling from a morphologically dilated skeleton of each region. The skeleton preserves geometrical and topological properties of the region. To further deal with the uncertainties in positions which are close to the region boundary, we use only the skeletons whose distance to the boundary exceeds a confidence, measured by a distance transform. A skeleton based propagation scheme was first proposed in [220] for similar reasons. However rather than propagating each label using just one flow vector from a single frame [167, 220], our approach diffuses labels across a distribution of directions (derived from multiple frames, subsec. 6.3.2) as we now explain.

$\mathcal{O}_n$ contains pixels J_k^{t-1} $(k = 1, 2, 3, \cdots, |\mathcal{O}_n|)$, where $|\mathcal{O}_n|$ is the cardinality of $\mathcal{O}_n$. The position of each pixel is denoted as z_k^{t-1}. For each pixel $J_k^{t-1} \in \mathcal{O}_n$ we predict its position z_k^t in frame I_t based on the motion vector from SIFT flow. As a perfect motion estimation is not available, the proposed model only assumes the motion estimation to be probabilistic. The *diffusion process* diffuses the propagated subset of pixels to close vicinity, treating the predicted position as the center of a Gaussian distribution,

$$p(z^t|J_k^{t-1}) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp(-\frac{||z^t - z_k^t||^2}{2\sigma_k^2}). \tag{6.5}$$

where z^t is a position in frame I_t. The variance σ_k reflects the error in motion estimation which is adaptively set for each pixel J_k^{t-1}. For motion estimation which is likely to contain large prediction errors, we set σ_k to large values.

Although Gaussian diffusion is frequently used to model uncertainty in tracking it has been explored only recently in the context of interactive video segmentation, for binary matteing [12]. The key to the robustness of our new multi-label diffusion approach is to propagate only a subset of pixels in regions to account for the imprecise motion estimates close to boundaries typically observed during our early experiments. Furthermore, using local motion coherence to encode the motion estimation error (as opposed to a global measurement of motion alignment error [12]) accommodates the per-pixel local motion estimation errors (σ_k) that can not necessarily be reflected by a global measurement or single propagation.

We now explain how to determine σ_k. The error in estimating the motion of a region of interest often causes discontinuities in the flow field. Such discontinuities are often referred to as *motion non-coherence*. A small portion of an moving object with rigid shape in a sequence often exhibits coherent motions. We correlate the prediction error with local motion non-coherence. For each pixel J_k^{t-1}, we consider the motion vectors in a 5×5 window centred at J_k^{t-1}. All motion vectors within this window are firstly quantized as N angles $\frac{2\pi}{N}$, $\frac{4\pi}{N}, \cdots\ 2\pi$. A quantized motion vector histogram h_k^{t-1} is computed across the local motion vectors. We define a motion coherence factor M_k^{t-1} by measuring the entropy of h_k^{t-1},

$$M_k^{t-1} = \min\{1, \frac{\log(N)}{-\sum_{i=1}^{N} H_k^{t-1}(i)\log(H_k^{t-1}(i))}\}. \tag{6.6}$$

where

$$H_k^{t-1}(i) = \frac{h_k^{t-1}(i)}{\sum_{i=1}^{N} h_k^{t-1}(i)}. \tag{6.7}$$

In information theory, entropy is a measure of the uncertainty associated with a random variable. Higher entropy of h_k^{t-1} indicates lower local motion coherence in the window, and thus smaller M_k^{t-1}. σ_k is computed as

$$\sigma_k = \theta_\gamma \exp(\theta_\mu M_k^{t-1}). \tag{6.8}$$

where θ_γ and θ_μ are constant parameters.

The probability of observing a pixel propagated from $\mathcal{O}_n$ (labelled as l_n) at location z^t on frame I_t is

$$p^{t-1}(z^t|\mathcal{O}_n) = \sum_{k=1}^{|\mathcal{O}_n|} p(J_k^{t-1})p(z^t|J_k^{t-1}). \tag{6.9}$$

where $p(J_k^{t-1}) = 1/|\mathcal{O}_n|$, assuming equal priors for every pixel in $\mathcal{O}_n$. As motions of l_n-labelled pixels are predicted based on $\mathcal{O}_n$, $p^{t-1}(z^t|\mathcal{O}_n)$ can be approximated as the labelling *prior* probability of label l_n at pixel z^t, i.e. $p^{t-1}(z^t|l_n)$.

6.3.2 Multi-Frame Motion Diffusion

We build a single-frame probabilistic motion diffusion model in Section 6.3.1 taking into account the estimation errors. As we later show, our diffusion model greatly enhances the coherence of skeleton based motion propagation [220] during occlusion and rapid movement. However, gross SIFT matching errors occasionally occur and may result in amplified errors in the propagation process.

To mitigate gross prediction errors, we adopt a multi-frame fusion scheme. We perform single-frame diffusion process on multiple successive frames I_{t-T}, I_{t-T+1}, $\cdots$ I_{t-1} in the sequence to acquire multiple diffusion probabilities $p^{t-T}(z^t|l_n)$, $p^{t-T+1}(z^t|l_n)$, $\cdots$ $p^{t-1}(z^t|l_n)$ and $p^1(z^t|l_n)$ regarding label l_i. Merging multiple frames' diffusion probabilities we have

$$p(z^t|l_n) = \sum_{s=1}^{T} w_{t-s} p^{t-s}(z^t|l_n). \tag{6.10}$$

where each frame contributes to the final fusion with weight w ($\sum_{s=1}^{T} w_{t-s} = 1$), which is inversely proportional to the alignment error in the scope of the region of interest Ω_n on each frame

$$w_{t-s} = 1/\sqrt{\frac{1}{|\Omega_n|} \sum_{z \in \Omega_n} ||I_{t-s}(z) - I'_{t-s}(z)||^2}. \tag{6.11}$$

where I'_{t-s} is the warped colour image from frame I_t to I_{t-s} by the SIFT flow. Accurate alignment generally indicates reliable SIFT flow and such frames thus contribute more to the probability fusion.

$p(z^t|l_n)$ reveals the likelihood of the pixel at z^t being assigned with label l_i propagated from previous frame in the sequence. This probabilitiy is encoded directly in the *unary* term of our energy function (Eq. 6.4), which comprises a sum of appearance and labelling potentials (described in Sec. 6.4, Eq. 6.12).

6.3.3 Incrementally Updated Colour Model

As we explain shortly (Sec. 6.4), the segmentation of I_t is dependent on the unary term of (6.4) comprising a per label appearance model built incrementally over time. A component of this model is a Gaussian Mixture Model (GMM), the parameters of which are written Θ^n_{col} for each label l_n, and which is initially built by sampling l_n-labelled pixels in starting key-frame I_1. We sample in the RGB colour space following the previous graph cut based methods. To avoid possible sampling errors caused by the imperfect region boundaries, we only use pixels whose spatial distance to the region boundary is larger than a confidence distance (3 pixels in our system) as the training data for the GMMs. As the colour distribution is normally simple in each appearance homogeneous region, the number of components in each GMM is set to 3.

To cope with luminance variations in the sequence, we update the colour model to achieve good segmentation by sampling the historical colours of labelled pixels over recent frames similarly as in Sec. 5.4.2 of Ch. 5.

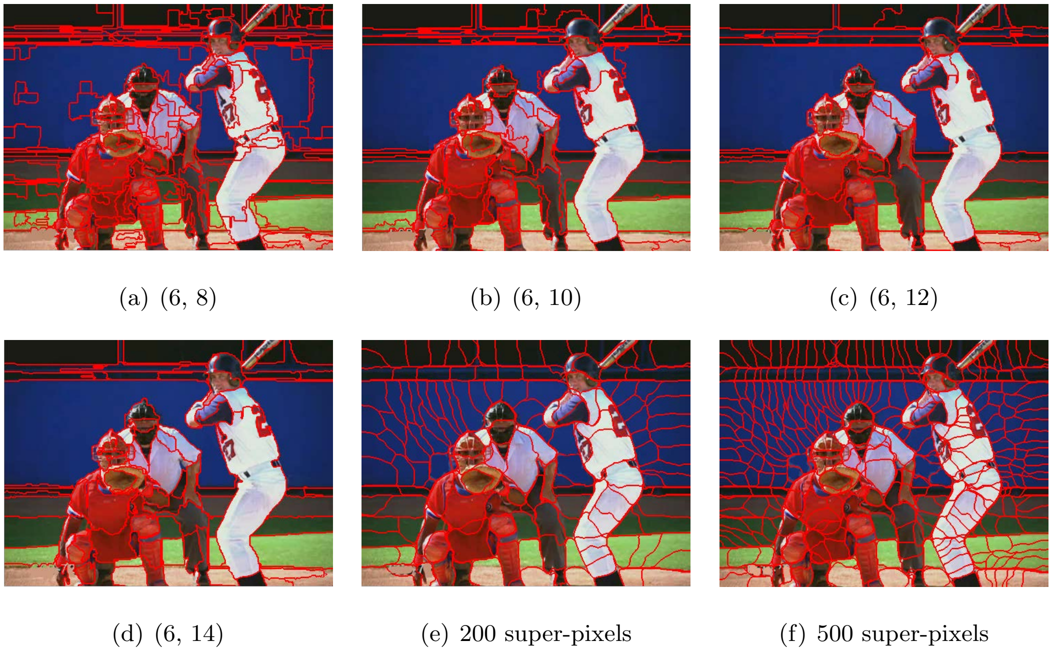
6.3.4 Label Management

If a region labelled n in I_t deviates significantly from its corresponding historic appearance model (determined via a threshold on the χ^2 distance between Θ^n_{col} at time t and $t-1$), then it is likely that the labelling is in error. Given that pixels matching the appearance of labels in the set are likely to be assigned correctly, we assume that significant changes are due to appearance of a new semantic region in the sequence. We therefore run our bootstrap procedure (e.g. mean-shift) over pixels putatively labelled n to create a new set of labels that are merged into $\mathcal{L}$. The frame I_t is then re-segmented using the enriched label set. Any superfluous region labels generated by this process are immediately merged into other similar labels measuring the distance between the GMM colour models via the graph-cut labelling process.

The related problem of label deletion is accommodated naturally within our framework as, depending on the pixel data, the multi-label graph cut may not assign a propagated label to the current frame.

6.3.5 Smoothing and Filtering

Due to visual ambiguities in low contrast areas, some pixels might be mis-labelled which results in unsatisfactory temporal coherence. We improve the temporal coherence by



(a) (6, 8)

(b) (6, 10)

(c) (6, 12)

(d) (6, 14)

(e) 200 super-pixels

(f) 500 super-pixels

Figure 6.2: Illustrating the multiple over-segmentations used to promote label consistency via the super-pixel potential in our energy term (Eq. 6.4), as governed by parameters documented in Sec.6.5.1. (a)-(d) are generated by mean shift segmentation algorithm with different parameters (h_s, h_r); (e)-(f) are generated by *Super-pixel* with particular number of super-pixels.

performing spatio-temporal smoothing operation introduced in Sec. 5.4.

6.4 Definition of Energy Potentials

We now describe how the diffused labelling priors are integrated into the unary, pair-wise and super-pixel consistency terms as defined respectively in (Eq. 6.4). We illustrate the importance of each in Fig. 6.4 where various terms are disabled to qualitatively demonstrate their contribution to segmentation coherence.

6.4.1 Appearance Model

The unary term $\psi_i(x_i)$ exploits the fact that different appearance homogeneous regions tend to follow different appearance models. This encourages assignment of pixels to the label following the most similar appearance model (we write the parameters of such models Θ). The unary term is defined as the negative logarithm of the likelihood of a label being assigned to pixel i. It can be computed from the appearance model for each label. To provide more discriminative power for accurate segmentation, the unary term incorporates colour and texture features as well as *prior* labelling probabilities. The unary term is defined as

$$\psi_i(x_i) = \theta_{col}\psi_{col}(x_i) + \theta_{tex}\psi_{tex}(x_i) + \theta_{lab}\psi_{lab}(x_i). \tag{6.12}$$

where θ_{col}, θ_{tex} and θ_{lab} are weights of colour potential $\psi_{col}(x_i)$, texture potential $\psi_{tex}(x_i)$ and *prior* labelling potential $\psi_{lab}(x_i)$ respectively.

Colour Potential

Colour potential is defined as:

$$\begin{aligned} \psi_{col}(x_i) &= -\log P_g(I_t(i)|x_i;\Theta_{col}). \\ P_g(I_t(i)|x_i = l_n;\Theta_{col}) &= \sum_{k=1}^{K_n} w_{nk}\mathcal{N}(I_t(i);\mu_{nk},\Sigma_{nk}). \end{aligned} \tag{6.13}$$

i.e. the colour model of the n^{th} label l_n is represented by a mixture of Gaussians (GMM), with parameters w_{nk}, μ_{nk} and Σ_{nk} representing the weight, the mean and the covariance of the k^{th} component. The parameters of all GMMs ($\Theta_{col} = \{w_{nk}, \mu_{nk}, \Sigma_{nk}, n = 1, \ldots, |L|, k = 1, \ldots, K_n\}$) are learned from historical observations of each region's colour distribution (sub-Sec. 6.3.3).

Texture Potential

Colour potential alone is not very discriminative and we incorporate texture potential to achieve more accurate segmentation. To this end, we adopt textons [128] which have been proven effective in categorizing materials [208] and generic object classes [225, 108, 190].

For extracting texton histograms, we use a filter bank made of 36 bar and edge filters, 1 Laplacian of Gaussian (LoG) and 1 Gaussian filter. The 36 bar and edge filters (6

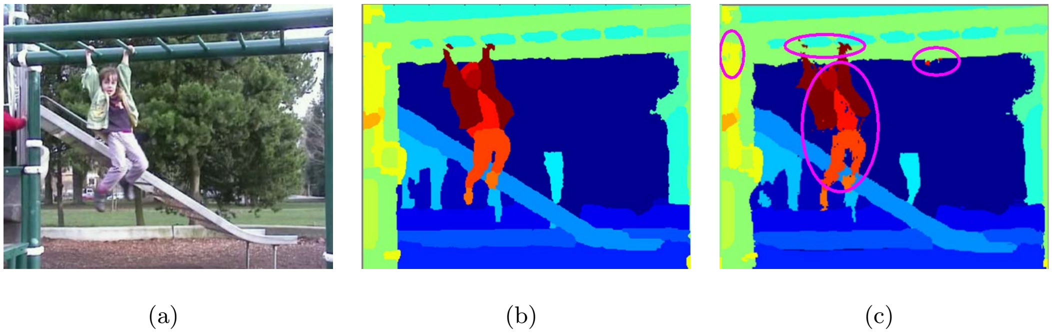


Figure 6.3: Segmentation of the "MONKEYBAR" video with and without the super-pixel consistency term. (a) Original frame; (b) Segmentation obtained with super-pixel potential present in Eq. 6.4 exhibits improved boundary stability when propagated over time, despite computing each frame's super-pixels being computed independently; (c) Segmentation obtained without the super-pixel constraint, differences highlighted in ellipses.

orientations and 3 scales for each) are applied to the L channel only, producing 36 filter responses. The Gaussian filter is applied to each CIELab channel, thus producing 3 filter responses. The LoG is also applied to the L channel only, thus producing 1 filter response. We quantize filter responses to 200 textons by running K-means clustering and each pixel in I_t is assigned to the nearest cluster center to generate the texton map T_t. We define texture potential as:

$$\begin{aligned} \psi_{tex}(x_i) &= -\log P_g(T_t(i)|x_i;\Theta_{tex}). \\ P_g(T_t(i)|x_i = l_n;\Theta_{tex}) &= \mathcal{H}^n(T_t(i)). \end{aligned} \tag{6.14}$$

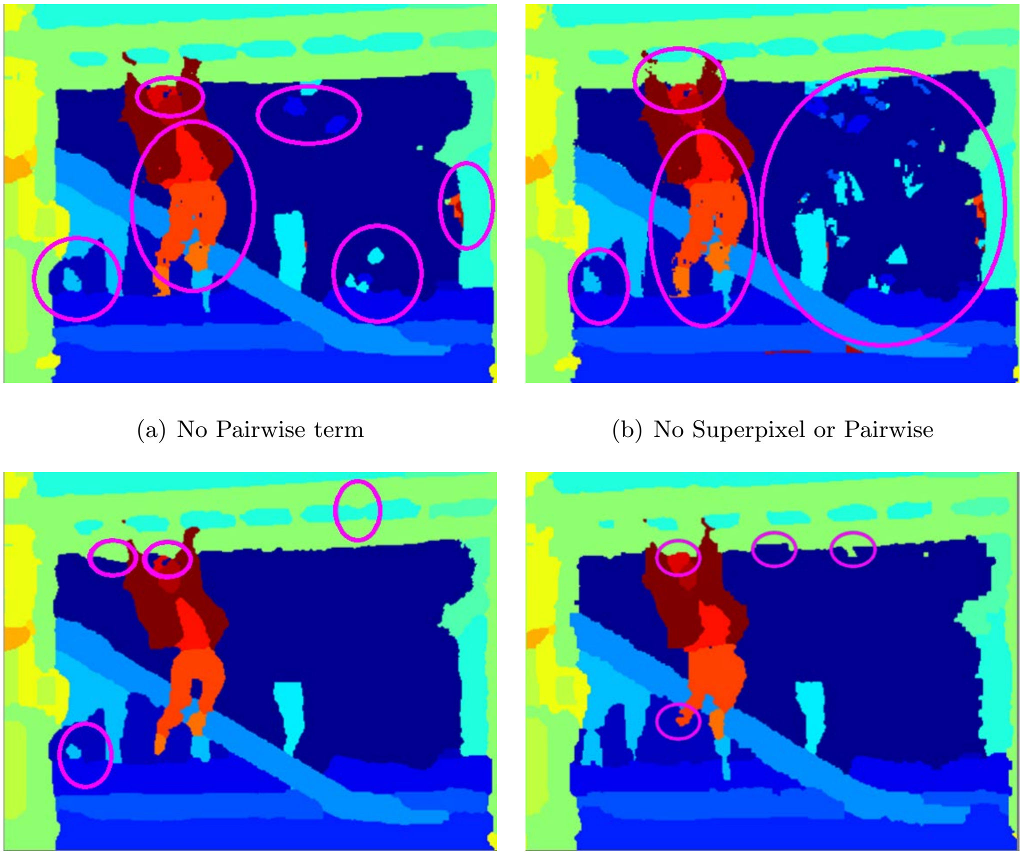
The texture model Θ_{tex} of the n^{th} label l_n is represented by a discrete probability model given the normalized texton histogram $\mathcal{H}^n$ learned from the textons map in the starting key-frame.

Labelling Potential

The labelling *prior* potential exploits the fact that pixels with a higher probability propagated from particular labelled region tend to have consistent label assignment. Unlike other interactive or automatic segmentation algorithms which use the labelling prior as a hard constraint, we incorporate labelling prior as a soft constraint which is inferred from a probabilistic motion estimation framework which inherently takes into account the motion estimation errors. The labelling potential $\psi_{lab}(x_i) = p(i|x_i)$ maps directly to $p(z^t|l_n)$ derived for each pixel, given a label, as defined in sub-Sec. 6.3.2, where $p(i|x_i)$ is the probability that label x_i is propagated to pixel i.

6.4.2 Encouraging Spatial Coherence

The pairwise term encourages coherence in region labelling and discontinuities to occur at high contrast locations, which is computed using RGB colour distance as in Grab-Cut



(a) No Pairwise term

(b) No Superpixel or Pairwise

(c) Only Meanshift Superpixel

(d) Only NCuts Superpixel

Figure 6.4: Illustrating the influence of the unary, pairwise and super-pixel (Spix) terms on segmentation coherence ("MONKEYBAR" sequence). Notable differences to proposed approach (Fig. 6.3b) highlighted in ellipses.

[178]:

$$\psi_{ij}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j, \\ \theta_\lambda e^{-\theta_\beta ||I_t(i) - I_t(j)||^2} & \text{if } x_i \neq x_j, \end{cases}$$

where θ_β is chosen to be contrast adaptive [21]:

$$\theta_\beta = \frac{1}{2} \langle ||I_t(i) - I_t(j)||^2 \rangle^{-1}. \tag{6.15}$$

where $\langle \cdot \rangle$ denotes expectation over an image sample.

6.4.3 Super-pixel Consistency Term

The super-pixel consistency term encourages the pixels belonging to a super-pixel to be assigned with the same label. We define this spatially 'higher order' term as:

$$\psi_c(x_i) = \begin{cases} 0 & \text{if } i \notin c, \\ \frac{\theta_c}{|c|} \sum_{j \in c} \psi_j(x_i) & \text{if } i \in c, \end{cases} \tag{6.16}$$

after [108], where θ_c is the parameter weighting the label consistency partly enforced by super-pixel c, and $|c|$ is the cardinality of super-pixel c. The expression $\sum_{j \in c} \psi_j(x_i)$ gives the label consistency cost, i.e. the cost if all pixels constituting super-pixel c are labelled as x_i (pixel i). $\psi_c(x_i)$ is thus defined as the weighted average unary potential of pixels in super-pixel c against label x_i. The indication is that an optimal label assignment to pixel i should also fit all pixels in super-pixel c as long as c has good homogeneity of visual appearance.

In practice, due to the non-homogeneity of visual appearance and parameter settings, the shapes of super-pixels may not always be consistent with the real object boundaries in only one over-segmentation or one unsupervised segmentation algorithm. Some super-pixels may quite often contain pixels belonging to multiple labels and will encourage an incorrect labelling. Therefore, following [181], multiple segmentations resulted from with different parameter sets of different unsupervised segmentation algorithms [45, 174] per frame are generated, so that although some super-pixels may fail to agree with object boundaries, the others would be good super-pixels that correspond to coherent boundaries. Different unsupervised segmentation algorithms promote differently featured homogeneous regions. Mean shift segmentation [45] generates regions with homogeneous colours, whereas *Super-pixel* [174] produces segmentations incorporating various Gestalt cues, i.e. contour, texture, brightness and good continuation.

Each super-pixel partly enforces the label consistency of regions with a weight. We correlate the weight with the quality of super-pixel from the over-segmentations. We adopt the super-pixel quality measure presented in [107], using the variance of unary potentials of all constituent pixels of a super-pixel as:

$$\sigma_c = \exp(-\frac{\theta_s}{|c|}\sum_{j\in c}(\psi_j(x_i) - \frac{\sum_{j\in c}\psi_j(x_i)}{|c|})^2). \tag{6.17}$$

and θ_c is defined as the normalized σ_c:

$$\theta_c = \frac{\sigma_c}{\sum_{c\in\mathcal{S}}\sigma_c}. \tag{6.18}$$

As opposed to other segmentation algorithms which use the hard label consistency in regions assuming that all pixels constituting a particular region are assigned with the same label, we use it as a soft label consistency constraint, similar to the Robust P^n model and non-parametric approaches of [108, 107]. Unlike the Robust P^n model which is based on the number of pixels in the super-pixel not taking the dominant label, we use the spatial constraint imposed by each super-pixel as soft constraint and naturally incorporate it to the unary term, and thus simplify the optimization without explicitly performing higher-order optimization (see 6.4.4).

6.4.4 Optimization

Although the proposed energy function Eq. 6.4 takes the similar form of the Robust P^n model in [108], the super-pixel consistency term is not based on the count of the number of labelled pixels within a single super-pixel. Rather, we define a *soft* constraint to reflect the label consistency enforced by different over-segmentations. We define this as the weighted average unary potential of pixels in each super-pixel. This definition is convenient as this spatially 'higher order' term does not take multiple numbers of variables in the clique, and so can effectively be further merged to unary term and the energy function Eq. 6.4 can be simplified to:

$$E(x) = \sum_{i \in \mathcal{V}} (\psi_i(x_i) + \frac{\theta_c}{|c|} \sum_{j \in c} \psi_j(x_i)) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j). \tag{6.19}$$

As the pairwise potentials of the energy function Eq. 6.19 is of the form of a Potts model it can be minimised using the α-expansion and $\alpha\beta$-swap algorithms [24]. An α-expansion iteration is a change of labelling such that p either retains its current value or takes the new label l_α. The expansion move proceeds by cycling the set of labels and performing an α-expansion iteration for each label until (6.4) cannot be decreased [24]. Each α-expansion iteration can be solved exactly by performing a single graph-cut using the min-cut/max-flow [23]. Convergence to a strong local optimum is usually achieved in 3-4 cycles of iterations over our label set. We use Alahari *et al.*'s [5] technique to improve the computation and memory efficiency of each iteration by reusing the flow at each iteration of the min-cut/max-flow algorithm, resulting in a two-fold speed-up.

6.5 Experiments and Comparisons

We apply our segmentation algorithm to several video clips exhibiting both slow moving and agile motion, and also a variety of occlusion conditions (no occlusion, self-occlusion, inter-object occlusion) — summarized in Table 6.1. We assess segmentation performance on both a subjective qualitative and objective quantitative basis; the latter using the methodology of the Berkeley Segmentation benchmark [144].

6.5.1 Parameter Settings

We first explain the parameter settings in unsupervised segmentation algorithms, i.e. mean shift and *Super-pixel*, that form the basis for the third term (the higher order constraint) in our optimization. There are two key parameters in mean shift algorithm; bandwidth in the spatial domain (h_s), and the range domain (h_r). A set of regions with various granulations are generated by varying h_s and h_r. As segmentations do not change dramatically with varying h_s on our NTSC resolution video frames we obtain 4 over-segmentations with parameters $(h_s, h_r) = \{(6, 8), (6, 10), (6, 12), (6, 14)\}$.

Clip	Motion	Occlusion
BOY (Fig. 6.5)	Slow	None
DANCE (Fig. 6.5)	Agile	Light
MONKEYBAR (Fig. 6.5)	Agile	Heavy
GARDEN (Fig. 6.9)	Slow	Light
WALKDOG (Fig. 6.10)	Slow	Heavy
YUNAKIM (Fig. 6.10)	Agile	Heavy
SKATEBOARD (Fig. 6.10)	Fast	Light
COWGIRL (Fig. 6.10)	Slow	Light
BASEBALL (Fig. 6.10)	Fast	Heavy

Table 6.1: Summary of video clips used in our evaluation, annotated as to motion and occlusion conditions present.

Super-pixel generates a large number of small nearly-uniform regions which has been shown to retain salient structure in real images. The only parameter in *Super-pixel* is the number of super-pixels or regions to be generated. We generate two sets of regions using *Super-pixel* with 200 and 500 super-pixels respectively. An example of multiple over-segmentations is shown in Fig. 6.2. Weighting parameters θ_{col}, θ_{tex} and θ_{lab} of colour potential $\psi_{col}(x_i)$, texture potential $\psi_{tex}(x_i)$ and *prior* labelling potential $\psi_{lab}(x_i)$ are chosen empirically, and we set $\theta_{col} = 0.31$, $\theta_{tex} = 0.56$ and $\theta_{lab} = 0.13$ respectively. θ_{λ} is set empirically to be 3 to obtain satisfactory segmentation. Other parameter settings are $\theta_s = 0.5$, $\theta_{\gamma} = 6$, $\theta_{\mu} = 2$.

6.5.2 Objective Evaluation

We first present the comparative objective evaluation of the proposed algorithm against two state-of-the-art video segmentation algorithms: Multi-label Propagation (MLP) [220], and spatial-temporal mean shift (STMS) [161]. These algorithms respectively represent an example of a 2D+t and 3D (spatio-temporal) video segmentation algorithm. We additionally compare against a state of the art hierarchical graph based (HGB) approach due to Grundmann *et al.* [76][1].

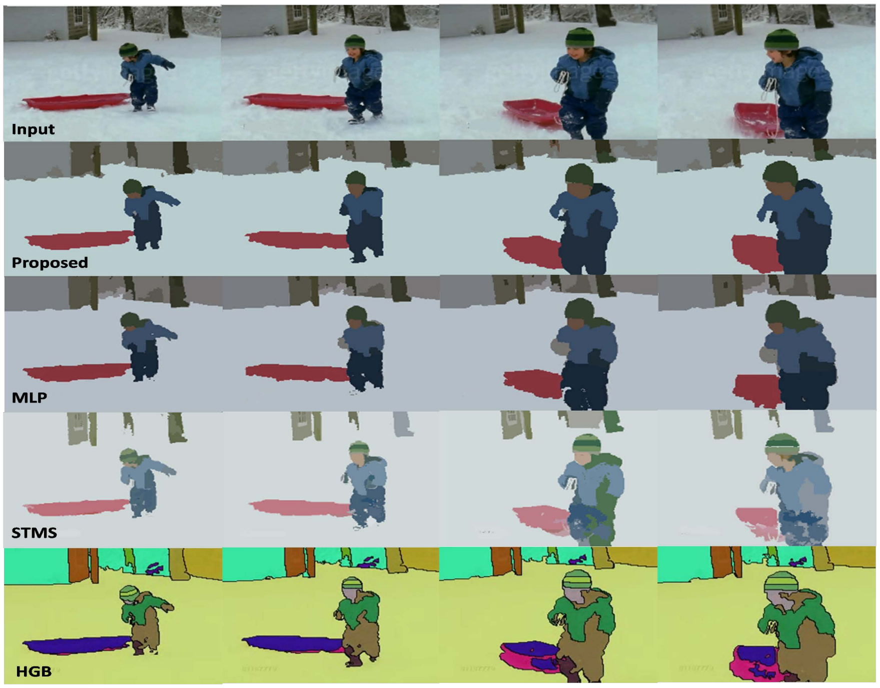
Benchmark

For objective evaluation, we adopt the Berkeley Segmentation Benchmark [144] to evaluate segmentation against manual ground-truth. This boundary-based evaluation methodology has become a standard benchmark. This framework considers two aspects of segmentation performance. Precision measures the fraction of true positives in the contours produced by a segmentation algorithm. Recall indicates the fraction of ground truth boundaries detected in the segmentation. The global F-measure, defined as the harmonic mean of precision and recall, provides a useful summary score for the segmentation algorithm.

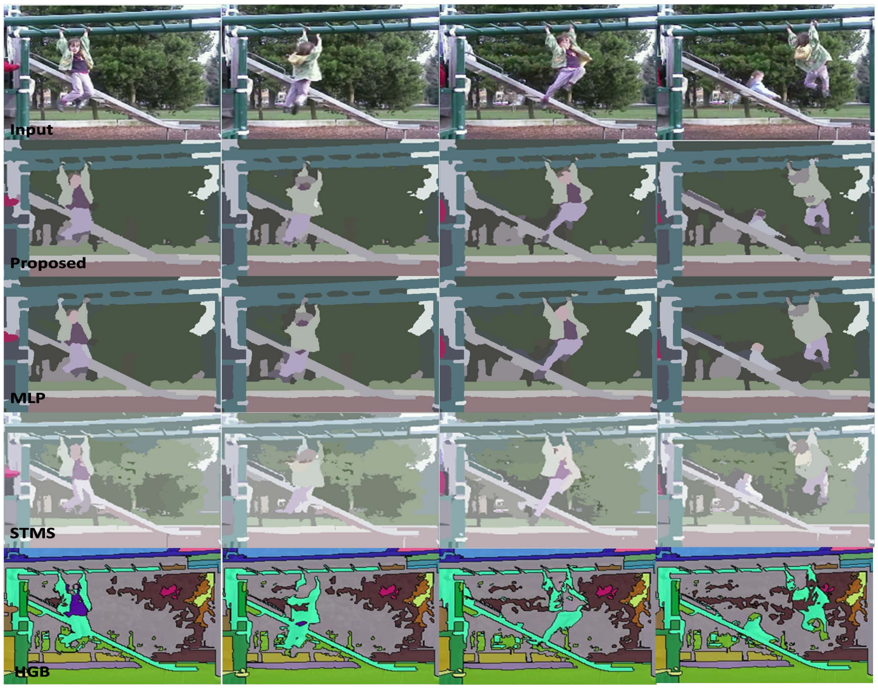
Ground Truth

In order to obtain a reliable estimate of segmentation accuracy under [144] we require ground truth region boundaries. We therefore hand labelled individual frames, seeking to preserve fine object boundaries present. Generating manual ground truth segmentations of all the frames of tested videos is very time consuming. Given the frame rate of 24 fps, we opted to hand label the ground truth every 10 frames, and made a second separate manual inspection visually verifying the boundary accuracy.

[1] Obtained via `http://neumann.cc.gt.atl.ga.us/segmentation/`

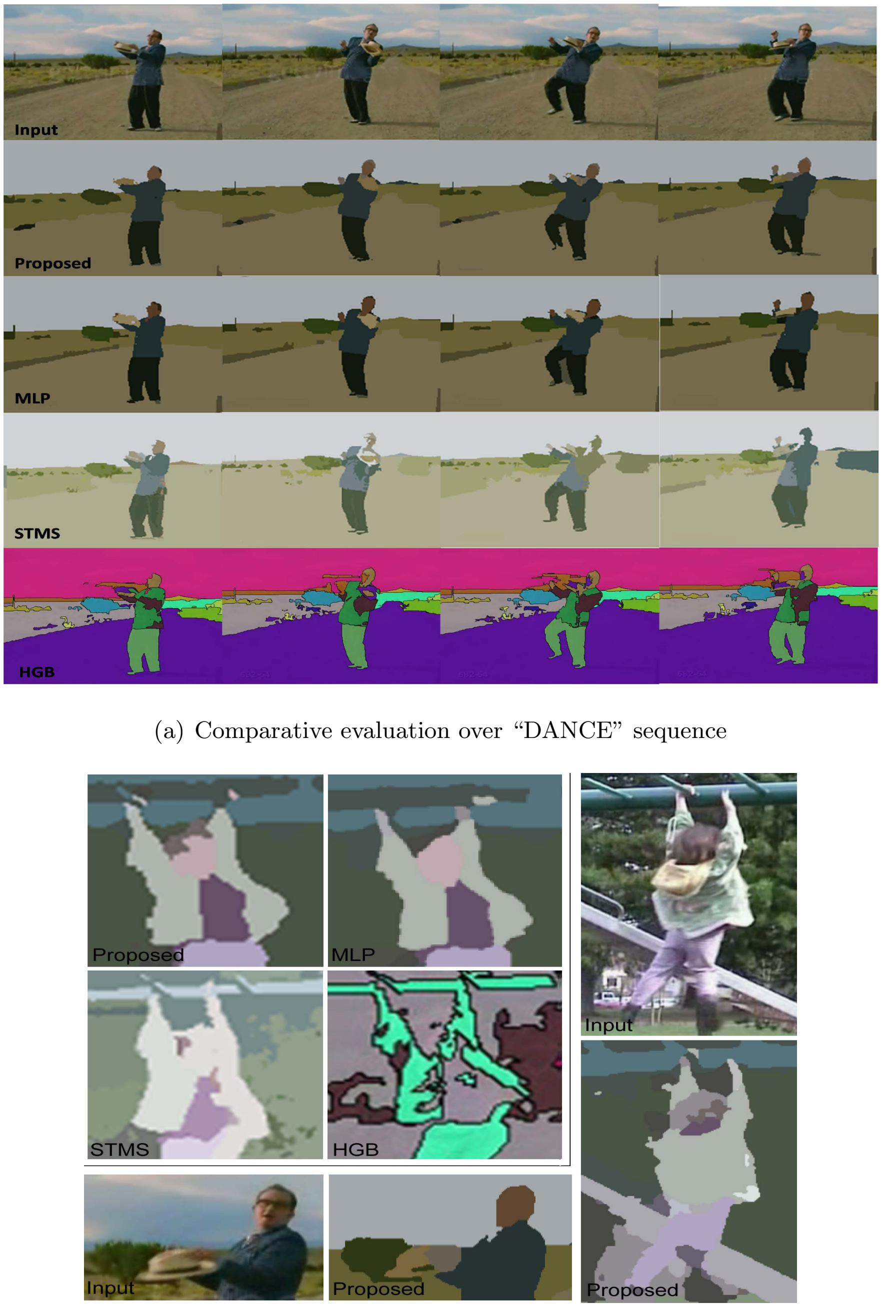


(a) Comparative evaluation over “BOY” sequence



(b) Comparative evaluation over “MONKEYBAR” sequence [212]

Figure 6.5: Comparing the accuracy and coherence of the proposed approach to MLP, STMS and HGB. Boundaries are less prone to variation in shape and topology. Sequences presented as follows: source (1st row); proposed approach (2nd row); MLP (3rd row); STMS (4th row); HGB (5th row).



(a) Comparative evaluation over "DANCE" sequence

(b) Close-ups of successes vs. inaccuracies

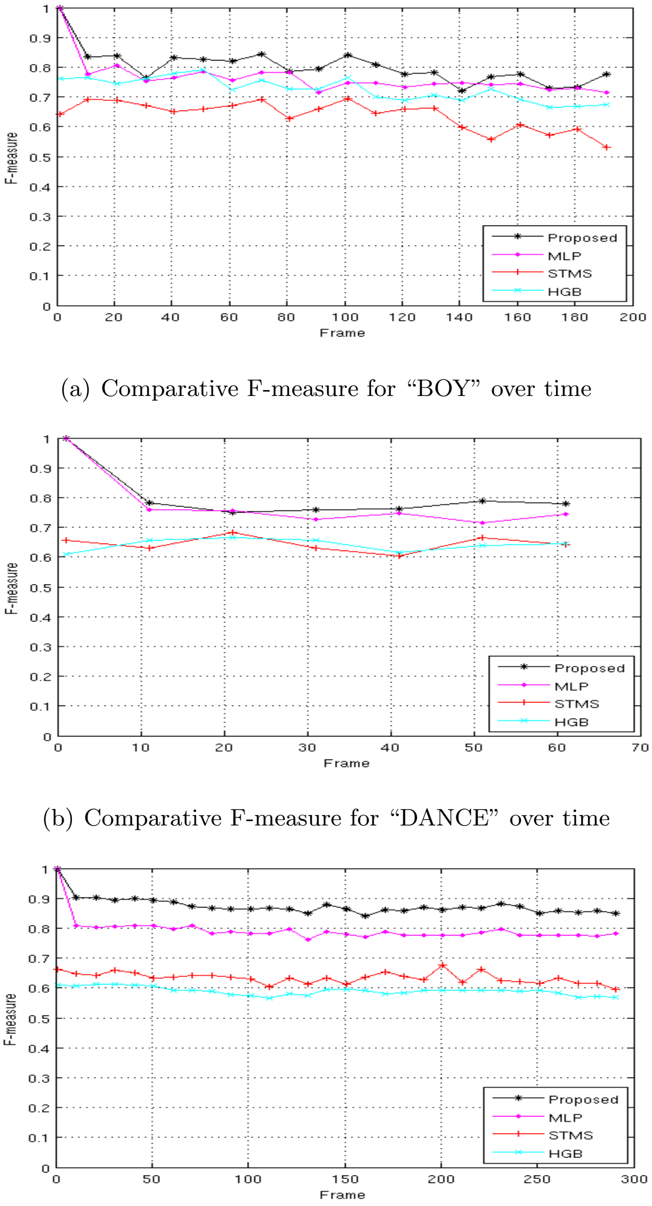
Figure 6.6: (a) Comparing the accuracy and coherence of the proposed approach to MLP, STMS and HGB (continued). Inset (b). 4 × 4: Improved performance of the proposed method vs. state of the art on face and hands in "MONKEYBAR". 2 × 1 Failures cases of the proposed approach, although outperforming compared methods some mislabelling of the hair in "MONKEYBAR" and loss of spatial coherence on hat in "DANCE" can be observed. In both cases these can be attributed to colour texture similarity in the presence of erratic motion.

Results

Figs. 6.7(a)-6.7(c) present the comparison between the proposed method and the MLP, STMS and HGB algorithms over clips “BOY” (192 frames), “DANCE” (62 frames) and “MONKEYBAR” (300 frames). According to the normalized F-measure with respect to manual ground-truth boundaries, our algorithm consistently outperforms the CRF based MLP algorithm, the graph-based HGB and the spatio-temporal STMS approach across the full duration of the clips. Incorporating labelling prior probability as well as the super-pixel consistency potential in (Eq. 6.4) has significantly increased the accuracy and coherence of segmented region boundaries.

Fig. 6.7(a) and 6.7(b) compare our proposed approach to MLP on clips “BOY” and “DANCE” [220]. We observe the region boundaries from our proposed method to exhibit improved stability and accuracy over time over STMS, HGB, and MLP according to the F-measure with respect to manual ground-truth boundaries. Adopting motion cues as a hard constraint in the CRF framework of the MLP algorithm cumulatively leads to mis-labellings close to boundaries; the non-discriminative colour model in MLP further deteriorates the segmentation quality in areas with low contrast or similar colour but different texture properties.

In Fig. 6.7(c) there is significant agile motion in “MONKEYBAR” — the girl twists and suffers frequent inter-occlusion over duration of the clip. Despite the adoption of a forward propagation (2D+t) strategy over several hundred frames of video there is no significant degradation of F-measure over time; the degradation is comparable to STMS (a spatio-temporal approach). The hard assignment propagation strategy of MLP leads to merging of regions, especially in the wake of moving limbs such as the legs (c.f. Fig. 6.9) resulting in a lower F-measure. We observe the HGB algorithm (also based on a form of hard assignment dense flow propagation) to fragment regions signficantly as the sequence progresses, whereas our approach does not, leading that method to produce consistently lower F-measure scores 6.7(c).



(a) Comparative F-measure for "BOY" over time

(b) Comparative F-measure for "DANCE" over time

(c) Comparative F-measure for "MONKEYBAR" over time

Figure 6.7: Evaluation of video segmentation algorithms against manual ground-truth on the Berkeley Segmentation Benchmark. Our proposed algorithm outperforms Multi-label Propagation (MLP) proposed in Sec. 5.4, Grundmann *et al.* [76], and spatial-temporal mean shift (STMS) [161] according to their F-measure (harmonic mean of precision and recall) with respect to manual ground-truth.

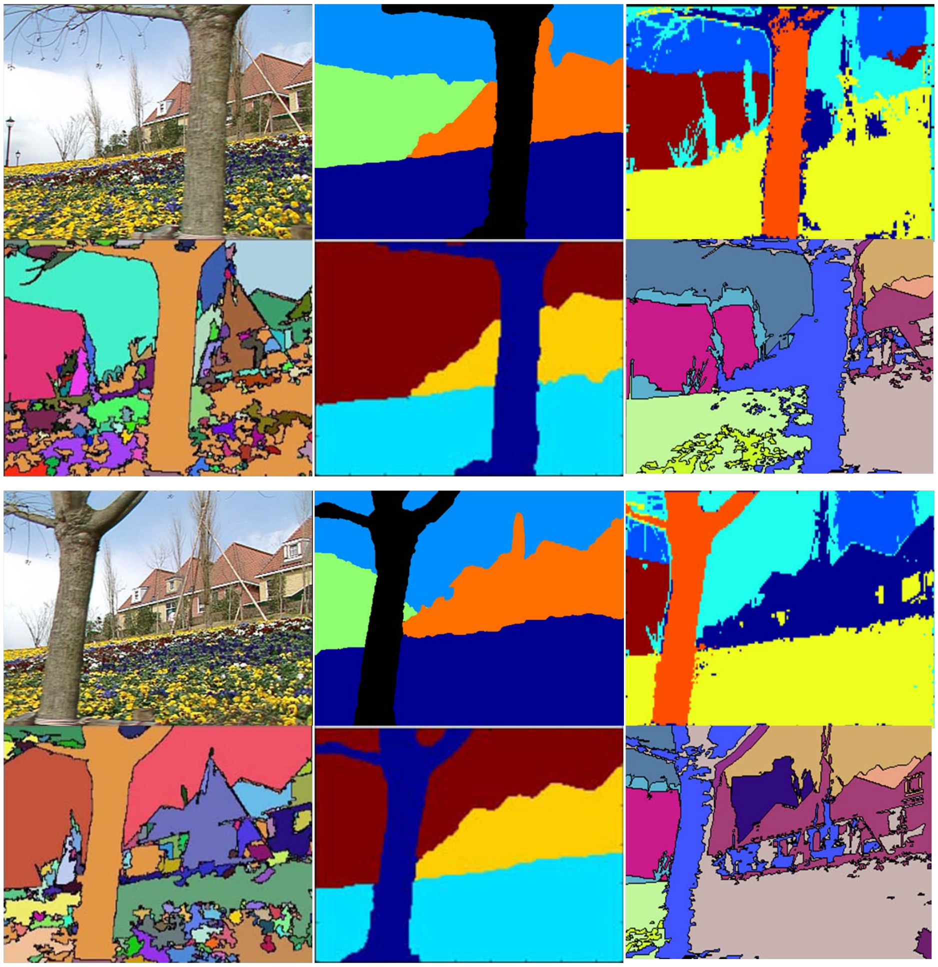


Figure 6.8: Additional qualitative comparison of performance on frames 1 and 30 of the "GARDEN" sequence. Order left-right, then top-bottom: Original; Proposed approach; [104]; [25]; [108]; [76].

6.5.3 Subjective Evaluation

We also demonstrate segmentation results on eight video clips. Each region is shaded with the mean colour of pixels in each labelled region on the starting key frame to evaluate longterm coherence and boundary consistency. Fig. 6.5 and 6.6 make qualitative comparison of the segmentation results of our proposed algorithm, MLP, HGB and STMS on clips "BOY", "DANCE" and "MONKEYBAR". We observe that the relative coherence and boundary accuracy match the objective evaluations in Sec. 6.5.2; for example see the zoomed inset (b). The ability to cope with fast motion and occlusions are significantly improved in the proposed segmentation algorithm over the state-of-the-art. A couple of failure cases are also indicated in Fig. 6.6b, in particular the body of the child ("MONKEYBAR") and the hand/hat of the dancer ("DANCE") are shown to deform unnaturally when undergoing erratic motion over background of similar colour and or texture.

An additional qualitative comparison on the "GARDEN" sequence is provided in Fig. 6.8, comparing against a further MRF/CRF based method [108], HBG [76] and another recently proposed video segmentation algorithm due to Brendel *et al.* [25]. Our method performs comparably to HBG on this sequence (though see other qualitative comparisons, Fig. 6.5) and retains a smaller number of coherent regions vs. [25, 108].

Fig. 6.9 directly compares our probabilistic diffusion ('soft') approach to motion propagation, with the hard-assignment strategy of [220]. The experiment is facilitated by temporarily modifying our approach to work with colour appearance only (no textons) and omitting the super-pixel term during optimization. The benefits of the probabilistic approach are observed on the feet of the child; hard assignment causes incorrect pixel assignments to cumulatively trail the feet over time. Although soft assignment alone causes minor loss of spatial coherence, this is avoided in our proposed system through incorporation of the super-pixel constraint to produce results such as those of Fig. 6.5(b).

Fig. 6.10 shows the remaining five segmentation results on clips "YUNAKIM" (560 frames), "COWGIRL" (224 frames), "BASEBALL" (171 frames), "SKATEBOARD" (146 frames), and "WALKDOG" (300 frames). Our segmentation algorithm exhibits consistent region identity and stable boundaries under conditions such as fast motion, low contrast, ambiguous colour, non-rigid shape, occlusions. Object boundaries are accurately preserved with colour and texture homogeneous regions grouped to ensure temporal and spatial coherence.

6.6 Conclusion

In this chapter, we presented a novel algorithm for video segmentation driven by multi-label graph-cut. Our core contribution was a multi-frame probabilistic motion diffusion

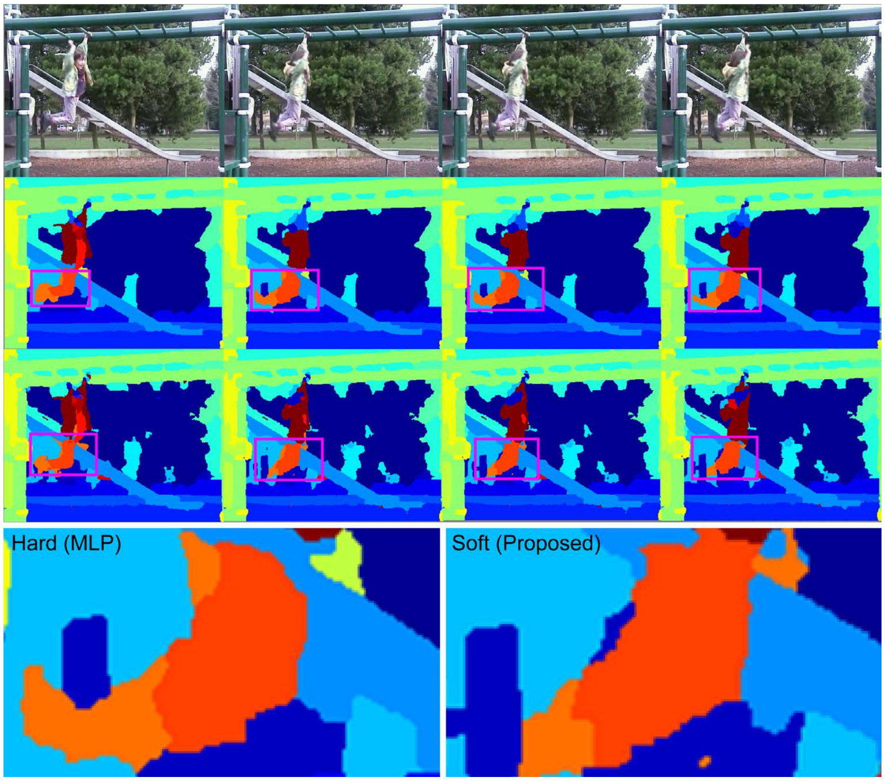
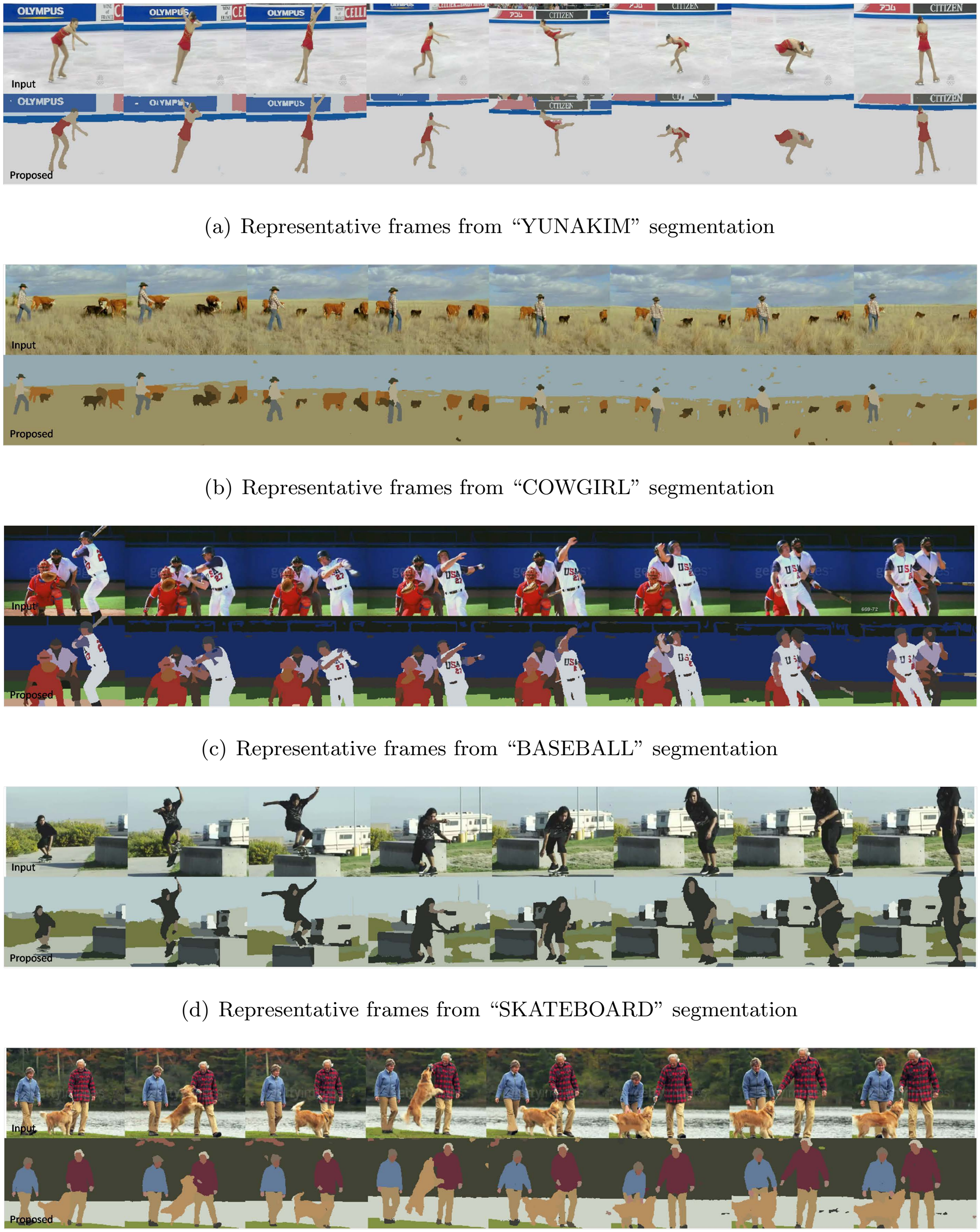


Figure 6.9: Comparison of motion propagation strategy; soft (proposed) vs. hard (Sec. 5.4) assignment. Textons and the super-pixel term are disabled to enable comparison between motion propagation strategies of Sec. 5.4 (row 2) and our approach (row 3); zoomed in sections of third frame sample (row 4). Note the cumulative errors of hard assignment incorrectly smear the feet (orange label) into elongated region over time (highlighted), where the region is correctly segmented using our proposed motion diffusion approach.

model to incorporate labelling priors from previous frames to influence the segmentation in new frame. Uniquely this diffusion model propagated a *per-pixel distribution of labelling priors* forward based on the probability distribution of motion vectors for that pixel. Motion flow estimation remains a challenging open problem in Computer Vision, and our approach mitigates against inaccuracy in such estimates via this “soft” propagation strategy. This was shown to improve temporal coherence over similar hard-assignment strategies [220], graph based schemes based on flow propagation [76] and spatio-temporal segmentation [161]. We combined this motion framework with a spatially ‘higher order’ constraint additionally imposing the soft label consistency constraint across image regions (super-pixels) obtained via various unsupervised segmentations — as is now common in image segmentation. By enforcing labelling consistency, both the spatial coherence and boundary accuracy of the segmentation was enhanced (demonstrated via comparison to a manually labelled ground truth). We demonstrated our algorithm on a variety of sequences exhibit both simple and challenging motion and occlusion conditions.

A current bottleneck in our system is the SIFT-flow estimation, which can take around 10 seconds in total to compute the flow between historic frames at the currently processed frame. Were our algorithm to be used for real-time segmentation, an alternative and perhaps less accurate optical flow method could be trivially substituted.

One interesting direction for future work would be to explore the possibility of propagation labelling priors both forward and backward in the sequence. This could provide an additional temporal constraint with the potential to further enhance temporal coherence. Currently our motion diffusion is Gaussian, and possibly some form of anisotropic diffusion in the direction of motion could further enhance motion coherence. However we do not believe such extensions are necessary to show the value of our motion diffusion model and segmentation framework which in their current form already exhibit improved accuracy on state of the art approaches under the Berkeley F-measure. Furthermore, the dependency on data from only previous time-steps preserves the future possibility of applying an optimized version of our algorithm to online (incrementally streamed) video data. Although our run-time complexity is currently tens of seconds per frame, GPU implementations of the bottle-neck in our system (α-expansion) are available. These future applications are in line with our original project motivations which are to to develop a coherent video object segmentation algorithm for multimedia graphics applications such as video stylisation [219].



(a) Representative frames from “YUNAKIM” segmentation

(b) Representative frames from “COWGIRL” segmentation

(c) Representative frames from “BASEBALL” segmentation

(d) Representative frames from “SKATEBOARD” segmentation

(e) Representative frames from “WALKDOG” segmentation

Figure 6.10: Additional segmentation results applying our approach to NTSC video sequences (source in top row, our result in bottom row). Please refer to **http://personal.ee.surrey.ac.uk/Personal/Tinghuai.Wang/TMM2011** for these and further results.

Part IV

Portrait Stylization

Chapter 7

Digital Raphael: Learnable Stroke Models for Example-based Portrait Painting

In this chapter we address the stylization of people, in particular portraits, which are frequently encountered in personal media collections yet which general AR algorithms perform particularly poorly on. The difficulty in rendering portraits is due to our strong cognitive prior to the structure of human face. A portrait rendering algorithm should account for facial structure to avoid any distortion or loss of detail of facial feature; it also should be able to learn how artists depict the structure with brush strokes and colour. We present a novel representation to interpret human facial features which drives a user trainable algorithm for stylizing photographs into portrait paintings. This composed facial feature representation not only accounts for global structure and higher-level semantics but also encodes local context and low-level visual feature, enabling a wide variety of artistic styles to be encapsulated in one system.

7.1 Introduction

The stylization of photographs into high quality digital paintings remains a challenging problem in computer graphics. In recent years, sophisticated *painterly rendering* algorithms have been proposed that rely increasingly upon computer vision to interpret visual structure and drive the rendering process [228]. Although such algorithms generate a pleasing aesthetic for many image classes e.g. scenic shots, they typically perform poorly on portraits. The human visual system has a strong cognitive prior for portraits, and is particularly sensitive to distortion or loss of detail around facial features [146]. Yet such artifacts are frequently observed when applying general purpose painterly rendering algorithms to photographs of faces. High quality rendering of faces is important, as many usage scenarios for artistic stylization focus upon movie post-production effects, or consumer media collections, which predominantly contain images of people.

This chapter contributes a new stroke-based rendering (SBR) algorithm for stylizing photographs of faces into portrait paintings. SBR algorithms create paintings by compositing a sequence of curved spline strokes on a 2D canvas. In contrast to SBR algorithms that encode various rendering heuristics to target a particular artistic style [228, 89], our algorithm learns the style of a human artist *by example*. Given a photograph, and an ordered list of strokes (and related attributes) captured from a training session in which an artist paints that photograph, we are able to learn the artist's style and render previously unseen photographs of faces into portraits with a similar aesthetic (Fig. 7.1).

Our algorithm is aligned with Image Analogies [93] and derivative techniques [124] that learn stroke models of image filters from a pair of unfiltered and filtered greyscale images. Such systems are able to learn filters, including edge preserving filters reminiscent of a painterly effect, by sampling pairs of corresponding patches from the two images. The learned filter is applied by looking up patches from the new image. Our approach differs as we train at the level of the stroke, learning how the placement and appearance of each brush stroke is modulated according to underlying features in the training image. Image features are composed using a Markov Random Field (MRF) model to warrant both the spatial coherence and structural awareness of feature for stroke learning and rendering. As such, our approach is specialized to the task of painting, enabling a wide variety of artistic styles. We specialize further to portraits by learning stroke models independently within semantic regions of the face, identified using an Active Shape Model (ASM) and Graph Cut. To the best our knowledge our system is the first to

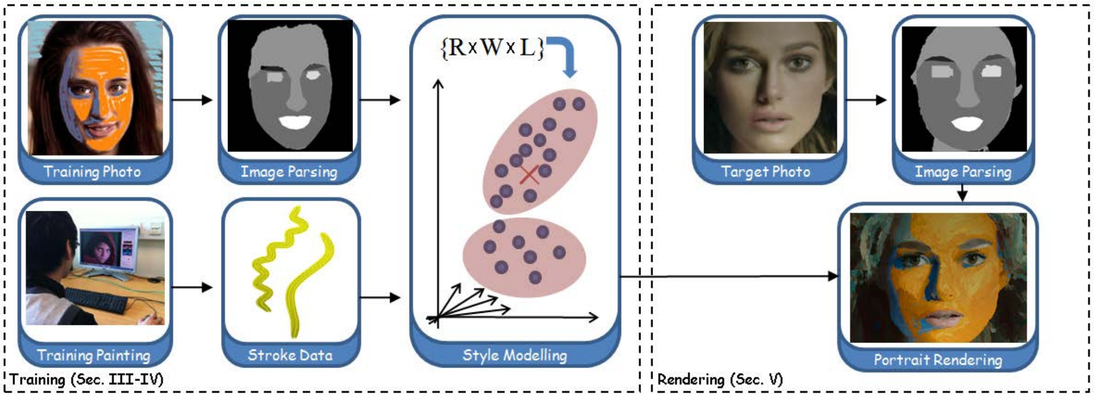


Figure 7.1: Overview of the Digital Raphael system. Stroke properties are learned from a training pair; a source photograph and an artist generated painting. The learned style model can be applied to new photographs to produce portraits in similar styles.

explore painterly stylization by example at the level of the stroke.

7.2 Facial Feature Extraction

Portraiture demands a careful composition of facial features (e.g. eyes, mouth) each differing in depiction style. In Sec. 7.3 we describe how models of stroke appearance are learned independently for each region corresponding to a facial feature in the training image. These models are used to drive the rendering process in (Sec. 7.4) within each region of the target image. For both training and rendering, we therefore perform segmentation on the input photograph to parse semantic regions corresponding to facial features and so label each pixel. In addition to this high level label assignment, we extract mid-level and low-level information to guide the learning process using texture and intensity information respectively. Each pixel of the image is therefore assigned a tuple $\{\mathcal{R}, \mathcal{W}, \mathcal{L}\}$ reflecting the local semantic feature, texture, and luminance. In addition we compute an orientation field Θ from edges and salient facial features.

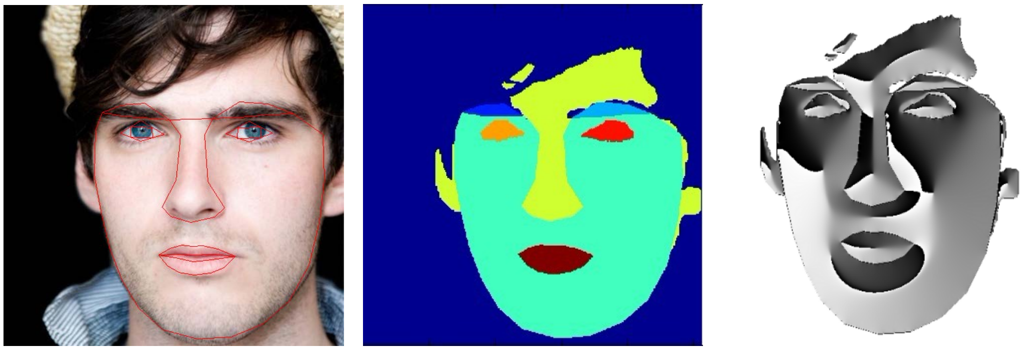


Figure 7.2: Feature extraction, from left to right: (a) Semantic regions formed from the landmarks of ASM; (b) Refined regions using Graph-Cut; (c) Interpolated orientation field in facial area.

7.2.1 Semantic Segmentation ($\mathcal{R}$)

We begin by fitting an Active Shape Model (ASM) [48] to the input image, comprising landmarks local to the eyes, eyebrows, nose, mouth and outline of the facial region. Polygons connecting these landmarks form rough contours of a subset of semantic facial regions (Fig. 7.2(a)) which we use as a basis for deriving a more complete facial representation.

Although the optimized ASM yields a reasonable approximation to feature positions, the model is insufficiently flexible to accurately reflect the shape of each facial feature. We therefore extract a spatial and color prior from the ASM regions to drive a Graph Cut segmentation local to the bounding box of each feature [22]. This refinement is performed for the mouth, eyes and eyebrows — where precision is particularly important in a portrait. The foreground and background color models are each represented by a Gaussian Mixture Model (GMM). The foreground model is learned from pixels within the feature being refined; these pixels are also used as the initial foreground labels. The background model is assumed to be a model of skin tone, and is bootstrapped from the nose region.

Regions of greater shape diversity such as the forehead, ears and neck can not be represented in the ASM and are addressed using a further segmentation over the entire image. Skin areas acquired from the ASM are labeled as foreground and used to train a foreground color GMM; pixels on image borders are labeled as background and used to train a background GMM. After applying Graph-Cut, pixels classified as foreground but exterior to the ASM facial area are labeled as the forehead, ears and neck respectively according to their spatial relationship. The remaining pixels (hair, clothes and background) are treated as one region. Thus the portrait is finally parsed into six regions $r \in \mathcal{R}$ (Fig. 7.2(b)).

7.2.2 Codebooking Visual Structure ($\mathcal{W}$)

Artists paint different types of image feature differently; for example long thin strokes along edges. Rather than prescribe such heuristics, we establish a basis upon which to learn this behavior by assigning each pixel a label reflecting the local image structure it contributes to. We densely sample a SIFT descriptor [140] at each pixel to characterize this contextual information. Dense SIFT has previously demonstrated its discriminative power in face recognition [137]. A dictionary of 20 visual words is built by running k-means over all descriptors sampled in the training image. Each descriptor is assigned a unique word $w \in \mathcal{W}$ in the dictionary. The dictionary is preserved for later use when rendering a new image, as dense SIFT features from that new image must be assigned to codewords in the same dictionary in order to create a basis for comparison with the training data. Note that we use considerably fewer visual codewords than that of large image database applications [182]. Accurate matching of codewords is not required (or desirable) given potential variation between portrait images, and a compact codebook also produces larger spatially coherent regions.

Descriptor-codeword assignment on a nearest-neighbor basis normally produces noisy and spatially uncoherent regions in the facial area due to the high similarity of SIFT features over skin (Fig. 7.3(a)). We adopt a Markov Random Field (MRF) model to optimise the labeling f which assigns codeword $w \in \mathcal{W}$ to each pixel with SIFT descriptor $s \in \mathcal{S}$. Let $\tilde{\mathcal{S}} \subseteq \mathcal{S}$ be the set of corresponding SIFT descriptors of the codebook. We assume that the codewords should vary smoothly almost everywhere but may change dramatically at some places where the local structure varies. The energy of labeling is given by

$$E(f) = \sum_{i \in \mathcal{I}} D_i(w_i) + \sum_{i \in \mathcal{I}, j \in \mathcal{N}_i} V_{i,j}(w_i, w_j)$$

where $\mathcal{N}_i$ denotes the set of four-connected neighbours of pixel i. $D_i(w_i)$ is the cost of assigning codeword w_i to pixel i, which provides the local evidence of the labeling. $V_{i,j}(w_i, w_j)$ is the cost of assigning codewords w_i and w_j to two neighboring pixels, which measures the neighbouring compatibility. Finding a labeling with minimum energy corresponds to the MAP estimation problem for an appropriately defined MRF.

Local Evidence

The goal of the local evidence term is to find a codeword w_i (SIFT descriptor $\tilde{s_i}$) in the codebook which is the nearest neighbour to the local descriptor s_i in feature space. We define the local evidence term as the truncated Euclidean distance between local descriptor and the descriptor of codeword,

$$D_i(w_i) = \min(\mu || s_i - \tilde{s_i} ||, \tau),$$

where μ (0.01 in our system) is the rate of increase in the cost, and τ (100 in our system) controls when the cost stops increasing.

Neighbouring Compatibility

The neighbouring compatibility term aims to make the neighboring estimated codeword map smooth whilst discontinuous at places where the local structure varies dramatically. In our model it is defined as the truncated Euclidean distance between the descriptors of neighbouring pixels in feature space,

$$V_{i,j}(w_i, w_j) = \min(\mu || s_i - s_j ||, \tau).$$

Belief Propagation

We use the max-product belief propagation (BP) [201] for performing inference on the Markov Random Field, which works by passing messages around the graph defined by

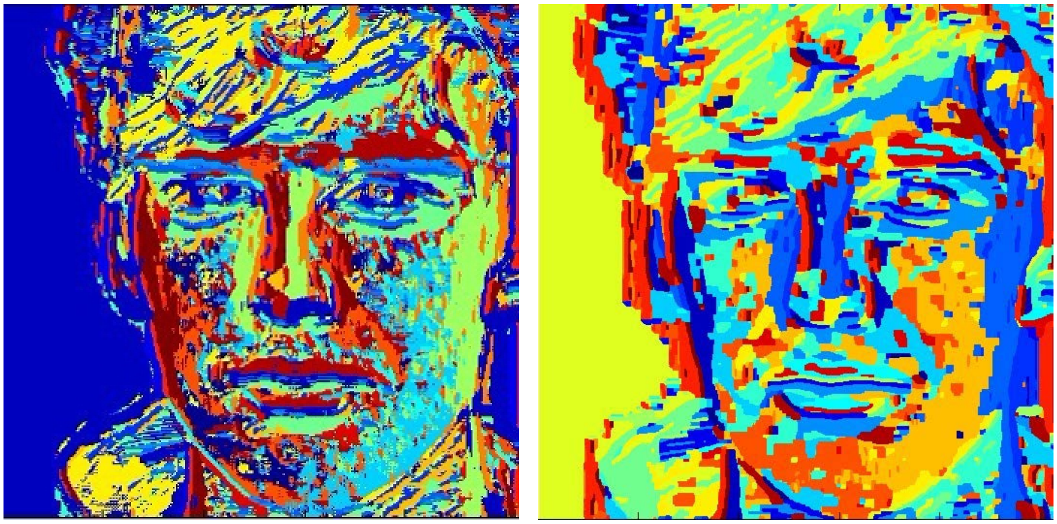


Figure 7.3: Codeword map formed by (a) hard quantisation of dense SIFT features and (b) proposed method using the Markov Random Field model.

the four-connected image grid. BP is an approximate optimisation while in most cases this relaxed solution is good enough for the MRF inference. Each message is a vector of dimension given by the number of possible labels.

Let M_{ij}^t be the message that pixel i sends to a neighbouring pixel j at time t, and at each iteration new messages are computed as

$$M_{ij}^t(w_j) = \min_{w_i}(D_i(w_i) + V_{i,j}(w_i, w_j) + \sum_{k \in \mathcal{N}_i \setminus j} M_{ki}^{t-1}(w_i))$$

where $\mathcal{N}_i \setminus j$ denotes the neighbours of i other than j. After T iterations a belief vector is computed for each pixel,

$$e_j(w_j) = D_j(w_j) + \sum_{i \in \mathcal{N}_j} M_{ij}^T(w_j).$$

Finally, the codeword w_j^* which minimises $e_j(w_j)$ individually at each pixel is chosen after 10 iterations. Fig. 7.3(b) illustrates the codeword map formed by our proposed method, which exhibits significantly improved coherence compared with the hard quantisation on a nearest-neighbor basis (Fig. 7.3(a)).

7.2.3 Orientation (Θ)

Stroke direction, and the use of light and shadow are critical components of portraiture. However image phenomena such as intensity level and orientation that typically correlate to these stroke attributes are not well represented by $\mathcal{W}$, due to the affine and illumination invariance of SIFT. As with subsec. 7.2.2 we desire spatial coherence of the low-level properties, to facilitate coherent variation of rendering parameters.

Orientation with the Facial Regions

We create an edge map $M(x,y) = \{0,1\}$ consisting of the contour of semantic facial regions and salient edges from Sobel operator, from which we interpolate an orientation field. Given this edge map, we compute a sparse field from the gradient of edge pixels $\theta[x,y] \mapsto \text{atan}\left(\frac{\delta M}{\delta x} / \frac{\delta M}{\delta y}\right), \forall_{x,y} M(x,y) = 1$. We define a dense orientation field Θ_{Ω^-} over all coordinates within the facial region Ω^-, minimizing:

$$\underset{\Theta}{\text{argmin}} \int\int_{\Omega^-} (\bigtriangledown\Theta - \mathbf{v})^2 \quad s.t. \;\; \Theta|_{\delta\Omega^-} = \theta|_{\delta\Omega^-}. \tag{7.1}$$

i.e. $\triangle\Theta = 0$ over Ω^- s.t. $\Theta_{\delta\Omega^-} = \theta|_{\delta\Omega^-}$ for which a discrete solution was presented by Perez *et al.* [164] solving Poisson's equation with Dirichlet boundary conditions. $\mathbf{v}$ represents the first order derivative of θ. Fig.7.2(c) shows the smooth interpolated orientation field in facial area, where the orientation field strongly correlates to the facial structure.

Orientation within the Non-facial Region

The texture of importance in the non-facial area in portrait rendering is the hair. Estimating the local orientation based on the eigenvalues of the structure tensor has been proven effective in modelling anisotropic textures [117]. We compute an orientation field Θ_{Ω^+} over all coordinates exterior to the facial region including the hair using the structure tensor. Suppose $f \in \mathbb{R}^3$ denotes the image, we first compute the structure tensor at a given point at $(i,j) \in \Theta_{\Omega^+}$ as

$$g_{i,j} = \begin{bmatrix} \frac{\partial f}{\partial x} \cdot \frac{\partial f}{\partial x} & \frac{\partial f}{\partial x} \cdot \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial x} \cdot \frac{\partial f}{\partial y} & \frac{\partial f}{\partial y} \cdot \frac{\partial f}{\partial y} \end{bmatrix} \tag{7.2}$$

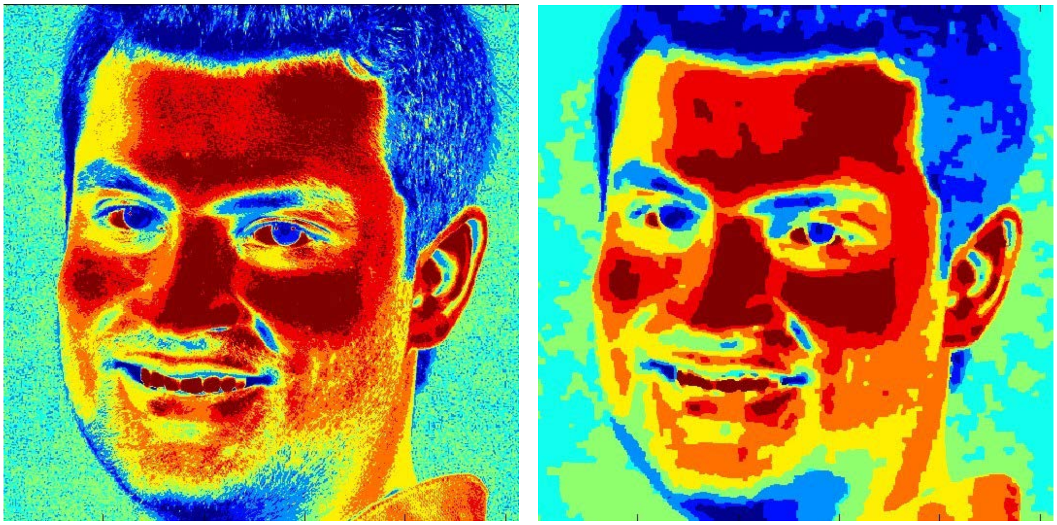


Figure 7.4: Luminance quantisation map formed by (a) hard quantisation of intensity and (b) proposed method using the Markov Random Field model.

The structure tensor at a given point measures the squared rate of change in $\mathbb{R}^3$ and the minimum rate of change is attained along the eigenvector $v_{i,j}$ of the minor eigenvalue of $g_{i,j}$. To compute Θ_{Ω^+}, the vector field formed by assigning $v_{i,j}$ to each point (i, j) is further smoothed by a 5×5 Gaussian kernel.

7.2.4 Intensity ($\mathcal{L}$)

We represent intensity level by quantising the luminance channel into 8 bins, assigning the bin number $l \in \mathcal{L}$ to each pixel. Due to the variation of lighting over face, hard quantisation may produce spatially noisy distribution of quantised levels (Fig.7.4(a)). We propose to optimise the quantisation of the luminance channel using the Markov Random Field model to account for both the local smoothness of quantisation level and the local evidence of intensity. Furthermore, we encode the local structure information to promote discontinuities at places where the SIFT feature changes dramatically.

To this end, we formulate intensity quantisation as a pixel-labeling problem of assigning each pixel $i \in \mathcal{I}$ with a value from the existing label set, i.e. the quantisation level, $\mathcal{L}$. Let $\mathcal{V}$ be the set of intensity values in the luminance channel, and $\tilde{\mathcal{V}} \subseteq \mathcal{V}$ be the set of intensity values in edges from hard quantisation. We assume that the quantisation level should vary smoothly almost everywhere but may change dramatically at some places where the intensity and local structure vary significantly. The energy of labeling is given by

$$E(f) = \sum_{i \in \mathcal{I}} D_i(l_i) + \sum_{i \in \mathcal{I}, j \in \mathcal{N}_i} V_{i,j}(l_i, l_j)$$

$D_i(l_i)$ is the cost of assigning quantisation level l_i to pixel i, which provides the local evidence of the labeling. $V_{i,j}(l_i, l_j)$ is the cost of assigning quantisation levels l_i and l_j to two neighboring pixels, which measures the neighbouring compatibility. The inference of MRF is performed by Belief Propagation [201]. Fig.7.4(b) demonstrates the intensity map formed by our proposed quantisation method, which exhibits improved structure-aware smoothness comparing with the hard quantisation (Fig.7.4(a)).

Local Evidence

The cost of assigning a particular quantised intensity for a pixel is based on the difference between that intensity and the observed value,

$$D_i(l_i) = \min(||v_i - \tilde{v_i}||, \lambda),$$

where λ (100 in our system) influences the point at which the cost stops increasing.

Neighbouring Compatibility

The artist might depict different local structures with similar illumination as totally different styles. We encode local structure information when deciding the luminance quantisation. To promote the discontinuities at places where the local structure changes dramatically, we define the neighbouring compatibility term to account for both the luminance smoothness and the SIFT feature variation,

$$V_{i,j}(w_i, w_j) = \min(||v_i - v_j||, \lambda) + \min(\mu||s_i - s_j||, \tau).$$

7.3 Learning Stroke Style

Our painterly rendering algorithm (Sec. 7.4) adopts the Stroke based Rendering (SBR) paradigm originally outlined by Haeberli [79]. A sequence of curved spline brush strokes are composited on a canvas to create a painting. Each stroke is represented using an Catmull-Rom (interpolating) piecewise cubic spline, comprising n control points $c_{1...n}$. Stroke properties such as geometry (i.e. stroke length and position), as well as stroke thickness and color, are determined using information sampled local to $c_{1...n}$ in the image.

Our training process operates by observing these properties in strokes within manually created training paintings. The system learns the mapping of these properties to pixel derived information within the training photograph (such as color, Θ) local to each stroke's control points $c_{1...n}$. i.e. we observe the stroke property set $\mathcal{P}$ given features $\mathcal{F}$ present in the image local to these points. The mapping is learned by modelling the distribution $p(\mathcal{F}|\mathcal{P})$ independently for each facial region ($\mathcal{R}$). The rendering process then estimates the stroke parameters $\mathcal{P}$ given features $\mathcal{F}$ observed in a new image via:

$$p(\mathcal{P}|\mathcal{F}) \propto p(\mathcal{F}|\mathcal{P})p(\mathcal{P}) \tag{7.3}$$

where we assume all stroke properties are equally likely (uniform prior). $p(\mathcal{F}|\mathcal{P})$ is learned independently for each $\mathcal{W}$ or $\mathcal{W} \times \mathcal{L}$ pair as we now describe.

7.3.1 Color Transfer Model

Particular features or visual structures may cause an artist to shift toward particular shades or hues; for example, complementary pairs of colors are often used by artists to emphasize light and shadow. We learn a color transfer function $\mathcal{F}_c : \{\mathcal{R}, \mathcal{W}, \mathcal{L}\} \mapsto \{\Delta a, \Delta b\}$ for each three-tuple, where $\{\Delta a, \Delta b\}$ indicates the deviation of training stroke color from the training source image in the a and b channels of $CIELab$ space. By learning for each three-tuple we sample a color transfer model for various illumination levels of each category of visual artifact ($\mathcal{W}$) — which are in turn, learned independently for each region ($\mathcal{R}$).

For a given tuple we learn the transfer function as follows. Our system accumulates the color deviation $\{\Delta a_{c_i}, \Delta b_{c_i}\}$ in the a and b channels of stroke color at each control point c_i from the underlying image to the tuple entry $\{r_{c_i}, w_{c_i}, l_{c_i}\}$. Color deviation of each tuple entry is averaged after the painting is finished to form a $2D$ vector

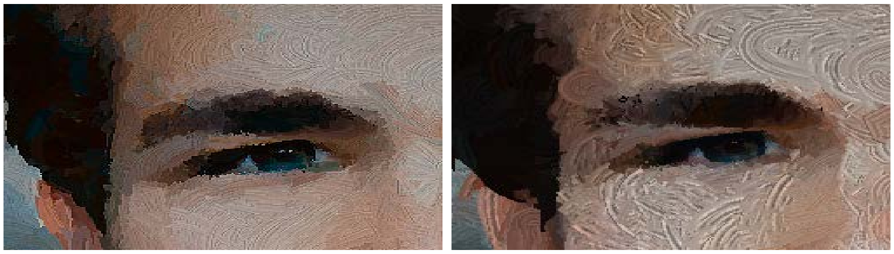


Figure 7.5: Learning stroke orientation. (a) strokes in the direction of intensity gradient, trained by Fig.7.11(e). (b) scribbled strokes in Fig.7.11(f) cause diverse, swirling orientations.

$\{\Delta a_{(r,w,l)}, \Delta b_{(r,w,l)}\}$, which encodes how the painter uses color to account for different semantic facial regions, local structures and luminance variations. In total $|\mathcal{R} \times \mathcal{W} \times \mathcal{L}|$ color transfer vectors may be generated. Due to the large tuple-space the color transfer functions are represented sparsely and interpolated at run-time (subsec. 7.4.4).

7.3.2 Stroke Orientation Model

We model the orientation deviation of strokes using field Θ (subsec. 7.2.3). Portraits of a specific style exhibit characteristic patterns of stroke orientations local to visual structure. As with color (subsec. 7.3.1) we learn orientation as a transfer function, here encoding deviation between Θ and stroke orientation.

We compute the local stroke orientation at control point c_i using the coordinates of two consecutive points as $\theta[c_i] \mapsto \text{atan}(c_{i-1} - c_i)$. Thus the deviation of stroke orientation from underlying orientation is computed as $\theta[c_i] - \Theta_\Omega[c_i]$, observations of which are per tuple entry $\{r_{c_i}, w_{c_i}\}$. Note we use a two-tuple rather than a three-tuple because the orientation model is mainly influenced by the perceptual structure $\mathcal{R}$ and local context $\mathcal{W}$ and assumed invariant to intensity change $\mathcal{L}$. Once strokes in the training process have been accumulated, the expectation μ and standard deviation σ of orientation deviations are computed for each tuple entry yielding a Gaussian model $\mathcal{N}^o_{(r,w)}(\mu, \sigma^2)$. Up to $|\mathcal{R} \times \mathcal{W}|$ models are learned.

7.3.3 Stroke Density Model

The density of painted strokes can vary according to the visual salience of depicted features, and their appearance. We learn a model of stroke density as the probability that a given point is seeding a stroke, subject to the semantic facial region and local structure.

We linearly interpolate the sequence of control points to extract a set of consecutive points $\{p_0, p_1, \ldots, p_N\}$. Each point p_i corresponds to a tuple entry $\{r, w\}$; a count $O_{(r,w)}$ is maintained for each tuple. The count is normalized by area on a per region basis, i.e. over all $\mathcal{W}$ for a given $\mathcal{R}$. A set of $|\mathcal{R} \times \mathcal{W}|$ stroke density probabilities are obtained.

7.3.4 Stroke Thickness and Length Models

We accumulate the thickness of strokes, creating a count on each tuple entry $\{r, w\}$ corresponding to the classification of the control point c_i. For example control point, the count is incremented by the ratio of the stroke thickness to the width of the face, to account for the scale variation of facial area. An identical procedure is undertaken to record stroke length. Stroke thickness counts are normalized by stroke length to prevent bias from over-long strokes. After the painting is finished, we calculate the expectation μ and standard deviation σ of stroke thickness per tuple entry to form a Gaussian model $\mathcal{N}^s_{(r,w)}(\mu, \sigma^2)$. A Gaussian model $\mathcal{N}^l_{(r,w)}(\mu, \sigma^2)$ is similarly learned for stroke length. Up to $|\mathcal{R} \times \mathcal{W}|$ stroke thickness and length models are generated.

7.4 Portrait Rendering

We present a novel digital portrait rendering system in this section which driven by the models of stroke properties learned in Sec. 7.3. Given a source portrait photograph to render, we perform identical preprocessing to the learning process — parsing a per-pixel three-tuple labelling $\{\mathcal{R}, \mathcal{W}, \mathcal{L}\}$ and extracting a composite orientation field Θ_Ω. We take a “single” layer approach to painterly rendering, as learning the order of training strokes is neither feasible nor necessary — our models learned local to structure feature naturally encode the stroke properties adopted by artist to depict salient feature or flat texture at various levels.

7.4.1 Seeding of Stroke Positions

Curve strokes are ‘grown’ bi-directionally from a number of seed locations, in an iterative process (subsec. 7.4.2). The layout of seed positions follows the stroke density model $P^d_{(\mathcal{R},\mathcal{W})}$ learned in Sec. 7.3.3. Given a point c_i and the associated tuple entry $\{r_{c_i}, w_{c_i}\}$, the probability that a stroke seed is generated at point c_i is $P^d_{(r_{c_i}, w_{c_i})}$. The main advantage of our method is that it strongly correlates to the stroke density pattern in the training painting with regard to generic facial and local structures. This approach to stroke spatial layout enables a variety of artistic styles and abstraction levels using different numbers of strokes.

7.4.2 Stroke Evolution

Each stroke is grown bi-directionally from a seed control point c_0, with two additional control points being placed a short ‘hop’ distance away — the direction of the hop is determined by vectors with orientation $\theta[c_0]$ and $\theta[c_0] + \pi$, after [89]. Orientation $\theta[c_0]$ is computed based on sampling from the learned orientation model $\mathcal{N}^o_{(r_{c_0}, w_{c_0})}(\mu, \sigma^2)$ trained in Sec. 7.2.3. $\{r_{c_0}, w_{c_0}\}$ is the tuple entry associated with c_0 in the parsed face representation. We use a truncated Gaussian $\mathcal{N}^o_{(r_{c_0}, w_{c_0})}(\mu, \sigma^2, a = -\sigma, b = \sigma)$. The maximum length l_{max} of the current stroke initiated from c_0 is generated from the truncated Gaussian $\mathcal{N}^l_{(r_0, w_0)}(\mu, \sigma^2, a = -\sigma, b = \sigma)$ learned in Sec.7.3. The thickness is similarly sampled from truncated Gaussian $\mathcal{N}^s_{(r_0, w_0)}(\mu, \sigma^2, a = -\sigma, b = \sigma)$. Note the orientation of each stroke fragment is updated on each new control point c_i following $\mathcal{N}^o_{(r_{c_i}, w_{c_i})}(\mu, \sigma^2, a = -\sigma, b = \sigma)$ whilst the maximum length and size of the stroke are fixed once the initial control point is determined.

The stroke grows from the initial seed c_0 to point c_{-1} and c_1 along $\theta[c_0]$ and $\theta[c_0] + \pi$ respectively with a minimum length of L_{min} (2 pixels in the system); this process iterates until any of the following criteria are violated. Growth halts if the new control point belongs to a different semantic region than c_0, or if the curvature change between a pair of consecutive stroke fragments is larger than a threshold T_a. If the color difference (in CIELab space) between the pixel at the new control point to the pixel at c_i is larger than a threshold T_c then growth halts. We use thresholds of $T_c = 20$ and $T_a = \frac{\pi}{2}$ determined empirically.

Each region is painted independently in order of area (largest first). Strokes are rendered using bump-mapping to enhance their painted appearance [91].

7.4.3 Stroke Color Transfer

After all the control points of a new stroke are generated, the associated code word $\tilde{w}_{c_0}$ with the highest occurrence, and thus the tuple entry $(r_{c_0}, \tilde{w}_{c_0})$ can be found to impose the structure constraint to partially determine the stroke color. However, the stroke color is strongly influenced by the low-level image data. Similar to the training process of Sec. 7.3, we quantize the averaged L channel of all the control points in $CIELab$ space as $\tilde{l}_{c_0}$. All together, we identify the tuple entry $\{r_{c_0}, \tilde{w}_{c_0}, \tilde{l}_{c_0}\}$ to index the color deviation model learned from Sec. 7.3.1.

Given the associated tuple entry $\{r_{c_0}, \tilde{w}_{c_0}, \tilde{l}_{c_0}\}$, the color deviation model of the current stroke is a $2D$ vector $\{\Delta a_{(r_{c_0}, \tilde{w}_{c_0}, \tilde{l}_{c_0})}, \Delta b_{(r_{c_0}, \tilde{w}_{c_0}, \tilde{l}_{c_0})}\}$. Let $\bar{L}_{c_0}$, $\bar{a}_{c_0}$, and $\bar{b}_{c_0}$ be the average Lab channels over the control points respectively, the color $C_s : \{\tilde{L}_{c_0}, \tilde{a}_{c_0}, \tilde{b}_{c_0}\}$ of the stroke which originates from c_0 is:

$$\tilde{L}_{c_0} = \bar{L}_{c_0} \tag{7.4}$$

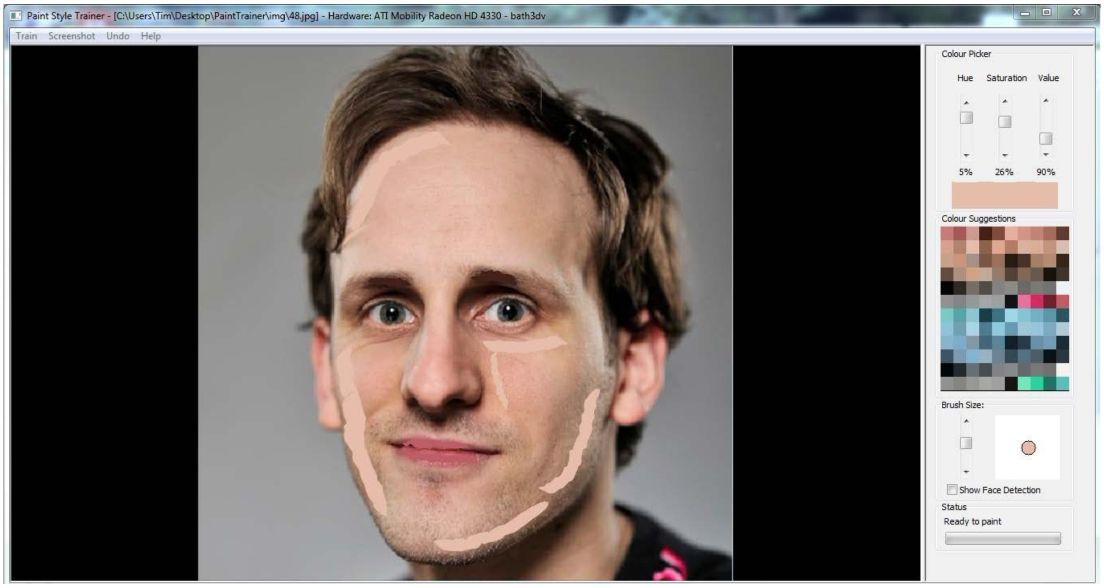
$$\tilde{a}_{c_0} = \bar{a}_{c_0} + \Delta a_{(r_{c_0}, \tilde{w}_{c_0}, \tilde{l}_{c_0})} \tag{7.5}$$

$$\tilde{b}_{c_0} = \bar{b}_{c_0} + \Delta b_{(r_{c_0}, \tilde{w}_{c_0}, \tilde{l}_{c_0})}. \tag{7.6}$$

After the color transfer, C_s is converted to RGB space and the values clamped.

7.4.4 Null Models

The sparseness of the training data can result in no model being recorded for particular tuple (null model). Tuples coded to null models may be encountered during rendering, and it is necessary to perform a lookup to identify the closest tuple with a model. This



Paint Style Trainer - [C:\Users\Tim\Desktop\PaintTrainer\img\48.jpg] - Hardware: ATI Mobility Radeon HD 4330 - bath3dv

Colour Picker

Hue	Saturation	Value
5%	26%	90%

☐ Show Face Detection

Status

Ready to paint

Figure 7.6: A snapshot of the user interface to capture brush strokes of artist, which consists of canvas (left), adjustable colour picker (top-right), colour palette (middle-right) and brush size (bottom-right).

also promotes spatially coherent appearance. For color transfer we must find the nearest tuple in $\{\mathcal{R} \times \mathcal{W} \times \mathcal{L}\}$ space; for the remainder of the properties in $\{\mathcal{R} \times \mathcal{W}\}$ space.

Prior to rendering, we pre-compute the nearest neighbor assignments for codewords $\mathcal{W}$ independently within each facial region $\mathcal{R}$. A matrix of distances between the cluster centers used to generate codebook $\mathcal{W}$ is computed to form an undirected graph. Distances incorporating a codeword for which no tuple has been learned for the particular region are zeroed. Having computed the matrix, the nearest D non-null models for each codeword can be computed efficiently at runtime by examining the D smallest values in the matrix for a given codeword (achieved by sorting each row of the matrix). Given a tuple (r, w) with null model encountered during painting, we consult this list to look-up nearby non-null models in the tuple space and average their Gaussian parameters. In practice, setting $D = 1$ is sufficient to ensure good coherence.

In the case of color transfer, we must lookup based on $\mathcal{L}$ for a given null tuple. This adds another layer of search to the process. We first find the nearest codeword using the matrix, specifically the (r, w) for which any models $(r, w, \mathcal{L})$ have been sampled.

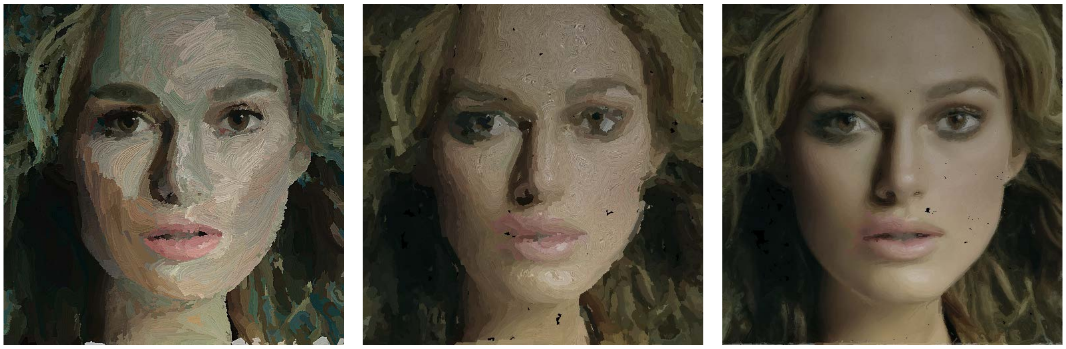


Figure 7.7: Portrait rendering result of our proposed algorithm (left), result of Hertzmann's multi-layer algorithm with brush radii $R = \{16, 8, 4\}$ (middle) and $R = \{8, 4, 2\}$ (right). Zoom to 400% to view details.

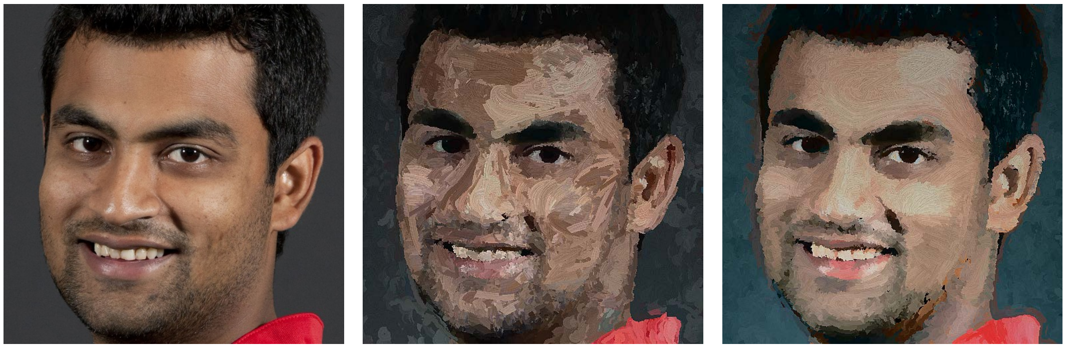


Figure 7.8: Comparison of the rendered portraits of source image (left) between our algorithm using hard quantized (middle) codeword and intensity features, and our proposed MRF approach (right). Zoom to 400% to view details.

With (r, w) fixed we identify the D closest non-null models and average their vector displacements in the color space.

7.5 Results and Discussion

We evaluate the Digital Raphael system, demonstrating improved preservation of salient features over the state of the art [89, 232] and the broad gamut of styles achievable. Our system is trained using a single photograph/painting pair. Brush strokes are captured using a bespoke user interface resembling a basic Photoshop-like painting environment

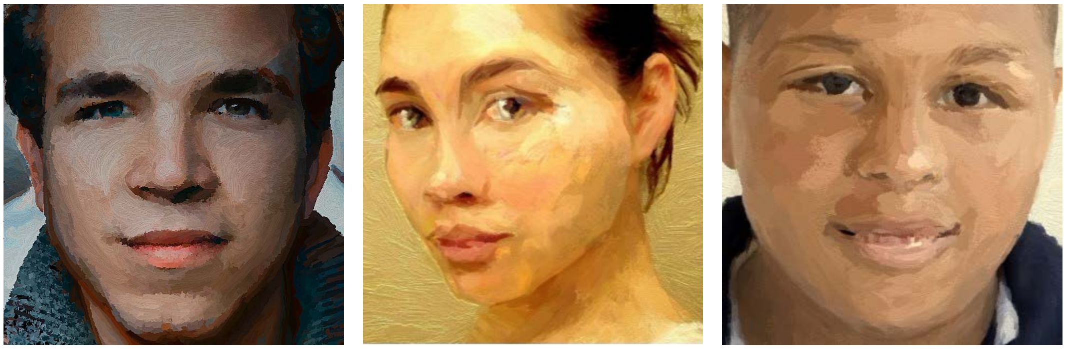


Figure 7.9: Comparison of the rendered portraits between our painting algorithm (left) and Zhao and Zhu [232] (middle and right) which warps pre-painted artist strokes to new faces. Zoom to 400% to view details.

with stroke color and size selection. A snapshot of the user interface is shown in Fig. 7.6. Strokes are painted on top of the original training photograph to provide a point of reference. The system also generates a color palette incorporating dominant colors in the photograph, and their complementary colors, by clustering RGB pixel values.

7.5.1 Comparison with Baseline Methods

We first compare our rendering algorithm with the general purpose multi-layer multi-scale painterly rendering algorithm by Hertzmann [89], commonly used as a baseline for painterly stylization. Fig. 7.7 provides a visual comparison, showing that this generic painting method either destroys important facial details through blurring or over-paint Fig. 7.7 (middle) or produces photorealistic renderings without insufficient painterly effect Fig. 7.7 (right) that tend back towards photorealism. Despite the multi-resolution approach taken in [89] to capture structure of different scales, the lack of higher-level semantic structure and stroke properties that adapt to local features causes loss of salient detail. The training image for our rendering is given in Fig. 7.11e, and is a very rough user depiction of the face using broad colored strokes. Even with such minimal, coarse training data our specialized portrait algorithm is able to produce high quality renderings that stylish salient facial details without loss of clarity.

Fig. 7.9 compares our results with Zhao and Zhu [232] where artist strokes are reproduced

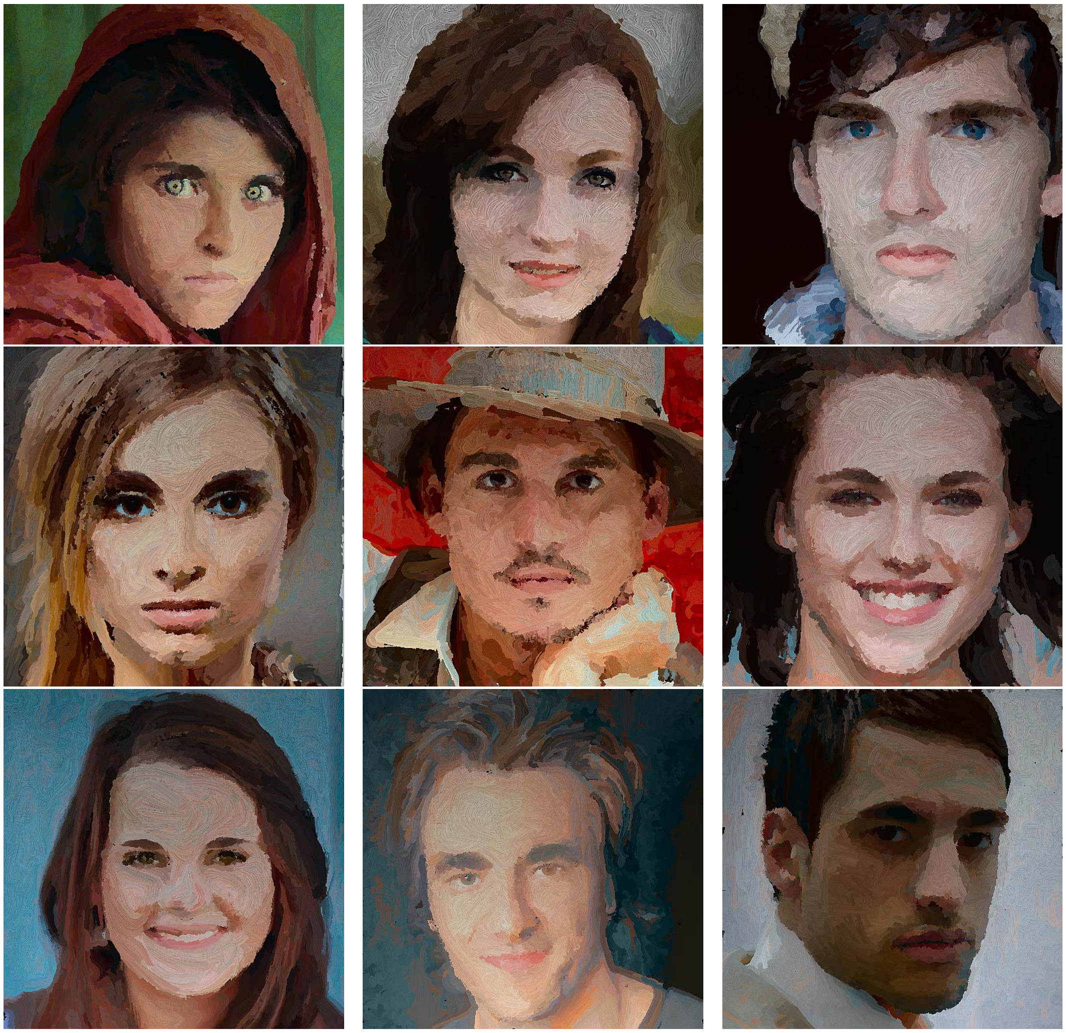
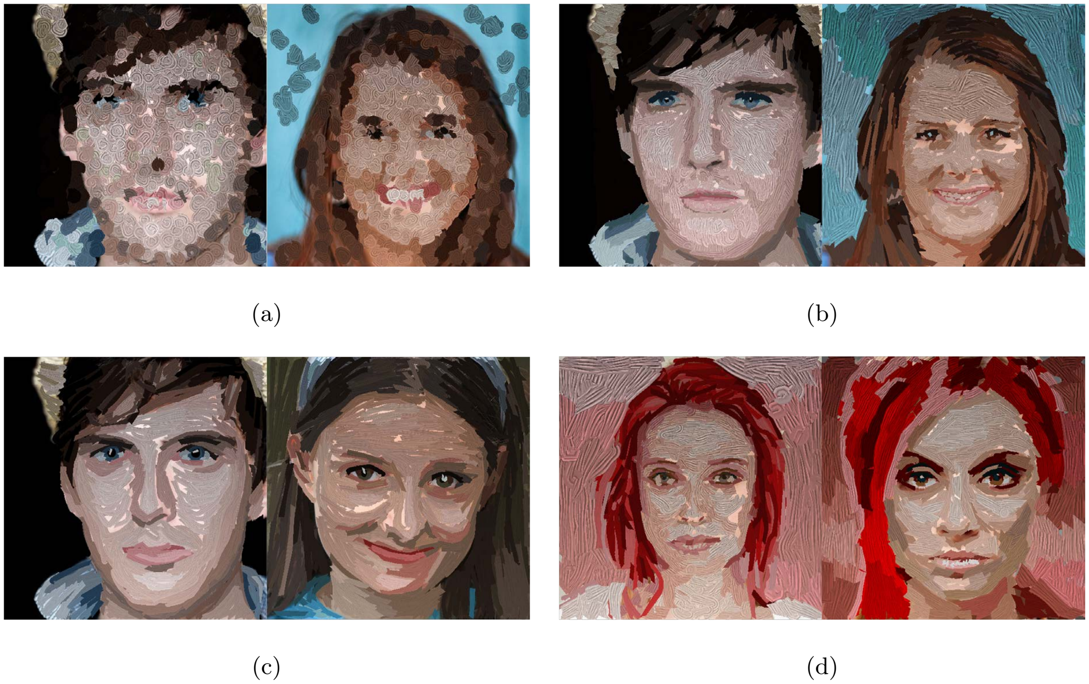


Figure 7.10: Portrait renderings by the proposed algorithm. Zoom to 400% to view details.

'verbatim' and warped to fit the face. This stroke-warping approach relies upon on a global facial model but, as the stroke positions are prescribed rather than algorithmically generated, they are unable to depict visual structures such as shadows or wrinkles. Detail in regions such as the eyes distinguishes our system's capability to adapt to delicate facial features. The phenomenon of "ghost teeth"[232] on the lips in Fig. 7.9 (right) is similarly caused by re-using captured strokes rather generating them. By contrast the result from our system (learned from 7.13e, with teeth) successfully depicts the delicate in eyes but also deals with changes in geometry, transferring the smiling face


(a)

(b)

(c)

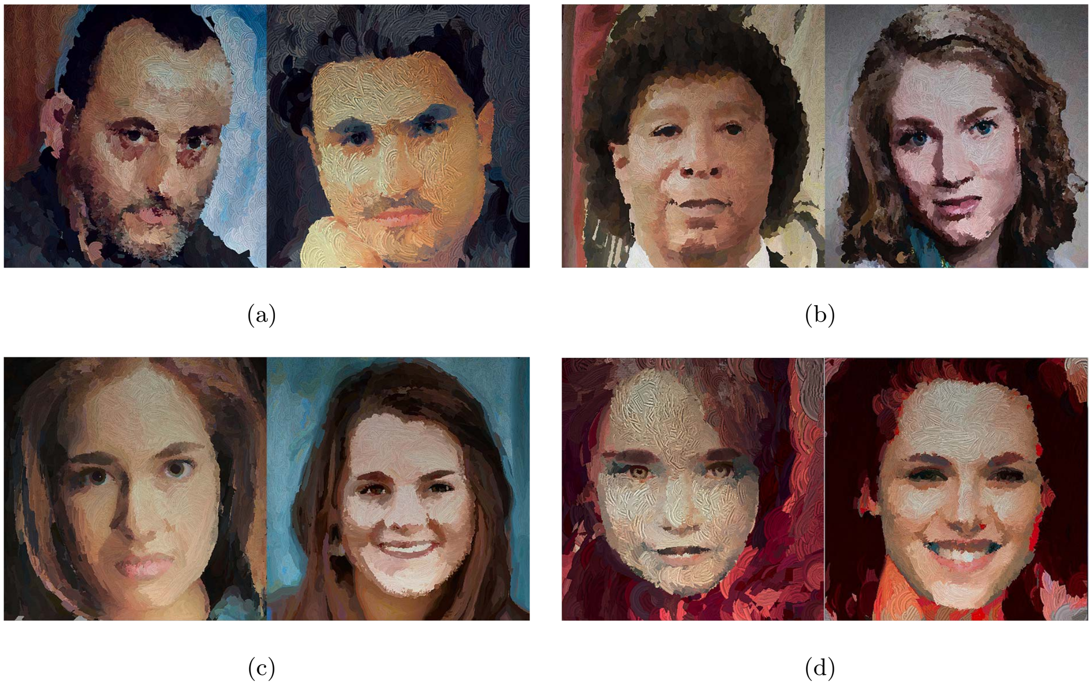
(d)

Figure 7.11: Training paintings of four styles: (a) Blobby strokes; (b) Medium strokes; (c) Long strokes; (d) Warm color. See corresponding renderings in Fig. 7.12.

with teeth to faces with no teeth showing. Note [232] also does not provide a solution to render hair, as we do, and paints this as a post-process.

7.5.2 Evaluation of MRF Style Coherence

Under our proposed learning and rendering framework, we evaluate the benefit of our MRF based feature composition which enforces spatial coherence in codeword ($\mathcal{W}$) and intensity ($\mathcal{L}$). Optimizing these fields to enforce spatial coherence directly influences the coherence of $\mathcal{P}$. A simpler alternative is simply to perform quantization of SIFT features according to the learned codebook, with no spatial coherence constraint under an MRF. Fig. 7.8 (middle) shows that the labels obtained using such quantization feature introduces spatial incoherence in the rendered style, whilst our MRF based approach (Fig. 7.8, right) to feature composition exhibits significantly improved aesthetics driven by the improved smoothness in variation of stroke parameters.



(a)

(b)

(c)

(d)

Figure 7.12: Portrait renderings using models learned from Fig. 7.11 (a) Blobby strokes; (b) Medium strokes; (c) Long strokes; (d) Warm color. Zoom to 400% to view details.

7.5.3 Style Repeatability and Diversity

By sampling models of stroke properties sourced from artists, our system is able to render a diverse range of painting styles unavailable to prior work. In order to evaluate these learning capabilities we asked a group of 10 people with varying levels of artistic experience, from professionally trained artists to amateurs, to produce various styles of portraits using our system. Fig. 7.11 shows representative training portraits covering four styles. We present two training paintings per style to verify the style consistency of learned models. The corresponding rendering results are in Fig. 7.12. Fig. 7.13 shows additional training portraits of six styles to further demonstrate the ability of our algorithm to learn a diverse range of styles and Fig. 7.14 presents the corresponding rendering results.

Fig. 7.11(a) and Fig. 7.12(a) show a couple of training examples and rendering results respectively demonstrating a painting style using thick, short strokes as color blobs to increase the abstract level, where the stroke length and thickness are correctly learned with regard to different local context without destroying the details. Fig. 7.11(b) uses medium size strokes to depict fine details of portrait, for instance the curly hair in Fig. 7.12(b). The long hair in Fig. 7.12(c) is featured by the learned long strokes from Fig. 7.11(c). Fig. 7.11(d) trains a style characterized by warm colors. The rendered result is in Fig. 7.12(d). The rendering results from every two training paintings of the same style exhibit similar stroke properties which demonstrate the style consistency during learning.

In contrast to Fig. 7.11(d), Fig. 7.13(a) and Fig. 7.14(a) use cold color to render the shade areas. Fig. 7.13(b) and Fig. 7.14(b) adopts a similar way to emphasis the shade areas; darker areas are painted with a purple tint and the relatively lighter areas are painted with the complementary color of orange. Only shadows with similar visual structure to the training image are painted in complementary color.

Fig. 7.14(c) demonstrates the ability to learn Pointillist rendering from training painting Fig. 7.13(c) using small, distinct strokes of pure color. We use pairs of colors for light/shade areas to train – orange/blue on facial area, and green/yellow for non-facial areas. Fig. 7.14(c) shows that the color transfer algorithm can preserve coherent color contrasts around local structures to create aesthetically pleasing effects even when very short strokes are used. Training Fig. 7.13(d) causes thick strokes with medium length to increase the abstract level in Fig. 7.14(d). Fig. 7.13(e) uses natural color and fine strokes to depict human portrait (Fig. 7.14(e)). Fig. 7.13(f) trains hair texture using long strokes which results in Fig. 7.14(f), benefiting from the tensor based orientation field.

Fig. 7.15 shows the rendering results on selected test images using models learned from the same images. The purpose of this experiment is to demonstrate that a painting used to train a system may be approximately reproduced from the learned style parameters. The rendering results exhibit similar styles, e.g. color tones, stroke orientation, thickness, and length, with the corresponding training paintings shown in Figs. 7.11(c) and 7.13(e). For example, the horizontal strokes on the the girl's cheek (Fig. 7.15(c)) in Fig. 7.15(a) have similar styles in Fig. 7.11(c); the style to depict the shade areas of hair and skin, even the eyelash (Fig. 7.15(d)) in Fig. 7.15(b), exhibits high similarity with the corresponding training painting Fig. 7.13(e). However note that we are not trying to reconstruct the training painting exactly, which is only possible by simply warping the training strokes as Zhao and Zhu [232] did — rather we are seeking similarity at a higher conceptual level, aiming to reconstruct the visual style of the training image.

Fig. 7.13 and Fig. 7.14 provide further examples of trained and reproduced styles. The variation of stroke orientation, thickness, length and color across the rendering results distinguishes each of the styles.

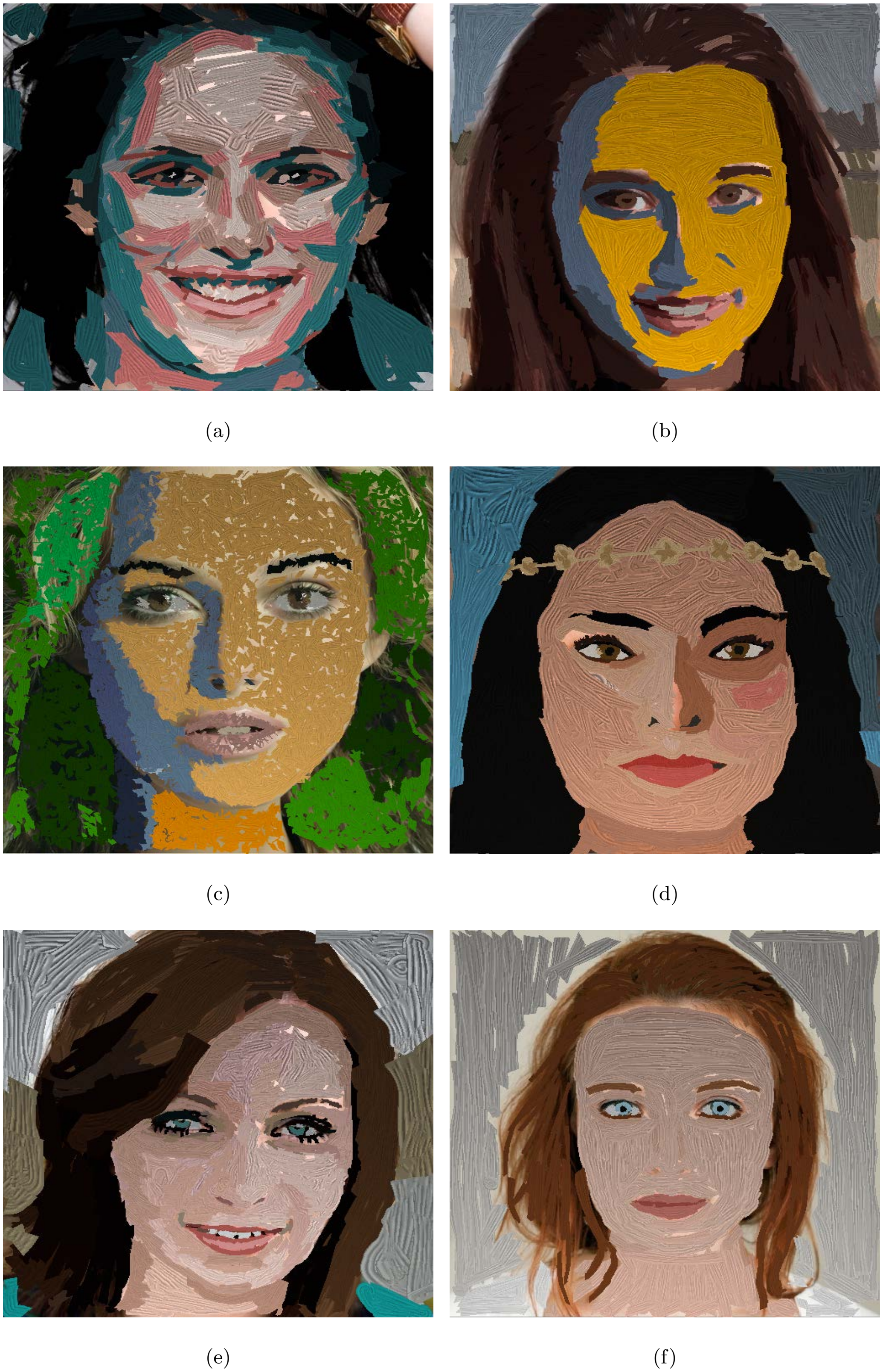
7.5.4 Learning with Sparse Training Data

As a final experiment we demonstrate that our algorithm is able to effectively learn certain styles of painting even when only limited training data is available. This may be the case in amateur paintings or where the user indicates only a few examples of style style per facial feature. The corresponding rendering results are shown in Fig. 7.17, where the learned stroke density is artificially scaled up for aesthetics, i.e. to deliver a full painting given the sparsity of strokes in the partial training example. Here the style variation within a facial feature is rather uniform, given the sparsity of training data. Nevertheless different stroke styles (round swirls, versus long curves) are learned and extrapolated over the image.

7.6 Conclusion

We have presented a user trainable algorithm for stylizing photographs into portrait paintings. This challenging Computer Graphics problem is addressed using Computer Vision to learn a flexible stroke model of artistic style by analyzing the global and local geometry as well as tone of brush strokes placed local to image features. Our facial feature representation (Sec. 7.2) which not only accounts for global structure and higher-level semantics but also encodes local context and low-level visual structure, enabling a wide variety of artistic styles to be encapsulated in one system. Further, our system is able to generalize from minimalist training data consisting of few strokes, to produce high quality paintings from data generated by users without extensive artistic training. Portraiture has previously proven to be a challenging domain for painterly rendering algorithms. Our solution is also able to depict faces without the loss of salient detail exhibited by more general painterly methods [89] and without relying on a pre-painted arrangement of strokes to warp over the face [232]. Rather, we compute the position of strokes as a function of image content.

In this chapter, we have demonstrated that modeling a visual structure of human facial features enabled portrait painting. This representation of visual structure underpinned domain specific models which drive effective per-part learning of how artists paint and thus provide more precise control over the portrait rendering without introducing any distortion or loss of details of facial feature. In future work, we would like to improve the accuracy of facial region parsing by introducing the 3D morphable model [19] which is capable to reconstruct 3D facial geometry from single image. Stroke style models learned based on 3D meshes may capture more accurate facial geometry, which conveys strong impression of 3D structures and enables pose-invariant portrait painting. The latter may serve the purpose of synthesising painterly facial animation or producing high quality painterly animation from video containing face.



(a)

(b)

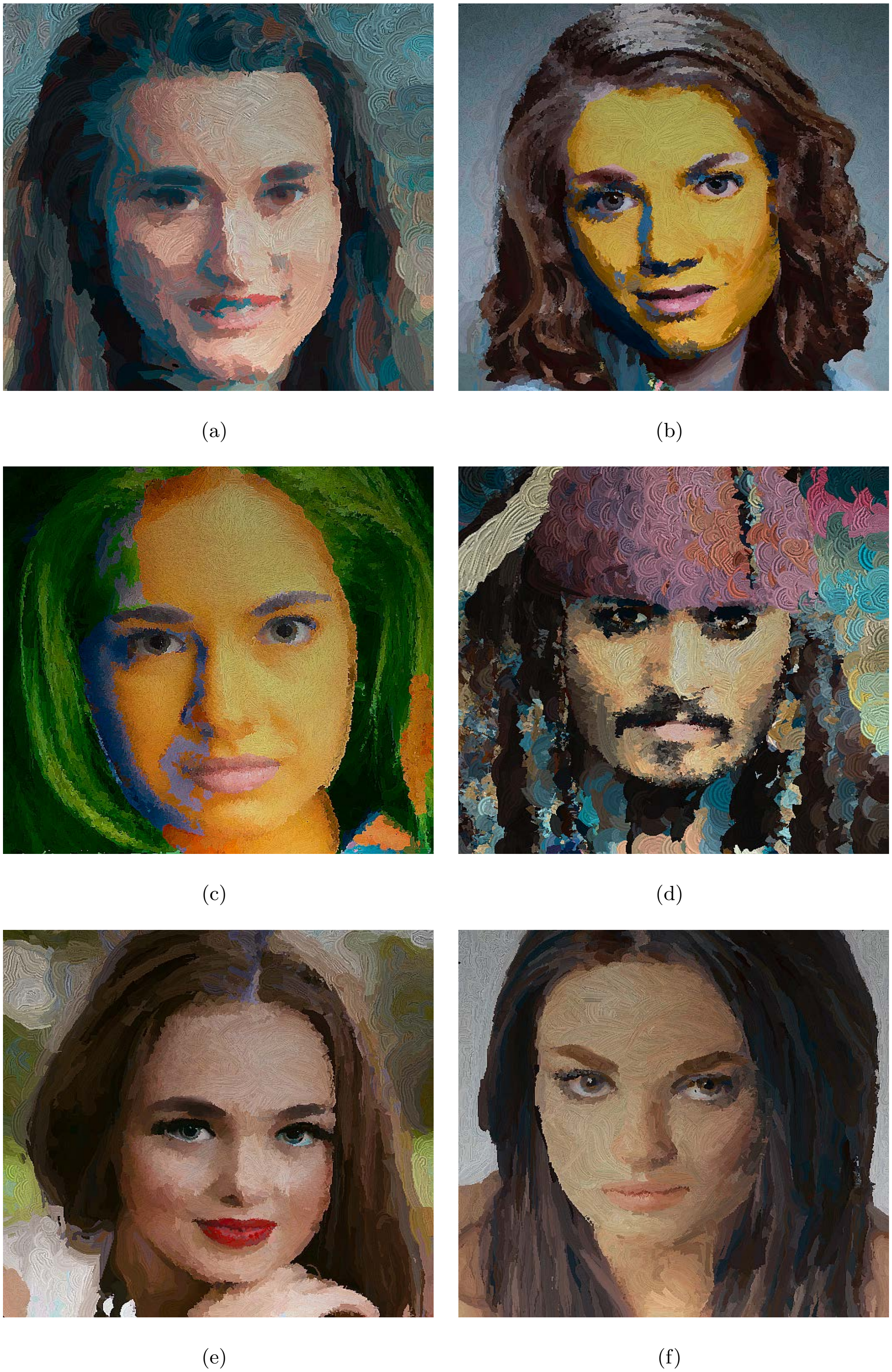
(c)

(d)

(e)

(f)

Figure 7.13: Additional training paintings of various styles: (a) Cold color; (b) Shading using complementary color; (c) Short strokes; (d) Thick, long strokes; (e) Natural color; (f) Long strokes for hair. See corresponding renderings in Fig. 7.14.



(a)

(b)

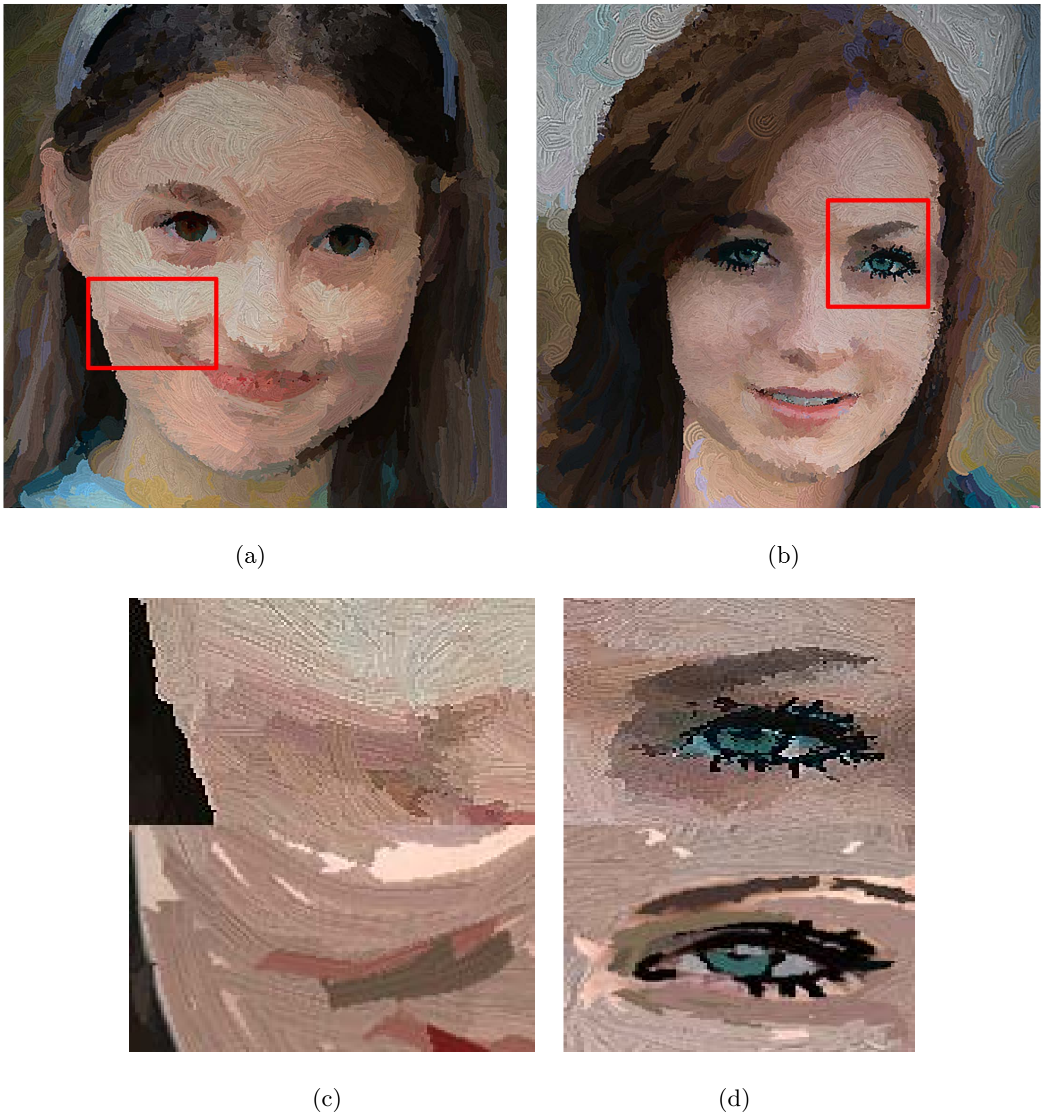
(c)

(d)

(e)

(f)

Figure 7.14: Portrait renderings using models learned from Fig. 7.13 (a) Cold color; (b) Shading using complementary color; (c) Short strokes; (e) Thick, long strokes; (d) Natural color; (f) Long strokes for hair. Zoom to 400% to view details.



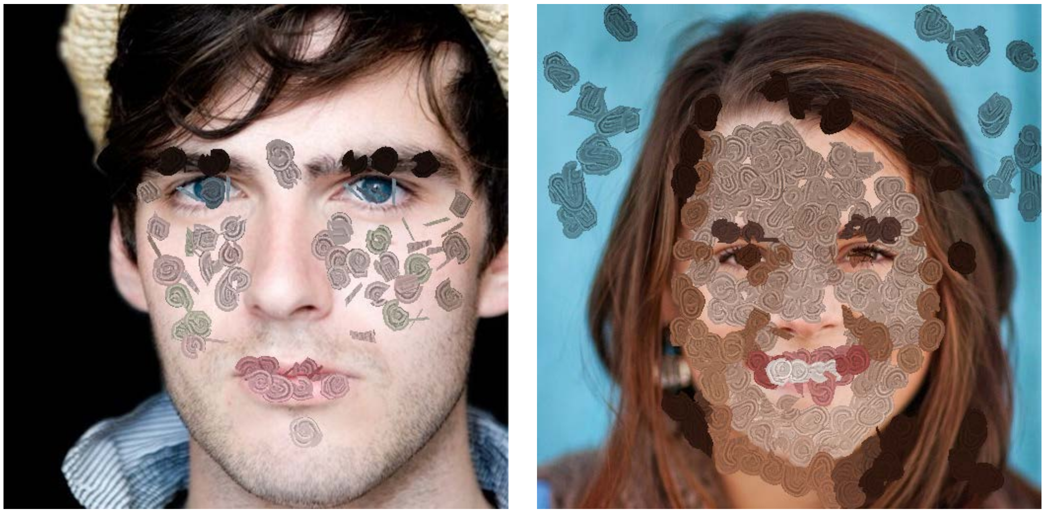
(a)

(b)

(c)

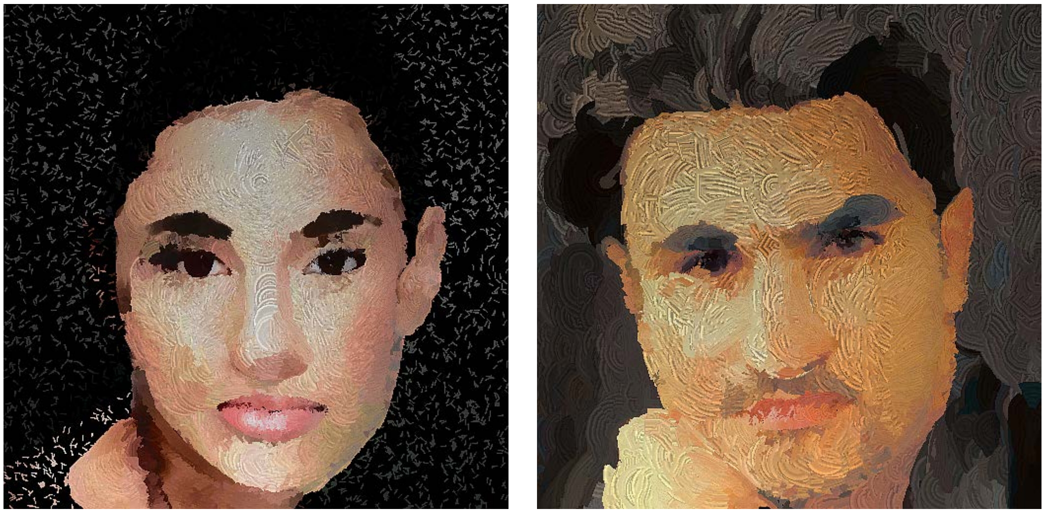
(d)

Figure 7.15: Portrait renderings of testing images using models learned from the same images (a) Long strokes from Fig. 7.11(c); (b) Natural color from Fig. 7.13(e); (c) Zoomed in section in (a) and the corresponding section in training painting Fig. 7.11(c) (d) Zoomed in section in (b) and the corresponding section in training painting Fig. 7.13(e). Zoom to 400% to view details.



(a) (b)

Figure 7.16: Partial training paintings taken from the process producing Fig. 7.11(a). Zoom to 400% to view details.



(a) (b)

Figure 7.17: Portrait renderings using models learned from partial training paintings Fig. 7.16. Stroke densities are scaled up to approximate a full painting learning in facial area. Zoom to 400% to view details.

Part V

Conclusions

Chapter 8

Conclusions and Further Work

In this chapter we summarise the contributions of the thesis, and discuss how the results of the algorithms we have developed support our central argument for the use of representation of visual structure in the stylisation and presentation of visual media. We suggest possible avenues for the future development of our work.

8.1 Summary of Contributions

This thesis addressed the problem of structured representation and stylisation for visual media collection; proposing algorithms for structurally representing, analysing and stylising the visual content. We performed a comprehensive review of related work and argued that the structured representation proves beneficial in terms of improving the understanding of visual media, and broadening the gamut of potential expressive styles. To support this argument we developed several novel algorithms which operate at different level of representations and render a wide range of expressive styles on image and video.

We now examine the algorithms developed in this thesis, and conclude as to how their results contributed to our central argument for structured representation and stylisation in visual media. Specifically we identify the improvements gained in each of the following areas.

8.1.1 Automatic Video Editing

In Chapter 3 we described a novel tree representation to bridge the gap between the low-level feature and high-level video editing operations, suitable for use in a Genetic Programming (GP) optimization framework. Our structured representation incorporates cutting, zooming and panning operations, which uniquely facilitates the search for a globally optimal video edit using GP, maximising both aesthetics and interest within the final clip. Our measures for aesthetics are grounded in common directing practice, and our measure for interest is based on the presence of people; the most common subject of interest for home videos. To capture the subjectivity of video aesthetic, our fitness function is governed by user parameters weighting desire for objects of interest against frequency of cuts, and motion. This system proved to be efficient over some representative examples of home video footage. The short optimization times enable user experimentation to taste. Our approach of video editing via defining a structured representation of editing operations increases the aesthetic value and interest in medium items, which supports our hypothesis in Sec. 1.1

- **H2.** Structured presentation and visual stylization of content in personal media collections enhances user engagement with that content.

8.1.2 Interactive Object Segmentation

In Chapter 4 we presented an object segmentation system driven by single finger touch using level set methods which integrates both low- and mid-level models of colour, texture and geometry. The core contribution is an edge-region-geometry based segmentation model to robustly tackle the interactive object segmentation problem — encoding boundary probabilities of color-texture homogeneous regions, and the statistical and geometric priors inferred from the user input. The proposed edge model observes the local colour distribution and thus provides robust description of the coherent colour-texture region which mitigates against the contour becoming stuck in local minima in the presence of noisy data. Colour information from user input augments this model, balancing the *a posterior* probabilities of region models inside and outside the putative object contour. We also demonstrated that our algorithm can be extended to segment video sequences into temporally coherent foreground and background region maps. We introduced a motion estimation enabled shape prior into the video adaptation to preserve temporal coherence when the foreground and background color distributions are indistinct. This gives rise to potential applications to video special effects (e.g. artistic stylization) with minimal user intervention, that may be suited to consumer touch-screen video cameras. This supports our argument that stable structures extracted from visual media facilitate the spatially localized media manipulation. A comprehensive comparison with previous techniques was presented, demonstrating the effectiveness of the proposed system at achieving high quality results, as well as the robustness of the system against limited inputs. This supports our hypothesis

- **H1.** Improving the stability of the structure extracted from video sequences beyond the state of the art enhances the temporal coherence of artistic renderings.

8.1.3 Video Segmentation

In Chapter 5 and Chapter 6 we presented two video segmentation algorithms to apply multi-label graph cut on successive frames, in which the segmentation of each frame is driven by motion flow propagated labelling priors and incrementally updated data model estimated from the past frames to improve the temporal coherence. The flow-propagated labels in the first algorithm are assumed to be hard constraints i.e. perfect estimates. The second algorithm also follows a flow-propagation strategy, but adopts ‘soft’ constraints on motion propagated priors.

Our core contribution in the latter was a multi-frame probabilistic motion diffusion model to incorporate labelling priors from previous frames to influence the segmentation in new frame. Uniquely this diffusion model propagated a *per-pixel distribution of labelling priors* forward based on the probability distribution of motion vectors for that pixel. Motion flow estimation remains a challenging open problem in Computer Vision, and our approach mitigates against inaccuracy in such estimates via this “soft” propagation strategy. This was shown to improve temporal coherence over hard-assignment strategies in our first algorithm, graph based schemes based on flow propagation [76] and spatio-temporal segmentation [161].

We combined this motion framework with a spatially 'higher order' constraint additionally imposing the soft label consistency constraint across image regions (super-pixels) obtained via various unsupervised segmentations — as is now common in image segmentation. By enforcing labelling consistence, both the spatial coherence and boundary accuracy of the segmentation was enhanced (demonstrated via comparison to a manually labelled ground truth).

Our novel video segmentation algorithms drive video stylization algorithms using mid-level representations of video parsed from footage. This representation allows us to establish correspondence between frames, enabling the coherent stylization of video objects with both shading and painterly effects, which proves our first hypothesis in Sec. 1.1

- **H1.** Improving the stability of the structure extracted from video sequences beyond the state of the art enhances the temporal coherence of artistic renderings.

8.1.4 Digital Ambient Displays of Visual Media Collections

In Chapter 5 we presented the *Digital Ambient Display* (DAD) that harnesses artistic stylization to create an abstraction of user's experiences through their home digital media collections. The DAD applies the proposed video segmentation algorithm to stylized animation. The scene structure is stably represented by parsing the video into coherent spatial segments. This representation not only enables the coherent video stylization, but also creating aesthetically pleasing video composition and transition effects between different video clips using region correspondence. This supports our hypotheses

- **H1.** Improving the stability of the structure extracted from video sequences beyond the state of the art enhances the temporal coherence of artistic renderings.

- **H3.** New approaches for parsing visual structure can unlock new forms of stylization so diversifying AR.

A further contribution of the thesis is a novel approach to structuring and navigating visual media collections. We described an algorithm for adaptively sequencing media items using graph optimization in a coarse-to-fine manner driven by user attention. By recursively clustering media items into a hierarchy, we were able to plan routes within clusters to display content of a common theme. We were also able to plan routes between clusters to summarise media within the collection. We deployed our system on dedicated hardware and undertook a small-scale user trial to validate the our content sequencing algorithm based on structured representation, which was shown to be more engaging than random photo slideshows, which supports our hypothesis

- **H2.** Structured presentation and visual stylization of content in personal media collections enhances user engagement with that content.

8.1.5 Portrait Painting

In Chapter 7 we have presented a user trainable algorithm for stylizing portrait photographs into paintings. Portraiture has previously proven to be a challenging domain for painterly rendering algorithms. This challenging Computer Graphics problem is addressed using Computer Vision to form a structured representation of facial feature and learn a flexible non-parametric model of artistic style by analyzing the global and local geometry as well as tone of brush strokes placed local to image features. Our structured facial feature representation which not only accounts for global structure and higher-level semantics but also encodes local context and low-level visual structure, enabling a wide variety of artistic styles to be encapsulated in one system. This supports our hypothesis

- **H3.** New approaches for parsing visual structure can unlock new forms of stylization so diversifying AR.

Further, our system is able to generalize from minimal training data consisting of few strokes, to produce high quality paintings from data generated by users without extensive artistic training.

8.2 Future Work

The techniques proposed in this thesis have raised a number of interesting possibilities for future work. Many of these have already been discussed in the conclusion section of the relevant chapters as they address specific incremental improvements which could be made to particular algorithms. In this section we highlight the more general directions which appear to hold the greatest potential.

Video segmentation is an under-constrained task, and there are generally two methods of introducing sufficient constraints to make the problem tractable; prescribed user heuristics and interaction. Heuristics refer to loosely applicable strategies or rules to control problem solving, and are assumed to be qualitatively applicable over all classes of footage processed. Interaction enables, by controls, constraints special to the piece of footage being processed. In this thesis, we have defined a variety of heuristics to underpin video segmentation, e.g. the propagation of motion and appearance priors enables the consistency and coherence of frame-by-frame segmentation. However, the task is still under-constrained since, in the absence of high level scene understanding, there can be more than one interpretation of pixels comprising the desired object of interest. The past decade has seen a trend toward better constraining the video segmentation task through the combination of high-level prior scene understanding via user interaction with low-level cues such as color, edges and motion observed in the sequence, sacrificing some level of automations. A balance between heuristics and interactions is one of the possible directions of our future work on video segmentation, which still remains limit in the specialism of temporal constraints interactions. For instance, our video segmentation algorithms might be driven by a scribble drawn by the user to indicate the rough trajectory of object movement to enhance the heuristics of motion propagation defined in the system. Such a motion scribble driven approach to video segmentation can be further enhanced by incorporating "occlusion boundaries" [199] discovered from motion disparity in the scene, and using these to compensate for any ambiguity in appearance between regions.

In Ch. 5 we have presented a novel hierarchical presentation for structuring media collections which facilitates the adaptive coarse-to-fine sequencing of media items driven by user attention. In our proof-of-concept system, we assumed there was only one single context in the media collections, e.g. "a trip to London". However, there might be more than one context in real world domestic media collections, in which case we would need to virtually structure the media collection into multiple but possibly overlapping hierarchies. We would like to investigate how these hierarchies may further constitute a graph consisting of nodes and edges, where nodes are hierarchies and edges measure the level of overlapping between hierarchies. A random walk over the graph sequences the media items belonging to single or multiple contexts.

We have proposed an approach to stylising portrait photographs into paintings in Chapter 7; we trained at the level of the stroke, learning how the placement and appearance of each brush stroke is modulated according to underlying features in the training image. Although integrating the evidence from depth cues to form a consolidated model of the visual world comes naturally to us, depth information has not played a major role in artistic rendering (though some progress has begun to be made [29]). We can speculate that exploiting the depth information from 3D geometry would significantly improve the robustness of learning portrait painting. Stroke style models learned based on 3D meshes may capture more accurate facial geometry, which conveys strong impression of 3D structures and enables pose-invariant portrait painting, without suffering from the visual ambiguities in 2D imagery. Learning portrait paintings from 3D geometry would open the door to synthesising portrait painterly animation or producing high quality painterly animation from video containing faces. General video stylisation algorithms perform particularly poorly on faces, which forms a barrier for domestic user to enhance the aesthetic value of their medium collection with expressive presentation forms. This would be another interesting question to be addressed in our future work. By building correspondence of 3D geometry between successive frames, stroke models learned taking advantage of depth information would generate stably morphing brush strokes exhibiting strong temporal coherence.

Throughout this thesis we have developed image and video segmentation algorithms for monocular view besides our portrait painting algorithm. Due to the increasing amount of stereoscopic 3D data now being produced, there have been segmentation algorithms to handle the multiple view data [168, 77]. Our monocular solutions can easily be extended to multiple view. Taking the video segmentation algorithm proposed in Ch. 6 for example, we can introduce inter- and intra-view spatial smoothness constraint correlating the local spatial coherence among multiple views derived from both object-like regions and dense feature matching. This approach would improve the spatio-temporal coherence and preserve the multiple view consistency which might be a solution for multiple view matting - a major application area for video segmentation, e.g in the creative industries.

Bibliography

[1] D. Adalsteinsson and J. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, pages 269–277, 1995.

[2] A. Agarwala, A. Hertzmann, D. Salesin, and S. M. Seitz. Keyframe-based tracking for rotoscoping and animation. *ACM Trans. Graph.*, 23(3):584–591, 2004.

[3] M. A. Ahmed, F. Pitie, and A. Kokaram. Extraction of non-binary blotch mattes. In *Proc. ICIP*, pages 2757–2760, 2009.

[4] M. Al-Hames, B. Hörnler, R. Müller, J. Schenk, and G. Rigoll. Automatic multimodal meeting camera selection for video-conferences and meeting browsers. In *Proc. ICME*, pages 2074–2077, 2007.

[5] K. Alahari, P. Kohli, and Torr. Reduce, reuse & recycle: Efficiently solving multi-label mrfs. In *Proc. CVPR*, pages 1–8, 2008.

[6] P. Arbelaez and L. D. Cohen. Constrained image segmentation from hierarchical boundaries. In *Proc. CVPR*, pages 1–8, 2008.

[7] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011.

[8] D. Arijon. *Grammar of the Film Language*. Silman-James Press, 1991.

[9] N. Arksey. Exploring the design space for concurrent use of personal and large displays for in-home collaboration. Master's thesis, University of British Columbia, Aug, 2007.

[10] C. Armstrong, B. Price, and W. Barrett. Interactive segmentation of image volumes with live surface. *Computers & Graphics*, 31(2):212–229, 2007.

[11] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *Proc. ICCV*, pages 1–8, 2007.

[12] X. Bai, J. Wang, and G. Sapiro. Dynamic color flow: A motion-adaptive color model for object segmentation in video. In *Proc. ECCV*, pages 617–630, 2010.

[13] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. In *Proc. SIGGRAPH*, pages 1–11, 2009.

[14] J. A. Bangham, S. E. Gibson, and R. Harvey. The art of scale-space. In *Proc. BMVC*, pages 569–578, 2003.

[15] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and texture-based image segmentation using em and its application to content-based image retrieval. In *Proc. ICCV*, pages 675–682, 1998.

[16] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:509–521, 2002.

[17] M. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proc. ICCV*, pages 231–236, 1993.

[18] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. In *Proc. ECCV*, pages 428–441, 2004.

[19] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proc. SIGGRAPH*, pages 187–194, 1999.

[20] A. Bousseau, F. Neyret, J. Thollot, and D. Salesin. Video watercolorization using bidirectional texture advection. *ACM Trans. Graph.*, 26(3):104:1–7, 2007.

[21] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *Int. J. Comput. Vision*, 2(70):109–131, 2006.

[22] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *Proc. ICCV*, pages 105–112. IEEE, 2001.

[23] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26:1124–1137, 2004.

[24] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23:1222–1239, 2001.

[25] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *Proc. ICCV*, 2009.

[26] X. Bresson, P. Vandergheynst, and J. Thiran. A priori information in image segmentation: Energy functional based on shape statistical model and image information. In *Proc. ICIP*, pages 425–428, 2003.

[27] T. Brox and D. Cremers. On the statistical interpretation of the piecewise smooth mumford-shah functional. In *Proc. SSVM*, pages 203–213, 2007.

[28] T. Brox and J. Weickert. Level set segmentation with multiple regions. *IEEE Trans. on Image Process.*, 15(10):3213–3218, 2006.

[29] Richardt C. *Colour videos with depth: acquisition, processing and evaluation.* PhD thesis, Cambridge University, March 2012.

[30] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[31] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proc. ICCV*, pages 694–699, 1995.

[32] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Process.*, pages 266–277, 2001.

[33] H. Chen, Z. Liu, C. Rose, Y. Xu, H.-Y. Shum, and D. Salesin. Example-based composite sketching of human portraits. In *Proc. NPAR*, pages 95–153, 2004.

[34] H. Chen, N.-N. Zheng, L. Liang, Y. Li, Y.-Q. Xu, and H.-Y. Shum. Pictoon: a personalized image-based cartoon system. In *Proc. MM*, pages 171–178, 2002.

[35] J. Chen, C.A Bouman, and J. C. Dalton. Hierarchical browsing and search of large image databases. *IEEE Trans. Image Process.*, 9:442–455, 2000.

[36] P. Chockalingam, S. Nalin Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *Proc. ICCV*, pages 1530–1537, 2009.

[37] P. Chopra and J. Meyer. Modeling an infinite emotion space for expressionistic cartoon face animation. In *Computer Graphics and Imaging*, pages 13–18, 2003.

[38] J. Collomosse. *Higher Level Techniques for the Artistic Rendering of Images and Video*. PhD thesis, University of Bath, May 2004.

[39] J. Collomosse and P. M. Hall. Painterly rendering using image salience. In *Proc. EGUK*, pages 122–128, 2002.

[40] J. Collomosse and P. M. Hall. Cubist style rendering from photographs. *IEEE Trans. Vis. Comput. Graph.*, 4(9):443–453, 2003.

[41] J. Collomosse and P. M. Hall. Genetic paint: A search for salient paintings. In *Proc. EvoMUSART*, pages 437–447, 2005.

[42] J. Collomosse, D. Rowntree, and P. M. Hall. Stroke surfaces: Temporally coherent artistic animations from video. *IEEE Trans. Vis. Comput. Graph.*, 11(5):540–549, 2005.

[43] J. P. Collomosse, G. McNeill, and Y. Qian. Storyboard sketches for content based video retrieval. In *Proc. ICCV*, 2009.

[44] J. P. Collomosse, D. Rowntree, and P.M. Hall. Stroke surfaces: Temporally coherent artistic animations from video. *IEEE Trans. Vis. Comput. Graph.*, 11:540–549, 2005.

[45] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:603–619, 2002.

[46] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[47] T. T. A. Combs and B. B. Bederson. Does zooming improve image browsing? In *Proc. DL*, pages 130–137. ACM, 1999.

[48] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape modelstheir training and application. *Comput. Vis. Image Underst.*, 61:38–59, January 1995.

[49] D. Cremers, F. R. Schmidt, and F. Barthel. Shape priors in variational image segmentation: Convexity, lipschitz continuity and globally optimal solutions. In *Proc. CVPR*, 2008.

[50] C. Curtis, S. Anderson, J. Seims, K. Fleischer, and D. Salesin. Computer-generated watercolor. In *Proc. SIGGRAPH*, pages 421–430, 1997.

[51] D. DeCarlo and A. Santella. Stylization and abstraction of photographs. In *Proc. SIGGRAPH*, pages 769–776. ACM, 2002.

[52] Y. Delignon, A. Marzouki, and W. Pieczynski. Estimation of generalized mixtures and its application in image segmentation. *IEEE Trans. Image Process.*, 6:1364–1375, 1997.

[53] D. Dementhon. Spatio-temporal segmentation of video by hierarchical mean shift analysis. In *Statistical Methods in Video Processing Workshop (SMVP)*, 2002.

[54] D. DeMenthon, V. Kobla, and D. Doermann. Video summarization by curve simplification. In *Proc. MM*, pages 211–218, 1998.

[55] Y. Deng, C. Kenney, M. Moore, and B. S. Manjunath. Peer group filtering and perceptual color image quantization. In *Proc. ISCAS*, pages 21–24, 1999.

[56] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(8):800–810, 2001.

[57] S. Dipaola. Painterly rendered portraits from photographs using a knowledgebased approach. In *Proc. SPIE Human Vision and Imaging*, 2007.

[58] P. Dollar. Supervised learning of edges and object boundaries. In *Proc. CVPR*, pages 1964–1971, 2006.

[59] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.

[60] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, 2004.

[61] V. Ferrari, M. J. Marín-Jiménez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proc. CVPR*, 2008.

[62] W. T. Freeman, J. B. Tenenbaum, and E. C. Pasztor. Learning style translation for the lines of a drawing. *ACM Trans. Graph.*, 22(1):33–46, 2003.

[63] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí. Yet another survey on image segmentation: Region and boundary information integration. In *Proc. ECCV*, pages 408–422, 2002.

[64] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[65] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

[66] A. Girgensohn, S. Bly, F. Shipman, J. Boreczky, and L. Wilcox. Home video editing made easy balancing automation and user control. In *Proc. INTERACT*, pages 464–471, 2001.

[67] A. Girgensohn, J. Boreczky, P. Chiu, J. Doherty, J. Foote, G. Golovchinsky, S. Uchihashi, and L. Wilcox. A semi-automatic approach to home video editing. In *Proc. UIST*, pages 81–89, New York, NY, USA, 2000. ACM.

[68] M. Gleicher. Image snapping. In *Proc. SIGGRAPH*, pages 183–190. ACM, 1995.

[69] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[70] J. Goldberger, S. Gordon, and H.K. Greenspan. Unsupervised image-set clustering using an information theoretic framework. *IEEE Trans. Image Process.*, 15:449–458, 2006.

[71] D. B. Goldman, B. Curless, S. Seitz, and D. Salesin. Schematic storyboarding for video visualization and editing. In *Proc. SIGGRAPH*, pages 862–871, 2006.

[72] B. Gooch, G. Coombe, and P. Shirley. Artistic vision: Painterly rendering using computer vision techniques. In *Proc. NPAR*, pages 83–90, 2002.

[73] B. Gooch, E. Reinhard, and A. Gooch. Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans. Graph.*, 23(1):27–44, 2004.

[74] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1768–1783, 2006.

[75] H. Greenspan, J. Goldberger, and A. Mayer. A probabilistic framework for spatio-temporal video representation and indexing. In *Proc. ECCV*, pages 461–475, 2002.

[76] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph based video segmentation. *Proc. CVPR*, 2010.

[77] J.-Y. Guillemaut and A. Hilton. Joint multi-layer segmentation and reconstruction for free-viewpoint video applications. *International Journal of Computer Vision*, 93(1):73–100, 2011.

[78] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *Proc. CVPR*, pages 3129–3136, 2010.

[79] P. Haeberli. Paint by numbers: Abstract image representations. In *Proc. SIGGRAPH*, pages 207–214, 1990.

[80] P. Haggerty. Almost automatic computer painting. *IEEE Comput. Graph. Appl.*, 11(6):11–12, 1991.

[81] P. M. Hall and Y. Hicks. CSBU-2004-03: A method to add gaussian mixture models. Technical report, Univ. Bath, 2004.

[82] J. Hays and I. Essa. Image and video based painterly animation. In *Proc. NPAR*, pages 113–120, 2004.

[83] J. Hays and I. Essa. Image and video based painterly animation. In *Proc. NPAR*, pages 113–120, 2004.

[84] X. He, R. S. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *Proc. ECCV*, pages 338–351, 2006.

[85] K. Heath, N. Gelfand, M. Ovsjanikov, M. Aanjaneya, and Guibas L. J. Image webs: Computing and exploiting connectivity in image collections. In *Proc. CVPR*, 2010.

[86] Paul Heckbert. Color image quantization for frame buffer display. Proc. SIGGRAPH, pages 297–307, 1982.

[87] M. Heiler and C. Schnoerr. Natural image statistics for natural image segmentation. In *Int. J. Comput. Vision*, pages 1259–1266, 2003.

[88] A. Herbulot, S. Jehan-Besson, M. Barlaud, and G. Aubert. Shape gradient for image segmentation using information theory. In *Proc. ICASSP*, pages 17–21, 2004.

[89] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proc. SIGGRAPH*, pages 453–460, 1998.

[90] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proc. SIGGRAPH*, pages 453–460, 1998.

[91] A. Hertzmann. Fast paint texture. In *Proc. NPAR*, pages 91–96, 2002.

[92] A. Hertzmann. Tutorial: A survey of stroke-based rendering. *IEEE Comput. Graph. Appl.*, 23(4):70–81, 2003.

[93] A. Hertzmann, C. Jacobs, N. Oliver, B. Curless, and D. Salesin. Image analogies. In *Proc. SIGGRAPH*, pages 327–340, 2001.

[94] A. Hertzmann and K. Perlin. Painterly rendering for video and interaction. In *Proc. NPAR*, pages 7–12, 2000.

[95] P. R. Hill, C. N. Canagarajah, and D. R. Bull. Image segmentation using a texture gradient based watershed transform. *IEEE Trans. Image Process.*, 12(12):1618–1633, 2003.

[96] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Proc. ICCV*, pages 654–661. IEEE, October 2005.

[97] T. M. Hospedales and O. Williams. An adaptive machine director. In *Proc. BMVC*, 2008.

[98] R. Hu, T. Wang, and J. Collomosse. A bag-of-regions approach to sketch-based image retrieval. In *Proc. ICIP*, pages 3661–3664, 2011.

[99] X.-S. Hua, L. Lu, and H. Zhang. Optimization-based automated home video editing system. *IEEE Trans. Circuits Syst. Video Techn.*, 14(5):572–583, 2004.

[100] M. Perďoch J. Čech, J. Matas. Efficient sequential correspondence selection by cosegmentation. In *Proc. CVPR*, 2008.

[101] M. Kagaya, W. Brendel, Q. Deng, T. Kesterson, S. Todorovic, P. J. Neill, and E. Zhang. Video painting with space-time-varying style parameters. *IEEE Trans. Vis. Comput. Graph.*, 17(1):74–87, 2011.

[102] E. Kalogerakis, D. Nowrouzezahrai, S. Breslav, and A. Hertzmann. Learning Hatching for Pen-and-Ink Illustration of Surfaces. *ACM Trans. Graph.*, 31(1), 2012.

[103] H. Kang and Seungyong Lee. Shape-simplifying image abstraction. *Computer Graphics Forum*, 27(7):1773–1780, 2008.

[104] S. Khan and M. Shah. Object based segmentation of video using color, motion and spatial information. In *Proc. CVPR*, 2001.

[105] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proc. ICCV*, pages 810–815, 1995.

[106] J. Kim, J. W. Fisher, A. Yezzi, M. Cetin, and A. S. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution. *IEEE Trans. Image Process.*, 14:1486–1502, 2005.

[107] T. H. Kim, K. M. Lee, and S. U. Lee. Nonparametric higher-order learning for interactive segmentation. In *Proc. CVPR*, pages 3201–3208, 2010.

[108] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. In *Proc. CVPR*, 2008.

[109] V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *Proc. ICCV*, pages 564–571, 2005.

[110] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *Proc. ICCV*, pages 1–8, 2007.

[111] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *Proc. SIGGRAPH*, pages 473–482, Jul 2002.

[112] J. Koza and R. Poli. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer, 2005.

[113] J. R. Koza. Genetic programming: a paradigm for genetically breeding populations of computer programs to solve problems. Technical report, Stanford, CA, USA, 1990.

[114] S. Krishnamachari and M. Abdel-Mottaleb. Image browsing using hierarchical clustering. *Computers and Communications, IEEE Symposium on*, pages 301–307, 1999.

[115] J.-E. Kyprianidis. Image and video abstraction by multi-scale anisotropic Kuwahara filtering. In *Proc. NPAR*, pages 55–64, 2011.

[116] J.-E. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg. State of the art: A taxonomy of artistic stylization techniques for images and video. *IEEE Trans. Vis. Comput. Graph.*, 2012.

[117] J.-E. Kyprianidis and J. Döllner. Image abstraction by structure adaptive filtering. In *Proc. EG UK TPCG*, pages 51–58, 2008.

[118] J.-E. Kyprianidis and J. Döllner. Image abstraction by structure adaptive filtering. In *Proc. EG UK Theory and Practice of Computer Graphics*, pages 51–58, 2008.

[119] J.-E. Kyprianidis and H. Kang. Image and video abstraction by coherence-enhancing filtering. *Computer Graphics Forum*, 30(2):593–602, 2011.

[120] J.-E. Kyprianidis, H. Kang, and J. Döllner. Image and video abstraction by anisotropic Kuwahara filtering. *Computer Graphics Forum*, 28(7):1955–1963, 2009.

[121] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, pages 282–289, 2001.

[122] D. A. Langan, J. W. Modestino, and J. Zhang. Cluster validation for unsupervised stochastic model-based image segmentation. *IEEE Trans. on Image Processing*, 7(2):180–195, 1998.

[123] S. Lankton and A. Tannenbaum. Localizing region-based active contours. *IEEE Trans. on Image Process.*, pages 2029–2039, 2008.

[124] H. Lee, S. Seo, S. Ryoo, and K. Yoon. Directional texture transfer. In *Proc. NPAR*, pages 43–50, 2010.

[125] Y. J. Lee and K. Grauman. Object-graphs for context-aware category discovery. pages 1–8, 2010.

[126] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. *Proc. ICCV*, pages 277–284, 2009.

[127] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *Proc. ICCV*, pages 277–284, 2009.

[128] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Comput. Vision*, 43:29–44, June 2001.

[129] C. Li, C. Kao, J. C. Gore, and Z. Ding. Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans. Image Process.*, 17(10):1940–1949, October 2008.

[130] C. Li, C. Xu, C. Gui, and M. D. Fox. Level set evolution without re-initialization: A new variational formulation. In *Proc. CVPR*, pages 430–436. IEEE, 2005.

[131] W. Li, M. Agrawala, B. Curless, and D. Salesin. Automated generation of interactive 3D exploded view diagrams. *ACM Trans. Graph.*, 27(3):101:1–7, 2008.

[132] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. In *Proc. SIGGRAPH*, pages 595–600, 2005.

[133] R. Lienhart. Abstracting home video automatically. In *Proc. MM*, 1999.

[134] L. Lin, K. Zeng, H. Lv, Y. Wang, Y. Xu, and S.-C. Zhu. Painterly animation using video semantics and feature correspondence. In *Proc. NPAR*, pages 73–80, 2010.

[135] T. Lin. Automatic video scene extraction by shot grouping. In *Proc. ICPR*, pages 39–42, 2000.

[136] P. Litwinowicz. Processing images and video for an impressionist effect. In *Proc. SIGGRAPH*, pages 407–414, 1997.

[137] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):978–994, 2011.

[138] C. Liu, J. Yuen, A. B. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *Proc. ECCV*, pages 28–42, 2008.

[139] H. Lombaert, Y. Sun, L. Grady, and C. Xu. A multilevel banded graph cuts method for fast image segmentation. In *Proc. ICCV*, pages 259–265, 2005.

[140] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, pages 1150–1157, Washington, DC, USA, 1999. IEEE.

[141] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, 2004.

[142] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li. A user attention model for video summarization. In *Proc. MM*, pages 533–542. ACM, 2002.

[143] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *Int. J. Comput. Vision*, 43(1):7–27, 2001.

[144] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. ICCV*, pages 416–423, 2001.

[145] D. R. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, 2004.

[146] E. McKone, N. Kanwisher, and C. Duchaine. Can generic expertise explain special processing for faces? *Trends in Cognitive Science*, 11:8–15, 2007.

[147] P. Mehrani and O. Veksler. Saliency segmentation based on learning and graph cut refinement. In *Proc. BMVC*, pages 110.1–12, 2010.

[148] T. Mei, X.-S. Hua, H.-Q. Zhou, and S. Li. Modeling and mining of users capture intention for home videos. *IEEE Trans. MM*, 9, 2007.

[149] B. J. Meier. Painterly rendering for animation. In *Proc. ACM SIGRGAPH*, pages 477–484, 1996.

[150] M. Meng, M. Zhao, and S.-C. Zhu. Artistic paper-cut of human portraits. In *Proc. MM*, pages 931–934, 2010.

[151] F. Moscheni, S. Bhattacharjee, and M. Kunt. Spatiotemporal segmentation based on region merging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20:897–915, 1998.

[152] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *Proc. CVPR*, pages 22–26. IEEE, 1985.

[153] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-video search for object appearances. In *Proceedings of the IFIP TC2/WG 2.6 Second Working Conference on Visual Database Systems II*, pages 113–127, 1992.

[154] R. Oami, A. B. Benitez, S.-F. Chang, and N. Dimitrova. Understanding and modeling user interests in consumer videos. In *Proc. ICME*, pages 1475–1478, 2004.

[155] M. Obaid, R. Mukundan, and M. Billinghurst. Rendering and animating expressive caricatures. In *Proc. ICCSIT*, pages 401–406, 2010.

[156] P. O'Donovan and A. Hertzmann. AniPaint: Interactive painterly animation from video. *IEEE Trans. Vis. Comput. Graph.*, 18(3):475–487, 2012.

[157] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.

[158] D. K. Panjwani and G. Healey. Markov random field models for unsupervised segmentation of textured color images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(10):939–954, 1995.

[159] N. Paragios and R. Deriche. A pde-based level-set approach for detection and tracking of moving objects. In *Proc. ICCV*, pages 1139–1145, 1998.

[160] N. Paragios and R. Deriche. Geodesic active regions: a new paradigm to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation*, pages 249–268, 2002.

[161] S. Paris. Edge-preserving smoothing and mean-shift segmentation of video streams. In *Proc. ECCV*, pages 460–473, 2008.

[162] S. Paris and F. Durand. A topological approach to hierarchical segmentation using mean shift. In *Proc. CVPR*, pages 1–8, 2007.

[163] I. Patras, E. A. Hendriks, and R. L. Lagendijk. Video segmentation by map labeling of watershed segments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1553–1567, 2001.

[164] P. Perez, M. Gangnet, and A. Blake. Poisson image editing. In *SIGGRAPH*, pages 313–318, 2003.

[165] R. Poli, W. B. Langdon, and N. F. McPhee. *A field guide to genetic programming.* Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008.

[166] T. Pouli and E. Reinhard. Progressive color transfer for images of arbitrary dynamic range. *Computers & Graphics*, 35(1):67–80, 2011.

[167] B. Price, B. Morse, and S. Cohen. Learning-based interactive video segmentation by evaluation of multiple propogated cues. In *Proc. ICCV*, 2009.

[168] B. L. Price and S. Cohen. Stereocut: Consistent interactive object selection in stereo image pairs. In *Proc. ICCV*, 2011.

[169] B. L. Price, B. S. Morse, and S. Cohen. Geodesic graph cut for interactive image segmentation. In *Proc. CVPR*, pages 3161–3168, 2010.

[170] C. Primo, A. Hernandez, and S. Escalera. Automatic user interaction correciton via multi-label graph cuts. In *Proc. ICCV Workshop on HCI in Computer Vision*, 2011.

[171] A. Protiere and G. Sapiro. Interactive image segmentation via adaptive weighted distances. *IEEE Trans. Image Process.*, 16, 2007.

[172] A. Rabinovich, S. Belongie, T. Lange, and J. M. Buhmann. Model order selection and cue combination for image segmentation. In *Proc. CVPR*, pages 1130–1137, 2006.

[173] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Colour transfer between images. *IEEE Comput. Graph. Appl.*, 21:34–41, 2001.

[174] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. ICCV*, volume 1, pages 10–17, 2003.

[175] M. Ristivojevic and J. Konrad. Space-time image sequence analysis: object tunnels and occlusion volumes. *IEEE Trans. Image Processing*, 15(2):364–376, 2006.

[176] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Does organisation by similarity assist image browsing? In *Proc. CHI*, pages 190–197. ACM, 2001.

[177] P. L. Rosin and Y.-K. Lai. Towards artistic minimal rendering. In *Proc. NPAR*, pages 119–127, 2010.

[178] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. In *Proc. SIGGRAPH.* ACM, 2004.

[179] C. Rother, V. Kolmogorov, Y. Boykov, and A. Blake. Interactive foreground extraction using graph cut. Technical Report MSR-TR-2011-46, Microsoft Research, UK, 2011.

[180] M. Rousson and N. Paragios. Shape priors for level set representations. In *Proc. ECCV*, pages 78–92. Springer, 2002.

[181] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proc. CVPR*, pages 1605–1614, 2006.

[182] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77:157–173, May 2008.

[183] A. Santella and D. DeCarlo. Abstracted painterly renderings using eye-tracking data. In *Proc. NPAR*, pages 75–82, 2002.

[184] G. Schaefer. A next generation browsing environment for large image repositories. *Multimedia Tools Appl.*, 47(1):105–120, 2010.

[185] A. Schodl, R. Skeliski, D. Salesin, and H. Essa. Video textures. In *Proc. SIGGRAPH*, pages 489–498, Jul 2000.

[186] L. Shafarenko, M. Petrou, and J. Kittler. *IEEE Trans. on Image Processing*, 6(11):1530–1544, 1997.

[187] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *Proc. ICCV*, pages 1154–1160, 1998.

[188] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[189] M. Shiraishi and Y. Yamaguchi. An algorithm for automatic painterly rendering based on local source image approximation. In *Proc. NPAR*, pages 53–58, 2000.

[190] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int. J. Comput. Vision*, 81:2–23, January 2009.

[191] M. Shugrina, M. Betke, and J. Collomosse. Empathic painting: Interactive stylization using observed emotional state. In *Proc. NPAR*, pages 87–96, 2006.

[192] B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW 2008*, pages 327–336, 2008.

[193] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *Proc. CVPR*, pages 119–127, 2008.

[194] D. Singaraju, L. Grady, and R. Vidal. P-brush: Continuous valued mrfs with normed pairwise distributions for image segmentation. In *Proc. CVPR*, pages 1303–1310, 2009.

[195] P. Sinha. Face recognition by humans: Nineteen results all computer vision researchers should know about. In *Proceedings of the IEEE*, pages 1948–1962, 2006.

[196] A. K. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *Proc. ICCV*, pages 1–8, 2007.

[197] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, volume 2, pages 1470–1477, 2003.

[198] Y. Song, P. M. Hall, P. Rosin, and J. Collomosse. Arty shapes. In *Proc. CAe*, pages 65–72, 2008.

[199] A. N. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *Int. J. Comput. Vision*, 82(3):325–357, 2009.

[200] S. Strassmann. Hairy brushes. In *Proc. SIGGRAPH*, pages 225–232, 1986.

[201] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Proc. ICCV*, pages 900–907, 2003.

[202] D. A. Tolliver and G. L. Miller. Graph partitioning by spectral rounding: Applications in image segmentation and clustering. In *Proc. CVPR*, pages 1053–1060, 2006.

[203] M. Tominaga, S. Fukuoka, K. Murakami, and H. Koshimizu. Facial caricaturing with motion caricaturing in PICASSO system. In *Proc. ICAIM*, page 30, 1997.

[204] M. Tominaga, J.-I. Hayashi, K Murakami, and H. Koshimizu. Facial caricaturing system PICASSO with emotional motion deformation. In *Proc. KES*, pages 205–214, 1998.

[205] S. M. F. Treavett and M. Chen. Statistical techniques for the automated synthesis of non-photorealistic images. In *Proc. EGUK*, pages 201–210, 1997.

[206] D. Tsai, M. Flagg, and J. Rehg. Motion coherent tracking with multi-label mrf optimization. In *Proc. BMVC*, pages 56.1–11, 2010.

[207] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.

[208] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *Int. J. Comput. Vision*, 62:61–81, 2005.

[209] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, May 2004.

[210] J. Wang, M. Agrawala, and M. F. Cohen. Soft scissors: an interactive tool for realtime high quality matting. In *Proc. SIGGRAPH*, pages 585–594. ACM, 2007.

[211] J. Wang, P. Bhat, A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. In *Proc. SIGGRAPH*, pages 585–594, 2005.

[212] J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. In *Proc. ECCV*, pages 238–249. Springer, 2004.

[213] J. Wang, Y. Xu, H. Shum, and M. Cohen. Video tooning. In *Proc. SIGGRAPH*, volume 23, pages 574–583, 2004.

[214] J.-P. Wang. Stochastic relaxation on partitions with connected components and its application to image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(6):619–636, 1998.

[215] L. Wang, B. Zeng, S. Lin, G. Xu, and H.-Y. Shum. Automatic extraction of semantic colors in sports video. In *Proc. ICASSP*, pages 617–620, 2004.

[216] T. Wang and J. Collomosse. Progressive motion diffusion of labeling priors for coherent video segmentation. *IEEE Transactions on Multimedia*, 14(2):389–400, April 2012.

[217] T. Wang, J. Collomosse, A. Hunter, and D. Greig. Digital raphael: Learnable stroke models for example-based portrait painting. *submitted to IEEE Trans. Vis. Comput. Graph.*, 2012.

[218] T. Wang, J. P. Collomosse, R. Hu, D. Slatter, D. Greig, and P. Cheatle. Stylized ambient displays of digital media collections. *Computers & Graphics*, 35(1):54–66, 2011.

[219] T. Wang, J. P. Collomosse, D. Slatter, P. Cheatle, and D. Greig. Video stylization for digital ambient displays of home movies. In *Proc. NPAR*, pages 137–146, 2010.

[220] T. Wang, J.-Y. Guillemaut, and J. P. Collomosse. Multi-label propagation for coherent video segmentation and artistic stylization. In *Proc. ICIP*, pages 3005–3008, 2010.

[221] T. Wang, B. Han, and J. Collomosse. Touchcut: Fast image and video segmentation using single-touch interaction. *submitted to Computer Vision and Image Understanding (CVIU)*, 2012.

[222] T. Wang, B. Han, and J. Collomosse. Touchcut: Single-touch object segmentation driven by level set methods. In *Proc. ICASSP*, 2012.

[223] T. Wang, A. Mansfield, R. Hu, and J. Collomosse. An evolutionary approach to automatic video editing. In *Proc. CVMP*, Nov 2009.

[224] X. Wang and X. Tang. Face photo-sketch synthesis and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(11):1955–1967, 2009.

[225] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. pages 1800–1807, 2005.

[226] H. Winnemoller, S.C. Olsen, and B. Gooch. Real-time video abstraction. In *Proc. SIGGRAPH*, pages 1221–1226, 2006.

[227] W. You, S. Feis, and R. Lea. Studying vision-based multiple-user interaction with in-home large displays. In *Proc.* 3rd *ACM workshop on Human-Centred Computing (HCC)*, pages 19–26, 2008.

[228] K. Zeng, M. Zhao, C. Xiong, and S.-C. Zhu. From image parsing to painterly rendering. *ACM Trans. Graph.*, 29(1):2:1–11, 2009.

[229] W. Zhang, X. Wang, and X. Tang. Lighting and pose robust face sketch synthesis. In *Proc. ECCV*, pages 420–433, 2010.

[230] H. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, pages 179–195, 1996.

[231] M. Zhao and S.-C. Zhu. Sisley the abstract painter. In *Proc. NPAR*, pages 99–107, 2010.

[232] M. Zhao and S.-C. Zhu. Portrait painting using active templates. In *Proc. NPAR*, pages 117–124, 2011.

[233] S. Zhu and A. Yuille. Region competition: unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 884–900, 1996.