UNIVERSITY OF
LEADING
THE WAY
WESTMINSTER

WestminsterResearch
http://www.westminster.ac.uk/research/westminsterresearch

Visual region understanding: unsupervised extraction and abstraction

Gaurav Gupta

School of Electronics and Computer Science

This is an electronic version of a PhD thesis awarded by the University of Westminster. © The Author, 2012.

This is an exact reproduction of the paper copy held by the University of Westminster library.

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners.
Users are permitted to download and/or print one copy for non-commercial private study or research. Further distribution and any use of material from within this archive for profit-making enterprises or for commercial gain is strictly forbidden.

Whilst further distribution of specific materials from within this archive is forbidden, you may freely distribute the URL of WestminsterResearch: (http://westminsterresearch.wmin.ac.uk/).

In case of abuse or copyright appearing without permission e-mail repository@westminster.ac.uk

Visual Region Understanding: Unsupervised Extraction and Abstraction

Gaurav Gupta

---

1. Supervisor: Dr. Alexandra Psarrou
2. Supervisor: Dr. Alexander Bolotov

A Thesis submitted in partial fulfilment of the
requirements of the University of Westminster
for the degree of Doctor of Philosophy

School of Electronics and Computer Science

August 2012

"I don't pretend we have all the answers. But the questions are certainly worth thinking about."

Arthur C. Clarke (British science fiction author, 1917-2008)

Affirmation

Herewith I declare that the work submitted is my own. Appropriate credit has been given to thoughts that were taken directly or indirectly from other sources.

Gaurav Gupta

This thesis was created at the School of Electronics and Computer Science (ECS) at University of Westminster between 2007 and 2012. During this time this work was supported internally by the Computer Vision and Imaging Research Group (CVIR).

Acknowledgements

This thesis was made possible by an Overseas Research Student (ORS) scholarship and maintenance funds awarded to me by the University of Westminster.

I am very grateful to my family, colleagues and friends for creating a wonderful environment that allowed me to enthusiastically pursue my research. My studies would not have been both fun and meaningful at the same time had it not been for the amazing levels of freedom, flexibility and drive afforded me by my primary research supervisor Dr. Alexandra Psarrou, whose gentle guidance took me in wonderful directions, by my employer Farnaz Fazaipour, whose support allowed fascinating levels of multitasking, and by my second supervisor Dr. Alexander Bolotov, whose calm logic was inspiring. A note of appreciation to Dr. Anastassia Angelopoulou, whose initial guidance gradually transformed into an important association and friendship that was crucial to the development of my research ideas. Deeply felt gratitude specially to my mother, Jhuma Gupta, without whose unwavering support this body of research would not have materialised.

The presence of close colleagues and friends also facilitated my progress. Numerous discussions and enjoyable times with friends and collaborators Sardar Zohaib Khan and Jae Young Park helped maintain a level of enthusiasm that was immensely helpful for my focus and drive. Miscellaneous thanks to Mehak Puri, Karolina Juszczak, and Eleftheria Mpouthalaki. Additional thanks to Dr. Sophie Triantaphillidou, Anastasia Tsifouti, Roger Campos, Aaron Licata, Tarek Shaaban and Dora Hadjinaki.

Dedicated to my mother.

Abstract

The ability to gain a conceptual understanding of the world in uncontrolled environments is the ultimate goal of vision-based computer systems. Technological societies today are heavily reliant on surveillance and security infrastructure, robotics, medical image analysis, visual data categorisation and search, and smart device user interaction, to name a few. Out of all the complex problems tackled by computer vision today in context of these technologies, that which lies closest to the original goals of the field is the subarea of unsupervised scene analysis or scene modelling. However, its common use of low level features does not provide a good balance between generality and discriminative ability, both a result and a symptom of the sensory and semantic gaps existing between low level computer representations and high level human descriptions.

In this research we explore a general framework that addresses the fundamental problem of universal unsupervised extraction of semantically meaningful visual regions and their behaviours. For this purpose we address issues related to (i) spatial and spatiotemporal segmentation for region extraction, (ii) region shape modelling, and (iii) the online categorisation of visual object classes and the spatiotemporal analysis of their behaviours. Under this framework we propose (a) a unified region merging method and spatiotemporal region reduction, (b) shape representation by the optimisation and novel simplication of contour-based growing neural gases, and (c) a foundation for the analysis of visual object motion properties using a shape and appearance based nearest-centroid classification algorithm and trajectory plots for the obtained region classes.

Specifically, we formulate a region merging spatial segmentation mechanism that combines and adapts features shown previously to be individually useful, namely parallel region growing, the best merge criterion, a time adaptive threshold, and region reduction techniques. For spatiotemporal region refinement we consider both scalar intensity differences and vector optical flow. To model the shapes of the visual regions thus obtained, we adapt the growing neural gas for rapid region contour representation and propose a contour simplication technique. A fast unsupervised nearest-centroid online learning technique next groups observed region instances into classes, for which we are then able to analyse spatial presence and spatiotemporal trajectories. The analysis results show semantic correlations to real world object behaviour. Performance evaluation of all steps across standard metrics and datasets validate their performance.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter discusses the motivation behind this body of research in unsupervised scene analysis and briefly discusses relevant issues in the areas of spatial and spatiotemporal segmentation, shape representation, similarity matching and categorisation. The main contributions are identified and an outline of the organisation of this thesis given.

1.1 Motivation

The question of how a machine can begin to learn a conceptual representation of its environment and reason with it is a very old and difficult one with no definite solution, hence it is termed ill-posed. It has only been relatively recently that rapidly increasing computational power and newer theories of unsupervised vision, learning, and reasoning have enabled new progress in this line of research. Yet there still does not exist a general theory of sensor based autonomous conceptual learning. Given the difficulty of the problem [63, 74], this work addresses a reduced version of it: semantic visual understanding.

Vision is a convenient source of information about the real world [160, 106], but it presents large streams of raw sensory data that are complex and hard to analyse computationally. Describing and categorising allow one to extract plausible patterns from an input stream, resulting in fewer components and reducing the analysis space. Visual semantic analysis, or visual intelligence [32], requires several processing steps. A feature extraction mechanism is required as the first subsymbolic level of abstraction. Features may be in the form of points or regions, spatial regions arising out of image segmentation [94, 140, 43, 77] and/or motion regions from motion segmentation [209, 199, 134]. Second, a matching/recognition scheme [24, 146, 115] is required in order to distinguish between new and previously encountered features. Third, various properties of each feature must be analysed and represented appropriately such that the abstract conceptualisation of different classes is possible [121]. Finally, there should be a way to evaluate system performance through its application [144] to one or more perceptual tasks and the measurement of its success at those tasks. Also, in order to be fully unsupervised, no training sets should be required to initialise shape categories.

A framework for the unsupervised semantic understanding of images and video is presented in this work. An unsupervised colour segmentation mechanism [90] is applied to spatially group pixels into regions. The spatial segments are further grouped into spatiotemporal regions according to motion information. Motion segments are comprised of multiple spatial segments linked together by their *common fate* of motion [188]. Similar spatiotemporal shapes are then categorised, with respect to their contour representation descriptions, into region classes that form the basis for visual behaviour analysis.

The segmentation step prepares us to carry out two levels of region behaviour modelling, one based on localised appearance tracking of scalar motion segments

across frames while the other, relying on vector motion information and region contour representation, produces shape-based spatiotemporal region class trajectories. The first depends on localised region tracking, involving colour, size and position features to describe each region. Between consecutive frames, these features are compared and matched within a bipartite graph consisting of regions discovered in each frame. We present this type of localised region recognition to demonstrate the usefulness of the results that we obtain even with a simple approach. The second, which is more representative of the main goals of this research, groups region instances into classes, and trajectories consisting of all instances within a class are established and analysed. This forms a primitive foundation for visual intelligence [32]. Applications of this framework include query by example [37, 152, 151] in extensions of content based image retrieval to video, automated scene understanding [19, 32], autonomous robot operation [49, 136] and visual learning [172, 79].

We begin with an overview of topics related to the design of such a framework.

1.2 Image segmentation

Splitting an image into a set of component parts is a first step without which much of higher level visual processing cannot be done. This splitting is refered to as image segmentation, for which which there exist many different computational methods. Image segmentation works on a stream or a grid of raw information and translates these to a smaller abstracted set of component regions. In general, the fewer the regions the higher the level of abstraction. The goal is to achieve a high level of abstraction by obtaining as few regions of homogeneity as possible without sacrificing what may be useful detail.

Stating the goal in such fashion raises the important question of how the optimal level of intraregion homogeneity is to be decided. A single pixel itself is perfectly homogeneous as is a group of pixels of exactly the same colour. Since almost all real world objects appear to have different shades [117, 202] at different locations on their surfaces this view of homogeneity would produce a very large number of regions, most of which would be too small to be useful. On the other hand, simply aiming to either minimise the number of regions or maximise average region size can give us the entire image as one big segment with large internal colour variations. How to strike a balance between homogeneity and the number of regions, or how to establish a stopping criterion [2], is a challenging question with no certain answer as yet. There are information theoretic solutions to this problem, but qualitative and quantitative evaluation of their segmentation results show that we do not yet know the solution that most closely matches human visual perception.

A related problem is what we define homogeneity in terms of. Since homogeneity is expressed as similarity between two sets of descriptive features[1], selecting an appropriate set of features and an appropriate function of distance between the feature sets should give us a suitable measure of homogenity. However, there are many features that can be constructed from raw pixel data, such as colour in different channels [139], texture, and measures of energy. While the most obvious solution is to use intensity or colour information and to measure distances between them in Euclidean space, there is no certain answer as to what is the most effective set of features nor what the optimal distance measure between feature

[1]While numerically different, it is convenient to think of feature similarity and feature distance as effectively equivalent, that is, region similarity can be interpreted as lack of region dissimilarity and vice versa.

sets is. Segmentation results are tied to colour representations [114] and geometric distances [86]. Studies have shown that different colour representations show different levels of perceptual uniformity [205]. Perceptual uniformity means that a certain change in a colour value should produce an approximately proportional change in visual importance. This poses a problem when we visually assess the result of a segmentation which has used a perceptually non-uniform colour representation and find that it does not match our expectations. Furthermore, various statistical summary descriptors for regions can be fed into distance measure functions to get different types of similarity values. The mean is the most common, but other statistical measures can also be used, with varying results.

Once a suitable distance measure and colour representation are chosen, the next question is how to select a threshold [162, 213] that decides whether a given distance is to indicate a merge or not. The threshold can be indepedent of the input, such as fixed or time-varying, or can be decided based on information theoretic measures [33], such as the ratio of intra-region homogeneity and inter-region heterogeneity.

1.3 Region growing

These and other questions influence the final result both in terms of quality and computational complexity and must be explored with the aid of a specific image segmentation tool. Out of the various approaches to segmentation, region growing [217] is the most intuitive procedure for grouping individual elements to form bigger regions and is one that best addresses direct proximity relations within immediate topological neighbourhoods. Due to its structure, which permits detailed intra-process control, region merging allows us to test hypotheses regarding

merging order, the order in and rate at which primitive regions are examined and merged, and varying thresholds.

Research into the human visual system has shown that our eyes jump briefly from point to point when exploring an object [96], called saccades. Regions can be grown by focusing on a specific pixel and growing a region around it as far as possible before moving on to another pixel, or smaller regions distributed around the image can be grown at a common rate. Alternatively, the rate of region growth can be equal for different image locations during a part of the region growing process and unequal for another. Different strategies produce different quality of results.

Further variations in the region merging framework are of interest. One is a changing region model [16]. When regions are pixel sized they are represented by the feature vector of the pixel itself. For small primitive regions the region model may be taken as a summary descriptor of the feature vectors of all component pixels. It is possible however, that when regions grow larger just a summary descriptor of component pixels may not be a good representation. Large regions can contain significant intra-region variation and most of the factors affecting further merge decisions reside at the boundaries of the region instead of being distributed over the entire region. Not only is there a question of whether primitive and more advanced regions should be represented using different region models, but we must also consider whether feature distances for the two should be calculated differently.

A segmentation typically stops when no more region changes are possible given a particular dissimilarity or distance measure and some threshold. Human perception is however able to identify varying levels of detail given the context and intention. Often when it may be possible to consider two regions as one, one may still identify them as distinct segments. Similarly, we may consider a critical

function for a region assuming a given intention and context [18, 119] which can override other criteria. We may wish, for example, to preferentially preserve an area of some type of detail even though regions within it may satisfy the default merge criteria. This raises the question of whether certain visual details are salient enough to be kept intact even if they demonstrate homogeneity with other image features.

Region growing considers one primitive region and its immediate neighbourhood at a time. A merge decision between any two neighbours may therefore not be optimal, if there exists a different better merge decision when looking at the whole picture. Also, assuming a globally optimal merge over the entire physical region space, there is the question of temporal optimality. Is it possible that an inferior merge decision now may eventually lead to an overall better segmentation map? The characteristic of looking at very local regions in space and time is termed a greedy search. A non-greedy search is one that truly discovers the best current choice given all spatial and temporal outcomes. A fully non-greedy search would however consume tremendous computational resources in a brute force search fashion. A convenient and effective strategy may be to design a semi-greedy search, one that assesses locally expanded spatial and temporal domains in order to arrive at a decision that appears optimal for a region as well as its neighbours.

1.4 Region restriction and region reduction

The growth of regions can be restricted using other information derived from the original input stream, for instance edge locations. Edge detection [178, 113] indicates pixel positions where boundaries are likely to exist, and this information can

be used to prevent or discourage merges from taking place at these positions, depending on edge strength. However, edge detectors operate by considering first and/or second order derivatives of images, which are decided by local changes between pixel values. Since a region growing procedure looks at pixel differences to start with, it may be that edges can be modeled intrinsically to the process. If this is the case, then hybrid techniques that use both edge and region information could be unified into a single framework. This could then further be extended to include other features such as corners and even salient points.

While region restriction enforces greater segment boundary accuracy by discouraging merges over strong boundary features, region reduction works in the opposite fashion, attempting to find some other common ground between regions not explained by the region model and the homogeneity criteria. For instance, regions completely enclosing another can be permitted to absorb the inner region if certain relaxed homogeneity conditions are met. Alternatively, occluded objects may appear as two distinct regions in which some criteria can be implemented to merge the two with the occlusion assumption.

Region reduction is a very important step. An accurate segmentation map can contain thousands of regions and if, for example, one is to match shapes between all regions in two image frames then the number of possible combinations is very large. If we are able to apply region reduction to bring down the number of regions to a few hundred then this task becomes much more tractable. It is also important to note that producing fewer segments simply by relaxing either the merge criteria or merge theshold is likely to indiscriminately violate local boundaries. While region reduction also violates certain local boundaries, it does so by looking at a larger context and is therefore more likely to be perceptually acceptable.

1.5 Temporal information

The colour values for pixels are the most basic pieces of information about an image, from which all spatial features are derived. Temporal data is an additional source of derived spatial information[1]. Intensity values considered for two images in a sequence, where one frame is some spatial transformation of the previous, can be combined to obtain motion features. Motion features reflect spatiotemporal coherence according to Gestalt principles of common fate [188].

Motion features can be obtained in two forms. Image differencing [157, 58] produces a map of areas that have changed between frames. This kind of feature set is incomplete and scalar in that one can tell which areas exhibit motion but not in which direction. Additionally one is not sure if the areas identified represent the complete set of motion pixels since non-textured object interiors do not respond to this technique.

To obtain more complete motion information there exist methods to compute optical flow [21], a set of vectors that show the estimated direction and magnitude of motion at a set of points. The lack of motion response [193] in non-textured surfaces is of concern here as well. Some techniques to compute optical flow produce sparse representations, that is not every pixel in the frame is guaranteed to have a flow vector computed for it, typically only motion pixels of large displacement [29] being included. Other methods however provide a dense optical flow map, where dense means every pixel is assigned a flow vector. Dense optical flow methods must 'guess' harder if they are to come up with a flow vector for non-textured

[1]This research restricts itself to work with a single camera input, however other work in the field using stereo imaging introduces yet another source of information: disparity between left and right frames, from which another type of information, depth, can be obtained and integrated with a region growing model.

surfaces, and therefore one must treat dense flow vectors as at least partially unreliable, but they do provide a complete flow map where one is required.

Using a dense optical flow map as additional input to our region growing framework allows us to estimate region similarity over the supplementary motion features. This is useful since in static images a computational algorithm cannot know if two or more segments of very different appearance actually belong together. By using the Gestalt principle of common fate, that is by saying that if two regions move similarly then they may actually be parts of the same object, we are able to further reduce the number of regions. This is a convenient approach to grouping together complex objects that appear cohesive only when they move relative to the viewing frame.

1.6 Shape representation and similarity

Using spatial and spatiotemporal segmentations one can extract meaningful regions from image and video visual inputs. In order to make sense of these regions one must represent them in terms of a set of discriminative properties. Much effort has been put into discovering invariants [135] such as scale invariant, translation invariant and rotation invariant properties, together termed shape invariants. While each of these invariances is individually important, we ideally want a set of properties that reflect each of these invariances such that it becomes possible to differentiate between a variety of objects.

It has been found to be difficult to reliably describe and match regions using such properties based on real visual data. There are two sources of this difficulty: region representation and region description. One is forming accurate shape representations in the presence of noise and the second is identifying a good set of

discriminative properties that work for artificially generated exact shapes as well as for real-world noisy regions.

The previously described steps of image and video segmentation are imperfect processes prone to subjective interpretation. An innacurate set of region maps will propagate errors to shape representation schemes. Thus one attempts to minimise region segmentation inaccuracies and apply noise/outlier-tolerant shape representation methods. Shapes may be represented through the analysis of global or structural information, or of contour or region information [210]. A global representation is generally more tolerant to outliers than a structural one, since outliers can be averaged or smoothed out over the entirety of the shape, but are less discriminative because local details are ignored.

When we think about shapes we think about global properties such as size, colour and symmetry, and we think about silhouettes or contours. Contours not only describe important variations in region appearances, but they also provide instructions for drawing the shapes of regions. Contour representations such as chain codes and self-organising maps are information preserving and allow one to reproduce approximations to object shapes in the absense of the original raw data. Reproducability, an indicator of the preservation of useful shape information, is much more difficult and computationally complex when performing shape representation by methods such as polygon decomposition, where regions are represented as a collection of simple geometric shapes, because it is more challenging to accurately and compactly describe a shape using polygons than it is to follow its contour and identify features along it.

Given that our source of shape information are image segments, we need to efficiently obtain contour plots from segmented regions. There are well established methods of obtaining such contours. Convex hulls [169], polygonal approximation

[5], chain codes [76], skeletal graphs [25] and self-organising maps [166, 171] are the most commonly applied techniques. Convex hulls calculate convex deficiencies or concavities, and representation accuracy is dependant on the level of, almost fractal-like, recursive analysis of these concavities to find greater detail. Polygonal approximation attempts to represent a curve in terms of a set of connected line segments. Chain codes use a fixed set of directions when following and representing a contour. Chain codes can encode a contour well when sharp, single pixel boundary map has been obtained, but is heavily dependant on boundary thinning to achieve this. Skeletal graphs work by thinning a solid shape down to a skeletal representation using an n-neighbour voxel distance heuristic. Convex hulls, chain codes and skeletal graphs however are all computationally demanding to generate and rely on intensive boundary or volumetric preprocessing. Additionally, with respect to shape contours, chain codes are directionally restricted typically to 8 directions, convex hulls are directionally more vague, and skeletal graphs significantly lose contour reproducability. A method that can work with incomplete boundary information and which is more balanced between degrees of directional freedom and contour representation accuracy is useful. A self-organising map (SOM) [108] is capable of this. As in the name, a SOM explores and discovers an input space and adjusts itself to match, within approximation bounds. The growing neural gas (GNG) [81] is specific self-organising map well adapted to this task.

The GNG can be used to quickly and efficiently encode a region boundary in terms of nodes distributed along the boundary and edges connecting these nodes. It is useful for shape representation since it is fast for small regions and obtains an abstraction of a region silhouette which contains global as well as local curvature information. The GNG has already been applied to polygonal approximation and medial axis extraction [214]. A GNG contour representation can also be interpreted

as a set of instructions to visually reproduce the silhouette. The GNG however has two shortcomings. First, its convergence speed can drop significantly for large regions, and second, in many cases there can be more than two edges for some of the nodes. We will investigate methods that a) greatly speed up the network performance with only slightly reduced topological correctness, and b) guarantee a maximum of two edges for each node in any network.

At this point we are able to segment regions and obtain contour maps for them. The next task is assessing the level of similarity between two or more such regions. Like shape representation techniques, shape similarity measures can also be global or structural, region or contour based. Apart from simple region properties, similarity measures are generally tied to the specific shape representation method being used. The usual approach with a contour-based representation is curvature analysis along the set of contour points. Curvature analysis however is made complicated by the fact that complete contours curve in on themselves in a closed loop of 360° [64], and therefore many summary statistics will smooth out local variations over this complete turn. Thus, while a large variety of shape descriptors can be found in the literature [130, 123, 192, 191, 210, 156, 104], one must choose carefully from them.

1.7 Dynamic grouping and description

The next level of abstraction involves categorisation. After obtaining region appearance descriptions and choosing discriminative similarity measures, we can group observed regions into different classes. These classes can then be used for similarity searching within video sequences, or content matching paradigms such as content based image retrieval (CBIR) [167, 190, 56], query by visual example

(QBVE) [98], query by semantic example (QBSE) [152] and query by contextual example (QBCE) [151]. An important consideration here is the classification speed. A single image can yield hundreds of regions, going up to several thousands of regions for even a short video. For a live system it is important to have a region categorisation technique that is based on online classification, using every observation to both classify an instance as well as to refine its class descriptor. However, many of the existing content based querying systems are based on offline or batch learning.

Online classification methods can be divided into instanced based learning [4, 57, 54] and eager learning systems, which comprise of all other learning systems. Instance based learning is refered to as lazy learning because it does not form abstractions during a series of observations but simply stores instances and waits until a new instance has to be classified at which point a local neighbourhood similarity search and majority class label assignment is done. In contrast, eager learning systems diverts effort to forming abstractions of observed instances and comparing new instances to the set of abstracted classes. Since instance based learning consumes more and more computational resources when searching through a growing set of observations, eager learning is more suitable for online systems that must stay operational indefinitely. While eager learning spends more resources on abstraction, it saves on both classification speed and on storage space. Both speed and space are of critical concern when dealing with learning regions and shapes from large video sequences, the total set of instances from which may rapidly become very large due to the number of regions identified from each frame and the overall number of frames. However, eager learning methods must commit to a single global approximation at the time of observation, a shortcoming that instance based learning does not share. While there have been attempts to combine instance

based and eager learning methods into hybrid systems [97], these systems are also characterised extended search times and large space requirements.

A popular eager learning approach is the nearest centroid method [142, 92, 120] which is fast and simple. A variant of this is the nearest shrunken centroid method [176, 120] which shrinks class centroids towards the overall centroid for all classes. Shifting centroids towards the overall mean reduces the sensitivity of the method to outliers.

1.8 Contributions

The contributions made in this body of research are as follows:

1. An efficient segmentation method via a novel region merging algorithm that combines and adapts into a single framework techniques that are found separately in previous literature. The adaptations include the more expensive best merge and a time-expanding theshold that allows cascaded region growth and simultaneously provides for the correction of errors inherent in a single pass scan.

2. A new performance summary indicator, relative performance RP, that is able to combine arbitrary sets of evaluation metrics into a single number, allowing the instant comparison of performances for different labeling methods.

3. Region reduction through the consideration of region flow, the overall optical flow of each spatial region, by the use of dense optical flow information derived from the sparse Lucas-Kanade flow estimation method. The components of the reduced set of spatiotemporal regions are linked by the Gestalt principle of common fate.

4. The application of the growing neural gas modelling technique to shape contour representation and the optimisation of the speed of its convergence as well as network simplification by the elimination of multiple node linkages.

5. The identification of discriminative sets of appearance and shape based region descriptors.

6. A fast centroid-based online classification scheme which allows the generalisation and categorisation of observed region instances into classes.

7. Three types of region class trajectory representations, spatial presence, horizontal variation and vertical variation, which together model the behaviour of instances in each region class as physical trajectories in the spatial and temporal domains and are shown to be semantically correlated to simple concepts of real-world object behaviour.

1.9 Thesis Outline

The remainder of this thesis is structured as follows. Chapter 2 discusses the relevant literature in each of the concerned areas, namely image segmentation and region growing, dense motion estimation, shape representation and similarity matching, and unsupervised scene analysis. Chapter 3 explores the various steps and components of region merging as a colour image segmentation technique and adapts them into a novel region merging framework. Region reduction methods, including the use of dense optical flow to obtain regions of greater perceptual significance, are presented. Chapter 4 shows the use of the growing neural gas for a contour representation that facilitates curvature analysis. The growing neural gas algorithm is simplified for greater speed as well as to guarantee a single linkage

contour representation. Features based on this network contour model are used to create information preserving region descriptions. Chapter 5 uses the region models in a fast online classification system that allows the abstraction of visual object categories and use of these for region class trajectory analyses. Finally, chapter 6 closes with a summary of the presented work and a discussion of further continuations to this research.

Chapter 2

Region extraction and analysis

Research in the cognitive sciences has produced terms such as dynamic grouping, neuronal synchrony, Gestalt principles, cascaded feature hierarchies, feature recognition, dynamic description, input mosaic, intermediate features and higher features [200]. These terms describe many parallels between our understanding of human perception and our efforts to duplicate the same in computers. This section will review relevant literature for the areas that concern this thesis: image segmentation and region merging, motion segmentation, shape representation and matching, and scene analysis.

2.1 Preliminary considerations

Machine vision techniques rely on a certain quality of images coming in from sensors to produce better processing results. In uncontrolled environments, faulty exposure and noise are two critical limiting factors when acquiring images.

The limited dynamic range of most cameras and colour representation schemes contribute to the effect known as overexposure or signal clipping, caused by an in-

coming signal exceeding the maximum measurable or representable value, and this effect is seen as patches of pure uniform white. While there exist software methods to compensate for this effect [161, 189, 163], there is no real way to correct clipping once it has occurred, and it is to the best interests of vision algorithms to rely on higher fidelity hardware. Conversely, the reverse effect is called under-exposure or crushed shadows. Either insufficient illumination, poor hardware response to existing illumination, or a limitation of a colour representation scheme, causes dark areas in an image to be zeroed out and therefore appear completely black. Again there is no way to accurately correct this phenomenon once it has occurred, and it is simply advisable to utilise the best available hardware possible for vision related computational tasks.

Camera noise [26] is another factor that can contribute to difficulties in image processing. Some noise or graininess is present in all devices that handle digital signals. When intense enough to be noticeable, it appears as random speckles on an image. This randomness can introduce anomalies when detecting motion via image differencing, since the noise may appear to be tiny movements. While various processing techniques [118, 129] are able to reduce noise, it is impossible to completely fix a noisy image. Various smoothing filters such as the gaussian filter are capable of reducing noise but have an adverse affect on edges. The median filter is much better at preserving edges and is particularly effective for salt and pepper noise, or impulsive noise.

Then there are other hardware related limitations, compared to the human visual system, such as camera jitter, sub-panoramic views, discrete instead of continuous signals, and resolution. There exists work attempting to compensate for each of these via software, but it is important to see developments in hardware technologies in these areas as well.

2.2 Image segmentation

Image segmentation is commonly defined as the identification of homogeneous regions within an image. The segmentation is then guided by the interpretation of homogeneity, usually involving colour or spatial distribution or both. Popular approaches, reviewed in [94, 140, 43, 77, 59], include region-based methods, edge-based methods, hybrid techniques incorporating both regions and edges, histogram-based methods, and graph-based methods.

Region-based methods [3, 38, 196, 46, 83] group pixels into segments based on some pixel similarity measure and threshold values to indicate whether the similarity test is passed. Edge-based methods [125, 103, 84], on the other hand, find region boundaries by applying edge detection mechanisms, and are limited due to the high number of edges found and by the need to have an effective mechanism to close edges and form contained regions. Histogram-based methods [45, 42] analyse peaks in dominant colours in order to establish cluster centers to which pixels are assigned. For true colour images, the histogram presents a huge number of colour combinations. Although images typically use a much smaller set of colours and methods such as non-parametric density estimation [111] can help reduce the methods complexity, histogram methods do not prioritise topological pixel relations. Hybrid methods [48, 66, 87] use both regions and edges but require complex mechanisms to draw correspondences between the two. Graph methods [174, 182, 72] represent pixels as nodes on a graph and pixel groupings as links between nodes. Graph-based methods are usually computationally complex due to the huge set of potential pixel relations.

Some of the leading approaches to image segmentation are graph or tree based [164, 72, 60, 95, 148], probabilistic [6, 39, 133], statistical [50, 141, 23, 47], and other

approaches such as using chain codes [150] and data compression [206].

Segmentation methods strive to achieve a balance between the resolution of the results and the generic applicability of the method. Different segmentation techniques, similar to different human operators [126], can be expected to segment a particular image in different ways. At one end of this spectrum lie methods that extract the primary salient regions of images. For instance, a salient region segmentation algorithm [111] finds broad regions that are most likely to capture human attention. The limitation of this is that the details contained in these broad regions are lost. Conversely, methods such as [174, 83] produce greater detail but the perceptual significance of each of their segments grows less obvious since objects that may be considered whole segments by humans can be split into multiple regions.

Previous work on the resolution versus accuracy issue has involved changing the merge criterion or distance threshold on the fly [159, 175, 40, 38] or applying a regularisation factor [201] to eliminate tiny segments which leads to a bias towards uniformly sized segments. We use a very computationally simple scheme for adapting the distance threshold as the segmentation progresses, very similar to the dynamic merge relaxation in [175], to allow the regions more time to move towards their cluster centers in feature space before the distance threshold is raised.

A general survey of image segmentation methods is available in [44] and a survey on unsupervised segmentation methods in [211].

2.3 Region merging

While the state of the art, with respect to both quality and speed of segmentation, mainly revolves around the mean-shift [50], graph based [164, 72] and similar techniques, one of the most commonly applied family of techniques is region merging,

where pixels are step-wise grouped into larger and larger segments.

Region merging imposes direct topological constraints during the process of building a segment map. Early work in region merging involved classical greedy merging using local information and a simple L-shaped scan [27], the scan order, the order in which pixels or regions are considered for merging, subsequently being recognized as an important factor [67, 75], there being an "inherent dependence on (...) the order in which pixels and regions are examined" [44]. In contrast to path-based labelling, seeded region growing [3, 67] segments images by establishing seed points at certain locations and then growing regions around these. Further work included a statistically-based reinterpretation [138], enlarging the search space of the classic greedy algorithm [34], and a reanalysis of the entire region merging framework [33].

The calculation of all relevant merge costs at each iteration is an expensive operation [53], either in terms of computer memory if a list of merge costs are maintained or in terms of processing speed if merge costs are recalculated at each step. A suitable neighbourhood scanning procedure produces gains in resource utilisation. A simple blob colouring template [27], which goes left to right and top to bottom through an image, considering only top and left neighbouring pixels for each position is fast but misses any diagonal merges. It is much more common to use 4 or 8 neighbourhood connectivity. For reasons of scan efficiency we use a rectangular L-shaped neighbourhood in a typical row-wise rightward scan path (see [75] for a discussion on paths).

The use of a single-pass row-wise scan leads to some trivial oversegmentation. Some regions paths are at least partially unconnected after the first pass (segments 2 and 4 in Figure 2) and a second pass is required [186] to correct this.

Region merging, a form of agglomerative hierarchical clustering, lags behind

the state of the art for the reason that it most commonly applies a greedy merge mechanism which produces a lower quality segmentation caused by a tradeoff between resolution and accuracy [9]. Attempts to adaptively change the threshold or the merge criteria on the fly [38, 181] have improved the results to an extent, but have not managed to match the best performing segmentation techniques. The problem may be attributed in part to the greedy nature of merging schemes. The best local merge is not guaranteed to be optimal in a global sense. However, greedy merging is what gives the technique its speed, and applying extended merging criteria [99, 33] reduces this advantage.

The region merging framework consists of three main components [33]: the region model, the merging criterion, and the merging order. Some common merging criteria are given in [68] and [33] which also explores various options and combinations of these components.

The work by Mignotte [132] compares the performance of several common distance measures over their segmentation results on the Berkeley Segmentation Dataset [127]. The measures compared are the Bhattacharya, Euclidean, Manhattan, Chord, Kolmogorov, Histogram intersect, Kullback, and Shannon-Jensen distances. Their results show that the Bhattacharya and Manhattan distances perform the best with respect to four important segmentation evaluation metrics, the PRI, VOI, GCE and BDE. Of the two, the Bhattacharya distance involves the summing of probabilities and is more complex to calculate than the Manhattan distance.

We improve this foundation by proposing a fast and effective novel region merging method that outperforms other state-of-the-art algorithms. Our method favours the best merge [203, 53] over the fast merge [38] and multiple merges can occur over each iteration as opposed to methods such as hierarchical stepwise optimisation [22] and region based automatic segmentation [1]. The resulting seg-

ments represent a first abstraction of the input as intermediate features according to a Gestalt-like common fate of colour homogeneity.

The region growing method proposed in this paper, semi-greedy adaptive-threshold method (SGAT), may be thought of as a refinement of the Beaulieu-Goldberg hierarchical stepwise optimisation (HSWO) algorithm [22]. The merging criterion used is the best merge [53] and the distance measure is the fast and effective [132] Manhattan distance. We preprocess the image with a median filter [154] to reduce noise. We also use a simple scheme for adapting the distance threshold as the segmentation progresses, similar to dynamic merge relaxation in [175], in order to achieve a balance between data reduction and correctness.

Two region reduction techniques in previous work are the phagocyte and the weakness heuristics [28]. The phagocyte heuristic acts so as to smoothen or shorten region boundaries while the weakness heuristic joins regions based on the strength of the boundary that separates them. The phagocyte heuristic is less general since objects in real world images are not always expected to have smooth boundaries. The weakness heuristic is more general in the sense that similar segments separated by a weak boundary are likely to belong together and thus may be merged. We use the weakness heuristic to clean up the segmentation and reduce the number segments.

2.4 Dense motion estimation

The literature broadly shows three approaches to motion segmentation. The first clusters motion-based feature points [179, 147, 61, 195, 112, 13], the second establishes motion contours and then performs boundary completion to get closed motion regions [71], and the third combines feature points and clustering to perform

region completion [88]. Problems faced by techniques in the literature include difficulty dealing with multiple motion regions, sparse representation, a prior assumption of low velocity motion, or high computational complexity.

Following the first approach, one such method [13] describes a motion superpixel adjacency graph on which graph cuts are applied to get a clustered set of superpixels, each representing individual object motion. Its drawbacks are halo effects and high complexity for more than two regions. A related method using tensor voting [61] establishes tensor points and applies the graph cuts thereafter. This has similar disadvantages, having a high complexity and additionally offering poor performance in low texture environments. Other recent examples of motion feature clustering [147, 112] follow the same general principle. Following the second approach, establishing region enclosing motion boundaries, is for instance the saliency based boundary completion scheme [71] which falters when there are large gaps in the motion boundary or when the motion is detected at coarse scales. The third approach, involving completion schemes to fill holes in motion features and solidify regions, is for instance the spatial clustering approach [88] to link motion features.

These and other approaches [105, 179, 195] face one or more of the following limitations: a) poor performance due to shadows/halos/outliers caused segmentation errors, b) difficulty in dealing with multiple motion regions, c) motion representation as clouds instead of solid regions, and d) high computational complexity.

The following general sources of problems are encountered in motion segmention (adapted from a list of background modelling problems [180]):

- Generalised aperture: Using small regions to identify motion reduces the quality of the motion segmentation due to a lower signal to noise ratio and insufficient spatial data for complete motion judgements. On the other hand,

large regions may consist of more than one motion, but to identify these as multiple motion we need the motion segmentation first.

- Waving trees: Constant periodic motion of background objects can blur the boundaries of foreground motion, since the periodicity of motion may lead to the background optical flow approximately coinciding with the foreground optical flow at regular intervals.

- Camouflage: A foreground object's pixels can sometimes have very similar intensity or pattern as the background object making the object difficult to detect. While this applies to the human visual system as well, the problem is more pronounced in software.

- Foreground aperture: When a homogeneously colored object moves, changes in interior pixels cannot be detected. Thus, the entire object may not appear as foreground but only parts thereof, usually defined internal or external contours.

- Sleeping person: When a foreground object stops moving it is hard to distinguish the motionless foreground object from other background objects using pixel differences or optical flow. This is technically valid grounds to 'lose' a motion object, however it is useful to be able to differentiate between when an object stops moving because it has disappeared entirely and when it has stopped moving and is simply hiding motionless with no pixel differencing response.

- Walking person: When an object starts moving, both the object and its newly exposed background appear as a motion response. Separating the response

region into foreground and background is not easy using motion appearances alone.

- Shadows: Moving objects often cast shadows and can result in identifying the shadow regions as foreground. Again this is technically valid, since the shadow is indeed moving as well, but it is convenient to be able to differentiate between 'solid' motion and transparent shadowy motion.

These problems apply to both sparse and dense motion estimates, but particularly affect image differencing [157, 58] based schemes, which at the most basic level simply subtract one image from another to get the scalar magnitude of intensity change for each pixel between frames. Optical flow [21] is the set of techniques that estimate both the magnitude as well as the direction of change for every pixel, and thus gives more complete information than image subtraction.

Optical flow attempts to track points over frames. Two of the best known algorithms are the classic Horn-Schunck [101, 102] and the Lucas-Kanade [124, 17] optical flow methods, many variants of both of which are described in the literature. Both are differential methods, the most widely used technique in optical flow. The Horn-Schunck algorithm is a global method which iteratively minimises a energy functional by assuming brightness constancy or flow smoothness. While it returns a fully dense flow field, it is sensitive to noise [20, 82] and its iterative nature makes it slow. The Lucas-Kanade method is a local method that solves flow equations for neighbourhood pixels using the least squares method. It also assumes a local brightness constancy and additionally small motions, but is more robust to noise has been generally[1] seen to work better in practice [20, 82]. It is also

[1] Although careful parameter tuning with the Horn-Schunck method has been shown to produce superior results in most cases [12]

possible to produce very fast implementation of this algorithm. Due to reasons of robustness to noise and faster performance, our optical flow technique of choice in Section 3.4 will be a modified dense implementation of the Lucas-Kanade method.

2.5 Segmentation evaluation and benchmarking

Various reviews [212, 216, 215] on segmentation evaluation methods identify several approaches to the evaluation of segmentation schemes. These approaches are subjective evaluation, system-level evaluation, analytical methods, supervised evaluation, and unsupervised evaluation. The most widely used is subjective evaluation, human subjects being by far the best suited to the task of segmentation. Subjective evaluation, while being intrinsically both time consuming and subjective, is commonly accepted as producing the highest-quality evaluation results. Analytic methods evaluate properties of the algorithm independent of the actual output produced, and are thus only applicable for algorithmic or implementation properties. Supervised methods compare the discrepancy between a given segmentation and its corresponding ground truth, usually obtained through manual segmentation. These methods provide a direct comparison between segmentation and ground truth and, while to a degree still being time consuming and subjective, is the most commonly used method for objective evaluation. Unsupervised, or empirical goodness, methods do not require a ground truth but instead evaluate properties of a segmentation according to certain mathematical characteristics of a good segmentation defined by humans. Unsupervised methods are quantitative and objective, but there is no guarantee that the mathematical property being measured is both a sound and complete indicator of segmentation quality, as all the measures developed so far have not been.

There are three reasons why the available unsupervised evaluation measures lack either or both properties of soundness and completeness. First, it is difficult to devise a measure that is rich enough to capture the complex motivations behind a human segmentation. Second, the various definitions of a good segmentation are largely heuristically motivated. For instance, one element of a good segmentation, in [94], is that region interiors should be simple and without holes. This heuristic would fail if trying to segment an image of a slice of Swiss cheese, an object which comes with lots of holes in it. Third, if there were to exist a sound and complete evaluation measure of a segmentation then the perfect segmentation algorithm could be obtained by expressing the measure as some function of pixels and simply optimising the function until the lowest error is achieved for a given image. However, unsupervised evaluations are still useful in that they allow some sort of comparison between methods, such as [170], even if it is true that one can not conclude which algorithm is better for the task purely based on such an evaluation.

Various unsupervised evaluation metrics are described in the literature, including Zeb [36], F_{RC} [155] and V_{CP} [51], which compared to some other evaluation metrics have been shown in [212] to be more balanced with respect to under-segmentation and over-segmentation, with only small biases. However, much of the recent work in various types of segmentation have used the following four measures to quantitatively evaluate performance: probabilistic Rand index PRI [149], variation of information VOI [131], global consistency error GCE [127] and boundary displacement error BDE [78].

Quantitative measures are typically calculated over standardised benchmark data sets so that different segmentation techniques can be compared, but there is a limited set of publicly available largescale general benchmark databases with

ground truths for segmentations on natural images and videos. The LabelMe dataset [158] is a collection of annotated images of natural and cluttered scenes from multiple views but it only provides rough boundary annotation for objects instead of fine contours. The Caltech 101 data set [70] provides fine contour annotations for objects centered in the images and not in natural contexts. The LHI Segmentation dataset [207] is a larger and more diverse dataset than the Caltech 101, following similar annotation principles but providing more diverity of views and contexts while still being limited in the number of components per image. The Caltech 256 [89] is a larger and more diverse version of the Caltech 101 but as of the time of writing lacks annotations, thus making it unsuitable for segmentation evaluation, but being suitable for semantic image interpretation since it consists of groups of categorised objects. The Berkeley Segmentation Dataset [127], while limited in scale and content, provides a well defined error control and benchmark procedure and additionally has been used in the evaluation of several segmentation algorithms making it easier to compare work. An advantage of this data set is the large number of regions available in the images and identified by the ground truth. The Hopkins 155 dataset [183] consists of motion sequence videos and ground truth, providing annotated sparse motion clusters, and is popular for the evaluation of feature tracking based motion segmentation. This however has to be done indirectly for dense motion estimation since its annotations are sparse. The Hopkins set is to the best of our knowledge the only publicly available motion segmentation dataset of its size, annotation resolution and contextual variability. Another motion segmentation dataset [122] provides an annotation tool for obtaining ground truths from videos, but also provides a few pre-annotated video sequences. Another large data set is the Corel Stock Photos collection, although it has no well-defined benchmark procedure or annotations.

2.6 Shape representation and matching

Much of the literature in shape analysis treats the two tasks of representation and similarity matching as one, as many of the review papers will demonstrate [130, 123, 192, 191, 210, 156, 104]. We differentiate between shape similarity measures and shape representation methods. While the two are often linked, there is a distinct difference between measures that numerically describe qualities of a shape and methods that help approximate the actual shape itself. This differentiation is pointed out by Mehtre [130] who classifies each approach as either unambigious/information preserving (IP) or ambiguous/non-information preserving (NIP). The cited review papers provide an extensive exposition on the various IP and NIP methods available.

Of particular interest to us are IP shape representations and those NIP measures that can be used alongside them. A good IP representation is one that is compact but which has high shape approximation accuracy. A compact representation reduces the data space, and it is then faster to calculate NIP measures from the reduced space. This is important since the demand for online image retrieval mechanisms places great emphasis on matching speed. The most prominent of IP shape representation methods are convex hulls [169], polygonal approximation [168, 5], chain codes [76, 204], medial axis transform or skeletal graphs [25, 173] and self-organising maps [110, 109, 166, 171]. These allow contour approximation with various degrees of ease. In contrast to the few main IP methods, there exist hundreds of NIP measures described in the literature. NIP measures are generally applied to an already identified solid shape, contour or feature points, and therefore depend on some prior segmentation mechanism. The following is an extensive, though by no means comprehensive, list of NIP measures, the detailed description of which

we leave to the cited literature [130, 145, 123, 192, 191, 156]: area, perimeter, other size functions, circularity, squareness, triangularity, rectangularity, rectilinearity, sigmoidality, chirality, eccentricity, ratio of principal axes, elongation, major axis orientation, euler number, concavity tree, holes, shape number, convexity, symmetry, compactness, circular variance, ellipticity, elliptic variance, bending energy, arc height, moments (invariant, Zernike, pseudo-Zernike), spherical harmonics, principal components, curvature scale space, voting schemes (geometric hashing, pose clustering, alignment), transformation space subdivision (Hough transform, Walsh transform, Wavelet transform, Fourier transform), boundary and region decomposition (finite point sets, corner, break point, smooth join, crank, end, bump), minimum weight matching, uniform matching, minimum deviation matching, distance (Euclidean, Hausdorff, Frechet, Minkowski, Bottleneck, Earth mover's, Chamfer, etc.), area of symmetric difference (template metric), tangent, acceleration, tangent angle, cumulative angle, periodic cumulative angle, other turning functions, signature function, affine arc length, reflection metric, shape histogram, and graph spectra.

Many of the NIP measures described above use angular, tangential or other curvature measures that rely on a connected point-based contour description. We had previously identified the main IP methods as including polygonal approximation, chain codes, skeletal graphs and self-organising maps. Out of these it is easiest to obtain a contour representation of connected points using self-organising maps (SOM). The SOM was introduced by Kohonen [108, 110, 109] and significantly later applied to shape matching [166, 171].

SOMs are neural networks that adapt to a set of inputs by modifying their connection weights, called training. To increase training flexibility the SOM was adapted into the Neural Gas (NG) [128] and further to the Growing Neural Gas

(GNG) [80, 81]. The GNG has only very recently been explicitly applied to the tasks of shape modelling and registration [69, 10, 214].

2.7 Scene analysis

Most work related to the semantics of visual information comes from the area of content based image retrieval (CBIR) [167, 190, 56]. While some formal languages such as description logics [14] and autoepistemic temporal modal logics [165, 116] attempt to model acquired information with respect to its semantics, they only address knowledge representation and its manipulation, and not the acquisition of such knowledge. CBIR, comprising any technology that helps organise visual data by content, on the other hand indirectly addresses semantic representations by using relatively low level feature comparisons to indicate the presence or absence of semantic similarity. These systems are split into three broad categories, which display varying levels of semantic expressiveness. These are Query By Visual Example (QBVE) [98], Query By Semantic Example (QBSE) [152] and Query By Contextual Example (QBCE) [151].

To assess similarity in CBIR, image or region features are extracted and compared between images. The first task is the mathematical representation of images or regions. Some features used colour, texture and/or simple shape-based information. The second task is the process of estimating similarity between signatures so as to maintain both abstract generality and discriminative ability. An extensive list of shape similarity features, measures and representation methods have been outlined in the previous section. Appropriate features make both tasks easier and more accurate. Colour and texture are very low level metrics to judge similarity by and therefore shape features are receiving more attention now. Shape-

based abstract signatures are more capable of capturing abstraction and present less computational load compared to texture scale selection and colour histogram comparison.

Three prominent, although vaguely similar, architectures in region-based CBIR systems are SIMPLIcity (Semantics-Sensitive Integrated Matching for Picture LIbraries) [198], Blobworld [35] and FRIP (Finding Regions In Pictures) [107]. The SIMPLIcity system first performs quick and approximate image segmentation and then calculates mean features of regions for distance comparisons. It represents region signatures using colour, texture and shape. Three colour channel values, three texture values, and shape indicators in the form of three orders of normalised inertia, are used. Colour and texture are emphasised more than shape due to only very simple shapes being used, and shape is ignored in particuular for textured images. The Blobworld system works comparably, using colour, texture and position to describe segmented regions. While shape is not used directly, for texture measurement a pixel-wise scale selection procedure selects that scale for which mean contrast is very low. Region signatures, a mean of all the member pixel features, are then comprised of three colour channel values, three texture measures and the two position coordinates. The three texture measures are polarity, anisotropy and the normalised texture contrast. The FRIP system works similarly as well, performing a quick image segmentation and then computing features for each region. FRIP features consist of thee colour channels, the Biorthogonal Wavelet Frame as a texture measure, the normalised area, location and two shape descriptors, the eccentricity and a Modified Radius-based Signature. As before, region signatures are derived from the mean of all the pixel features contained in the region, and regions are compared pair-wise.

Scene analysis and specifically content based image retrieval involve the classi-

fication of visual components reached via segmentation and representation. Classification methods can be divided into instanced based learning [4, 57, 54] and eager learning systems, which comprise of all other learning systems. Instance based learning is refered to as lazy learning because it does not form abstractions during a series of observations but simply stores instances and waits until a new instance has to be classified at which point a local neighbourhood similarity search and majority class label assignment is done. In contrast, eager learning systems diverts effort to forming abstractions of observed instances and comparing new instances to the set of abstracted classes. Since instance based learning consumes more and more computational resources when searching through a growing set of observations, eager learning is more suitable for online systems that must stay operational indefinitely. While eager learning spends more resources on abstraction, it saves on both classification speed and on storage space. Both speed and space are of critical concern when dealing with learning regions and shapes from large video sequences, the total set of instances from which may rapidly become very large due to the number of regions identified from each frame and the overall number of frames. However, eager learning methods must commit to a single global approximation at the time of observation, a shortcoming that instance based learning does not share. While there have been attempts to combine instance based and eager learning methods into hybrid systems [97], these systems are also characterised extended search times and large space requirements.

A popular eager learning approach is the nearest centroid method [142, 92, 120] which is fast and simple. A variant of this is the nearest shrunken centroid method [176, 120] which shrinks class centroids towards the overall centroid for all classes. Shifting centroids towards the overall mean reduces the sensitivity of the method to outliers.

2.8 Summary

In this chapter we have reviewed some of the relevant literature in image segmentation, focusing on region merging methods and covering region reduction, dense motion estimation, segmentation evaluation methods and benchmarks, shape representation and shape similarity, and finally scene analysis.

In region growing, we find different sources sharing a common foundation but focusing on different elements of the region merging paradigm, such as the best merge and a time-varying threshold, and identify a need to combine some of these different elements into a single model. Also, region reduction is found to usually be expressed in very different terms than the core region growing step.

Motion information has often been used to group segmented regions, but which we will express in later chapters as an extension of the main region reduction step. In motion estimation, the literature shows the importance of dense motion calculation which offers more complete information than a sparse calculation, but dense motion estimation methods are seen to be less reliable and more sensitive to noise than sparse methods.

In shape similarity, we typically find the two tasks of shape representation and shape matching being very closely linked, with the representation method deciding the matching technique that is used. In scene analysis, the literature shows mostly use of NIP measures as shape and image descriptors, with some application of region information to the task, but little use of IP shape representation methods to perform content based similarity searches.

Chapter 3

Segmentation by region merging

This chapter presents a generalised region merging framework, consisting of multi-stage merging that incorporates adaptations such as the more expensive best merge and a time-expanding theshold. The core of the framework is a hierarchical parallel merging model and region reduction techniques. Based on the general framework, a fixed-threshold region merging special case is discussed. All segmentation results are qualitatively and quantitatively assessed across standardised data sets and four evaluation metrics along with a proposed performance summary indicator. The evaluation results demonstrate the superior performance of the proposed segmentation framework.

3.1 Introduction

The region merging framework consists of three main components: the region model, the merging criterion, and the merging order. We work on this foundation and propose a fast and effective novel region merging method that outperforms other state-of-the-art algorithms. Our method favours the best merge [9, 10]

over the fast merge [11] and multiple merges can occur over each iteration as opposed to methods such as [12] and [13]. The resulting segments represent a first abstraction of the input as intermediate features according to a Gestalt-like common fate of colour homogeneity. The full spatial segmentation, applying a semi-greedy adaptive-threshold path based merging scheme (SGAT) and region reduction techniques, consists of the following phases: 1. Algorithmic region merging 2. Region reduction (a) Weakness heuristic region reduction (b) Small segment reduction (c) Enclosed region absorption.

The proposed system aims to obtain a set of regions that reflect regions of primary saliency in the input, but which also retains a level of detail where the visual importance of localised zones is high. It is region-based for reasons of low complexity and the need to prioritise topological proximity. The system's two-stage operation first quickly establishes class labels from neighbouring pixel colour distances, and then further merges the preliminary set of classes or segments. Since we consider both colour and segment size in the second merging stage, fewer and larger segmented regions are obtained except where significant local features force the separation of smaller areas.

3.2 Preprocessing

Some problems are encountered, before the actual segmentation step, when attempting to process natural images taken live from a web camera. For instance, there is a some amount of noisy irregularity within captured images due to hardware constraints. Illumination levels are also affected by changes in natural and artifical lighting present at different times and by miscellaneous transient light sources. Moreover, the colour space used to represent the input image plays a

part in deciding the segmentation pathway, some formats being more perceptually uniform than others at the cost of computational efficiency and some separating out information such as luminance and chrominance. We next discuss how we deal with these problems.

The image is first convolved with a 3x3 median filter to reduce "shot" noise and to provide some median blurring. This helps improve immediate neighbour pixel grouping and, due to the edge preserving nature of the median filter, without significantly reducing the separation between local zones of region dissimilarity. The median filter [154] is represented as:

$$y(m, n; W) = med\{x(m - k, n - l), (k, l) \in W\}, \tag{3.1}$$

where W in this case is a 3x3 window or filter mask. The window size is kept at the non-trivially smallest possible value so that shot noise reduction and very localised smoothing occurs without affecting much of the image details.

At this stage we select a suitable colour space to work in. Among others experimented with, the primary candidates were RGB, CIE XYZ, HSV, CIE Luv and CIE Lab. RGB is the most common format in use for documents, monitors and the Internet. To deal with the problem of negative weights in the RGB colour model, CIE's 1931 XYZ format was introduced. While RGB is considerd psychologically non-intuitive [177], both RGB and CIE XYZ lack perceptual uniformity in Euclidean space. The HSV format is a linear transformation from RGB and is a "phenomenal" colour space, being a natural way for humans to describe colours. However it too is perceptually non-uniform, and additionally poses a poor correlation between computed and perceived lightness [177]. Both CIE Luv and CIE Lab were proposed by CIE as perceptually uniform colour spaces, the main difference between them being CIE Lab normalizes its values by division with the white point while CIE Luv normalizes its values by subtraction of the white point [177].

To convert an RGB image into either of these two perceptually uniform colour spaces requires device dependency considerations, a reference white $\{X_r, Y_r, Z_r\}$ for the XYZ representation of the image, and a set of transformations involving expensive operations such as divisions and a cube root. The extra computational burden of transformations from RGB to CIE XYZ and then to CIE Luv/CIE Lab is justified if it produces a noticeable difference in the quality of the results, but as verified from our tests, this is not the case for the specific algorithms considered in this research, and we therefore continue with the most commonly encountered RGB colour space.

We therefore have a three component $\{R, G, B\}$ feature vector, where feature distances are calculated between individual pixels or groups of pixels. To keep the complexity of the method low, an efficiently computed distance measure must be established. We use the Manhattan distance,

$$dM = \sum_{i=0}^{n} |U_i - V_i|, \tag{3.2}$$

where U and V are the feature vectors for the two sets of pixels being compared, and n is the length the feature vector, which in our case is 3.

3.3 A generalised region merging framework

A number of region growing methods in the literature bear anywhere from a passing to a striking resemblance. The Beaulieu-Goldberg hierarchical stepwise optimisation (HSWO) [22] and the Adams seeded region growing (SRG) [3] methods each start with a set of initial regions and iteratively merge pixels to them one at a time based on the smallest neighbouring distance out of all the regions and their immediate neighbours, the difference being HSWO starts with every pixel as a

"seed" while SRG uses a smaller selection of seeds. SRG starting with 100% seed density is nearly equivalent to HSWO. Bailey's raster based method (RB) [15] is similar to both HSWO and SRG, except it performs parallel merges at each iteration. While HSWO and SRG indirectly perform a best merge [203, 53], since at each iteration only the pair of regions with the overall minimum pairwise distance are merged, RB uses a fast merge [38]. In fast fuzzy C-means (FFCM) [23] a very similar procedure of scanning and labelling is followed to set up a JND (just noticeable difference) histogram, after which histogram agglomeration (bin merging) helps the algorithm to proceed. It too uses the fast merge distance measure for histogram computation, that is a colour is added to the first bin that it is similar to, not the nearest bin. FFCM has similarities to HSWO, SRG and RB, the difference being it operates in colour feature space while the other three algorithms work in physical space with immediate neighbourhood constraints.

Consider also the following equivalence. Region merging using a region adjacency graph (RAG) is described as follows [27]: "Task: Merge neighbouring regions R_i and R_j. Phase 1. Update the region-adjacency graph. 1. Place edges between R_i and all neighbouring regions of R_j (excluding, of course, R_i) that do not already have edges between themselves and R_i. 2. Delete R_j and all its associated edges". Step 1 is equivalent to R_i adopting all the boundaries that R_j has, which also incidentally means if R_i were to be previously connected to a third region of which both R_i and R_j are neighbours then R_i would now be disconnected from it. Step 2 is then equivalent to finishing the R_i-R_j merger joining the interior of R_j to R_i as well.

Brox's multistage region merging (MRM) [30] also uses a RAG to perform merging, "the algorithm proceeds by continuously searching for the edge with the lowest dissimilarity value and merging the two regions". This is equivalent to a

sequential minimum neighbouring distance merge algorithm along the lines of HSWO and SRG. Brox however identifies some alternative distance measures for the merge criterion. The graph based method in [62] uses a texture based variation for the merge criterion.

Similar equivalences are seen to apply to tree based region growing [34, 60], adaptive threshold region growing [38], and other approaches [73, 65], to name a few. While many of them use different distance measures and merging criteria, they share a very similar foundation. The aggressive region growing (ARG) approach [41] is different in the sense that a single region is grown to its maximum extent before any others are considered. While such variations on a common theme are expected, it is helpful to recognise them as variations that share a common foundation. This shared foundation motivates the formalisation of a single general region growing framework.

We now propose a general region growing framework, semi-greedy adaptive-threshold region growing (SGAT), which performs region merging on N clusters by calculating at each iteration the distance measures $C_{i,j} = d(S_i, S_j)$ for all cluster pairs (S_i, S_j), with the most similar pair of clusters being merged and this procedure being repeated until a stopping criterion is satisfied. The merging criterion used is the best merge [53] which requires the consideration of only a pixels neighbours and its neighbours' neighbours, thus allow multiple best merges to take place during a single iteration. The distance measure is the fast and effective [132] Manhattan distance, $d(S_i, S_j) = \sum_{i=0}^{n} |U_i - V_i|$, where U_i and V_i are the feature vectors for segments S_i and S_j respectively, and n is the length the feature vector. We preprocess the image with a 5×5 median filter to reduce noise, the window size being determined by visually assessing the effect of various filter windows on the overall noise level in the picture content when using the experimental hardware.

3.3.1 Segmentation strategy

Using an L-neighbourhood[1] row-wise scan offers up to L local merges, but a potential merge (S_i, S_j) for segment i may be suboptimal for neighbouring segment j if there exists a neighbour k of j for which $C_{j,k} < C_{i,j}$ is true, indicating a better merge. Thus a merge over an L-neighbourhood should therefore also check the full neighbourhood of the target of the merge, such semi-greedy behaviour resulting in a more optimal merge.

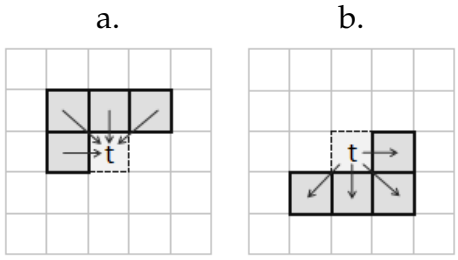

Figure 3.1: Neighbours of pixel t for row-wise rightward scans. a) Seeking labels from $\{TL, T, TR, L\}$, b) Pushing labels onto $\{R, BL, B, BR\}$

We encode the eight possible neighbours for any pixel t as $\{TL, T, TR, L, R, BL, B, BR\}$, (top left, top, top right, left, right, bottom left, bottom and bottom right respectively). For computational efficiency and algorithmic correctness we wish to have the smallest neighbourhoods that exhibit two characteristics: a) the neighbourhood should contain either only already-labelled or only yet-unlabelled pixels, and b) the neighbourhood should not miss any possible merge pathways. An L-shaped neighbourhood reduces scan redundancy while not missing any merge pathways. The two minimum L-neighbourhoods for a row-wise scan are $\{TL, T,$

[1] An L-neighbourhood consists of some subset of the following immediate neighbours: top-left (TL), top (T), top-right (TR), left (L), right (R), bottom-left (BL), bottom (B) and bottom-right (BR).

TR, $L\}$ and $\{R, BL, B, BR\}$, see Figure 3.1. Given the usual row-wise nature of image scans, the former set is suited for pull-labelling, in which the pixel currently being considered seeks to adopt or pull a label from one of the L-neighbours, while the latter set is suited for push-labelling, in which the current pixel seeks to propagate or push its own label onto one or more of the L-neighbours. In our work we use the 4-neighbourhood $\{TL, T, TR, L\}$ to scan for a potential merge and the 8-neighbourhood $\{TL, T, TR, L, R, BL, B, BR\}$ to check if it is the best merge for both involved segments.

The feature distance $d(S_i, S_j)$ between two segments and the cost $C_{i,j}$ of merging them are equivalent. Merges take place whenever $d(S_i, S_j) < d_{max}$. We use a simple scheme for adapting the distance threshold as the segmentation progresses, similar to *dynamic merge relaxation* in [175], in order to achieve a balance between data reduction and correctness.

The full segmentation consists of the following two phases:

1. Algorithmic region merging
2. Region reduction
   (a) Weakness heuristic region reduction
   (b) Small segment reduction
   (c) Enclosed region absorption

The key controller of the row-wise scan is the variable t which is the pixel index currently being considered. Incrementing t has the effect of moving through the image row-wise from left to right, the minimum L-neighbourhood resulting from looking only at all the pixels already labeled in the past within the current iteration. The variables involved are as follows:

1) B_i, the set of the segments adjacent to S_i, called the neighborhood,

2) D_i, the parameters that describe the segment S_i, e.g. the segment R, G, B

means,

3) $C_{i,j} = C(D_i, D_j)$, the cost of merging segment S_i with S_j, where S_j is contained in B_i,

4) d_{curr}, the distance threshold that restricts merges if the cost of merging is greater than this value. The threshold grows after each iteration as $d_{curr} = d_{curr} + d_{step}$.

5) d_{max}, the maximum allowable distance threshold such that: $d_{curr} \leq d_{max}$.

The region merging algorithm is then defined as:

I. Initialise:

(i) $Ind = \{1, 2, \cdots n\}$ (image pixel indices).

(ii) $P^0 = \{S_1, S_2, \cdots S_n\}$ (initial partition).

(iii) Label(t), $t \in Ind$ (segment label for pixel t).

(iv) $k = 0$, $m = n$ and $d_{curr} = 0$.

(v) $\forall S_i \in P^0$, calculate D_i and B_i.

(vi) $hasMerged = false$

II. Merge, $\forall t \in Ind$, 4-neighbour scan $\{TL, T, TR, L\}$:

(i) $i =$ Label(t).

(ii) calculate $CS_i = \{C_{i,j} | S_j \in B_i\}$.

(iii) find $C_{u,v} =$ Minimum($C_{i,j}$) where $C_{i,j} \in CS_i$, and Minimum($C_{v,t}$) $\geq C_{u,v}$ where $C_{v,t} \in CS_v$ in a full 8-neighbourhood $\{TL, T, TR, L, R, BL, B, BR\}$.

(iv) if $C_{u,v} \leq d_{curr}$, do Merge(S_u, S_v) as follows:

a) $k = k + 1$ and $m = m + 1$.

b) $P^k = (P^{k-1} \cup \{S_m\}) \cap \overline{\{S_u, S_v\}}$.

c) calculate D_m from D_u and D_v.

d) $B_m = (B_u \cup B_v) \cap \overline{\{S_u, S_v\}}$.

e) $\forall S_j \in B_m$, $B_j = (B_j \cup \{S_m\}) \cap \overline{\{S_u, S_v\}}$

f) $hasMerged = true$

III. Stopping condition:

(i) if $hasMerged == true$, do the following:

a) $hasMerged = false$.

b) $d_{curr} = \text{Min}(d_{curr} + d_{step}, d_{max})$.

c) go to step II.

(ii) stop.

The following section discusses the region reduction post-processing that we next apply.

3.3.2 Region reduction

We use the weakness heuristic [28] to clean up the segmentation and reduce the number segments. The variables involved in the weakness heuristic region reduction step are as follows:

1) for two regions S_i and S_j where $S_j \in B_i$, let $L_i = \{l_{i,1}, l_{i,2}, \cdots l_{i,q}\}$ and $L_j = \{l_{j,1}, l_{j,2}, \cdots l_{j,r}\}$ be the set of all boundary pixels in S_i and S_j respectively,

2) let Adjacent($l_{i,u}$, $l_{j,v}$) be a boolean function determining whether pixels $l_{i,u} \in L_i$ and $l_{j,v} \in L_j$ are adjacent. We find three adjacent pixels in S_j for every pixel in S_i along smooth boundaries, which we compensate for in the boundary strength computation.

3) then $L_i^j \subset L_i$, where $\exists u \in L_i, v \in L_j$ for which Adjacent($l_{i,u}$,$l_{j,v}$) $= true$.

4) d(r,s) is the feature distance between pixels r and s.

5) $f_{i,j} = |L_i^j|$, the approximated length of the common boundary between S_i and S_j

6) $\forall l_{i,u} \in L_i^j$, mean boundary distance between segments S_i and S_j, $CT_{i,j} =$

$\frac{1}{3}\frac{1}{f_{i,j}}\sum \mathrm{d}\left(l_{i,u}, l_{j,v}\right)$, where $\exists u \in L_i, v \in L_j$ such that Adjacent($l_{i,u}$,$l_{j,v}$) $= true$.

7) $d_{weakness} = \alpha \times d_{max}$, where $0 < \alpha < 1$ is the weakness control factor, some fraction of d_{max} since weak segments are more similar, that controls the merging of weak segments.

In this step we use only a 1-neighbour TL scan for faster processing without significant degradation in quality[1]. Since we proceed from the main segmentation step, no initialisation is required for the region reduction post-processing phase, which is given by the following:

I. Region reduction, $\forall t \in Ind$, 1-neighbour scan $\{TL\}$:

(i) $hasMerged = false$.

(ii) $i =$ Label(t).

(iii) calculate $C_{u,v}$ as $CT_{i,j}|S_j \in B_i$.

(iv) if $C_{u,v} \leq d_{weakness}$, Merge($S_u$, S_v) as follows:

a) $k = k + 1$ and $m = m + 1$.

b) $P^k = (P^{k-1} \cup \{S_m\}) \cap \overline{\{S_u, S_v\}}$.

c) calculate D_m from D_u and D_v.

d) $B_m = (B_u \cup B_v) \cap \overline{\{S_u, S_v\}}$.

e) $\forall S_j \in B_m$, $B_j = (B_j \cup \{S_m\}) \cap \overline{\{S_u, S_v\}}$.

f) $hasMerged = true$.

(v) if $hasMerged = true$, repeat from step I.

Next we reduce the number of small segments by performing a single iteration of the main region merging algorithm via a full 8-neighbour scan (to avoid missing possible merge pathways) by setting $d_{curr} = \infty$ and considering $i =$ Label(t) in a

[1] A 1-neighbour (top-left) row-wise scan using the weakness heuristic has a 'blind spot': segment boundaries aligned at a perfect $45°$ on the left diagonal, which are unlikely to occur in natural images.

row-wise scan only for segments S_i such that the segment size $|S_i|$ satisfies the condition $|S_i| \leq \gamma \times \frac{|Ind|}{|P^{k-1}|}$, where the total image size is represented by the number of pixels contained $|Ind|$ and the control parameter γ decides the smallness to be merged in terms of some fraction of the ratio between image size and the number of segments obtained thus far.

Finally, we extend the weakness heuristic to include regions completely enclosed by another region, many of which represent less salient details that may be merged away. We use a second weakness control factor β to reduce the cost of merging such regions, leading to only salient enclosed regions remaining unmerged. The main region merging algorithm is repeated for one iteration via a full 8-neighbour scan after setting $d_{curr} = \beta \times d_{max}$ and considering $i = \text{Label}(t)$ in a row-wise scan only for segments S_i in which $|L_i^j| = |L_i|$.

3.3.3 Results

Four commonly used segmentation evaluation metrics are the probabilistic Rand index (PRI) [149], variation of information (VOI) [131], global consistency error (GCE) [127] and boundary displacement error (BDE) [78], each having established figures for results using important segmentation methods. Since we wish to compare segmentation performances based on values for these four different metrics, we propose a new performance summary indicator as a function of multiple evaluation measures. This performance indicator uses the results for human segmentation as a baseline. In some evaluation metrics lower values show better performance while for others higher values show better performance. We therefore use a scheme that adds a positive term to the performance indicator for metric results better than the baseline and penalises metric results worse than the baseline. We also take into account the fact that different metrics produce varying ranges of val-

ues and differences in some metrics may have less influence than others. We thus normalise each metric result by the relative importance of that metric with respect to the complete set of considered metrics.

Desirable properties of the performance indicator are:

1) the indicator should have a zero value when the baseline is compared against itself.

2) the indicator should have a specific constant value when the same pair of human and algorithm results are being compared, independent of the number of algorithms compared.

We call this new indicator the relative performance (RP), which we now define. Let H_i and A_i respectively be the baseline (human) and challenger (algorithm) results for metric i for n different metrics, also let $\lambda_i = 1$ when higher values are better and $\lambda_i = -1$ when lower values are better for metric i. Then the relative weight of each metric in terms of the baseline is given by $W_i = \frac{H_i}{\sum_{j=0}^{n} H_j}$.

Then RP is defined as:

$$RP = \frac{1}{n} \sum_{i=0}^{n} \frac{\lambda_i \times (A_i - H_i)}{W_i} \tag{3.3}$$

We benchmark the spatial and spatiotemporal segmentation methods against the Berkeley Segmentation Dataset (BSDS) [127] according to the four evaluation metrics of PRI, VOI, GCE, BDE and the summary indicator RP. We use these measures to quantitatively evaluate our segmentation results against the figures reported in [206, 95, 150, 133, 47, 39]. The segmentation methods compared against are the following: Felzenszwalb & Huttenlocher Graph-based (FH) [72], Mean Shift (MS) [50], Normalised Cuts (NC) [164], Multiscale NCut (MNC) [52], Markov Chain Monte Carlo (MCMC) [185], Fusion of Clustering Results (FCR) [132], Compression based Texture Merging (CTM) [206], Ultrametric Contour Maps (UCM)

	Performance Measures				
Algorithms	PRI ↑	VoI ↓	GCE ↓	BDE ↓	RP ↑
Human	0.8754	1.1040	0.0797	4.9940	0
FH [72]	0.7841	2.6647	0.1895	9.9497	-6.86
MS [50]	0.7550	2.4770	0.2598	9.7001	-8.08
NC [164]	0.7229	2.9329	0.2182	9.6038	-7.92
MNC [52]	0.7559	2.4701	0.1925	15.10	-8.49
MCMC [185]	0.768	2.261	-	-	-
FCR [132]	0.7882	2.3035	0.2114	8.9951	-6.42
$CTM_{\gamma=0.1}$ [206]	0.7561	2.4640	**0.1767**	9.4211	-6.12
$CTM_{\gamma=0.2}$ [206]	0.7617	**2.0236**	0.1877	9.8962	**-5.82**
UCM [11]	0.77	2.11	-	-	-
TBES [133]	**0.807**	**1.705**	-	-	-
HMC [95]	0.7816	3.8700	0.3000	**8.9300**	-10.87
MCSpec [208]	0.7357	2.6336	0.2469	15.40	-10.10
NormTree [197]	0.7521	2.4954	0.2373	16.30	-9.95
BW [35]	0.7138	2.6295	-	-	-
SWC [184]	0.7644	3.0266	-	-	-
$SGAT_{[1]}$	**0.7946**	3.5026	**0.1396**	**5.0237**	**-5.33**
$SGAT_{[2]}$	0.7886	2.8482	0.1911	5.2839	-5.53

Table 3.1: Quantitative comparison of SGAT segmentation results with other methods. Average performance on the BSDS shown. PRI $[0, 1]$, higher is better. VoI $[0, \infty]$, lower is better. GCE $[0, \infty]$, lower is better. BDE $[0, \infty]$, lower is better. Figures not available are marked as '-'. RP $[-\infty, 0]$, values rounded to 2 decimal places. The two best values for each measure are shown in bold, considering only the better performer out of $SGAT_{[1]}$ and $SGAT_{[2]}$.

[11], Texture and Boundary Encoding-based Segmentation (TBES) [133], Hierarchical Markov Clustering (HMC) [95], Multiclass Spectral Clustering (MCSpec) [208], Normalised Tree Partitioning (NormTree) [197], Blobworld (BW) [35], Swendsen-Wang Cuts (SWC) [184].

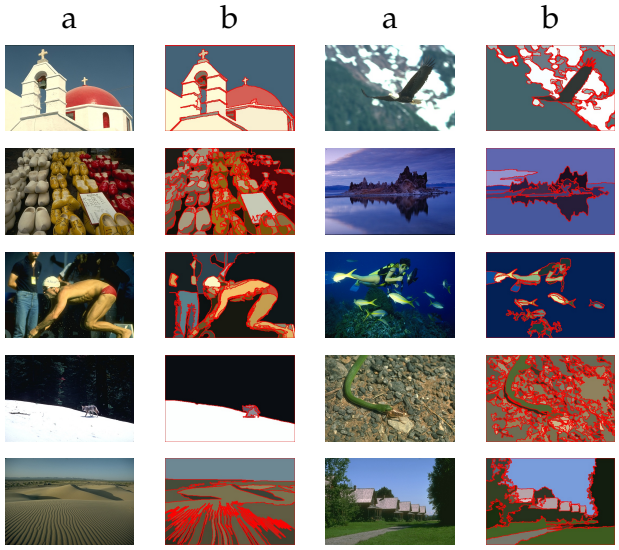

Figure 3.2: Segmentation results on some BSDS images using SGAT$_{[2]}$. a) Original, b) Segmented

We have experimentally determined the following parameter values for SGAT region merging to produce robust results:

SGAT$_{[1]}$: d_{max}=35, d_{step}=3, α=0.55, γ=2, β=10.

SGAT$_{[2]}$: d_{max}=45, d_{step}=3, α=0.50, γ=2, β=10.

As we can see from the RP column in Figure 3.1, the relative performance indicator allows us to compare overall performance for an algorithm across several metrics. Higher values (closer to zero) are better[1]. Both variations of the proposed

[1]A positive value for RP indicates a better performance than the baseline.

SGAT method show better overall performance with respect to the available data for the other methods. We also see that $\text{SGAT}_{[1]}$ produces better quantitative results than $\text{SGAT}_{[2]}$, but both score higher on the overall performance indicator than the other algorithms. All experiments were run on a 2.26 Ghz dual core laptop computer. Average region merging frame rates on the 481×321 BSDS images were 0.5 to 1 fps.

3.3.4 Segmentation special case

We now describe a special case of the general region framework framework described in the previous section. The merging strategy consists of two broad stages: a preliminary pixel-level class label assignment stage, followed by an iterative class merging stage.

The merging procedure is dependent on two threshold values, the pixel merging distance d_p and the segment merging distance d_s. Experimental tuning sets these parameters to $d_p = 10.5$ and $d_s = d_p * r$, where the segmentation factor (inversely proportional to the resolution of the segmentation) $r = 15 * 10^6$ provides a good segmentation of large salient regions while maintaining some smaller regions of high importance.

Intially, a simple Canny edge detector with window size 3 and threshold values 80 and 240 is applied to the image to get a set of edge labels, E where $E_{m,n} = 1$ indicates pixel $P_{m,n}$ has been identified as an edge, and $E_{m,n} = 0$ indicates otherwise.

In stage 1, we carry out a preliminary class label assignment for each pixel based on a threshold value between immediate neighbours. We start with an empty set S of class labels, and move through the entire image row-by-row from top left to bottom right, using a label assignment strategy to populate S with possible class labels s_i, where i is the label counter. On the very first pixel $P_{m,n}, m = 1, n = 1$,

the label counter i is set to 1 and a new element s_1 is inserted into S. Thus $P_{1,1}$ is assigned the class label s_1. We then carry out the following steps until there are no more image pixels to process:

I. Try to assume a neighbouring pixel label using the procedure $Seek$. If this succeeds, move on to the next pixel.

II. If it fails, increment the label counter i and assign the new label s_i to the current pixel, inserting this label into the set of labels S.

A $k \times l$ kernel window K, where $k \bmod 2 \neq 0, l \bmod 2 \neq 0$ and $k > 1, l > 1$, is employed at several stages of the segmentation. We keep the window at the smallest possible non-trivial size, which is $k = 3, l = 3$. When K is positioned over pixel $P_{m,n}$ of image I, the kernel window coordinates are represented by $K_{x,y}$.

$Seek$ procedure: We center the kernel window K over the currently considered pixel and proceed to compare the feature distance, dM (the Manhattan distance) between the center pixel and other pixels of the kernel that lie within the image region. Since our kernel is 3×3, we have eight possible neighbours for each center pixel considered, and thus eight neighbour distances dM_c, c=1:8. For each of these eight, the class label corresponding to the smallest dM_c that falls within the allowable pixel merging threshold d_p is assigned to the center pixel. If none of the dM_c values that satisfy the threshold already possess a class label then this procedure fails.

By the end of this process, we get a label map for all the pixels in the image. However, there remains a problem. Since we proceed from left to right, top to bottom, there are cases when the class labelling splits what should be a single segment into multiple classes. Figure 3.3 shows a simple example where the $Seek$ procedure assigns class labels 2 and 4 to pixels actually belonging to a single class.

The $Seek'$ procedure corrects this and the label 2 is dropped from the set of labels, the entire segment now being assigned the label 4.

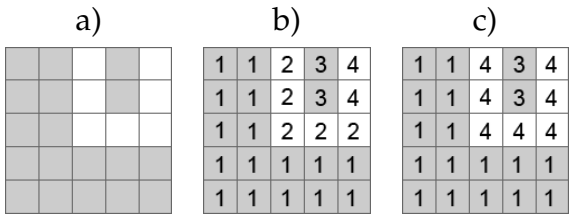

Figure 3.3: Label assignment: a) Image, b) $Seek$, c) $Seek'$

Using $Seek'$, we run through all the pixels on a second pass, this time merging segments indicated by neighbouring pixels satisfying the pixel merging threshold but with different class labels. Although this step could have been avoided by making the first step more complex, we make significant performance gains by having two low complexity passes instead of a single complex pass. Figure 3.4 shows the $Seek'$ pass correcting the initial label assignments established by the $Seek$ pass.

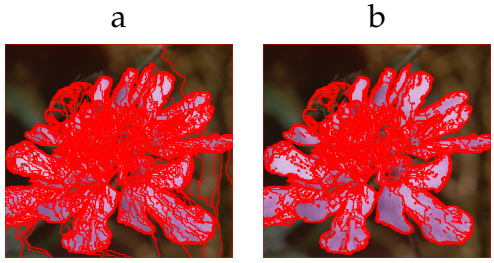

Figure 3.4: Label correction between $Seek$ (a) and $Seek'$ (b)

$Seek'$ procedure: We again center the kernel window K over the currently considered pixel and proceed to compare the feature distance, dM between the center

pixel and other pixels of the kernel that lie within the image region. From the eight neighbour distances $dM_c, c = 1 : 8$, we eliminate those that correspond to pixels with different class labels from the center pixel and get an updated set of neighbour distances dM'_c. Now considering this set, the class label corresponding to the smallest dM'_c that falls within the allowable pixel merging threshold d_p is assigned to the center pixel, using the $Merge$ procedure. If none of the dM'_c values that satisfy the threshold already possess a class label then this procedure fails.

$Merge$ procedure: When merging two pixels or pixel regions with classes u and v, the pixels belonging to both classes are all set to u. Thus the two segments previously identified by labels u and v are now identified by a single label u.

Next we move to stage 2 of the segmentation, which is similar to stage 1, with three differences that redefine the functionality of the $Seek'$ procedure to get a new procedure $Refine$. First, we now only consider the kernel window and pixel level information in order to identify neighbouring segments. Neighbours are now defined as, for kernel K centered at image coordinates $P_{m,n}$, segments for which K contains pixels from each segment and at least one of the segments places a pixel at the kernel center, $K_{((m+1)/2,(n+1)/2)}$. Secondly, the distance measure dM is now between neighbouring segments instead of neigbouring pixels. The feature vector for each segment is recomputed as the mean of the individual feature vectors of their component pixels. The third difference is in the merging threshold, which is now d_s instead of d_p. This threshold d_s is further magnified by the sizes of the particular segments being considered for the merge. The modified procedure is as follows:

$Refine$ procedure: For kernel K at $P_{m,n}$, if segment label s_U at center pixel $K_{((m+1)/2,(n+1)/2)}$ differs from that at another $K_{(m,n)}$, then s_U and s_V are neighbouring segments. For each pair of neighbouring segments identified by an instance of

K, we get a set of distance measures dM_c. The distance measures are calculated as the distance between the means of the feature vectors of all the pixels belonging to each segment. Each distance measure is further multiplied by the number of pixels in each of the two segments being considered, i.e. for segments s_U and s_V with number of pixels N_a and N_b respectively, $dM_c = dM_c * N_a * N_b$.

Finally, the edges are factored in and if for either considered pixel the edge label $E_{m,n} = 1$, we apply a magnifying factor, experimentally determined to be optimal at 500, to get $dM_c = dM_c * 500$. The segment corresponding to the smallest dM_c, and which falls within the allowable segment merging threshold d_s, is merged using procedure $Merge$ with the segment identified by the class label at the center of K. The segment corresponding to the smallest dM_c, and which falls within the allowable segment merging threshold d_s, is merged using procedure $Merge$ with the segment identified by the class label at the center of K.

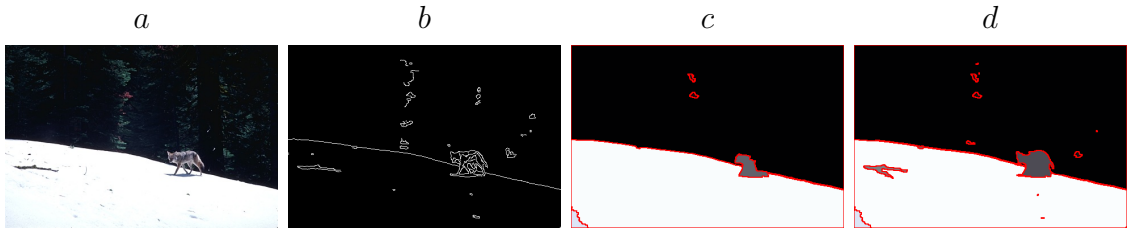

Figure 3.5: a: Original. b: Edges detected. c: Region growing segmentation. d: Edge-enhanced region growing, with increased contour correctness.

For segments larger than 50 pixels we then insert a clause immediately before the $Seek$, $Seek'$ and $Refine$ phases which has the effect of lowering the pixel and segment merging thresholds d_p^M and d_S^M for edge pixels. We use a threshold scaling factor z to control this lowering effect. Assuming the edge map represents pixel edge information $X_{i,j}$ at pixel location i, j as 1 if it is an edge and 0 otherwise, the

merging thresholds become

$$d_p^M = \frac{d_p^M}{z}, for X_{i,j} = 1 \tag{3.4}$$

and

$$d_S^M = \frac{d_S^M}{z}, for X_{i,j} = 1 \tag{3.5}$$

We set z to a high value between 7 and 10. Lowering z scales the merging threshold less and makes it easier for boundary merges to occur, conversely setting it high enough makes it impossible for any boundary merges to take place.

If none of the dM_c values satisfy the threshold then this procedure fails. Thus, starting at the top left of the image and proceeding row-by-row to the bottom right, we iterate through the following steps until in any single run through the entire image no segment merges occur:

I. For the current segment, try to assume a neighbouring segment label using the new procedure $Refine$, modified in the three ways as described above. Move to the next pixel if it *fails*.

II. If it succeeds, recompute properties global to the newly merged segment and its feature vector as the mean of the feature vectors of its component pixels. Move to the next pixel.

The segmentation is now complete. Figure 3.6 shows the two-stage results. For $r = 15 * 10^6$, we typically get within ten and twenty final segments. Note that Figure 3.6-b represents the same processing stage as Figure 3.4-b.

In our system, segments represent zones of interest, and regions where multiple segments are concentrated represent possible points of gaze fixation. Images in which there are fewer and weaker dominant points of fixation are indicated by

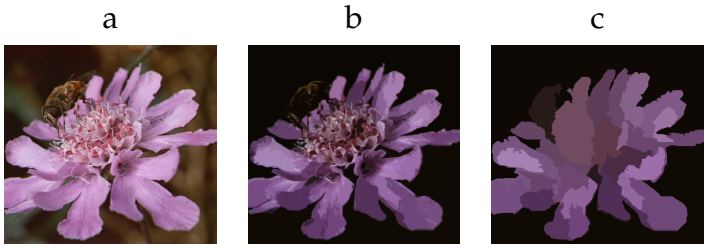

Figure 3.6: Segmentation stages: a) Original image, b) Oversegmented results after Stage-1 $Seek'$, c) Final results after Stage-2 processing

a segmentation set consisting of fewer and larger regions, similar to the human perception taking a little longer to identify something significant to look at in the image.

As seen from the results, we sacrifice uniformly spread local detail in order to gain global saliency zones. The system keeps intact localised small segments only where it finds the visual importance of the segment to be very high. We note however that once broad regions of interest are identified, it is possible to apply the same system at higher resolutions to pick up greater detail from those regions.

In evaluating the performance of our algorithm, we use visual comparisons. As pointed out in [85], benchmark-based segmentation evaluation usually suffers from the trade-off between objectivity and generality. General-purpose segmentations may not have parallel well-defined ground truths. While we have used images from the Berkeley Segmentation Data Set [126], and obtained the corresponding F-measure of the segmentation results (maximal F-measure of 0.512032), "trivial segmentations, where each segment only contains one pixel or the whole image is a single segment, always produce perfect 100% segmentation accuracy in this benchmark" [85]. In this vein, increasing our resolution parameters provides better quantitative results, but makes the output visually less pleasing. Therefore, in our evaluation we visually compare some of our results against those of other

popular algorithms.

Figure 3.7 shows some segmentation results presented alongside the results from three other algorithms ([100], [174] and [111]). The images chosen represent medium to high complexity. As can be seen from the results, our segmentation maintains the balance between saliency and detail, the visual quality of the results being comparable in positive light against the results of the other algorithms.

We have presented a natural colour image segmentation method that is fast and maintains a level of perceptual correlation with the input. The method produces segments that signify salient image regions but which does not eradicate smaller intra-object regions should they present sufficiently salient features.

3.4 Motion-based region reduction

The analysis of motion allows pixel grouping by a further Gestalt principle of similar motion. That is, for video data, segmentation must be extended to consider spatiotemporal or motion information. The simplest indicator of motion is intensity differences between two frames of a video sequence. Motion regions thus identified represent another level of intermediate features in the abstraction hierarchy.

3.4.1 Scalar motion from image differences

Image differences between frames provide the most straightforward indication of motion, and is thus used as the basis for almost all motion segmentation schemes. While image differences are very easy to obtain, interpreting them is much more involved. Three main characterstics contribute to this difficulty. From the list of common motion segmentation problems identified earlier, these are *foreground*

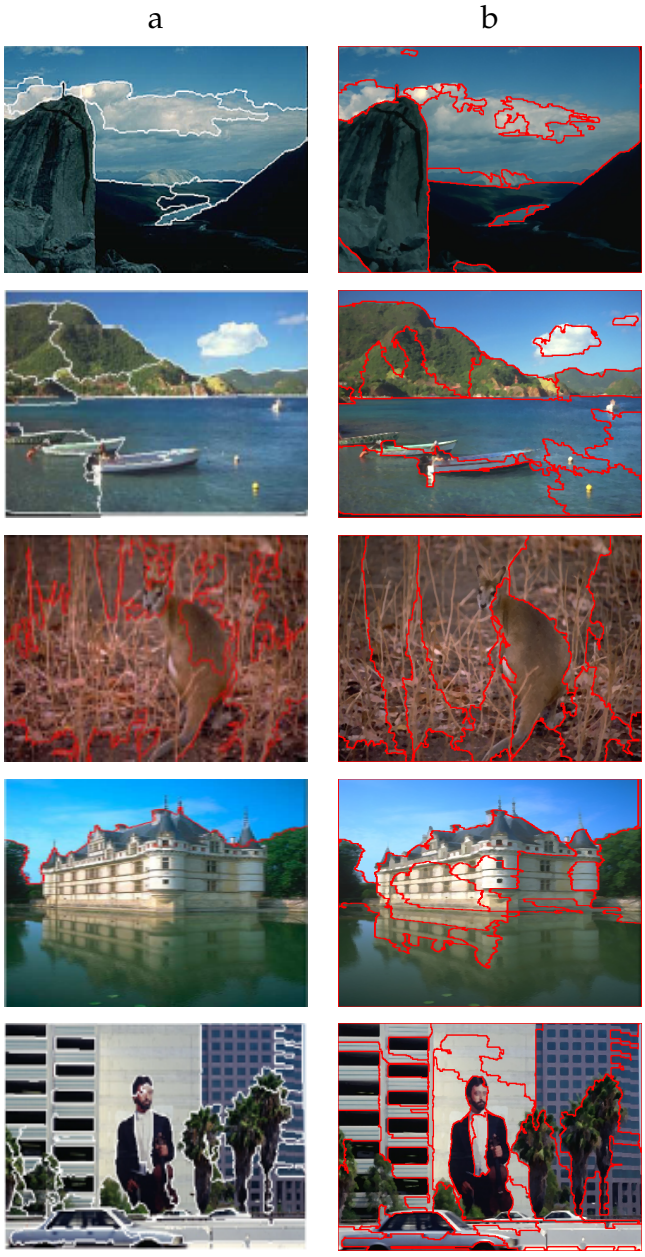

Figure 3.7: Segmentation performance for difficult segmentation problems compared against the performance of other algorithms. Column a) Other algorithms (Row 1: [100], Row 2 [174], Row 3 [111], Row 4 [111], Row 5 [100]), Column b) Our algorithm.

aperture, *sleeping person* and *walking person*. The first of these, *foreground aperture*, is an important property of image differences, describing how boundaries or salient edges of a moving object primarily contribute to the difference image, with the object interior appearing very similar to the other stationary areas of the image. The second, *sleeping person*, simply represents the fact that once an object stops moving or drops below the motion sensitivity threshold it can no longer be detected by image differencing alone. The third, *walking person*, is conceptually a sort of reverse of the aperture problem, appearing as motion response from both a moving object as well as its immediate background which was vacated due to the motion.

These problems motivate the combination of image subtraction with spatial segments, and also a single-step "memory" of previous frame motion and segment information. Solid spatial regions provide an opportunity to fill in the thin contours arising from image subtraction, while motion memory provides the opportunity to retain a temporary lock on a motion segment even when it suddenly stops moving.

The motion segmentation mechanism combines spatial segmentation with temporal image differencing to help get past these problems. While any segmentation technique can be used, the higher the perceptual correlation of the segments the better the motion segmentation results. The following list summarises the motion segmentation procedure, after which a more formal representation is provided.

I. For each frame, obtain a region-based spatial segmentation and a thresholded image difference between the current and previous frames. A each time step t complete information about the previous frame is retained, namely the spatial segmentation, the image difference, and the calculated motion segmentation.

II. Calculate a measure of global spread of the motion response as the ratio of the number of motion pixels to the total image size. Also calculate a *derived* motion spread measure for the previous frame, calculated as the ratio of the sum of all previous-frame motion segment sizes to the total image size.

III. Overlap the current image difference with the current segmentation to identify the strength of motion response, or segment motion spread, within each segment. If the ratio of segment motion spread to segment size is greater than the global spread ratio (considered also as segment motion threshold), then mark the entire segment as exhibiting motion.

IV. To additionally stabilise the motion segmentation with regards to segment and motion memory of the previous frame, similarly to point 3, consider derived segment motion spreads by taking current spatial segments and the motion segments from the *previous* frame instead of current motion response. Since with previous-frame motion information we expect motion responses to be slightly displaced from segment locations, we shift the global spread ratio closer to 1 to ensure greater correctness of overlaps, as a new segment motion threshold, before checking whether the ratio of derived motion segment spread to segment size exceeds this threshold. If yes, these segments are marked as derived motion segments.

V. The union of regular and derived motion segments forms the final motion segmentation. In cases where the set of motion segments includes the entire image, either due to camera motion or several large objects in near-view moving simultaneously, we make the set of motion segments a null set and clear previous frame motion memory, to avoid motion segments being the entire image.

A formal description of the above summary is now presented. The processing starts with two sequential image frames I_{t-1} and I_t, their region-based spatial segmentations S_{t-1} and S_t, and their image difference $D_t(I_t, I_{t-1})$. All references to pixels p_i indicate the location of the i^{th} pixel as defined by $i = ((y-1) \times w) + x$, where x and y are cartesian coordinates and w is the width of the image.

A spatial segmentation $S(I_t)$ of image $I_t[p_1 : p_m]$, where m is the number of pixels in the image, is an n-dimensional vector $S(I_t)[s_1^t : s_n^t], s_i^t \in I$, where n is the number of segments discovered within the image at time t, and s_i^t is a set of pixel locations for spatial segment i.

The image difference D is taken to be a subset of image pixels as follows:

$$D_t(I_t, I_{t-1}) = p_i | p_i \in |I_t - I_{t-1}|, p_i > \phi \tag{3.6}$$

The differencing threshold ϕ can be set to small non-zero value and is useful since typically image differences are highly sensitive it may be desirable to ignore very small pixel differences possibly caused by illumination fluctuations or noise.

An indicator of global motion spread A_g for image I_t, is calculated as:

$$A_g(I_t) = \frac{|D_t|}{|I_t|} \tag{3.7}$$

For each $s_i \in I_t$, the segment motion spread $A_s(s_i^t)$ is:

$$A_s(s_i^t) = \frac{|s_i^t \cap D_t|}{|s_i^t|} \tag{3.8}$$

For each $s_i \in I_t$, regular segment motion flag $m_i^r(s_i^t)$ is established by considering segments crossing the threshold of the global motion spread:

$$m_i^r(s_i^t) = \begin{cases} 1 & \text{if } A_s(s_i^t) > A_g(I_t), \\ 0 & \text{if } A_s(s_i^t) \leq A_g(I_t). \end{cases} \tag{3.9}$$

The total regular motion segmentation set then is:

$$M_t^r = s_1^t \cup s_2^t ... \cup s_n^t, \text{ where } m_i^r(s_i^t) = 1 \tag{3.10}$$

For the previous frame I_{t-1}, the derived global motion spread $A_g^d(I_{t-1})$ considers previous frame regular motion segments M_{t-1}^r, instead of the raw motion response D_{t-1}, and is represented by:

$$A_g^d(I_{t-1}) = \frac{\left|M_{t-1}^r\right|}{|I_{t-1}|} \tag{3.11}$$

For each $s_i \in I_t$, the derived segment motion spread $A_s^d(s_i^t)$ also considers previous frame regular motion segments M_{t-1}^r and is given by:

$$A_s^d(s_i^t) = \frac{\left|s_i^t \cap M_{t-1}^r\right|}{|s_i^t|} \tag{3.12}$$

For each $s_i \in I_t$, the derived segment motion flag $m_i^d(s_i^t)$ is established by considering segments crossing a modified threshold obtained from the global motion spread:

$$m_i^d(s_i^t) = \begin{cases} 1 & \text{if } A_s^d(s_i^t) > \frac{A_g(I_t)+1}{2}, \\ 0 & \text{if } A_s^d(s_i^t) \leq \frac{A_g(I_t)+1}{2}. \end{cases} \tag{3.13}$$

Here the segment motion threshold is increased from $A_g(I_t)$ as in equation 3.9 to $\frac{A_g(I_t)+1}{2}$, since when combining previous frame motion pixels with current frame spatial segments there is a greater possiblity of uncorrelated motion and segment overlaps, therefore boosting the previously used segment motion threshold by half the distance to its maximum value helps improve robustness.

The final motion segmentation is then:

$$M_t^f = M_t^r(s_i^t) \cup M_t^d(s_i^t) \tag{3.14}$$

To avoid extended false positives based on motion memory, and also to avoid motion segments equalling the entire image itself, one last step remains:

$$M_t^f = D_{t-1} = \emptyset \text{ if } \left| M_t^f \right| = |I_t| \tag{3.15}$$

The complexity of the motion segmentation scheme is bounded by the efficiency of the spatial segmentation, the remaining computation being of low complexity. We obtain frame rates of 8Hz to 10Hz for 320×240 video images. Also in addition to computational efficiency, the method improves five out of the seven common motion segmentation problems identified earlier. The *shadows* problem is indirectly addressed by the fact that moving objects and their shadows are usually picked up as distinct spatial regions. Only the *camouflage* problem of the foreground and background being similarly coloured or textured, which the human visual system has difficulty solving as well, causes a deterioration of results.

The *generalised aperture* problem arises from a required balance between aperture size and multiple motion differentiation ability. This cannot be termed a trade-off since reducing aperture size only makes it more difficult to obtain an accurate motion segmentation lock in the first place. The proposed approach addresses this by using maximum aperture size at the outset and effectively sharpening focus on the basis of spatial segments instead.

The *waving trees* problem is symptomised by failure to distinguish between foreground and moving background. The presented method is capable of distinguishing between different objects moving independently due to its segment based motion tracking. This would of course fail if the segmentation confused the foreground and the background in the first place, but the selected segmentation mechanism is seen to be fairly robust. Thus the final motion segmentation would only merge foreground and background if *both* the following conditions were to hold true: 1. the foreground and background appear very similarly coloured and

textured, and 2. the foreground and background exhibit motion simultaneously. Also, it is necessary for the appearances or motion characteristics of any two regions to diverge only momentarily in order for us to be able to "remember" them and pick out matches based on previous frame motion memory if the appearances and motion do become indistinguishably similar for the subsequent frame.

The problem of *foreground aperture* arises from another trademark characteristic of image differencing, that is the subtraction process fails to find differences on most of the interior of a moving object, since between frames most locations on the interior are likely to be occupied by another location still lying within the interior, thus yielding very similar features, except for highly textured local areas. In other words, image differencing yields ghostly contours, with thicker contours for faster moving objects since for any particular object more of the interior has had the opportunity to be replaced by a previously exterior location. Also of importance is the fact that, again except for highly textured areas and a "wall" of hardware/illumination noise, only objects contours that are non-parallel to the motion will be detectable using image differencing. Moreover, the greater the angle of the contour to the direction of motion, the greater the response to differencing. These characteristics prove useful for potential future work in which a template based recognition system can "know" the object concerned and thus infer the properties of the motion based on how prominent known contours are in the differenced image, limited of course by object non-rigidity.

Our method addresses the problem of *foreground aperture* by using contours active in the image difference to identify which whole regions from the spatial segmentation correspond to motion, thus supplying the us with the "missing" motion interior for intra-object untextured motion unresponsiveness.

Next we consider the *sleeping person* problem. This is a result of absence of de-

tectable motion, which zeroes out the image differencing response, excluding mild pervasive noise. An immediate solution is to directly reuse motion response from an earlier frame instead of that from the current one, but this presents a problem. An object may move very slightly so as to either not exceed any noise threshold or such that the motion is beneath the sensitivity of the sensor hardware. In this case an object may creep gradually across an otherwise static scene and never be detected, since the current frame difference has a flat response and much older motion information may be continually replacing the current motion response in order to retain a lock on the target. There are two problems here. A false negative would involve losing the object when it stops moving or moves very slowly, while a false positive would involve assuming the existence of a previously detected segment when in reality the object of interest has by some means disappeared from the image location.

The presented method addresses both these problems by considering its memory of the previous frame region and motion information. The use of previous motion response pixels helps reduce false negatives by still picking up a motion object even when its motion response in the current frame has dropped, provided there exists a motion response in the previous frame. False positives are reduced due to the consideration of motion response within entire segments. While an object disappearing entirely will suddenly revealed background objects to suddenly jump to attention, unrelated segments will not be positively identified. That is, if an object disappears from in front of a smooth stretch of background wall, no part of the entire wall will appear as a motion object since the motion response is not spread over a large surface area of the wall. False positives can also come about in rare cases if the motion memory persistently causes the motion object to be retained due to actual motion in one frame and then noisy or random motion

response in subsequent frames, but in this case we are able to differentiate between regular motion segments and memory-derived motion segments.

Finally, the problem of *walking person* is dealt with as a direct result of the nature of the spatial segmentation mechanism. When a foreground object starts to move and reveals background differences as well, there is motion activity detected in both regions, the new location of the foreground and the newly revealed background. In most cases the strength of motion response in the revealed background will not be strong enough compared to the size of the background segment, and it will therefore not be identified as a motion segment. However, in the case that both foreground and revealed background are detected as motion objects, we know from the spatial segmentation that these are two distinct regions. From the results in figure 3.8 we see that none of the segmented objects have a 'motion trail', indicating that the revealed backgrounds from motion objects were successfully ignored.

We have described a scalar motion segmentation method that addresses several of the main problems in foreground segmentation and yields very good results at close range, provided the background is not extremely cluttered. The results shown in Figure 3.9 were seen to hold steady over lengthy extended video segments, losing the foreground target very occasionally. In the left image we see that the ears and spectacles existing as separate segments in the spatial segmentation have been combined into a whole segment with respect to their common motion. Similarly, in the right image the sets of segments representing the face, hair, arms and body have all been merged into a single region. While this method of motion segmentation is restricted to extracting only a single foreground motion object from the image, thus limiting its applicability, the quality and stability of the results at close range views potentially makes it worthwhile for webcam, green

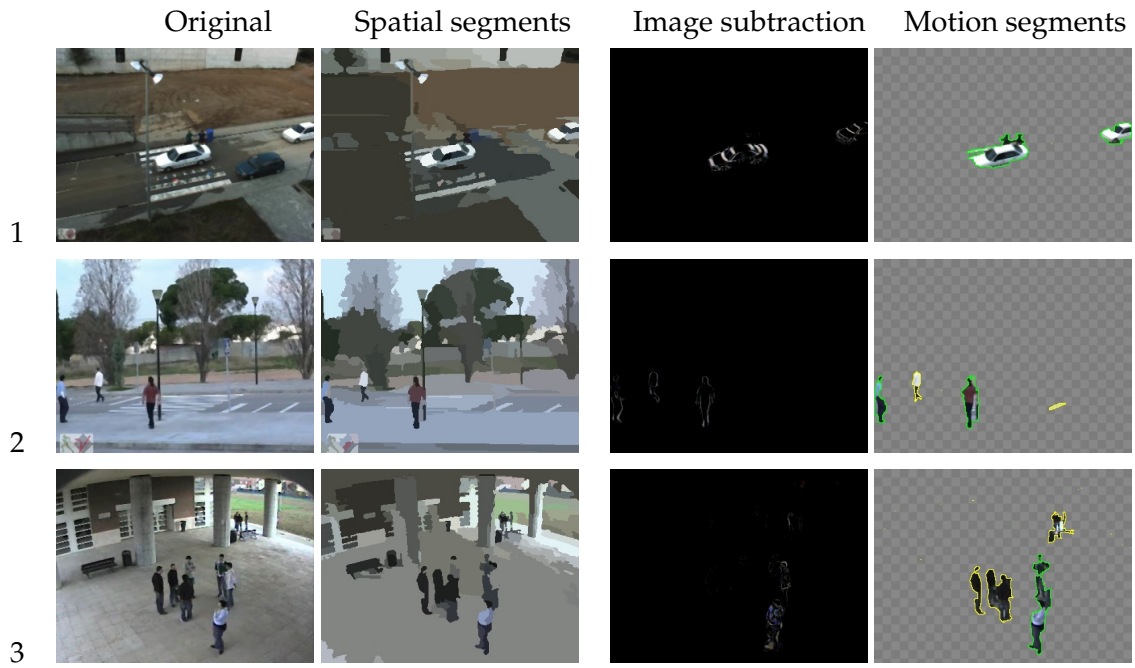

Figure 3.8: Motion segmentation from OpenVisor [194] sequences. 1: ISELab sequence Hermes_Outdoor_cam1, 2: ISELab sequence CVC_Zebra, 3: Outdoor Unimore D.I.I. sequence seq01_cam1_300305_A

screen or other types of applications which share a close range viewing distance and a relatively uncluttered background.

3.4.2 Vector motion from optical flow

While intensity differencing produces a dense map of pixel-level changes, this information is scalar, one does not know in which direction each pixel is likely to have moved. The solution is to use optical flow, which attempts to track points over frames. A popular and long used technique for estimating optical flow is the Lucas-Kanade method [22], from which using mean region optical flow information we apply our semi-greedy merging scheme to get a spatiotemporal seg-

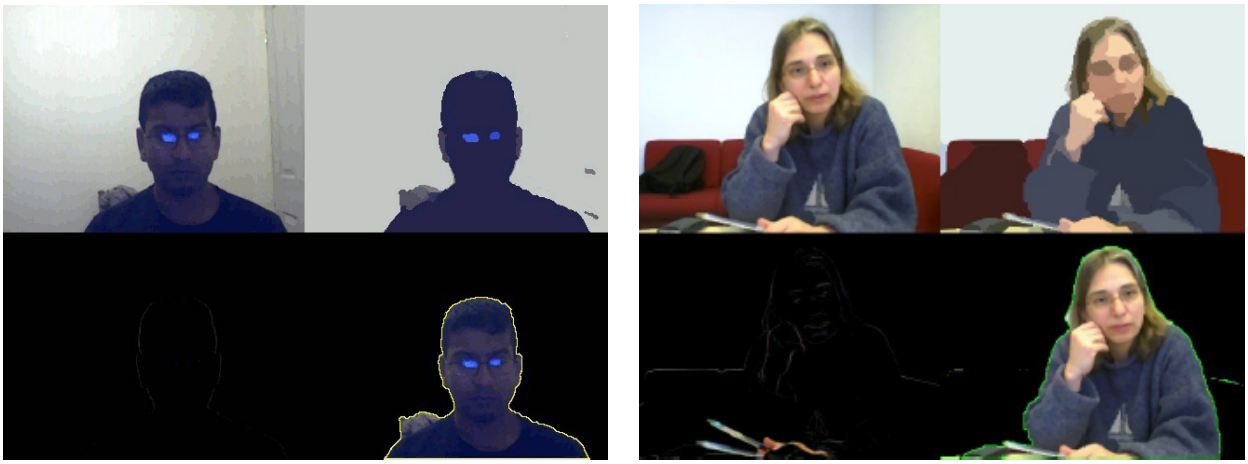

Figure 3.9: Scalar motion segmentation at webcam distance. Within each image, top left shows the original input, top right shows the spatial segmentation, bottom left shows scalar image differencing with respect to the previous frame, and bottom right shows the spatiotemporal motion segmentation.

ment map. Using this, we now propose a spatiotemporal segmentation method that is significant in the following ways: a) unrestricted by the number of motion segments, b) dense representation, c) no assumption about velocity of motion, d) low complexity, executes fast, e) fully unsupervised, f) fully online, no training required, g) works with a minimum of only two frames.

The proposed method, segmentation by dense region flow (SDRF), involves computing dense optical flow and combining this with region information from the spatial segmentation to obtain region flow vectors which are then used to merge regions that appear to exhibit similar motion. Quantitative and qualitative evaluation of segmentation and shape matching results are given and an overall evaluation measure, the relative performance (RP), proposed to summarise quantitative performance across multiple metrics.

The simplest indicator of motion is intensity differences between two frames

of a video sequence. While intensity differencing produces a dense map of pixel-level changes, this information is scalar, one does not know in which direction each pixel is likely to have moved. The solution is to use optical flow, which attempts to track points over frames. We use the Lucas-Kanade (LK) method [124] which is a Gauss-Newton gradient descent non-linear optimization algorithm and is known to produce an overall best performance [82]. While the LK method is described in more detail in [17], we paraphrase a brief summary here.

The goal of the LK image alignment method is to align a template image $T(\mathbf{x})$ to an input image $I(\mathbf{x})$, where $\mathbf{x} = (x, y)^T$ is a column vector containing the pixel coordinates. Let $\mathbf{W}(\mathbf{x}; \mathbf{p})$ denote the parameterized set of allowed warps, where $\mathbf{p} = (p1, ...pn)^T$ is a vector of parameters. The warp $\mathbf{W}(\mathbf{x}; \mathbf{p})$ takes the pixel $\mathbf{x}$ in the coordinate frame of the template T and maps it to the sub-pixel location $\mathbf{W}(\mathbf{x}; \mathbf{p})$ in the coordinate frame of the image I. For 2D optical flow, the warps represent the translations: $\mathbf{W}(\mathbf{x}; \mathbf{p}) = \left(\begin{smallmatrix} x+p1 \\ y+p2 \end{smallmatrix}\right)$.

The aim is to minimize the sum of squared error between two images, the template T and the image I warped back onto the coordinate frame of the template: $\sum_{\mathbf{x}}[I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - T(\mathbf{x})]^2$. Warping I back to compute $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$ requires interpolating the image I at the sub-pixel locations $\mathbf{W}(\mathbf{x}; \mathbf{p})$. The minimization of the expression $\sum_{\mathbf{x}}[I(\mathbf{W}(\mathbf{x}; \mathbf{p} + \mathbf{\Delta p})) - T(\mathbf{x})]^2$ is performed with respect to $\mathbf{p}$ and the sum is performed over all of the pixels $\mathbf{x}$ in the template image $T(\mathbf{x})$. The algorithm assumes that a current estimate of $\mathbf{p}$ is known and then iteratively solves for increments to the parameters $\mathbf{\Delta p}$ as $\mathbf{p} \leftarrow \mathbf{p} + \mathbf{\Delta p}$. These two steps are iterated until the estimates of the parameters $\mathbf{p}$ converge to the threshold $\|\mathbf{\Delta p}\| \leq \epsilon$.

We use the openCV[1] function $cvCalcOpticalFlowLK$ for LK flow estimation, forcing the inclusion of every pixel as a feature, to get a dense flow map consisting

[1]Open Source Computer Vision Library, http://www.intel.com/research/mrl/research/opencv

of flow vectors for each pixel in the first of a pair of frames.

After obtaining $\mathbf{\Delta p}$ for each pixel, we are able to proceed to calculating the *region flow* warp $\mathbf{V}(\mathbf{r}, \mathbf{\Delta p'}_r)$ for region $\mathbf{r}$ where $\mathbf{\Delta p'}_r$ is the mean flow vector for $\mathbf{r}$ calculated as the mean of all $\mathbf{\Delta p} \in \mathbf{r}$, as $\mathbf{\Delta p'}_r = \frac{1}{|r|} \sum [\mathbf{\Delta p} \in \mathbf{r}]$. The cost $CF_{i,j}$ of merging two regions S_i and S_j based on flow information is thus given by $CF_{i,j} = |\mathbf{\Delta p'}_i - \mathbf{\Delta p'}_j|$. The threshold d_{flow} restricts region merges based on optical flow information if the cost of merging is greater than or equal to this value. The rest of the variables are similar to those applied in the spatial region reduction step (Section 3.3.2).

I. Region reduction, $\forall t \in Ind$, 4-neighbour scan $\{TL, T, TR, L\}$:

(i) $hasMerged = false$.

(ii) $i =$ Label(t).

(iii) calculate $C_{u,v}$ as $CF_{i,j} | S_j \in B_i$.

(iv) if $C_{u,v} < d_{flow}$, Merge(S_u, S_v) as follows:

a) $k = k + 1$ and $m = m + 1$.

b) $P^k = (P^{k-1} \cup \{S_m\}) \cap \overline{\{S_u, S_v\}}$.

c) calculate D_m from D_u and D_v.

d) $B_m = (B_u \cup B_v) \cap \overline{\{S_u, S_v\}}$.

e) $\forall S_j \in B_m$, $B_j = (B_j \cup \{S_m\}) \cap \overline{\{S_u, S_v\}}$.

f) $hasMerged = true$.

(v) if $hasMerged = true$, repeat from step I.

This completes the spatiotemporal segmentation. We next test the SDRF spatiotemporal segmentation using the Berkeley Motion Segmentation Dataset (BMS-DS) [31], which provides 26 video sequences with dense segmentation annotations of moving objects in 204 of the frames across all sequences. We experimentally find good values of threshold d_{flow} to be in the range $[0.3, 2.0]$. From Figure

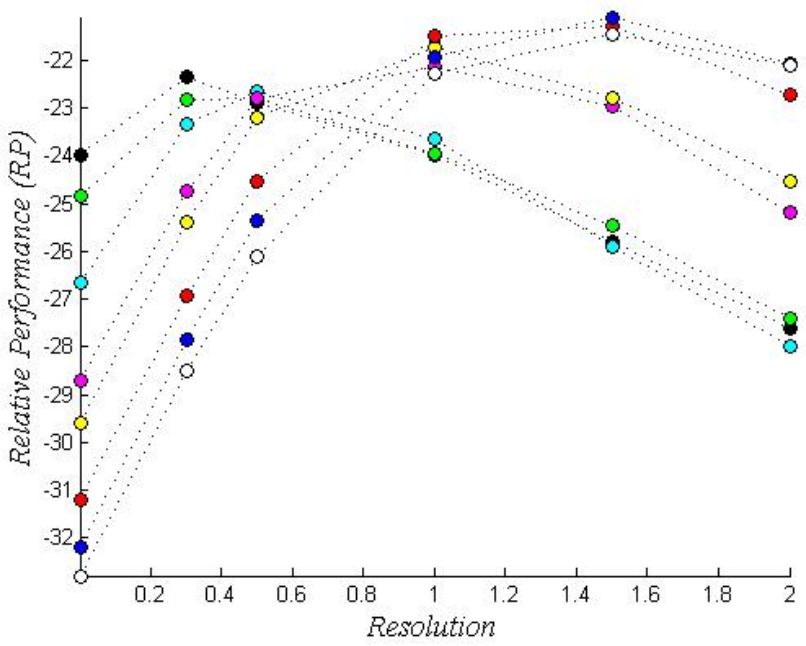

Figure 3.10: Relative Performance (RP) of SDRF segmentation on the Berkely Motion Segmentation Dataset (BMSDS) with various values for a) initial d_{curr}, b) d_{max}, and c) d_{flow}. Parameter ranges: initial $d_{curr} = \{0, 3, 5, 10, 15, 30, 50, 70\}$ respectively for white, blue, red, yellow, magenta, cyan, green and black plots; $d_{step} = 5$; $d_{max} = d_{curr} + 5$; region flow resolution $d_{flow} = \{0, 0.3, 0.5, 1.0, 1.5, 2.0\}$ on the x axis.

3.10 we see that the best spatiotemporal segmentation relative performance is obtained from the blue, red and white plots, which represent initial d_{curr} values of 3, 5, and 0 respectively. Within these plots the best relative performance is obtained from spatiotemporal region flow resolution d_{flow} values in the range $[1.0, 2.0]$. This shows that using region flow to group spatial regions using optical flow information works best with a high resolution spatial segmentation and moderate to high values for the region flow threshold d_{flow}. However, using lower spatial resolution

gives more visually pleasing results.

A comparison between the spatial and spatiotemporal segmentations is found on the $Resolution$ axis in Figure 3.10, where $Resolution = 0$ marks the point on each curve at which there is no spatiotemporal merging. We can see that, for every curve, this point is not the highest on the RP scale. Only for three curves, cyan, green and black, corresponding to high spatial merging thresholds, does any point on the curve dip below the zero spatiotemporal resolution performance. For the curves with the globally highest RP values, the white, blue and red curves, the best RP is well above the RP at $Resolution = 0$. This clearly indicates the superior performance of spatiotemporal merging, when temporal data is available, compared to only spatial merging.

Figure 3.11 shows a comparison between a low resolution spatial segmentation and its spatiotemporal counterpart. We can see that the spatiotemporal segmentation merges the spatial regions making up the rear wheel into one segment. Average spatiotemporal segmentation frame rates, in addition to spatial segmentation time, on the 640×480, 352×288 and 532×380 BMSDS frames were 5 to 10 fps.

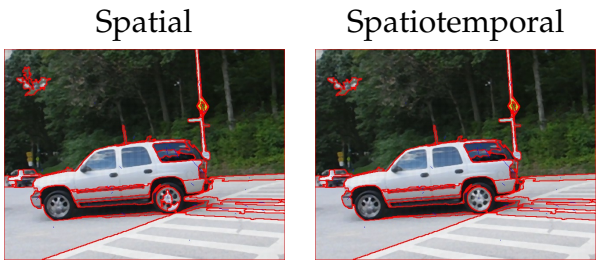

Figure 3.11: Spatial vs. spatiotemporal segmentations for BMSDS sequence $cars1$. a) SGAT, b) SDRF. Parameters: $d_{curr} = 70$, $d_{step} = 5$, $d_{max} = d_{curr} + d_{step}$. Note single segment rear wheel in b) as opposed to a).

3.5 Discussion

In this chapter we have described a set of grouping techniques that allow us to convert a sequence of input frames into a set of cohesive regions. We have identified median filtered XYZ images to be well suited to such grouping tasks. We have demonstrated the effectiveness and speed of a Seek/Refine region merging mechanism consisting of a pixel-based initial labelling followed by a region-based further grouping of regions. Many considerations discussed separately in the literature affect a segmentation outcome, such as initial seeding, scan order, grouping order, neighborhoods and adjacency, parallel or sequential growing, fixed and varying distance thresholds, optimality of local merges, and region reduction heuristics. We combine these into a single region merging algorithm, SGAT (semi-greedy adaptive-threshold) region growing, a crude approximation of which is the three stage Seek/Refine segmentation previously described. We assess the performance of the SGAT segmentation framework using four standard evaluation metrics, and additionally propose a summary indicator, the relative performance RP, which combines individual evaluation metrics into a single value that allows a quick comparison between different segmentation methods.

We have explored two further ways of grouping regions by their common fate using spatiotemporal or motion information. Based on the SGAT spatial segmentation we have considered motion-based region reduction using two types of spatiotemporal information, scalar information in the form of temporal intensity differences and vector information in the form of pixel-level optical flow vectors. We have then proposed a novel spatiotemporal segmentation by dense region flow method (SDRF) which uses dense pixel-level optical flow information and combines them using the basic SGAT mechanism to obtain overall region flow vectors

for each segment, which are then used for further region merging. We have qualitatively and quantitatively assessed the performance of both spatiotemporal region reduction mechanisms.

Chapter 4

Matching shape appearances

From segmented regions we move on to region representation. This chapter describes region contour modelling and the evaluation of contour-based shape descriptors. Contour modelling is done using the growing neural gas, a type of self-organising map, to which adaptations are made for improved execution speed and for network simplification into a double-linkage single-chain representation that facilitates shape analysis. While the growing neural gas has been extensively applied to shape modelling, to the best of our knowledge it has never explicitly been used for region similarity comparison via contour description and curvature analysis. A set of 30 descriptors, derived from the properties of regions and their GNG shape representations, are evaluated and two sets of descriptors are established using the methods of feature subset selection and variable ranking.

4.1 Introduction

After obtaining segmented regions using spatial or spatiotemporal segmenta-

tions, we need a way to represent, describe and compare their shapes and appearances. Shape analysis is a very difficult problem, motivating a large volume of research, many of the concepts of which are described in various review papers [130, 123, 192, 191, 210, 156, 104]. While various shape representations and descriptors have been proposed, not all of them fit the general concensus of what a good representation or descriptor should be. For instance, according to [130], "A 2-D shape descriptor should be insensitive to: Translation, Scale changes (uniform in both the X-coordinate and the Y-coordinate), Rotations", and although it is debatable whether it is best if descriptors are completely insensitive to such changes or whether they simply should be less sensitive, not all shape representation schemes satisfy this condition.

The following are some other general criteria for shape evaluation [104]:

- Scope: Applicability to all types of shapes
- Uniqueness: Similar shapes should have similar descriptions that are different from other types of shapes
- Stability: Minor changes in a shape should not affect its description much
- Sensitivity: Minor but salient changes in a shape *should* affect its description
- Efficiency: Descriptors should be computationally easy to calculate and compare
- Multi-scale support: It should be possible to use a description to analyse a shape at various levels of abstraction
- Local support: It should be possible to compute and effectively compare descriptions when the input is either coarse or fine grained.

From our experiments, described later, we see that the uniqueness and stability factors are very difficult to balance. Unique descriptors, ones that vary noticeably between different classes of shapes, also tend to vary within each class, appearing to be both sensitive and unstable.

4.2 Contour modelling

In thinking about shapes the first few properties that come to mind are shape contours as well as global properties such as size, colour, etc. Contours describe important variations in figure and also represent instructions for approximately reproducing figures. To discover a contour, sets of boundary pixels need to be assimilated algorithmically. An efficient way of doing this is through self-organised learning, for which the self-organising map (SOM) [108, 110, 109] is a well-suited tool. The SOM was later adapted to the neural gas (NG) [128] and subsequently to the growing neural gas (GNG) [81]. We apply the topology preserving GNG to the task of contour modelling and shape analysis.

There are very few cases of the application in shape representation of SOMs in general, such as [166, 171], and the GNG in particular, such as [69, 10, 214, 9, 8, 7]. To the best of our knowledge, the GNG has previously never been applied explicitly to the task of shape curvature analysis.

In order to proceed, we must first briefly define the GNG algorithm. The following are the steps of the original algorithm as described in [81]:

0. Start with two units a and b at random positions w_a and w_b in $\mathbf{R}^n$.

1. Generate an input signal ξ according to $p(\xi)$.

2. Find then nearest unit s_1 and the second-nearest unit s_2.

3. Increment the age of all edges emanating from s_1.

4. Add the squared distance between the input signal and the nearest unit in input space to a local counter variable: $\Delta error(s_1) = \|w_{s_1} - \xi\|^2$.

5. Move s_1 and its direct topological neighbours towards ξ by fractions ϵ_b and ϵ_n, respectively, of the total distance: $\Delta w_{s_1} = \epsilon_b(\xi - w_{s_1})$, and $\Delta w_n = \epsilon_n(\xi - w_n)$ for all direct neighbours n of s_1.

6. If s_1 and s_2 are connected by an edge, set the age of this edge to zero, else create it.

7. Remove edges with an age larger than a_{max}. If this results in points having no emanating edges, remove them as well.

8. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows:

   - Determine the unit q with the maximum accumulated error.
   - Insert a new unit r halfway between q and its neighbour f with the largest error variable: $w_r = 0.5(w_q + w_f)$.
   - Insert edges connecting the new unit r with units q and f, and remove the original edge between q and f.
   - Decrease the error variables of q and f by multiplying them with a constant α. Initialise the error variable of r with the new value of the error variable of q.

9. Decrease all error variables by multiplying them with a constant β.

10. If a stopping criterion is not yet fulfilled go to step 1.

4.2.1 Efficient contour representation with the growing neural gas

While the GNG is good at learning topologies through vector-quantised Hebbian learning-based induced Delaunay triangulation to achieve Voronoi tesselation, it is a neural network and, like all neural networks, tends to converge relatively slowly. While parameter selection can optimise learning times, setting it to adapt very rapidly can destroy the Delaunay triangulation completely. In the context of unsupervised scene analysis, where we are faced with modelling contours of hundreds of regions per frame within sequences which consist of thousands of frames, modelling speed is extremely critical. We therefore describe certain modifications to the original GNG algorithm that speed up its performance.

The first modification concerns the distance measure used. Every time a signal is generated the GNG algorithm must compute the Euclidean distance between the signal and all the nodes in order to the identify the two nodes nearest to the signal. We propose the use of the faster Manhattan distance [132] instead.

The second modification deals with the stopping criteria. In order to ensure a consistent number of nodes per local contour feature for every region, we express the stopping criteria in terms of node density. Node density, expressed as a fraction in the range $[0, 1]$, is defined as the ratio of nodes to pixels. For instance, a node density of 0.1 means the network will stop developing when, with N nodes, there is at least 1 node for every 10 pixels of the input space.

The third modification involves the number of starting nodes and their connectedness. The standard GNG starts off with two nodes connected by an edge and grows gradually towards the final number of nodes. In order to try and speed up network development we initialise with some fraction f, in the range $[0, 1]$, of the total number of nodes. The number of starting nodes is then $n_{start} = \lfloor f \times n_{stop} \rfloor$.

These are connected sequentially[1], that is, every $node_i$ is connected to $node_{i-1}$ for $2 < i < n_{start}$.

The following GNG parameter values were used in all experiments: $\lambda = 700$, $\epsilon_b = 0.05$, $\epsilon_n = 0.0006$, $a_{max} = 30$, $\alpha = 0.5$, and $\beta = 0.995$, with the maximum number of nodes N being selected as described above.

Changing the distance measure and the initial number of nodes and their connectedness can have a direct impact on the Delaunay triangulation, which satisfies the condition that no node is inside the circumcircle of a triangle formed by any other set of nodes that are immediately connected to each other, and can thus affect the topological correctness of the network. We test the potential loss in accuracy both by visual comparison with the original GNG, shown in Figure 4.1, as well as by quantitative measures, shown in Table 4.1. While we try to summarise the quantitative gains by the mean and overall gain figures, assessing the individual results shows marginally higher error rates, a nearly five-fold speedup, and nearly identical[2] network node distributions.

Two measures of topological correctness we use are the mean quantisation error (qe) and the topological error (te) [187], shown in Equations 4.1 and 4.2 respectively. There are N pixels, or data vectors $\vec{x}_i$, representing the input space in a GNG network. The nodes, or prototype units, form the output space. The best matching unit (BMU) $m_{\vec{x}_i}$ for each data vector is the data vector nearest it in Euclidean space, the first BMU. The second BMU is the data vector nearest the input vector after excluding the first BMU. For the calculation of topographic error, there is a function $u(\vec{x}_i)$ that is 1 if $\vec{x}_i$ data vector's first and second BMUs are adjacent and 0

[1]Full connectivity between all starting nodes is computationally expensive both in terms of initialisation as well as in the deleting of edges. We have experimentally found it to be slower than sequential connectivity.

[2]Some variations are due to the random input distribution used in every run of the GNG

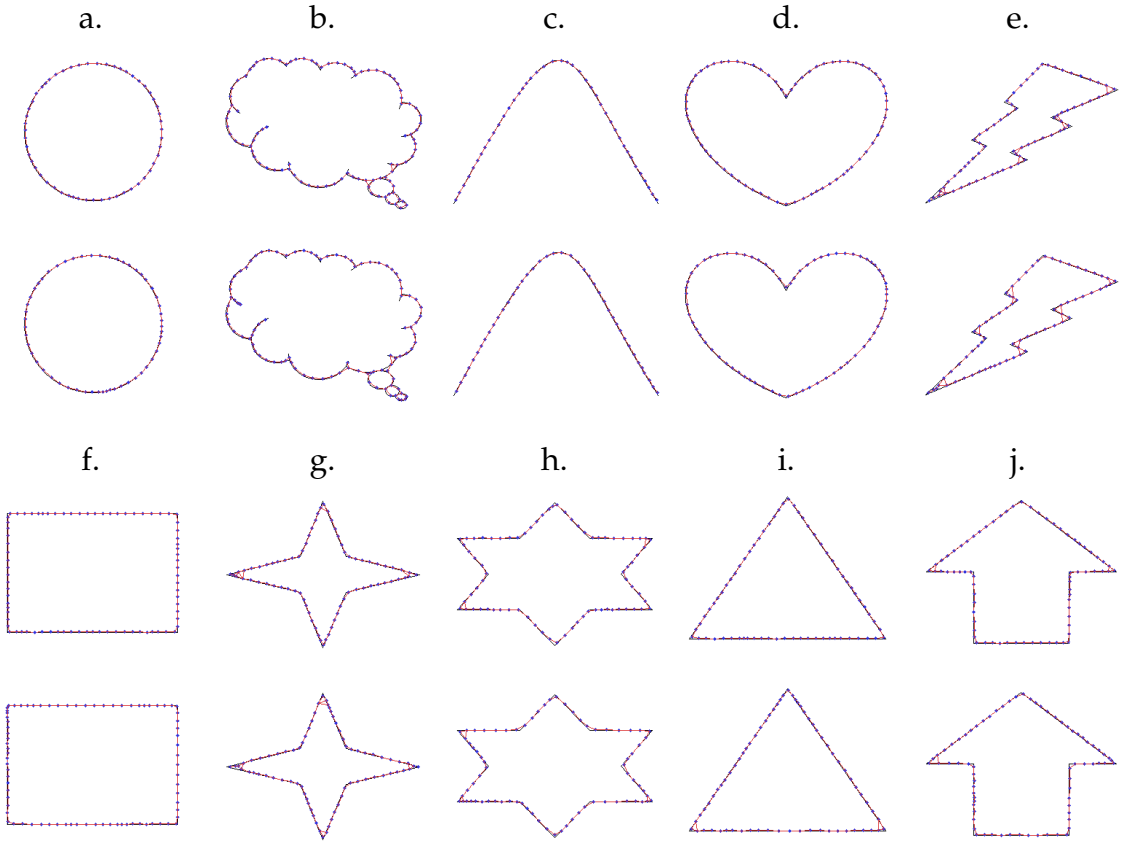

Figure 4.1: Comparison of GNG network formation between the original algorithm and our optimised implementation using 10 simple shapes. The optimisations produce a significant speedup and there is little difference in representation except on very close inspection.

		Original			**Optimised**		
Shape	Nodes	Fps	QE	TE	Fps	QE	TE
Circle	54	2.22	2.8016	0	7.94	2.8714	0
Cloud	97	0.61	2.6552	0	5.26	2.7275	0
Curve	42	4.20	3.0364	0	12.5	3.0789	0
Heart	70	1.38	2.9337	0	5.24	2.9648	0
Lightning	71	1.04	2.8138	0	6.99	2.9391	0
Rectangle	80	1.07	2.5235	0	6.25	2.5547	0
Star-4	71	1.16	2.6375	0	7.30	2.7551	0
Star-6	74	1.11	2.9073	0	6.06	2.9564	0.0014
Triangle	66	1.71	2.8816	0	7.14	2.9278	0
Arrow	72	1.11	2.7424	0.0014	6.67	2.7717	0.0014
Mean		**1.56**	**2.7933**	**0.0001**	**7.14**	**2.8547**	**0.0003**
Optimised/Original: $gain_{fps}$, $gain_{qe}$, $gain_{te}$					4.58	1.0220	3.0000
Overall gain: $gain_{fps}/(gain_{qe} \times gain_{te})$					1.49		

Table 4.1: Original vs. optimised GNG with respect to frames per second (fps), quantisation error (qe), and topographic error (te). Mean and overall gain shown as a summary statistic. The optimised version produces a significant speed increase, with little visual difference and a tolerable rise in error levels.

otherwise.

$$qe = \frac{1}{N} \sum \|\vec{x}_i - m_{\vec{x}_i}\| \tag{4.1}$$

$$te = \frac{1}{N} \sum_{i=1}^{N} u(\vec{x}_i) \tag{4.2}$$

4.2.2 Simplifying the network

It is useful to obtain a contour composed solely of sequentially linked nodes, so that angles of curvature can be analysed. In many cases, the GNG can be formed with more than two edges emanating from some nodes, such as in Figure 4.1.b, 4.1.e, and 4.1.g. This can happen when there are either sharp corners or complicated junctions in the silhouette, particularly when the network is not given enough time or sufficient number of nodes to model the contour more accurately. Even when the GNG is allowed a long time to converge, through a high value for λ and a high maximum number[1] of nodes N, there is no guarantee that multiple connections will be avoided, as this depends on the structure of the input space. As explained previously, in online systems performance is critical, and therefore there is greater reason to run the GNG with a short convergence cycle and fewer nodes. Thus we need a method to convert a complicated network into one comprised only of sequentially linked nodes.

The easiest procedure is to simply delete all nodes that have more than two edges, but doing this leads to a contour mapping that is full of gaps, the network being composed of a fragmented set of segments with many edges missing that

[1] A special case is when N equals the size of the input space, which means there is one node for every pixel of input space.

could otherwise have been preserved. The following is a straightforward algorithm we use for the network simplication task that preserves as many edges as it can while eliminating multiple connections as well as attempting to keep the network as a single connected segment:

1. Begin with all edges unmarked.
2. Identify nodes which have two or less edges emanating from them and mark their edges.
3. Delete all unmarked edges.
4. Delete all nodes which still have more than two edges.
5. Remove any 'hanging' edges, those without nodes at both ends.
6. Remove any 'orphan' nodes, those with no emanating edges.
7. For every node with a single edge, if its nearest other single-edged node is less than half the distance to its second-nearest other single-edged node, then connect the node to its nearest single-edged node. This step attempts to close some gaps left by edges previously deleted.

Figure 4.2 shows the results of this network simplification algorithm on the 'lightning' shape. We see that the network preserves the contour with only a little of the top right corner detail being lost. More importantly, all the multiple connections have been reduced to a sequence of double connected nodes. For the purpose of a fast topological representation, running the simplification algorithm is less expensive than either using more nodes in the network or allowing the network more time to converge.

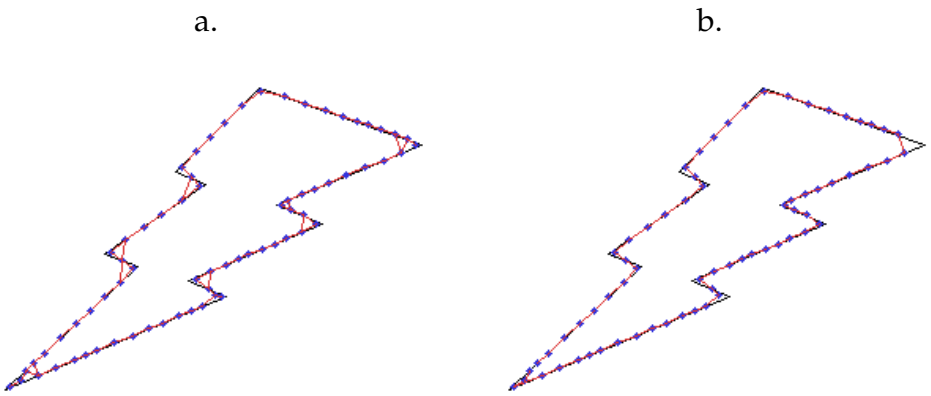

Figure 4.2: Simplification of the GNG network to eliminate multiple connections and to attempt to reduce the network to a single series of sequentially linked nodes: a) Original, b) Our simplification algorithm.

4.3 Curvature based contour features

Now that we have arrived at a fast and effective contour representation method, we next analyse shape-discriminative features that can be derived through contour analysis. Since the GNG network consists of nodes connected by edges, we can readily obtain angles of turn when going from node to node around a contour. Given a set of nodes and edges, we are able to approximately reconstruct the original contour, and therefore such a representation is information preserving and represents a set of instructions for the approximate reproduction of the input space. Also, it is generally the case that a GNG network consists of roughly even spacing between nodes, with only minor local variation. Thus by knowing a set of angles of turn, and assuming a fixed spacing between nodes, we should still be able to reconstruct a scale-varying but morphologically similar contour approximation. This guides the assumption that a set of curvature angles alone is sufficient to differentiate between a large variety of shapes.

We next test this hypothesis by calculating a large variety of statistics based on the angles of curvature for shapes from a standard dataset and by measuring intra-class variance for each statistic. A good shape-discriminative measure should demonstrate high inter-class distance but low intra-class variance, and so statistics for which intra-class variance is high can be discarded from consideration. We thus model shapes from the COIL-100 [137] dataset, for which object categories are known, using our optimised GNG and then apply many different statistical measures to the angles of curvature. The goal is to identify the set of statistics that display a low intra-class variance and, since each measure can be expect to discriminate between certain types of shapes but not all, a moderate to high inter-class distance.

The 100 object Columbia Object Image Library (COIL-100) dataset consists of colour images of 72 different poses for each object. The poses correspond to 5° rotation intervals and the objects are captured against a dark background, making the primary shape relatively easy to separate from the background without the need for a full segmentation. This makes the COIL-100 dataset convenient for shape analysis.

We perform a simple R, G, B threshold segmentation, with $R > 40$, $B > 40$ and $G > 40$, in order to separate the object from the background which is dark but contains varying shades. In a few cases this fixed thesholding of the background leads to the object shape being imperfect, either excluding a small portion of the object or including a small portion of the background. This variation is within acceptable limits and the imperfections actually help simulate the uncertainties present in a full segmentation mechanism. Figure 4.3 shows some of the typical shapes extracted in this way and their corresponding simplified GNG networks.

We also need to calculate the turning angles from node to node within a sim-

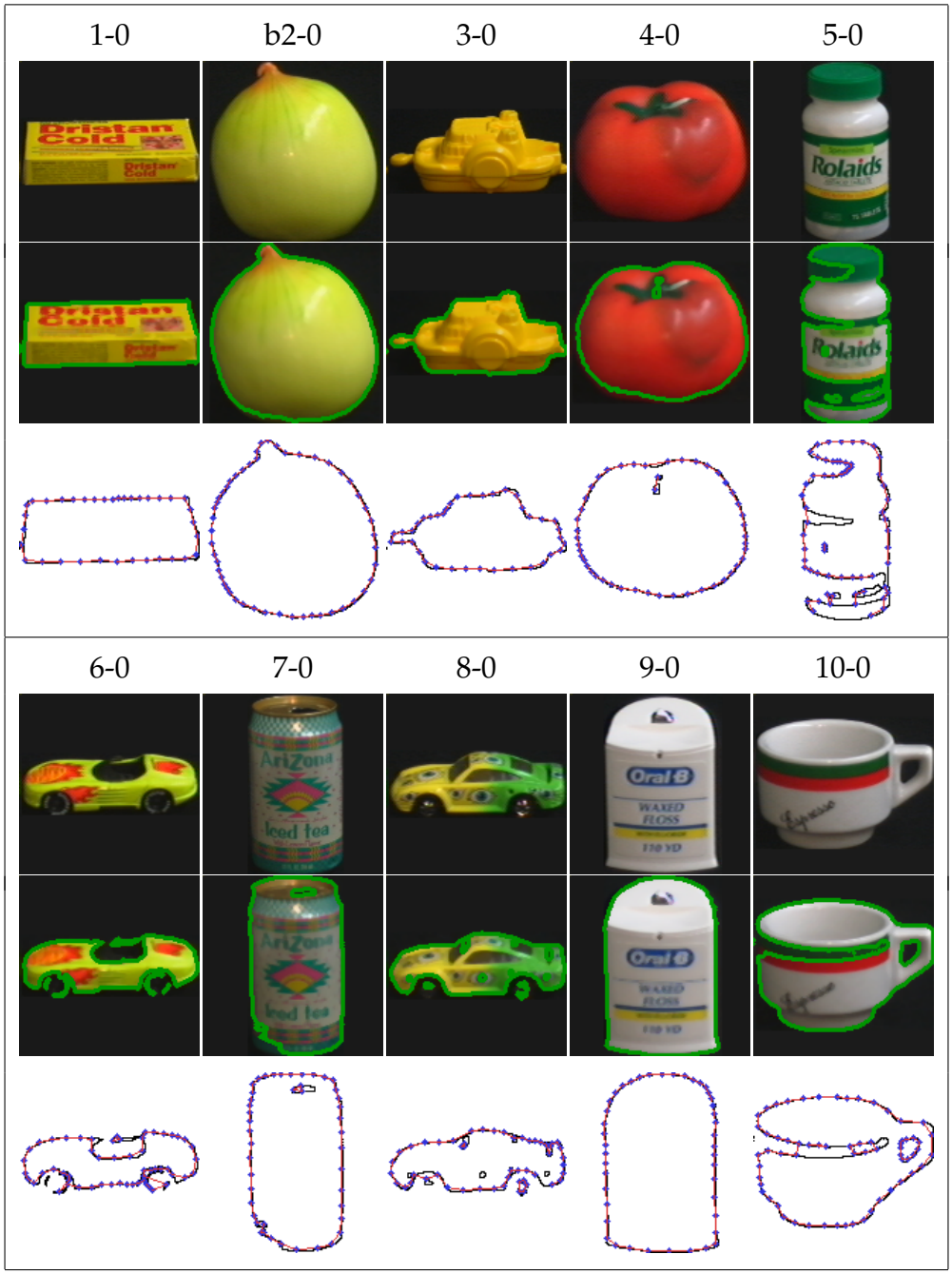

Figure 4.3: First shape of each of the first ten objects in COIL-100, showing the original image, the thresholded region, and the GNG contour representation.

plified GNG network representing a shape contour so that statistics based on the curvature angles may be calculated and used as shape descriptors. We now describe the turning angle used in our shape representation experiments, and the algorithm to calculate it for a set of sequentially connected contour nodes.

4.3.1 Calculating node-by-node turning angles

The exterior angle between any two sides of a polygon is the angle formed by the first side and a line extended from the second side. This involves three nodes at points A, B and C, our desired direction of travel being $\overrightarrow{AB}+\overrightarrow{BC}$, with the angle of turn being between $\overrightarrow{AB}$ and the the extended reversed length of the second vector $\overrightarrow{CB}$. This represents the vertically equivalent minimum turning angle to rotate the vector $\overrightarrow{AB}$ to point in the direction of $\overrightarrow{BC}$. Locally convex sections of contours have a positive turning angle while locally concave sections have a negative turning angle.

The exterior angle $\angle ABC$ defined between points $A(a_x,a_y)$, $B(b_x,b_y)$ and $C(c_x,c_y)$ is calculated according to the following:

I. Perform vector subtractions to get vector between points.

(i) $BA_x = b_x - a_x$ and $BA_y = b_y - a_y$

(ii) $CA_x = c_x - a_x$ and $CA_y = c_y - a_y$

II. $dot = (BA_x \times CA_x) + (BA_y \times CA_y)$

III. $pcross = (BA_x \times CA_y) - (BA_y \times CA_x)$

IV. Calculate angle in degrees: $angle = atan2(pcross, dot) \times \frac{180}{PI}$

V. If $angle < 0$, do $angle = -180 - angle$, else $angle = 180 - angle$

The following algorithm describes the calculation of all turning angles starting with a sequence of contour nodes connected by edges. We start with a set of nodes n_i defined by coordinates (x_i, y_i) and edges $E(n_s, n_t), s \neq t$, and aim to represent these in terms of a sequence of lengths l_j and turning angles θ_j. The distance between two nodes n_a and n_b is their euclidean distance $d(n_a, n_b)$.

I. If $\exists n_i \in S : S = \{E(n_i, n_y) \cup E(n_y, n_i)\}, |S| = 1$, then the contour network has hanging ends consisting of some nodes with only one emanating edge. We select one of these as the starting node. If not, then every node n_i has two neighbours, the contour network being closed, and we arbitrarily select some starting node n_i. Starting this way guarantees we will pass through all the nodes in the network by simply following the edge to the next neighbour for the current node.

II. Initialise new representation with $n_s = n_i$, $j = 0$ and previously considered node $p = -1$.

III. Set $\theta_j = 0$ and $l_j = d(n_i, n_y)$.

IV. Set $p = i$ and $i = y$.

V. Find new neighbour n_z of node n_i along edge $E(n_i, n_z)$ such that $z \neq p$. If $i = s$, stop.

VI. Increment j.

VII. Find the angle between nodes n_p, n_i and n_z according to the exterior angle calculation function described above in Section 5. Set $\theta_j = \Theta(n_p, n_i, n_z)$ and $l_j = d(n_i, n_z)$.

VIII. Set $y = z$ and unset z.

IX. Repeat from Step 4.

At the end of this procedure we get a set of $[l_j, \theta_j]$ that represent complete instructions for tracing a path all the way around a region contour. The first turning angle θ_0 is always set to 0 for a rotation invariant representation, that is in order to represent the region at its original pose we simply need to additionally store the angle that the first vector makes with repect to either axis. We also note that due to the Voronoi tesselation properties of the GNG, most of the l_j values are similar, thus putting more importance on the sequence of turning angles, rather than the length between turns, in the representation and analysis of region shape.

4.3.2 Shape features

We then determine an extensive set of statistics based on the curvature angles and calculate intra-class variance, using the known COIL-100 object categories, for each measure. To make the analysis of shape-discriminative statistics more complete, we also include six global descriptors (1-4) that are unrelated to curvature angles but are calculable from the GNG network. The measures we consider are the following:

1. Mean R, G, B: Mean region colour, the most basic visual descriptor.
2. Area: Region size in number of pixels.
3. Number of network nodes: An indicator of the amount of GNG resources required to model the shape.
4. Number of network labels: An indicator of network complexity. The occurence of several multiply connected nodes could result in a greater number of network segments after the simplification process.

5. Mean absolute angle: The overall turning tendency
6. Minimum absolute angle: The smallest turn in the network
7. Median absolute angle: Overall turning tendency, but more resilient to noise
8. Maximum absolute angle: The largest turn in the network
9. Variance, Standard deviation: Measures of angle spread.
10. Mean absolute deviation, median absolute deviation: Measures of spread of absolute angles.
11. Angle range, interquartile range: Total range and midspread.
12. Number of inflexion points: Points of inflexion are points at which the curvature changes sign, indicating 'S' shaped contour locations.
13. Mean absolute inflexion: Mean absolute change in curvature at the inflexion points, indicating the overall strength of the 'S' shaped contour features.
14. First and second eigenvalues: Eigenvectors and eigenvalues represent important characteristics of matrices, being associated with image moments, orientation and shape information, and it is possible that the eigenvalues of a set of contour points could yield useful information.
15. Skewness: Absolute angle distribution asymmetry.
16. Mean - median: Central drift, also an indicator of skewness.
17. Kurtosis: Measure of extreme deviation of absolute angles.
18. Circularity: Shape roundness as described by curvature angles.

19. Curvature acceleration: Mean rate of change of curvature angles.

20. Second central moment: Variance using divisor of n instead of $n - 1$.

21. Number of angles in the ranges $[135, 179]$, $[90, 134]$, $[45, 89]$ and $[0, 44]$: Chain code-like turn classification.

We compute descriptor values over all COIL-100 shapes and calculate the mean intraclass variance and the mean interclass distance. Every descriptor has its own range of values and for the purpose of comparison we first normalise all the data according to the following formula, thereby bringing every descriptor value within the range $[0, 1]$:

$$x_i = \frac{x_i - min(x)}{max(x) - min(x)} \tag{4.3}$$

where x_i is a descriptor value and $max(x)$ and $min(x)$ are calculated over all instances of that descriptor.

Intraclass variance (ICV) and interclass distances (ICD) are also calculated for three control variables which are the object ID ($group$), object instance ($instance$), and a random number ($rand$). The summary measure $\frac{ICV}{ICD}$ indicates the discriminative strength of the descriptor. Low values correspond to low intraclass variance and high interclass distance, characterising effective descriptors, specially since many of the thresholded shape segmentations are very noisy.

Table 4.2 shows all calculated quantities. As expected, the control variables $group$ and $instance$ produce zero and infinite values for $\frac{ICV}{ICD}$ respectively, since $group$ is the ground truth itself and $instance$ is unique for every object instance within a group. Also as expected, the control variable $rand$ shows the highest $\frac{ICV}{ICD}$ of all the other descriptors, indicating poor discriminative power.

#	Descriptor	ICV	ICD	$\frac{ICV}{ICD}$
1	Object $group^{++}$	0	33.6667	*0*
2	Object $instance^{++}$	438	0	*Inf*
3	Mean B	0.0012	0.2771	**0.0044**
4	Mean G	0.0021	0.2278	**0.0092**
5	Mean R	0.0011	0.1856	**0.0061**
6	Pixel $area$	0.0114	0.1823	**0.0625**
7	Number of $nodes$	0.0038	0.1120	**0.0338**
8	Number of $labels$	0.0053	0.0938	**0.0560**
9	First eigenvalue of nodes $eig1$	0.0136	0.1668	**0.0817**
10	Second eigenvalue of nodes $eig2$	0.0082	0.0763	0.1070
11	Angle var	0.0081	0.0894	**0.0907**
12	Angle $range$	0.0264	0.1028	0.2565
13	Angle iqr	0.0054	0.0790	**0.0686**
14	$Skew$ness	0.0215	0.1023	0.2104
15	$Kurt$osis	0.0256	0.1107	0.2309
16	$Mean$ absolute angle	0.0091	0.1202	**0.0753**
17	Min absolute angle	0.0063	0.0423	0.1484
18	Med absolute angle	0.0068	0.0887	**0.0766**
19	Max absolute angle	0.0537	0.0765	0.7027
20	$Mean - Med$	0.0122	0.0995	0.1225
21	Angle std	0.0118	0.1120	0.1053
22	Inflexion points $numInflex$	0.0094	0.1006	**0.0934**
23	Mean absolute inflexion $meanInflex$	0.0081	0.0862	**0.0943**
24	Number of $[135°, 179°]$ angles $num135$	0.0123	0.0283	0.4343
25	Number of $[90°, 134°]$ angles $num90$	0.0148	0.0742	0.1988
26	Number of $[45°, 89°]$ angles $num45$	0.0085	0.0920	**0.0922**
27	Number of $[0°, 44°]$ angles $num0$	0.0068	0.0780	**0.0876**
28	$Circ$ularity	0.0025	0.0693	**0.0359**
29	$Curv$ature	0.0068	0.1043	**0.0650**
30	Angle mad_0	0.0081	0.1203	**0.0674**
31	Angle mad_1	0.0067	0.0912	**0.0739**
32	Second central moment $mom2$	0.0135	0.0933	0.1447
33	Random number $rand^{++}$	0.0818	0.0354	*2.3084*

Table 4.2: Appearance and curvature descriptors and their ICV/ICD clustering strength indicators calculated over the COIL-100 dataset. 30 descriptors (3 to 32) and 3 control variables (1, 2 and 33) marked $^{++}$.

From the $\frac{ICV}{ICD}$ values we establish that all the descriptors in Table 4.2, excluding the three control variables, are at least partially capable of region contour discrimination. They all lie between 0, a perfect descriptor as indicated by the ground-truth variable $group$, and 2.3084, a non-descriptor indicated by the random control variable $rand$. Additionally, all the values occur very near the 0 (very discriminative) end of the range, with the few exceptions being max, $num135$, $range$, $skew$ and $kurt$. These exceptions, while still being much smaller than the control $rand$ value of 2.3084, are significantly larger than the other $\frac{ICV}{ICD}$ values.

At this point, we need to shortlist a set of descriptors that are likely to perform well when applied to a higher level computer vision task. There are two ways we may do this, feature selection and variable ranking. Feature subset selection ideally involves an exhaustive search through all possible descriptor combinations, based on which selection procedures such as minimum-redundancy-maximum-relevance (mRMR) [143] or correlation feature selection (CSF) [93] are run. We will prefer to follow a related but simpler approach due to the sheer number of subsets we can generate from our list of 30 descriptors. On the other hand, variable ranking uses a set of input and output variables and a scoring function to empirically determine an ordered ranking of decreasing variable usefulness. This method is simple and scaleable but can lead to the selection of a redundant subset [91]. We will experiment with feature selection in addition to variable ranking due to the following observations presented in the work in [91]:

- Noise reduction and consequently better class separation may be obtained by adding variables that are presumably redundant
- Very high variable correlation (or anti-correlation) does not mean absence of variable complementarity

- A variable that is completely useless by itself can provide a significant performance improvement when taken with others

As mentioned, the exhaustive feature subset search approaches in the literature (see [55] and [93] for reviews) very quickly become intractable as the number of features grows beyond a few variables. A subset selection procedure of greatly reduced complexity is to select descriptors that fall within an acceptable performance measure, for us ICV/ICD, range and then to refine these by eliminating those that are closely correlated with, but slightly weaker than, one or more other descriptors within the same set. The other approach, an extension of basic variable ranking, is to create an ordered list according to ascending ICV/ICD values and then to evaluate incrementally expanding sets starting with the best (at the top of the list) and working our way down. As we keep evaluating expanding sets, the point at which the evaluation measures indicate a peak in performance would indicate a good descriptor set. Since we rely on experimental results from a higher order vision task to perform feature set selection using variable ranking, we discuss this in further detail in Chapter 5, and establish a feature set here using the first method of feature subset selection.

Following the subset selection approach, we set our initial selection criterion as $\frac{ICV}{ICD} < 0.1$, based on the observation that intuitively good non-shape descriptors of B, G, R and $area$ all satisfy this. Applying this initial selection condition to the values in Table 4.2, we arrive at the following initial subset of 19 features: a) B, b) G, c) R, d) $area$, e) $nodes$, f) $labels$, g) $eig1$, h) var, i) iqr, j) $mean$, k) med, l) $numInflex$, m) $meanInflex$, n) $num45$, o) $num0$, p) $circ$, q) $curv$, r) mad_0, s) mad_1. Some of the descriptors in this list may be redundant if there is a strong correlation with one or more other descriptors. To reduce the effects of redundancy or feature interdependence, we refine this list by simplifying strongly correlated groups of

descriptors through a process of elimination of the weakest.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
a	-																		
b		-																	
c			-																
d				-			x												
e					-	x										x			
f					x	-										x			
g				x			-								x				
h								-		x			x				x	x	
i									-	x	x						x	x	x
j								x	x	-	x		x				x	x	x
k									x	x	-						x	x	x
l												-							
m								x		x			-				x	x	
n														-			x	x	
o							x								-				
p					x	x										-			
q								x	x	x	x		x	x			-	x	x
r								x	x	x	x		x	x			x	-	x
s									x	x	x						x	x	-

Table 4.3: Correlations for region descriptors that satisfy $\frac{ICV}{ICD} < 0.1$. a) B, b) G, c) R, d) $area$, e) $nodes$, f) $labels$, g) $eig1$, h) var, i) iqr, j) $mean$, k) med, l) $numInflex$, m) $meanInflex$, n) $num45$, o) $num0$, p) $circ$, q) $curv$, r) mad_0, s) mad_1.

For this purpose we analyse pairwise linear correlations and flag correlations for which $corr^2 > 0.5$ (a standard statistical test for strong correlation), shown in Table 4.3[1], and select only one descriptor from each correlated group, the one with the strongest discriminative ability as decided by the clustering performance measure $\frac{ICV}{ICD}$. For instance, $area$ and $eig1$ are correlated, as are $nodes$, $labels$ and $circ$.

[1]See Table A.1 in Appendix A for the complete set of appearance descriptor correlations.

We select $area$ from the first group and $nodes$ from the second, both of which have lower $\frac{ICV}{ICD}$ than the others which we discard in their respective groups. Through this subset selection mechanism[1] we arrive at the following feature vector that contains low redundancy:

$$F_{corr} = \{B, G, R, area, nodes, numInflex, num0, curve\}$$

4.4 Summary

In this chapter we have applied a self organising neural network, the growing neural gas, to shape contour modelling. We first propose a modification to the GNG that allows faster modelling. We test to see if this optimisation causes any noticeable loss in topology preservation of the map by calculating and comparing the frame rate, quantisation error and topological error for ten different artificially generated shapes. The optimised version is seen to produce a near fivefold speed increase with negligible change in the quantisation error and a threefold increase in the topological error. The overall speed increase to accuracy loss ratio is approximately 1.5. Visual inspection of the generated contour maps using the optimised GNG also show negligible differences. Since speed is critical in real-time segmentation and region analysis, this gain of 1.5 justifies using the optimised version over the original.

We then propose another modification to the GNG that simplifies the contour to consist of only sequentially connected nodes. This facilitates extracting angles of curvature, or turning angles, from lengthwise contour segments. We deal with the problem of simplification where multiple connections exist between nodes, when

[1] See Appendix B for an algorithmic representation of the correlation-based subset selection procedure.

this task is more difficult since one must then decide which edge to follow if faced with several emanating edges. The simplification step consists of a combination of marking single connected nodes to preserve them, deleting nodes with multiple edges, and rejoining nearby hanging nodes.

Next, the shape discriminative strength of various turning angle statistical measures are evaluated. Optimised GNG maps with the contour simplification prcedure are used to model region shapes from the COIL-100 dataset. Based on the region contour representations, a set of 30 appearance and shape descriptors are established. The ground truth is used to calculate intraclass variance and interclass distances for these descriptors and the $\frac{ICV}{ICD}$ ratio is taken to indicate the shape discriminative strength of each. Two feature selection approaches are considered, subset selection and variable ranking. The first is used to draw up a descriptor shortlist which will be tested in the following chapter against a different descriptor set established using the second approach via experimental evaluation. Following the subset selection approach, the initial selection criterion is set as a low value for the ratio ICV/ICD, after which this initial list is refined by considering pairwise feature correlations and setting up groups of correlated descriptors. Where high correlation indicates feature interdependence within a group, the feature with the lowest $\frac{ICV}{ICD}$ value is retained and the others discarded.

The following chapter combines the techniques discussed so far in this work for the purpose of automatic shape categorisation and scene analysis.

Chapter 5

Visual understanding via region appearances

In this chapter we describe the application of a nearest-centroid based eager learning method to automatically categorise observed shapes from image sequences or video. Using this categorisation technique and the COIL-100 dataset, the classification performance of the initial list of shape descriptors previously identified are compared against other descriptor sets established experimentally according to the second feature selection approach defined in Chapter 4. The evaluation measures that were used to quantify spatial segmentation performance are again applied to compare the class labels obtained by the proposed categorisation technique to the ground truth. The methods previously described in this work are also integrated to allow us to perform (a) localised region tracking using segmentation and scalar motion information, and (b) region class behaviour analysis through the categorisation of visual regions and with the aid of three types of trajectory plots and seven motion descriptors.

5.1 Introduction

Various types of visual abstractions can be drawn from image sequences or videos, such as object tracking, learning object classes and summarising video content. Our starting point is always a spatial/spatiotemporal segmentation of the input frames, followed by appearance analyses of the segmented regions. In real world situations, a spatial segmentation can represent complex objects as multiple regions, which we can group using similar motion features. Chapter 3.4 discusses two types of motion information, scalar and vector, that we can use. In simpler environments the use of scalar motion segmentation produces stable results at high speeds, but is unstable when analysing complex motion. Vector-based spatiotemporal region formation may work better in these cases. Also, the required level of region analysis is task dependent. Non-shape based region descriptors are more appropriate to localised object tracking through sequential frames that have a separation between motion components, while shape based region abstraction is more likely to contribute to the learning of the environment through the categorisation of observed visual regions.

An important factor in visual abstraction is the learning mechanism used to build a set of categories. This chapter begins with a non-shape based region tracking application and, to allow for more advanced shape based visual understanding, goes on to describe a fast eager learning method using region class centroids and shape-centric region descriptors, with which region classes are learnt from the COIL-100 dataset. The effectiveness of the class learning method and and the performance of sets of shape descriptors are evaluated through the classification accuracy achieved. Finally, shape-based region class trajectory analysis is applied to demonstrate the potential for a higher level scene understanding framework.

5.2 Non-shape appearance based object tracking

In this section we demonstrate a form of scene understanding using only spatial segmentation, scalar motion-based region reduction, and a primitive set of region descriptors. A region growing segmentation (Section 3.3.4) is first performed on each frame of the video and the number of regions reduced by merging regions further according to scalar motion information (Section 3.4.1). The regions are then tracked as follows:

I. Initialise tracker variable $c = 0$ and frame position $t = 0$.

II. Perform spatial segmentation to get a set R_t of regions.

III. Perform region reduction using scalar motion information to get a reduced set of regions R'_t.

IV. Assign new tracking label for each region, or transfer over the existing label for regions detected to have existed in the previous frame $t - 1$.

(i) If $t = 0$, for every region $r \in R'_0$ assign tracking value c, incrementing c after each assignment. Thus c is incremented $|R'_0|$ times.

(ii) If $t > 0$, $\forall r_t \in R'_t$, $\forall r_{t-1} \in R'_{t-1}$: calculate the tracking distance using standard Manhattan (L1) distances as $d(r_t, r_{t-1}) = ||y(r_t) - y(r_{t-1})||$, where y is a region descriptor feature vector composed of the x coordinate, y coordinate, size, and $\{B, G, R\}$ colour channels, defined by $y = \{X, Y, SIZE, B, G, R\}$. All six components of the feature vector are scaled to the interval $[0, 255]$ in order to give them equal weight in the Manhattan distance calculation. The tracking is controlled by a threshold d_{track}, experimentally set to a value of 60. If $\exists r_t, r_{t-1} : d(r_t, r_{t-1}) <$

d_{track} assign tracking label for r_{t-1} to r_t, else assign assign tracking label c to r_t and increment c.

V. Increment t and repeat from step 2 until the end of the video sequence is reached.

At the end of this procedure, all regions with the same tracking label correspond to the same tracked object. Some results are summarised in figure 5.1. Three frames from the Video Surveillance Online Repository (VISOR) [194] HighwayII video of cars moving down a highway are shown, below which are some of the regions or 'objects' segmented and tracked over several frames.

Figure 5.1: Local tracking of regions using non-shape descriptors. Top: Three example frames from the VISOR HighwayII sequence. Bottom: Some objects segmented and tracked via region similarity comparison.

In this experiment we assume a continuous existence of an object across several frames and also relatively small object translations from frame to frame. In this type of sequence all the moving objects of interest, the cars, are rectangular blobs, however because of the stated assumptions we are able to perform tracking without considering region shapes. In principle, not accounting for errors, this approach allows one to both track and count the number of different occurences of a certain type of object within a video sequence. From Figure 5.1 we can see that a number of different classes of vehicles are separately tracked across several frames, even though the video is noisy and of low resolution. Additionally, near real-time performance is achieved, an approximate average frame rate of 5 being achieved for the entire segmentation and tracking procedure.

5.3 Bootstrapping categories

Image sequence abstraction and summarisation tasks require the explicit learning of object categories. The nearest-centroid learning method is an eager learning approach, in which observations are abstracted and stored while the details are immediately discarded. This approach results in low storage requirements and rapid recall rates, while more effort is spent in the learning of classes. Shape descriptors using GNG contour representation and curvature analysis, as described in the previous chapter, are calculated and are used to learn shape categories.

5.3.1 Centroid classification

The centroid classification method [142, 92, 120] uses cluster means to determine the class of a new observed sample. The first observation sets up a single cluster center which is the sample point itself. If the next observation lies outside a given

threshold distance from the existing cluster center or centroid, a new centroid is set up and the observation allocated to it. If not, the point is assigned to the existing cluster and the cluster mean updated. All subsequent observations x_i are tested for proximity to any of the existing centroids C_j and where the proximity function is satisfied the cluster mean μ_{C_j} is updated to include the new sample point.

A cluster center C_j is defined as the arithmetic mean of its sample point members:

$$\mu_{C_j} = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i \tag{5.1}$$

Cluster membership for a new observation is decided by a distance function $\mathrm{d}(x, y)$:

$$C(x) = \underset{C_j}{\operatorname{argmin}}\, \mathrm{d}\big(\mu_{C_j}, x\big) \tag{5.2}$$

We use a modified version of the Manhattan (L1) distance for the distance function:

$$\mathrm{d}(\mu, x) = ||\mu - x||_{AND} \tag{5.3}$$

where,

$$||y||_{AND} = \begin{cases} \sum_1^N |y(i)| & \forall i : y_i < d_t \\ \infty & \exists i : y_i \geq d_t \end{cases} \tag{5.4}$$

$y(i)$ being the value of the i^{th} feature, and d_t being a fixed threshold.

This distance measure is more effective for classifying high dimensional feature vectors since each the distance between each dimension must be within the

defined threshold otherwise the distance is set to infinity. In contrast to using vector means, this distance calculation ensures any differences are distributed in small amounts across all features instead of being concentrated in large amounts in a few channels.

The number of cluster centers grows as new instances are observed and similar shapes are grouped together. Every new instance must be compared against all the existing cluster centers in order to decide membership. This comparison can involve many expensive logical operations since we use a logical AND connecting individual feature distances. To reduce this, we can expand the distance calculation to first check the absolute difference between the mean feature value of the new instance and the cluster center being compared against. If the test of similarity of means fails then we no longer require additional checks for individual feature distances. This helps prune the set of distance calculations involved in classifying any new instance, being particularly useful when the number of existing cluster centers has already grown large.

The difference of feature means between an observed instance and a cluster mean is:

$$\mathrm{d}_{mean}(\mu, x) = \frac{1}{N}\left|\sum_{1}^{N}(\mu_i - x_i)\right| \tag{5.5}$$

The extended distance measure, replacing Equation 5.3, is then calculated as:

$$\mathrm{d}(\mu, x) = \begin{cases} ||\mu - x||_{AND} & \mathrm{d}_{mean}(\mu, x) < d_t \\ \infty & \mathrm{d}_{mean}(\mu, x) \geq d_t \end{cases} \tag{5.6}$$

The choice of d_t affects the number of clusters we obtain, with higher values producing fewer clusters with greater intraclass variance and lower values producing a larger number of more homogeneous clusters. We apply this learning

mechanism with various values for d_t to learn object types from the COIL-100 dataset in an unsupervised fashion.

5.3.2 Feature sets and classification performance

In order to facilitate quantitative evaluation of the categorisation results we treat the set of labelled instances as a segmentation of the data space, thus allowing us to evaluate performance using the evaluation measures described in Chapter 3.3.3. For each d_t, we quantify the performance of the labelling correctness using measures used for the spatial segmentation evaluation, namely PRI, VOI, GCE, BDE, and additionally the proposed summary indicator RP. As described in Chapter 3.3.3, these measures assess the extent to which a certain labelling conforms to the ground truth, and are easily adapted for this evaluation task. We use the term RP in this chapter to indicate $RP_{PRI,VOI,GCE,BDE}$.

We first apply the proposed online categorisation technique to COIL-100 shapes using the descriptor set F_{corr} initially shortlisted in Chapter 4. The results are presented numerically in Table 5.1 and summarised graphically in Figure 5.3. We see that overall accuracy $RP = -3.4492$ peaks at $d_t = 0.05$ for which the number of clusters $N_c = 3668$. From the ground truth we know that there are 100 object types, and N_c should therefore ideally be 100. However, when we force $N_c \approx 100$, such as at $d_t = 0.26$ where $N_c = 102$, then overall accuracy drops to $RP = -22.3632$. The implication of this is that the learning mechanism combined with the shortlisted shape descriptors is able to group together small sets of similar shapes, but that the error increases significantly when a larger merging threshold forces bigger clusters to be formed.

Compression ratio, C, is a measure of the reduction in data representation size and can be used to compare the grouping power of classification mechanisms.

d_t	N_c (≈ 100)	PRI ($\uparrow$)	VOI ($\downarrow$)	GCE ($\downarrow$)	BDE ($\downarrow$)	$RP_{PRI,VOI,GCE,BDE}$ ($\uparrow$)
0	7200	0.99014	6.1699	0	0.00013889	-4.3335
0.005	7169	0.99014	6.1606	0	6.94E-005	-4.3186
0.01	7039	0.99015	6.1187	0	6.94E-005	-4.2516
0.02	6465	0.9902	5.9059	0.00013889	1.39E-004	-3.9147
0.03	5718	0.99027	5.6395	0.0051731	6.94E-005	-3.6005
0.04	4993	0.99036	5.3934	0.019978	6.94E-005	-3.5348
0.05	**3668**	0.99058	4.8831	0.052965	1.39E-004	**-3.4492**
0.06	3047	0.99066	4.714	0.096301	0	-4.1377
0.07	2362	0.99075	4.5199	0.15474	6.94E-005	-5.1203

Table 5.1: Performance evaluation of region descriptor set F_{corr}: $\{B$, G, R, $area$, $nodes$, $numInflex$, $num0$, $curve\}$. Clustering threshold d_t vs. number of clusters N_c and evaluation measures PRI, VOI, GCE, BDE and RP. Desired values in parentheses.

Compression ratio is given by $C = \frac{\text{Compressed size}}{\text{Uncompressed size}}$. The data compression ratio obtained using the descriptor set F_{corr} with the optimal d_t indicated by best RP is $C \approx 0.51$.

We now proceed, due to insufficient clustering strength using F_{corr}, to test the categorisation performance of different descriptor sets following the variable ranking approach previously discussed. Rearranging the descriptors in Table 4.2 according to the scoring function as ICV/ICD, where lower values are better, we get the following ordering of the best seven[1] along with their respective ICV/ICD values:

$\{B, R, G, numNodes, circ, numLabels, curv\}$

$\{0.0044, 0.0061, 0.0092, 0.0338, 0.0359, 0.0560, 0.0650\}$

[1]We will not require any more than this, since we will locate the optimal set within the first seven descriptors.

Starting with a set containing only the best descriptor and incrementally adding subsequent ones, we obtain evaluation measures for a range of d_t values for each, keeping tracking of each maximum RP value to see for which descriptor set it is the best. The expanding ranked descriptor sets start with the set 1, $\{B\}$, and incrementally add to it the descriptors R, G, $numNodes$, $circ$, $numLabels$ and $curv$, to get seven different sets 1 through 7, each of which is one element bigger than the previous and the final set has all top seven descriptors. While our experiments have involved the full range of expanding descriptor sets and a larger range of d_t values, for the sake of concisenes we only show the RP results for sets 3 to 7 since this range clearly shows at which set the optimal RP value is reached.

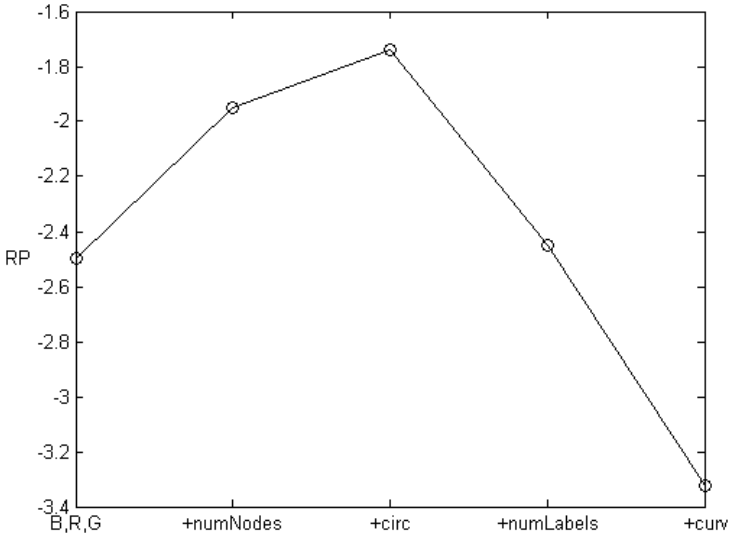

Figure 5.2: Performance summary indicator RP vs. ranked expanding descriptor sets.

According to the data we thus obtain[1], we summarise in Figure 5.2 the effec-

[1]See Appendix B for the complete set of evaluation statistics for expanding descriptor sets using variable ranking.

d_t	N_c (≈ 100)	PRI ($\uparrow$)	VOI ($\downarrow$)	GCE ($\downarrow$)	BDE ($\downarrow$)	$RP_{PRI,VOI,GCE,BDE}$ ($\uparrow$)
0	7200	0.99014	6.1699	0	0.00013889	-4.3335
0.005	5783	0.99028	5.6536	0.00034722	6.94E-005	-3.5163
0.01	3405	0.99075	4.6062	0.012469	2.08E-004	-2.1107
0.02	1661	0.99153	3.6118	0.067515	6.94E-005	**-1.7386**
0.03	1003	0.9918	3.2483	0.13583	1.39E-004	-2.669
0.04	638	0.99186	3.0686	0.22047	1.39E-004	-4.2542
0.05	440	0.99115	3.0555	0.30196	0	-6.0377
0.06	328	0.98988	3.2813	0.39515	1.39E-004	-8.4626
0.07	246	0.98859	3.3213	0.46168	0.00013889	-10.0011

Table 5.2: Performance evaluation of region descriptor set F_{rank}: $\{B,\ R,\ G,\ numNodes,\ circ\}$. Clustering threshold d_t vs. number of clusters N_c and evaluation measures PRI, VOI, GCE and BDE. Desired values in parentheses.

t on RP of gradually expanding the ranked descriptor set. When using set 5, $\{B, R, G, numNodes, circ\}$, RP reaches a clear maximum of -1.7386, for which $d_t = 0.02$ and the number of classes $N_c = 1661$, as Figure 5.2 shows. At this d_t, we obtain a data compression ratio of $C \approx 0.23$, a large increase in compression compared to the feature set arrived at through subset selection. After this point, adding more descriptors to the set results in a steady decrease in RP. Thus, feature selection by variable ranking gives us the following optimal set of descriptors:

$$F_{rank} = \{B, R, G, numNodes, circ\}$$

It is interesting to find that the most useful descriptors thus established consist of three intuitively good non-shape variables R, B and G, a property of the GNG contour network $numNodes$ that implies both the size and irregularity of a shape (complex shapes with more extending "arms" requiring a larger number of nodes

to represent than another simpler shape of exactly the same visual surface area), and only a single shape descriptor $circ$.

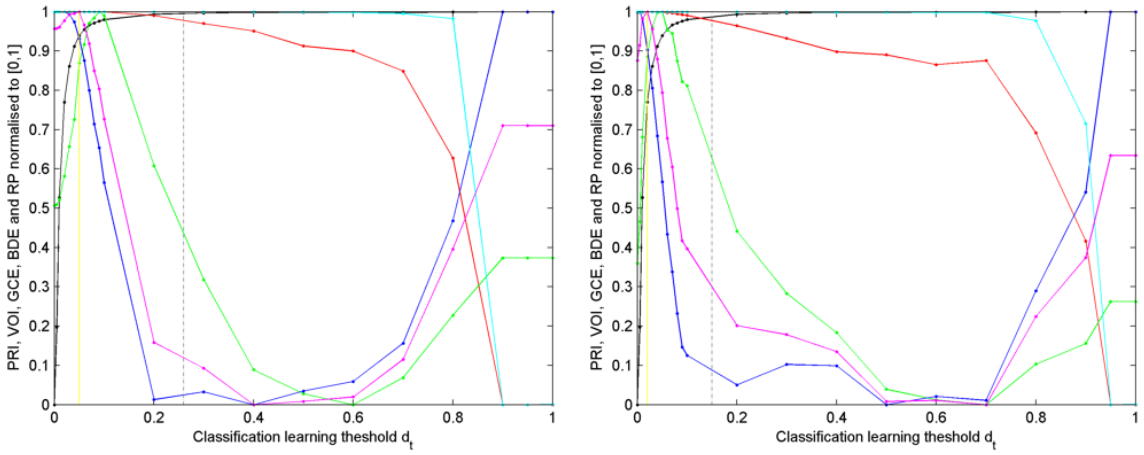

Figure 5.3: Performance characteristics curves for descriptor sets F_{corr} (left) and F_{rank} (right). Threshold d_t vs. evaluation measures normalised to $[0, 1]$ to allow simultaneous visual comparison. Black: N_c. Red: PRI. Green: VOI. Blue: GCE. Cyan: BDE. Magenta: RP. The yellow vertical line marks the d_t value that corresponds to best RP where the magenta curve peaks, while the dashed grey line marks the d_t value at which $N_c \approx 100$, the ideal number of clusters.

Figure 5.3 normalises the values for number of class centroids N_c and the evaluation measures, PRI, VOI, GCE, BDE and RP, according to Equation 4.3, for both descriptor sets F_{corr} and F_{rank} in order to superimpose them on a single plot for comparison. This comparison plot includes a full range of d_t values in order to show the characteristic response of each curve through the complete range of clustering resolutions. Also, since we favour lower values for VOI, GCE, BDE and N_c, we have inverted their plots using the transformation $x_i = max(x) - x_i$ before the values are normalised to $[0, 1]$. Visually, therefore, higher points along

the curves indicate better performance. The distance between the yellow vertical, line marking the point of best performance, and the dashed grey line, marking the point at which the ideal number of clusters is obtained, is smaller for descriptor set F_{rank} than for set F_{corr}. In terms of performance evaluation metrics, F_{rank} outperforms F_{corr}. However, we will see if these findings are supported when both are applied to the higher level task of scene component modelling, which we present in the next section.

We also note that the measure RP depends on a baseline score across the evaluation metrics considered, and since there are no baseline figures for such a classification task, we use the image segmentation baseline values. A segmentation task is clearly very different from an object classification task, both in terms of ambiguity and complexity, nonetheless by using its baseline figures we are able to calculate the performance summary indicator RP. The effect of adopting the segmentation baseline values for the classification task is that, while we are unable to compare classification accuracy against human performance for the same task, it still allows us to compare visual region classification performance with different threshold d_t settings and using different descriptor sets.

5.4 Region-based visual understanding

We now analyse region occurrences and region class behaviours over time for more sophisticated visual understanding. While the localised appearance-based region tracking demonstrated in Section 5.2 followed each region between consecutive frames as single trajectories, in this section we follow the behaviour of different region classes, as determined by shape categorisation, over arbitrary frames within a video segment. The region behaviour analysis starts with a segmentation of

regions from video sequences, with each grouped region then being modelled by our modified growing neural gas and the shapes for all frames being categorised according to the online centroid based learning described earlier. Region classes need not occur in sequential frames in order to be followed, since shape categorisation occurs over the content of all frames in a sequence. Instances belonging to the same region class are connected together, in order of temporal occurrence, on three different graphical plots: (i) spatial presence, and temporal presence consisting of a (ii) horizontal variation (x plot), and (iii) vertical variation (y plot).

The spatial presence graph plots the mean x versus the mean y coordinate of each class member, indicating the locations of instantiation, or existence, of different region classes. The horizontal and vertical variation graphs plot time versus region instance x position and time versus region instance y position respectively, showing the nature of horizontal and vertical class behaviours over time. A set of visual regions or instances belonging to a particular region class are represented on the graphs as dots, and these dots are connected by lines in order of temporal occurence. Instances in the same class occuring at the same time step are connected arbitarily.

We intuitively expect individual trajectories for videos in which primarily horizontal activity occurs to (i) be broader on the spatial presence plot, (ii) be taller on the horizontal variation plot, and (iii) be flatter on the vertical variation plot. Conversely, we expect trajectories for primarily vertical activity sequences to be (i) taller on both the spatial presence and vertical variation plots and (ii) flatter on the horizontal variation plot. Additionally, the density of points on each graph indicates the level of *busyness* that a scene contains.

We expect the set of semantics we may conclude from region class trajectory plots to include the following:

I. Frequency of occurrence: A lower density of dots on a given trajectory at earlier time steps connected to a higher density of dots at later time steps conceptually indicates a more steady later reocurrence of a type of visual object, whether due to a genuine steadiness of existence or due to segmentation or classification error[1].

II. Divergence and convergence: A single dot at a certain time step connected to multiple dots at another fixed time step would indicate the divergence, splitting, or multiplication, of instances from a region class, while multiple siimultaneous dots joined to fewer later dots would indicate a convergence, merging or elimination of instances from a region class.

III. Direction of motion: Steeper trajectories on the x plot indicate a primarily horizontal activity, while steeper trajectories on the y plot indicate a primarily vertical activity.

IV. Harmonic motion: A regular pattern exhibited by a trajectory in either the horizontal or vertical plots would indicate a repeated characteristic behaviour of a region class, this behaviour having its own amplitude and wavelength.

V. Steady component: Plots with the most dense horizontal trajectories at a fixed point on both the x and y plots indicate the existence of a set of steady background or foreground objects, dependent on whether the video is static and contains foreground movement, or is moving along with a foreground that is fixed relative to the camera. Steady components on the spatial presence plot are indicated by connected dots concentrated around small localised

[1] In general, the more dots there are on a given line, the more certain we may be of the validity of the information the trajectory represents, since occasional system errors are likely to randomise a pattern rather than strengthen it.

regions.

VI. Relative location of activity: Regions of varying density of dots at different x and y coordinates indicate the localisation of motion activity relative to the viewing window. We can get a visual summary of this from the spatial presence plot as the degree and type of spread of the surface area covered by different connected sets of components.

VII. Relative length of activity: The length on the $time$ axis of connected components in the x and y plots shows the temporal existence of different region classes.

While these are very simple concepts that humans process with barely any noticeable conscious thought, it would be a firm step in generalised scene understanding if we were able to establish a procedure to recognise them from within arbitrary video sequences. In the analysis that follows we refer to both weak and strong examples of conceptual abstraction of semantic video content from various video sequences from the UCF50 [153] standard motion activity dataset. Due to the complexity of the various algorithmic components interacting to provide the final result, we were unable to automate the procedure to obtain graphical results for all the videos contained in that dataset, however we have selected from this dataset a diverse set of examples, containing a mixture of noisy and clear videos containing either random or clearly defined motion, and representing semantically different types of activities. We make the assumption that higher densities of points represent stronger data than sparsely filled areas of points since outlier errors are more likely to randomise a pattern rather than emphasise it.

Figure 5.4 compares the spatial presence trajectories resulting through processing spatiotemporal (left) and spatial (right) regions respectively. We present this

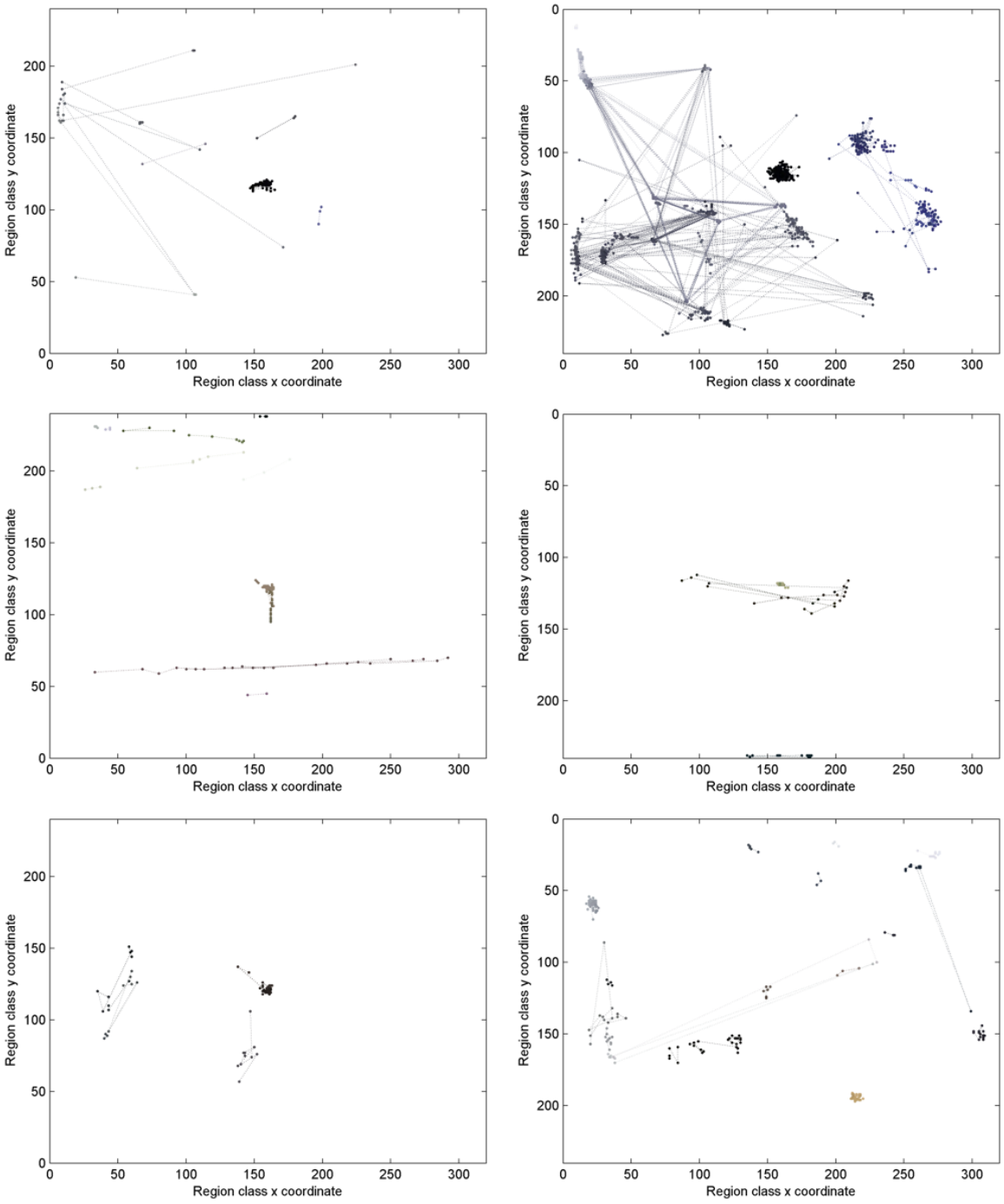

Figure 5.4: Region class x vs. y trajectories. Left: Spatiotemporal analysis. Right: Spatial analysis. UCF50 sequences: v_Drumming_g13_c01 (top), v_HorseRace_g01_c01 (middle), and v_BenchPress_g01_c01 (bottom).

to show that the region reduction effect of spatiotemporal grouping greatly simplifies the region class trajectory plots to the point at which only the most salient moving regions are extracted. We compare only the spatial presence plots because their complexity are also representative of the level of activity present in the other two types of trajectory information. With respect to the low resolution dataset we use here, the spatiotemporal grouping reflects the filtering out motion components that are less certain, and combined with the stronger grouping tendency of the descriptor set F_{rank}, this leads to very simplistic trajectory information. The result is fewer trajectory components than would allow us to draw up generalisations regarding the characteristics of such plots. Bearing in mind the poor resolution and high levels of noise, spurious visual effects and artefacts in these video sequences, we observe that the spatiotemporal analysis would be the prefered application method on visual systems of an acceptable standard of hardware quality and once we have developed sufficiently robust rules of trajectory characterisation. At this stage we are yet unaware of such rules, and to the best of our knowledge no such work is shown in the literature. We must uncover these rules through the analysis of the spatial trajectories which are dense and therefore allow us to detect patterns more easily. The primary target in this chapter is to uncover rudimentary rules governing the characteristics of region class trajectories, to form the foundation of trajectory analysis allowing automation of scene modelling and comparison in the future. All subsequent figures therefore refer to trajectory analyses based on spatial grouping.

To facilitate visual comparison in each case we present the spatial presence information superimposed over a mean of all frames in each sequence. The mean image provides an indication of the overall extent and range of motion activity within the videos, to which the spatial presence information can then be related.

In each mean image, blurred local regions indicate areas of most motion activity, to which corresponding sets of dots (region instances) should appear, their connecting lines implying region instance translocation or multiple occurrences of a region class. The horizontal and vertical trajectory variation information relate to physical space only in one dimension each, and we are unable to superpose such data on the original frames. We interpret the data for these two plots as indicating the extent and direction of translation in one dimension over time (as the video progresses).

Figure 5.5 and 5.6 show the trajectory plots for two UCF50 drumming videos. A summary inspection shows some immediately observable common characteristics. The spatial presence plots for both contain sharp angular (triangular or polygonal) connections between dense sets of instances, which are steady components, focused around small localised regions. The steady components are also seen in the x and y plots as narrow bands of instance occurrences.

The trajectories for both drumming sequences show neither a preference to horizontal or vertical activity, which coincides with the nature of drumming as an activity. Both sequences also show dense intricate trajectories, signifying a complex motion pattern (although sequence $v\_Drumming\_g11\_c01$ is simpler than $v\_Drumming\_g13\_c01$), and the sharp angles between connected trajectory components coincide with rapid changes in motion direction.

Figures 5.7 and 5.8 show the trajectories for two UCF50 pull-up videos. Sequence $v\_Pullup\_g10\_c01$ shows interesting trajectory patterns, partially attributable to the video being relatively clear and free of camera shake. On the spatial presence plot we see clear instance occurrences at $x \approx 140$ marking the motion of the lower body of the subject (dark coloured dots) and a part of the upper body (pink dots). Also, the central steady components on the x plot show two undulations of

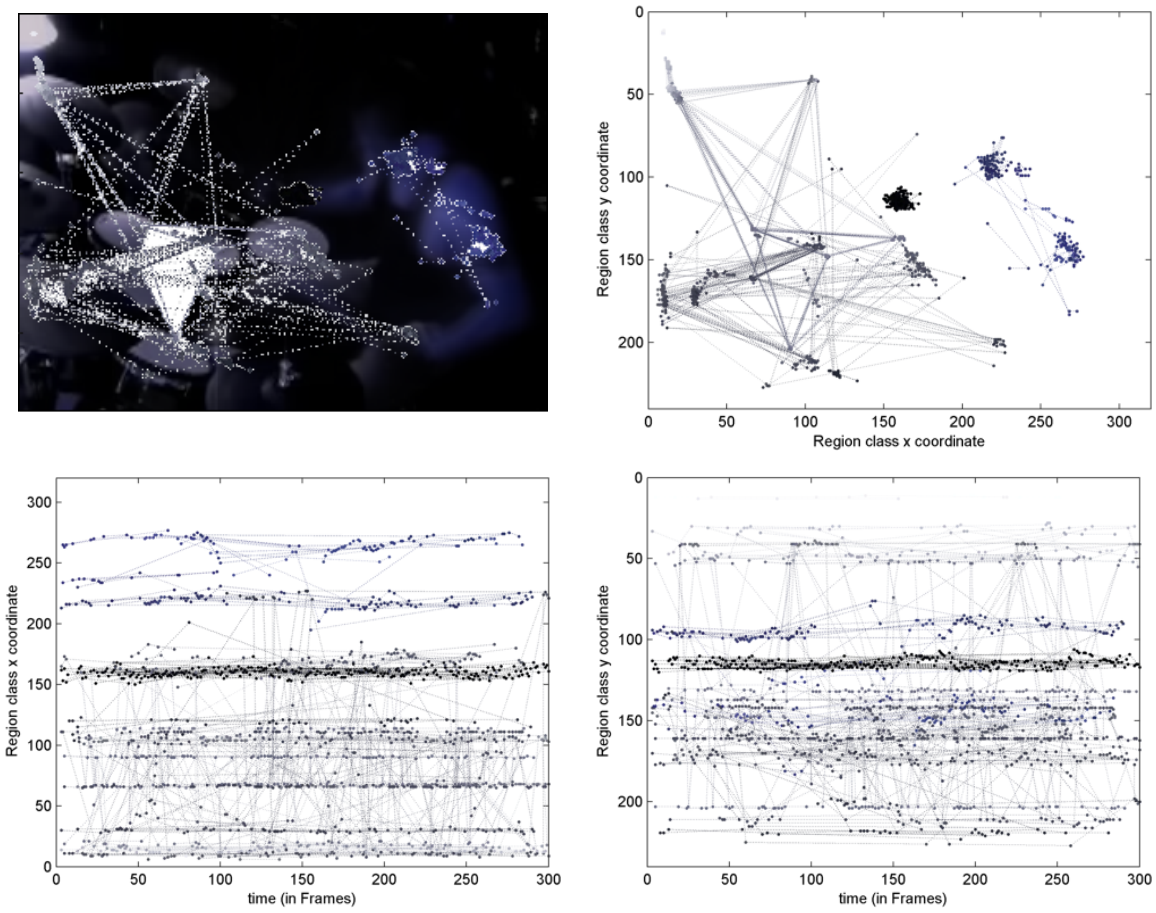

Figure 5.5: UCF50 video sequence v_Drumming_g13_c01. Top left: Mean of video frames with spatial trajectories superimposed. Top right: Spatial trajectories, x vs. y. Bottom left: x trajectories vs. time. Bottom right: y trajectories vs. time.

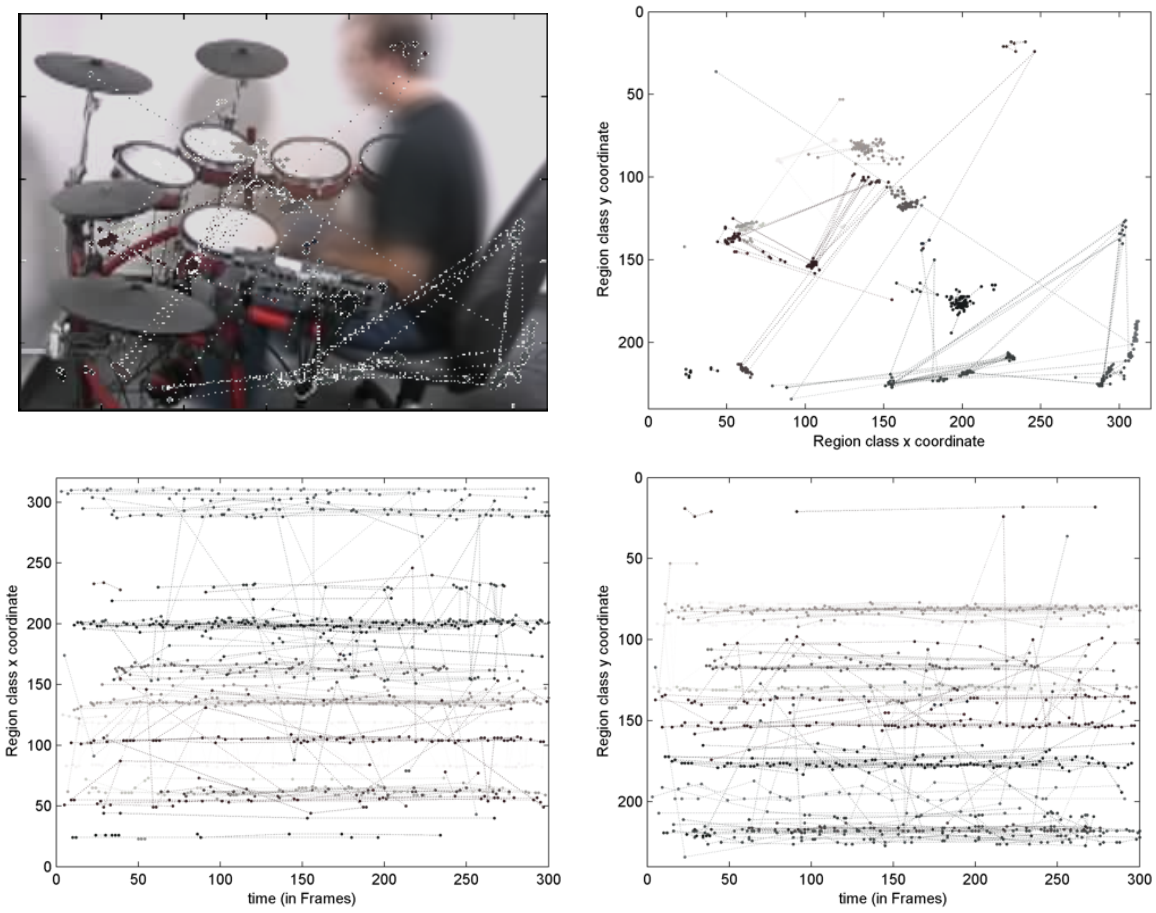

Figure 5.6: UCF50 video sequence v_Drumming_g11_c01. Top left: Mean of video frames with spatial trajectories superimposed. Top right: Spatial trajectories, x vs. y. Bottom left: x trajectories vs. time. Bottom right: y trajectories vs. time.

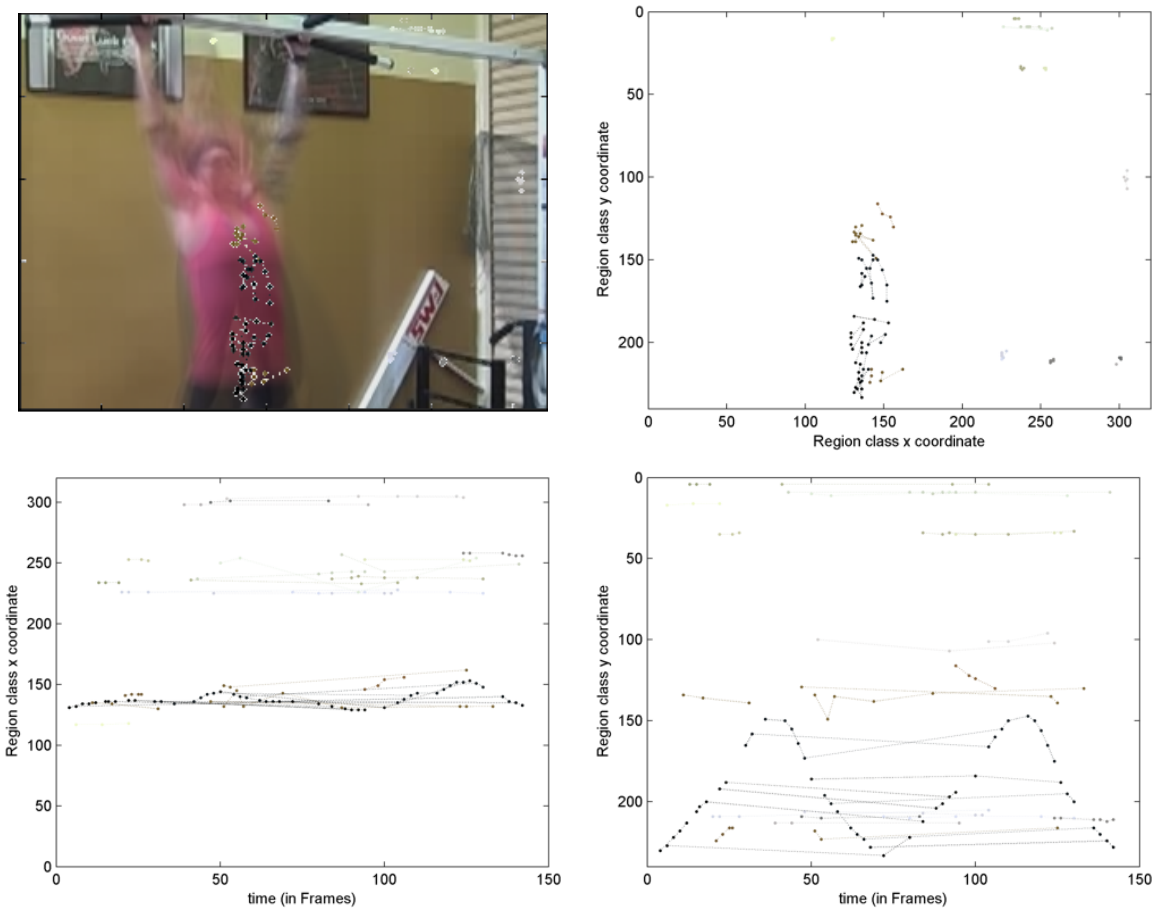

Figure 5.7: UCF50 video sequence v_Pullup_g10_c01. Top left: Mean of video frames with spatial trajectories superimposed. Top right: Spatial trajectories, x vs. y. Bottom left: x trajectories vs. time. Bottom right: y trajectories vs. time.

increasing intensity at $time \approx 50$ and $time \approx 130$, corresponding to the subject's legs swinging forth on the first pull-up and swinging forward again with greater intensity on the second pull-up, where greater physical effort presumably necessitates a larger swing. Similarly, on the y plot, the steady components resembling a sine wave clearly mark out both pull-ups as the subject goes up and down twice.

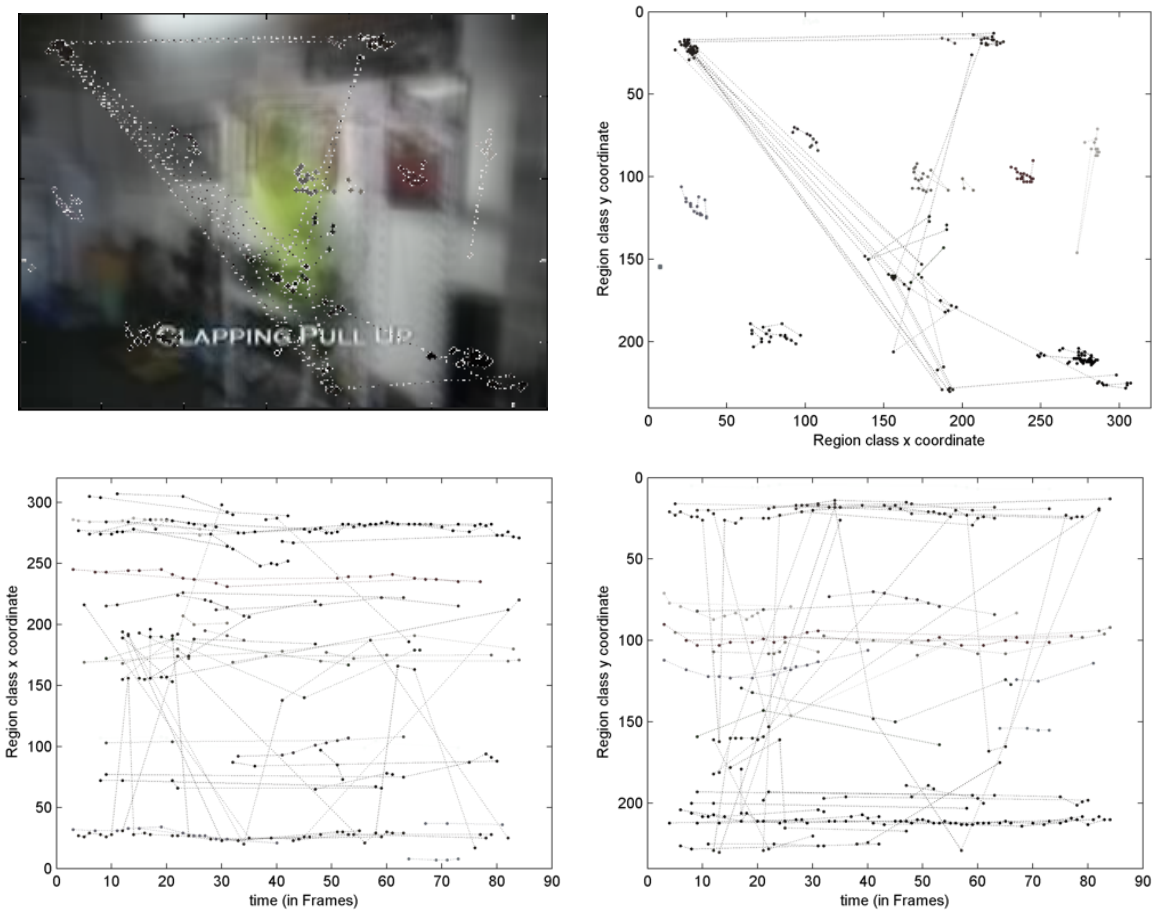

Figure 5.8: UCF50 video sequence v_Pullup_g06_c01. Top left: Mean of video frames with spatial trajectories superimposed. Top right: Spatial trajectories, x vs. y. Bottom left: x trajectories vs. time. Bottom right: y trajectories vs. time.

Sequence $v\_Pullup\_g06\_c01$ shows mostly noise, caused by very rapid motion and by intense camera shake. We observe that when there is significant camera motion then the system locks on to textured background objects since relative to the camera the background is indeed moving. In this sequence, dark objects of similar appearance on the wall form the majority of the steady components and perceived motion regions. This is also supported by the series of nearly vertical lines on the x and y plots, representing not genuine activity, due to the limitations of the system preventing it from detecting any real very fast motion, but the multiple occurrence of region class instances at physically separated but static locations. While its spatial presence plot shows an angular set of component connections very different from the first pull-up sequence and slightly resembling the drumming videos, it can be differentiated from the drumming sequences by the lack of dense central steady components.

Figures 5.9 and 5.10 show the trajectories for two UCF50 horse race videos. Sequence $v\_HorseRace\_g02\_c01$ is busier than $v\_HorseRace\_g01\_c01$ but both share divergences from the central components on the x and y plots as expected due to the spreading of the pack of horses. In the first sequence, three locations of separate activity are represented, corresponding to the spectator lane at the top, the central raceway, and the text caption at the bottom. The plots for both sequences share larger variations in the x plot than the y plot, which matches the semantic understanding of horse racing as typically describing horizontal motion. In sequence $v\_HorseRace\_g02\_c01$, the smaller length of the red steady components in both plots represent the shorter temporal existence of the text caption compared to the horses.

The following are some general observations we can make about the properties of region class trajectory plots. Vertical or nearly vertical lines in literal terms

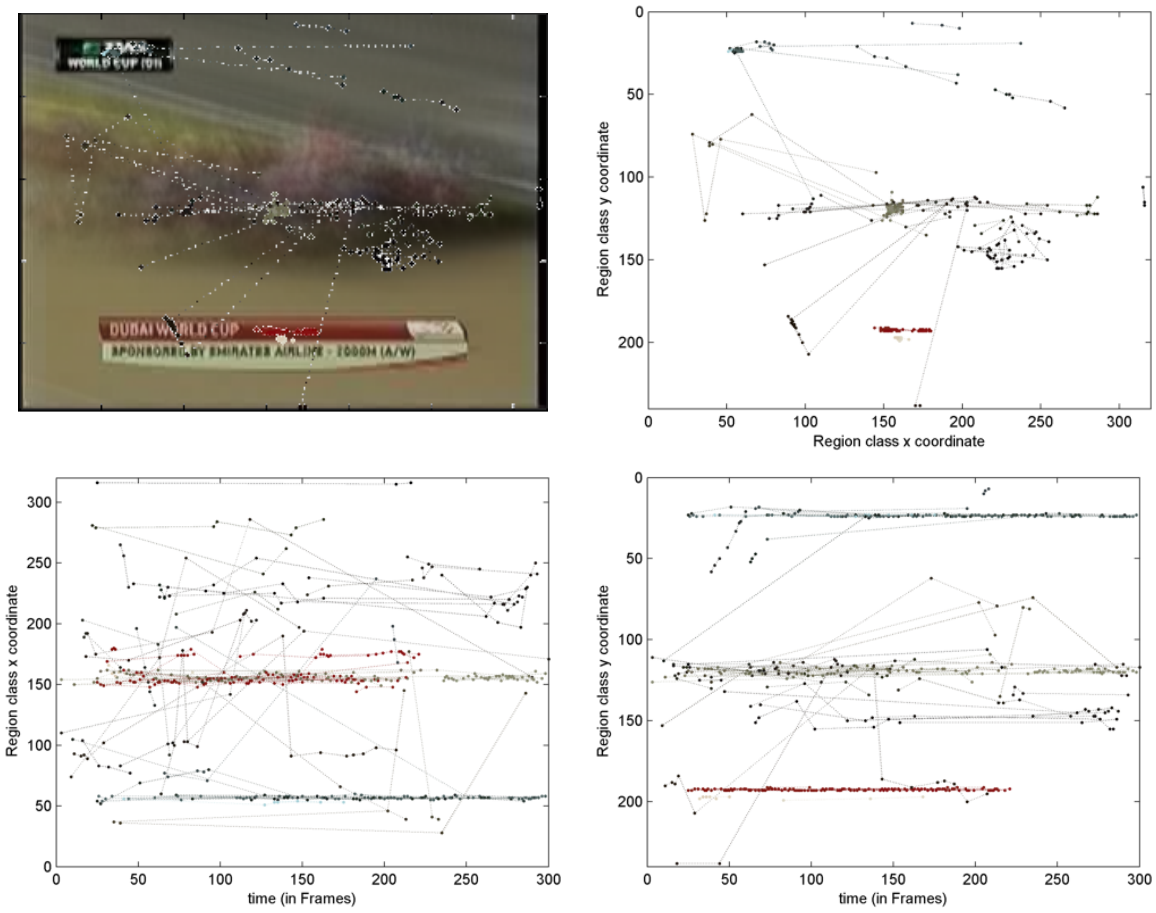

Figure 5.9: UCF50 video sequence v_HorseRace_g02_c01. Top left: Mean of video frames with spatial trajectories superimposed. Top right: Spatial trajectories, x vs. y. Bottom left: x trajectories vs. time. Bottom right: y trajectories vs. time.

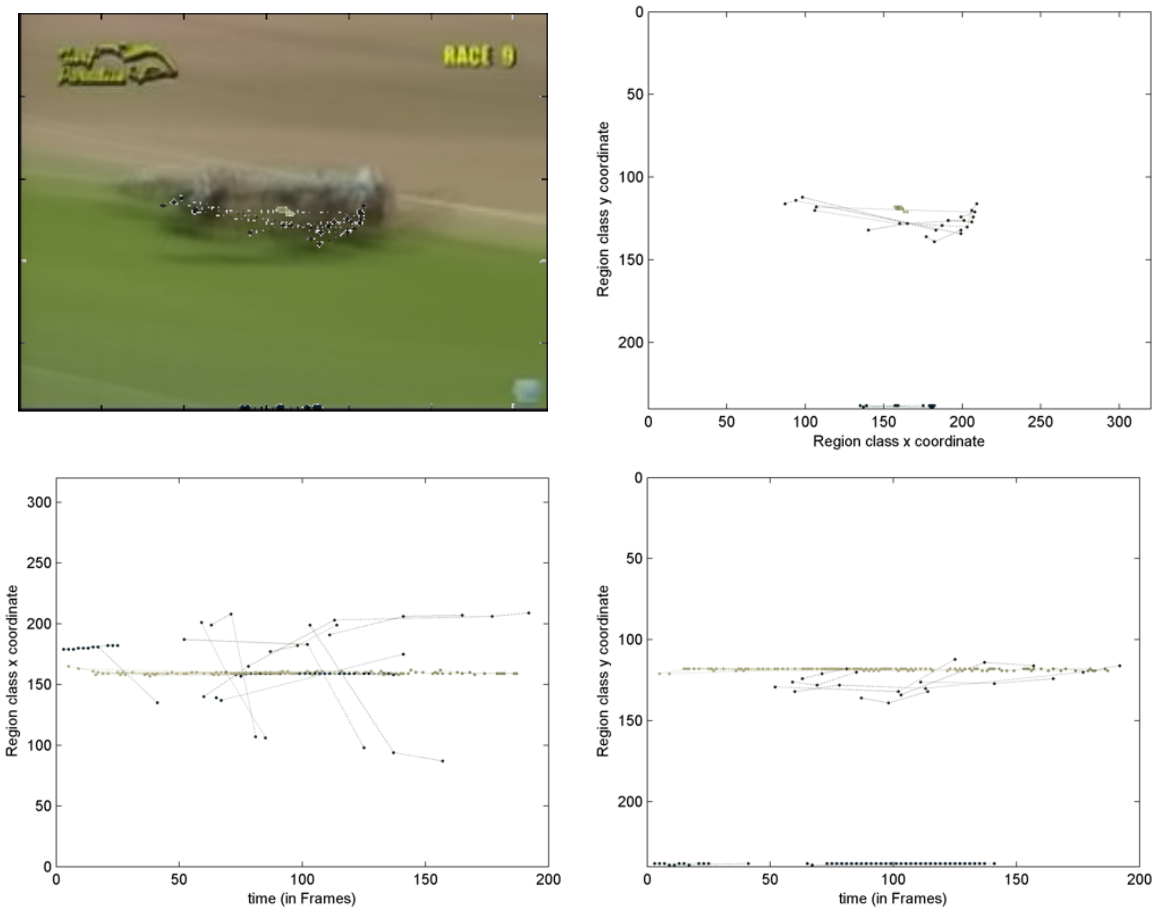

Figure 5.10: UCF50 video sequence v_HorseRace_g01_c01. Top left: Mean of video frames with spatial trajectories superimposed. Top right: Spatial trajectories, x vs. y. Bottom left: x trajectories vs. time. Bottom right: y trajectories vs. time.

show large fluctuations in region locations, however in real video sequences these are more likely the co-occurrence of similar shapes at different locations rather than regions moving about at high speeds. Also helping us differentiate between angular components of motion are dense horizontal steady components appearing in one plot that correspond to scattered trajectories in the other. Horizontal steady components in either an x or y plot indicate that the coordinate holds steady over time, while the corresponding trajectories in the other plot indicate that the other coordinate is changing, thus indicating horizontal or vertical movement of regions. The mean gradient of any single region class trajectory indicates the level of change with respect to either the x or y coordinates, and thus taking a ratio of x trajectory mean gradient to y trajectory mean gradient gives us the overall "flatness" or "tallness" of a particular visual subevent.

There are more details describing a motion that can be found when looking at individual trajectory components. Harmonic motion, for instance a regular up and down movement, can be identified as a set of alternating gradients in one or both of the temporal activity plots. While harmonicity is hard to reliably extract with the present system configuration, there are other descriptors describing the characteristics of motion that we can compute. Based on our visual analysis of trajectory plots, we propose the following set of motion descriptors:

I. Instance Busyness M_{IB}: number of region instances or dots on any of the three plots

II. Class Busyness M_{CB}: number of trajectories, equivalent to the number of region classes

III. Fragmentedness M_F: the lack of connectedness calculated as the ratio of dots to lines, equivalent to the average size of connected trajectories

IV. Orientation M_O: mean of absolute x and y gradients from the spatial presence plot

V. Orientation Variation M_{OV}: variability of trajectory orientation means, calculated as the ratio of standard deviation of x and the standard deviation of y from the spatial presence plot

VI. Turn Sharpness M_{TS}: average absolute angle of trajectory turn

VII. Turn Variability M_{TV}: standard deviation of means of absolute turning angles for individual trajectories

	Motion descriptors						
Sequence	M_{IB}	M_{CB}	M_F	M_O	M_{OV}	M_{TS}	M_{TV}
v_TrampolineJumping_g01_c01	307	68	4.5147	2.9157	10.3202	0.90175	0.39055
v_TrampolineJumping_g02_c01	155	29	5.3448	1.2553	1.1957	0.59718	0.37837
v_Pullup_g10_c01	136	35	3.8857	0.81646	0.72387	0.90607	0.36443
v_Pullup_g06_c01	279	51	5.4706	0.94654	0.82011	1.337	0.60844
v_BenchPress_g01_c01	265	54	4.9074	1.6455	2.8025	1.3215	0.42347
v_BenchPress_g20_c01	196	34	5.7647	0.50679	0.56948	1.1331	0.39187
v_Fencing_g09_c01	148	28	5.2857	1.3676	1.2094	1.0249	0.62173
v_Fencing_g18_c01	175	35	5	1.0286	0.87877	0.97323	0.5201
v_HorseRace_g01_c01	180	38	4.7368	7.023	6.8331	0.61938	0.23832
v_HorseRace_g02_c01	497	105	4.7333	3.2697	2.3557	0.94484	0.48666
v_Billards_g01_c01	126	36	3.5	1.5461	2.3588	0.93952	0.1708
v_Billards_g05_c01	710	132	5.3788	1.4367	1.0999	1.3844	0.6599
v_Drumming_g13_c01	1492	202	7.3861	1.2942	1.3862	1.0066	0.67922
v_Drumming_g11_c01	954	141	6.766	1.5367	1.3361	0.99201	0.61149

Table 5.3: Motion descriptors calculated on 14 sequences from the UCF50 dataset.

We calculate values for these descriptors several UCF50 video sequences, shown in Table 5.3. The values show patterns particularly in sequences that show better defined visual patterns in the trajectory plots. For instance, both drumming sequences have very similar values for M_F, M_O, M_{OV}, M_{TS} and M_{TV}, also true for both fencing sequences. The pull-up sequences are also similar over the same set of descriptors excluding M_{TV}.

In order to test whether video classification using these descriptors produces meaningful results over a standard dataset, we run an action recognition experiment by selecting 5 video classes, $Drumming$, $Billiards$, $HorseRace$, $BenchPress$ and $Pullup$, from the UCF50 dataset, and select 50 videos per class. We follow the processing steps required to obtain for each video the 7 motion descriptors listed above, normalising all values to within the range $[0, 1]$ to allow for fair clustering distance computations. Then we use the mean shift clustering [50] to group together points from the data cloud. Mean shift follows a gradient ascent procedure to find the modes of local estimated density and has no embedded assumptions on the shape of the distribution nor the number of clusters. We run the mean shift algorithm using the Gaussian kernel and a range of bandwidths $\{0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$, since we are dealing with the range $[0, 1]$. In all cases we find that the computational grouping has very weak correlation to the ground truth, however this is not surprising. Our identification of motion descriptors has been targeted for the differentiation of certain semantic content that is not always reflected by the grouping of the sequences within the dataset. The UFC50 dataset groups together videos of different motion intensity, direction, angle of view, etc. as long as the contained objects and general activity type are the same, while our motion descriptors are designed to represent "textural" properties of contained actions. Thus the values we obtain will not necessarily correspond to

the dataset grouping, but may indicate other similarities that correspond to our observations about the video content. For example, the M_{CB} values identify sequences $Drumming$ 13, $Drumming$ 11, $Billards$ 05 and $HorseRace$ 02, to be relatively busier in terms of motion activity, and this can be confirmed through visual assessment.

Having discussed the relationship between the plotted region class trajectory information and the semantic content of each of the video sequences, we conclude that region class trajectory characteristics and a set of appropriately selected motion descriptors can provide indications of semantic content and future work into the formal specification of the algorithmic analysis of such trajectories can lead to a well developed theory of fully unsupervised and general scene understanding. The implementation we use to obtain experimental results, such as those above, is prone to errors, due to spurious visual content such as information dialogues and text, lighting effects, and pixelation, as well as due to imperfections that are cascaded through the individual algorithmic stages of segmentation, shape representation and region categorisation. However, despite such difficulties we have demonstrated the ability of the framework to represent simple physical concepts differently and to differentiate between types of observed activity with respect to semantic content.

5.5 Summary

In this chapter we have presented an online unsupervised centroid based categorisation method, that needs no training and refines its categories as each new instance of data becomes available, and established optimal values for its classification parameter d_t for descriptor sets F_{corr} and F_{rank}, of which F_{rank} proves supe-

rior. Optimal classification settings are identified according to best accuracy and classification power over well known evaluation metrics and using data from the standard COIL-100 dataset. We have then demonstrated the ability to track spatial regions using simple appearance and locality information, and have gone on to extend this to shape-based region behaviour analysis.

Through the online learning mechanism, we have performed region categorisation using visual segments as the input. Regions in the same category are graphically plotted as region class trajectories in order to demonstrate correlation between trajectory characteristics and video semantic content. Shape-based region behaviour analysis is more generalised in applicability than localised appearance-based spatial region tracking since trajectories are no longer limited to consecutive frames, and the trajectories represent greater abstraction in that they represent the behaviour of region classes rather than region instances. We propose three types of graphical plots that aid the visual analysis of region class trajectories and, based on our analysis of these over several UCF50 video sequences, we propose a set of seven motion descriptors for which we present values for each sequence.

Chapter 6

Conclusions

This chapter concludes with a discussion of the step-by-step approach to the unsupervised identification of semantic concepts from visual scenes that we have demonstrated in this thesis. Each stage reduces the data space through data generalisation, allowing the next to work on the abstracted information to form higher level generalisations. Spatial segments generalise pixels as regions according to mean colour properties, spatiotemporal segments generalise groups of spatial regions as motion objects according to mean motion properties, GNG contour networks generalise region shapes as structures of nodes and edges, learnt region classes generalise visual instances according to colour, size and shape, and finally region class trajectories generalise visual object classes with respect to their appearance and motion.

6.1 A visual abstraction framework

The presented work consists of multistage information abstraction starting with raw pixels and ending with visual trajectories and their characteristics. Spatial seg-

mentation combines collections of pixels into regions, to which spatiotemporal region reduction is applied to obtain larger regions. Region shapes are then modelled using the GNG, reducing shape contour data space. Finally, a nearest-centroid online classification method groups visual regions into classes, the instance-wise physical trajectories of which are then followed over the length of video sequences and characterised by motion descriptors, representing the highest level of visual information abstraction. We have described a multi-component framework using which we achieve this extraction of abstract semantic concepts from video data.

The input is first broken up into regions in Chapter 3 via a spatial colour image segmentation method SGAT (Section 3.3), which consists of multi-stage merging, the stages being region formation, refinement and reduction. The segmentation method describes an efficient region merging algorithm that combines and adapts into a single framework techniques that are found separately in previous literature. The adaptations include the more expensive best merge and a time-expanding theshold that allows cascaded region growth and simultaneously provides for the correction of errors inherent in a single pass scan. A new performance summary indicator, relative performance RP (Section 3.3.3), is also proposed. RP is able to combine arbitrary sets of evaluation metrics into a single number, allowing the instant comparison of performances for different labeling methods, of which image segmentation is a type. Quantitative and qualitative evaluations of SGAT segmentation prove its superiority to other leading segmentation methods.

For video sequences, spatial regions from each frame are placed in temporal context (Section 3.4) using two types of motion information: scalar intensity differences and vector optical flow. Further region reduction is carried out based on mean motion information for spatial segments. Grouping regions using scalar motion information (Section 3.4.1) provides stable results for the segmentation of a

set of non-adjacent foreground objects in front of a low to medium cluttered background, particularly when the foreground object is to be held steady even when they briefly stop moving, such as in webcam applications. However, it is vector motion information (Section 3.4.2) that is applied towards the later stage of semantic video analysis, due to its ability to discriminate between adjacent spatial segments of different motion directions. Optical flow vectors are computed for every pixel using a dense Lucas-Kanade flow estimation technique. The mean optical flow vector for every spatial segment, called region flow, is calculated and used as the basis of further region reduction to achieve spatiotemporal segmentation.

To allow the inclusion of shape as an element of appearance, Chapter 4 explores region modelling and description. Visual region contours are modelled using an optimised GNG shape representation (Section 4.2) and the resulting GNG networks simplified through a novel procedure. The optimisations to the original GNG, concerning the distance measure, stopping criterion and starting configuration, produce a nearly five-fold speedup in network convergence with only a minimal effect on topology correctness. The novel network simplification technique, via the elimination of complex sets of edges connections and the rejoining of hanging network segments, also aids performance by reducing the contour representation to one or more chain-like structures for which many curvature features are easily established. The discriminative power of 30 appearance and contour-based shape descriptors are then experimentally evaluated (Section 4.3). A complete set of inter-descriptor correlations are assessed and subset selection performed in order to identify a low redundancy feature set.

Chapter 5 moves on to the final stage of visual scene understanding. Section 5.2 shows a method for tracking and extracting objects of interest from video sequences using only non-shape appearance and location properties of scalar motion

segments, a natural extension of scalar temporal segmentation for basic scene analysis which assumes sequential existence of objects. However, for more advanced scene understanding we require the ability to identify the discontinuous existence of types of visual objects or region classes. To this end, a rapid centroid-based learning scheme (Section 5.3) then categorises all the observed regions into classes. Three types of trajectory representations are proposed: spatial presence, horizontal variation and vertical variation. These representations model the behaviour of instances in each region class as physical trajectories in the spatial and temporal domains, and are a higher level of abstraction describing the original input. Trajectory analyses (Section 5.4) allow us to draw conclusions regarding scene content in terms of some simple semantic concepts relating to the physical world, such as direction and intensity of object motion, motion regularity, and spatiotemporal locations of heightened activity.

As with all algorithmic systems, there are limitations in each processing step. The spatial segmentation is reliant on parameter settings, but is also robust to small parameter changes, as seen from the $SGAT_1$ and $SGAT_2$ performance statistics shown in Figure 3.2. In addition to parameters intrinsic to the algorithm, there are a number of extrinsic variables as well. The median filtering preprocessing step can affect the outcome depending on the size of the filter mask. For edge-based merge restriction, the spatial resolution of edge detection is another such variable, which is heavily scene dependent and impossible to select a single generic value for. In the spatiotemporal segmentation, the algorithm for which is similar in principle to the spatial segmentation, the choice of optical flow computation method can also influence the final result. Also, in comparison to a sparse optical flow computation, there are more statistical errors resulting from forcing a dense flow map given uncertain localised motion information. Finally, the output of each processing step

heavily influences the performance of the next. For contour representation and shape learning, a poor segmentation severely limits perceptual correlation to real world objects and hence the calculation of useful motion trajectories.

In summary, the framework we have proposed for automatic semantic abstraction of visual sensory streams is dependent on a series of cascaded algorithmic steps, starting with segmentation and ending with the analysis of region trajectories. While dependent, the framework also displays a robustness to variations within each of the steps, such as variations in segmentation resolution, the extent of region reduction, the convergence characteristics of the GNG, as well as to variations in the centroid based region class learning parameters. An aspect of this robustness can be seen from the useful trajectory information that is present whether directly analysing the spatial segmention or whether including the spatiotemporal region reduction.

While such dependencies and corresponding robustness are open to further exploration, we already see that there emerges within this framework a redundancy and flexiblity comparable to primitive biological systems. It is a plausible hypothesis that multistage cascaded levels of information abstraction from raw inputs leads to a capacity for generic functioning that is tolerant to noise, unpredictability and the diversity generally present in real world visual events that biological entities are regularly required to process.

Applications of the framework include surveillance and alert systems via categorisation of visual events and the general advancement of computational visual understanding. Individual components of the framework are also highly applicable to current science and industry. Image segmentation is frequently used for a wide range of vision tasks, where performance speed and perceptual correlation to human perception are of importance. The scalar motion segmentation can be ap-

plied to the manipulation of entire backgrounds in a range of situations including video conversations over digital devices, interactive gaming, movie editing, etc. Shape representation and analysis itself finds further applications in a variety of areas such as gesture and pose recognition, behaviour modelling, and the general tasks of object detection and recognition.

The following section highlights possible directions of future research that may shed further light on the intricacies of region merging, spatiotemporal grouping, appearance modelling, unsupervised visual understanding, and multi-component abstraction-forming systems in general.

6.2 Future Work

Each component of the proposed scene understanding framework has the potential to be improved further as we work to get a better understanding on the mechanisms underlying each technique. While the possiblities for improvement are seemingly endless, we highlight in the following paragraphs a few of the most prominent considerations that became apparent during the course of our research.

In region merging we need to gain a better understanding of what characterises a good "region" to the human visual perception system, and to carry out experiments to identify better region descriptors that can then be used in the calculation of inter-region distances for merging purposes. Some possibilities include textural information, region boundary statistics and lateral multi-interaction between descriptors for neighbouring segments. Identifying a region description that includes the properties of neighbouring regions would possibly indicate a default merging stopping criterion as one in which further merges would violate the integrity of a region's description. However, it is possible that an information theoretic stop-

ping criterion established along these lines would depend on feedback from higher level processes of abstraction, such as contextual scene understanding, that help confirm or reject automatic region hypotheses triggered at the merging level. This relates to the question of whether human perception of objects is sensitive to contour changes or more reliant on higher level appearance abstractions. It would also be useful to improve and extend our proposed relative performance summary indicator to work with extended sets of diverse evaluation metrics in order to be better indicative of overall segmentation quality.

We also propose further research into whether motion information is best analysed at a low level, such as computing optical flow between frames, or whether it is a result of higher order processes such as the categorisation of region class and region instance trajectories, or a combination of both low and high level mechanisms. Since the proposed trajectory plots cover entire sequences of visual frames, motion information deduced from these would be drawn from considerations global to not only a single frame or a pair of frames but indeed global to the entire sequence. This points to the question of how to split a continuous video input stream into separate sequences, however individual video sequences from standard datasets allow the assumption of a pre-split video stream.

Finally, a natural extension of the work presented here would be to establish descriptors for the region class trajectories themselves, and then categorise these to obtain abstractions about groups and types of observed physical events. The work presented here lays a promising foundation for the development of a fully unsupervised generic scene understanding system. For this to happen we must first formalise a theory of semantic conceptualisation from spatial and spatiotemporal trajectories, starting with simple physical concepts and gradually expanding to more sophisticated or even compound semantics. While a few clear patterns in

the trajectory plots corresponding to basic semantics of video content are apparent to the human observer, we do not yet have a precise set of rules guiding the algorithmic detection and interpretation of such patterns. Nor are we certain whether other patterns exist in the trajectory plots that are not apparent to the human observer but which may be algorithmically detected and included within a future framework of automatic scene interpretation. Sufficiently developed conceptual understanding mechanisms could find applications in event detection systems for monitoring and security and also allow further developments in generalised computational visual understanding.

Appendix A

Shape descriptor correlations

This appendix shows the complete pairwise linear correlation figures for 30 region appearance features combining colour and size information with shape descriptors calculated from the GNG network applied to region contours.

A.1 Region appearance descriptor correlations

In Chapter 4 we had started with a list of candidate region appearance descriptors and tested their discriminative performance. Items from this list were then discarded according to their pairwise correlations, and Chapter 4 shows the pairwise correlation values for the shortlisted feature set. In order to get a more complete picture of the interdependence between each of the features, we present in this appendix the complete set of pairwise descriptor correlations. Pairs that satisfy the standard test of correlation strength, $corr^2 > 0.5$, are marked.

Table A.1 shows pairwise correlations between all 30 appearance descriptors (3 to 32) as well as the three control variables (1, 2 and 33). The three control variables are $group$, $instance$ and $rand$, which are present for verification purposes.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	-																																
2		-																															
3			-																														
4				-																													
5					-																												
6						-			X																								
7							-	X																			X						
8							X	-																			X						
9						X			-																		X						
10										-																							
11											-	X				X					X		X					X	X		X		
12											X	-							X		X										X		
13													-			X		X										X	X	X			
14														-	X																		
15														X	-																		
16											X		X			-		X			X		X					X	X	X	X		
17																	-																
18													X			X		-										X	X	X			
19												X							-					X							X		
20																				-	X										X		
21											X	X				X				X	-		X					X	X		X		
22																						-											
23											X					X					X		-					X	X				
24																			X					-									
25																									-								
26																										-			X	X			
27									X																		-						
28							X	X																				-					
29											X		X			X		X			X		X			X			-	X	X	X	
30											X		X			X		X			X		X			X			X	-	X	X	
31													X			X		X											X	X	-		
32											X	X				X			X	X	X								X	X		-	
33																																	-

Table A.1: Correlations for region descriptors: 1. $group^{++}$, 2. $instance^{++}$, 3. B, 4. G, 5. R, 6. $area$, 7. $nodes$, 8. $labels$, 9. $eig1$, 10. $eig2$, 11. var, 12. $range$, 13. iqr, 14. $skew$, 15. $kurt$, 16. $mean$, 17. min, 18. med, 19. max, 20. $mean - med$, 21. std, 22. $numInflex$, 23. $meanInflex$, 24. $num135$, 25. $num90$, 26. $num45$, 27. $num0$, 28. $circ$, 29. $curv$, 30. $mad0$, 31. $mad1$, 32. $mom2$, and 33. $rand^{++}$. Control variables are marked with $^{++}$.

Four other variables, B, G, R and $area$ are non-shape appearance descriptors, and are independent of the GNG contour representation. The non-shape descriptors are nonetheless present in both tables in order to observe any correlation with the shape descriptors. Rows with no markings represent independent variables. Leaving aside the three control variables, the set of appearance descriptors contain four variables B, G, R and $numInflex$ that are independent of all the others.

Appendix B

Descriptor set evaluation and selection

This appendix discusses the feature subset and variable ranking methods for descriptor selection. The selection precodures are described and performance evaluation statistics presented for each method.

B.1 F_{corr}

In Chapter 4 we had identified 19 region descriptors, out of the 30 we had considered, to satisfy the selection criterion $ICV/ICD < 0.1$, based on the observation that this criterion holds for the four most intuitively relevant features, B, G, R and $area$ in Table 4.2. The 19 descriptors were: a) B, b) G, c) R, d) $area$, e) $nodes$, f) $labels$, g) $eig1$, h) var, i) iqr, j) $mean$, k) med, l) $numInflex$, m) $meanInflex$, n) $num45$, o) $num0$, p) $circ$, q) $curv$, r) mad_0, s) mad_1. Many of these are correlated (see Table 4.3) and this set therefore contains high redundancy. To minimise such redundancy we identify strongly correlated feature subgroups and select from each

d_t	N_c (≈ 100)	PRI (↑)	VOI (↓)	GCE (↓)	BDE (↓)	$RP_{PRI,VOI,GCE,BDE}$ (↑)
0	7200	0.99014	6.1699	0	0.00013889	-4.3335
0.005	7169	0.99014	6.1606	0	6.94E-005	-4.3186
0.01	7039	0.99015	6.1187	0	6.94E-005	-4.2516
0.02	6465	0.9902	5.9059	0.00013889	1.39E-004	-3.9147
0.03	5718	0.99027	5.6395	0.0051731	6.94E-005	-3.6005
0.04	4993	0.99036	5.3934	0.019978	6.94E-005	-3.5348
0.05	**3668**	0.99058	4.8831	0.052965	1.39E-004	**-3.4492**
0.06	3047	0.99066	4.714	0.096301	0	-4.1377
0.07	2362	0.99075	4.5199	0.15474	6.94E-005	-5.1203
0.08	1884	0.99066	4.4746	0.22175	0.000069444	-6.5307
0.09	1587	0.99058	4.4189	0.2685	0.00020833	-7.4763
0.1	1188	0.99037	4.4549	0.33669	0.00013889	-9.0427
0.2	209	0.9812	5.8103	0.76461	6.94E-005	-20.6933
0.26	*102*	*0.9738*	*6.5965*	*0.78265*	*9.03E-004*	*-22.3632*
0.3	59	0.96115	6.841	0.74912	0.00090301	-22.0374
0.4	31	0.94269	7.6556	0.77471	0.0021534	-23.9424
0.5	18	0.90487	7.8726	0.74775	0.005906	-23.77
0.6	16	0.89255	7.9715	0.72899	0.011687	-23.5396
0.7	11	0.84233	7.7259	0.65332	0.030844	-21.5812
0.8	4	0.62567	7.163	0.41249	0.11801	-15.8213
0.9	1	0.0098625	6.6439	0	6.5917	-9.3923
0.95	1	0.0098625	6.6439	0	6.5917	-9.3923

Table B.1: Clustering performance of descriptor set F_{corr}: $\{B, G, R, area, nodes, numInflex, num0, curve\}$. Clustering threshold d_t vs. number of clusters N_c and evaluation measures PRI, VOI, GCE, BDE and RP. Desired values in parentheses. $d_t \approx 0.05$ at best RP, $d_t \approx 0.26$ at best N_c.

subgroup only that feature with the best ICV/ICD value, discarding the rest. By this process we reduce the set of 19 features to a set of 8, which gives us descriptor set $F_{corr} = \{B, G, R, area, nodes, numInflex, num0, curve\}$ with low redundancy. Table B.1 shows extended clustering performance evaluation statistics using F_{corr} to learn categories from the COIL-100 dataset.

B.2 F_{rank}

From the 30 features considered in Chapter 4, we use a variable ranking approach to establish an alternate descriptor set F_{rank} against which the performance of the previously determined set F_{corr} may be compared. To do this, we first rank all 30 features in ascending ICV/ICD order. We start with the feature at the top of the list, to get a feature vector of size 1, and evaluate clustering performance for a range of d_t values, noting the maximum RP achieved, and setting this also as the globally best RP_{best}. Then we add the second feature in the list to the first and evaluate this new feature vector of size 2 through the same range of d_t. If the maximum RP for any d_t in this set is greater than RP_{best}, then this replaces the existing RP_{best} value. We continue adding one feature at a time from the ranked list and evaluate each incrementally expanding feature set, updating RP_{best} as we proceed. At the end of this process, the feature set for which the maximum RP equals RP_{best} is considered the optimal variable ranked descriptor set F_{rank}.

We find the global optimum for RP_{best} across the full set of 30 descriptors to occur at the seventh variable ranked incremental set, and we thus present clustering evaluation statistics for sets 3 to 7 (Tables B.2, B.3, 5.2, B.5 and B.6). These allow us to observe the peak in RP_{best} which gives us the variable ranked descriptors F_{rank}: $\{B, R, G, numNodes, circ\}$, for which extended evaluation statistics are given.

d_t	N_c ($\approx$ 100)	PRI ($\uparrow$)	VOI ($\downarrow$)	GCE ($\downarrow$)	BDE ($\downarrow$)	$RP_{PRI,VOI,GCE,BDE}$ ($\uparrow$)
0	7200	0.99014	6.1699	0	0.00013889	-4.3335
0.005	4656	0.9904	5.2707	0.017424	0.00013889	-3.2823
0.01	1940	0.99113	3.9897	0.074528	0.00013889	**-2.4983**
0.02	758	0.99187	3.12	0.16199	6.94E-005	-3.0427
0.03	441	0.99181	2.9253	0.25134	6.94E-005	-4.7085
0.04	284	0.9912	2.9658	0.34468	6.94E-005	-6.8396
0.05	201	0.98928	3.0348	0.42703	0	-8.7753
0.06	156	0.98729	3.258	0.5199	1.39E-004	-11.1905
0.07	117	0.98564	3.4965	0.57951	6.94E-005	-12.8937

Table B.2: Performance evaluation of region descriptor set: $\{B, R, G\}$. Clustering threshold d_t vs. number of clusters N_c and evaluation measures PRI, VOI, GCE and BDE. Desired values in parentheses.

d_t	N_c ($\approx$ 100)	PRI ($\uparrow$)	VOI ($\downarrow$)	GCE ($\downarrow$)	BDE ($\downarrow$)	$RP_{PRI,VOI,GCE,BDE}$ ($\uparrow$)
0	7200	0.99014	6.1699	0	0.00013889	-4.3335
0.005	5702	0.99027	5.6402	0.0015741	0.00013889	-3.522
0.01	3252	0.99079	4.5563	0.018034	0.00020833	-2.154
0.02	1538	0.99149	3.6066	0.077328	1.39E-004	**-1.9475**
0.03	908	0.99181	3.2203	0.1498	6.94E-005	-2.9333
0.04	599	0.99184	3.0553	0.22391	6.94E-005	-4.3094
0.05	410	0.99108	3.056	0.31202	0	-6.2611
0.06	299	0.99013	3.1389	0.39052	6.94E-005	-8.1322
0.07	239	0.98896	3.3666	0.46134	0.00020837	-10.0652

Table B.3: Performance evaluation of region descriptor set: $\{B, R, G, numNodes\}$. Clustering threshold d_t vs. number of clusters N_c and evaluation measures PRI, VOI, GCE and BDE. Desired values in parentheses.

d_t	N_c (≈ 100)	PRI (↑)	VOI (↓)	GCE (↓)	BDE (↓)	$RP_{PRI,VOI,GCE,BDE}$ (↑)
0	7200	0.99014	6.1699	0	0.00013889	-4.3335
0.005	5783	0.99028	5.6536	0.00034722	6.94E-005	-3.5163
0.01	3405	0.99075	4.6062	0.012469	2.08E-004	-2.1107
0.02	1661	0.99153	3.6118	0.067515	6.94E-005	**-1.7386**
0.03	1003	0.9918	3.2483	0.13583	1.39E-004	-2.669
0.04	638	0.99186	3.0686	0.22047	1.39E-004	-4.2542
0.05	440	0.99115	3.0555	0.30196	0	-6.0377
0.06	328	0.98988	3.2813	0.39515	1.39E-004	-8.4626
0.07	246	0.98859	3.3213	0.46168	0.00013889	-10.0011
0.08	205	0.98655	3.666	0.5362	0.00013891	-12.2045
0.09	173	0.98456	3.9177	0.59581	6.94E-005	-13.9292
0.1	146	0.98246	3.9716	0.61054	0.00020837	-14.3453
0.12	103	0.9764	4.3462	0.60777	0.00041676	*-14.8948*
0.2	49	0.95687	5.7714	0.66303	0.001251	-18.4333
0.3	23	0.92525	6.5411	0.62654	0.0029182	-18.9194
0.4	14	0.8919	7.027	0.62907	0.0068797	-19.8203
0.5	12	0.88425	7.7269	0.69814	0.0085417	-22.4822
0.6	14	0.85969	7.8579	0.6834	0.014655	-22.417
0.7	12	0.86988	7.9181	0.69035	0.01263	-22.6456
0.8	7	0.68911	7.4159	0.49597	0.15033	-17.9557
0.9	4	0.41842	7.1604	0.32042	1.8782	-14.819
0.95	1	0.0098625	6.6439	0	6.5924	-9.3926

Table B.4: Performance evaluation of region descriptor set F_{rank}: $\{B,\ R,\ G,\ numNodes, circ\}$. Clustering threshold d_t vs. number of clusters N_c and evaluation measures PRI, VOI, GCE and BDE. Desired values in parentheses. $d_t \approx 0.02$ at best RP, $d_t \approx 0.12$ at best N_c.

d_t	N_c (≈ 100)	PRI ($\uparrow$)	VOI ($\downarrow$)	GCE ($\downarrow$)	BDE ($\downarrow$)	$RP_{PRI,VOI,GCE,BDE}$ ($\uparrow$)
0	7200	0.99014	6.1699	0	0.00013889	-4.3335
0.005	6089	0.99024	5.7616	0.00034722	1.39E-004	-3.6888
0.01	4307	0.99063	4.9808	0.0088877	6.94E-005	-2.63
0.02	2764	0.99117	4.2268	0.055305	0.00E+000	**-2.4515**
0.03	2058	0.99136	3.9032	0.11222	2.08E-004	-3.1934
0.04	1580	0.99118	3.8752	0.19124	6.94E-005	-4.8973
0.05	915	0.99088	3.5863	0.2747	2.08E-004	-6.2831
0.06	698	0.98995	3.6474	0.36611	0.00013891	-8.4047
0.07	529	0.98916	3.7669	0.43157	6.94E-005	-10.0455

Table B.5: Performance evaluation of region descriptor set: $\{B$, R, G, $numNodes$, $circ$, $numLabels\}$. Clustering threshold d_t vs. number of clusters N_c and evaluation measures PRI, VOI, GCE and BDE. Desired values in parentheses.

d_t	N_c (≈ 100)	PRI ($\uparrow$)	VOI ($\downarrow$)	GCE ($\downarrow$)	BDE ($\downarrow$)	$RP_{PRI,VOI,GCE,BDE}$ ($\uparrow$)
0	7200	0.99014	6.1699	0	0.00013889	-4.3335
0.005	6911	0.99016	6.0761	0.00018519	6.94E-005	-4.1876
0.01	5986	0.99025	5.7306	0.001412	6.94E-005	-3.6628
0.02	4493	0.99049	5.156	0.02769	1.39E-004	**-3.326**
0.03	3531	0.99072	4.7717	0.068272	1.39E-004	-3.6097
0.04	2773	0.99081	4.5401	0.13068	1.39E-004	-4.6203
0.05	1727	0.99093	4.1555	0.20068	0.00E+000	-5.5543
0.06	1379	0.99049	4.1588	0.28002	0	-7.3158
0.07	1049	0.99018	4.1729	0.35372	1.39E-004	-8.9695

Table B.6: Performance evaluation of descriptor set: $\{B$, R, G, $numNodes$, $circ$, $numLabels$, $curv\}$. Clustering threshold d_t vs. number of clusters N_c and evaluation measures PRI, VOI, GCE and BDE. Desired values in parentheses.

Appendix C

Publications

This thesis includes work that was authored or coauthored and published externally within the duration of its creation, presented below in reverse chronological order.

2012

G. Gupta and A. Psarrou. Semi-Greedy Adaptive-Threshold Region Merging via Path Scanning. In *IEEE International Conference on Image Processing, ICIP'12*, 2012.

G. Gupta, A. Psarrou, and A. Angelopoulou. Image Segmentation based on Semi-Greedy Region Merging. In *IET Conference on Image Processing, IPR'12*, pages 1-4, ISBN: 978-1-84919-632-1, 2012.

2010

A. Angelopoulou, A. Psarrou, J. García, and G. Gupta. Tracking Gestures using a Probabilistic Self-Organising Network. In *International Joint Conference on*

Neural Networks (IJCNN'10), IEEE WCCI'10, pages 1–7, IEEE Catalogue Number: CFP10IJS-DVD, ISBN: 978-1-4244-6917-8, 2010.

2009

G. Gupta, A. Psarrou, and A. Angelopoulou. Generic colour image segmentation via multi-stage region merging. In 10^{th} *International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'09*, pages 185–188, IEEE Xplore, 2009.

2008

A. Angelopoulou, A. Psarrou, G. Gupta, and J. García. Active-GNG: Model Acquisition and Tracking in Cluttered Backgrounds. In *ACM Workshop on Vision Networks for Behaviour Analysis, VNBA'08, in conjunction with the ACM Multimedia*, pages 17–22, 2008.

2007

A. Angelopoulou, A. Psarrou, G. Gupta, and J. García. Nonparametric Modelling and Tracking with Active-GNG. In *IEEE International Workshop on Human Computer Interaction, ICCV-HCI'07, in conjunction with the ICCV 2007, LNCS 4796*, pages 98–107, Springer, 2007.

A. Angelopoulou, A. Psarrou, G. Gupta, and J. García. Robust Modelling and Tracking of NonRigid Objects Using Active-GNG. In *IEEE Workshop on Non-rigid Registration and Tracking through Learning, NRTL'07, in conjunction with the ICCV 2007*, IEEE Xplore, 2007.

Bibliography

[1] T. Adamek, N. O'Connor, and N. Murphy. Region-based segmentation of images using syntactic visual features. In *International Workshop on Image Analysis for Multimedia Interactive Services*, WIAMIS'05, pages 1–4, 2005.

[2] T. Adamek and N. E. O'Connor. Stopping region-based image segmentation at meaningful partitions. In *International Conference on Semantic Multimedia, Semantic and Digital Media Technologies*, SAMT'07, pages 15–27, 2007.

[3] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:641–647, 1994.

[4] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

[5] H. Alt, B. Behrends, and J. Blömer. Approximate matching of polygonal shapes. *Annals of Mathematics and Artificial Intelligence*, 13(3-4):251–266, 1995.

[6] M. Andreetto, L. Zelnik-Manor, and P. Perona. Non-parametric probabilistic image segmentation. In *IEEE International Conference on Computer Vision*, ICCV'07, pages 1–8, 2007.

[7] A. Angelopoulou, A. Psarrou, J. Garcia Rodriguez, and G. Gupta. Active-GNG: Model acquisition and tracking in cluttered backgrounds. In *ACM*

Workshop on Vision Networks for Behavior Analysis, VNBA'08, pages 17–22, 2008.

[8] A. Angelopoulou, A. Psarrou, G. Gupta, and J. García-Rodríguez. Nonparametric modelling and tracking with Active-GNG. In *IEEE International Conference on Human-Computer Interaction*, HCI'07, pages 98–107, 2007.

[9] A. Angelopoulou, A. Psarrou, G. Gupta, and J. G. Rodriguez. Robust modelling and tracking of nonrigid objects using active-gng. In *IEEE International Conference on Computer Vision*, ICCV'07, pages 1–7, 2007.

[10] A. Angelopoulou, A. Psarrou, J. Rodrguez, and K. Revett. Automatic landmarking of 2D medical shapes using the growing neural gas network. In *Computer Vision for Biomedical Image Applications*, volume 3765 of *Lecture Notes in Computer Science*, pages 210–219. 2005.

[11] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'09, pages 2294–2301, 2009.

[12] B. Atcheson, W. Heidrich, and I. Ihrke. An evaluation of optical flow algorithms for background oriented schlieren imaging. *Experiments in Fluids*, 46:467–476, 2009.

[13] A. Ayvaci and S. Soatto. Motion segmentation with occlusions on the superpixel graph. In *IEEE International Conference on Computer Vision Workshops, Workshop on Dynamical Vision*, ICCV'09, pages 727–734, 2009.

[14] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, 2007. ISBN: 0521876257.

[15] D. Bailey. Raster based region growing. In *New Zealand Image Processing Workshop*, pages 21–26, 1991.

[16] R. Bajcsy. Active perception. *Proceedings of the IEEE, Special Issue on Computer Vision*, 76(8):966–1005, 1988.

[17] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[18] A. Baraldi, P. Blonda, F. Parmiggiani, and G. Satalino. Contextual clustering for image segmentation. *Optical Engineering*, 39:907–923, 2000.

[19] P. Barral, G. Dorme, and D. Plemenos. Scene understanding techniques using a virtual camera. In *Eurographics 2000, Short Presentations, Rendering and Visibility*, 2000.

[20] J. Barron, D. Fleet, S. Beauchemin, and T. Burkitt. Performance of optical flow techniques. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'92, pages 236–242, 1992.

[21] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27:433–466, 1995.

[22] J.-M. Beaulieu and M. Goldberg. Hierarchy in picture segmentation: A stepwise optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:150–163, 1989.

[23] K. Bhoyar and O. Kakde. Colour image segmentation using fast fuzzy C-means algorithm. *Electronic Letters on Computer Vision and Image Analysis*, 9(1):18–31, 2010.

[24] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.

[25] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, 1967.

[26] R. Boie and I. Cox. An analysis of camera noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:671–674, 1992.

[27] C. Brice. Region growing. In D. Ballard and C. Brown, editors, *Computer Vision*, pages 149–165. Prentice Hall, 1982.

[28] C. R. Brice and C. L. Fennema. Scene analysis using regions. *Artificial Intelligence*, 1(3):205–226, 1970.

[29] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'09, pages 41–48, 2009.

[30] T. Brox, D. Farin, and P. H. N. de With. Multi-stage region merging for image segmentation. In *Symposium on Information Theory in the Benelux*, pages 189–196, 2001.

[31] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *European Conference on Computer Vision: Part V*, ECCV'10, pages 282–295, 2010.

[32] H. Buxton. Learning and understanding dynamic scene activity: a review. *Image and Vision Computing*, 21(1):125–136, 2003.

[33] F. Calderero and F. Marques. Region merging techniques using information theory statistical measures. *IEEE Transactions on Image Processing*, 19:1567–1586, 2010.

[34] J. Cardelino, V. Caselles, M. Bertalmío, and G. Randall. A contrario hierarchical image segmentation. In *IEEE International Conference on Image Processing*, ICIP'09, pages 4041–4044, 2009.

[35] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[36] S. Chabrier, B. Emile, H. Laurent, C. Rosenberger, and P. Marche. Unsupervised evaluation of image segmentation application to multi-spectral images. In *International Conference on Pattern Recognition*, volume 1 of *ICPR'04*, pages 576–579, 2004.

[37] S.-F. Chang, J. R. Smith, M. Beigi, and A. Benitez. Visual information retrieval from large distributed online repositories. *Communications of the ACM*, 40(12):63–71, 1997.

[38] Y. L. Chang and X. Li. Adaptive image region-growing. *IEEE Transactions on Image Processing*, 3(11):868–872, 1994.

[39] S. Chen, L. Cao, Y. Wang, J. Liu, and X. Tang. Image segmentation by MAP-ML estimations. *IEEE Transactions on Image Processing*, 19:2254–2264, 2010.

[40] S.-Y. Chen, W.-C. Lin, and C.-T. Chen. Split-and-merge image segmentation based on localized feature analysis and statistical tests. *CVGIP: Graphical Models and Image Processing*, 53:457–475, 1991.

[41] Y. Chen, R. Yin, P. Flynn, and S. Broschat. Aggressive region growing for speckle reduction in ultrasound images. *Pattern Recognition Letters*, 24:677–691, 2003.

[42] K. S. Chenaoua, A. Bouridane, and F. Kurugollu. Unsupervised histogram based color image segmentation. *IEEE International Conference on Electronics, Circuits and Systems*, pages 240–243, 2003.

[43] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang. Color image segmentation: Advances and prospects. *Pattern Recognition*, 34(12):2259–2281, 2001.

[44] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang. Color image segmentation: Advances and prospects. *Pattern Recognition*, 34(12):2259–2281, 2001.

[45] H. D. Cheng, X. H. Jiang, and J. Wang. Color image segmentation based on homogram thresholding and region merging. *Pattern Recognition*, 35:373–393, 2002.

[46] S. C. Cheng. Region-growing approach to colour segmentation using 3-D clustering and relaxation labeling. *IEE Proceedings - Vision, Image, and Signal Processing*, 150(4):270–276, 2003.

[47] M. Cho and K. M. Lee. Authority-shift clustering: Hierarchical clustering by authority seeking on graphs. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'10, pages 3193–3200, 2010.

[48] C. C. Chu and J. K. Aggarwal. The integration of image segmentation maps using region and edge information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1241–1252, 1993.

[49] D. Chung, R. Hirata, T. N. Mundhenk, J. Ng, R. J. Peters, E. Pichon, A. Tsui, T. Ventrice, D. Walther, P. Williams, and L. Itti. A new robotics platform for neuromorphic vision: Beobots. In *International Workshop on Biologically Motivated Computer Vision*, BMCV'02, pages 558–566, 2002.

[50] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[51] P. Correia and F. Pereira. Objective evaluation of video segmentation quality. *IEEE Transactions on Image Processing*, 12(2):186–200, 2003.

[52] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2 of *CVPR'05*, pages 1124–1131, 2005.

[53] D. Crisp and T. Tao. Fast region merging algorithms for image segmentation. In *Asian Conference on Computer Vision*, ACCV'02, pages 412–417, 2002.

[54] W. Daelemans, S. Buchholz, and J. Veenstra. Memory-based shallow parsing. *The Computing Research Repository (CoRR)*, 1999.

[55] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(1-4):131–156, 1997.

[56] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008.

[57] K. Deng and A. W. Moore. Multiresolution instance-based learning. In *International Joint Conference on Artificial Intelligence*, volume 2 of *IJCAI'95*, pages 1233–1239, 1995.

[58] S. Desa and Q. Salih. Image subtraction for real time moving object extraction. In *International Conference on Computer Graphics, Imaging and Visualization*, CGIV'04, pages 41–45, 2004.

[59] K. S. Deshmukh. Color image segmentation: A review. In *International Conference on Digital Image Processing*, volume 7546 of *Proceedings of the SPIE*, pages 754624–754624–6, 2010.

[60] M. Dimiccoli and P. Salembier. Hierarchical region-based representation for segmentation and filtering with depth in single images. In *IEEE International Conference on Image Processing*, ICIP'09, pages 3497–3500, 2009.

[61] T. Dinh and G. Medioni. Two-frames accurate motion segmentation using tensor voting and graph-cuts. In *IEEE Workshop on Motion and Video Computing*, WMVC'08, pages 1–8, 2008.

[62] M. V. Droogenbroeck and H. Talbot. Segmentation by adaptive prediction and region merging. In *International Conference on Digital Image Computing: Techniques and Applications*, DICTA'03, pages 561–570, 2003.

[63] J. Eakins. Towards intelligent image retrieval. *Pattern Recognition*, 35(1):3–14, 2002.

[64] D. Eppstein. Dynamic connectivity in digital images. *Information Processing Letters*, 62:121–126, 1997.

[65] G. M. Espindola, G. Camara, I. A. Reis, L. S. Bins, and A. M. Monteiro. Parameter selection for region-growing image segmentation algorithms using spatial autocorrelation. *International Journal of Remote Sensing*, 27(14):3035–3040, 2006.

[66] J. Fan, D. K. Y. Yau, A. K. Elmagarmid, and W. G. Aref. Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Transactions on Image Processing*, 10(10):1454–1466, 2001.

[67] J. Fan, G. Zeng, M. Body, and M.-S. Hacid. Seeded region growing: An extensive and comparative study. *Pattern Recognition Letters*, 26:1139–1156, 2005.

[68] D. S. Farin. *Automatic Video Segmentation Employing Object/Camera Modeling Techniques*. PhD thesis, Technische Universiteit Eindhoven, 2005. ISBN: 90-386-2381-X.

[69] E. Fatemizadeh, C. Lucas, and H. Soltanian-Zadeh. Automatic landmark extraction from image data using modified growing neural gas network. *IEEE Transactions on Information Technology in Biomedicine*, 7(2):77–85, 2003.

[70] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:594–611, 2006.

[71] D. Feldman and D. Weinshall. Motion segmentation and depth ordering using an occlusion detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1171–1185, 2008.

[72] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 2004.

[73] C. Fiorio and R. Nock. A concentration-based adaptive approach to region merging of optimal time and space complexities. In *British Machine Vision Conference*, volume 2 of *BMVC'00*, pages 775–784, 2000.

[74] D. A. Forsyth. Benchmarks for storage and retrieval in multimedia databases. In *SPIE Proceedings 4676*, Storage and Retrieval for Media Databases, pages 240–247, 2002.

[75] C. Fredembach and G. Finlayson. Path based colour image segmentation. In *European Conference on Color in Graphics, Imaging and Vision*, CGIV'06, pages 382–386, 2006.

[76] H. Freeman. On the encoding of arbitrary geometric configurations. *IRE Transactions on Electronic Computers*, EC-10:260–268, 1961.

[77] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí. Yet another survey on image segmentation: Region and boundary information integration. In *Computer Vision - ECCV 2002*, volume 2352 of *Lecture Notes in Computer Science*, pages 21–25. 2002.

[78] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí. Yet another survey on image segmentation: Region and boundary information integration. In *European Conference on Computer Vision - Part III*, ECCV'02, pages 408–422, 2002.

[79] M. Fritz, G.-J. M. Kruijff, and B. Schiele. Tutor-based learning of visual categories using different levels of supervision. *Computer Vision and Image Understanding, Special issue on Intelligent Vision Systems*, 114(5):564–573, 2010.

[80] B. Fritzke. Fast learning with incremental RBF networks. *Neural Processing Letters*, 1:2–5, 1994.

[81] B. Fritzke. A growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems 7*, NIPS'95, pages 625–632, 1995.

[82] B. Galvin, B. Mccane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: An analysis of eight optical flow algorithms. In *British Machine Vision Conference*, BMVC'98, 1998.

[83] L. Garcia-Ugarriza, E. Saber, V. Amuso, M. Shaw, and R. Bhaskar. Automatic color image segmentation by dynamic region growth and multimodal merging of color and texture information. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP'08, pages 961–964, 2008.

[84] J. M. Gauch. Image segmentation and analysis via multiscale gradient watershed hierarchies. *IEEE Transactions on Image Processing*, 8(1):69–79, 1999.

[85] F. Ge, S. Wang, and T. Liu. Image-segmentation evaluation from the perspective of salient object extraction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1 of *CVPR'06*, pages 1146–1153, 2006.

[86] T. Georgiou, O. Michailovich, Y. Rathi, J. G. Malcolm, and A. Tannenbaum. Distribution metrics and image segmentation. *Linear Algebra and its Applications*, 425(2-3):663–672, 2007.

[87] T. Gevers. Adaptive image segmentation by combining photometric invariant region and edge information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):848–852, 2002.

[88] R. Girisha and S. Murali. Motion segmentation from surveillance videos using varied number of frames. *International Journal of Recent Trends in Engineering*, 2(2):60–65, 2009.

[89] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. http://authors.library.caltech.edu/7694.

[90] G. Gupta, A. Psarrou, and A. Angelopoulou. Generic colour image segmentation via multi-stage region merging. *International Workshop on Image Analysis for Multimedia Interactive Services*, pages 185–188, 2009.

[91] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[92] L. O. Hall, D. Bhadoria, and K. W. Bowyer. Learning a model from spatially disjoint data. In *IEEE International Conference on Systems, Man & Cybernetics*, SMC'04, pages 1447–1451, 2004.

[93] M. A. Hall. *Correlation-based feature subset selection for machine learning*. PhD thesis, University of Waikato, Hamilton, New Zealand, 1998.

[94] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29(1):100–132, 1985.

[95] R. Hedjam and M. Mignotte. A hierarchical graph-based markovian clustering approach for the unsupervised segmentation of textured color images. In

IEEE International Conference on Image Processing, ICIP'09, pages 1357–1360, 2009.

[96] J. Henderson. Human gaze control during real-world scene perception. *Trends in Cognitive Sciences*, 7(11):498–504, 2003.

[97] I. Hendrickx and A. van den Bosch. Hybrid algorithms for instance-based classification. In *European Conference on Machine Learning*, volume 3720 of *Lecture Notes in Computer Science*, pages 158–169, 2005.

[98] K. Hirata and T. Kato. Query by visual example - content based image retrieval. In *International Conference on Extending Database Technology: Advances in Database Technology*, EDBT'92, pages 56–71, 1992.

[99] S. Hojjatoleslami and J. Kittler. Region growing: A new approach. *IEEE Transactions on Image Processing*, 7(7):1079–1084, 1998.

[100] Y. Hong, J. Yi, and D. Zhao. Improved mean shift segmentation approach for natural images. *Applied Mathematics and Computation*, 185(2):940–952, 2007.

[101] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, pages 185–203, 1981.

[102] B. K. P. Horn and B. G. Schunck. "Determining optical flow": A retrospective. *Artificial Intelligence*, pages 81–87, 1993.

[103] G. Iannizzotto and L. Vita. Fast and accurate edge-based segmentation with no contour smoothing in 2-D real image. *IEEE Transactions on Image Processing*, 9(7):1232–1237, 2000.

[104] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. Three-dimensional shape searching: State-of-the-art review and future trends. *Computer Aided Design*, 37:509–530, 2005.

[105] M. Kampel, H. Wildenauer, P. Blauensteiner, and A. Hanbury. Improved motion segmentation based on shadow detection. *Electronic Letters on Computer Vision and Image Analysis*, 6(3):1–12, 2007.

[106] T. Kanade. Region segmentation: Signal vs semantics. *Computer Graphics and Image Processing*, 13(4):279–297, 1980.

[107] B. Ko and H. Byun. Integrated region-based image retrieval using region's spatial relationships. In *International Conference on Pattern Recognition*, volume 1 of *ICPR'02*, page 10196, 2002.

[108] T. Kohonen. Automatic formation of topological maps of patterns in a self-organizing system. In *Scandinavian Conference on Image Analysis*, 2SCIA, pages 214–220, 1981.

[109] T. Kohonen. *Self-Organising Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, 2001. ISBN: 9783540679219.

[110] T. Kohonen and E. Oja. Visual feature analysis by the self-organising maps. *Neural Computing & Applications*, 7:273–286, 1998.

[111] Y. Kuan, C. Kuo, and N. Yang. Color-based image salient region segmentation using novel region merging strategy. *IEEE Transactions on Multimedia*, 10(5):832–844, 2008.

[112] D. Kulic, D. Lee, and Y. Nakamura. Whole body motion primitive segmentation from monocular video. In *IEEE International Conference on Robotics and Automation*, ICRA'09, pages 558–564, 2009.

[113] M. Kunt. Edge detection: A tutorial review. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP'82, pages 1172–1175, 1982.

[114] N. Kwok, Q. Ha, and G. Fang. Effect of color space on color image segmentation. In *International Congress on Image and Signal Processing*, CISP'09, pages 1–5, 2009.

[115] A. Laika and W. Stechele. A review of different object recognition methods for the application in driver assistance systems. In *IEEE International Workshop on Image Analysis for Multimedia Interactive Services*, WIAMIS'07, pages 10–, 2007.

[116] G. Lakemeyer. All they know: A study in multi-agent autoepistemic reasoning. In *International Joint Conference on Artifical Intelligence*, IJCAI'93, pages 376–381, 1993.

[117] H.-C. Lee, E. J. Breneman, and C. P. Schulte. Modeling light reflection for computer color vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:402–409, 1990.

[118] J. Lee. Digital image enhancement and noise filtering by use of local statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(2):165–168, 1980.

[119] J. Letham, N. Robertson, and B. Connor. Contextual smoothing of image segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW'10, pages 7–12, 2010.

[120] I. Levner. Feature selection and nearest centroid classification for protein mass spectrometry. *BMC Bioinformatics*, 6:68, 2005.

[121] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.

[122] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss. Human-assisted motion annotation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'08, pages 1–8, 2008.

[123] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.

[124] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, volume 2 of *IJCAI'81*, pages 674–679, 1981.

[125] W. Y. Ma and B. S. Manjunath. Edge flow: A technique for boundary detection and image segmentation. *IEEE Transactions on Image Processing*, 9(8):1375–1388, 2000.

[126] 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 *IEEE International Conference on Computer Vision*, volume 2 of *ICCV'01*, pages 416–423, 2001.

[127] 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 *IEEE International Conference on Computer Vision*, volume 2 of *ICCV'01*, pages 416–423, 2001.

[128] T. Martinetz and K. Schulten. A "neural-gas" network learns topologies. *Artificial Neural Networks*, I:397–402, 1991.

[129] G. A. Mastin. Adaptive filters for digital image noise smoothing: An evaluation. *Computer Vision, Graphics, and Image Processing*, 31:103–121, 1985.

[130] B. M. Mehtre, M. S. Kankanhalli, and W. F. Lee. Shape measures for content based image retrieval: A comparison. *Information Processing & Management*, 33(3):319–337, 1997.

[131] M. Meilă. Comparing clusterings: An axiomatic view. In *International Conference on Machine learning*, ICML'05, pages 577–584, 2005.

[132] M. Mignotte. Segmentation by fusion of histogram-based k-means clusters in different color spaces. *IEEE Transactions on Image Processing*, 17(5):780–787, 2008.

[133] H. Mobahi, S. Rao, A. Y. Yang, S. S. Sastry, and Y. Ma. Segmentation of natural images by texture and boundary compression. *The Computing Research Repository (CoRR)*, abs/1006.3679, 2010.

[134] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126, 2006.

[135] J. L. Mundy and A. Zisserman, editors. *Geometric invariance in computer vision*. MIT Press, 1992. ISBN: 0-262-13285-0.

[136] L. Natale, S. Rao, and G. Sandini. Learning to act on objects. In *International Workshop on Biologically Motivated Computer Vision*, BMCV'02, pages 567–575, 2002.

[137] S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (COIL-100). Technical Report CUCS-006-96, Department of Computer Science, Columbia University, 1996.

[138] R. Nock and F. Nielsen. Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1452–1458, 2004.

[139] Y.-I. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, 13(3):222–241, 1980.

[140] N. Pal and S. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.

[141] S. Paris and F. Durand. A topological approach to hierarchical segmentation using mean shift. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[142] H. Park, L. M. Jeon, and J. B. Rosen. Lower dimensional representation of text data based on centroids and least squares. *BIT Numerical Math*, 43:427–448, 2003.

[143] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8):1226–1238, 2005.

[144] S. Pereira, S. Voloshynovskiy, M. Madueno, S. Marchand-Maillet, and T. Pun. Second generation benchmarking and application oriented evaluation. In *International Workshop on Information Hiding*, IHW'01, pages 340–353, 2001.

[145] M. Peura and J. Iivarinen. Efficiency of simple shape descriptors. In C. Arcelli, L. P. Cordella, and G. S. di Baja, editors, *Advances in Visual Form Analysis*, pages 443–451. World Scientific, 1997.

[146] A. Pinz. Object categorization. *Foundations and Trends in Computer Graphics and Vision*, 1(4):255–353, 2005.

[147] S. J. Pundlik and S. T. Birchfield. Real-time motion segmentation of sparse feature points at any speed. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 38(3):731–742, 2008.

[148] A. K. Qin and D. A. Clausi. Multivariate image segmentation using semantic region growing with adaptive edge penalty. *Transactions on Image Processing*, 19:2157–2170, 2010.

[149] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[150] S. Rao, H. Mobahi, A. Y. Yang, S. Sastry, and Y. Ma. Natural image segmentation with adaptive texture and boundary encoding. In *Asian Conference on Computer Vision*, ACCV'09, pages 135–146, 2009.

[151] N. Rasiwasia and N. Vasconcelos. Image retrieval using query by contextual example. In *ACM International Conference on Multimedia Information Retrieval*, MIR'08, pages 164–171, 2008.

[152] N. Rasiwasia, N. Vasconcelos, and P. J. Moreno. Query by semantic example. In *ACM International Conference on Image and Video Retrieval*, LNCS 4071, pages 51–60, 2006.

[153] K. K. Reddy. UCF 50 human action dataset, 2010. vision.eecs.ucf.edu/data/UCF50.rar.

[154] S. J. Reeves. On the selection of median structure for image filtering. *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, 42(8):556–558, 1995.

[155] C. Rosenberger and K. Chehdi. Genetic fusion: Application to multi-components image segmentation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP'00, pages 2223–2226, 2000.

[156] P. Rosin. Computing global shape measures. In C. Chen and P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*, pages 177–196. John Wiley & Sons, Inc., 3rd edition, 2005.

[157] P. L. Rosin. Thresholding for change detection. *Computer Vision and Image Understanding*, 86(2):79–95, 2002.

[158] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

[159] P. K. Sahoo, S. Soltani, A. K. Wong, and Y. C. Chen. A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing*, 41:233–260, 1988.

[160] S. Santini. Semantic modalities in content-based retrieval. In *IEEE International Conference on Multimedia and Expo*, volume 2, pages 683–686, 2000.

[161] I. Scollar, B. Weidner, and T. Huang. Image enhancement using the median and the interquartile distance. *Computer Vision, Graphics, and Image Processing*, 25(2):236–251, 1984.

[162] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–168, 2004.

[163] C. Shi, K. Yu, J. Li, and S. Li. Automatic image quality improvement for videoconferencing. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2:701–704, 2004.

[164] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[165] Y. Shoham. Nonmonotonic logics: meaning and utility. In *International Joint Conference on Artificial Intelligence*, IJCAI'87, pages 388–393, 1987.

[166] R. Singh, V. Cherkassky, and N. Papanikolopoulos. Self-organizing maps for the skeletonization of sparse shapes. *IEEE Transactions on Neural Networks*, 11(1):241–248, 2000.

[167] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[168] R. W. Smith. Computer processing of line images: A survey. *Pattern Recognition*, 20(1):7–15, 1987.

[169] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision*. Chapman and Hall Computing Series. Chapman & Hall, 1995. ISBN: 978-0-412-45570-4.

[170] P. Soundararajan and S. Sarkar. An in-depth study of graph partitioning measures for perceptual organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):642–660, 2003.

[171] P. N. Suganthan. Shape indexing using self-organizing maps. *IEEE Transactions on Neural Networks*, 13:835–840, 2002.

[172] J. Sun, J. L. Moore, A. Bobick, and J. M. Rehg. Learning visual object categories for robot affordance prediction. *International Journal of Robotics Research*, 29(2-3):174–197, 2010.

[173] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Shape Modeling International*, pages 130–139, 2003.

[174] W. Tao, H. Jin, and Y. Zhang. Color image segmentation based on mean shift and normalized cuts. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 37(5):1382–1389, 2007.

[175] R. W. Taylor, M. Savini, and A. P. Reeves. Fast segmentation of range imagery into planar regions. *Computer Vision, Graphics, and Image Processing*, 45:42–60, 1989.

[176] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Class prediction by nearest shrunken centroids, with applications to dna microarrays. *Statistical Science*, 18(1):104–117, 2003.

[177] M. Tkalcic. Colour spaces - perceptual, historical and applicational background. In *IEEE Region 8 EUROCON 2003: Computer as a Tool*, volume 1, pages 304–308, 2003.

[178] V. Torre and T. A. Poggio. On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):147–163, 1986.

[179] M. Toussaint, V. Willert, J. Eggert, and E. Krner. Motion segmentation using inference in dynamic bayesian networks. In *British Machine Vision Conference*, BMVC'07, pages 4.1–4.10, 2007.

[180] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *IEEE International Conference on Computer Vision*, volume 1 of *ICCV'99*, pages 255–261, 1999.

[181] A. Tremeau and N. Borel. A region growing and merging algorithm to color segmentation. *Pattern Recognition*, 30(7):1191–1203, 1997.

[182] A. Tremeau and P. Colantoni. Regions adjacency graph applied to color image segmentation. *IEEE Transactions on Image Processing*, 9:735–744, 2000.

[183] R. Tron and R. Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'07, pages 1–8, 2007.

[184] Z. Tu. An integrated framework for image segmentation and perceptual grouping. In *IEEE International Conference on Computer Vision*, ICCV'05, pages 670–677, 2005.

[185] Z. Tu, S.-C. Zhu, and H.-Y. Shum. Image segmentation by data driven markov chain monte carlo. In *IEEE International Conference on Computer Vision*, volume 2 of *ICCV'01*, pages 131–138, 2001.

[186] S. E. Umbaugh. *Computer Vision and Image Processing: A Practical Approach Using Cviptools with Cdrom*. Prentice Hall PTR, 1st edition, 1997. ISBN: 0132645998.

[187] E. A. Uriarte and F. D. Martn. Topology preservation in SOM. *International Journal of Applied Mathematics and Computer Sciences*, 1(1):19–22, 2005.

[188] W. R. Uttal, L. Spillmann, F. Strzel, and A. B. Sekuler. Motion and shape in common fate. *Vision Research*, 40(3):301–310, 2000.

[189] M. Uyttendaele, A. Eden, and R. Szeliski. Eliminating ghosting and exposure artifacts in image mosaics. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2 of *CVPR'01*, pages 509–516, 2001.

[190] N. Vasconcelos. From pixels to semantic spaces: Advances in content-based image retrieval. *Computer*, 40(7):20–26, 2007.

[191] R. C. Veltkamp. Shape matching: Similarity measures and algorithms. In *International Conference on Shape Modeling and Applications*, SMI'01, pages 188–197, 2001.

[192] R. C. Veltkamp and M. Hagedoorn. *State of the art in shape matching*, pages 87–119. Springer-Verlag, 2001. ISBN: 1-85233-381-2.

[193] A. Verri and T. Poggio. Motion field and optical flow: qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):490–498, 1989.

[194] R. Vezzani and R. Cucchiara. Video surveillance online repository (ViSOR): An integrated framework. *Multimedia Tools and Applications*, 50(2):359–380, 2010.

[195] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using PowerFactorization and GPCA. *International Journal of Computer Vision*, 79(1):85–105, 2008.

[196] S. Y. Wan and W. E. Higgins. Symmetric region growing. *IEEE Transactions on Image Processing*, 12(9):1007–1015, 2003.

[197] J. Wang, Y. Jia, X.-S. Hua, C. Zhang, and L. Quan. Normalized tree partitioning for image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'08, pages 1–8, 2008.

[198] J. Wang, J. Li, and G. Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:947–963, 2001.

[199] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, 2003.

[200] R. J. Watt and W. A. Phillips. The function of dynamic grouping in vision. *Trends in Cognitive Sciences*, 4(12):447–454, 2000.

[201] J. Weickert. Efficient image segmentation using partial differential equations and morphology. *Pattern Recognition*, 34(9):1813–1824, 1998.

[202] I. Weiss. Invariants for recovering shape from shading. In *Applications of Invariance in Computer Vision*, volume 825 of *Lecture Notes in Computer Science*, pages 185–198, 1994.

[203] M. Willebeek-LeMair and A. P. Reeves. Solving nonuniform problems on SIMD computers: Case study on region growing. *Journal of Parallel and Distributed Computing*, 8:135–149, 1990.

[204] P. Wong and J. Koplowitz. Chain codes and their linear reconstruction filters. *IEEE Transactions on Information Theory*, 38:268–280, 1992.

[205] G. Xiong, D.-J. Lee, S. Fowers, J. Gong, and H. Chen. Using perceptual color contrast for color image processing. In *Advances in Visual Computing*, volume 6455 of *Lecture Notes in Computer Science*, pages 407–416. 2010.

[206] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110:212–225, 2008.

[207] B. Yao, X. Yang, and S.-C. Zhu. Introduction to a large scale general purpose ground truth dataset: methodology, annotation tool, and benchmarks. In *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, EMMCVPR'07, pages 169–183, 2007.

[208] S. X. Yu and J. Shi. Multiclass spectral clustering. In *IEEE International Conference on Computer Vision*, ICCV'03, pages 313–319, 2003.

[209] D. Zhang and G. Lu. Segmentation of moving objects in image sequence: A review. *Circuits, Systems, and Signal Processing*, 20(2):143–183, 2001.

[210] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.

[211] H. Zhang, J. E. Fritts, and S. A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008.

[212] H. Zhang, J. E. Fritts, and S. A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008.

[213] X. Zhang, W. Lin, and P. Xue. Just-noticeable difference estimation with pixels in images. *Journal of Visual Communication and Image Representation*, 19:30–41, 2008.

[214] Y. Zhang, G. Liu, X. Fang, and B. Chen. Medial axis extraction using growing neural gas. In *International Conference on Artificial Intelligence and Computational Intelligence*, volume 2 of *AICI'09*, pages 544–548, 2009.

[215] Y. J. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, 1996.

[216] Y. J. Zhang. A review of recent evaluation methods for image segmentation. In *International Symposium on Signal Processing and its Applications*, ISSPA'01, pages 148–151, 2001.

[217] S. W. Zucker. Region growing: Childhood and adolescence. *Computer Graphics and Image Processing*, 5(3):382–399, 1976.

"Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain."

Alan Turing (British computer scientist, "Computing Machinery and Intelligence", 1912-1954)