OBJECT RECOGNITION AND SEGMENTATION VIA SHAPE MODELS

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

BY

METIN BURAK ALTINOKLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2016

Approval of the thesis:

OBJECT RECOGNITION AND SEGMENTATION VIA SHAPE MODELS

submitted by **METIN BURAK ALTINOKLU** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

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

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. İlkay Ulusoy Parnas
Supervisor, **Electrical and Electronics Eng. Dept., METU** _____

Prof. Dr. Sibel Tarı
Co-supervisor, **Computer Engineering Department, METU** _____

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Dept., METU _____

Assoc. Prof. Dr. İlkay Ulusoy Parnas
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Uğur Halıcı
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering Dept., METU _____

Assist. Prof. Dr. Aykut Erdem
Computer Engineering Dept., Hacettepe University _____

Date: **2. 2. 2016**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: METIN BURAK ALTINOKLU

Signature :

ABSTRACT

OBJECT RECOGNITION AND SEGMENTATION VIA SHAPE MODELS

Altınoklu, Metin Burak
Ph.D., Department of Electrical and Electronics Engineering
Supervisor : Assoc. Prof. Dr. İlkay Ulusoy Parnas
Co-Supervisor : Prof. Dr. Sibel Tarı

February 2016, 84 pages

In this thesis, the problem of object detection, recognition and segmentation in computer vision is addressed with shape based methods. An efficient object detection method based on a sparse skeleton has been proposed. The proposed method is an improved chamfer template matching method for recognition of articulated objects. Using a probabilistic graphical model structure, shape variation is represented in a skeletal shape model, where nodes correspond to parts consisting of lines and edges correspond to pairwise relation between parts. For edge support function of lines, directional chamfer matching cost is calculated. The performance of the new method has been evaluated with experiments using databases especially suitable for shape based object detection methods. The proposed method performs well, and it is much faster as compared to related methods.

Keywords: Object detection, object recognition, object segmentation, skeleton, graphical models, shape matching

ÖZ

ŞEKİL MODELLERİ ARACILIĞIYLA NESNE TANIMA VE BÖLÜTLEME

Altınoklu, Metin Burak
Doktora, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. İlkay Ulusoy Parnas
Ortak Tez Yöneticisi : Prof. Dr. Sibel Tarı

Şubat 2016 , 84 sayfa

Bu tez çalışmasında, bilgisayar görmesi alanında şekil tabanlı olarak nesne bulma, tanıma ve bölütleme problemi çözülmektedir. Seyrek iskelet tabanlı verimli bir nesne tespit yöntemi önerilmiştir. Önerilen yöntem, oluk şablon eşleştirme yönteminin eklemli nesneleri tanımak için geliştirilmişidir. Bir olasılıksal grafik model yapısı ile, şekilde değişikler bir iskeletsel şekil modelinde tanımlanmıştır, düğümler doğrulardan oluşan parçalara, kenarlar da bu parçalar arasındaki ikili ilişkiye karşılık gelmektedir. Doğruların kenar destek fonksiyonu için, yönlü oluk eşleştirme maliyeti hesaplanmaktadır. Bu yeni yöntem şekil tabanlı nesne tanıma yöntemlerine uygun veritabanında yapılan deneylerle değerlendirilmiştir. Önerilen yöntem yeterince başarılıdır ve ilgili diğer yöntemlere göre daha hızlı çözüm bulmaktadır.

Anahtar Kelimeler: Nesne tespiti, nesne tanıma, nesne bölütleme, iskelet, grafiksel modeller, şekil eşleştirme

To my family and people who are reading this thesis

ACKNOWLEDGMENTS

I would like to thank my supervisor Associate Professor İlkay Ulusoy Parnas and co-supervisor Professor Sibel Tarı for their constant support, guidance and friendship. It was a great honor to work with them. I would also like to thank to the committee members for improving the thesis with valuable feedback.

I owe thanks to Yasin Yazıcı for performing experiments with convolutional net.

I would also like to thank to the members of EA 308 laboratory. My sincerest thanks goes to each of my family members for supporting me all the way. Lastly, I would like to thank friends (Vesall, Elif, Seda, Eren) who believed I could achieve my goal.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

AAM	Active Appearance Models
ASM	Active Shape Models
AT	Ambrosio - Torterelli
CCM	Contour Chamfer Matching
CNN	Convolutional Neural Network
CRF	Conditional Markov Random Field
DCM	Directional Chamfer Matching
DCE	Discrete Curve Evolution
DT	distance transform
FFPI	false positives per image
HOG	Histogram of Oriented Gradients
IOU	intersection over union
kAS	k Adjacent Segments
MAP	Maximum a posteriori
MRF	Markov Random Fields
PAS	Pairwise Adjacent Segments
PDE	Partial Differential Equations
PCA	Principal Component Analysis
PGM	Probabilistic Graphical Model
POM	Probabilistic Object Models
RANSAC	Random sample consensus
ROC	receiver operating characteristic
OCM	Oriented Chamfer Matching
SIFT	scale-invariant feature transform
SVM	Support Vector Machine
TSP	Tari-Shah-Pien

CHAPTER 1

INTRODUCTION

One important goal of computer vision is to develop systems that can interpret images. To attain this goal, we need to address two interrelated research problems. In the first problem, known as object detection and recognition, the aim is to determine whether an object of a certain category exists or not in a given image and providing the bounding box of the object. This problem is challenging for real images especially when objects are occluded, of varied intra-class attributes, in varying poses, in cluttered scenes, at different illumination conditions and camera viewpoints. Given that the object exists, the second problem is to delineate the object boundaries, this is known as the object segmentation. Segmentation is also a hard problem since there is not a universal criterion to be applied to every segmentation problem.

Traditionally, the recognition literature has been mostly dominated by appearance based methods, where the cues for detection are texture, color and interest points. However, the appearance based methods have faced limitations, for example, if intra-class variation is more than inter-class variation for exemplars or if key points cannot be reliably extracted. For many natural images, objects of the same class may have a dissimilar appearance, but it is possible to recognize objects by their shape. For many datasets, indeed satisfactory object detection are not achieved unless shape cues are used. In fact, shape is the most distinctive and inherent property of an object class. Besides, shape-based approaches are also used for complementing appearance-based approaches. Until the recent advance, good edge detectors which are vital for achieving successful results with shape based object detection methods were not available. After this advancement in edge detection methods, there has been an increase in num-

ber of shape based object detection papers. Shape based methods either use a hand drawn model, or construct a shape model in the learning stage. Then, objects are detected and segmented by shape matching of the model in the query image.

Despite being an important aspect of computer vision, shape does not have a universal definition. One definition of shape by statistician Kendall is 'all the geometrical information that remains when location, scale and rotational effects are filtered out from an object' [35]. In practical applications, shape is defined in various ways. Boundary based /contour representations, which are commonly used, consider shape as boundary of objects. Boundary based representations include point distribution models, shape context, k-adjacent segments, boundary fragment models. One important property of shape is that it is generally perceived as a whole, rather than being local. Regional representations instead consider shape as a property of the region encircled by the object. One particular regional representation is skeleton (medial axis) which was introduced by Blum [11]. It has been used in shape analysis tasks such as retrieval, matching, classification. Skeleton is defined as the locus of center of maximal inscribed circles of a closed object. It is a compact representation and provides robustness against articulation. However, skeleton based matching algorithms are almost exclusively used for silhouettes/binary images because there is no reliable way of extracting skeletons for cluttered images.

Recently, skeleton has also been applied to the object detection. Trinh and Kimia [67] and Bai et al. [5] developed methods which utilize skeleton as a means of generating shape hypotheses. Then, matching scores are computed by chamfer matching. Chamfer matching is a classical technique of object detection using distance transforms. Another variant of chamfer matching is the method of Nyugen [51]. He defines a small band by adding parallels around the shape template, and formulates object detection using a Markov Random Field (MRF) and uses the variational mean field approximation for the inference.

In this thesis, we mainly deal with shape based object recognition and segmentation. Our main motivation is to bridge the gap between skeleton based shape matching in silhouettes and object detection methods in real images. Even though papers on this application of skeletons are rare, we believe it is very useful to use skeletons for

matching shapes in natural images as it naturally captures the shape structure. As part of the thesis, a new method for object detection has been proposed. This method is based on a sparse skeleton model. There are two main contributions. The first contribution is a skeleton based template generation procedure. From the training images, skeletal parameters are extracted by computing skeletons for all training binary images. A parameter set is extracted from the skeletons, by sampling some skeletal points, this is used as the shape model. Next, a template composed of lines is constructed by connecting the generating points corresponding to the sampled skeletal points. Our second contribution is that we propose a probabilistic graphical model (PGM) based shape matching using this sampled skeleton. As the extraction of skeleton on real images is hard and cannot be reliably used in object detection in real images, we need a probabilistic modeling approach for applying skeletons to shape matching in real images. The shape model is represented as a MRF. We consider the location of skeletal points as hidden nodes. Our interpretation of object detection is to find the maximum a posteriori (MAP) estimation of the associated parameters so that dissimilarity with the query edge image is minimized and the similarity to the shapes in training is maximized. For the image based similarity checking, we adapt the directional chamfer matching (DCM). In addition, we conducted a research on complementing the shape based cue with an appearance cue. For the appearance cue, we trained a convolutional neural network (CNN).

The proposed method is tested on some classes (giraffe and swan classes- these classes contain especially articulated objects) of the ETHZ shape dataset. The performance is superior as compared to similar chamfer template matching approaches. It was demonstrated that our model can handle variations in shape of various exemplars under even large articulations. As compared to previous work, our method has some significant advantages. In Trinh and Kimia's method [67], a probabilistic interpretation is missing. Every, possible state of shape fragments is matched independently and with a uniform prior, however, this causes the shape to vary more than needed, even a shape not belonging to the object category can be obtained. Besides, it explores a very large space and thus it is slow. First of all, our method is faster. Second, we constrain the shape so that no irregular shape is possible unless a cost is paid. Nyugen's model [51] uses only the local variations of a single template and it does

not search the whole shape space of the object class; thus its performance is limited as compared to our method.

The rest of this thesis is organized as follows. In Chapter 2, a literature survey of object detection and recognition methods using several shape-based and appearance based features is provided. In Chapter 3, skeleton extraction and matching techniques are reviewed for mainly silhouette based shape problems. In Chapter 4, boundary detection, skeletal methods for object detection, matching and MRFs have been reviewed. The proposed method is presented in Chapter 5 and the comparison to existing methods is described in details. In chapter 6, properties of databases, evaluation criteria and experimental results are provided. Finally, Chapter 7 concludes the thesis.

CHAPTER 2

RELATED WORK

An object recognition system may use two different types of cues, appearance and shape. In this section, firstly, methods using contours for representing the shape of the object are reviewed. Secondly, some selected appearance based methods are summarized. Finally, some other related methods are examined.

2.1 Shape Based Methods

Object detection methods that mainly relies on shape cues are reviewed here. Some mainly shape based methods also use an appearance based feature. The section is divided into subsections according to different shape features used.

2.1.1 Contour Segments

One particular strategy for contour based detection is using contour segments as the shape feature. An example is k Adjacent Segments (kAS); the segments are formed by short chains of k connected, roughly straight contour segments. Ferrari et al. [28] used this shape feature for object detection, and they also collected a set images for testing (known as ETHZ shape dataset). Over this kAS segments, a contour network is built. Object detection is formulated as finding the paths that resemble the the single hand drawn object model. Their continuing [26] work also uses kAS, but uses a linear support vector machine (SVM) window classifier for detecting objects.

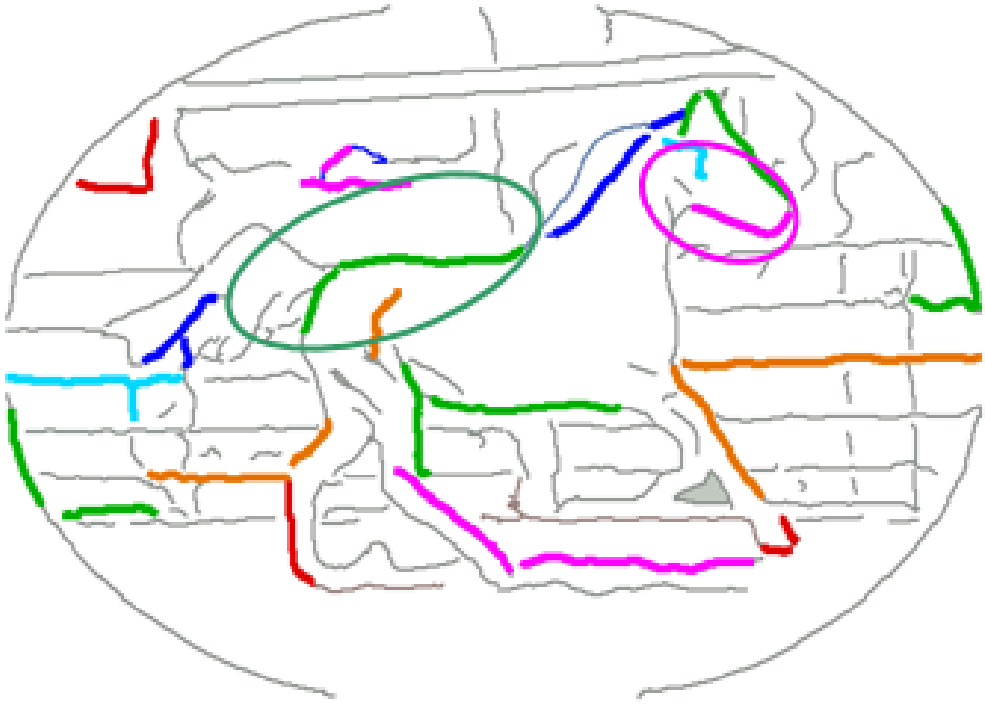


Figure 2.1: Example PAS. (Photo taken from [26])

Ferrari et al. [27] later used pairwise adjacent segments (PAS) (kAS, when k=2, as shown in Figure 2.1) for a statistical shape based object detection method. In training, only bounding boxes of objects are provided. A prototype shape is assembled from the computed PAS. The statistical shape variation is obtained by Principal Component Analysis (PCA) (see Appendix A). In testing, first some hypotheses as rectangles around objects are obtained, when PAS in the shape model to PAS in test images are matched by a 3-dimensional (position, scale) Hough-style voting [43]. Then, the statistical shape model is used for verifying object detection results by deforming the object shape to the boundaries of objects inside rectangles by a modified thin plate spline Robust Point Matching algorithm [18].

A recent work by Guo et al. [33] improved detection performance by introducing importance of parts. A conditional entropy formulation is used to describe the uncertainty in shape reconstruction. Then, a shape part importance measure is defined from the uncertainty. The parts which have lower reconstruction uncertainty is considered of higher importance. Given a small shape part, it is possible to reconstruct the whole shape using this formulation. In object detection, they obtain object candidates as

in Ferrari et al [27], and rank the candidates using a SVM; however, the votes are weighted by calculated part importances.

2.1.2 Template Based Methods

One strategy for object detection is to use complete shape templates. Then, by a template based detection method, the object is located to the location in the query image which provides a good match. Chamfer matching is the classical technique used for measuring the matching quality between the model and query images in template matching. The fastest chamfer matching method is DCM [45] which is based on representation of edge and query with lines. In the template matching methods, generally, one hand drawn shape example per class is used. Chamfer matching is examined in details in Chapter 4.

2.1.3 Boundary Fragments

A different idea is to use boundary fragments in the shape model, as in the similar works of Opelt et al. [53] and Shotton et al. [59]. Fragments are group of edges, they are extracted from training images. Using these fragments, a shape codebook is constructed, the location of the object centroid is also stored. Then, a category specific object detector is learned using a boosting algorithm. The object is detected by calculating the classifier's output for an object centroid. Shotton uses a star shape constellation model for handling the geometric variation in an object class, while Opelt uses a Hough voting method for detection. Fragment to query edge matching is measured with the proposed oriented chamfer matching (OCM) in Shotton's work, and with chamfer matching in Opelt's work.

Later, Opelt et al. [52] improved their shape based object categorization and detection method by adding an appearance cue. So, a codebook is constructed with two kinds of alphabet entries which are boundary-fragments (basic shape feature) and patches (appearance feature). Similarly, Shotton et al. [60] improved their initial contour-based work; by introducing a new multiscale OCM technique for matching contour fragments, and adding a step of bootstrapping technique to the training. Chia et al.

[16] used a similar approach, where shape codebooks are constructed using extracting line segments and ellipses.

Yarlagadda [72] formulated object detection as the proper placement of model contours in the image. A dictionary of model contours are learned in training. Contours are found by checking points of high curvature in normalized bounding boxes. Contours are clustered by calculating a dissimilarity matrix and using Ward's method. Entries of the matrix consist of DCM cost, location of the match, and shift from contour centroid to the bounding box's center. The best joint placement of contours are found by multiple-instance learning SVMs, using both positive and negative bounding boxes. In testing, contours in the codebook are matched to the edge map by DCM and a set of scores are obtained. A joint placement of contours that find the best spatial consistency and DCM cost is found by optimization.

2.1.4 Active Shape and Appearance Models

An active shape model (ASM) [21] is a PCA based statistical shape analysis technique. (Active appearance models (AAM) [19] are similar, but includes appearance (texture) information.) The primary technique used in these model is a variant of Procrustes analysis [36]. The shape model is obtained at the training phase by learning a point distribution model of a finite set of 'landmark points' in the shape boundary. All points are aligned and dimension reduction is performed on the aligned point set by PCA. The landmark points are then represented/approximated using eigenvalues and shape coefficients. The model is used for locating the object in a test image by matching the appearance of it to objects in the training set. In the test phase, active shape model is constrained to stay close to the shapes in the model. Specifically, the ASM works in these two steps, in an iterative procedure: It generates a shape by determining the best position of the points. The points are then moved so that the suggested shape is constrained to the point distribution model.

Kokkinos and Maragos [39] presented an detection and segmentation method, using AAM [20] for object modeling, expectation maximization for inference. Shape and texture basis elements are acquired in the learning step. Then, EM algorithm is used in testing. In the E-step (the segmentation step), image observations are assigned

to object hypothesis, either calculating foreground probabilities in regions or using a front propagation. In the M-step (the recognition step), shape and appearance object models are fitted to observations by solving shape fitting equations (obtained by taking derivatives with respect to shape and texture). A log-likelihood is used for probabilistic modeling the object, while the background is represented by a Gaussian model as a piecewise constant image corrupted by noise.

2.1.5 Long Contours / Tokens

Many shape-based object detection method relies on extracted long salient contours. One example is Zhu et al. [74]'s method which seeks a correspondence of control points sampled both from the model and the long contours. They define control selection variables such that a contour is either selected or not. Then, a set-to-set contour matching is performed by minimizing cost function over the correspondences and contour selection. The final optimization problem is relaxed to be a linear programming problem.

Felzenszwalb and Schwartz [25] proposed an approach based on a hierarchical structure called shape-tree. For each open curve specified by a set of points, the mid-point is used to divide the curve into two curves. At detection stage, firstly, the salient curves are found from the soft edge map. Then, image contours are matched to model contours hierarchically using dynamic programming, and the object is detected as composition of these matches by using another dynamic programming.

Payet and Todorovic [56] also uses long, open contours for object detection. Their method is based on the observation that clutter object boundaries will be similar in layout and shape despite boundaries change from instance to instance. The contours are represented by their Beam Angle Histogram descriptor. Then, a graph of matching contours is built using this shape descriptor. The object is detected by finding the MAP multi-coloring assignment of the graph.

Srinivasan [62] develops a partial matching approach, using long, salient, bottom-up image contours. A many-to-one matching of object contours to learned model parts is suggested. The many-to-one matching scores are tuned by a latent SVM, using a

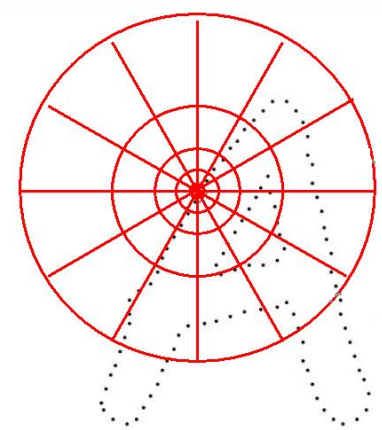


Figure 2.2: Shape context example. (Taken from [7])

max-margin setting. Object contours and part locations, as well as object locations are to be inferred. The optimization is formulated as a linear programming. It takes minutes per image to perform the object detection task.

Kokkinos and Yuille [40] proposed an hierarchical graphical shape model; the elementary unit is image tokens (edge segments and ridges), the higher levels are contours, parts, and the object (in this order). Given the bounding boxes, a shape model is learned by AAMs and clustering of parts. The object detection is formulated as the search of the assignment of tokens to the object contours, the pose (a state variable consisting of position and scale) of nodes, and determining existing nodes. A cost function is defined as the combination of an object configuration term and an image fidelity term comparing model and image contours; and the lowest value of the cost function is found using an A* search algorithm.

2.1.6 Shape Context

The shape context is defined as a shape matching feature for a shape representation using a point set. The vectors from a particular point to remaining points are considered. This particular point is seen as the center of a circle, then, the shape context is simply the log-polar histogram of these vectors (see Figure 2.2). The angle is quantized into 12 bins, and log-distance is quantized into 5 bins. This descriptor has been invented and successfully used in shape matching by Belongie et al. [8].

Bai et al. [4] has proposed a two step object detection method. In the final step, the shape context is used for measuring the similarity between the formed shape and the template. At the initial step, the image is scanned with shape bands. A shape band is nothing but a search window around the object, corresponding to local different deformations of the template. Inside a shape band, using gradients at each point on the contour as the features, candidate object positions are obtained that will be input to the final step.

2.1.7 Landmark Points / Sampled Points

The shape can be represented by landmarks around the contour. Heitz et al. [34] has developed a probabilistic method for descriptive classification using landmark positions learned by an automatic landmark selection process. In the object representation, a conditional Markov Random Field (CRF) defined over set of landmarks is used. The associated cost function consists of 3 terms, object shape model, landmark detector features, and the gradient feature. The object shape model is represented by a Gaussian of landmark locations obtained from the training set. The landmark detector features, namely, boundary fragments, shape templates, filter response patches, and scale-invariant feature transform (SIFT) descriptors are constructed using a strong boosted classifier. This method has also been unified with the appearance based Texton Boost method by Packer et al. [54] for figure-ground segmentation (object localization and segmentation) in cluttered natural scenes.

Wang [69] proposed a flexible shape model known as the fan shape model. Given a training shape, a root node (point) is selected, and the contour is sampled with equidistant points. The correspondences between points in different training shapes are found via a dynamic programming approach. The probability of an edge point belonging to a ray is calculated based on its tangential angle, inclined angle, distance to the root node, and the neighbor classifier using SIFT features. Firstly, given a set of discrete scales, the scale that yields the maximum value of product of maximum probabilities of edges belonging to a ray is found. Then, the edge point that yields the maximum probability of these 4 features is found for each point inside a ray. For the final score, this probability is multiplied by a term penalizing matches due

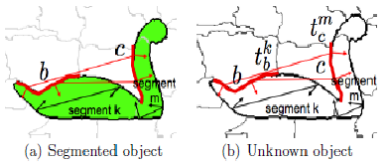
to clutter, and with another term measuring the distance between points (distance between points should be equal as in the model).

2.1.8 PDE equations

Gorelick and Basri [31] proposed an object segmentation method via grouping some superpixels as foreground. Firstly, a hierarchical segmentation of the image is obtained and each superpixel is matched to a database of object silhouettes exemplars based on the regional and contour shape information. The regional shape descriptor is a partial differential equation, called the Poisson equation (reviewed in Chapter 3). The boundary information is compared by chamfer distance, while the regional information (local shape orientation) is compared by a regional dissimilarity measure. By using a PGM and a Hough voting scheme, the probability of observing an object of a particular class at a particular location is obtained and the object is delineated by labeling a group of superpixels as foreground.

2.1.9 Chordiogram

Toshev et al. [65] defined a global shape descriptor using chords. A chord is a pair of boundary edges on object outlines. Geometric features obtained from chords are quantized into bins, and its K dimensional histogram called the chordiogram is the shape descriptor (shown in Figure 2.3). In testing, images are segmented into several superpixels. The object is assembled by grouping of superpixels that corresponds to the minimum of a cost function. The cost function includes a term for measuring the distance between the object model and the selected object, a term measuring the coherency of the appearance of the selected object, and a boundary grouping term (penalizing low values of Pb edge detector). The problem is a type of NP-hard integer quadratic program, however, a relaxation to linear constraints that leads to optimization by semi definite programming is possible. In addition, in a reranking procedure, where the top detection is used as the positive example, chordiograms are used to train one-vs-all SVMs for each category.



(a) Segmented object (b) Unknown object

Figure 2.3: Chordiogram descriptor. (Drawing taken from [65])

2.1.10 Skeleton Based Methods

Using more than one template is needed in the case of a large intra-class shape variation, as the single template has a limited capacity for representing the shape variation. One way of generating shapes is using a skeletal model. Active skeleton of Bai et al. [5] constructs a tree union of skeletons, and it searches for the contour that best aligns to the edges at each part of the tree. Initially, an OCM cost is calculated for measuring the similarity of image edges and parts, and the branch to skeleton branch cost is added. After the tree union-structure is searched by the max-sum algorithm, the best location is found via changing the location by mean-shift algorithm, and rotating the object slightly. Trinh and Kimia [67] proposed a skeleton based object search algorithm. Possible objects are hypothesized by a generative shape model, obtained from their shock graphs. The object search is expressed as finding the optimal parameters of the model via a dynamic programming. The support of edges to the model template is measured by the Contour/Partitioned chamfer matching (CCM) cost. These skeleton based methods and CCM method have been examined in details in Chapter 4.

Adluru and Latecki [1], uses skeleton in a different framework, for simultaneously estimating skeletons and the contour of the object. Firstly, edges are detected and linked, then edge links are split into scale adaptive edge segments. The skeleton consists only of a single branch. In addition, the skeleton is not defined with circles, rather it is defined with ellipses, because the problem does not permit the application of the latter one. At the final step, boundary and skeleton of the object of interest are grouped jointly using these edge segments in a probabilistic framework using a particle filter.

There has been an attempt by Ozcanli and Kimia [73] to use shock graphs as shape features for object recognition in real images. The method is based on a bag-of-

visual fragments approach. Shock patch fragment is a patch of the image with shock subgraph model. After computing shock patch fragments in real images, the shock patches are matched to the model, using edit-distance algorithm based on shock transitions. This is particularly useful since the structure captures the articulation in the shape.

2.2 Appearance Based Methods

Key-points are widely used in object recognition. A well-known key-point descriptor is Lowe's SIFT [46]. This approach starts with scale-space extrema detection step, in which keypoints are extracted by successive difference of Gaussian-blurred images. Then, the points with low contrast or along an edge are discarded. Based on image gradients, dominant orientations are assigned to keypoints. Thus, the most stable points for matching and recognition are extracted. A histogram of the sub-block around the keypoint is used as the keypoint descriptor.

The Texton Boost method developed by Shotton et al.[61] performs well even in highly textured or textureless objects, as it is based on dense features rather than sparse features. In this method, a new discriminative model for automatic semantic labeling (segmentation) of photographs is introduced using a new feature called the texture-layout filter. This filter jointly captures texture (based on texton, obtained after applying a set of seventeen filters including Gaussian, derivative of Gaussians and Laplacian of Gaussians with different scales), layout and context information. Features such as color, location, texture and edge potentials are combined to a single model through a Conditional Random Field (CRF).

Berg et al. [9], proposes a shape matching approach, where the shape of an object is represented by a set of points sampled from the contour. He uses a local appearance detector named as the geometric blur (smoothed signal around a feature point) for matching feature points of the image and the model. This shape feature captures smooth portions of object contours. The similarity of the correspondence points and the spatial arrangement of feature points is measured. An integer quadratic programming problem is solved for the minimization of the associated cost function.

The geometric blur is also used in Maji and Malik's work [49]. In training, a codebook of features is extracted. In testing, local features are extracted and matched to codebook entries. Then, Hough transform voting of implicit shape model framework of [43] is used for detection: Weighted votes are casted for object centers, based on the learned distributions. The weights used in the matching are learned in a discriminative setting in a max-margin framework.

2.3 Some Other Methods

There are, of course, many other object detection approaches in the literature. For example, Carreira et al. [13] proposed a united segmentation-detection approach for figure-ground assignment using both appearance and shape cues. This method uses SIFT and three pyramid histogram of oriented gradients (HOGs). Firstly, multiple figure-ground segmentations of an image are obtained by a segmentation algorithm called the constrained parametric min cuts algorithm [14], where a sequence of constrained parametric min-cut problems are solved. Then, the quality of being a member of a particular class for each segment is measured for separate classes. Quality functions in these two steps are obtained using a SVM approach.

Chen et al. [15] proposes an object classification, segmentation, and recognition method by learning Probabilistic Object Models (POMs), POM interest points, POM masks and POM-edgelets. The task of learning POMs is a structure induction problem. A knowledge propagation strategy is used to send information from POMs to otherPOMs. The shape of the object is represented with the combination of POM interest points and POM-mask.

CHAPTER 3

SKELETAL SHAPE REPRESENTATIONS

Skeletal shape representations are widely used in shape analysis tasks such as binary shape recognition, retrieval, matching, classification. This chapter reviews them.

3.1 Medial Axis

The medial axis was proposed by Blum [11] for biological shape recognition. A skeleton (medial axis) can be defined in a few different but equivalent ways. In the first definition, an analogy of grassfire starting at boundaries and propagating isotropically is used. The points where grassfires meet are equidistant to the boundary, these points are called sym-points (skeletal points). Skeleton can also be formulated via a maximal disc. A maximal disc is a disc contained in the shape that cannot be contained with another disc inside the shape. Then, the sym-points lie at the centers of maximal discs.

More notions about the geometry of a medial axis can be defined. A pan-normal is the shortest path from the object's boundary to a sym-point. At sym-points, the length of all pan-normals originating from distinct boundary points are equal. This equal length is called sym-dist. Sym-dist can also be seen as the radius of maximal disc centered at the skeletal point. Sym-axis (sym-ax) is the locus of sym-points. If the input shape is a circle, sym axis will be a point; for a rectangle as the input shape, the sym-axis is shown in Figure 3.1. The sym-ax and the sym-dist constitute the sym-function, which is the sym-transformation of the shape. The inverse transform of a sym-transformation is possible, this is known as the shape reconstruction. The object

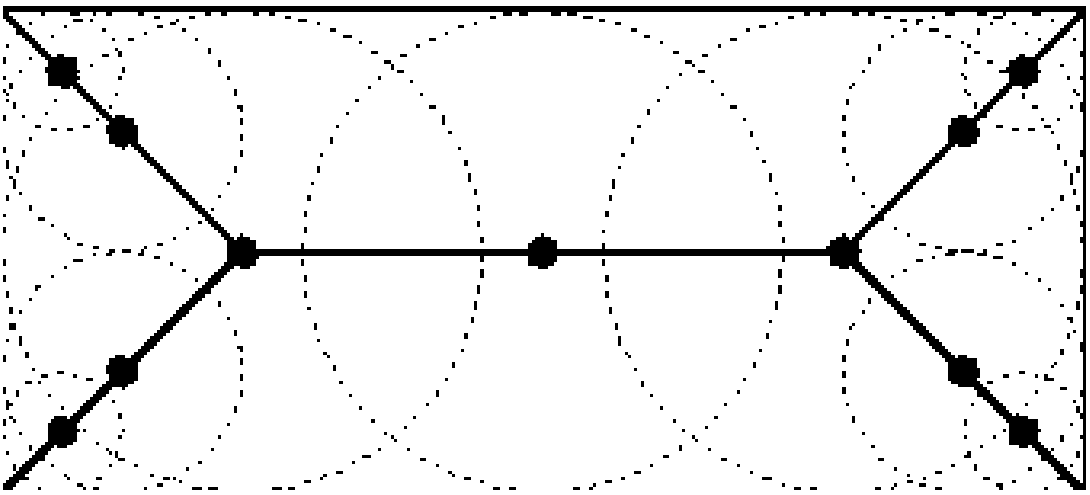


Figure 3.1: Medial axis representation for a rectangle (taken from [29]).

shape is constructed by the union of maximal discs of sym-dist placed at each sym-point of the sym-ax. A sym-point that has n sym-ax in the neighbourhood is called a n-sym point. Angles between tangents to the parallels (the boundary of maximal discs) are sym-ax angles. Interval angle is the angle between pannormals. The contact angle is the angle equal to 360 degrees minus the interval angle. For the case of 2 sym-points, we define the positive sym-ax angle as the object angle.

Many different approaches for the computation of the skeleton have been developed. Unfortunately, there is an instability problem of the skeletons: The skeleton topology changes with respect to small changes at object boundaries caused by noise or rotations. However, for proper application of skeletons to shape analysis, the stability should be ensured.

3.2 Bai's Skeleton

Bai [2] proposed a skeleton pruning method as a solution to the instability problem. This method grows the skeleton according to a contour partitioning algorithm, so that unnecessary details in the skeleton caused by small changes in object contour are removed. The contour partitioning algorithm divides the contour into parts by approximating the shape as a polygon, the higher the number of partitioning points in the algorithm, better the polygon approximates the shape. The partitioning algorithm

Figure 3.2: Skeleton extraction with respect to the polygonal approximation contour segments obtained by the DCE. (Illustration is taken from [2])

used in their paper is Discrete Curve Evolution (DCE). The particular skeletonization algorithm used in the skeleton pruning method extracts the skeleton by a connectivity criterion [17]. The algorithm starts at the maximum value of the distance transform of the shape and grows in either direction until reaching the end points of the shape. The starting point is considered as the root point of the skeleton. For detecting a point as a skeletal point, it should satisfy the condition of being a ridge point geometrically. In addition, Bai forces the sym-points to have corresponding boundary touching points from different partitions of the curve based on the partitioning algorithm's output. With this modification, the algorithm does not produce skeletons with spurious branches. An illustration of this is supplied in their paper (see Figure 3.2), note that by using less points in the polygonal approximation, less details remain in the final skeleton.

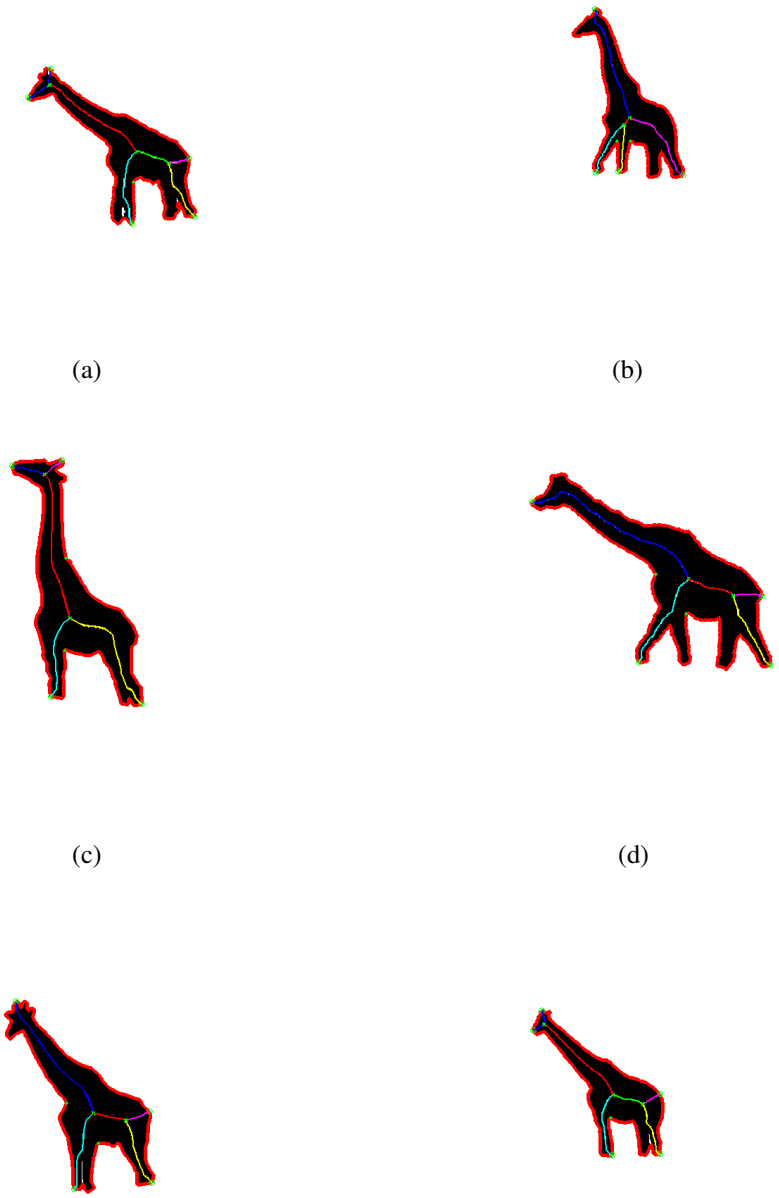
For some images in the giraffe class, skeletons are computed and end point-junction point connections are obtained by computing Bai's skeleton. The path (list of skeletal points) between two endpoints is calculated by depth-fist search search by the code of authors, however it is rather slow. Instead, bwtraceboundary of MATLAB gives the point list in the skeleton between input points, in a faster computational time if end

and junction points are supplied. From skeletons, end and junction points are found by checking 3 by 3 neighbourhoods of each point by findendjunctions.m (obtained from Kovesi image processing functions [42]). A skeleton path is defined as the shortest path (skeletal points) between two endpoints. We trace skeletal paths between these points by depth-first-search algorithm (using Bai's code).

Some computed skeletons are shown in Figure 3.3. Each path is plotted in a different color, while the DCE vertices are shown by green markers. The examples are from the giraffe class. Note that the skeleton topology is not fixed for the same class; the number of branches and junction points vary in different exemplars. For some shapes, there are two branches in head, for others there is only one (compare shape (a) and shape (e)). We see that in shape (a), there are 7 branches at total, and for shape (c) there are only 5 branches (in c the branch between the center and the right of the body is missing and the branch from the center to the right leg is not separated into 2 branches). Number of junction points are 2 in shape (c) and 3 in shape (a). However, we observe that a basic topology is preserved in all shapes: one branch at the neck, one branch for the leg, one branch for the other leg, and one branch for the center of the shape.

3.3 MAP skeleton

Skeleton extraction has been explored through many different techniques, but these techniques are generally deterministic. In the deterministic approaches, the skeleton is affected from the noise at the boundary so it is not easy to capture the true structure of object parts. Feldman and Singh devised a probabilistic skeleton extraction, named as the MAP skeleton [24]. They define the skeleton extraction as a Bayesian estimation problem, probability of a skeletal description is expressed by Bayes' rule. The posterior probability is the product of the likelihood term (how likely each shape given a skeleton, expressed as the product of likelihoods associated with all skeletal points) and a prior term which discourages too many details such as curvy branches. The skeletal parameters, radius of maximal discs and rib direction (the direction of the vector from the generating points to skeletal points) are modeled with Gaussian and Von Moses distributions, respectively. The skeletal description with the least negative



(a) (b)

(c) (d)

(e) (f)

Figure 3.3: Computed skeleton and paths in giraffe class shapes.

logarithm of the posterior probability is computed via an expectation -maximization like algorithm. Initially, the skeleton is computed using the classical Voronoi-based medial axis transform. Some axes are pruned (if it can pass a Bayesian test of importance). Then, it is organized it into a hierarchical structure. Skeletons are estimated by repeating these steps: Each boundary point is assigned to the axis point with the highest likelihood term. One step is taken down towards the gradient of negative logarithm of the posterior probability.

3.4 Shock Graph

A shock graph (Kimia et al. [38]) is an abstraction of a skeleton to a directed graph. The graph consists of shock nodes and shock edges (shock segments) between them. Shocks (singularities) are formed during propagation of grassfire from boundaries. Skeletal points are grouped according to the direction of the increase in the radius function (or equivalently velocity of flow of grassfire waves). The Figure 3.4 shows a shock graph example (from [30]). There are 6 types of shock points, 5 shock point types are considered as shock nodes: 1. the point where the medial axis has an end point, 2. a type of junction point (two flows enter and one flow quit), 3. another type of junction point (three branches enter the points), 4. a typical smooth mid-segment points of the medial axis, for which the radius of the maximum disc has either maximum, 5. a smooth mid-segment point same where the associated radius is minimum. Finally, there is one more typical smooth mid-segment shock point type, for which the radius of the maximum disc has no extremum, these points form shock edges that connect shock nodes.

An available code for computing shock graphs is Macri's code C++ code [48]. The code provides 2 options for computing skeletons, either the flux-based skeletons as formulated by Dimitrov et al. [23] or A* Fast Marching Method skeletons developed by Telea [64], and these skeletons are converted to shock graphs. Figure 3.5 shows computed shock graphs of shapes in 5 ETHZ object categories using this code.

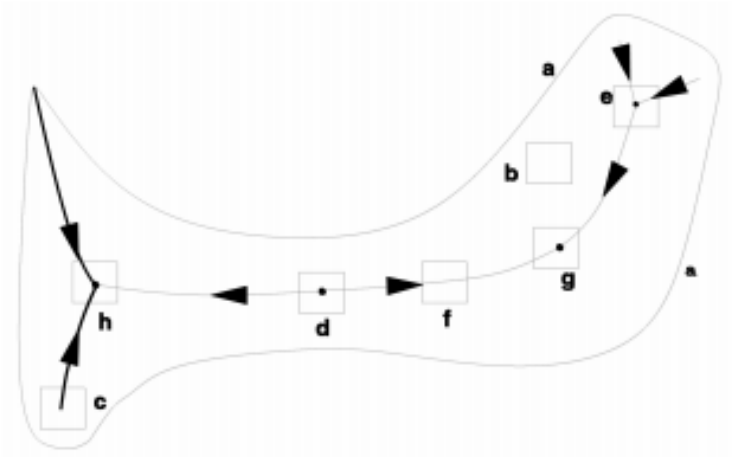


Figure 3.4: A shock formation example (from [30]). For shape (a), these are the shock types: (c) is the end point of the medial axis, (d) is a typical point point which correspond to a local minimum of radius, (f) is a typical point for which the radius of the maximum disc has no extremum, (g) is a typical point for which the radius of the maximum disc has a local maximum, point (e) and (h) are junction points.

3.5 Variational and PDE-Methods

In variational methods, we start an energy functional, the associated Euler-Lagrange equations are found with the calculus of variations. The minimum of the energy functional is found by gradient descent by introducing an artificial time parameter, that yields a PDE equation. The Partial Differential Equations (PDE)-based methods directly start at a PDE equation. PDE equations are solved by a numerical scheme. These methods are applied to shape analysis of binary images as shape representations.

3.5.1 Ambrosio - Torterelli functional and Tari-Shah-Pien Method

Mumford and Shah's image segmentation functional (given in equation 3.1) [50] searches for the optimal piecewise smooth approximation of the image (u) and an edge set (Γ), given an input image u_0. The functional consists of three terms, a data fidelity term, a term that measures the piecewise smoothness of u in each Ω_i, a penalty term for the length of edge set $|\Gamma|$. Omitting any of these 3 terms yields trivial solu-

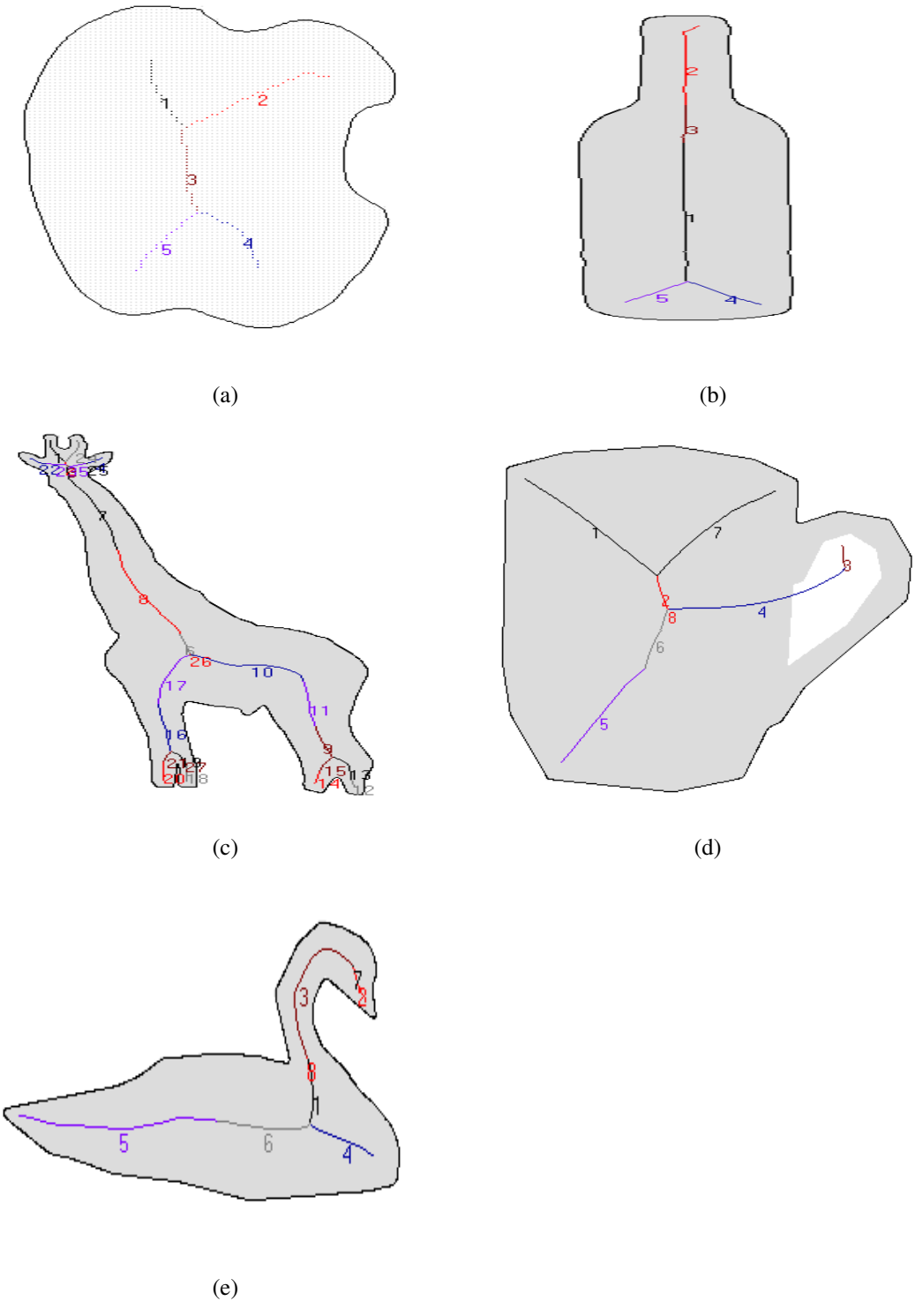


Figure 3.5: Shock graph examples for different object classes, in this order, apple-logo, bottle, giraffe, mug, swan.

tions; only if all three terms are kept, the image segmentation problem can be solved.

$$E_{MS}(u,\Gamma) = \beta \int_{\Omega} (u-u_0)^2 dxdy + \alpha \int_{\Omega\setminus\Gamma} |\nabla u|^2 dxdy + \nu|\Gamma| \tag{3.1}$$

The model can also be seen as a two parameter model, where $1/\sqrt{2\alpha\rho}$ is the contrast threshold (a threshold of the gradient value at the boundary between noise and real edges), α/β is the smoothing scale (a parameter for choosing the region size affected by smoothing).

The MS functional has two unknowns of different natures (a function and a set), hence it is impossible to minimize it in the standard way of calculus of variations. Calculus of variations can be applied only after approximation by a regular functional in gamma - convergence framework. The most common approximation is Ambrosio - Torterelli (AT) functional (see 3.2) which γ-converges to the MS functional as ρ goes to zero.

$$E_{AT}(u,v) = \int_{\Omega} \left(\beta \left(u-u_0\right)^2 + \alpha \left(v^2 |\nabla u|^2\right) + \frac{1}{2}\left(\rho|\nabla v|^2 + \frac{(1-v)^2}{\rho}\right)\right) dxdy \tag{3.2}$$

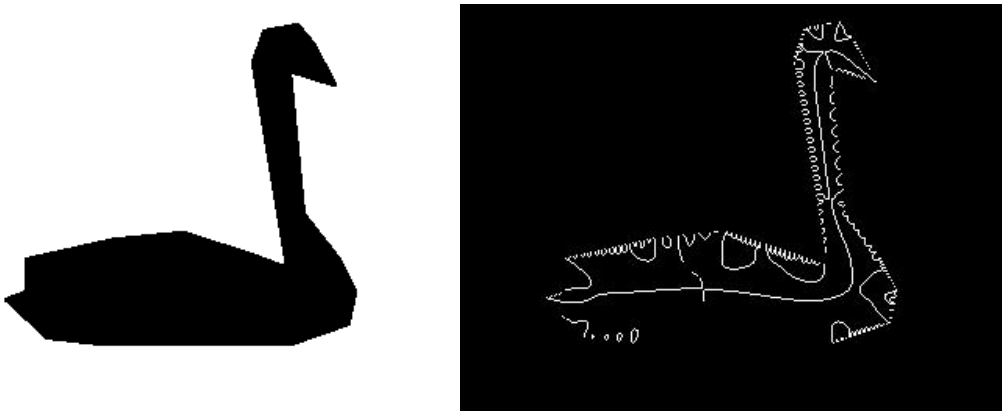
Besides the application in the segmentation problem, the AT model can be used for representing shapes. Indeed, a shape representation in the form of a PDE equation is used in the Tari-Shah-Pien (TSP) method [63].

$$\int_{\Omega} \frac{1}{2}\left(\rho|\nabla v|^2 + \frac{(v)^2}{\rho}\right) dxdy \tag{3.3}$$

$$\nabla v^2 - \frac{v}{\rho^2} = 1 \tag{3.4a}$$

$$v\,|_{\beta} = 0 \tag{3.4b}$$

The main idea of TSP method is using AT PDE equations with a large value of ρ, in this case, v is seen as a measure of the pixel belonging to the boundary. Thus, AT



(a) (b)

Figure 3.6: For an input swan image shown in (a), (b) presents computed skeleton (ρ is set to 100)

model can be also used for extraction of skeletons in gray-scales images. Indeed, we can extract the skeleton as the zero-crossings of $d|\nabla v|/ds = v_{\eta\xi}$ by a zero-crossing detector where

$$v_{\eta\xi} = \frac{\left\{\left(v_y^2 - v_x^2\right) v_{xy} - v_x v_y \left(v_{yy} - v_{xx}\right)\right\}}{|\nabla v|^2}. \tag{3.5}$$

The variables in $v_{\eta\xi}$ are discretized with central differences as follows:

$$v_x^n = \frac{v_{i,j+1}^n - v_{i,j-1}^n}{2} \tag{3.6a}$$

$$v_y^n = \frac{v_{i+1,j}^n - v_{i-1,j}^n}{2}, \tag{3.6b}$$

$$v_{xx}^n = \frac{v_{i,j-1}^n - 2v_{i,j}^n + v_{i,j+1}^n}{2} \tag{3.6c}$$

$$v_{yy}^n = \frac{v_{i+1,j}^n - 2v_{i,j}^n + v_{i-1,j}^n}{2} \tag{3.6d}$$

$$v_{xy}^n = \frac{v_{i+1,j+1}^n + v_{i-1,j-1}^n + v_{i+1,j-1}^n + v_{i-1,j+1}^n}{4} \tag{3.6e}$$

$$|\nabla v|^2 = v_{xx}^{n2} + v_{yy}^{n2} \tag{3.6f}$$

In Figure 3.6 is shown the obtained skeleton for an input binary swan image.

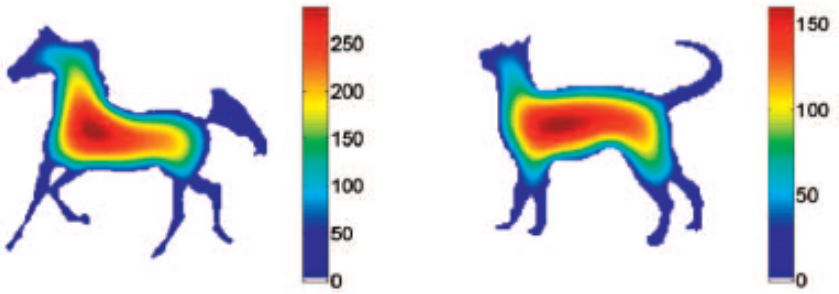


Figure 3.7: Solutions of the Gorelick's PDE for two exemplar shapes. (Taken from [32])

3.5.2 Poisson PDE equations

Gorelick's [32] Poisson PDE equations are a special form of TSP PDE equations. It is used in shape characterization of segmented complete silhouettes. For a silhouette, a particle is considered to be in a random walk starting at a internal point and reaching the boundary. Let S denote the shape and U denote the particular measure. The value of U is found by solving the PDE equations (see equation 4.2). High values of U are observed towards the center of the shape, while U values are relatively lower for the limbs, head, and tail parts, as can be seen in Figure 3.7. Thus, the U function reflect the global properties of the object shape.

$$\triangle U = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = -1 \tag{3.7}$$

$$U\left(x, y\right) |_{\partial S} = 0 \tag{3.8}$$

3.6 Skeleton-based Shape Matching

Skeleton based shape matching are used for classifying shapes (in the form of silhouettes). Sebastian [58] uses an edit-distance algorithm for matching skeletons for shock graph structures. For matching of a query shock graph to a template shock graph, the cost of edit operations (splice, contract, deform) are computed. Deform edit cost

matches two shock segments by finding the optimal alignment by comparing length and boundary curvature differences of associated boundary segments, width and relative orientation differences.

Bai and Latecki [3]'s path similarity based shape matching method can deal with the problem caused by changing topologies in the same class. The matching is based on minimal paths. The graph matching algorithm computes distance between two paths from two different skeletons by comparing normalized radius and length differences of sampled points in the path. Using a geodesic path based matching (rather than using a topological graph comparison approach) provides a stable matching method. Radius information of sampled skeletal points is taken as Euclidean distance transform (DT) values at those points and it is normalized by dividing each radius value to the sum of all DT values of points in the shape. The shape dissimilarity between two paths of M equidistant points, with normalized radius sequences r, p and normalized lengths l, k, is expressed as

$$sd = \sum_{i=1}^{M} \frac{(r_{i-}p_i)^2}{r_i + p_i} + \alpha \frac{(l-k)^2}{l+k}. \tag{3.9}$$

An important aspect of the method is finding correspondence between end points. Suppose we are matching two skeletons, query and target skeletons, A and B. Also suppose that for skeleton A, there are 3 end points, $A1$, $A2$, $A3$ and there are 3 end points of B, $B1$, $B2$, $B3$. As we do not know the correspondences of these critical points, for end point $A1$, the path between $A1$-$A2$, and $A1$-$A3$ can correspond to $B1$-$B2$, $B1$-$B3$, or $B1$-$B3$, $B1$-$B2$ in skeleton B. The correspondences between the points can be found by an assignment algorithm known as optimal subsequence bijection. Either the average of $A1$-A2, $B1$-$B2$ or $A1$- $A3$, $B1$-$B3$ (the case with the lower cost is chosen here) differences yield the dissimilarity value for $A1$-$B1$ path. After calculating dissimilarity values associated with each path, a dissimilarity matrix is obtained, and the dissimilarity value between two shapes is found by solving the assignment problem by a combinatorial optimization algorithm, the Hungarian algorithm.

CHAPTER 4

OBJECT CLASS DETECTION BACKGROUND

An object detection system consists of a boundary detection method, a shape model, and a matching method. In Section 4.1, we review some boundary detection methods. In Section 4.2, we examine generative shape models based on skeletons. Finally, we deal with matching; in Section 4.3, an overview of template matching methods, and in Section 4.4, an overview of MRFs is provided.

4.1 Boundary Detection

Boundary detection is a fundamental task in low level computer vision. Traditionally, boundary detection algorithms were failed in cluttered images. Recently, some new methods has reported improving boundary detection in cluttered and texture images towards human-level performance.

4.1.1 Canny Edge Detector

Canny edge detector [12] is a successful multistage algorithm for finding the edges of objects especially for images with solid regions. The image is smoothed with a Gaussian filter in the reprocessing step. Then, intensity gradients which will be used as edge strength are computed. Canny edge detector does not merely threshold the gradient values, rather spurious edges are removed by non-maximum suppression with a double threshold operation. Strong edges are those edges that have higher gradient value than the second threshold, weak edges are those edges that have gradient value

between thresholds, remaining edges are suppressed. The final edge set is obtained via hysteresis thresholding in which weak edges that are not connected to strong edges are removed.

4.1.2 Berkeley Edge Detector

For textured images, classical edge detection methods have limited success. We need more powerful tools than the tools which simply use the image gradient, because textured regions have patterns which behave like edges for image gradient based edge detection algorithms, but in fact they are repeated texture structures. One such powerful detection method is the Berkeley/Pb edge detector of Martin et al. [22]. Their formulation is basically based on a multichannel discontinuity checking on the local neighborhood of each pixel. Channels consist of two brightness features, one color feature and one texture feature (texton). The first brightness feature is the oriented energy, which is used to detect composite edges. Let f^e and f^o denote a quadrature pair of even and odd-symmetric filters at orientation and scale, a Gaussian second-derivative and its Hilbert transform is used for even and odd filters, respectively; then oriented energy is computed as a filtering operation:

$$OE_{\theta,\sigma} = \left(I * f^e_{\theta,\sigma}\right) + \left(I * f^o_{\theta,\sigma}\right) \tag{4.1}$$

Other measures are brightness, color, and texture gradients. For these measures, local discontinuities are computed by comparing contents of two disc halves of a certain radius, and orientation centered at each pixel. The discontinuity measure is a Chi-Squared distance of histograms (given in equation 4.2) of all three different features, in these discs. The method works in a L*a*b colorspace. Histogram of L* values are used as the brightness feature while histogram of a*b values are used (histograms are in 2-D space in color case) for the color feature. The texture gradient is computed as the response to a 13-element filter bank, 6 pairs of even and odd filters (here we use the filters in oriented energy computations at different orientations), and a center surround filter (difference of Gaussians). Then, the data vector is input to k-means algorithm, and filter responses are clustered. Textons are centers of these clusters.

Oriented energy, brightness gradient, color gradient, and texture gradient complete the final set of features, each at eight orientation and three half-octave scales. The system will provide edge detection probabilities by combining these cues. The parameters in these features, like scale and orientation are optimized by applying coordinate-ascent on each cue for good quality precision-recall curves as the objective. The cues are finally combined in a supervised learning framework, in which a classifier is trained with human-marked segmentations. Different classifiers such as classification trees, density estimation, logistic regression, hierarchical mixture of experts, and support vector machines are used in the learning task.

$$\chi^2(g,h) = \frac{1}{2}\sum \frac{(g_i - h_i)^2}{g_i + h_i} \tag{4.2}$$

4.1.3 Generalized Boundary Edge Detector

Generalized Boundary detector [44] is another successful edge detection algorithm for textured images. Boundaries at different layers of images are found, from low to mid level layers. Using more than layer is advantageous because one layer could extract boundaries for only some images, but for others other layers may be useful. Boundaries in one layer may not coincide with boundaries in other layers, thus it is necessary to combine them. Specifically, color, soft-segmentation and optical flow are used in layers. The outputs of all layers are appropriately scaled and used as input to a matrix equation. Then, by Solving the eigenvalue problem of this matrix, the optimal continuous boundary orientation and strength of the edge (probability of being an edge) are found. The edges are grouped into contour links based on the boundary orientation. Finally, the edge probabilities of long and smooth contours are set to higher probabilities, this further improves the result. As compared to Pb edge detector, this technique is reported to be of lower computational cost and run time, but of comparable boundary detection performance.

4.1.4 Edge Detection Results and Discussion

For higher level applications, like object detection and recognition, the edge set of a test image should be extracted via a edge detector that performs well in textured and cluttered images. For comparison, for a test image from ETHZ dataset, Canny, Berkeley/Pb, and Generalized Boundary edge detector results are shown in Figure 4.1. Furthermore, for test images of five classes of ETHZ dataset, the results obtained with Berkeley edge detector are shown in Figure 4.2 and the results obtained with Generalized Boundary edge detector are shown in Figure 4.3. The results obtained with these 2 methods with default parameters are similar visually, except small differences. We decided to use Berkeley edge detector, as it can provide an edge map similar to human observer's contour detection, and especially for consistency since almost all articles in the shape based object recognition literature obtain the edge set by the Pb/Berkeley edge detector.

4.2 Skeletal Shape Models

This thesis concentrates on generative shape models based on skeletons. There are two examples in the literature.

4.2.1 Tree Union Model of Skeletons

Active skeleton of Bai et al. [5] is a tree based skeleton search approach for non-rigid object detection. For the task, it builds an exemplar based shape model. Being entirely based on exemplars is its limitation, i.e., this method does not utilize the whole shape space of an object class. In the model, the medial axis is used for storing object parts in a tree union for the object detection. Boundaries and skeleton edges are attached together, each skeleton edge has an associated boundary that represents a part of the contour template. In the formulation, missing branches are allowed, for example a horse shape can have one less leg, it may change skeleton's topology.

For every shape instance, we should have the same tree structure. There is an algorithm for it. First, a skeleton is manually labeled and is used as the reference. For

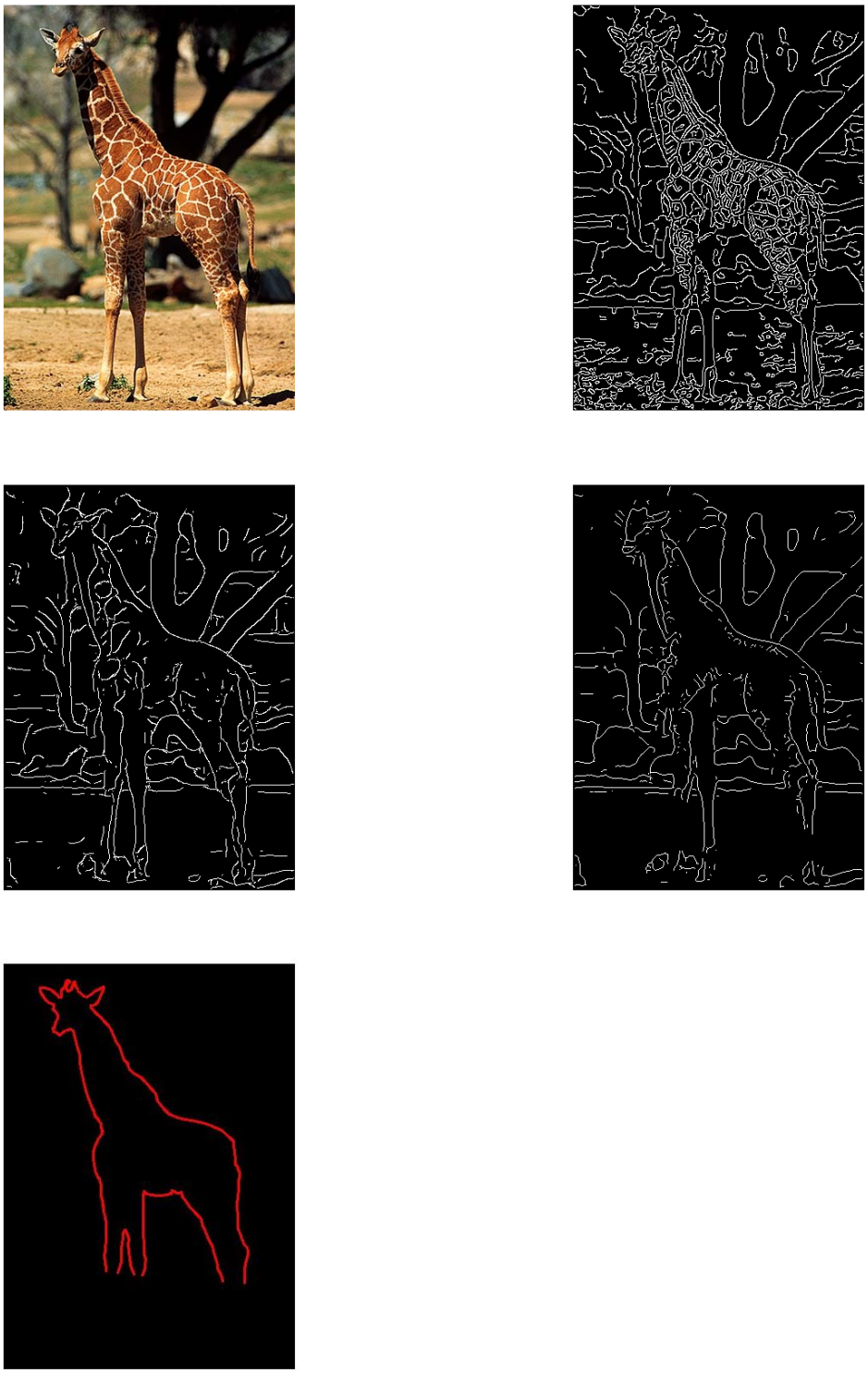
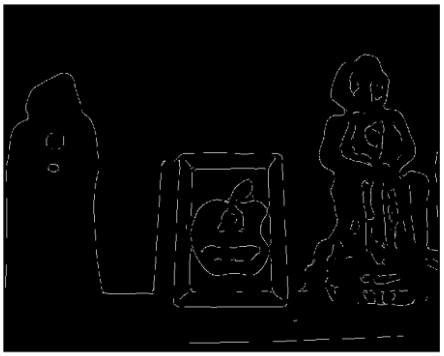
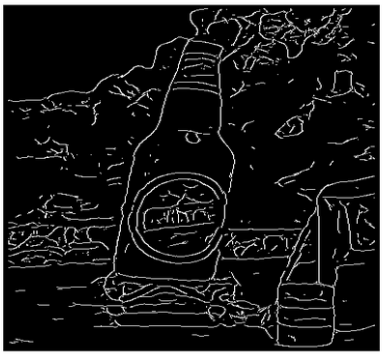


Figure 4.1: For an input giraffe image shown in (a), (b) Canny, (c) thresholded Pb, (d) thresholded Generalized Boundary detector edge detector results, (e) the mask of the object.

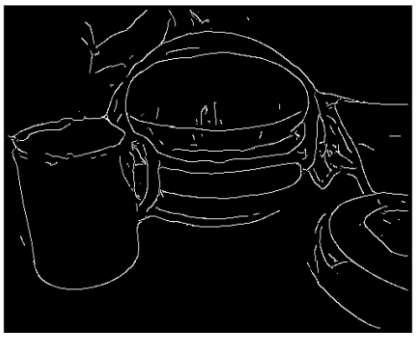


(a)



(b)

(c)



(d)

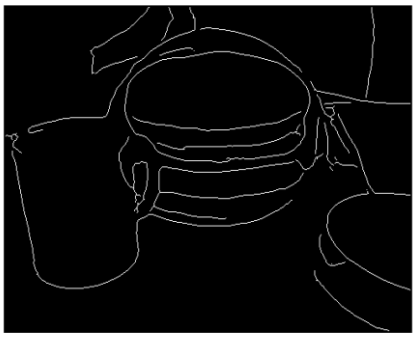
(e)

Figure 4.2: For images from 5 classes, Berkeley edge detection results

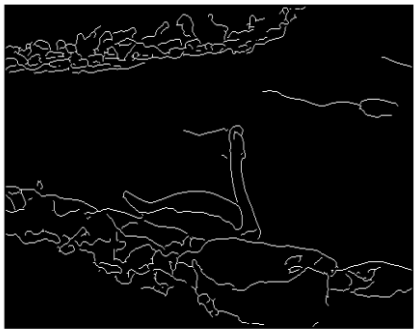
(a)

(b)

(c)



(d)



(e)

Figure 4.3: For images from 5 classes, Generalized Boundary detector edge detection results

every exemplar, skeleton matching is performed with skeleton paths. By an incremental strategy, if the skeleton matching score is below a threshold, the tree union is extended. It is also required that junction points on the paths should match junction points. In the result, every extracted skeleton has the same topology.

4.2.2 Shock Graph-based Generative Shape Model

A shock graph based representation has been used for the generative shape model of Trinh and Kimia [67]. Each shape is represented with its shock graph, and the associated parameter set. Each object class is represented by an interval of shock graph parameters.

In the parameter set, first, there are position coordinates (x, y) of the shock node. Then, there are shock radius r, angle -ψ, and ϕ–angles. Note that angle-ψ is the tangent angle of the medial axis at the shock node with respect to x-axis, it is the slope of the vector between the node and its closest point in the medial axis. Defined as the angles between normals to the maximal inscribed circle and the shock tangent, ϕ–angles are exactly object angles in Blum's paper [11]. The degree of a shock node is the number of adjacent nodes. A shock node is connected to 2 or 3 boundary points. For degree 2 nodes, there are two boundary touching points and two associated ϕ–angles; for degree 3 nodes, there are three boundary touching points and three associated ϕ–angles. For degree 2 nodes, the tangent angle determines the symmetry axis between ϕ_1 and ϕ_2; so if it is given, only one ϕ angle is required. However, for degree 3 nodes, ϕ angle is not useful, the tangent angle does not yield the remaining ϕ–angles given only one ϕ–angle. In our implementation, for a degree 1 or 2 node i, the parameter set is given as $\{x_i, y_i, r_i, \psi_i, \phi_{1_i}\}$, while the parameter set is given as $\{x_i, y_i, r_i, \phi_{1_i}, \phi_{2_i}, \phi_{3_i}\}$ for a degree-3 node i.

Computed shock graph topologies are not fixed for an object class so it is needed that a reference shock graph topology is manually chosen. In their conference work [66], all parameters were determined manually. An automated rule is suggested in their journal work. They first match each exemplar shock graph to the reference shock graph and initialize the parameter set. Then, parameters are optimized with gradient descent to find the minimum distance to the exemplar's boundaries.

A simple shape prior of the object class is constructed from the computed parameters. Each parameter is limited between the calculated minimum and maximum values plus a %50 extension. Between these possible values, the parameter range is uniformly sampled to obtain different values. Changing the value of a parameter corresponds to a different shape (object template), so many instances of the class can be constructed. This method results with a space greater than the space of shapes spanned by the training data; nevertheless in the matching stage, most of irregular shapes are removed from the solution if they do not accidentally match the edge set.

In our experiments, only for a limited number of exemplars in each class the computed shock graph has the same topology as the fixed topology, for others, topologies do not match. For the former case, we calculated the parameters set from the computed shock graph, using the vectors from the shock node location to its nearest point in the medial axis and a node and its associated boundary touching points. For the latter case, the parameter set is manually chosen (by trial and error) so that the boundary reconstructed by the parameter set matches the boundary of the object as good as possible.

An object shape is constructed from the parameters in a two step procedure. In the first step, the points where maximal inscribed circles touches the boundary and the tangent angles associated with them are found by equations in 4.3.

$$T = (cos(\psi_i), sin(\psi_i)) \tag{4.3a}$$

$$N = (-sin(\psi_i), cos(\psi_i)) \tag{4.3b}$$

$$P_1 = (X_i + r_i * cos\,(\phi_{1_i}) * T(1) - r_i * sin(\phi_{1_i}) * N(1), \\ Y_i + r_i * cos\,(\phi_{1_i}) * T(2) - r_i * sin(\phi_{1_i}) * N(2)) \tag{4.3c}$$

$$\theta_1 = \psi_i - \phi_{1_i} + \pi/2 \tag{4.3d}$$

$$P_2 = (X_i + r_i * cos(\phi_{1_i}) * T(1) + r_i * sin\,(\phi_{1_i}) * N(1), \\ Y_i + r_i * cos(\phi) * T(2) + r_i * sin(\phi_{1_i}) * N(2)) \tag{4.3e}$$

$$\theta_2 = \psi_i + \phi_{1_i} + \pi/2 \tag{4.3f}$$

In the second step, the boundary of a fragment is obtained by fitting a curve between boundary points of two adjacent nodes. The curve between two points, point P_1 (x_1, y_1), and point P_2 (x_2, y_2) that have corresponding tangent values θ_1, θ_2 is constructed by smooth bi-arc interpolation algorithm (Kimia et al. [37]). This algorithm finds the curve as composed of two arcs coinciding at a midpoint, (this midpoint is also found by the algorithm). The problem is solved by computing curvature (K_1, K_2) and arc length parameters (L_1, L_2) of these two arcs, with the least possible total curvature derivative variation. This leads to an Euler Spiral solution. The point where two arcs meet has three parameters, x and y coordinates of the point, x_m, y_m, and tangent angle, θ_m. Assuming circular arcs (i.e., the curvatures are not zero), the circular arc construction equations for the first and second arc are given by,

$$x(s) = x_1 + \frac{1}{K_1}\left(\sin\left(K_1 s + \theta_0\right) - \sin\theta_0\right) \tag{4.4a}$$

$$y(s) = y_1 - \frac{1}{K_1}\left(\cos\left(K_1 s + \theta_0\right) - \cos\theta_0\right) \tag{4.4b}$$

$$x(s) = x_m + \frac{1}{K_2}\left(\sin\left(K_2 s + \theta_m\right) - \sin\theta_m\right) \tag{4.4c}$$

$$y(s) = y_m - \frac{1}{K_2}\left(\cos\left(K_2 s + \theta_m\right) - \cos\theta_m\right) \tag{4.4d}$$

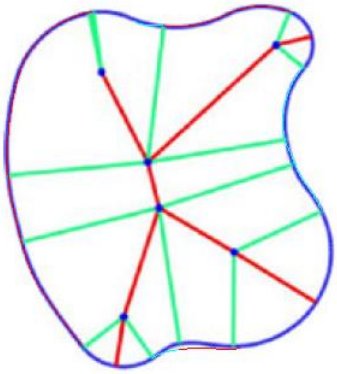
respectively. Here, s denotes the arclength which is between 0 to L_1 for the first arc, and is between 0 to L_2 for the second arc. These equations lead to three equations (involving two coordinate parameters and one tangent angle parameter) to be satisfied, the arc equation should start at the first point and the first tangent angle and should end at the second point and the second tangent angle. There are 4 unknowns (K_1, K_2, L_1, L_2). In the general case, the equations can be parametrized in terms of some variables, and the associated solutions are obtained. Let n_1 and n_2 denote integers. If $K_1 \neq 0$, L_1 is calculated from $\theta_m = K_1 * L_1 + \theta_1 + 2 * \pi * n_1$. If $K_2 \neq 0$, L_2 is calculated from $\theta_m = K_2 * L_2 + \theta_2 + 2 * \pi * n_2$. These two arcs are constructed

with the calculated curvature and arc length values using the circular arc construction equations. If curvature is found to be zero, the corresponding arc is drawn using a line equation since the curvature is zero for a line. The final shape consists of a set of curve arcs, each curve arc is either circular or a line.

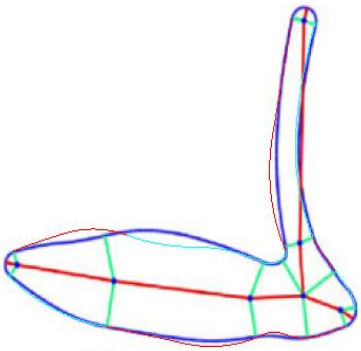
The topologies and reconstruction on top of them are shown in Figure 4.4, this is an example of almost exact reconstruction. In addition, two examples of inaccurate reconstructed boundaries for apple-logo and bottles classes are shown in Figure 4.5. For the first apple-logo example, the difference between the reconstructed object and original object is small, which does not cause a problem for our task. However, for the second apple-logo example, difference between the chosen topology and the topology of the exemplar causes a problem at reconstruction, in this case certainly we could not recover the given shape. It is claimed that the curves should yield the original boundary associated with each shock fragment, however if the boundary points of adjacent nodes are not enough close to each other, the reconstructed shape clearly deviates from the original boundary. Our conclusion is that a one-to-one reconstruction is not guaranteed, since the reconstruction procedure does not always yield the original shape using a set of skeletal points.

To make shape independent of position and scale, the parameter set is divided into two disjoint sets, intrinsic and extrinsic parameters. Intrinsic parameters specifies the shape, while extrinsic parameters puts the specified shape in a specified location and scale. For removing the effect of scale, the obtained shapes are normalized so that each one has an area of 64X64. For removing the effect of position, the distance (L) and the angle (α) between child and root nodes are used in the shape model.

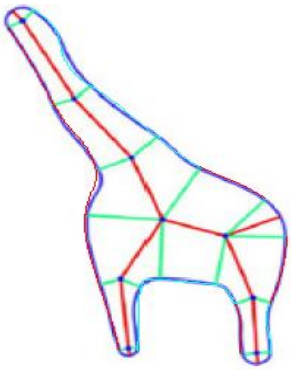
Lastly, I would like to review what modifications of the method was made in our implementation. First, we did not use authors' formulation via using ψ–angle for degree 3 nodes, instead we used three ϕ angles while tangent angles are fixed to zero (as it is not needed). For smooth arc interpolation, there was a typo in atan2 formula in their paper, atan2 is written as $\cos(x) / \sin(x)$ instead of $\sin(x) / \cos(x)$ and it was corrected in our implementation. In addition, there was a conceptual mistake that would cause the algorithm to generate unacceptable shapes. There are more than one value for arc lengths that satisfies the constraints. We should select only the one



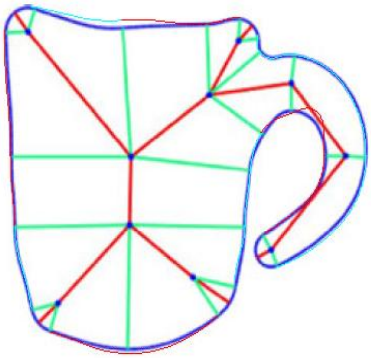
(a)



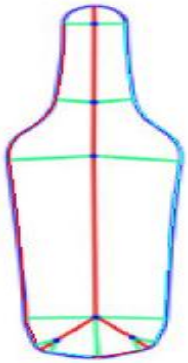
(b)



(c)

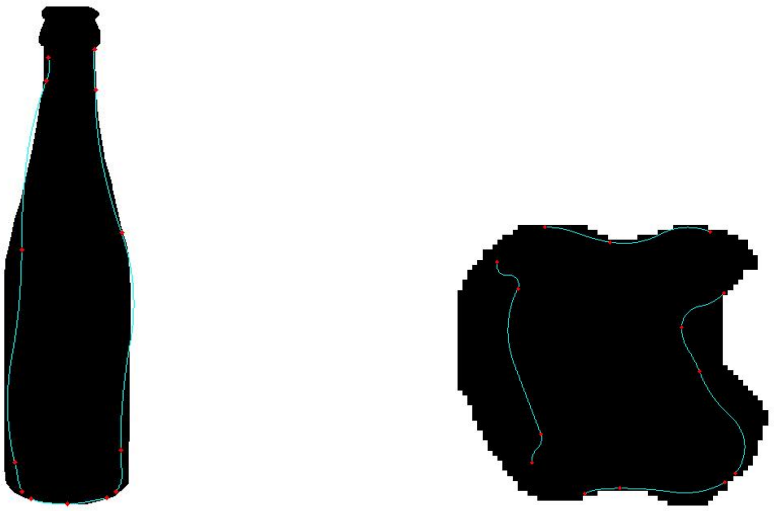


(d)



(e)

Figure 4.4: Reconstruction examples for different object classes, in this order, apple-logo, swan, giraffe, mug, bottle



(a) Reconstruction for computed shock graph parameters

(b) Reconstruction for manually selected shock graph parameters

Figure 4.5: Reconstruction of two different apple-logo shapes

that finds the shortest curve, since bigger absolute values in L_1 and L_2 forces the curve to be S-shaped. The algorithm in the paper finds the positive solution by fixing n_1 and n_2 as integers; if only n_1 (and similarly n_2) is selected suitably for the smallest absolute value of L_1 (and similarly of L_2), the curves could look natural.

4.3 Matching

The features in the query image is matched to the features in the model, and a measure for evaluating the matching is needed.

4.3.1 Chamfer Matching

Chamfer Matching is a simple and fast method for computing the similarity between a template and query image [6]. It is based on sliding a template in the image and finding the nearest edge to each template point. Objects are detected by thresholding the distance between template and nearest edge points. The computation is done

with a distance transform (DT) of the query image, from which a distance image is extracted. The values for each pixel in the distance image is distance to the closest pixel in the edge set.

4.3.2 Oriented Chamfer Matching

Shotton has proposed OCM [59], which both uses angular distance and the spatial distance. First, the best match, i.e., the edge point with minimum distance to each template point is found by using a distance transform. Then by adding orientation difference, the final detection score is obtained. Let U represent the template, and V represent the query edge set, and let $n = |U|$. The edge that is closest to to a template contour point u_i is found by minimizing the Euclidean distance via

$$v_{j^*} = \min_{v_j \epsilon V} |u_i - v_j|. \tag{4.5}$$

Then, OCM's cost function between template and query images is given by

$$d_{OCM}\left(U, V\right) = \frac{1}{n} \sum_{u_i \epsilon U} \left(1 - \lambda\right) |u_i - v_{j^*}| + \lambda |\phi\left(u_i\right) - \phi\left(v_{j^*}\right)|. \tag{4.6}$$

4.3.3 Contour/Partitioned Chamfer Matching

In OCM, an image edge can contribute to multiple contour points, there may be edges better than those matched to the contours, or, the contour can be fitted in accident to a place where each individual edge align to contour well but not as a whole. Trinh and Kimia [67] introduced CCM as a solution to these drawbacks, and report better performance than OCM, especially when there are spurious and missing edges.

For every contour point in the model, γ_i, a thin strip between middle points of neighboring contour points which has height of 12 pixels is selected. The aim is to find the edge pixel e_j in this strip that minimizes a distance function consisting of location closeness and orientation terms. If there is no edge in strip, the function takes the maximum value 1. Otherwise, the edge pixel that minimizes the distance measure given in the below formula

$$d\left(\gamma_i, e_j\right) = w_d * \min\left(\left(d(\gamma_i, e_j)/\tau_d\right), 1\right) + w_\theta * \min\left(\frac{\alpha_i - \bar{\alpha}_j}{\tau_\theta}, 1\right) \tag{4.7}$$

is selected, where α_i is the tangent angle of the contour point, and $\bar{\alpha}_j$ is the orientation of the edge pixel, and w_α and τ_θ are weights. For further checking the orientation difference of model contour and the contour defined by selected edges ($\bar{\beta}_j$ is the tangent angle of edge pixels in the contour), the zig-zag distance is calculated as

$$d\left(\gamma_i, e_j\right) = w_\alpha * \min\left(\frac{\alpha_i - \bar{\beta}_j}{\tau_\theta}, 1\right) \tag{4.8}$$

where w_α and τ_θ are the weights. The total CCM cost is the sum of the distance calculated by 4.7 and the zig-zag distance in 4.8 , divided by the total number of contour points. In addition, the objective function is tailored to each fragment, using probability distribution of each fragment's CCM cost, since the same value can correspond to good and bad matches at different fragments, for example necks can be easily detected than other parts.

Objects are searched at different scales (images are resized). For every scale, a multi-stage procedure is used for object search. The location of root node is searched in a 4X4 image grid. Given the location of the root node, possible values (within the limits) of locations of its child nodes are obtained for a set of distance values. The optimal solution is seeked by tracing the tree structure from parent to child nodes (finding the locations of child nodes given the parent nodes). For each root node location, the parameters of shock graph parameters that yields the minimum value of the chamfer function is found. As the shape consists of nodes in a tree structure, the cost can be represented as a sum, and the parameters are searched with dynamic programming (Viterbi algorithm). Among root node locations in a neighbourhood, only the locations that yields the local minimum of chamfer function are kept (non-maxima suppression). The next step is to reduce the solution set by removing all but one root node locations which yield overlapping (overlapping criterion is the 0.5 IOU criterion) region boundaries. Finally by thresholding the costs at the remaining solution set, we obtain a set of object detections for each test image. A successful object detection result for a test image (from applelogo class) with our implementation of

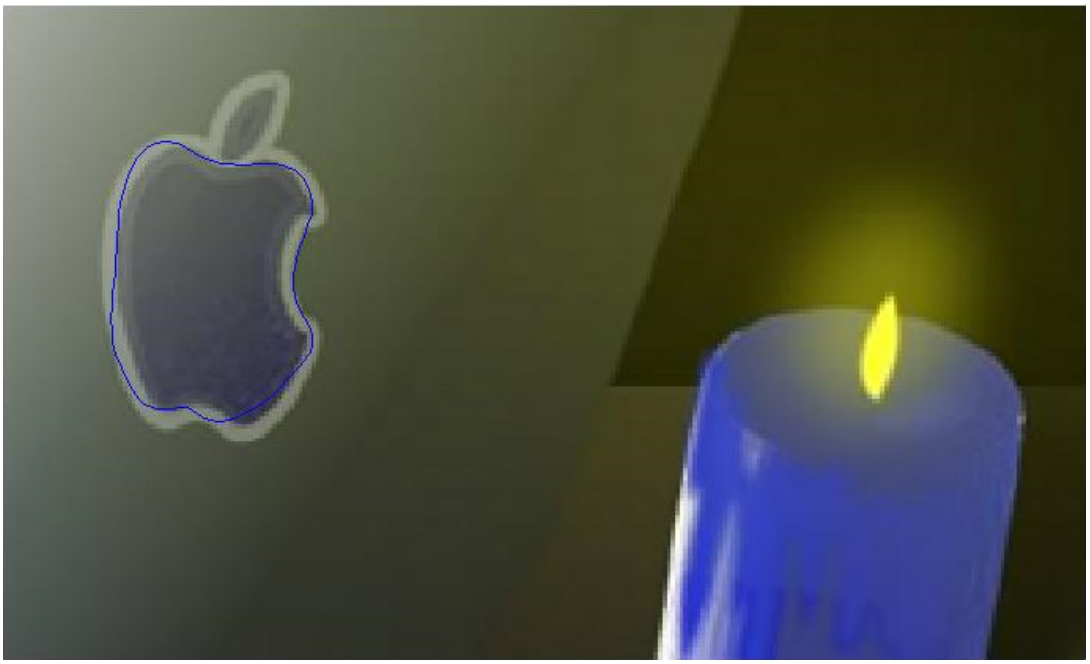


Figure 4.6: The object detection result for skeleton search

the skeleton search method is shown in Figure 4.6.

4.3.4 Directional Chamfer Matching

Liu et al. [45]'s DCM method jointly minimizes location and orientation difference terms in a 3 dimensional space to find the nearest template edge point. Let U represent the template, and V represent the query, and let $n = |U|$. The DCM's cost function between template and query images is given by

$$d_{DCM}(U, V) = \frac{1}{n} \sum_{u_i \epsilon U} \min_{v_j \epsilon V} |u_i - v_j| + \lambda |\phi(u_i) - \phi(v_j)|. \tag{4.9}$$

There are some simplifications in the algorithm that make it fast. Firstly, edge images are turned to line segments via Random sample consensus (RANSAC). The provided hand drawn templates for each class are also turned to the line representation by RANSAC, each line is represented by its start and end points, then the template is represented by $\left\{l_{[s_j, e_j]}\right\}_{j=1,\ldots,m}$. Orientations are quantized into q discrete channels evenly in $[0, \pi]$ range. The quantized orientation of each line is easily obtained, as every point in a line has the same orientation.

DCM computes a 3 dimensional tensor consisting of locations and quantized edge orientations. At the beginning, q 2D distance transforms are computed. Then, the 3-dimensional tensor is computed by solving a second order clockwise dynamic program for each pixel. Let $DT_{V\{\hat{o}\}}$ be the distance transform of edge points at a particular orientation. For a template point u_i of orientation o_i, three dimensional distance transform is given by,

$$DT3_V\left(u_i\right) = \min_{\hat{o}\in\hat{O}} \left\{ DT_{V\{\hat{o}\}}\left(u_i\right) + \lambda|o_i - \hat{o}| \right\}. \tag{4.10}$$

The orientation of a line in the template is the quantized value, $\widehat{\phi}\left(l_j\right)$. Then, the DCM cost of a line l_j can be rewritten as

$$C\left(I, l_j\right) = \frac{1}{n} \sum_{u_i \epsilon l_i} DT3_V\left(u_i, \widehat{o}\left(l_j\right)\right). \tag{4.11}$$

and overall DCM cost can be rewritten as

$$d_{DCM}\left(U, V\right) = \frac{1}{n} \sum_{lj\epsilon L_U} \sum_{u_i \epsilon l_j} DT3_V\left(u_i, \widehat{\phi}\left(l_j\right)\right). \tag{4.12}$$

For further reducing the computation time of summation operation, integral images are used. Let x_0 be the intersection of the boundary of the image and the line passing through point x with orientation $\widehat{\phi_i}$. Then, the distance transform entry of the integral image at point x and direction $\widehat{\phi_i}$ is given by

$$IDT\left(x, \widehat{\phi_i}\right) = \sum_{x_j \in l[x_0, x]} DT\left(x_j, \widehat{\phi_i}\right). \tag{4.13}$$

Then, the DCM cost can be easily calculated by

$$d_{DCM}\left(U, V\right) = \frac{1}{n} \sum_{l[s_j, e_j] \in L_U} \left[ID\left(e_j, \widehat{\phi}\left(l_{[s_j, e_j]}\right)\right) - ID\left(s_j, \widehat{\phi}\left(l_{[s_j, e_j]}\right)\right) \right]. \tag{4.14}$$

A sliding window approach is used for scanning the image, so the single template is shifted to different locations in the image. In total, 8 different scales and 3 different aspect ratios are used for searching the object. For eliminating false positives near the correct detection, a non-maxima suppression is performed, among the overlapping detections (at IOU=0.2), only the detection with the lowest score is kept.

4.3.5 Chamfer Matching Using Variational Mean Field

While DCM and OCM detects objects using fixed templates; Nyugen [51]'s method searches for the local deformations of the template in a small band. Due to clutter, chamfer matching has a high false alarm rate. Nyugen's chamfer matching method using variational mean field reduces the false alarm rate by incorporating the local variants of the shape. Yet, Nyugen's method does not bring a very successful improvement for articulated object detection.

For each template point/line (t_i), a set of parallel points/lines are added to the template. Each template point/line is considered as a hidden node (h_i), the best matching edge point/line for a hidden node is considered as an observed node (v_i). The best matching edge point/line is found by a OCM or DCM. Now, the matching cost is formulated as the solution of a MAP estimation problem of finding the shape $(H(T))$ that best matches the edge set.

$$M\left(I,T\right)=\max_{H(T)}P(V|H(T))P(H(T)) \tag{4.15}$$

There are two probabilities which are jointly maximized. The likelihood of obtaining the observed edge point given a template point is denoted by $p(v_i|h_i)$. If independence of each point/line is assumed, and the DCM cost is denoted by $C\left(I,T\right)$, this likelihood is given by,

$$\prod_{i=1}^{T}p\left(vi|ti\right)=\exp\left(-\alpha C\left(I,T\right)\right). \tag{4.16}$$

The second probability term is a shape prior of local deformations of the template (constrains it to be similar to the initial template). Only points/nodes in a fixed neighbourhood of each node is taken into account, because of the Markov assumption. For the case of OCM, vectors between the current point and its neighbour point is defined for the initial and deformed templates. For the case of DCM, vectors are defined as the the connection between the middle point of the lines h_i, and h_j or the lines t_i, and t_j. The angle between these two vectors $\overrightarrow{h_ih_j}$, $\overrightarrow{t_it_j}$ is measured with a function Θ, and the shape prior term is constructed as,

$$p\left(h_i, h_j | t_i, t_j\right) = \exp\left[-\beta | \Theta\left(\overrightarrow{h_i h_j}, \overrightarrow{t_i t_j}\right) |\right]. \tag{4.17}$$

The complexity of the problem is high for an exact solution, since there are many states to be searched for each template point. The graph structure does not have the form of a tree, so an exact solution via belief propagation is not possible. Fortunately, the problem can be solved with a fast inference technique known as variational mean field, in which a simpler factorisable Q is used to approximate P (see the next section for variational inference). Then, the cost $M\left(I, T\right)$ is approximated as $\prod Q_i\left(h_i^*\right)$.

4.4 Markov Random Fields

MRF is a probabilistic graphical model which is described by an undirected graph, $G = (V, E)$, where V denotes the set of nodes and E denotes the edges. Each node is associated with a random variable, and edges encode contextual relations between random variables. It satisfies the Markov property, non-adjacent variables are conditionally independent given all other variables. There are a number of nodes, $X = \{X_1, ..., X_N\}$, each X_i takes values from a set (state space). The objective in a MRF is described as a probability, see equation 4.18. Our goal is to maximize this probability with respect to $X = \{X_1, ..., X_N\}$. [41, 10]

$$P\left(X\right) = \frac{1}{Z} \prod_{\{i,j\} \in E} P\left(X_i, X_j\right) \prod_i P\left(X_i\right) \tag{4.18}$$

The constant Z is called the partition function and it is used for normalizing the probability mass function. Generally, unary potentials are expressed as an exponential of an energy function multiplied with a constant, $P\left(X_i\right) = \exp\left(-\alpha E\left(X_i\right)\right)$. The pairwise potential checks the compatibility of the values of X_i and X_j.

MRF presents a Bayesian approach to a computer vision problem. In a Bayesian approach, it is assumed that prior knowledge is available as a probability distribution and the value that minimizes the posterior probability is searched. MRF has also the advantage of providing a graph structure of the model, hence making it possible to

represent the relationships between variables. MRF is widely used in computer vision. For example in image denoising, pixels/superpixels can be considered as nodes, then edges correspond to pairwise relation (like smoothness of adjacent pixels). In a part-based object detection, we consider object parts as nodes; and MRF is suitable for representing the object model. In this case, the MRF keeps the structure to be close to possible configurations among parts of the object. The proposed template matching method in this thesis is in the context of a part-based object detection.

MRF can be exactly or approximately solved by a suitable fast algorithm. If the MRF has a tree structure, belief propagation also known as sum-product message passing finds the exact solution. The algorithm is time linear in the number of nodes. In belief propagation, a node's belief is the product of the unary potential at that node and all messages coming from parent nodes. Mathematically speaking, message passing equations are given by,

$$m_{ij}\left(X_j\right) = \max P\left(X_i\right) \prod_{k \in Nb(i) \setminus j} m_{ki}\left(X_i\right). \tag{4.19}$$

and beliefs (the predicted marginal probabilities), are given by,

$$b_i\left(X_i\right) = P\left(X_i\right) \prod_{j \in Nb(i)} m_{ji}\left(X_i\right). \tag{4.20}$$

If MRF does not a have tree structure, the belief propagation can be used as heuristics for finding a solution. This is, however, not guaranteed to be the exact solution. Alternatively, we can use variational inference techniques. In a variational approximation, we use a factorisable distribution $Q\left(H\right) = \prod Q_i\left(X_i\right)$ that approximates $P\left(X\right)$. The Kullback–Leibler divergence between Q and P is optimized globally and two coupled equations are found. At the beginning, the states are initialized with the same probabilities. Then, the coupled equations are solved iteratively until convergence.

CHAPTER 5

THE PROPOSED METHOD

The proposed method captures the shape variability using a sparse skeleton structure and searches for the object by using a MRF model of the object. Then, we complement the shape cue with an appearance cue. The flowchart of the proposed method has been shown in Figure 5.1.

5.1 A Skeletal Generative Shape Model

We develop a generative shape model, based on the skeleton. In the model construction process, starting from at least one example shape, we arrive at the MRF representation of the object shape. The first step is to extract skeletons from training shape(s). We fix a certain topology for each class, and sample a set of skeletal points from each shape. Then using the set of skeletons, we compute possible variations for the skeleton parameters. We explain the details of our generative shape model, in the following three steps.

5.1.1 Representation of the shape via a set of skeleton points

We extract the skeleton of a training shape using Bai's method [2]. Because we use Directed Chamfer Matching later in the search step, the shape model needs to be approximated in the form of a collection of a sufficient number of line segments. Hence, instead of using the entire skeleton, we sample it. For the purpose of sampling, we use the critical point detector of Kovesi [42] providing us with end and junction

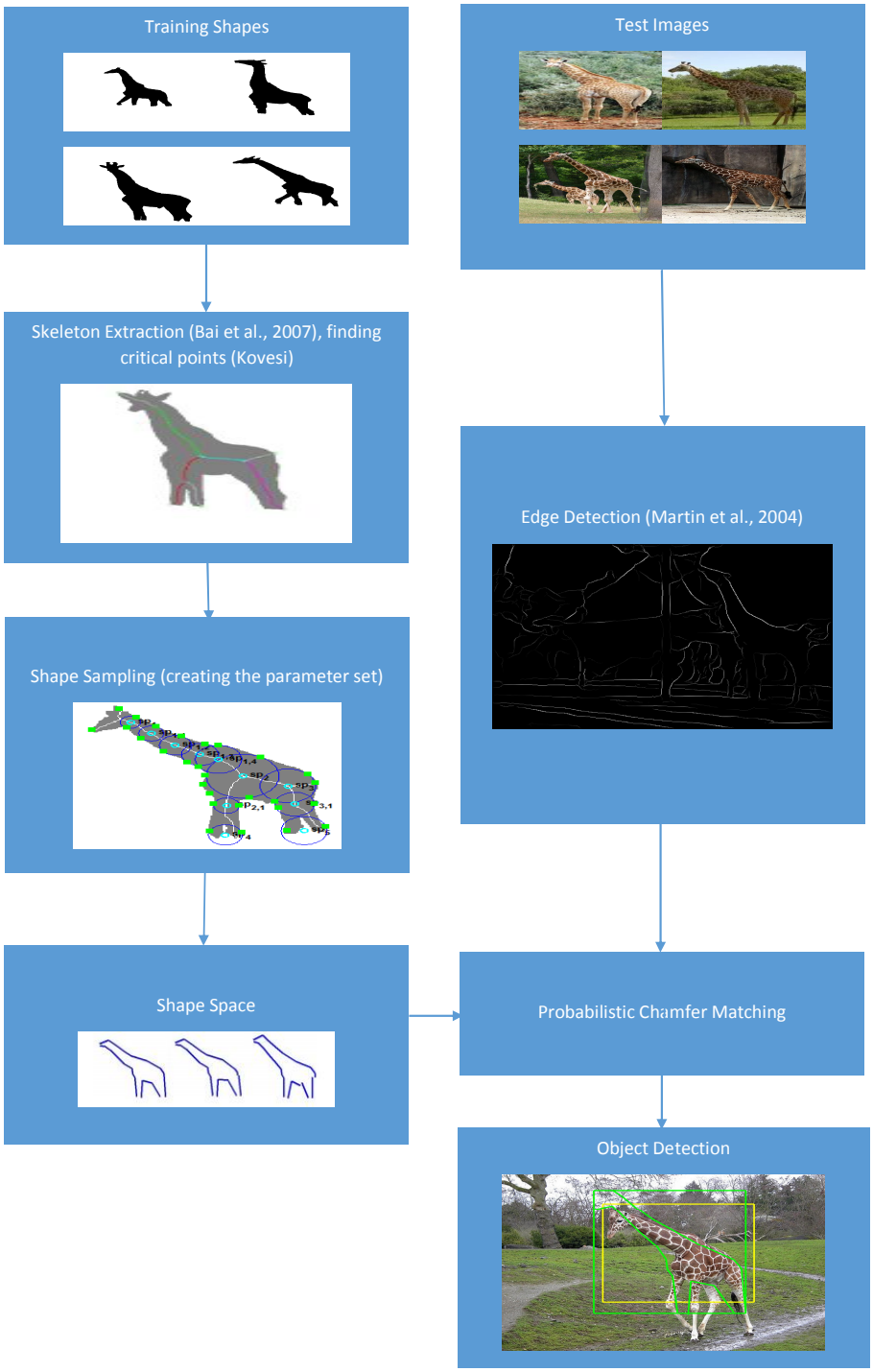


Figure 5.1: Flowchart of the proposed method.

points. By tracing the path between these critical points, the skeletal branches are obtained. In this way, an hierarchical representation of the object shape in the form of a tree is obtained by taking the branches as nodes.

From this skeleton representation, a coarse form of the shape can be reconstructed by connecting the respective boundary points associated with each of the retained skeletal point. However, this reconstruction is typically too coarse, especially if the boundaries of branches are not linear. Thus, we fix a reference resolution as a part of the model building step. To upgrade the initial coarse approximation to this resolution level, we add more samples. The best partitioning of the shape contour is not required. It is sufficient to construct a reasonable approximation. Thus, we simply add new skeleton points by sampling a long branch at equidistant intervals. From the collected point set, we obtain a linear representation of an object mask, by collecting a set of skeletal points.

We may use an empirical rule for collecting a set of skeletal points. Each skeleton branch is sampled at the same corresponding points for every object instance. We want the linear approximation be a satisfactory approximation of the input curve. For each branch, we include at least 2 points corresponding to the first and final point of the branch (for example giraffe's torso has only 2 sampled skeletal points). As for long parts, more points may be included. The neck part of the giraffe has been sampled with 4 points; the points in neck-torso branch are ordered from neck to torso and the first selected point is the point whose index is round of 0.05 of number of points in this branch, and other points are selected at 0.2, 0.4, 0.6 of the remaining part of the branch. It is possible that some critical points of the skeleton may be missing (like the junction point at the torso), the skeletal point that is most close to the point's possible location is determined by an user input. The selected skeletal for one giraffe template example is visualized in Fig. 5.2.

An automatic skeletal point sampling procedure is also possible. There are two objectives in shape sampling. First, we need to obtain the same number of points for each shape in an object. Each shape has different number of skeletal points, we should fix a certain rule for having the same number of points. Thus, a correspondence between different instances can be obtained and a model can be constructed. The second objec-

tive is to approximate the shape as good as possible with a limited number of skeletal points. We only need to select two skeletal points to be adjacent if their connection in the original shape is almost linear. It does not have to be exactly same as the original shape. To check the approximation quality, our proposed measure is symmetric area difference between the original and reconstructed shapes. For this, we find the set of points which are either in the original shape and not in the reconstructed shape, or not in the original shape but in the reconstructed shape, and calculate the cardinality of the set. Thus, for a sufficient number of points, this measure is low, but it is high for a too sparse set. For each object part, we search for a suitable approximation by firstly sampling the branch with 30 points. We calculate the approximation measure by removing a point in the branch one by one, and remove the sample point that yields the lowest approximation cost without using it. We repeat until a fixed number of points are removed. This operation is performed for 2-30 points to be kept in the branch. The cost curve saturates at a point, where no more points are needed for the approximation. This is selected by finding the point where the difference between adjacent costs over the previous cost is less than a threshold (0.1).

The set of sampled skeleton skeleton points is denoted by, $\mathbf{sp}_j$, $j = 1, ..., Np$, where Np is the number of samples. Through the rest of paper, only to the sampled skeletal points, we will be refer to as skeletal points. A skeletal point is specified by its x and y coordinates: $(sp\{x\}, sp\{y\})$. Each skeletal point belongs to at least one branch, a junction point is included in all branches which meet at that point. The skeletal points in a branch are denoted by $\{\mathbf{sp}_j\}$, $j \in V_i$, where V_i is the index set of skeletal points in branch i, consisting of Np_i entries. The skeleton structure consists of four branches/parts on giraffe class, and two branches/parts on swan class. The branches and skeletal points are shown in Figure 5.2 (b) for the giraffe class. The branch in the torso has two skeletal points, $V_3 = \{2, 3\}$.

There are 2 boundary points linked to a skeletal point in Bai's skeleton algorithm. Besides these 2 boundary points, we may associate 1-3 points at circular parts (like the boundary of the giraffe torso). The more points linked to a skeletal point, the better the constructed template approximates the shape. For obtaining additional boundary points, we draw a circle of the calculated radius. On this circle, we choose the angular coordinates such that these additional points are near to the boundary but

away from 2 boundary points that generated the skeletal point and also away from each other (based on their angle difference); we represent these additional points with a different radial coordinate than the maximal disc radius if there is a considerable difference between these radii. The boundary points linked to a skeletal point $\mathbf{sp}_i$, is denoted by $\mathbf{bp}_{i,j}$. Through the rest of paper, boundary points linked to skeletal points will be simply referred to as boundary points. The boundary points can also be represented in polar coordinates with the radius of the maximal disc and angles, (r_i, Θ_i). Boundary points ($\mathbf{bp}_{i,1}$, $\mathbf{bp}_{i,2}$) can be represented in a polar coordinate via angles, $\Theta = (\theta_{i,1}, \theta_{i,2})$. Then, the boundary point coordinates', $\mathbf{bp}$ $(bp\{x\}, bp\{y\})$ for parameters r, and θ are given by,

$$bp\{x\} = sp\{x\} + r\cos(\theta), \tag{5.1a}$$

$$bp\{y\} = sp\{y\} - r\sin(\theta). \tag{5.1b}$$

The template is represented with a set of lines. By joining boundary points associated with nearby skeletal points, as shown in Fig. 5.2 (d), the lines are constructed. A line is defined by its start and end point, $\mathbf{l}_{i,k} = (\mathbf{s}_{i,k}, \mathbf{e}_{i,k})$, where i shows the index of the part, k shows the index of the line in the part. Consider joining the boundary points linked to the adjacent skeletal points at a part i. Let $V_i(n)$, $V_i(n+1)$ denote two adjacent skeletal points. Assuming there are 2 boundary points linked to them, 2 lines between these points are obtained via: $\mathbf{l}_{i,k} = \left(\mathbf{bp}_{V_i(n),1}, \mathbf{bp}_{V_i(n+1),1}\right)$, and $\mathbf{l}_{i,k+1} = \left(\mathbf{bp}_{V_i(n),2}, \mathbf{bp}_{V_i(n+1),2}\right)$, where k and $k+1$ denotes the indexes of these lines in the part. In Fig. 5.2 (c), 2 skeletal points at the end of neck part, the boundary points linked to skeletal points and the extraction of lines between boundary points are illustrated.

5.1.2 The variability of the skeleton parameters

A generative shape model has been constructed for generating flexible shape templates via allowing the width and the pose of parts to vary. The shape variation of an object class can be captured from a training set consisting of the randomly se-

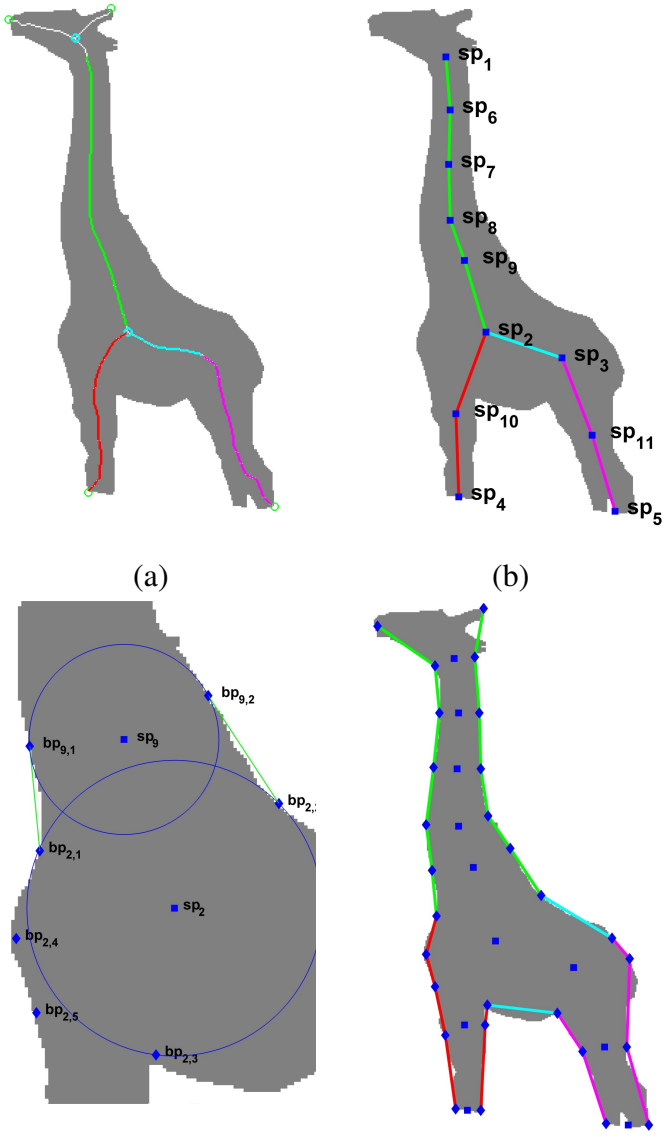
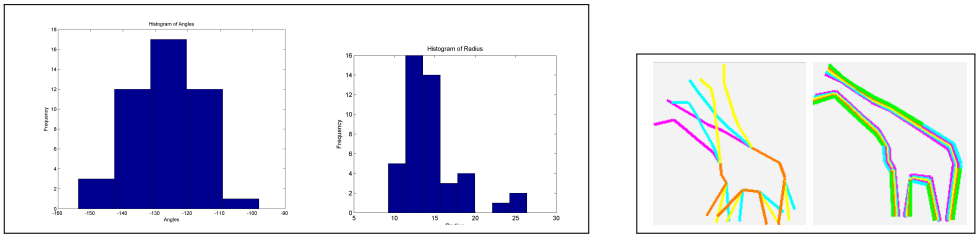


Figure 5.2: Skeletal structure for a training image. (a) Extracted skeleton. End points are shown in green circles and junction points in cyan circles. (b) The sampled skeleton. (c) Connecting lines between associated boundary points of the adjacent sampled skeleton points. (d) Conversion of the mask to a template of lines.

lected half of the images in an object class. For each shape in training, the skeleton is extracted, and the associated parameter set is calculated (the coordinates of skeletal points, and calculated radius and Φ angles). Then, all shapes are translated to the top-left, and they are normalized such that the difference between the x-coordinates of two selected skeletal points ($\mathbf{sp}_2$ and $\mathbf{sp}_3$) are the same for all instances. For articulated parts, we divide parts associated with shape instances into bins, based on the position of skeletal points. Each bin corresponds to a possible state of skeletal point corresponding to a different pose obtained by the articulated motion of the object. The state of skeletal point location corresponding to a bin is obtained by averaging the x and y coordinates of skeletal points of all instances in the part. In addition, for each skeletal point location, a corresponding Θ angle vector is found by finding the mean angle of the associated angles. The set of radii values associated with skeletal points are fit into a Gaussian probability density function (μ_i, σ_i). At generated templates, the circle centered at point i is allowed to take radii values from this set: $\{\mu_i - 1\sigma_i,\ \mu_i - 0.5\sigma_i,\ \mu_i,\ \mu_i + 0.5\sigma_i,\ \mu_i + 1\sigma_i\}$. Then, the state vector for radii of circles is given by $\mathbf{r_i} = \{r_j\}\, j \in V_i$, all circles in a part take the radius value $\mathbf{r}$ corresponding to the same state of circle size. Once the parameters are obtained, the location of the boundary points are found by inserting the skeletal point locations $\mathbf{sp}$, Θ and $\mathbf{r}$ to the equation 5.1, and finally the set of lines corresponding to different states are obtained. Since, the parts are considered as separate nodes, for the circle centered at the skeletal point joining 2 parts, its size can be larger at one part than its adjacent part. Histograms of neck's orientation (defined as the angle between two skeletal points, $\mathbf{sp}_1$ and $\mathbf{sp}_2$) and the histogram of radii variation are shown in Figure 5.3 (a); 3 states corresponding to different poses and 5 states corresponding to different radii are shown in Figure 5.3 (b). For the case with single object template, we empirically determined the location of skeletal points, and Θ angles corresponding to different poses of the articulated parts and radii variation corresponding to different widths of all parts; this is referred to as the manual parameters case.

5.1.3 From skeletons to MRF of the object's shape

Once a flexible model for a linearized shape is constructed, the detection is done by scanning the image edge map for possible locations where a hypothesized instance



(a) (b)

Figure 5.3: Distribution of shape parameters (radius and orientation) (a) and generating shapes with changing the skeletal point coordinates and skeletal radius (b).

of the model may exist. The aim is to find the most likely skeleton that could have generated the searched object by estimating its associated parameter set. To facilitate the search in a probabilistic framework, the flexible model is expressed as a MRF. In the MRF representation, each node denotes a part (e.g., leg, neck, or torso for the giraffe class) as shown in (Fig. 5.4). The edges denote pairwise relations between parts. Associated with each node i is a random variable $X_i = [\mathbf{r}_i, \mathbf{p}_i]$. Each node i is also associated with a collection of lines, $(l_{i,1}, l_{i,2}, ..., l_{i,N_i})$. A set of states is constructed for different poses, $\mathbf{p}_i$ and skeletal radii, $\mathbf{r}_i$. The lines take values over a set which consists of entries corresponding to different states of the random variable. The generated necks for the giraffe class are illustrated in Fig. 5.3 (b). Note that all lines in the part are obtained by using the random variable in the same state, i.e., all circles in the neck is either at the largest size, or at the smallest size. The set of lines in a node i corresponding to a state X_i is denoted by $\{l_{i,k}(X_i)\}$, where $k \in 1, ..., N_i$ denotes the index of any line in that node.

5.2 Detecting the object in an image using the sparse skeleton

We formulate detecting the object in an image using our MRF model. The probability of the existence of an object is defined by a joint distribution over the set of random variables $X = \{X_1, ..., X_{Nn}\}$,

$$\phi(X) = \frac{1}{Z} \prod_{\{i,j\} \in N_b} \phi(X_i, X_j) \prod_i \phi(I|X_i)\, \phi(X_i) \tag{5.2}$$

where N_b represent the set of all neighboring nodes, and Z is a normalization constant. The model contains unary and pairwise potentials. The unary potential is the

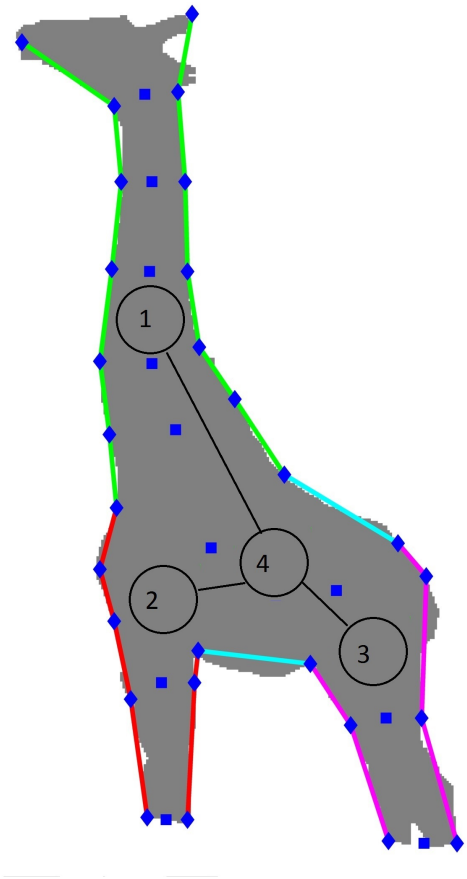


Figure 5.4: MRF structure for the giraffe class.

product of a likelihood term based on the image evidence and a shape prior term. The pairwise potential defines the relation between parts.

Likelihood potential Given the state of part $X_i = \lfloor \mathbf{r}_i, \mathbf{p}_i \rceil$, the likelihood for a test image I, $\phi\left(I|X_i\right)$ is an image fidelity term that compares the template to the query (edges of the image). The lines corresponding to possible states of X_i was pre-computed. Then, the likelihood can be obtained by summing the DCM cost of all lines in a node, and inserting the sum into an exponential equation,

$$\phi\left(I|X_i\right) = \exp\left(\sum_k -\alpha_{i,k} C\left(I, l_{i,k}\left(X_i\right)\right)\right). \tag{5.3}$$

where, $\alpha_{i,k}$ is a constant, and $C\left(I, l_{i,k}\left(X_i\right)\right)$ denotes the DCM cost of a line in the part, which is given by

$$C\left(I, l_{i,k}\right) = \frac{1}{n} \sum_{u_t \epsilon l_{i,k}} DT3_{E(I)}\left(u_t, \widehat{o}\left(l_{i,k}\right)\right). \tag{5.4}$$

Prior potentials The unary prior potential is about the radii $\phi\left(X_i\right) = \phi\left(\mathbf{r}_i\right)$ (states of skeletal point positions are considered of equal probability). For the shape prior of the radius, a Gaussian distribution was assumed: $\phi\left(r_{i,k}\right) \propto \mathcal{N}\left(r_{i,k}|\mu_{i,k}, \sigma_{i,k}\right)$ For making it to be comparable to the data fidelity term, it is input to an exponential equation. Let $\beta_{i,k}$ denote a constant, the unary prior potential is then given by

$$\phi\left(X_i\right) = \exp\left(\sum_k -\beta_{i,k} \log \phi\left(r_{i,k}\right)\right). \tag{5.5}$$

The pairwise potential $\phi\left(X_i, X_j\right) = \phi\left(\mathbf{r}_i, \mathbf{r}_j\right)$ constrains neighboring nodes to have comparable widths. It is assumed that neighboring nodes should have proportional radius values, such that if one part's width increases, its neighbor part also should increase. The pairwise potential $\phi\left(X_i, X_j\right)$ penalizes the distortion that may be caused by width difference of neighboring nodes i and j via,

$$\phi\left(X_i, X_j\right) = \exp\left(-\gamma_{i,j} \left| \frac{r_{i,k} - \mu_{i,k}}{\sigma_{i,k}} - \frac{r_{j,k} - \mu_{j,k}}{\sigma_{j,k}} \right|\right), \tag{5.6}$$

where $\gamma_{i,j}$ is a constant, k denotes the index of a skeletal point in the node, (the index among a node does not matter since the same potential is obtained for any point in the same node).

Inference In each sliding window, the object configuration that best fits the edge set is found. This accounts to the MAP estimation, i.e., finding the assignment of variables that maximizes joint posterior probability (5.2). We find the exact solution using max-product belief propagation algorithm as our model has a tree structure. Indeed, our model has a tree structure; for giraffe, the torso is designated as the root, and other nodes are leaves; for swan, the body is designated as the leaf, and the neck is the root.

The max-product belief propagation algorithm works in two stages [10]. In the first stage, messages are passed from the child nodes to their parents, from the leaf nodes towards the root node. Let N_c denote the children of a node. The message passing equation from a node i to its parent node j is given in equation 5.7 and equation 5.8, for the case i is a leaf node, and for other cases of i, respectively. The state that corresponds to the MAP estimate for the root node r is found via equation 5.9.

$$m_{i\to j}\left(X_j\right) = \max_{X_i} \phi\left(X_i, X_j\right) \phi\left(X_i\right) \phi\left(I|X_i\right) \tag{5.7}$$

$$m_{i\to j}\left(X_j\right) = \max_{X_i} \phi\left(X_i, X_j\right) \phi\left(X_i\right) \phi\left(I|X_i\right) \prod_{k\in N_c} m_{k\to i}\left(X_i\right) \tag{5.8}$$

$$\hat{X}_r = \arg\max_{X_r} \left(\prod_{k\in N_c} m_{k\to r}\left(X_r\right) \right) \phi\left(X_r\right) \phi\left(I|X_r\right) \tag{5.9}$$

Finally, after calculating the final MAP estimate at the root node, the states of other nodes can be found via backtracking: For each parent node state, the child node state which has the highest probability is saved. Given the MAP estimate of the parent node, the corresponding state of the child nodes is found using these saved states. The computational complexity of inference is $O\left(N_n N_s^2\right)$, where N_n is the number of nodes, N_s is the number of states for each node. The computational complexity is low and the algorithm is fast as long as there are not enormous number of states.

The setting of parameters Recently, Guo et al. [33] observed that shape parts should have different importances. This is inspired by human's improved ability to

recognize objects by some special parts, it is considered that different object parts do not have equal importances for object recognition. Besides, the edge map for certain parts of object are more noisy, this is another motivation for using an importance idea. For incorporating importance, we appropriately determine the parameters, α for object parts. For example, we set a higher α for giraffe's neck as compared to other parts.

5.3 Convolutional Neural Networks for Computing the Appearance Based Score

As humans use both the shape and appearance information for object detection and recognition, an automated recognition system should also use both cues. We also compute an appearance based cue using a Convolutional Neural Network (CNN) for object detection. A CNN is a deep-Learning architecture that uses the convolution operation, a typical CNN as shown in Fig. 5.5. We use the publicly available CNN code of [55].

For each class, the half of positive images and the half of negative images are used in training, and the other halves are used for testing. But the dataset is so small as compared to usual deep learning datasets. We had to pre-train the CNN with CIFAR-10 dataset. Then, we use images from ETHZ dataset. First, each training image is converted to grayscale. Around each ground truth, 10 randomly selected windows are selected. Each window is resized to 32 by 32. In addition, the mean is subtracted from the image function, and it is divided by the variance; so the data input to the layers are without any bias and variance. The labels of positive images are [1 0], and negative images' labels are [0 1]. For the loss function, the sum of squared differences between predicted and the actual label are used. In the convolutional layer, the image is convolved with a small matrix of weights, and the result is used as the input of a non-linear function (sigmoid function). In the subsampling layer, averaging of max-pooling operation is used. The last layer is the fully connected layer which is connected to all outputs in the previous layer and has 2 outputs (for 2 classes). In the last layer, softmax is used as the non-linear function. In the forward pass, the loss function is computed. In the backward pass (Backpropagation), weights are adjusted using the derivative of the loss function with respect to weights.

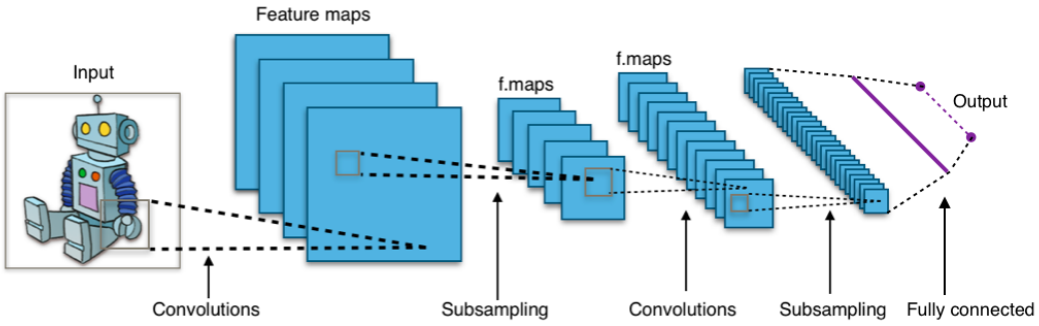


Figure 5.5: A typical CNN structure. (Illustration taken from [70])

In testing, first, we calculate the shape probability by applying the skeleton based object detection to the test image. Then, if the shape probability P_s is higher than a fixed threshold on a bounding box, we proceed with the CNN. As in the training, the image inside the box is reduced to 32X32, and the mean is subtracted, and it is divided by variance. Then, CNN calculates the output function which is viewed as the probability of belonging to the class based on appearance (denoted by P_a). If both shape probability and appearance probability from CNN are higher than some certain thresholds, we calculate the overall probability P_o by the following formula, $P_o = P_s * P_a/0.9$. Otherwise, the shape probability is take as the overall probability $P_o = P_s$. By this operation, scores of many true positives' get better, but less false positives get better scores. Now, we can only use the half of negative images in plotting ROC curves for overall detection score since the other half was used in training of the CNN.

CHAPTER 6

EXPERIMENTAL EVALUATION

6.1 Evaluation Criteria

For each class, we plot receiver operating characteristic (ROC) curve: detection rate versus false positives per image. In plotting ROC, we need to define detection criteria. We consider two separate criterion for this purpose, both of which are based on intersection over union (IOU) ratio with the ground truth bounding box. According to the first criterion, a detection is considered as successful if IOU is at least 0.2. The first criterion is also referred as Ferrari criterion. The second criterion, which is also referred as PASCAL criterion, is more conservative. It requires at least 0.5 for IOU to consider a detection as successful. To compare our work against similar methods in the literature, we report the detection rate of our method at false positives per image (FFPI) values equal to 0.3 and 0.4 for IOU ratio of 0.2 and 0.5.

6.2 Shape Datasets

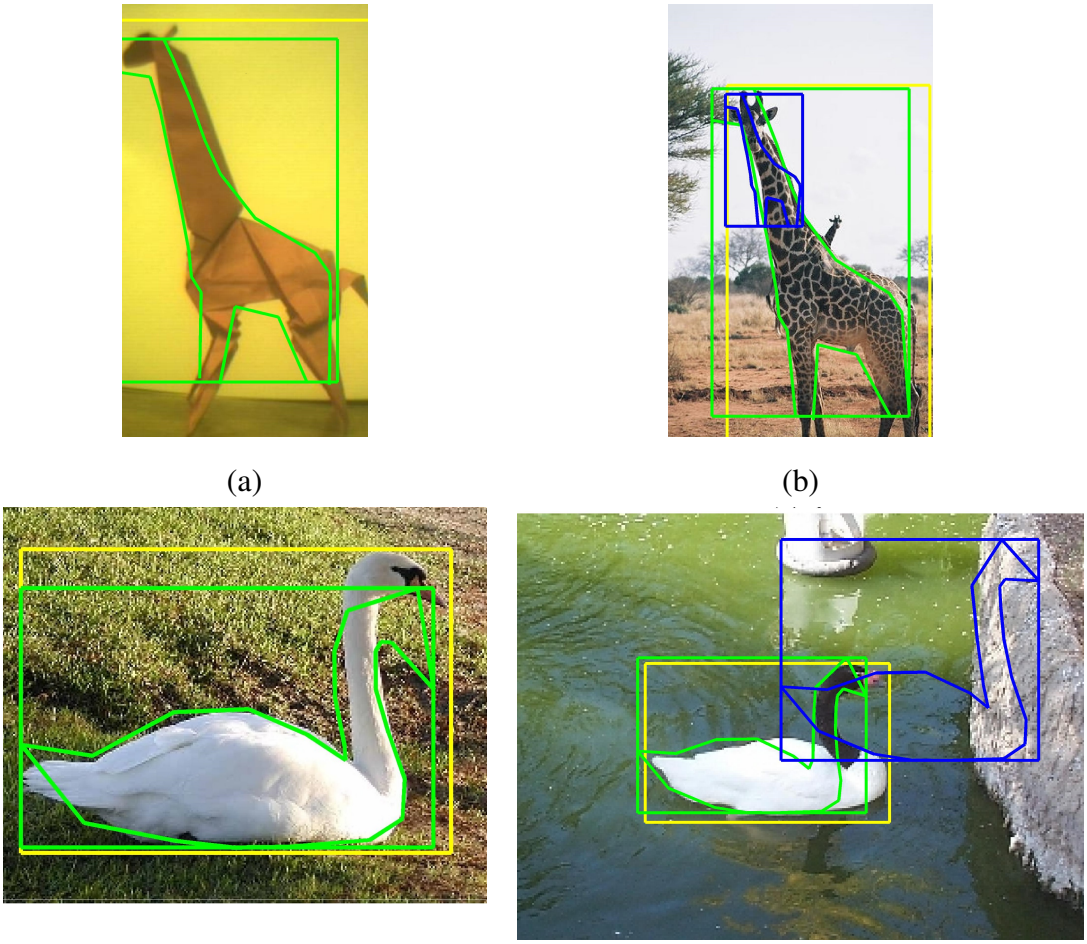
Databases especially suitable for object detection and recognition with shape are Weizmann Horse Dataset, INRIA Horse dataset, UIUC Car Dataset, Caltech Motorbike Dataset, and ETHZ Shape Database. The ETHZ dataset is perhaps the most commonly used datasets for evaluating shape based object detection tasks. It consists of a total of 255 images from 5 object categories: applelogo (40 images), bottle (48 images), giraffe (87 images), mug (48 images), and swan (32 images). Some of the 255 images contain more than one instance of the respective class. For example,

the giraffe class has 91 giraffes in 87 images. Objects from different classes do not appear on the same image. On one hand, the ETHZ dataset is particularly a challenging dataset because of cluttered backgrounds; due to which, there are many spurious edges. An additional challenge in this dataset is due to the fact that some of the objects of interest comprise only a fraction of the total image area. In this dataset, the intra-class shape variation is also high. The images contain instances of objects with varying scales. On the other hand, the images in the ETHZ dataset does not contain many objects with occlusion and viewpoint variations are relatively low. Weizmann Horse Dataset consists of 628 images, half of them contain horses, remaining half are from Caltech background images. 100 images are used in training, another 100 images are used in validation, and remaining images are used in testing. INRIA horse dataset consists of 340 images; 170 images are from horse category (at several scales and against cluttered backgrounds), other 170 images do not contain horses (negative images).

6.3 Experimental Results

Experiments with ETHZ Dataset For test images, we obtained the edge image by thresholding the output of the Berkeley edge detector using 0.15 as the threshold and then reducing some clutter edges. As in the DCM [45], we search the image with a sliding window, at different scales and aspect ratios. For scale change, the query image (edge map) is scaled; for changing the aspect ratio, the template's width to height ratio is varied. We used 3 aspect ratios for both classes while 10 scales for the giraffe and 9 scales for the swan. For DCM, we use its publicly available code. The ratios between consecutive scales and the aspect ratio are set to 1.2 and 1.1, respectively (the default settings). For plotting ROC curves, different detection thresholds of intervals 0.05 are used.

In each window, the probabilistic chamfer matching cost is calculated by solving the MAP problem, and each variable is assigned to their MAP estimates, and an object detection window corresponding to the assigned values is found. The detection which satisfies the desired IOU ratio and has the best score among other detections is considered as true detection. All other detections are considered as false positives.



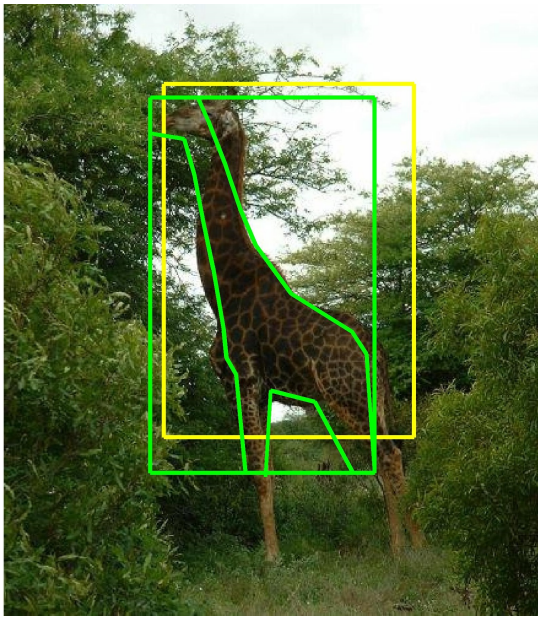
(a) (b)

(c) (d)

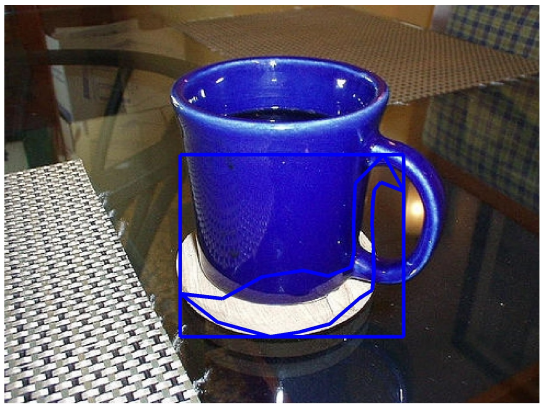
Figure 6.1: Correct detections are shown in green boxes and the false positive(s) in blue. The ground truth is the yellow box.

Some detection examples are shown in Figure 6.1. The detection results illustrate that the method can successfully detect objects that exhibit large shape variation due to articulation, even in cluttered images.

We observe that the false positives (accidental alignments) are caused mainly by the texture edges inside the objects or background clutter. This is the main limitation of this template matching approach. A false positive example is shown in Figure 6.2. In addition, there is a false negative, for the giraffe detection in Figure 6.2. In this case, the detection score is not good enough for including the detection at the threshold corresponding to the 0.4 FFPI, since the constructed shape does not fit the shape in the test image well.



(a) failure case (object is in giraffe class)



(b) false positive from an image from mug class (object is in swan class)

Figure 6.2: Proposed method's failure cases. Green shows false positives.

Firstly, the performance of manual parameters case has been evaluated. The ROC curves of DCM are shown in Figure 6.3. In each of the four figures, there are 2 curves, the blue one showing existing DCM method with the fixed template case, the cyan one shows the result of our method when the shape parameters are set manually. It is observed that the proposed method already improves the detection rate as compared to existing DCM method with the same template. This performance increase is mainly due to the method's capacity for handling shape variations of the class.

If the model parameters are estimated from the data by using half of the data for training, the detection rates may increase further. We also tested weighting the α parameter in equation 5.3, and observed some additional increase in the performance. By adding an appearance based term we see an additional improvement in the ROC curve, as for the same detection rate we get a lower FFPI number. In Figure 6.3, there are 4 ROC curves. The blue one again shows the result of existing method. The result of our shape based method first with all α are equal is shown in yellow, with α are varied is shown in magenta, and the result of the joint shape and appearance based detector is shown in green.

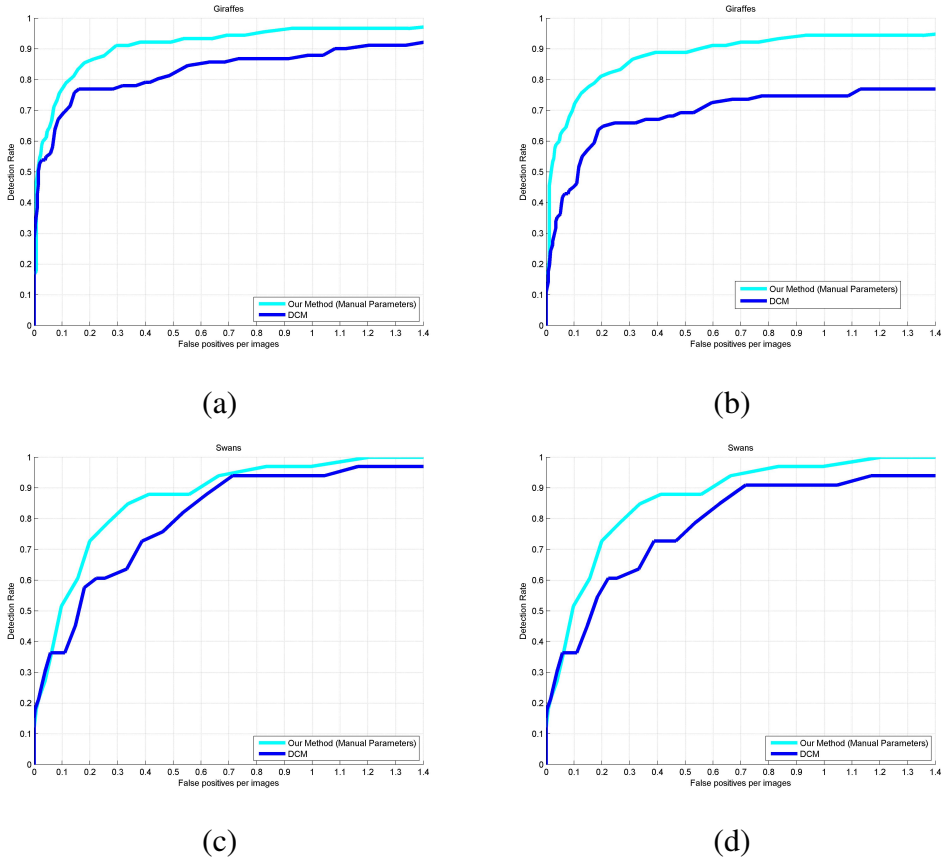


Figure 6.3: ROC curves of the proposed method as compared to DCM (detection with a hand drawn template) for IOU=0.2 (left column) and IOU=0.5 (right column.) for giraffe (top row) and swan (bottom row).

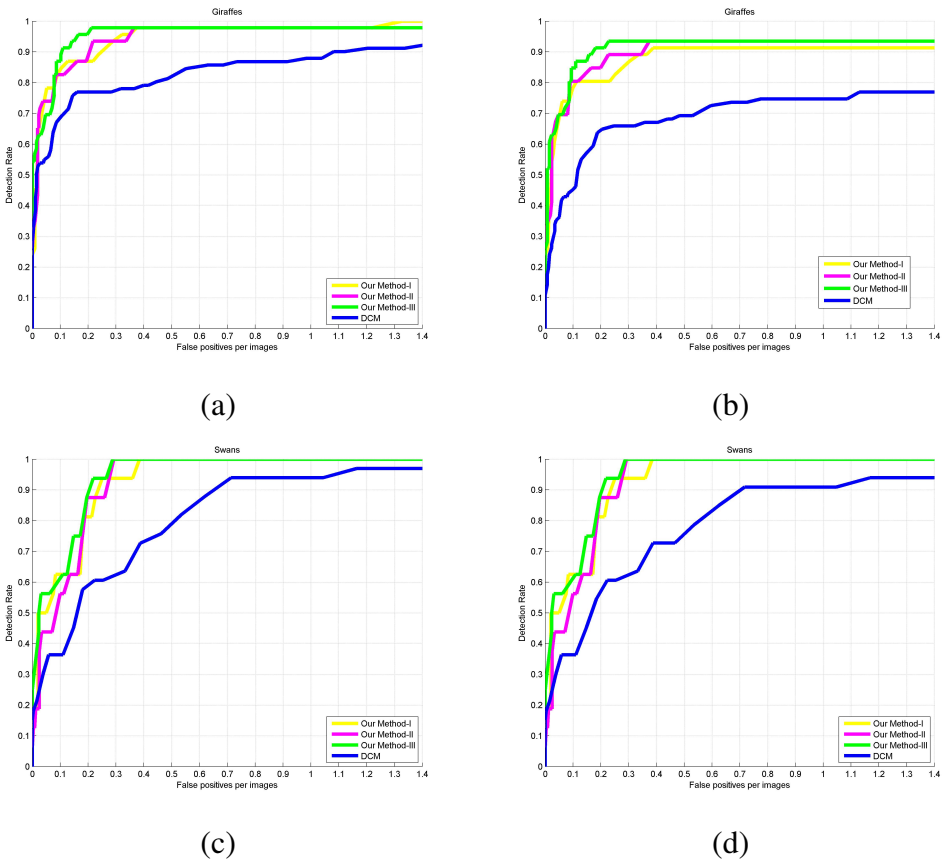


Figure 6.4: ROC curves for IOU=0.2 (left column) and IOU=0.5 (right column) for giraffe (top row) and swan (bottom row). The method-I means α equal, method-II α weighted, method-III means appearance added. DCM shows the result of a detection with a hand drawn template.

Overall, for giraffe, if IOU=0.5 is adopted as the criterion, the detection rates for FFPI rates of 0.3 and 0.4 are 0.891/ 0.935. With CNN added, the rates increase to 0.935/0.935. This rate increases to 0.935/0.978 with shape only detection, and to 0.9783/0.9783 with CNN added detection, if IOU=0.2 is adopted as the criterion. For swan, the respective rates are 1/1 (at IOU=0.5 or 0.2), for both cases.

Comparison to Related Work In Tables 6.1, 6.2, we give a comparison of detection rates at FFPI rates of 0.3 and 0.4 for the two IOU criteria. Starred methods use appearance cues along with shape. The *proposed method* is weighted constants case, and *proposed method + CNN* means the overall probability is computed using both cues. According to the results reported in Table 6.1, among 11 methods, our method

Table 6.1: For IOU=0.5, comparison of the methods for 0.3/0.4 FPPI. Starred methods use cues other than shape (appearance, texture).

Method	Giraffe	Swan
Proposed Method	0.891/**0.935**	**1.000/1.000**
* Proposed Method +CNN	**0.935/0.935**	**1.000/1.000**
Ferrari et al. [27]	0.399/0.445	0.632/0.705
Zhu et al. [74]	0.681/0.681	0.824/0.824
Lu et al. [47]	0.734/0.770	0.938/0.938
Yang et al. [71]	0.769/0.792	0.909/0.941
Riemenschneider et al. [57]	0.792/0.819	0.926/0.926
* Toshev et al. [65]	0.813/0.868	0.934/0.934
Srinivasan et al. [62]	0.872/0.896	**1.000/1.000**
Maji et al. [49]	0.896/0.896	0.882/0.882
Trinh and Kimia [67]	0.898/0.918	0.941/**1.000**
* Wang et al. [69]	0.920/0.920	0.940/0.940

outperformed most of the methods in these two classes, for FFPI=0.4 and IOU=0.5. According to the results reported in Table 6.2, the proposed method significantly outperformed existing shape based methods for IOU=0.2, except [65]. Note that [65] uses additional cues. For swan, at FFPI 0.3, we achieve the rate obtained by [47], yet we outperform if FFPI is increased to 0.4.

Though the chamfer matching considerably tolerates moderate misalignment in position, and rotation, it is severely affected by large deformations of object shapes. Moreover, the accidental alignment with image clutter may lead to spurious detections. Our skeleton-based probabilistic method is able to overcome these problems in Chamfer matching and finds objects. The importance of using skeleton-based shape is revealed when our detection rates are compared to the detection rates for the method of [51]. On the average, our rate is 25% higher.

Finally, the proposed method has been compared to related methods in terms of computational complexity and time. The run times of the methods for one image are provided in Table 6.3. The most similar method [67] reports the run time as 10-20 minutes. Our implementation of Trinh and Kimia's method in C++ has 30 minutes run time. Although that method uses dynamic programming, it uses a very large

Table 6.2: For IOU=0.2, comparison of the methods for 0.3/0.4 FPPI. Starred methods use cues other than shape (appearance, texture).

Method	Giraffe	Swan
Proposed Method	0.935/**0.978**	**1.000/1.000**
* Proposed Method + CNN	**0.978/0.978**	**1.000/1.000**
Ferrari et al. [27]	0.399/0.445	0.632/0.705
Nyugen et al. [51]	0.681/0.681	0.824/0.824
Lu et al. [47]	0.734/0.770	0.938/0.938
* Toshev et al. [65]	**0.978/0.978**	0.934/0.934

,

Table 6.3: Computational Time Comparison

Method	Run Time
Proposed Method	2-30 s
Ferrari et al. [27]	Not reported
Zhu et al. [74]	Not reported
Lu et al. [47]	Not reported
Yang et al. [71]	30-40 s
Riemenschneider et al. [57]	5.3 s (average)
* Toshev et al. [65]	30-45 s
Srinivasan et al. [62]	Several minutes
Maji et al. [49]	Not reported
Trinh and Kimia [67]	10-20 minutes
* Wang et al. [69]	Not reported

state space, so the inference is slow. Instead, we allow a much smaller state space, this means a faster inference. The reduction of the computational burden (on the same computer and language, our runs only take 2 to 30 seconds) is our method's advantage. We also believe that the gained computational resources should be used to integrate color and texture cues to further increase the detection performance. For the other methods, we could not directly make a comparison. Most methods do not report their run time, and for the methods that reported computational times, we do not have the code to check computational times in our computer. However, we can at least say that our method is one of the fastest methods. For example, the run time of [62] is several minutes, but our method's run time is in seconds.

CHAPTER 7

CONCLUSION

In this Ph. D. study, shape based object detection and recognition has been the main focus. Object detection and recognition is a difficult problem because it has to deal with many kinds of objects with different properties; objects may have different positions, scales, may appear in viewpoints, and have a significant intra-class variation. There is not a single straightforward approach that can solve problem generally. Object detection methods based on texture and local features generally work well for many object classes. However, when the objects are characterized by certain shapes, other provided descriptions are not useful for achieving a satisfactory performance. In that case, shape information can be used as a good measure to detect and segment objects as a whole. Our particular concern is to use skeletons in object detection. In the literature, skeleton has been mostly used for shape analysis in silhouettes, but there are rare applications of skeletons in object detection and recognition too.

We proposed a new method for matching templates in real images using a sparse skeleton. There are two contributions. As the first contribution, a generative shape model based on a sparse skeleton has been proposed. As the second contribution, a probabilistic approach based on the skeletal object structure has been developed. The proposed method is simple yet seemingly effective for shape cue based object detection. The object shape is modeled as a MRF where each node of the field is a collection of a few sampled skeleton points. For skeleton extraction, the method of Bai et al. [5] is used as it provides a simple yet effective means to calculate a clean skeleton. The skeletons are sampled to simplify the shape model, reducing the boundary to a collection of line segments. Shape hypotheses, in the form of bound-

ary line segments, are generated by sampling the MRF. Regularity of the generated hypotheses is attained by suitably defined priors for pairs of MRF nodes. Matching a generated hypothesis to an image edge map is solved in a probabilistic framework where chamfer cost is employed to measure the mismatch. We also weighted the role of different nodes (for example neck will have more weight as compared to other parts), motivated by observations that some parts of the shape are more important for recognition. One further development is that we combined appearance based features from the CNNs with the proposed shape based method for calculating the final detection cost.

Our experimental results show that the approach, despite its simplicity and computational efficiency, provides an effective means of integrating shape cue into object detection. Even better performance is observed with further developments, like weighting and using CNNs. The superiority of the proposed method as compared to many the state-of-the-art methods has been proved. Our method has significant advantages as compared to related work with respect to computational time. We also implemented the most similar work, the skeleton search of Trinh and Kimia [67]. With the same machine and with same programming language (C++), our method brought a striking improvement in computational time (from 30 minutes to 2-30 seconds).

In the future work, there are a few directions to be explored. Firstly, we could also a false positive rejection term to the cost function to deal with false positives. Secondly, we could apply the method to other databases. Finally, we could consider the Chamfer matching as the first step, and add a verification stage where detection is scored with a sensitive algorithm.

REFERENCES

[1] N. Adluru and L. J. Latecki. Contour grouping based on contour-skeleton duality. *International Journal of Computer Vision*, 83(1):12–29, 2009.

[2] X. Bai, L. J.Latecki, and W. Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):449–462, March 2007.

[3] X. Bai and L. J. Latecki. Path similarity skeleton graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1282–1292, July 2008.

[4] X. Bai, Q. Li, L. J. Latecki, W. Liu, and Z. Tu. Shape band: A deformable object detection approach. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 1335–1342, June 2009.

[5] X. Bai, X. Wang, L.J. Latecki, W. Liu, and Z. Tu. Active skeleton for non-rigid object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 575–582, Sept 2009.

[6] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'77, pages 659–663, San Francisco, CA, USA, 1977. Morgan Kaufmann Publishers Inc.

[7] S. Belongie, J. Malik, and J. Puzicha. Matching with shape contexts, 2001. [Online; accessed 1-February-2016].

[8] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:509–522, 2001.

[9] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 26–33 vol. 1, 2005.

[10] A. Blake, P. Kohli, and C. Rother. *Markov Random Fields for Vision and Image Processing*. The MIT Press, 2011.

[11] H. Blum. Biological shape and visual science (part I). *Journal of Theoretical Biology*, 38:205–287, 1973.

[12] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, June 1986.

[13] J. Carreira, F. Li, and C. Sminchisescu. Object recognition by sequential figure-ground ranking. *International Journal of Computer Vision*, 98(3):243–262, 2012.

[14] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1312–1328, July 2012.

[15] Y. Chen, L. Zhu, A. Yuille, and H. J. Zhang. Unsupervised learning of probabilistic object models (poms) for object classification, segmentation, and recognition using knowledge propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1747–1761, Oct 2009.

[16] A .Y. Chia, D. Rajan, M. K. Leung, and S. Rahardja. Object recognition by discriminative combinations of line segments, ellipses, and appearance features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1758–1772, Sept 2012.

[17] W. Choi, K. Lam, and W. Siu. Extraction of the euclidean skeleton based on a connectivity criterion. *Pattern Recognition*, 36(3):721 – 729, 2003.

[18] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, February 2003.

[19] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *European Conference on Computer Vision (ECCV)*, pages 484–498, 1998.

[20] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, Jun 2001.

[21] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Compututer Vision Image Understanding*, 61(1):38–59, January 1995.

[22] R. M. David, C. F. Charless, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, May 2004.

[23] P. Dimitrov, C. Phillips, and K. Siddiqi. Robust and efficient skeletal graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 417–423 vol.1, 2000.

[24] J. Feldman and M. Singh. Bayesian estimation of the shape skeleton. *Proceedings of the National Academy of Sciences*, 103(47):18014–18019, 2006.

[25] P. F. Felzenszwalb and J. D. Schwartz. Hierarchical matching of deformable shapes. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 1–8, June 2007.

[26] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):36–51, Jan 2008.

[27] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *International Journal of Computer Vision*, 87(3):284 –303, may 2010.

[28] V. Ferrari, T. Tuytelaars, and L. van Gool. Object detection by contour segment networks. In *European Conference on Computer Vision (ECCV)*, 2006.

[29] R. Fisher, S. Perkins, A. Walker, and E. Wolfart. Skeletonization/medial axis transform, 2003. [Online; accessed 1-February-2016].

[30] P. J. Giblin and B. B. Kimia. On the intrinsic reconstruction of shape from its symmetries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):895–911, July 2003.

[31] L. Gorelick and R. Basri. Shape based detection and top-down delineation using image segments. *International Journal of Computer Vision*, 83(3):211–232, 2009.

[32] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt. Shape representation and classification using the poisson equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1991–2005, 2006.

[33] G. Guo, Y. Wang, T. Jiang, A. Yuille, F. Fang, and W. Gaon. A shape reconstructability measure of object part importance with applications to object detection and localization. *International Journal of Computer Vision*, 108(3):241–258, July 2014.

[34] G. Heitz, G. Elidan, B. Packer, and D. Koller. Shape-based object localization for descriptive classification. *International Journal of Computer Vision*, 84(1):40–62, 2009.

[35] D. G. Kendall. Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121, Aug 1984.

[36] D. G. Kendall. A Survey of the Statistical Theory of Shape. *Statistical Science*, 4(2):87–99, 1989.

[37] B. B. Kimia, I. Frankel, and A. M. Popescu. Euler spiral for shape completion. *International Journal of Computer Vision*, 54(1-3):159–182, 2003.

[38] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker. Shapes, shocks, and deformations i: The components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15(3):189–224, 1995.

[39] I. Kokkinos and P. Maragos. Synergy between object recognition and image segmentation using the expectation-maximization algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1486–1501, Aug 2009.

[40] I. Kokkinos and A. Yuille. Inference and learning with hierarchical shape models. *International Journal of Computer Vision*, 93(2):201–225, 2011.

[41] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2011.

[42] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

[43] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *European Conference on Computer Vision (ECCV), Workshop on Statistical Learning in Computer Vision*, pages 17–32, May 2004.

[44] M. Leordeanu, R. Sukthankar, and C. Sminchisescu. Efficient closed-form solution to generalized boundary detection. In *European Conference on Computer Vision (ECCV)*, pages 516–529, Berlin, Heidelberg, 2012. Springer-Verlag.

[45] M. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1696–1703, June 2010.

[46] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[47] C. Lu, L. J. Latecki, N. Adluru, X. Yang, and H. Ling. Shape guided contour grouping with particle filters. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2288–2295, Sept 2009.

[48] D. Macrini. Shapematcher 5. `http://www.cs.toronto.edu/~dmac/ShapeMatcher/index.html`, 2005.

[49] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1038–1045, June 2009.

[50] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[51] D. T. Nguyen. A novel chamfer template matching method using variational mean field. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2425–2432, June 2014.

[52] A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *International Journal of Computer Vision*, 80(1):16–44, 2008.

[53] A. Opelt and A. Zisserman. A boundary-fragment-model for object detection. In *In European Conference on Computer Vision (ECCV)*, pages 575–588, 2006.

[54] B. Packer, S. Gould, and D. Koller. A unified contour-pixel model for figure-ground segmentation. In *European Conference on Computer Vision (ECCV)*, 2010.

[55] R. B. Palm. Prediction as a candidate for learning deep hierarchical models of data. Master's thesis, Technical University of Denmark, Denmark, 2012.

[56] N. Payet and S. Todorovic. From a set of shapes to object discovery. In *European Conference on Computer Vision (ECCV)*, volume 6315 of *Lecture Notes in Computer Science*, pages 57–70. Springer Berlin Heidelberg, 2010.

[57] H. Riemenschneider, M. Donoser, and H. Bischof. Using partial edge contour matches for efficient object category localization. In *European Conference on Computer Vision (ECCV)*, volume 6315 of *Lecture Notes in Computer Science*, pages 29–42. Springer Berlin Heidelberg, 2010.

[58] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):550–571, May 2004.

[59] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *IEEE International Conference on Computer Vision, (ICCV)*, volume 1, pages 503–510 Vol. 1, Oct 2005.

[60] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1270–1281, July 2008.

[61] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2007.

[62] P. Srinivasan, Q. Zhu, and J. Shi. Many-to-one contour matching for describing and discriminating object shape. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1673–1680, June 2010.

[63] S. Tari, J. Shah, and H. Pien. Extraction of shape skeletons from grayscale images. *Computer Vision and Image Understanding*, 66(2):133–146, 1997.

[64] A. Telea and J. J. Van Wijk. An augmented fast marching method for computing skeletons and centerlines. In *Proceedings of the symposium on Data Visualisation 2002*, pages 251–ff. Eurographics Association, 2002.

[65] A. Toshev, B. Taskar, and K. Daniilidis. Shape-based object detection via boundary structure segmentation. *International Journal of Computer Vision*, 99(2):123–146, September 2012.

[66] N. H. Trinh and B. B. Kimia. Category-specific object recognition and segmentation using a skeletal shape model. In *proceedings of the British Machine Vision Conference (BMVC)*, 2009.

[67] N. H. Trinh and B. B. Kimia. Skeleton search: Category-specific object recognition and segmentation using a skeletal shape model. *International Journal of Computer Vision*, 94(2):215–240, 2011.

[68] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1991.

[69] X. Wang, X. Bai, T. Ma, W. Liu, and L .J. Latecki. Fan shape model for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 151–158, June 2012.

[70] Wikipedia. Convolutional neural network, 2016. [Online; accessed 1-February-2016].

[71] X. Yang, H. Liu, and L. J. Latecki. Contour-based object detection as dominant set computation. *Pattern Recognition*, 45(5):1927–1936, May 2012.

[72] P. Yarlagadda and B. Ommer. From meaningful contours to discriminative object shape. In *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, ECCV'12, pages 766–779, Berlin, Heidelberg, 2012. Springer-Verlag.

[73] Ö. C. Özcanlı and B. B. Kimia. Generic object recognition via shock patch fragments. In *Proceedings of the British Machine Vision Conference*, pages 1030–1039. Warwick Print, 2007.

[74] Q. Zhu, L. Wang, Y. Wu, and J. Shi. Contour context selection for object detection: A set-to-set contour matching approach. In *European Conference on*

Computer Vision (ECCV), volume 5303 of *Lecture Notes in Computer Science*, pages 774–787. Springer Berlin Heidelberg, 2008.

APPENDIX A

STATISTICAL METHODS USED IN SHAPE ANALYSIS

In shape analysis, since we have several exemplars as input data, dimension reduction is necessary for obtaining compact representation of the shape. Dimension reduction seeks to find a linear mapping of the data from a high dimension space to a lower dimension space while keeping the variance of the data in the mapped representation. One possible dimension reduction technique is the PCA, which is closely related to Karhunen–Loève transform. PCA is applied to other computer vision problems, for example to face recognition ([68]).

Given a data set of M high dimensional signal samples $\{x_i\}_{i=1}^{M}$, where $x_i \epsilon \Re^N$, the data $\bar{x}$ (consisting of concatenation of all $\{x_i\}$) is projected onto a low dimensional space using PCA. Firstly, the mean and covariance of data are computed as follows $\bar{\mu} = \frac{1}{N}\sum_{i=1}^{M} x_i$, and $K = \frac{1}{N}\sum_{i=1}^{M}(x_i - \mu)(x_i - \mu)^T$. Eigendecomposition is obtained by $Ke = \lambda e$, where sorted eigenvalues are $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$, and associated eigenvalues are $e_1, e_2, ..., e_N$.

The data can be projected onto a space spanned by only keeping k largest eigenvectors of covariance matrix, if if $M < N$. An approximation of data is found by computing projection coefficients $\alpha_v = (\bar{x} - \bar{\mu}) \cdot \bar{e}_v$ (where $v = 1, ..., k$) and using them for reconstruction of data:

$$\bar{x} \approx \bar{\mu} + \sum_{v=1}^{k} \alpha_v \bar{e_v}$$

We may fix a variance threshold for eigenvalues selected over all eigenvalues, and the reduced dimension can be chosen for satisfying this threshold. By using all eigen-

vectors and all coefficients, no data reduction is obtained (the variance ratio is 1), and consequently the reconstructed data is exactly equal to x:

$$\bar{x} = \bar{\mu} + \sum_{v=1}^{N} \alpha_v \bar{e_v}$$

In an ASM [21], PCA is used for the statistical shape representation. Let mean of data be denoted by $\overline{x}$, eigenvectors be denoted by E_s, projection coefficients representing shape be denoted by b_s. The landmark points are then represented/approximated by $x = \overline{x} + (E_s * b_s)$.

Ferrari's method [27] also applies PCA as in ASM of Cootes. A point distribution model from training shapes is obtained, and the point coordinates matrix is computed. The mean shape (S), eigenvectors (stored as a matrix, E) and associated eigenvalues λ_i are obtained from the point coordinates matrix. Only the n largest eigenvectors that account for the 95% of variance are selected, the shape is represented by the vector b in the reduced dimension and each coefficient in b is bounded between $-3\lambda_i$ to $3\lambda_i$.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Altınoklu, Metin Burak
Nationality: Turkish (TC)
Date and Place of Birth: 04.05.1984, Ankara
Marital Status: Single
Phone: 0 312 2554181
email: altinokluburak@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
M.S.	METU	2009
B.S.	METU	2006
High School	Gazi Anatolian High School	2002

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2007-2012	METU	Teaching Assistant

PUBLICATIONS

International Journal Publications

Metin Burak Altınoklu, İlkay Ulusoy, Sibel Tarı, A probabilistic sparse skeleton based object detection. Pattern Recognition Letters (submitted).