Graph-Based Representations in Pattern Recognition

7th IAPR-TC-15 International Workshop, GbRPR 2009
Venice, Italy, May 26-28, 2009
Proceedings
Volume Editors

Andrea Torsello
Department of Computer Science
“Ca’ Foscari” University of Venice
Venice, Italy
E-mail: torsello@dsi.unive.it

Francisco Escolano
Department of Computer Science
and Artificial Intelligence
Alicante University
Alicante, Spain
E-mail: sco@dccia.ua.es

Luc Brun
GreyC
University of Caen
Caen Cedex, France
E-mail: luc.brun@greyc.ensicaen.fr

Library of Congress Control Number: Applied for

CR Subject Classification (1998): I.5, I.3, I.4, I.2.10, G.2.2

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

ISSN 0302-9743
ISBN-10 3-642-02123-9 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

springer.com
© Springer-Verlag Berlin Heidelberg 2009
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12682966 06/3180 5 4 3 2 1 0
Preface

This volume contains the papers presented at the 7th IAPR-TC-15 Workshop on Graph-Based Representations in Pattern Recognition – GbR 2009. The workshop was held in Venice, Italy between May 26–28, 2009. The previous workshops in the series were held in Lyon, France (1997), Haindorf, Austria (1999), Ischia, Italy (2001), York, UK (2003), Poitiers, France (2005), and Alicante, Spain (2007).

The Technical Committee (TC15, http://www.greyc.ensicaen.fr/iapr-tc15/) of the IAPR (International Association for Pattern Recognition) was founded in order to federate and to encourage research work at the intersection of pattern recognition and graph theory. Among its activities, the TC15 encourages the organization of special graph sessions in many computer vision conferences and organizes the biennial GbR Workshop.

The scientific focus of these workshops covers research in pattern recognition and image analysis within the graph theory framework. This workshop series traditionally provide a forum for presenting and discussing research results and applications in the intersection of pattern recognition, image analysis and graph theory.

The papers in the workshop cover the use of graphs at all levels of representation, from low-level image segmentation to high-level human behavior. There are papers on formalizing the use of graphs for representing and recognizing data ranging from visual shape to music, papers focusing on the development of new and efficient approaches to matching graphs, on the use of graphs for supervised and unsupervised classification, on learning the structure of sets of graphs, and on the use of graph pyramids and combinatorial maps to provide suitable coarse-to-fine representations. Encouragingly, the workshop saw the convergence of ideas from several fields, from spectral graph theory, to machine learning, to graphics.

The papers presented in the proceedings have been reviewed by at least two members of the Program Committee and each paper received on average of three reviews, with more critical papers receiving as many as five reviews. We sincerely thank all the members of the Program Committee and all the additional referees for their effort and invaluable help. We received 47 papers from 18 countries and 5 continents. The Program Committee selected 18 of them for oral presentation and 19 as posters. The resulting 37 papers revised by the authors are published in this volume.

March 2009

Andrea Torsello
Francisco Escolano
Luc Brun
Organization

General Chairs

Andrea Torsello, Università Ca’ Foscari di Venezia, Italy
Francisco Escolano, Universidad de Alicante, Spain
Luc Brun, GREYC ENSICAEN, France

Program Committee

I. Bloch, TELECOM ParisTech, France
H. Bunke, University of Bern, Switzerland
S. Dickinson, University of Toronto, Ontario, Canada
M. Figueiredo, Instituto Superior Técnico, Portugal
E. R. Hancock, University of York, UK
C. de la Higuera, University of Saint-Etienne, France
J.-M. Jolion, Universite de Lyon, France
W. G. Kropatsch, Vienna University of Technology, Austria
M. Pelillo, Università Ca’ Foscari di Venezia, Italy
A. Robles-Kelly, National ICT Australia (NICTA), Australia
A. Shokoufandeh, Drexel University, PA, USA
S. Todorovic, Oregon State University, OR, USA
M. Vento, Università di Salerno, Italy
R. Zabih, Cornell University, NY, USA

Organizing Committee

S. Rota Bulò, Università Ca’ Foscari di Venezia, Italy
A. Albarelli, Università Ca’ Foscari di Venezia, Italy
E. Rodolà, Università Ca’ Foscari di Venezia, Italy

Additional Referees

Andrea Albarelli, Daniela Giorgi, Emanuele Rodolà
Xiang Bai, Michael Jamieson, Samuel Rota Bulò
Sebastien Bougleux, Jean-Christophe Janodet, Émilie Samuel
Gustavo Carneiro, Rolf Lakemper, Cristian Sminchisescu
Fatih Demirci, Miguel Angel Lozano
Aykut Erdem, James Maclean
Sébastien Fourey, Anand Rangarajan

Sponsoring Institutions

Dipartimento di Informatica
Università Ca’ Foscari di Venezia, Italy
# Table of Contents

## Graph-Based Representation and Recognition

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching Hierarchies of Deformable Shapes</td>
<td>1</td>
</tr>
<tr>
<td><em>Nadia Payet and Sinisa Todorovic</em></td>
<td></td>
</tr>
<tr>
<td>Edition within a Graph Kernel Framework for Shape Recognition</td>
<td>11</td>
</tr>
<tr>
<td><em>François-Xavier Dupé and Luc Brun</em></td>
<td></td>
</tr>
<tr>
<td>Coarse-to-Fine Matching of Shapes Using Disconnected Skeletons by</td>
<td>21</td>
</tr>
<tr>
<td>Learning Class-Specific Boundary Deformations</td>
<td></td>
</tr>
<tr>
<td><em>Aykut Erdem and Sibel Tari</em></td>
<td></td>
</tr>
<tr>
<td>An Optimisation-Based Approach to Mesh Smoothing: Reformulation and</td>
<td>31</td>
</tr>
<tr>
<td>Extensions</td>
<td></td>
</tr>
<tr>
<td><em>Yskandar Hamam and Michel Couprie</em></td>
<td></td>
</tr>
<tr>
<td>Graph-Based Representation of Symbolic Musical Data</td>
<td>42</td>
</tr>
<tr>
<td><em>Bassam Mokbel, Alexander Hasenfuss, and Barbara Hammer</em></td>
<td></td>
</tr>
<tr>
<td>Graph-Based Analysis of Nasopharyngeal Carcinoma with Bayesian</td>
<td>52</td>
</tr>
<tr>
<td>Network Learning Methods</td>
<td></td>
</tr>
<tr>
<td><em>Alex Aussem, Sergio Rodrigues de Morais, Marilys Corbex, and Joël Favrel</em></td>
<td></td>
</tr>
<tr>
<td>Computing and Visualizing a Graph-Based Decomposition for Non-manifold Shapes</td>
<td>62</td>
</tr>
<tr>
<td><em>Leila De Floriani, Daniele Panozzo, and Annie Hui</em></td>
<td></td>
</tr>
<tr>
<td>A Graph Based Data Model for Graphics Interpretation</td>
<td>72</td>
</tr>
<tr>
<td><em>Endre Katona</em></td>
<td></td>
</tr>
<tr>
<td>Tracking Objects beyond Rigid Motion</td>
<td>82</td>
</tr>
<tr>
<td><em>Nicole Artner, Adrian Ion, and Walter G. Kropatsch</em></td>
<td></td>
</tr>
<tr>
<td>Graph-Based Registration of Partial Images of City Maps Using</td>
<td>92</td>
</tr>
<tr>
<td>Geometric Hashing</td>
<td></td>
</tr>
<tr>
<td><em>Steffen Wachenfeld, Klaus Broelemann, Xiaoyi Jiang, and Antonio Krüger</em></td>
<td></td>
</tr>
</tbody>
</table>

## Graph Matching

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Polynomial Algorithm for Submap Isomorphism: Application to</td>
<td>102</td>
</tr>
<tr>
<td>Searching Patterns in Images</td>
<td></td>
</tr>
<tr>
<td><em>Guillaume Damiand, Colin de la Higuera, Jean-Christophe Janodet, Émilie Samuel, and Christine Solnon</em></td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>A Recursive Embedding Approach to Median Graph Computation</td>
<td>113</td>
</tr>
<tr>
<td><em>M. Ferrer, D. Karatzas, E. Valveny, and H. Bunke</em></td>
<td></td>
</tr>
<tr>
<td>Efficient Suboptimal Graph Isomorphism</td>
<td>124</td>
</tr>
<tr>
<td><em>Kaspar Riesen, Stefan Fankhauser, Horst Bunke, and Peter Dickinson</em></td>
<td></td>
</tr>
<tr>
<td>Homeomorphic Alignment of Edge-Weighted Trees</td>
<td>134</td>
</tr>
<tr>
<td><em>Benjamin Raynal, Michel Couprie, and Venceslas Biri</em></td>
<td></td>
</tr>
<tr>
<td>Inexact Matching of Large and Sparse Graphs Using Laplacian Eigenvectors</td>
<td>144</td>
</tr>
<tr>
<td><em>David Knossow, Avinash Sharma, Diana Mateus, and Radu Horaud</em></td>
<td></td>
</tr>
<tr>
<td>Graph Matching Based on Node Signatures</td>
<td>154</td>
</tr>
<tr>
<td><em>Salim Jouili and Salvatore Tabbone</em></td>
<td></td>
</tr>
<tr>
<td>A Structural and Semantic Probabilistic Model for Matching and Representing a Set of Graphs</td>
<td>164</td>
</tr>
<tr>
<td><em>Albert Solé-Ribalta and Francesc Serratosa</em></td>
<td></td>
</tr>
<tr>
<td>Arc-Consistency Checking with Bilevel Constraints: An Optimization</td>
<td>174</td>
</tr>
<tr>
<td><em>Aline Deruyver and Yann Hodé</em></td>
<td></td>
</tr>
</tbody>
</table>

**Graph Clustering and Classification**

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pairwise Similarity Propagation Based Graph Clustering for Scalable Object Indexing and Retrieval</td>
<td>184</td>
</tr>
<tr>
<td><em>Shengping Xia and Edwin R. Hancock</em></td>
<td></td>
</tr>
<tr>
<td>A Learning Algorithm for the Optimum-Path Forest Classifier</td>
<td>195</td>
</tr>
<tr>
<td><em>João Paulo Papa and Alexandre Xavier Falcão</em></td>
<td></td>
</tr>
<tr>
<td>Improving Graph Classification by Isomap</td>
<td>205</td>
</tr>
<tr>
<td><em>Kaspar Riesen, Volkmar Frinken, and Horst Bunke</em></td>
<td></td>
</tr>
<tr>
<td>On Computing Canonical Subsets of Graph-Based Behavioral Representations</td>
<td>215</td>
</tr>
<tr>
<td><em>Walter C. Mankowski, Peter Bogunovich, Ali Shokoufandeh, and Dario D. Salvucci</em></td>
<td></td>
</tr>
<tr>
<td>Object Detection by Keygraph Classification</td>
<td>223</td>
</tr>
<tr>
<td><em>Marcelo Hashimoto and Roberto M. Cesar Jr.</em></td>
<td></td>
</tr>
<tr>
<td>Graph Regularisation Using Gaussian Curvature</td>
<td>233</td>
</tr>
<tr>
<td><em>Hewayda ElGhawalby and Edwin R. Hancock</em></td>
<td></td>
</tr>
</tbody>
</table>
Characteristic Polynomial Analysis on Matrix Representations of Graphs ...................................................... 243
   Peng Ren, Richard C. Wilson, and Edwin R. Hancock

Flow Complexity: Fast Polytopal Graph Complexity and 3D Object Clustering .................................................. 253
   Francisco Escolano, Daniela Giorgi, Edwin R. Hancock,
   Miguel A. Lozano, and Bianca Falcidieno

Pyramids, Combinatorial Maps, and Homologies

Irregular Graph Pyramids and Representative Cocycles of Cohomology Generators .................................................. 263
   Rocio Gonzalez-Diaz, Adrian Ion, Mabel Iglesias-Ham, and
   Walter G. Kropatsch

Annotated Contraction Kernels for Interactive Image Segmentation ..... 273
   Hans Meine

3D Topological Map Extraction from Oriented Boundary Graph ......... 283
   Fabien Baldacci, Achille Braquelaire, and Guillaume Damiand

An Irregular Pyramid for Multi-scale Analysis of Objects and Their Parts ............................................................. 293
   Martin Drauschke

A First Step toward Combinatorial Pyramids in n-D Spaces .......... 304
   Sébastien Fourey and Luc Brun

Cell AT-Models for Digital Volumes ........................................ 314
   Pedro Real and Helena Molina-Abril

From Random to Hierarchical Data through an Irregular Pyramidal Structure ...................................................... 324
   Rimon Elias, Mohab Al Ashraf, and Omar Aly

Graph-Based Segmentation

Electric Field Theory Motivated Graph Construction for Optimal Medical Image Segmentation ........................................ 334
   Yin Yin, Qi Song, and Milan Sonka

Texture Segmentation by Contractive Decomposition and Planar Grouping ......................................................... 343
   Anders Bjorholm Dahl, Peter Bogunovich, and Ali Shokoufandeh
Image Segmentation Using Graph Representations and Local Appearance and Shape Models ........................................ 353  
  Johannes Keustermans, Dieter Seghers, Wouter Mollemans, Dirk Vandermeulen, and Paul Suetens

Comparison of Perceptual Grouping Criteria within an Integrated Hierarchical Framework .................... 366  
  R. Marfil and A. Bandera

Author Index ............................................................ 377
Matching Hierarchies of Deformable Shapes

Nadia Payet and Sinisa Todorovic
Oregon State University, Corvallis, OR 97331, USA
payetn@onid.orst.edu, sinisa@eecs.oregonstate.edu

Abstract. This paper presents an approach to matching parts of deformable shapes. Multiscale salient parts of the two shapes are first identified. Then, these parts are matched if their immediate properties are similar, the same holds recursively for their subparts, and the same holds for their neighbor parts. The shapes are represented by hierarchical attributed graphs whose node attributes encode the photometric and geometric properties of corresponding parts, and edge attributes capture the strength of neighbor and part-of interactions between the parts. Their matching is formulated as finding the subgraph isomorphism that minimizes a quadratic cost. The dimensionality of the matching space is dramatically reduced by convexifying the cost. Experimental evaluation on the benchmark MPEG-7 and Brown datasets demonstrates that the proposed approach is robust.

1 Introduction

This paper is about shape matching by using: (1) a new hierarchical shape representation, and (2) a new quadratic-assignment objective function that is efficiently optimized via convexification. Many psychophysical studies suggest that shape perception is the major route for acquiring knowledge about the visual world [1]. However, while humans are very efficient in recognizing shapes, this proves a challenging task for computer vision. This is mainly due to certain limitations in existing shape representations and matching criteria used, which typically cannot adequately address matching of deformable shapes. Two perceptually similar deformable shapes may have certain parts very different or even missing, whereas some other parts very similar. Therefore, accounting for shape parts in matching is important. However, it is not always clear how to define a shape part. The motivation behind the work described in this paper is to improve robustness of shape matching by using a rich hierarchical shape representation that will provide access to all shape parts existing at all scales, and by formulating a matching criterion that will account for these shape parts and their hierarchical properties.

We address the following problem: Given two shapes find correspondences between all their parts that are similar in terms of photometric, geometric, and structural properties, the same holds recursively for their subparts, and the same holds for their neighbor parts. To this end, a shape is represented by a hierarchical attributed graph whose node attributes encode the intrinsic properties of corresponding multiscale shape parts (e.g., intensity gradient, length, orientation), and edge attributes capture the strength of neighbor and part-of interactions between the parts. We formulate shape matching as finding the subgraph isomorphism that preserves the original graph connectivity and minimizes a quadratic cost whose linear and quadratic terms account for differences.
between node and edge attributes, respectively. The cost is defined so as to be invariant
to scale changes and in-plane rotation of the shapes. The search in the matching space
of all shape-part pairs is accelerated by convexifying the quadratic cost, which also re-
duces the chances to get trapped in a local minimum. As our experiments demonstrate,
the proposed approach is robust against large variations of individual shape parts and
partial occlusion.

In the rest of this paper, Sec. 2 points out main contributions of our approach with re-
spect to prior work, Sec. 3 describes our hierarchical representation of a shape, Sec. 4.1
specifies node and edge compatibilities and formulates our matching algorithm, Sec. 4.2
explains how to convexify and solve the quadratic program, and Sec. 5 presents experi-
mental evaluation of our approach.

2 Our Contributions and Relationships to Prior Work

This section reviews prior work and points out our main contributions. Hierarchical
shape representations are aimed at efficiently capturing both global and local properties
of shapes, and thus facilitating their matching. Shortcomings of existing representations
typically reduce the efficiency of matching algorithms. For example, the arc-tree [2,3]
trades off its accuracy and stability for lower complexity, since it is a binary tree, gener-
ated by recursively splitting the curve in two halves. Arc-trees are different for similar
shapes with some part variations, which will be hard to match. Another example is the
curvature scale-space [4,5] that loses its descriptive power by pre-specifying the degree
of image decimation (i.e., blurring and subsampling), while capturing salient curvature
points of a contour at different degrees of smoothing. Also, building the articulation-
invariant, part-based signatures of deformable shapes, presented in [6], is sensitive to
the correct identification of the shape’s landmark points and to the multidimensional
scaling and estimating of the shortest path between these points. Other hierarchical
shape descriptions include the Markov-tree graphical models [7], and the hierarchy of
polygons [8] that are based on the restrictive assumptions about the number, size, and
hierarchy depth of parts that a curve consists of. The aforementioned methods encode
only geometric properties of shape parts, and their part-of relationships, yielding a strict
tree. In contrast, we use a more general, hierarchical graph that encodes the strength of
all ascendant-descendant and neighbor relationships between shape parts, as well as
their geometric and photometric properties. The sensitivity of the graph structure to
small shape variations is reduced, since we estimate the shape’s salient points at multi-
ple scales. Also, unlike in prior work, the number of nodes, depth, and branching factor
in different parts of the hierarchical graph are data dependent.

Graph-based shape matching has been the focus of sustained research activity for more
than three decades. Graph matching may be performed by: (i) exploiting spectral prop-
erties of the graphs’ adjacency matrices [9,10]; (ii) minimizing the graph edit-distance
[11,12]; (iii) finding a maximum clique of the association graph [13]; (iv) using the
expectation-maximization of a statistical, generative model [14]. Regardless of a particu-
lar formulation, graph matching in general can be cast as a quadratic assignment problem,
where a linear term in the objective function encodes node compatibility functions, and
a quadratic term encodes edge compatibility functions. Therefore, approaches to graph
matching mainly focus on: (i) finding suitable definitions of the compatibility functions; and (ii) developing efficient algorithms for approximately solving the quadratic assignment problem (since it is NP-hard), including a suitable reformulation of the quadratic into linear assignment problem. However, most popular approximation algorithms (e.g., relaxation labeling, and loopy belief propagation) critically depend on a good initialization and may be easily trapped in a local minimum, while some (e.g., deterministic annealing schemes) can be used only for graphs with a small number of nodes. Graduated nonconvexity schemes [15], and successive convexification methods [16] have been used to convexify the objective function of graph matching, and thus alleviate these problems. Since it is difficult to convexify matching cost surfaces that are not explicit functions, these methods resort to restrictive assumptions about the functional form of a matching cost, or reformulate the quadratic objective function into a linear program. In this paper, we develop a convexification scheme that shrinks the pool of matching candidates for each individual node in the shape hierarchy, and thus renders the objective function amenable to solution by a convex quadratic solver.

3 Hierarchical Shape Representation

In this paper, a shape (also called contour or curve) is represented by a hierarchical graph. We first detect the contour’s salient points at multiple scales, which in turn define the corresponding shape parts. Then, we derive a hierarchy of these shape parts, as illustrated in Fig. 1.

**Multiscale part detection.** A data-driven number of salient (or dominant) points along the contour are detected using the scale-invariant algorithm of [17]. This algorithm does not require any input parameters, and remains reliable even when the shape is rich in both fine and coarse details, unlike most existing approaches. The algorithm first determines, for each point along the curve, its curvature and the region of support, which jointly serve as a measure of the point’s relative significance. Then, the dominant points are detected by the standard nonmaximum suppression. Each pair of subsequent dominant points along the shape define the corresponding shape part. The end points of each shape part define a straight line that is taken to approximate the part. We recursively apply the algorithm of [17] to each shape part whose associated line segment has a

![Fig. 1. An example contour: (left) Lines approximating the detected contour parts are marked with different colors. (right) The shape parts are organized in a hierarchical graph that encodes their part-of and neighbor relationships. Only a few ascendant-descendant and neighbor edges are depicted for clarity.](image)
larger approximation error than a pre-set threshold. This threshold controls the resolution level (i.e., scale) at which we seek to represent the contour’s fine details. How to compute this approximation error is explained later in this section. After the desired resolution level is reached, the shape parts obtained at different scales can be organized in a tree structure, where nodes and parent-child (directed) edges represent the shape parts and their part-of relationships. The number of nodes, depth, and branching factors of each node of this tree are all automatically determined by the shape at hand.

**Transitive closure.** Small, perceptually negligent shape variations (e.g., due to varying illumination in images) may lead to undesired, large structural changes in the shape tree (e.g., causing a tree node to split into multiple descendants at multiple levels). As in [18], we address these potential structural changes of the shape tree by adding new directed edges that connect every node with all of its descendants, resulting in a transitive closure of the tree. Later, in matching, the transitive closures will allow that a search for a maximally matching node pair is conducted over all descendants under a visited ancestor node pair, rather than stopping the search if the ancestors’ children do not match. This, in turn, will make matching more robust.

**Neighbors.** Like other strictly hierarchical representations, the transitive closure of the shape tree is capable of encoding only a limited description of spatial-layout properties of the shape parts. For example, it cannot distinguish different layouts of the same set of parts along the shape. In the literature, this problem has been usually addressed by associating a context descriptor with each part. In this paper, we instead augment the transitive closure with new, undirected edges, capturing the neighbor relationships between parts. This transforms the transitive closure of the shape tree into a more general graph that we call the shape hierarchy.

**Node Attributes.** Both nodes and edges of the shape hierarchy are attributed. Node attributes are vectors whose elements describe photometric and geometric properties of the corresponding shape part. The following estimates help us define the shape properties. We estimate the contour’s mean intensity gradient, and use this vector to identify the contour’s direction – namely, the sequence of points along the shape – by the right-hand rule. The principal axis of the entire contour is estimated as the principal axis of an ellipse fitted to all points of the shape. The attribute vector of a node (i.e., shape part) includes the following properties: (1) length as a percentage of the parent length; (2) angle between the principal axes of this shape part and its parent; (3) approximation error estimated as the total area between the shape part and its associated straight line, expressed as a percentage of the area of the fitted ellipse; (4) signed approximation error is similar to the approximation error except that the total area between the shape part and its approximating straight line is computed by accounting for the sign of the intensity gradient along the shape; and (5) curvature at the two end points of the shape part. All the properties are normalized to be in $[0, 1]$.

**Edge Attributes.** The attribute of an edge in the shape hierarchy encodes the strength of the corresponding part-of or neighbor relationship. Given a directed edge between a shape part and its descendant part, the attribute of this edge is defined as the percentage that the length of the descendant makes in the length of the shape part. Thus, the shorter descendant or the longer ancestor, the smaller strength of their interaction. The attribute
of an undirected edge between two shape parts can be either 1 or 0, where 1 means that the parts have one common end point, and 0 means that the parts are not neighbors.

4 Shape Matching

Given two shapes, our goal is to identify best matching shape parts and discard dissimilar parts, so that the total cost is minimized. This cost is defined as a function of geometric, photometric, and structural properties of the matched parts, their subparts, and their neighbor parts, as explained below.

4.1 Definition of the Objective Function of Matching

Let \( H = (V, E, \psi, \phi) \) denote the shape hierarchy, where \( V = \{v\} \) and \( E = \{(v, u)\} \subseteq V \times V \) are the sets of nodes and edges, and \( \psi \) and \( \phi \) are functions that assign attributes to nodes, \( \psi : V \rightarrow [0, 1]^d \), and to edges, \( \phi : E \rightarrow [0, 1] \). Given two shapes, \( H \) and \( H' \), the goal of the matching algorithm is to find the subgraph isomorphism, \( f: U \rightarrow U' \), where \( U \subseteq V \) and \( U' \subseteq V' \), which minimizes the cost, \( C \), defined as

\[
C = \left[ \beta \sum_{v \in V} c_1(v, f(v)) + (1 - \beta) \sum_{(v,u) \in E} c_2(v, f(v), u, f(u)) \right],
\]

where \( c_1 \) is a non-negative cost function of matching nodes \( v \) and \( v' = f(v) \), and \( c_2 \) is a non-negative cost function of matching edges \( (v, u) \) and \( (v', u') \), and \( \beta \in [0, 1] \) weights their relative significance to matching. To minimize \( C \), we introduce a vector, \( X \), indexed by all node pairs \( (v, v') \in V \times V' \), whose each element \( x_{vv'} \in [0, 1] \) encodes the confidence that pair \( (v, v') \) should be matched. Matching can then be reformulated as estimating \( X \) so that \( C \) is minimized. That is, we use the standard linearization and relaxation of (1) to obtain the following quadratic program (QP):

\[
\min_X \left[ \beta A^T X + (1 - \beta) X^T B X \right],
\]

s.t. \( \forall (v, v') \in V \times V', x_{vv'} \geq 0 \), \( \forall v' \in V' \), \( \sum_{v \in V} x_{vv'} = 1 \), \( \forall v \in V \), \( \sum_{v' \in V'} x_{vv'} = 1 \),

where \( A \) is a vector of costs \( a_{vv'} \) of matching nodes \( v \) and \( v' \), and \( B \) is a matrix of costs \( b_{vu'u'} \) of matching edges \( (v, u) \) and \( (v', u') \). We define \( a_{vv'} = \frac{1}{d} \| \psi(v) - \psi(v') \|_2 \), where \( d \) is the dimensionality of the node attribute vector. Also, we define \( b_{vu'u'} \) so that matching edges of different types is prohibited, and matches between edges of the same type with similar properties are favored in (2): \( b_{vu'u'} = \infty \) if edges \( (v, u) \) and \( (v', u') \) are not of the same type; and \( b_{vu'u'} = | \phi(v, v') - \phi(u, u') | \) if edges \( (v, u) \) and \( (v', u') \) are of the same type.

The constraints in (2) are typically too restrictive, because of potentially large structural changes of \( V \) or \( E \) in \( H \) that may be caused by relatively small variations of certain shape parts. For example, suppose \( H \) and \( H' \) represent similar shapes. It may happen that node \( v \) in \( H \) corresponds to a subgraph consisting of nodes \( \{v'_1, \ldots, v'_m\} \) in \( H' \), and vice versa. Therefore, a more general many-to-many matching formulation would be more appropriate for our purposes. The literature reports a number of heuristic approaches to many-to-many matching [19][20][21], which however are developed only for weighted graphs, and thus cannot be used for our shape hierarchies that have
attributes on both nodes and edges. To relax the constraints in (2), we first match \( H \) to \( H' \), which yields solution \( X_1 \). Then, we match \( H' \) to \( H \), which yields solution \( X_2 \). The final solution, \( \tilde{X} \), is estimated as an intersection of non-zero elements of \( X_1 \) and \( X_2 \).

Formally, the constraints are relaxed as follows: (i) \( \forall (v, v') \in V \times V', x_{vv'} \geq 0 \); and (ii) \( \forall v \in V, \sum_{v' \in V'} x_{vv'} = 1 \) when matching \( H \) to \( H' \); and \( \forall v' \in V', \sum_{v \in V} x_{vv'} = 1 \) when matching \( H' \) to \( H \).

### 4.2 Convexification of the Objective Function of Matching

The QP in (2) is in general non-convex, and defines a matching space of typically \( 10^4 \) possible node pairs in our experiments. In order to efficiently find a solution, we convexify the QP. This significantly reduces the number of matching candidates.

Given \( H \) and \( H' \) to be matched, for each node \( v \in V \) of \( H \), we identify those matching candidates \( v' \in V' \) of \( H' \) that form the region of support of the lower convex hull of costs \( \{a_{vv'}\}_{v' \in V'} \), as illustrated in Fig. 2. Let \( \tilde{V}'(v) \subset V' \) denote this region of support of the convex hull, and let \( \hat{V}'(v) \subset V' \) denote the set of true matches of node \( v \) that minimize the QP in (2) (i.e., the solution). Then, by definition, we have that \( V'(v) \subseteq \hat{V}'(v) \), i.e., the true matches must be located in the region of support of the convex hull. It follows, that for each node \( v \in V \), we can discard those matching candidates from \( V' \) that do not belong to \( \hat{V}'(v) \). In our experiments, we typically obtain \( |\hat{V}'(v)| \ll |V'| \), which leads to a dramatic reduction of the dimensionality of the original matching space, \(|V \times V'|\).

In summary, we compute \( \tilde{A}, \tilde{X}, \tilde{B} \) from original \( A, X, B \), respectively, by deleting all their elements \( a_{vv'}, x_{vv'}, b_{vv'uv'w} \), for which \( v' \notin \hat{V}'(v) \). Then, we use the standard interior-reflective Newton method to solve the following program:

\[
\min_{\tilde{X}} \left[ \beta \tilde{A}^T \tilde{X} + (1 - \beta) \tilde{X}^T \tilde{B} \tilde{X} \right],
\]

s.t. \( \forall (v, v') \in V \times \hat{V}'(v), \ x_{vv'} \geq 0, \ \forall v \in V, \ \sum_{v' \in \hat{V}'(v)} x_{vv'} = 1. \) (3)

### 5 Results

This section presents the experimental evaluation of our approach on the standard MPEG-7 and Brown shape datasets [12]. MPEG-7 has 1400 silhouette images showing 70 different object classes, with 20 images per object class, as illustrated in Fig. 3. MPEG-7 presents many challenges due to a large intra-class variability within each class, and small differences between certain classes. The Brown shape dataset has 11 examples from 9 different object categories, totaling 99 images. This dataset introduces
additional challenges, since many of the shapes have missing parts (e.g., due to occlusion), and the images may contain clutter in addition to the silhouettes, as illustrated in Figs. 1, 4, 5. We use the standard evaluation on both datasets. For every silhouette in MPEG-7, we retrieve the 40 best matches, and count the number of those that are in the same class as the query image. The retrieval rate is defined as the ratio of the total number of correct hits obtained and the best possible number of correct hits. The latter number is \(1400 \cdot 20\). Also, for each shape in the Brown dataset, we first retrieve the 10 best matches, then, check if they are in the same class as the query shape, and, finally, compute the retrieval rate, as explained above. Input to our algorithm consists of two parameters: the fine-resolution level (approximation error defined in Sec. 5) of representing the contour, and \(\beta\). For silhouettes in both datasets, and the approximation error (defined in Sec. 5) equal to 1%, we obtain shape hierarchies with typically 50-100 nodes, maximum hierarchy depths of 5-7, and maximum branching factors of 4-6. For every query shape, the distances to other shapes are computed as the normalized total matching cost, \(D\), between the query and these other shapes. If \(X\) is the solution of our quadratic programming, then \(D = \left[\beta A^T X + (1 - \beta) X^T B X\right] / \left[|V| + |V'|\right]\), where \(|V|\) is the total number of nodes in one shape hierarchy. Matching two shape hierarchies takes about 5-10 sec in MATLAB on a 3.1GHz, 1GB RAM PC.

**Qualitative evaluation.** Fig. 3 shows a few examples of our shape retrieval results on MPEG-7. From the figure, our approach makes errors mainly due to the non-optimal pre-setting of the fine-resolution level at which contours are represented by the shape hierarchy. Also, some object classes in the MPEG-7 are characterized by multiple

![Fig. 3. MPEG-7 retrieval results on three query examples and comparison with [6]. For each query, we show 11 retrieved shapes with smallest to highest cost. (top) Results of [6]. (bottom) Our results. Note that for deer we make the first mistake in 6th retrieval, and then get confused with shapes whose parts are very similar to those of deer. Mistakes for other queries usually occur due to missing to capture fine details of the curves in the shape hierarchy in our implementation.](image-url)
disjoint contours, whereas our approach is aimed at matching only one single contour at a time. Next, Fig. 4 shows four example pairs of silhouettes from the same class, and their matched shape parts. Similar shape parts at multiple scales got successfully matched in all cases, as expected. Fig. 5 presents two example pairs of silhouettes that belong to different classes. As in the previous case, similar shape parts got successfully matched; however, since there are fewer similar than dissimilar parts the normalized total matching cost in this case is larger. This helps discriminate between the shapes from different classes in the retrieval.

Quantitative evaluation. To evaluate the sensitivity of our approach to input parameter $\beta$, we compute the average retrieval rate on the Brown dataset as a function of input $\beta = 0.1 : 0.1 : 0.9$. The maximum retrieval rate of 99% is obtained for $\beta=0.4$, while for $\beta = \{0.3, 0.5, 0.6\}$ we obtain the rate of 98%, while for other values of $\beta$ the retrieval rate gracefully decreases. This suggests that both intrinsic properties of shape parts and their spatial relations are important for shape matching, and that our
algorithm is relatively insensitive to small changes of $\beta$ around 0.4. However, as any hierarchical approach, ours also seems to be sensitive to the right choice of the finest resolution at which the shape is represented. As mentioned above, different values of this input parameter may result in large variations of the number of nodes in the shape hierarchy, which, in turn, cause changes in computing the normalized total matching cost. If the right choice is selected separately for each class of MPEG-7, using validation data, then we obtain the retrieval rate of 88.3%. If this parameter is set to 1%, as stated above, for all classes, then our performance drops to 84.3%. This is comparable to the state of the art that achieves the rates of 85.40% in [6] and 87.70% in [3]. Table 1 summarizes our retrieval rates on the Brown dataset after first top 1 to 10 retrievals, for $\beta = 0.4$ and shape-resolution level fixed over all classes. Again, this retrieval improves if we select a suitable value for the resolution parameter for each class separately, using validation data.

### 6 Conclusion

Matching deformable shapes is difficult since they may be perceptually similar, but have certain parts very different or even missing. We have presented an approach aimed at robust matching of deformable shapes by identifying multiscale salient shape parts, and accounting for their intrinsic properties, and part-of and neighbor relationships. Experimental evaluation of the proposed hierarchical shape representation and shape matching via minimizing a quadratic cost has demonstrated that the approach robustly deals with large variations or missing parts of perceptually similar shapes.

### References

Edition within a Graph Kernel Framework for Shape Recognition

François-Xavier Dupé* and Luc Brun

GREYC UMR CNRS 6072,
ENSICAEN-Université de Caen Basse-Normandie,
14050 Caen France
{francois-xavier.dupe,luc.brun}@greyc.ensicaen.fr

Abstract. A large family of shape comparison methods is based on a medial axis transform combined with an encoding of the skeleton by a graph. Despite many qualities this encoding of shapes suffers from the non continuity of the medial axis transform. In this paper, we propose to integrate robustness against structural noise inside a graph kernel. This robustness is based on a selection of the paths according to their relevance and on path editions. This kernel is positive semi-definite and several experiments prove the efficiency of our approach compared to alternative kernels.

Keywords: Shape, Skeleton, Support Vector Machine, Graph Kernel.

1 Introduction

The skeleton is a key feature within the shape recognition framework [1,2,3]. Indeed, this representation holds many properties: it is a thin set, homotopic to the shape and invariant under Euclidean transformations. Moreover, any shape can be reconstructed from the maximal circles of the skeleton points.

The set of points composing a skeleton does not highlight the structure of a shape. Consequently, the recognition step is usually based on a graph comparison where graphs encode the main properties of the skeletons. Several encoding systems have been proposed: Di Ruberto [4] proposes a direct translation of the skeleton to the graph using many attributes. Siddiqi [2] proposes a graph which characterises both structural properties of a skeleton and the positive, negative or null slopes of the radius of the maximal circles along a branch. Finally this last encoding has been improved and extended to 3D by Leymarie and Kimia [6].

The recognition of shapes using graph comparisons may be tackled using various methods. A first family of methods is based on the graph edit distance which is defined as the minimal number of operations required to transform the graph encoding the first shape into the graph encoding the second one [2,3]. Another method, introduced by Pelillo [1], transforms graphs into trees and then

* This work is performed in close collaboration with the laboratory Cycéron and is supported by the CNRS and the région Basse-Normandie.
models the tree matching problem as a maximal clique problem within a specific association graph. A last method proposed by Bai and Latecki \[7\] compares paths between end-node (node with only one neighbor) after a matching task on the end-nodes. Contrary to previously mentioned approaches this last method can deal with loops and may thus characterize holed shapes.

All the above methods perform in the graph space which almost contains no mathematical structure. This forbids many common mathematical tools like the mean graph of a set which has to be replaced by its median. A solution consists to project graphs into a richer space. Graph kernels provide such an embedding: by using appropriate kernels, graphs can be mapped either explicitly or implicitly into a vector space whose dot product corresponds to the kernel function.

Most famous graph kernels are the random walk kernel, the marginalized graph kernel and the geometric kernel \[8\]. A last family of kernels is based on the notion of bag of paths \[9\]. These methods describe each graph by a subset of its paths, the similarity between two graphs being deduced from the similarities between their paths. Path similarity is based on a comparison between the edges and nodes attributes of both paths.

However, skeletonization is not a continuous process and small perturbations of a shape may produce ligatures and spurious branches. Graph kernels may in this case lead to inaccurate comparisons. Neuhaus and Bunke have proposed several kernels (e.g. \[10\]) based on the graph edit distance in order to reduce the influence of graph perturbations. However the graph edit distance does not usually fulfills all the properties of a metric and the design of a definite positive kernel from such a distance is not straightforward. Our approach is slightly different. Indeed, instead of considering a direct edit distance between graphs, our kernel is based on a rewriting process applied on the bag of paths of two graphs. The path rewriting follows the same basic idea than the string edit distance but provides a definite positive kernel between paths.

This paper follows a first contribution \[11\] where we introduced the notion of path rewriting within the graph kernel framework. It is structured as follows: first, we recall how to construct a bag of path kernel \[9,11\] (Section 2). Then, we propose a graph structure (Section 3) which encodes both the structure of the skeleton and its major characteristics. This graph contains a sufficient amount of information for shape reconstruction. We then extend the edition operations (Section 4) by taking into account all the attributes and by controlling the effect of the edition on them. Finally, we present experiments (Section 5) in order to highlight the benefit of the edition process.

## 2 Bag of Path Kernel

Let us consider a graph $G = (V, E)$ where $V$ denotes the set of vertices and $E \subset V \times V$ the set of edges. A bag of paths $P$ associated to $G$ is defined as a set of paths of $G$ whose cardinality is denoted by $|P|$. Let us denote by $K_{\text{path}}$ a generic path kernel. Given two graphs $G_1$ and $G_2$ and two paths $h_1 \in P_1$ and $h_2 \in P_2$ of respectively $G_1$ and $G_2$, $K_{\text{path}}(h_1, h_2)$ may be interpreted as
a measure of similarity between $h_1$ and $h_2$. The aim of a bag of path kernel consists to aggregate all these local measures between pairs of paths into a global similarity measure between the two graphs. Such a kernel differs from random walk kernels where all the paths of the two graphs are compared.

### 2.1 Change Detection Kernel

Desobry [12] proposed a general approach for the comparison of two sets which has straightforward applications in the design of a bag of path kernel (bags are sets). The two bags are modelled as the observation of two sets of random variables in a feature space.

Desobry proposes to estimate a distance between the two distributions without explicitly building the pdf of the two sets. The considered feature space is based on a normalised kernel ($K(h, h') = K_{path}(h, h')/\sqrt{(K_{path}(h, h)K_{path}(h', h'))}$). Using such a kernel we have $\|h\|^2_K = K(h, h) = 1$ for any path. The image in the feature space of our set of paths lies thus on an hypersphere of radius 1 centered at the origin (Fig. 1(a)). Using the one-class $\nu$-SVM, we associate a set of paths to a region on this sphere. This region corresponds to the density support estimate of the set of paths’ unknown pdf.

Once the two density supports are estimated, the one-class SVM yields $w_1$ (resp. $w_2$), the mean vector, and $\rho_1$ (resp. $\rho_2$), the ordinate at the origin, for the first bag (resp. the second bag). In order to compare the two mean vectors $w_1$ and $w_2$, we define the following distance function:

$$d_{mean}(w_1, w_2) = \arccos \left( \frac{w_1^t K_{1,2} w_2}{\|w_1\| \|w_2\|} \right),$$

where $K_{1,2}(i, j) = K(h_i, h_j)$, $h_i \in P_1$, $h_j \in P_2$ and $w_1^t K_{1,2} w_2$ is the scalar product between $w_1$ and $w_2$. This distance corresponds to the angle $\alpha$ between the two mean vectors $w_1$ and $w_2$ of each region (Fig. 1(a)). Then we define the kernel between two bags of path $P_1$ and $P_2$ as $1)$ the product of a Gaussian RBF kernel associated to $d_{mean}(w_1, w_2)$ and $2)$ a Gaussian RBF kernel associated to

![Fig. 1. (a) Separating two sets using one-class SVM. The symbols $(w_1, \rho_1)$ and $(w_2, \rho_2)$ denote the parameters of the two hyperplanes which are represented by dashed lines. Influence of small perturbation on the skeleton (in black) ((b),(c) and (d)).](image)
the difference between the two coordinates at the origin ($\rho_1$ and $\rho_2$):

$$K_{\text{change}}(P_1, P_2) = \exp \left( \frac{-d_{\text{mean}}^2(w_1, w_2)}{2\sigma_{\text{mean}}^2} \right) \exp \left( \frac{-(\rho_1 - \rho_2)^2}{2\sigma_{\text{origin}}^2} \right).$$ (2)

Finally, we define the kernel between two graphs $G_1$, $G_2$ as the kernel between their two bags of path: $K_{\text{change}}(G_1, G_2) = K_{\text{change}}(P_1, P_2)$.

The distance between the mean vector $s$ is a metric based on a normalized scalar product combined with arccos which is bijective on $[0, 1]$. However, the relationship between the couple $(w, \rho)$ and the bag of path being not bijective, the final kernel between bags is only semi positive-definite [13]. Though, in all our experiments run so far the Gram matrices associated to the bags of paths were positive-definite.

2.2 Path Kernel

The above bag of path kernel is based on a generic path kernel $K_{\text{path}}$. A kernel between two paths $h_1 = (v_1, \ldots, v_n)$ and $h'_1 = (v'_1, \ldots, v'_p)$ is classically [14] built by considering each path as a sequence of nodes and a sequence of edges. This kernel denoted $K_{\text{classic}}$ is defined as 0 if both paths have not the same size and as follows otherwise:

$$K_{\text{classic}}(h, h') = K_v(\varphi(v_1), \varphi(v'_1)) \prod_{i=2}^{\vert h \vert} K_e(\psi(e_{v_{i-1}v_i}), \psi(e'_{v'_{i-1}v'_i})) K_v(\varphi(v_i), \varphi(v'_i)),$$ (3)

where $\varphi(v)$ and $\psi(e)$ denote respectively the vectors of features associated to the node $v$ and the edge $e$. The terms $K_v$ and $K_e$ denote two kernels for respectively node’s and edge’s features. For the sake of flexibility and simplicity, we use Gaussian RBF kernels based on the distance between the attributes defined in section 3.2.

3 Skeleton-Based Graph

3.1 Graph Representations

Medial-axis based skeleton are built upon a distance function whose evolution along the skeleton is generally modeled as a continuous function. This function presents important changes of slope mainly located at the transitions between two parts of the shape. Based on this remark Siddiqi and Kimia distinguish three kind of branches within the shock graph construction scheme [2]: branches with positive, null or negative slopes. Nodes corresponding to these slope transitions are inserted within the graph. Such nodes may thus have a degree 2. Finally, edges are directed using the slope sign information.

Compared to the shock graph representation, we do not use oriented edges since small positive or negative values of the slope may change the orientation of an edge and thus alter the graph representation. On the other hand, our set of nodes corresponds to junction points and to any point encoding an important
change of slope of the radius function. Such a significant change may encode a change from a positive to a negative slope but also an important change of slope with a same sign (Fig. 2(a)). Encoding these changes improves the detection of the different parts of the shape. The main difficulty remains the detection of the slope changes due to the discrete nature of the data. The slopes are obtained using regression methods based on first order splines [15]. These methods are robust to discrete noise and first order splines lead to a continuous representation of the data. Moreover, such methods intrinsically select the most significant slopes using a stochastic criterion. Nodes encoding slope transitions are thus located at the junctions (or knot) between first order splines.

3.2 Attributes

The graph associated to a shape only provides information about its structural properties. Additional geometrical properties of the shape may be encoded using node and edge attributes. From a structural point of view, a node represents a particular point inside the shape skeleton and an edge a branch. However, a branch also represents the set of points of the shape which are closer to the branch than any other branch. This set of points is defined as the influence zone of the branch and can be computed using SKIZ transforms [16]. Descriptors computed from the influence zone are called local, whilst the ones computed from the whole shape are called global. In [3] Goh introduces this notion and points out that an equilibrium between local and global descriptors is crucial for the efficiency of a shape matching algorithm. Indeed local descriptors provide a robustness against occlusions, while global ones provide a robustness against noise.

We have thus selected a set of attributes which provides an equilibrium between local and global features. Torsello in [17] proposes as edge attribute an approximation of the perimeter of the boundary which contributes to the formation of the edge, normalized by the approximated perimeter of the whole shape. Suard proposes [9] as node attribute the distance between the node position and the gravity center of the shape divided by the square of the shape area. These two attributes correspond to our global descriptors.
Goh proposes several local descriptors for edges based on the evolution of the radius of the maximal circle along a branch. For each point \((x(t), y(t))\) of a branch, \(t \in [0, 1]\), we consider the radius \(R(t)\) of its maximal circle. In order to normalize the data, the radius is divided by the square root of the area of the influence zone of the branch. We also introduce \(\alpha(t)\), the angle formed by the tangent vector at \((x(t), y(t))\) and the \(x\)-axis. Then we consider \((a_k)_{k \in \mathbb{N}}\) and \((b_k)_{k \in \mathbb{N}}\) the coefficients of two regression polynomials that fit respectively \(R(t)\) and \(\alpha(t)\) in the least square sense. If both polynomials are of sufficient orders, the skeleton can be reconstructed from the graph and so the shape (Section 1). Following Goh [3], our two local descriptors are defined by: \(\sum_k a_k/k\) and \(\sum_k b_k/k\).

The distance associated to each attribute is defined as the absolute value of the difference between the values \(a\) and \(b\) of the attribute: \(d(a, b) = |a - b|\). As the attributes are normalized, the distances are invariant to change of scale and rotation. Such distances are used to define the Gaussian RBF kernels (\(\exp\left(-\frac{d^2}{2\sigma^2}\right)\)) used to design \(K_{path}\) (Section 2.2).

### 4 Hierarchical Kernels

The biggest issue with skeleton-based graph representation is the non-negligible effect of small perturbations on the shape [2]: Fig. 1 shows two deformations of the skeleton of a circle (Fig. 1(b)) one induced by a small bump (Fig. 1(c)) and one by an elongation (Fig. 1(d)). On complex shapes, severe modifications of the graphs may occur and lead to inaccurate comparisons.

From a structural point of view, perturbations like bumps (Fig. 1(c)) create new nodes and edges. In contrast, the principal effect of an elongation (Fig. 1(d)) is either the addition of an edge inside the graph or the extension of an existing edge. So shape noise mainly induces two effects on paths: addition of nodes (Fig. 1(c)) and addition of edges (Fig. 1(d)). This leads to the two editions operations: node suppression and edge contraction. Note that, as the compared structure are paths, the relevance of these operations should be evaluated according to the path under study.

#### 4.1 Elementary Operations on Path

The node suppression operation removes a node from the path and all the graph structures that are connected to this path by this node. Within the path, the two edges incident to the nodes are then merged. This operation corresponds to the removal of a part of the shape: for example, if we remove the node 2 in Fig. 2(b1), a new shape similar to Fig. 2(b2) is obtained.

The edge contraction operation contracts an edge and merges its two extremity nodes. This results in a contraction of the shape: for example, if we contract the edge \(e_{1,2}\) of the shape in Fig. 2(b1) then the new shape will be similar to Fig. 2(b3).
Since each operation is interpreted as a shape transformation, the global descriptors must be updated. From this point of view our method may be considered as a combination of the methods of Sebastian [2] and Goh [3] who respectively use local descriptors with edit operations and both local and global descriptors but without edit operations.

4.2 Edition Cost

In order to select the appropriate operation, an edition cost is associated to each operation. Let us consider an attribute weight associated to each edge of the graph which encodes the relevance of its associated branch. We suppose that this attribute is additive: the weight of two consecutive edges along a path is the sum of both weights.

Note that, we consider the maximal spanning tree \( T \) of the graph \( G \). As skeletonization is an homotopic transform, a shape with no hole yields \( T = G \). Let us consider a path \( h = (v_1, \ldots, v_n) \) within \( T \). Now, an edition cost is assigned to both operations within \( h \):

- Let us consider a node \( v_i \), \( i \in \{2, \ldots, n-1\} \) of the path \( h \) (extremity nodes are not considered). The cost of the node suppression operation on \( v_i \) must reflect two of its properties: 1) the importance of the sub-trees of \( T \) connected to the path by \( v_i \) and 2) the importance of the slope changes (Section 3.1) between the two branches respectively encoded by the edges \( e_{v_{i-1}v_i} \) and \( e_{v_iv_{i+1}} \).

  The relevance of a sub-tree is represented by its total weight: for each neighbor \( v \) of \( v_i \), \( v \notin h \), we compute the weight \( W(v) \) defined as the addition of the weight of the tree rooted on \( v \) in \( T \setminus \{e_{v_i,v}\} \) and the weight of \( e_{v_i,v} \). This tree is unique since \( T \) is a tree. The weight of the node \( v_i \) is then defined as the sum of weights \( W(v) \) for all neighbors \( v \) of \( v_i \) (\( v \notin h \)) and is denoted by \( \omega(v_i) \).

  We encode the relevance of a slope change by the angle \( \beta(v_i) \) formed by the slope vectors associated to \( e_{v_{i-1}v_i} \) and \( e_{v_iv_{i+1}} \). An high value of \( \beta(v_i) \) encodes a severe change of slopes and conversely. Since slopes are approximated using first-order polynomials (section 3.1), the angle \( \beta(v_i) \) is given by

  \[
  \beta(v_i) = \arccos\left(\frac{1+a_1\cdot a'_1}{\sqrt{1+a_1^2}\sqrt{1+a'_1^2}}\right)
  \]

  where \( a_1 \) and \( a'_1 \) are the first order coefficients of the regression polynomials.

  Finally the edition cost of the suppression of a node is defined by \( (1 - \gamma)\omega(v_i) + \gamma\beta(v_i)/\pi \), where \( \gamma \) is a tuning variable.

- The cost of the edge contraction operation encodes the importance of the edge inside the graph \( T \), this is the purpose of the weight. So, the edition cost of contracting an edge is defined as its weight.

Concerning weight any additive measure encoding the relevance of a skeleton’s branch may be used. We choose to use the normalized perimeter as computed by Torsello [17], because of its resistance to noise on the shape boundary.
4.3 Edition Path Kernel

Let us denote by $\kappa$ the function which applies the cheapest operation on a path and $D$ the maximal number of reductions. The successive applications of the function $\kappa$ associate to each path $h$ a sequence of reduced paths $(h, \kappa(h), \ldots, \kappa^D(h))$. Each $\kappa^k(h)$ is associated to a cost: $\text{cost}_k(h)$ defined as the sum of the costs of the $k$ operations yielding $\kappa^k(h)$ from $h$. Using $K_{\text{classic}}$ for the path comparison, we define the kernel $K_{\text{edit}}$ as a sum of kernels between reduced paths. Given two paths $h$ and $h'$, the kernel $K_{\text{edit}}(h, h')$ is defined as:

$$K_{\text{edit}}(h, h') = \frac{1}{D+1} \sum_{k=0}^{D} \sum_{l=0}^{D} \exp \left( -\frac{\text{cost}_k(h) + \text{cost}_l(h')}{{2\sigma}_{\text{cost}}} \right) K_{\text{classic}}(\kappa^k(h), \kappa^l(h')), \quad (4)$$

where $\sigma_{\text{cost}}$ is a tuning variable. This kernel is composed of two parts: a scalar product of the edition costs in a particular space and a path kernel. For a small value of $\sigma_{\text{cost}}$ the behavior of the kernel will be close to $K_{\text{classic}}$ as only low editions cost will contribute to $K_{\text{edit}}(h, h')$. Conversely, for a high value every editions will contribute to $K_{\text{edit}}(h, h')$ with an approximately equal importance.

The kernel $K_{\text{classic}}$ is a tensor product kernel based on positive-definite kernels (Section 2.2), so it is positive-definite. The kernel over edition costs is constructed from a scalar product and is thus positive-definite. These two last kernels form a tensor product kernel. Finally $K_{\text{edit}}$ is proportional (by a factor $D+1$) to a $R$-convolution kernel [18, Lemma 1], thus this kernel is positive-definite.

5 Experiments

For the following experiments, we defined the importance of a path as the sum of the weights of its edges. For each graph, we first consider all its paths composed of at most 7 nodes and we sort them according to their importance using a descending order. The bag of paths is then constructed using the first 5 percent of the sorted paths. For all the experiments, the tuning variable of the deformation cost $\gamma$ (Section 4.2) is set to 0.5.

The first experiment consists in an indexation of the shapes using the distances induced by the kernels, i.e. $d(G, G') = k(G, G) + k(G', G') - 2k(G, G')$ where $k$ is a graph kernel. The different $\sigma$ of the attributes RBF kernels involved in $K_{\text{classic}}$ (Section 3.2) are fixed as follows: $\sigma_{\text{perimeter}} = \sigma_{\text{radius}} = \sigma_{\text{orientation}} = 0.1$ and $\sigma_{\text{gravity center}} = 0.2$. Note that $K_{\text{classic}}$ constitutes the basis of all the kernels defined below. The parameters of $K_{\text{change}}$ are set to: $\sigma_{\text{mean}} = 1.0$, $\sigma_{\text{origin}} = 20$ and $\nu = 0.9$. The maximal number of editions is fixed to 6. Let us consider the class tool from the LEMS database [19] of 99 shapes with 11 elements per class. Two kind of robustness are considered: robustness against ligatures and perturbations and robustness against erroneous slope nodes. Ligatured skeletons of the shapes are created by varying the threshold parameter $\zeta$ of the skeletonization algorithm [17], high values lead to ligatured skeletons while low value tend to remove relevant branches. Skeletons with erroneous slope nodes are created by varying the parameter of our slope detection algorithm. This detection is based on the BIC criterion which uses the standard error of the noise $\sigma_{\text{BIC}}$. 
A small value of $\sigma_{BIC}$ makes the criterion sensitive to small changes of slopes and gives many slope nodes, while high value makes the criterion insensitive to slope changes. Four kernels are compared: random walk kernel [8], $K_{\text{change}}$ with $K_{\text{classic}}$ (denoted as $K_{\text{change,classic}}$) and 2 kernels using $K_{\text{change}}$ with $K_{\text{edit}}$ (with $\sigma_{\text{cost}} = 0.1$ for $K_{\text{change,edit1}}$ and $\sigma_{\text{cost}} = 0.2$ for $K_{\text{change,edit2}}$). Using the distances induced by the kernels, shapes are sorted in ascending order according to their distance to the perturbed tool. Fig. 3(a) shows the mean number of tools inside the first 11 sorted shapes for an increasing value of $\sigma_{BIC}$. Fig. 3(b) shows the same number but for a decreasing threshold value $\zeta$. The two edition kernels show a good resistance to perturbations and ligatures as they get almost all the tools for each query. Their performances slightly decrease when shapes become strongly distorted. The kernel $K_{\text{change,classic}}$ gives the worst results as the reduction of the bag of paths leads to paths of different lengths which cannot be compared with $K_{\text{classic}}$ (Section 2.2). The random walk kernel is robust against slight perturbations of the shapes but cannot deal with severe distortion.

In the second experiment, we strain kernels by separating 49 dogs from 49 cats using a $\nu$-SVM. The three considered kernels are $K_{\text{change,classic}}$, $K_{\text{change,edit}}$ (with $\sigma_{\text{cost}} = 0.5$) and random walk. The different $\sigma$ of the attributes RBF kernels (Section 3.2) are fixed as follows: $\sigma_{\text{perimeter}} = \sigma_{\text{radius}} = \sigma_{\text{orientation}} = 0.1$ and $\sigma_{\text{gravity center}} = 0.5$. The parameters of $K_{\text{change}}$ are set to: $\sigma_{\text{mean}} = 5.0$, $\sigma_{\text{origin}} = 20$ and $\nu = 0.9$. We compute the ROC curves produced by kernels using a 10-fold cross-validation. Fig. 3(c) presents the three ROC curves. The random walk kernel gives correct results, whilst the $K_{\text{change,classic}}$ kernel confirms its poor performance. The $K_{\text{change,edit}}$ kernel shows the best performances and a behaviour similar to the random walk kernel. Furthermore, on our computer a Core Duo 2 at 2GHz, the computational burden of the 98x98 Gram matrix is of approximately 23 minutes for $K_{\text{change,edit}}$ and of 2.5 hours for the random walk kernel.
6 Conclusion

We have defined in this paper a positive-definite kernel for shape classification which is robust to perturbations. Our bag of path contains the more important paths of a shape below a given length in order to only capture the main information about the shape. Only the $K_{edit}$ kernel provides enough flexibility for path comparison and gives better results than the classical random walk kernel. In a near future, we would like to improve the selection of paths. An extension of the edition process on graphs is also planned.

References

Coarse-to-Fine Matching of Shapes Using Disconnected Skeletons by Learning Class-Specific Boundary Deformations

Aykut Erdem and Sibel Tari

1 Dipartimento di Informatica, Università Ca’ Foscari di Venezia
Via Torino 155, Mestre, Venezia, 30172, Italy
erdem@dsi.unive.it

2 Department of Computer Engineering, Middle East Technical University
Inonu Bulvari, 06531, Ankara, Turkey
stari@metu.edu.tr

Abstract. Disconnected skeleton [1] is a very coarse yet a very stable skeleton-based representation scheme for generic shape recognition in which recognition is performed mainly based on the structure of disconnection points of extracted branches, without explicitly using information about boundary details [2,3]. However, sometimes sensitivity to boundary details may be required in order to achieve the goal of recognition. In this study, we first present a simple way to enrich disconnected skeletons with radius functions. Next, we attempt to resolve the conflicting goals of stability and sensitivity by proposing a coarse-to-fine shape matching algorithm. As the first step, two shapes are matched based on the structure of their disconnected skeletons, and following to that, the computed matching cost is re-evaluated by taking into account the similarity of boundary details in the light of class-specific boundary deformations which are learned from a given set of examples.

1 Introduction

There is a long history of research in computer vision on representing generic shape since shape information is a very strong visual clue in recognizing and classifying objects. A generic shape representation should be insensitive to not only geometric similarity transformations (i.e. translation, rotation, and scaling) but also visual transformations such as occlusion, deformation and articulation of parts. Since their introduction by Blum in [4], local symmetry axis based representations (commonly referred to as shape skeletons), have attracted and still attracts many scientists in the field, and became a superior alternative to boundary-based shape representations. These representation schemes naturally capture part structure by modeling any given shape via a set of axial curves, each of which explicitly represents some part of the shape. Once the relations among extracted shape primitives, i.e. the skeleton branches, are expressed in terms of a graph or a tree data structure (e.g. [5,6,7]), resulting shape descriptions are insensitive to articulations and occlusions.