Preface

Graph theory has strong historical roots in mathematics, especially in topology. Its birth is usually associated with the "four-color problem" posed by Francis Guthrie in 1852,[1] but its real origin probably goes back to the Seven Bridges of Königsberg problem proved by Leonhard Euler in 1736.[2] A computational solution to these two completely different problems could be found after each problem was abstracted to the level of a *graph model* while ignoring such irrelevant details as country shapes or cross-river distances. In general, a graph is a nonempty set of points (*vertices*) and the most basic information preserved by any graph structure refers to adjacency relationships (*edges*) between some pairs of points. In the simplest graphs, edges do not have to hold any attributes, except their endpoints, but in more sophisticated graph structures, edges can be associated with a direction or assigned a label. Graph vertices can be labeled as well. A graph can be represented graphically as a drawing (vertex = dot, edge = arc), but, as long as every pair of adjacent points stays connected by the same edge, the graph vertices can be moved around on a drawing without changing the underlying graph structure.

The expressive power of the graph models placing a special emphasis on connectivity between objects has made them the models of choice in chemistry, physics, biology, and other fields. Their increasing popularity in the areas of computer vision and pattern recognition can be easily explained by the graphs' ability to represent complex visual patterns on one hand and to keep important structural information, which may be relevant for pattern recognition tasks, on the other hand. This is in sharp contrast with the more conventional feature vector or attribute-value representation of patterns where only unary measurements – the features, or equivalently, the attribute values – are used for object representation. Graph representations also have a number of invariance properties that may be very convenient for certain tasks.

[1] Is it possible to color, using only four colors, any map of countries in such a way as to prevent two bordering countries from having the same color?

[2] Given the location of seven bridges in the city of Königsberg, Prussia, Euler has proved that it was not possible to walk with a route that crosses each bridge exactly once, and return to the starting point.

As already mentioned, we can rotate or translate the drawing of a graph arbitrarily in the two-dimensional plane, and it will still represent the same graph. Moreover, we can stretch out or shrink its edges without changing the underlying graph. Hence graph representations have an inherent invariance with respect to translation, rotation and scaling – a property that is desirable in many applications of image analysis. On the other hand, we have to pay a price for the enhanced representational capabilities of graphs, viz. the increased computational complexity of many operations on graphs. For example, while it takes only linear time to test two feature vectors or two tuples of attribute-value pairs, for identity, all available algorithms for the equivalent operation on general graphs, i.e., graph isomorphism, are of exponential complexity. Nevertheless, there are numerous applications where the underlying graphs are relatively small, such that algorithms of exponential complexity are applicable. In other problem domains, heuristics can be found that cut significant amounts of the search space, thus rendering algorithms with a reasonably high speed. Last but not least, for more or less all common graph operations needed in pattern recognition and machine vision, approximate algorithms have become available meanwhile, which can be substituted for their exact versions. As a matter of experience, often the performance of the overall task is not compromised by using an approximate algorithm rather than an optimal one.

This book intends to cover a representative, but in no way exclusive, set of novel graph-theoretic methods for complex computer vision and pattern recognition tasks. The book is divided into three parts, which are briefly described below.

Part I includes three chapters applying graph theory to low-level processing of digital images. The first chapter by Walter G. Kroptasch, Yll Haxhimusa, and Adrian Ion presents a new method for partitioning a given image into a hierarchy of homogeneous areas (“segments”) using graph pyramids. A graphical model framework for image segmentation based on the integration of Markov random fields (MRFs) and deformable models is introduced in the chapter by Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas. In the third chapter, Alain Bretto studies the relationship between graph theory and digital topology, which deals with topological properties of 2D and 3D digital images.

Part II presents four chapters on graph-theoretic learning algorithms for high-level computer vision and pattern recognition applications. First, a survey of graph based methodologies for pattern recognition and computer vision is presented by D. Conte, P. Foggia, C. Sansone, and M. Vento. Then Gabriel Valiente introduces a series of computationally efficient algorithms for testing graph isomorphism and related graph matching tasks in pattern recognition. Sebastien Sorlin, Christine Solnon, and Jean-Michel Jolion propose a new graph distance measure to be used for solving graph matching problems. Joseph Potts, Diane J. Cook, and Lawrence B. Holder describe an approach, implemented in a system called Subdue, to learning patterns in relational data represented as a graph.

Finally, Part III provides detailed descriptions of several applications of graph-based methods to real-world pattern recognition tasks. Thus, Gian Luca Marcialis, Fabio Roli, and Alessandra Serrau present a critical review of the main graph-based and structural methods for fingerprint classification while comparing them with the classical statistical methods. Horst Bunke et al. present a new method to visualize a time series of graphs, and show potential applications in computer network monitoring and abnormal event detection. In the last chapter, A. Schenker, H. Bunke, M. Last, and A. Kandel describe a clustering method that allows the use of graph-based representations of data instead of the traditional vector-based representations.

We believe that the chapters included in our volume will serve as a foundation for a variety of useful applications of the graph theory to computer vision, pattern recognition, and related areas. Our additional goal is to encourage more research studies that will deal with the methodological challenges in applied graph theory outlined by this book authors.

October 2006

Abraham Kandel
Horst Bunke
Mark Last

http://www.springer.com/978-3-540-68019-2

Applied Graph Theory in Computer Vision and Pattern Recognition
Kandel, A.; Bunke, H.; Last, M. (Eds.)
2007, X, 266 p., Hardcover
ISBN: 978-3-540-68019-2