Luc Brun
Mario Vento (Eds.)

LNCS 3434

Graph-Based Representations in Pattern Recognition

5th IAPR International Workshop, GbRPR 2005
Poitiers, France, April 2005
Proceedings

Springer

Luc Brun Mario Vento (Eds.)

Graph-Based Representations in Pattern Recognition

5th IAPR International Workshop, GbRPR 2005
Poitiers, France, April 11-13, 2005
Proceedings

Springer

Volume Editors

Luc Brun
GREYC, ENSICAEN
6 Boulevard du Maréchal Juin, 14050 Caen, France
E-mail: luc.brun@greyc.ensicaen.fr

Mario Vento
Università di Salerno, MIVIA Lab
Dipartimento di Ingegneria dell'Informazione ed Ingegneria Elettrica
Via Ponte Don Melillo, 1, 84084 Fisciano (SA), Italy
E-mail: mvento@unisa.it

Library of Congress Control Number: 2005922822

CR Subject Classification (1998): I.5, I.4, I.3, G.2.2, E.1

ISSN 0302-9743
ISBN-10 3-540-25270-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-25270-8 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11405405 06/3142 5 4 3 2 1 0

Preface

Many vision problems have to deal with different entities (regions, lines, line junctions, etc.) and their relationships. These entities together with their relationships may be encoded using graphs or hypergraphs. The structural information encoded by graphs allows computer vision algorithms to address both the features of the different entities and the structural or topological relationships between them. Moreover, turning a computer vision problem into a graph problem allows one to access the full arsenal of graph algorithms developed in computer science.

The Technical Committee (TC15, http://www.iapr.org/tcs.html) of the IAPR (International Association for Pattern Recognition) has been funded in order to federate and to encourage research work in these fields. Among its activities, TC15 encourages the organization of special graph sessions at many computer vision conferences and organizes the biennial workshop GbR. While being designed within a specific framework, the graph algorithms developed for computer vision and pattern recognition tasks often share constraints and goals with those developed in other research fields such as data mining, robotics and discrete geometry. The TC15 community is thus not closed in its research fields but on the contrary is open to interchanges with other groups/communities. Within this framework, the TC15 community decided to organize the fifth edition of its workshop jointly with the international conference Discrete Geometry for Computer Imagery (DGCI) organized by TC18 of the IAPR. Indeed, within the pattern recognition field, many graph-based algorithms are used to analyze the structures of the underlying objects. On the other hand, many algorithms of discrete geometry aim at finding the structures of unstructured sets of pixels or voxels. From this point of view, both communities aim at studying the structures of discrete objects. Both conferences were held in Poitiers, during the same week, with a common session on Wednesday 13th of April.

This volume contains the papers presented at the 5th Workshop on Graph-Based Representations in Pattern Recognition (GbR) organized by the IAPR TC15. The workshop was held at the University of Poitiers, France during April 11–13, 2005. The previous workshops in the series were held in Lyon, France (1997), Haindorf, Austria (1999) [3], Ischia, Italy (2001) [2], and York, UK (2003) [1].

The papers presented during this workshop, while all based on graphs, cover a wide range of research fields related to image processing and understanding. Indeed, one paper presented by Alain Bretto and Luc Gillibert uses graphs for low image processing such as noise attenuation and edge detection. Then several papers present several segmentation methods based on graphs together with improved graph data structures to encode fine properties of the partitions. Graphs or hierarchical graph data structures may thus be used to encode fine properties of the image's content. However, graphs may also be used to encode shape information. Many papers presented during this workshop encode a shape using either its skeleton or a set of points characterizing it. Given a graph describing an object (a shape, an image, a graphic, etc.) the next step consists of determining a measure of similarity between these graphs in order to derive a similarity measure between the underlying objects. Several papers devoted to graph matching attack this difficult problem using either exact or inexact algorithms. Algorithms based on graph kernels and the heat kernel equations provide alternative and interesting approaches to graph matching. Graph-matching algorithms may be pushed one step further by studying not only the matching between two graphs but also the classification of a set of graphs or the analysis of a sequence of graphs. Several papers presented during the workshop present novel and interesting ideas on these topics.

The papers presented here have all been reviewed by two reviewers and revised by their authors. The 50 papers submitted to the GbR were written by authors coming from 20 different countries located on five different continents. Based on these 50 submitted papers the Program Committee selected 18 of them as full papers and 17 of them as posters. We would therefore like to thank the members of the Program Committee and the additional reviewers for their help in ensuring that the papers were given a thorough and critical evaluation. We would also like to thank our sponsors who provided the material and financial help for the organization of this workshop.

April 2005

Luc Brun
Mario Vento

References

1. Edwin Hancock and Mario Vento (Eds.) *Graph-Based Representations in Pattern Recognition, GbRPR 2003*, volume 2726 of *LNCS*, York, UK, June/July 2003. IAPR-TC15, Springer. ISBN 3-540-40452-X
2. Jean-Michel Jolion, Walter G. Kropatsch, and Mario Vento, editors. *Graph-Based Representations in Pattern Recognition, GbR 2001*, Ischia, Italy, May 2001. IAPR-TC15, CUEN. ISBN 88-7146-579-2.
3. Walter G. Kropatsch and Jean-Michel Jolion, editors. *Graph-Based Representations in Pattern Recognition, GbR 1999*, Haindorf, Austria, May 1999. IAPR-TC15, OCG. ISBN 3-85403-126-2.

Organization

Program Co-chairs

Luc Brun
GREYC
ENSICAEN
6 Boulevard du Maréchal Juin
14050 Caen
France

Mario Vento
MIVIA Lab - Università di Salerno
Dipart. di Ing. Dell'Informazione ed Ingegneria Elettrica
Università di Salerno
Via Ponte Don Melillo, 1 Fisciano (SA) Italy

Program Committee

Horst Bunke	University of Bern, Switzerland
Terry Caelli	University of Alberta, Edmonton, Canada
Guillaume Damiand	Université de Poitiers, France
Luigi Pietro Cordella	Università di Napoli, Naples, Italy
Jean Michel Jolion	INSA Lyon, Lyon, France
Walter Kropatsch	Vienna Univ. of Technology, Vienna, Austria
Marcello Pellilo	Università Ca' Foscari di Venezia, Venice, Italy

Referees

I. Bloch (France)
A. Braquelaire (France)
L. Brun (France)
H. Bunke (Switzerland)
T. Caelli (USA)
L.P. Cordella (Italy)
G. Damiand (France)
P. Foggia (Italy)
F. Fontanella (Italy)
E.R. Hancock (UK)
C. Irniger (Switzerland)
J.-M. Jolion (France)
W.G. Kropatsch (Austria)
O. Lezoray (France)
M. Neuhaus (Switzerland)
M. Pellilo (Italy)
M. Revenu (France)
C. Sansone (Italy)
C. De Stefano (Italy)
M. Vento (Italy)

Sponsoring Institutions

ENSICAEN - 6 Bd. Maréchal Juin, 14050 Caen, France
GREYC - 6 Bd. Maréchal Juin, 14050 Caen, France
Université de Poitiers, France

Table of Contents

Graph Representations

Graphs and Linear Representations

Combinatorial Maps

Matching

Hierarchical Graph Abstraction and Matching

Inexact Graph Matching

Learning

Graph Sequences

Graph Kernels

Graphs and Heat Kernels

Hypergraph-Based Image Representation

Alain Bretto and Luc Gillibert

Université de Caen, GREYC CNRS UMR-6072, Campus II, Bd Marechal Juin BP, 5186, 14032 Caen cedex, France
{alain.bretto, lgillibe}@info.unicaen.fr

Abstract. An appropriate image representation induces some good image treatment algorithms. Hypergraph theory is a theory of finite combinatorial sets, modeling a lot of problems of operational research and combinatorial optimization. Hypergraphs are now used in many domains such as chemistry, engineering and image processing. We present an overview of a hypergraph-based picture representation giving much application in picture manipulation, analysis and restoration: the Image Adaptive Neighborhood Hypergraph (IANH). With the IANH it is possible to build powerful noise detection an elimination algorithm, but also to make some edges detection or some image segmentation. IANH has various applications and this paper presents a survey of them.

Keywords: Image Processing, Image Model, Segmentation, Edge Detection, Noise Cancellation, Hypergraph, Graph, Neighborhood Hypergraph.

1 Introduction

Graphs are very powerful tools for describing many problems and structures in computer sciences but also in physic and mathematics. But graphs only describe some binary relations and are not always sufficient for modeling some complex problems or data. Hypergraph theory, originally developed by C. Berge [8] in 1960, is a generalization of graph theory. The idea consists in considering sets as generalized edges and then calling a hypergraph the family of these edges. This concept models more general types of relations than graph theory do. In the last decades, the theory of hypergraphs has proved to be of a major interest in applications to real-world problems. These mathematical frameworks can be used to model networks, data structures, process scheduling, computations, and a variety of other systems where complex relations between the objects in the system play a dominant role.

To any digital image, a hypergraph, the Image Adaptive Neighborhood Hypergraph (IANH), can be associated and used for image processing. Many publications were written about this hypergraph model and many applications were found [1, 2, 4, 6]. This paper is a survey of different methods about image analysis and treatment based on this adaptive neighborhood hypergraph model.

L. Brun and M. Vento (Eds.): GbRPR 2005, LNCS 3434, pp. 1–11, 2005.

First, we give basic definitions about hypergraphs and the definition of the IANH. Then we present an algorithm building the IANH and we study its properties and its complexity. Finally we illustrate some applications of the IANH to the image segmentation, the edge detection and the noise cancellation, three of the most important low level image processings. We give some powerful algorithms always based on the adaptive neighborhood hypergraph associated to an image. A set of examples is shown to illustrate the effectiveness of the algorithms.

2 Definitions

The general terminology concerning graphs and hypergraphs is similar to [8, 7]. All graphs in this paper are, finite, undirected, connected with no isolated vertex and simple, *i. e.* graphs with no loops or multiple edges. We denote a graph $G = (V; E)$. Given a graph G, we denote by $\Gamma(x)$ the *neighborhood* of a vertex x, *i. e.* the set consisting of all vertices adjacent to x which is defined by $\Gamma(x) = \{y \in V, \{x, y\} \in E\}$.

A *hypergraph* H on a set $\mathbf{X}$ is a family $(E_i)_{i \in I}$ of non-empty subsets of $\mathbf{X}$ called *hyperedges* with;

$$\bigcup_{i \in I} E_i = \mathbf{X}, \quad I = \{1, 2, \dots, n\}, \quad n \in \mathbf{N}$$

Let us note $H = (\mathbf{S}; (E_i)_{i \in I})$. For $x \in \mathbf{S}$, a *star* of H (with center x) is the set of hyperedges which contains x, and is called $H(x)$. The *degree of* x is the cardinality of the star $H(x)$ denoted by $dx = Card(H(x))$.

Let $H = (\mathbf{S}; E = (E_i)_{i \in I})$ be a hypergraph, the *dual hypergraph* H^* is the hypergraph such that the set of vertices is the set of hyperedges, and the set of hyperedges is the set of stars of H. We can represent a hypergraph as in figure 1-(a).

A hyperedge E_i is *isolated* if and only if:

$$\forall j \in I,\ j \neq i,\ if\ E_i \cap E_j \neq \emptyset\ then\ E_j \subseteq E_i$$

An important structure from a hypergraph is the notion of *intersecting family.* A family of hyperedges is an intersecting family if the hyperedges from this family intersect two by two. We can distinguish two types of intersecting families:

- Intersecting families with an empty intersection.
- Intersecting families with an non empty intersection.

A hypergraph has the HELLY *property* if each family of hyperedges intersecting two by two (intersecting family) has a non empty intersection (belongs to a star). As example in figure 1-(a) the hypergraph has the HELLY property. Figure 1-(b) shows these two types of intersecting hyperedges. To each graph one can associate a hypergraph. Indeed, let $G = (X, E)$ be a graph, the hypergraph having the vertices of G as vertices and the neighborhood of these vertices as hyperedges (including these vertices) is called the *neighborhood hypergraph* of G and is denoted by:

$$H_G = (X, (E_x = \{x\} \cup \Gamma(x)))$$

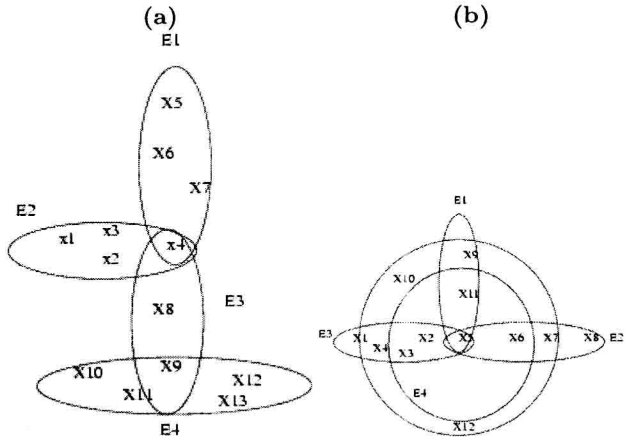

Fig. 1. (a) Example of hypergraph, the set of vertices is $\{x_1, x_2, \ldots, x_13\}$ and the set of hyperedges is $\{E_1, E_2, E_3, E_4\}$. (b) We have two types of intersecting families the first is the star the second has an empty intersection

3 Image Adaptive Hypergraph Model

First we recall some definitions about digital images. A distance d' on X defines a *grid* (a graph connected, regular, without both loop and multi-edge). A *digital image* (on a grid) is a two-dimensional discrete function that has been digitized both in spatial coordinates and in magnitude feature value. Throughout this paper a digital image will be represented by the application $I : X \subseteq Z^2 \rightarrow \mathcal{C} \subseteq Z^n$ with $n \geq 1$, where $\mathcal{C}$ identifies the *feature intensity level* and X identifies a set of points called *image points*. The couple $(x, I(x))$ is called a *pixel*. Let d be a distance on $\mathcal{C}$, we have a neighborhood relation on an image defined by:

$$\forall x \in X,\ \Gamma_{\alpha,\beta}(x) = \\ \{x' \in X, x' \neq x \mid d(\mathcal{I}(x), \mathcal{I}(x')) < \alpha \text{ and } d'(x, x') \leq \beta\} \qquad (1)$$

The neighborhood of x on the grid will be denoted by $\Gamma_\beta(x)$. So to each image we can associate a hypergraph called *Image Adaptive Neighborhood Hypergraph* (IANH): $H_{\alpha,\beta} = \big(X, (\{x\} \cup \Gamma_{\alpha,\beta}(x))_{x \in X}\big)$. The attribute α can be computed in an adaptive way depending on local properties of the image. If α is constant the hypergraph is called the *Image Neighborhood Hypergraph* (INH). Throughout this paper α will be estimated by the standard deviation of the pixels $\{x\} \cup \Gamma_\beta(x)$.

Algorithm: Image Adaptive Neighborhood Hypergraph
Construction of the hypergraph $H_{\alpha,\beta}$.
 Data: Image I of size $m_x \times m_y$, and neighborhood order β
 $X = \emptyset$;
 For each pixel x **of** I, **do** ;
  $\alpha =$ the standard deviation of the pixels $\{x\} \cup \Gamma_\beta(x)$;

$\Gamma_{\alpha,\beta}(x) = \emptyset$;
For each pixel y **of** $\Gamma_\beta(x)$, **do**
if $d(I(x), I(y)) \leq \alpha$ **then**
$\Gamma_{\alpha,\beta}(x) = \Gamma_{\alpha,\beta}(x) \cup \{y\}$;
end if
end for
$X = X \cup \{x\}$; $E_{\alpha,\beta}(x) = \{\Gamma_{\alpha,\beta}(x) \cup \{x\}\}$;
end for
$H_{\alpha,\beta} = (X, (E_{\alpha,\beta}(x))_{x \in X})$;
End

Data Structures Used: For each x, $\Gamma_{\alpha,\beta}(x)$ is a table of booleans, so $E_{\alpha,\beta}$ is a $m_x \times m_y$ table of tables. The set X is a $m_x \times m_y$ table of booleans.

Proposition 1. *Given β, the algorithm converges to a unique solution. Its complexity is in $O(n)$ (n standing for the pixel number of the image). (For the proof report to [2]).*

4 Detection of Impulsive Noise

A common type of corruption that occurs in image data is corruption by an impulsive noise process. Attenuation of noise and preservation of details are usually two contradictory aspects of image. Various noise reduction algorithms make various assumptions, depending on the type of imagery and the goals of the restoration [9],[10].

In this section, we present a noise cancellation algorithm that exploits a lack of homogeneity criterion. We consider that the global homogeneity characterizes regions, local homogeneity characterizes edges, no homogeneity characterizes a noise. A noise reduction algorithm is based on the following criterion: binary classification of hyperedge of image (H_0 noisy hyperedge and H_1 no noisy hyperedge) and filtering the noisy parts.

Noise Definition - We will call disjoined chain a ordered succession of hyperedges disconnected two by two and build on some adjacent pixels. A disjoined chain is *thin* if the cardinality of each hyperedge is equal to 1. To model a noise we propose the following definition:

We say that $E_{\alpha,\beta}(x)$ is a *noise hyperedge* if it verifies one of the two conditions:

- The cardinality of $E_{\alpha,\beta}(x)$ is equal to 1 and $E_{\alpha,\beta}(x)$ is not contained in disjoined thin chain having five elements at least.
- $E_{\alpha,\beta}(x)$ is an isolated hyperedge and there exists an element y belonging to the open neighborhood of $E_{\alpha,\beta}(x)$ on the grid, such that $E_{\alpha,\beta}(y)$ is isolated. (i.e. $E_{\alpha,\beta}$ is isolated and it has an isolated hyperedge in its neighborhood on the grid).

This definition allows a good discrimination between edge pixels and noisy pixels. The lemma below shows that a noisy hyperedge must be isolated.

Lemma 1. *If the cardinality of a hyperedge is equal to one, then this hyperedge is isolated. (For the proof report to [6]).*

With the noise definition above, a noise detection algorithm is simple. The IANH is built and all the hyperedges satisfying the conditions of the noise definition are selected. This selection is separated in two step, first the detection of the isolated hyperedges and then, in the set of the isolated hyperedges, the detection of the noisy hyperedges.

Algorithm: Noise Detection
Data: Image I of size $m_x \times m_y$, IANH $H_{\alpha,\beta}$.
Determination of isolated hyperedges of $H_{\alpha,\beta}$

```
For each vertex x of H_{α,β}, do ;
    E'_x = ⋃_{y∈E_{α,β}(x)} E_y;
        If E'_x == E_x, (E_x is an isolated hyperedge) then
            If Cardinality of E_x is equal to one then
                ISO[x] = E_x;
            Else
                IS[x] = E_x;
            end if
        end if
end for
```

Detection of noise hyperedges of $H_{\alpha,\beta}$

```
For each E_x of ISO[], do
    For each E_y of ISO[], and x ≠ y do
        For each E_z of ISO[], and x ≠ z and y ≠ z do
            If y, z ∉ Γ_β(x) then
                NH[x] = E_x;
            end if
        end for
    end for
end for
For each E_x of IS[], do
    If there is y ∈ Γ°(E_x) such as E_{α,β}(y) ∈ ISO[] ∪ IS[], then
        NH[x] = E_x;
    end if
end for
End
```

Data Structures Used: The data structures used for the IANH and its hyperedges are the same that in the IANH construction algorithm. The sets ISO, ES and NH are some $m_x \times m_y$ tables of hyperedges.

The complexity of this algorithm is in $O(n^3)$. This algorithm has been tested on several images in order to show how effective our method is. This method has a great advantage over the class of linear filters; it preserves the edges, so the additional complexity provides some additional results. Some experimental results can be found in [6]. Some visual examples are shown in figure 2.

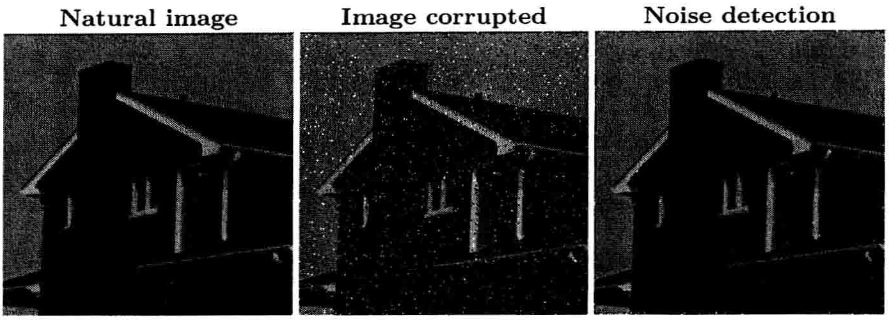

Fig. 2. Example of IANH-based noise detection and cancellation

5 Segmentation

One of the first major step of low level vision is segmentation. Segmentation is the process which consists in partitioning an image into some non-intersecting regions such that each region is homogeneous and that the union of two adjacent regions is never homogeneous. The algorithm below will give us the segmentation of an image. This algorithm is based on the detection of stars in the hypergraph model.

The algorithm process can be divided into in two main parts. In the first part, a covering of the image by a minimal set of stars is computed. In the second part, selected stars are aggregated to obtain the regions. This regions are the segments of the image.

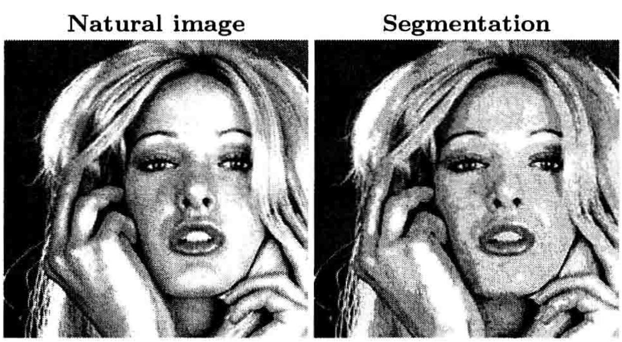

Fig. 3. Example of IANH-based image segmentation

Algorithm: Covering and Selection for Segmentation

Data: Image I of size $m_x \times m_y$, IANH $H_{\alpha,\beta}$.

Choosing a cover of the image by a minimal set of stars.

Chose a minimal cover of the image, $E = \{H(x_1), H(x_2), \ldots H(x_n)\}$, such that any pixel of the image belongs to at most one hyperedge of at least one star of the set E.

Building aggregate areas.
For each $H(x_i)$ in E, **do**
$I(x_i) =$ grey level of the center of the star $H(x_i)$;
$Agg_i = \emptyset$ (initialization on a new aggregation area)
For each $H(x_j)$ intersecting with the star $H(x_i)$, **do**
If $I(x_j)$ in $[I(x_i) - \alpha, \quad I(x_i) + \alpha]$, **then**
$Agg_i = Agg_i \cup$ (vertices of $H(x_j) \cup H(x_i)$);
end if
end for
end for
Reducing the number of areas.
For each aggregate area Agg_i, **do**
$g_i =$ center of gravity of Agg_i;
$min_i =$ minimum grey level of the centers of the stars of Agg_i;
$max_i =$ maximum grey level of the centers of the stars of Agg_i;
$med_i =$ medium grey level of the centers of the stars of Agg_i;
end for
For each aggregate area Agg_i, **do**
For each area Agg_j intersecting with Agg_i, **do**
If g_i **or** g_j is in $Agg_i \cup Agg_j$,
and min_i in $[min_j - \alpha,\ min_j + \alpha]$,
and max_i in $[max_j - \alpha,\ max_j + \alpha]$,
and med_i in $[med_j - \alpha,\ med_j + \alpha]$, **then**
Aggregate Agg_i and Agg_j;
end if
end for
end for
Assigning each star to an aggregation area to obtain a partition.
Chose the area Agg_i containing the greatest number of stars
For each star H in Agg_i, **do**
Assign the star H to Agg_i;
For each aggregate area $Agg_j \neq Agg_i$, **do**
Remove H from Agg_j;
end for
end for
Repeat chose until all the stars have been assigned.
Assigning the pixels generating edges
For each pixel x in I, **do**
If x in several stars, **then**
assign x to the area of the star center whose grey level is the closest
end if
end for

Finally, the pixels lefts are assigned to the area of the neighboring pixel which has already been assigned, and whose grey level is the closest.