CPRW'99, Peršlak , CZ, February 2-4, 2000

Hierarchies of Combinatorial Maps

Walter G. Kropatsch*and Luc Brun
Vienna Univ. of Technology, Inst. of Computer Aided Automation, 183/2
Pattern Recognition + Image Processing Group
Favoritenstr.9 / A-1040 WIEN / Austria
Fax.+43(1)58801x18392
krw@prip.tuwien.ac.at

Abstract *Hierarchies of graphs can be generated by dual graph contraction. The goal is to reduce the data structure by a constant reduction factor while preserving certain image properties like connectivity. Since these graphs typically samplings of the plane they are by definition plane. The particular embedding can be represented in different ways, e.g. a pair of dual graphs relating points and faces through boundary segments. Combinatorial maps determine the embedding by explicitely recording the orientation of edges around vertices. We summarize the formal framework which has been set up to perform dual graph contraction with combinatorial maps. Contraction is controlled by kernels that can be combined in many ways. We have shown that kernels producing a slow reduction rate can be combined to speed up reduction. Or, conversely, kernels decompose into smaller kernels that generate a more gradual reduction.*

1. Introduction

This paper surveys a framework for building irregular pyramids with combinatorial maps. The formal definitions and theorems are given in the two technical reports [1, 2]. Here we summarize the motivation, the major concepts and the planned applications.

We start with a comparison of different ways to embed structural descriptions of images in the plane (section 2.). We then outline the basic terms from combinatorial maps in section 3.. The basic operations corresponding to dual graph contraction [6] are formulated in terms of combinatorial maps in section 4.. The further concept of equivalent contraction kernel as presented in [6] can be expressed in terms of successor and inclusion kernels (section 5.). We conclude by comparing the benefits of combinatorial maps with respect to dual irregular pyramids.

*This work was supported by the Austrian Science Foundation under grant S 7002-MAT

2. Embedding in the image plane

Objects that are mapped into the image plane induce spatial relations among each other and between their parts. Geometrical measurements derived from a digital image are very sensitive to errors due to noise, discrete sampling and motion inaccuracies. However these structural and topological relations are inherent to the objects and their arrangement in the image and mostly do not depend on the particular imaging situation. This is the background of several recent contributions describing spatial/structural representations and transformations preserving existing topological relations in the image plane. Following list enumerates a few possibilities to preserve structural relations into a more abstract representation:

1. The simplest uses coordinates as vertex attributes of an attributed relational graph. This immediate representation depends on the particular mapping geometry. For well controlled environments (e.g. geographic information systems) it is widely used due to its simplicity.
2. Another approach [8] considers local deformations of digital curves that preserve an implictely given topology. The idea is that images showing the same topological arrangement of regions and curves can be transformed into each other. An interesting extension to higher dimension is presented by Fourey and Malgouyres [3].
3. A pair of plane dual graphs is the base of an irregular graph pyramid built by repeated dual graph contractions [5]. It differs from the previous approach that the transformed data are reduced at each step by a factor which is the origin of its computational efficiency.
4. Topological and combinatorial maps have been investigated in [4] and [7]. There the embeddeing is determined by the local orientation of the structural elements. These works are the basis of this

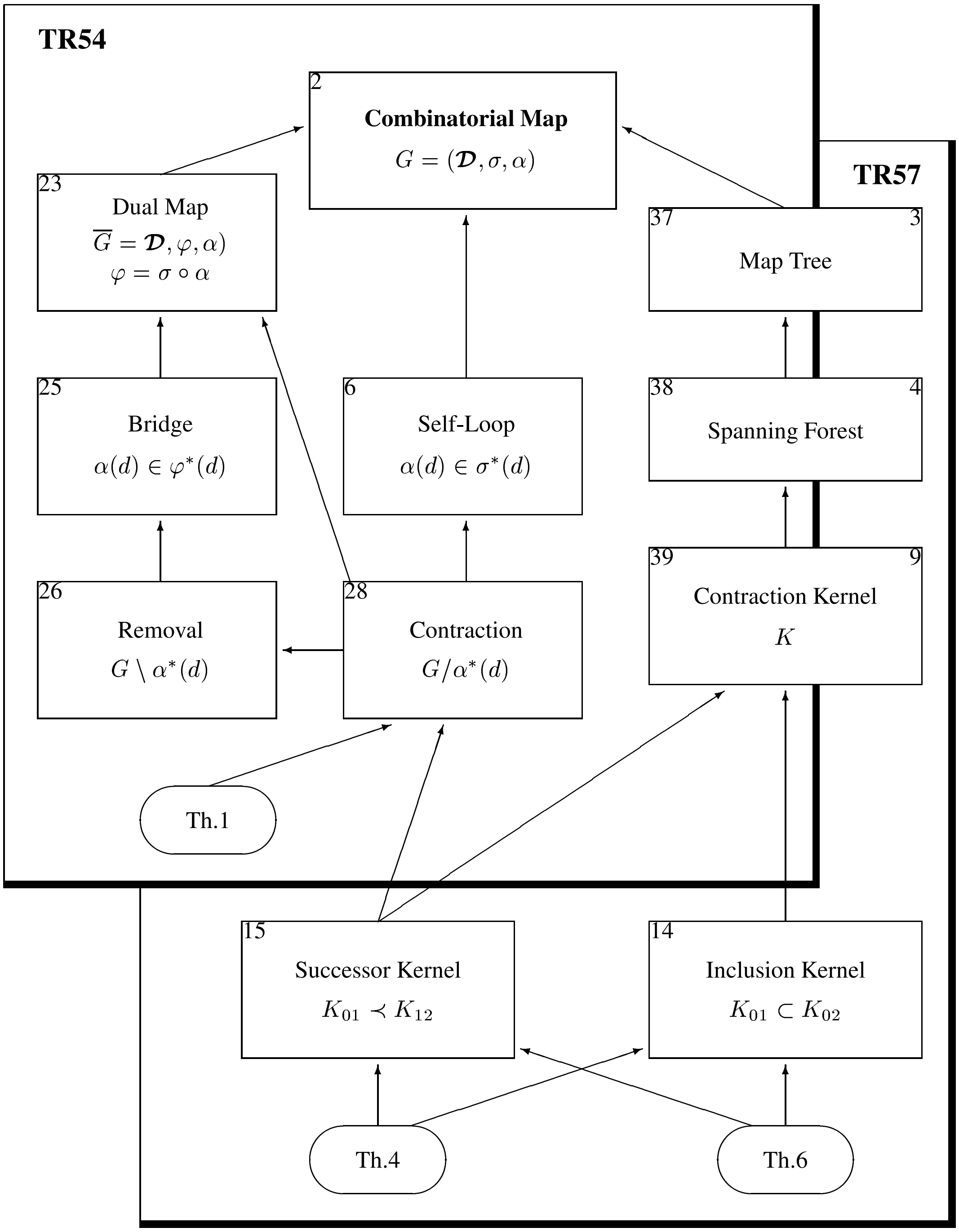
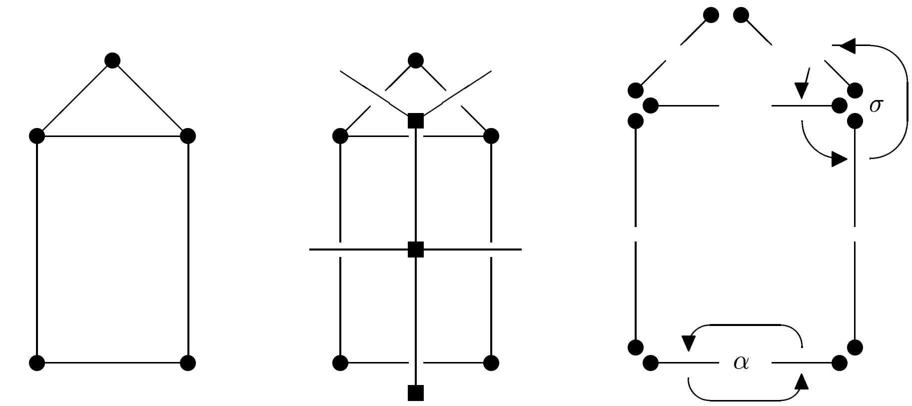


Figure 1: Main definitions and theorems of TR-54 and TR-57



(a) A plane graph (b) decomposed along dual edges (c) combinatorial map

Figure 2: From a plane graph to a combinatorial map

papers where we combine dual graph contraction with the representation of combinatorial maps.

3. Combinatorial Maps

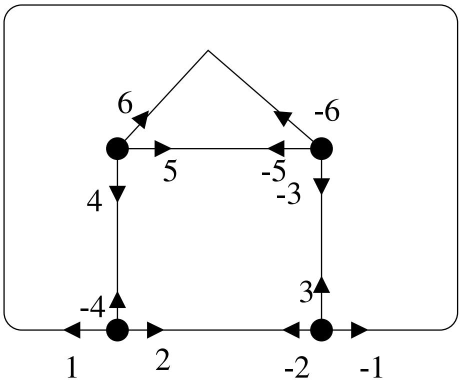
A combinatorial map may be seen as a planar graph encoding explicitly the orientation of edges around a given vertex. Thus all graph definitions used in irregular pyramids [6] such as end vertices, self loops, or degrees may be retrieved easily.

Figures 2 and 3 demonstrate the derivation of a combinatorial map from a plane graph. First edges are split where their dual edges cross (see Figure 2-b). That decomposes the graph into connected parts of half-edges that surround each vertex. These half edges are called *darts* and have their origin at the vertex they are attached to. The fact that two half-edges (darts) stem from the same edge is recorded in the **reverse permutation** α. A second permutation σ, called the **successor permutation**, defines the (local) arrangement of darts around a vertex. Counterclockwise ordering is assumed here (see Figure 3). The symbols $\alpha^*(d)$ and $\sigma^*(d)$ stand, respectively, for the α and σ orbits of the dart d. More generally, if d is a dart and π a permutation we will denote the π-orbit of d by $\pi^*(d)$. The cardinal of this orbit will be denoted $|\pi^*(d)|$.

A **combinatorial map** G is the triplet $G = (\mathcal{D}, \sigma, \alpha)$, where $\mathcal{D}$ is the set of darts and σ, α are two permutations defined on $\mathcal{D}$ such that α is an involution:

$$\forall d \in \mathcal{D} \quad \alpha^2(d) = d$$

If the darts are encoded by positive and negative integers, the permutation α can be implicitly encoded by $\alpha(d) = -d$ (see Figure 3). In the following, we will use alternatively both notations, the notation $\alpha(d) = -d$ will be often use for practical results linked to the implementation of our model. Indeed, if the permutation α is implicitly encoded, the combinatorial map may be implemented by a basic array of integers encoding the permutation σ, which looks as follows for Fig. 3:



$$\sigma = (1, 2, -4)(-2, -1, 3)(-3, -6, -5)(4, 5, 6)$$

Figure 3: *The permutation σ*

d	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
$\sigma(d)$	-5	-3	1	-6	-1	3		2	-4	-2	5	6	4

Following concepts from graph theory that are needed later for structure preserving operations can be expressed in terms of combinatorial maps: self-loop, duality, and bridge. An edge $\alpha^*(d)$ is called a self loop, iff: $-d \in \sigma^*(d)$. Or, if the two endpoints of an edge are the same vertex.

A face of a planar graph is defined by the set of edges which surround it. Using a combinatorial map, one dart per edge is sufficient to encode a face, since for each dart the involution α allows us to retrieve the other dart defining the edge. Moreover, the ordered sequence of darts around a vertex encoded by permutation σ induce an order in the sequence of faces encountered when turning around a face. This order is encoded thanks to the permutation $\varphi = \sigma \circ \alpha$: Given a combinatorial map $G = (\mathcal{D}, \sigma, \alpha)$, the combinatorial map $\overline{G} = (\mathcal{D}, \varphi, \alpha)$ is called **dual combinatorial map** of G. The orbits of

φ encode the faces of G. Note that the function φ is a permutation, since it is the composition of two permutations on the same set. Using a clockwise orientation for permutation σ all the faces of the combinatorial map except one are counter-clockwise oriented. The clockwise oriented face is called the infinite face. The dual map of Fig. 3 is given as follows:

d	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
$\varphi(d)$	4	6	5	-2	-4	2		3	-1	-6	1	-3	-5

The connectivity of a graph (or a subgraph representing an object) is an essential structural property. Since our goal is to successively remove unnecessary parts the connectivity can be lost by these operations. Before disconnecting a graph into two components these two components will be connected by a single edge which is called a **bridge** which can be characterized by

$$\alpha(d) \in \varphi^*(d)$$

4. Contraction and Removal

This section is devoted to the definition and the properties of the operations that will be used in irregular pyramids. Given a combinatorial map a first useful operation is the removal of an edge $\alpha^*(d)$. The resulting combinatorial map may be defined as a sub-combinatorial map deduced from the original one by simply removing the darts d and $\alpha(d)$ from its set of darts . In order to preserve the number of connected components of the original combinatorial map bridges must be excluded from removal operations. Furthermore self-direct loops, which are a special case of a self-loop with $\sigma(d) = \alpha(d)$, are excluded from this general operation, and are treated as special cases. With these restrictions the formal definition of the removal operation may be written in terms of modifications of permutation σ

Given a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ and a dart $d \in \boldsymbol{\mathcal{D}}$ which is neither a bridge nor a self-direct-loop, the **contracted submap** $G \setminus \alpha^*(d) = (\boldsymbol{\mathcal{D}} - \alpha^*(d), \sigma', \alpha)$ is defined by:

$$\left\{ \begin{array}{lll} \forall d' \in \boldsymbol{\mathcal{D}} - \{\sigma^{-1}(d), \sigma^{-1}(-d)\} & \sigma'(d') = \sigma(d') & \\ \sigma'(\sigma^{-1}(d)) & = & \sigma(d) \\ \sigma'(\sigma^{-1}(-d)) & = & \sigma(-d) \end{array} \right.$$

Given a partition of an image, merging two regions may be considered in two different ways: First we can consider that the two regions are merged by removing one of their common boundaries. This operation is encoded in our combinatorial map formalism by the edge removal. Secondly, we can also consider that the two regions are merged by identifying the two regions and removing one of their common boundaries. This dual point of view is encoded in our formalism by the contraction operation.

Using the duality we define the **contraction** of dart d of a given combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ which is not a self loop. The result is the following graph

$$G' = G/\alpha^*(d) = \overline{\overline{G} \setminus \alpha^*(d)}$$

Note that this operation is well defined since d is a self-loop in G iff it is a bridge in $\overline{G}$.

Note that, under the same hypothesis, we have:

$$\overline{\overline{G}/\alpha^*(d)} = G \setminus \alpha^*(d)$$

Thus the two dual points of view on merging regions are performed by two dual operations on the combinatorial map and its dual. Thus many particular cases of one operation may be retrieved thanks to the particular cases of the other. For example, since bridges are forbidden for removal operation the dual of a bride, i.e. a self-loop, is forbidden for contraction.

5. Equivalent Contraction Kernels

The concept of a tree and of a forest are used to define a contraction kernel that collects a set of darts that can be contracted independently of each other without destroying the connectivity structure of the graph. A sequence of merging segments of a partition may be encoded by a sequence of contractions of the combinatorial map encoding the partition. Since the contraction operation is forbidden for self-loops the set of darts involved in such a sequence of contractions must not contain a circuit. Thus the set of edges involved in such a contraction may be encoded by a tree which is a submap of the combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ with only one φ'-orbit. The only dual face of a tree is the background face.

More generally, if we contract a set of vertices into a given set of surviving vertices, the set of darts involved in such contractions may be encoded by a forest $F = (\boldsymbol{\mathcal{D}}_1, \ldots, \boldsymbol{\mathcal{D}}_n)$ which is a collection of non-overlapping trees spanning the given combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$.

The forest $K = (\boldsymbol{\mathcal{D}}_1, \ldots, \boldsymbol{\mathcal{D}}_n)$ of G will be called a **contraction kernel** iff:

$$\mathcal{SD} = \boldsymbol{\mathcal{D}} - \bigcup_{i=1}^{n} \boldsymbol{\mathcal{D}}_i \neq \emptyset$$

The set $\mathcal{SD}$ is called the set of surviving darts.

We can apply successively two (and more) contraction kernels K_{01} and K_{12} to a given combinatorial map G_0: $G_1 = G_0/K_{01}$ and $G_2 = G_1/K_{12}$. The same result can be achieved by applying a bigger kernel only once: $G_2 = G_0/K_{02}$. Conversely, a contraction kernel may be decomposed into two smaller ones. The successive application of the resulting contraction kernels is equivalent to the application of the initial one. Different contraction kernels on the same combinatorial map G_0 may be related by **inclusion**, successive kernels give rise to **predecessor** and **successor** relations which allow us to formulate the above mentionned equivalences:

Inclusion of Contraction Kernels: Let us consider two different contraction kernels K_{01} and K_{02} defined on a combinatorial map G_0. We will say that the contraction kernel K_{02} includes K_{01} iff $K_{01} \subset K_{02}$. In this case each connected component, a tree $\mathcal{T}_1$ of K_{01} is included in exactly one connected component, a tree $\mathcal{T}_2$ of K_{02}:

$$\forall \mathcal{T}_1 \in \mathcal{CC}(K_{01}) \exists!\, \mathcal{T}_2 \in \mathcal{CC}(K_{02}) \text{ s.t. } \mathcal{T}_1 \subset \mathcal{T}_2.$$

Predecessor and Successor Kernels Given a combinatorial map $G_0 = (\mathcal{D}, \sigma, \alpha)$, a contraction kernel K_{01} of G_0 and the contracted combinatorial map $G_1 = G_0/K_{01}$. If K_{12} is a contraction kernel of G_1 then we say that K_{01} is the predecessor of K_{12}, or that K_{12} is the successor of K_{01}. This relation will be denoted $K_{01} \prec K_{12}$.

The successive application of K_{01} and K_{12} forms a new operator on G_0 denoted by $K_{12} \circ K_{01}$.

Based on these two definitions two theorems could be formulated in TR-57 [2] that relate composition and decomposition of contraction kernels:

Theorem 4 derives inclusion kernels from successor kernels:

$$K_{01} \prec K_{12} \Longrightarrow K_{01} \subset K_{02} = K_{01} \cup K_{12}$$
$$\text{with } (G_0/K_{01})/K_{12} = G_0/K_{02}.$$

The kernel K_{02} combines kernel K_{01} with the subtrees of K_{12} such that that the result of contracting G_0 with K_{02} is the same as if G_0 is contracted with K_{01} and with K_{02} in succession.

Theorem 6 derives successor kernels from inclusion kernels:

$$K_{01} \subset K_{02} \Longrightarrow K_{01} \prec K_{12} = K_{02} - K_{01}$$
$$\text{with } G_0/K_{02} = (G_0/K_{01})/K_{12}.$$

Given two contraction kernels K_{01}, K_{02} for G_0, K_{01} being included in K_{02}, the larger kernel K_{02} can be decomposed into K_{01} and the successor kernel K_{12} which can be used after contracting G_0 with K_{01} to yield the same result.

6. Conclusion

Combinatorial maps and a pair of plane dual graphs are equivalent representations. One can be transformed into the other without loss of information. They differ in what they represent explicitely and that implicitely defined information needs additional retrieval processes to be accessible.

Combinatorial maps code the embedding of a planar graph using explicitely the orientation of the edges around a vertex. This coding allows the construction of the dual graph without additional information. Hence it does not necessitate the storage of the dual graph as in dual irregular pyramids. However certain structural entities are not represented explicitely: e.g. vertices and faces are implicitely defined and need extra processes to be identified, or to receive further attributes like in attributed relational graphs.

Another advantage of combinatorial maps for building hierarchies of graphs is related to the definition of darts: since a surviving dart is always linked to its vertex which must survive by definition, redefinition and renaming of the surviving darts is not needed. Hence the equivalent contraction kernels can be simply expressed by simple subset relations. This fact will be exploited in the future to efficiently represent complete irregular pyramids by labels attached to the darts of the base graph.

References

[1] Luc Brun and Walter G. Kropatsch. Dual Contraction of Combinatorial Maps. Technical Report PRIP-TR-54, Institute f. Computer Aided Automation 183/2, Pattern Recognition and Image Processing Group, TU Wien, Austria, 1999. Also available through http://www.prip.tuwien.ac.at/ftp/pub/publications/trs/tr54.ps.gz.

[2] Luc Brun and Walter G. Kropatsch. Pyramids with Combinatorial Maps. Technical Report PRIP-TR-57, Institute f. Computer Aided Automation 183/2, Pattern Recognition and Image Processing Group, TU Wien, Austria, 1999. Also available through http://www.prip.tuwien.ac.at/ftp/pub/publications/trs/tr57.ps.gz.

[3] Sébastien Fourey and Rémy Malgouyres. Intersection Number of Paths Lying on a Digital Surface and a New Jordan Theorem. In Gilles Bertrand, Michel Couprie, and Laurent Perroton, editors, *Discrete Geometry for Computer Imagery, DGCI'99*, volume Vol. 1568 of *Lecture Notes in Computer Science*, pages 104–117, Marne-la-Vallée, France, 1999. Springer, Berlin Heidelberg, New York.

[4] A. Jones Gareth and David Singerman. Theory of maps on orientable surfaces. volume 3, pages 273–307. London Mathematical Society, 1978.

[5] Walter G. Kropatsch. Equivalent contraction kernels to build dual irregular pyramids. *Advances in Computer Science*, Advances in Computer Vision:pp. 99–107, 1997.

[6] Walter G. Kropatsch. From equivalent weighting functions to equivalent contraction kernels.

In *CZECH PATTERN RECOGNITION WORKSHOP'97*, pages 1–13. Czech Pattern Recognition Society, February 1997.

[7] P. Lienhardt. Subdivisions of n-dimensional spaces and n-dimensional generalized maps. In Kurt Mehlhorn, editor, *Proceedings of the 5th Annual Symposium on Computational Geometry (SCG '89)*, pages 228–236, Saarbrücken, FRG, June 1989. ACM Press.

[8] Azriel Rosenfeld and Akira Nakamura. Local deformations of digital curves. *Pattern Recognition Letters*, 18:613–620, 1997.