Discrete Maps: a framework for region segmentation algorithms

Luc Brun, Jean-Philippe Domenger and Jean-Pierre Braquelaire,

Laboratoire Bordelais de Recherche en Informatique - URA 1304
Université Bordeaux I
351, cours de la Libération
33405 Talence, France Fax Number: 33 5 56 84 66 69
Phone Number: 33 5 56 84 24 31

{`Luc.Brun | Jean.Philippe.Domenger | Jean.Pierre.Braquelaire` }`@labri.u-bordeaux.fr`

In this paper, we present different recent segmentation works based on discrete maps. Discrete maps provide an efficient framework for region based segmentation methods. A discrete is a mixed model. It combines an encoding of the discrete boundaries of the image regions with topological graphs which represents the topology of the image.

Keywords

Topological graph, discrete map, segmentation.

1 Introduction

Region-based algorithms are based on the partition of the image into nonempty connected sets of pixels, called *region*. The segmentation processing consists in modifying the partition by *splitting* a region into sub-regions or by *merging* two adjacent regions to produce another region. Region selection is done using an evaluation criterion computed on one region or on one region and one of its adjacent regions. Among region-based algorithms, we distinguish three main approaches:

- The *top-down* approach [18, 16, 7] starts with an under-segmentation of the image and iterates split operations;
- The *bottom-up* approach [3, 10] starts with an over-segmentation of the image and iterates merge operations;
- The *mixed* approach [11, 12, 4, 8, 19] modifies the partition by combining split and merge operations.

The data structures commonly used by the top-down approach are hierarchical structures like *quadtrees* [9, 20] or *pyramids* [4, 19]. The splitting is efficiently performed by a refinement of the subdivision, but the computation of the adjacency of regions involves complex and costly processing. Another drawback implied by a regular subdivision is the "square aspect" of the final segmented image. The bottom-up approach is frequently implemented by *an array of labels* [17] combined with a *region adjacency graph*(RAG) [8, 3]. When the mixed approach is implemented both with *regular pyramids* and RAG, region merging must preserve the hierarchical structure. In this case, the merging operation is called *restricted merge*. Due to the incompatibility of data structures used by the split and the merge algorithms, the mixed approach is mainly divided in two parts: the first part alternates splits and restricted merges; the second part generates a RAG and allows merging of any adjacent regions.

Another region-based approach is proposed by Jolion [13]. This approach is based on the adaptive pyramid. During a first stage (bottom-up) an adaptive pyramid is built. The roots of the pyramid define the regions of the image. A second stage (top-down), allows the construction of a graph describing the components.

In this paper, we sum up different recent segmentation works [2, 5, 6] based on a data structure, called *discrete maps* [2], that combines an explicit encoding of the geometry of region boundaries with a topological description based on *topological graphs*. This data structure is well-adapted to mixed approach because the split and merge algorithm use the same data structure and these algorithms can be alternated without overhead. Moreover, this structure allows us to consider regions as 4-connected set of pixels having free geometry.

Section 2 briefly sums up the model of discrete maps. Section 3 presents the ability of this model to be used in a mixed approach. Section 4 is devoted to the presentation of an example of segmentation, using the model of discrete maps.

2 Representation of an image with discrete maps

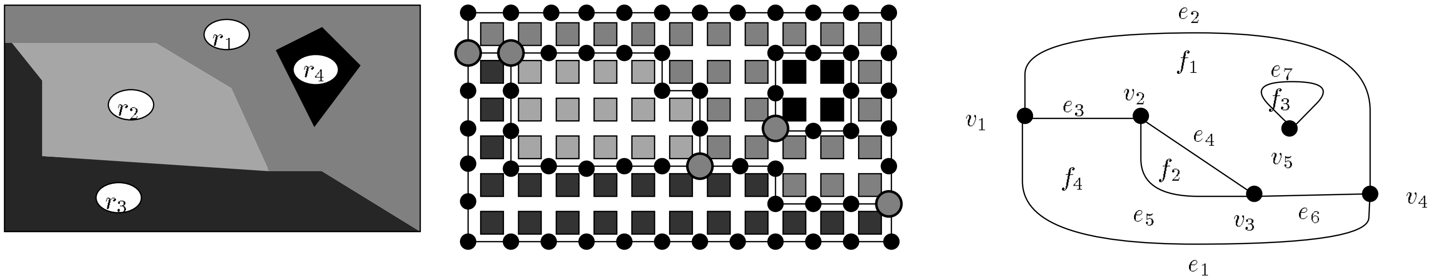
The topology of the subdivision of the Euclidean plane into connected components can be represented by a *topological graph* [21]. The aim of *discrete maps* is to associate a topological graph to a segmented image so that each face of this graph is homeomorphic to one region of the segmented image. To obtain this result, it is necessary to embed the segmented image into a digital topological plane having correct topological properties. By embedding the image in the plane $(\frac{1}{2}\mathbb{Z})^2$ provided with Khalimsky topology [14, 15], these following properties are established by Braquelaire and al. [2]

1. The boundary of a region r, denoted by $\partial(r)$, is made of discrete Jordan curves.
2. If r and r' are two 4-adjacent regions of the image, then $\partial(r) \cap \partial(r')$ is a set of Jordan arcs. Each arc contains at least three points.

Both of these properties allow us to define the topological graph of a segmented image subdivided into simply connected regions. The *faces* of this graph correspond to regions of the images; the edges correspond to Jordan arcs(called *segments*) shared by two 4-adjacent regions; the *vertices* are either points belonging at least to the boundaries of three regions, or arbitrary points selected on a closed Jordan arc included in the boundary of two regions. The geometrical element of a vertex is called a *node*.

A *discrete map* is defined as the drawing of a topological graph in the Khalimsky plane. If the image has n holed regions, its boundary set is the drawing of $n + 1$ discrete maps. We call the *infinite face* of a discrete map, the face representing the exterior of the discrete maps. Hence, an image can be represented by a set of discrete maps augmented with an inclusion relation linking each infinite face with the finite face that contains it.

Note that is not necessary to encode the geometry of the discrete maps in the plane $(\frac{1}{2}\mathbb{Z})^2$. In fact, it is sufficient to encode the boundary of regions in the plane corresponding to the closed points of the Khalimsky topology. This plane, called the *boundary plane*, is generated from $\mathbb{Z}^2$ by a a translation of a vector $(-\frac{1}{2}, -\frac{1}{2})$. The geometry of a discrete map encoding an image of size $h \times w$, is represented by an array $\mathcal{B}$ having a size $(h+1) \times (w+1)$. Let p be a boundary point; the value $\mathcal{B}[p]$ encodes the links which connect p to its neighboring boundary points. Thus, four bits are sufficient to encode an entry of $\mathcal{B}$ [2]. The figure 1 presents the discrete maps associated with an image composed with four regions labelled by $\{r_1, r_2, r_3, r_4\}$.



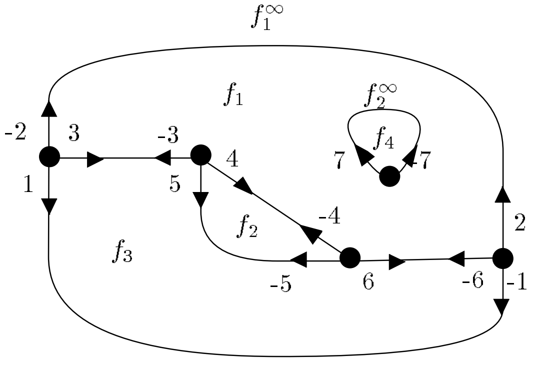
(a) Segmented Image (b) geometry of the discrete maps (c) Topological graphs of the discrete maps

Fig. 1. Part (a) shows the segmented image composed by four regions respectively labeled by r_1, r_2, r_3, r_4. Part (b) represent the boundary set of regions. The big circles are the nodes and the black circles are boundary points of segments. Part (c) describes the topological maps of the image, there exists two connected components in the graph because one region is holed.

A topological graph can be efficiently encoded using a **combinatorial map**. A combinatorial map is a tuple $(\mathcal{D}, \alpha, \sigma)$ such that $\mathcal{D}$ is a set of elements called darts, α and σ are two permutations defined on $\mathcal{D}$. Each dart represents an oriented edge of the topological map. An edge of the topological graph corresponds to an orbit of α, a vertex corresponds to an orbit of σ, an orbit of permutation $\varphi = \sigma \circ \alpha$ encodes a face. Note that if the darts are encoded by positive and negative integers, the permutation α can be implicitly encoded by $\alpha(d) = -d$. Let d be a dart and π a permutation, the orbit of π generated from d is denoted by $\pi^*(d)$.

Each face is identified by a *label*. A labeling function λ defined on $\mathcal{D}$ associates a label to each dart of face. Conversely the function λ^{-1} associates the label of a face with a canonical dart.

The inclusion relation between discrete maps is encoded by the function ***Parent*** and ***Children*** defined on the set of face labels. Let f be a finite face and $\{f_1^\infty, \ldots, f_n^\infty\}$ be the set of infinite faces included in f; then we have $\boldsymbol{Children}(f) = \{f_1^\infty, \ldots, f_n^\infty\}$ and $\forall\ f_i^\infty, \boldsymbol{Parent}(f_i^\infty) = f$. The combinatorial maps representing the topology of the image showed in figure 1 is displayed in figure 2. The two connected components of the graph are respectively associated with the combinatorial maps (σ_1, α_1) and (σ_2, α_2). The infinite face of the combinatorial map (σ_2, α_2) is $\varphi_2^*(-7) = (-7)$, so we have $\lambda(7) = f_2^\infty$ and $\boldsymbol{Parent}(f_2^\infty) = \lambda(-2) = f_1$.



$$\varphi_1 = (1\ 2)(-2\ -6\ -4\ -3)(-1\ 3\ 5\ 6)(4\ -5)$$
$$\sigma_1 = (-2\ 1\ 3)(-3\ 5\ 4)(-5\ 6\ -4)(-1\ 2\ -6)$$
$$\varphi_2 = (7)(-7)$$
$$\sigma_2 = (7\ -7)$$

Fig. 2. This figure represents the two combinatorial maps on the image displayed in figure 1. For instance, we have $e_1 = \alpha_1^*(1)$, $v_2 = \sigma_1^*(-3)$, the face $f_1 = \varphi^*(-2) = (-2\ -6\ -4\ -3)$

.

Using the array $\mathcal{B}$, an oriented segment S_d corresponding to a dart d may be traversed started from its node n if we know knowing its first move m. The association between the topology and the geometry of the discrete maps is given by the functions Σ and Σ^{-1} such that $\Sigma[(n,m)] = d$ and $\Sigma^{-1}[d] = (n,m) = S_d$. Let r be a simply connected region labeled by f_r, if $f_r = \varphi^*(\lambda(f_r)) = (d_1, \ldots, d_n)$ then

$$\partial(r) = S_{d_1} \oplus \ldots \oplus S_{d_n}$$

where $\oplus$ symbolizes the concatenation of oriented segments. In the case where the region is not simply connected, the other curves of the boundary are associated in the same way with the cycle of infinite faces $(\varphi^*(\lambda(f_i^\infty))$, with $f_i^\infty \in \boldsymbol{Children}(f_r))$. Since each closed boundary component validates the Jordan theorem, the region r_f can be traversed using the scan-line of seed-fill algorithms [2].

The model of discrete maps allows the computation of a complex homogeneity criterion on a region. We denote by $\mathcal{I}(r)$, the information attached to a region r allowing the computation of an homogeneity criterion. For instance, the homogeneity criterion defined by Beverige [1] requires, for a region r belonging to a grayscale image,

- μ, the mean gray value of r;
- σ, the standard deviation ;
- $|r|$, the number of pixels of r;
- $\partial(r)$, the boundary of r;
- $|\partial(r)|$, the perimeter of r;
- $|\partial(r) \cap \partial(r_i)|$, the length of the boundary shared by r and one of its adjacent region r_i.

The information $\mathcal{I}(r)$ can be subdivided in two parts:

1. Part $(\mu, \sigma, |r|)$ associated with the label f_r;
2. Part $(\partial(r))$ distributed on each segment S_{di} such that each di belongs either to $\varphi^*(\lambda(f_r))$ or to $(\varphi^*(\lambda(f_i^\infty))$ with $f_i^\infty \in \boldsymbol{Children}(f_r))$.

3 Region-based algorithms with discrete maps

Region-based algorithms are implemented with two operations, region splitting and region merging. The selection of the operation depends on two homogeneity criteria C_s and C_m. The criterion $C_s(r)$ determines the splitting of a region r; r must be split if $C_s(r) = false$. The criterion $C_m(r, r')$ determines if two adjacent region r and r' can be merged; r and r' are merged if $C_m(r, r') = false$. We suppose that the computation of these both criteria depends on the information $\mathcal{I}$ associated with regions. An image is wholly segmented when the both following conditions are validated:

1. For all regions r_i, $C_s(r_i) = true$
2. For all regions r_i, $C_m(r_i, r_j) = true$ with r_j an adjacent region of r_i.

A split and merge algorithm consists in the selection of the most heterogeneous region. The selected region is split into $\{r_1, \ldots, r_k\}$. Then, for each region r_i we merge r_i with one of its adjacent region r_j if $C_m(r_i, r_j) = false$. This algorithm is applied recursivly until both of the previous conditions are validated.

The main requirements for the split algorithm are:

1. Select a region r labeled by f_r such that $C_s(r) = false$;
2. Compute the boundary $\partial(r)$;
3. Traverse r in order to produce the list $\{S_1, \ldots, S_n\}$ of splitting segments.
4. Insert each segment S_i in the data structure;
5. Compute the list of new regions $r_1, \ldots, r_k$;
6. Update the inclusion relation between discrete maps;
7. Evaluate $\mathcal{I}(r_i)$ for each new region r_i.

As mentioned in section 2, the darts defining the boundary $\partial(r)$ are computed by traversing the orbits of $\varphi^*(\lambda^{-1}(f_r))$ and $\varphi^*(\lambda^{-1}(f_r^\infty))$ with $f_r^\infty \in \boldsymbol{Parent}(f_r)$. The insertion of segment S in the geometry data structure is done in a time complexity equal to $|S| + 2 * \times |S_{mean}|$, where $|S_{mean}|$ is the mean length of a segment (see [5] for further information). The insertion of an edge in the combinatorial map is direct. The updating of the inclusion relation between discrete maps, requires for each infinite face $f_i^\infty \in \boldsymbol{Children}(f_r)$ to recompute the function $\boldsymbol{Parent}$. The function $\boldsymbol{Parent}(f_i^\infty)$ is updated by the localization of the region containing a point belonging to the boundary of f_i^∞. The time complexity of the point localization is bounded by the diameter of the region r. The evaluation of $\mathcal{I}(r_i)$ may imply a traversal of the region r_i. Note that the information evaluated on the boundary can be computed during insertion of the segment. If we except this last requirement, the time complexity of the split is lower than $2 \times |r|$.

The main requirements for the merge algorithm are

1. Select a region r labeled by f_r;
2. Compute the adjacency-region set $\mathcal{V}(f_r)$ of the region r;
3. Select a region $r_i \in \mathcal{V}(f_r)$ such that $C_m(r, r_i) \; = \; false$;
4. Suppress the edges belonging to $\partial(r) \cap \partial(r_i)$
5. Update $\mathcal{I}(r) = \mathcal{I}(r \cup r_i)$.

The adjacency-region set is computed on the data structure by the adjacency relation between faces expressed by the following definition

Definition 1. Two finite faces f and f' are adjacent if they verify one of the following properties:

1. $\exists d \in \varphi^*(\lambda^{-1}(f))$ such that $-d \in \varphi^*(\lambda^{-1}(f'))$
2. $\exists d \in \varphi^*(\lambda^{-1}(f))$ and $\exists! f^\infty \in \boldsymbol{Children}(f')$ such that $-d \in \varphi^*(\lambda^{-1}(f^\infty))$
3. $\exists d \in \varphi^*(\lambda^{-1}(f'))$ and $\exists! f^\infty \in \boldsymbol{Children}(f)$ such that $-d \in \varphi^*(\lambda^{-1}(f^\infty))$

The time complexity of the computation of $\mathcal{V}(f_r)$ is linear in $\sum |\varphi^*(d_i)|$ where $d_0 = \lambda^{-1}(f_r)$ and $d_i = \lambda_1(f_r^\infty)$. Note that, with the same algorithm, we identify for each face $f_{r_i} \in \mathcal{V}(f_r)$ the edges belonging to $\partial(r) \cap \partial(r_i)$. The removal of an edge in the topological graph is direct, and the removal of a segment S is in $|S|$ because S must be traversed in order to be removed in the array $\mathcal{B}$ [5]. The updating of the inclusion relation is direct. If the information attached to two adjacent regions r and r' verifies $\mathcal{I}(r \cup r') = \mathcal{I}(r) \cup \mathcal{I}(r')$, then the fourth requirement of the merge can be done immediately.

4 A split and merge algorithm using discrete maps

The homogeneity criterion, used by the split and merge algorithm presented in [6], is based on the squared error defined by:

$$\boldsymbol{SE}(FS_r) = \sum_{x \in FS_r} g(x) ||x - \mu(r)||^2 \tag{1}$$

where FS_r denotes the feature space associated to region r, this feature space may be a set of values between $\{0, \ldots, 255\}$ for grayscale images or a three dimensional set for color images. The value $g(x)$ is the number of pixels with color x in region r and $\mu(r)$ is the mean of the set FS_r. Now, we will denote $\boldsymbol{SE}(r)$ for $\boldsymbol{SE}(FS_r)$. The criterion C_r is defined by

$$C_m(r_i, r) \; = \; false \Longleftrightarrow \boldsymbol{SE}(r) > \alpha \times \boldsymbol{SE}(I)$$

where α is a parameter given by the user, and $\boldsymbol{SE}$(I) is the square error of the initial image. At each step this algorithm split the region r_{max} having the greatest squared error. The splitting of r_{max} generates a set of sub-regions $\{r_1, \ldots, r_n\}$. Then, we apply a merger on the regions $\{r_1, \ldots, r_n\}$ using the criterion C_m defined by:

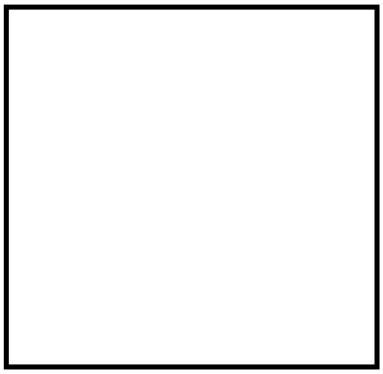
$$C_m(r_i, r) = false \Longleftrightarrow \boldsymbol{SE}(r \cup r_i) = Max_{j \in \{1, \ldots, n\}, r \in V(r_j)} \boldsymbol{SE}(r_j \cup r)$$

Note that this merge algorithm may merge one region r_i with a region which does not belong to $\{r_1, \ldots, r_n\}$. The resulting region will be added to the set $\{r_1, \ldots, r_n\}$. In order to restrict the set of regions considered by the merge algorithm, we use an array of marks indicating if a region can be merged or not.

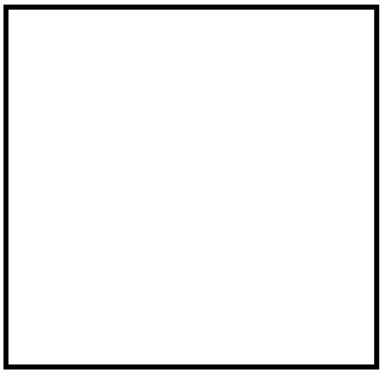
On the example displayed in figure 3, a region is marked if it results from one split step. Therefore, a region generated from a split step may be reconsidered in the following merge step.

5 Conclusion

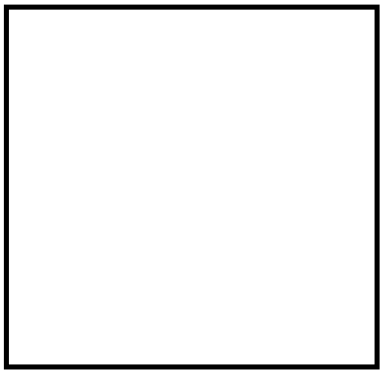
The model of discrete maps provides a framework to implement any region-based algorithms. The homogeneity criterion may be computed from information defined on region and on region boundary. Moreover, this model can be used in an interactive segmentation processing. Thus, the user may reduce the segmentation to an image part having free geometry and not necessary connected. It may also modifie the result of a segmentation, by incremental addition or removal of segments.



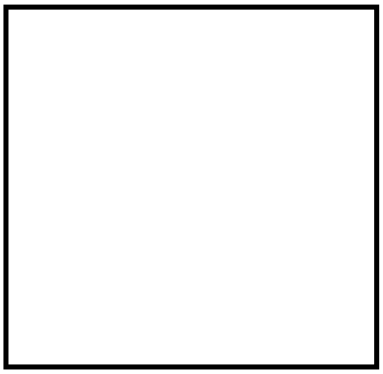
(a) Initial image



(b) First splitting step
2342 Nodes, 3662 Segments
2935 Finite faces, 1064 infinite faces



(c) First merging step
36 Nodes, 54 Segments
19 Finite faces



(d) Final image obtained after twenty iterations
515 Nodes, 778 Segments
288 Finite faces, 24 Infinite faces

Fig. 3. This figure displayed differents steps of the segmentation processing and indicates the main parameters of the discrete map model

References

1. J.R Beveridge, J. Griffith, R. Kohler, A.R. Hanson, and E.M. Riseman. Segmenting images using localized histograms and region merging. *International Journal of Computer Vision*, 2:311–347, 1989.
2. J.P. Braquelaire and J.P. Domenger. Representation of region segmented images with discrete maps. Technical report, RR-112797 LaBRI, available on http://www.labri.u-bordeaux.fr/LaBRI/Publications/index.html, June 1996.
3. R. Brice and C.L. Fennema. Scene analysis using regions. *Artificial intelligence*, 1:205–226, 1970.
4. J.D. Browning. Segmentation of pictures into regions with tile-by-tile method. *Pattern Recognition*, 15(1):1–10, 1982.
5. L. Brun and J.P. Domenger. Incremental modifications on segmented image defined by discrete maps. Technical report, RR-112696 LaBRI, available on http://www.labri.u-bordeaux.fr/LaBRI/Publications/index.html, may 1996.
6. Luc Brun and Jean Philipe Domenger. A new split and merge algorithm with topological maps and inter-pixel boundaries. To be published in WSCG'97, may 1996.
7. Mehmet Celenk. A color clustering technique for image segmentation. *Computer Vision, Gaphics, and Image Processing*, 52:145–170, 1990.
8. F. Cheevasuvut, H. Maitre, and D. Vidal-Madjar. A robust method for picture segmentation based on a split and merge procedure. *CVGIP*, 34:268–281, 1986.
9. R.C. Dyer, A Rosenfeld, and S Hanan. Region representation: Boundary codes from quadtrees. *ACM: Graphics and Image Processing*, 23(3):171–179, March 1980.
10. C. Fiorio and J. Gustedt. Two linear time union-find strategies for image processing. Technical Report 375/1994, Technische Universitat Berlin, 1994.

11. S.L. Horowitz and T. Pavlidis. Picture segmentation by a directed split-and-merge procedure. Technical report, Departement of Electrical Engineering, Princeton University , N. J. 08540, 1975.
12. S.L. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *Journal of The Association for Computing Machinery*, 23(2):368–388, April 1976.
13. J.M. Jolion and A. Montanvert. The adaptative pyramid : A framework for 2d image analysis. *Computer Vision, Graphics, and Image Processing*, 55(3):339–348, May 1992.
14. E. Khalimsky, R. Kopperman, and P.R. Meyer. Boundaries in digital planes. *Journal of applied Math. and Stocastic Analysis*, 3:27–55, 1990.
15. T.Y. Kong and A. Rosenfeld. Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48:357–39, 1989.
16. C.-H. Lee. Recursive region splitting at hierarchical scope views. *Computer Vision, Graphics, and Image Processing*, 33:237–258, 1986.
17. C. J. Nicol. A systolic approach for real time connected component labeling. *Computeer vision and Image understanding*, 61(1):17–31, January 1995.
18. R. Ohlander, K. Price, and D.R. Reddy. Picture segmentation using a recursive region splitting method. *Computer Graphics, and Image Processing*, 8:313–333, 1978.
19. M. Pietikainen, A. Rosenfeld, and I. Walter. Split and link algorithms for image segmentation. *Pattern Recognition*, 15(4):287–298, 1982.
20. H. Samet. Region representation: Quadtrees from boundary codes. *ACM : Graphics and Image Processing*, 23(3):163–170, March 1980.
21. W.T. Tutte. A census of planar maps. Canad.J.Math., 15:249–271, 1963.