Extensive Partition Operators, Gray-Level Connected Operators, and Region Merging/Classification Segmentation Algorithms: Theoretical Links

Daniel Gatica-Perez, *Student Member, IEEE*, Chuang Gu, *Member, IEEE*, Ming-Ting Sun, *Fellow, IEEE*, and Salvador Ruiz-Correa

***Abstract*—The relation between morphological gray-level connected operators and segmentation algorithms based on region merging/classification strategies has been pointed out several times in the recent literature. However, to the best of our knowledge, the formal relation between them has not been established. This paper presents the link between the two domains based on the observation that both connected operators and segmentation algorithms share a key mechanism: they *simultaneously operate on images and on partitions*, and therefore they can be described as operations on a joint image-partition model. As a result, we analyze both segmentation algorithms and connected operators by defining operators on complete product lattices, that explicitly model gray-level and partition attributes. In the first place, starting with a complete lattice of partitions, we initially define the concept of *segmentation model* as a mapping in a product lattice, whose elements are three-tuples consisting of a partition, an image that models the partition attributes, and an image that represents the gray-level model associated to the segmentation. Then, assuming a conditional ordering relation, we show that any region merging/classification segmentation algorithm can be defined as an extensive operator in such a complete product lattice. In the second place, we proposed a very similar lattice-based *extended representation of gray-level functions* in the context of connected operators, that highlights the mathematical analogy with segmentation algorithms, but in which the ordering relation is different. We use this framework to show that every region merging/classification segmentation algorithm indeed corresponds to a connected operator. While this result provides an explanation to previous work in the area, it also opens possibilities for further analysis in the two domains. From this perspective, we additionally study some theoretical properties of a general region merging segmentation algorithm.**

***Index Terms*—Extensive partition operators, gray-level connected operators, mathematical morphology, operators on complete lattices, region merging/classification segmentation algorihtms.**

Manuscript received June 21, 2000; revised May 16, 2001. D. Gatica-Perez was supported by the Fulbright-CONACYT Ph.D. Scholarship Program and by the National University of Mexico. S. Ruiz-Correa was supported by the Fulbright-CONACYT Ph.D. Scholarship Program. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Scott T. Acton.

D. Gatica-Perez and S. Ruiz-Correa are with the Department of Electrical Engineering and the Human Interface Technology Laboratory, University of Washington, Seattle, WA 98195 USA (e-mail: danielgp@hitl.washington.edu).

C. Gu is with Microsoft Corporation, Redmond, WA 98052 USA.

M.-T. Sun is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA.

Publisher Item Identifier S 1057-7149(01)07470-X.

I. INTRODUCTION

MATHEMATICAL morphology [25] pursues the design of nonlinear operators to process and analyze images and extract useful information from them, based on the development of both image models and a mathematical theory that describes fundamental properties of the desired image operators. Complete lattices constitute one appropriate algebraic framework for such task [26], [15], thus framing mathematical morphology as the study of operators on complete lattices.

Connected operators represent one of the main contributions of mathematical morphology to binary and gray-level image processing and analysis [28]. They are characterized by the peculiar property of preserving image contours, because they operate on a *regional* basis: they transform an image by selectively altering the intensity of connected sets of pixels of constant grey-level, called *flat zones*. This important characteristic makes this class of operators ideal for image simplification and for other applications where contour preserving is necessary, like image and video segmentation [7], [22], [23].

In the same morphological framework, the fundamental, yet unsolved problem of *image segmentation* can be conceived as an issue of designing *operators in lattices of partitions*. This formalism has attracted recent attention for video analysis [8] and video object extraction [9].

The relation between image segmentation and connected operators goes beyond the application domain. In fact, the similarities between connected operators and segmentation algorithms based on *region merging/region classification* have been pointed out several times in the literature [5], [8], [9]. On one hand, the flat zones of a gray-level image induce a partition of the image support, and the definition of a connected operator itself is based on the relation between the flat-zone partitions induced by the input and output images. In other words, connected operators create a hierarchy of flat-zone partitions, embedded one in each other by an ordering relation. On the other hand, region merging algorithms also create a hierarchy of partitions [20], [33], [13]. The relation between connected operators and segmentation techniques was exploited in [5], in which a region merging segmentation method based on the flat zone concept was proposed. Segmentation is viewed as a region number reduction problem, in which flat zones of an image are either preserved or merged; therefore this algorithm behaves as a connected operator. The intuitive links between the two domains was further employed in [8], which proposed similar algorithms for definition of both connected operators and region merging methods. However, gray-level connected operators and partition lattice operators map elements in different domains, and, to the best of our knowledge, the formal relation between them has not been established.

In this paper, we propose a way of interpreting such connection, based on the observation that connected operators and segmentation algorithms share a key mechanism: they simultaneously operate on an image *and* on a partition, and therefore, they can be described as operations on a joint image-partition model. As a result, we analyze both segmentation algorithms and connected operators by defining operators on complete product lattices, that explicitly model gray-level and partition attributes. In the first place, starting with a complete lattice of partitions, we initially define the concept of *segmentation model* as a mapping in a product lattice, whose elements are three-tuples consisting of a partition, an image that models the partition attributes, and an image that represents the gray-level model associated to the segmentation. Then, assuming a conditional ordering relation, we show that any region merging/classification segmentation algorithm can be defined as an extensive operator in such a complete product lattice. In the second place, we proposed a very similar, *extended representation of gray-level functions* in the context of connected operators, that highlights the mathematical analogy with segmentation algorithms. Such a representation allows for the formulation of the main result of this paper, namely, that each generic region merging/classification segmentation operator indeed corresponds to a connected operator. While this result provides an explanation to previous work in the area [5], [8], it also opens possibilities of further analysis in the two domains. From this perspective, we additionally study some theoretical properties of a general region merging segmentation algorithm, as proposed in [8].

The rest of the paper is organized as follows. Section II reviews the concepts from complete lattice theory that are necessary for our development. Section III proposes the complete lattice framework for segmentation models and generic region merging algorithms. Section IV reviews the basic notions of gray-level connected operators, and then presents a product lattice model for gray-level functions that is convenient to analyze the links between connected operators and segmentation algorithms. Section V presents the analysis of the links between the different types of operators. From this formulation, Section VI further illustrates the approach by presenting the analysis of a general region merging operator. Section VII provides some final thoughts. Finally, an Appendix concentrates the mathematical details, summarizing notation and proofs for all the stated properties.

II. Fundamentals of Complete Lattice Theory

A. Definitions

In this section we present some basic definitions from complete lattice theory for sake of completeness and clarity of notation, as discussed in [26], [6], [15], [19]. The reader is referred to these references for further treatment. For a quick reference, the Appendix contains a glossary of the main symbols used in the rest of the paper.

Definition 1: A **complete lattice** $\mathcal{L}$ is a set with a partial order relation $\leq$ for which each of its subsets $\mathcal{K} \subset \mathcal{L}$ has an *infimum* (greatest lower bound) $\wedge\mathcal{K}$, and a *supremum* (least upper bound) $\vee\mathcal{K}$. When $\mathcal{K} = \{X_1, X_2, \ldots, X_n\}$, infimum and supremum are denoted by $X_1 \wedge X_2 \wedge \cdots \wedge X_n$, and $X_1 \vee X_2 \vee \cdots \vee X_n$, respectively. $\wedge$ and $\vee$ are binary operations on $\mathcal{L}$, hence $(\mathcal{L}; \wedge, \vee)$ constitutes an algebraic structure.

Let $\mathcal{L}$ and $\mathcal{M}$ be complete lattices. It can be shown that the set of all **mappings** or **operators** $\Gamma = \{\gamma_i\}$ from $\mathcal{L}$ to $\mathcal{M}$ inherits the partial ordering structure by defining $\gamma_i \leq \gamma_j \leftrightarrow \gamma_i(X) \leq \gamma_j(X), \forall X \in \mathcal{L}$, and also constitutes a complete lattice. A number of properties of operators on complete lattices that will be studied in the rest of the paper can be enumerated as follows.

Definition 2: Let $\mathcal{L}, \mathcal{M}$, and $\mathcal{S}$ be complete lattices.

1) The **identity operator** on $\mathcal{L}$ $(\mathrm{id}_{\mathcal{L}})$ maps every element of $\mathcal{L}$ onto itself.
2) The **composition** of operators is defined in the usual way: if $\alpha : \mathcal{L} \rightarrow \mathcal{M}$, and $\beta : \mathcal{M} \rightarrow \mathcal{S}$, then the operator $\psi : \mathcal{L} \rightarrow \mathcal{S}$ is defined by $\psi(X) = \beta(\alpha(X)) = (\beta\alpha)(X), \forall X \in \mathcal{L}$.
3) An operator $\psi : \mathcal{L} \rightarrow \mathcal{M}$ is called
   a) **increasing** iff it is order-preserving, i.e., $X \leq Y \leftrightarrow \psi(X) \leq \psi(Y), \forall X, Y \in \mathcal{L}$;
   b) **lattice isomorphism** iff it is a bijection (injective or one-to-one, and surjective or onto), and if both ψ and its inverse ψ^{-1} are increasing. Lattice isomorphisms preserve infima and suprema, that is, $\psi(X \wedge Y) = \psi(X) \wedge \psi(Y)$, and $\psi(X \vee Y) = \psi(X) \vee \psi(Y)$.
4) An operator $\psi : \mathcal{L} \rightarrow \mathcal{L}$ is called
   a) **extensive** iff $X \leq \psi(X)\ \forall X \in \mathcal{L}$. Similarly, decreasing and anti-extensive operators can be defined;
   b) **idempotent** iff $\psi(\psi(X)) = \psi(X)\ \forall X \in \mathcal{L}$;
   c) **morphological filter** iff it is increasing and idempotent;
   d) **opening** (resp. **closing**) iff it is an anti-extensive (resp. extensive) morphological filter.
5) A **left inverse** $\psi^{\leftarrow}$ of an injective operator $\psi : \mathcal{L} \rightarrow \mathcal{M}$ is a surjection $\psi^{\leftarrow} : \mathcal{M} \rightarrow \mathcal{L}$ such that the composition $\psi^{\leftarrow}\psi$ is the identity on $\mathcal{L}, \mathrm{id}_{\mathcal{L}}$.
6) A **right inverse** $\psi^{\rightarrow}$ of a surjective operator $\psi : \mathcal{L} \rightarrow \mathcal{M}$ is an injection $\psi^{\rightarrow} : \mathcal{M} \rightarrow \mathcal{L}$ such that the composition $\psi\psi^{\rightarrow}$ is the identity on $\mathcal{M}, \mathrm{id}_{\mathcal{M}}$.
7) The **invariance domain** $\mathrm{Inv}(\psi)$ of an operator ψ is the set of all elements $X \in \mathcal{L}$ such that $\psi(X) = X$. Each X is called a **fixpoint** or **invariant** under ψ.

B. Lattices of Gray-Level Functions and Partitions

Gray-level (color) images and image segmentation have a clear formulation from a lattice-theoretic perspective, by using complete lattices of scalar (vector) functions and of partitions, respectively. We review such definitions in the following.

Definition 3: Let $\mathcal{L}$ be a complete lattice and E an arbitrary nonempty set. Then the **set of all functions** $f_i : E \rightarrow \mathcal{L}$, denoted by $\mathrm{Fun}(E, \mathcal{L})$, is a complete lattice of mappings, with the partial order relation defined by $f_i \leq f_j \leftrightarrow f_i(x) \leq f_j(x), \forall x \in E, f_i, f_j \in \mathrm{Fun}(E, \mathcal{L})$ [15]. The supremum and infimum are also pointwise-defined:

$$\left(\bigwedge_i f_i\right)(x) = \bigwedge_i (f_i(x));$$
$$\left(\bigvee_i f_i\right)(x) = \bigvee_i (f_i(x)), \quad x \in E.$$

The particular case in which $E \subset \mathcal{Z}^2$, where $\mathcal{Z}$ denotes the set of integers, and $\mathcal{L} = \{0, 1, \ldots L-1\}$, defines a **complete lattice of discrete, gray-level images** quantized to L levels, which is also denoted by $\mathrm{Fun}(\mathcal{Z}^2)$, or simply by $\mathrm{Fun}(\cdot)$ in this paper. Furthermore, the case $L = 2$ corresponds to binary images.

Definition 4: Given a set E and its powerset $\mathcal{P}(E)$, a **partition** of E is a mapping $P : E \to \mathcal{P}(E)$ such that $\forall x, y \in E$, (i) $x \in P(x)$, and (ii) $P(x) = P(y)$ or $P(x) \cap P(y) = \emptyset$ [26]. $P(x)$ is called the *zone* or *region* of P that contains x. It can be proved that the set of all partitions of the space E constitutes a complete lattice of mappings [30], denoted by Π, where the partial ordering relation is defined as $P_i \leq P_j \leftrightarrow P_i(x) \subseteq P_j(x), \forall x \in E, P_i, P_j \in \Pi$. In this case, P_i is said to be *finer* than P_j. The infimum of a set of partitions indexed by i is given by

$$\left(\bigwedge_i P_i\right)(x) = \cap_i P_i(x) \quad \forall x \in E,$$

that is, the infimum is the partition made of the intersections of all regions in the original set of partitions. On the other hand, the supremum is given by

$$\left(\bigvee_i P_i\right)(x) = \cap\{B : B = \cup_i \cup_{y \in B} P_i(y), x \in B, B \in \mathcal{P}(E)\},$$

which is the finest partition that is larger than each of the individual P_i. For the two-partition case, $(P_i \vee P_j)(x) = (P_i \vee P_j)(y)$ if $P_i(x) = P_i(y)$ or $P_j(x) = P_j(y)$. Additionally, the least and greatest elements of Π correspond to the finest partition P_O and the coarsest partition P_I, such that $P_O(x) = x$ and $P_I(x) = E$ for all $x \in E$. In the case of a discrete image, P_O corresponds to the partition that has as many regions as pixels in the image, and P_I corresponds to a partition that consists of a single region that includes all pixels.

The previous two Definitions constitute the basic models for image processing and analysis, namely, *images and partitions of the image support*. On one hand, a lattice of partitions represents a proper framework to analyze segmentation operations [26], [5], [8], and will be analyzed in more detail in Section III. On the other hand, there has been considerable progress on the analysis and design of operators on $\mathrm{Fun}(\mathcal{Z}^2)$, also known as *gray-scale morphology* [26], [15]. In particular, the class of connected operators, discussed in Section IV, has also demonstrated its applicability for image simplification and segmentation purposes [28], [4], [16]. The next two sections will highlight the symmetries between the two domains, by developing a representation for both classes of operators, helpful for disclosing their relations.

III. Extensive Partition Operators and Segmentation Algorithms

As a blackbox, segmentation algorithms essentially establish a rule of correspondence between the original partition of an image support-usually the finest partition P_{O}– and a new one [30]. In morphological terms, this can be defined as follows [26].

Definition 5: A **partition lattice operator** ψ is a mapping from a complete lattice of partitions into itself $\psi : \Pi \to \Pi$.

Some basic operators were originally proposed in [26]. More recent studies include [8] and [9]. As discussed in these references, the segmentation techniques based on *region merging* [12], [8], [13] and *region classification* [2], [3], [11], [9] correspond to *extensive* partition operators, as the original partitions are finer than the transformed ones $(P_i \leq \psi(P_i), \forall P_i \in \Pi)$.

However, as we discuss in the following, a region merging/classification segmentation algorithm makes use of additional information other than the sole partition information. On one hand, region merging methods are iterative, and require three elements for their specification: a *region model* that represents the attributes (gray-level/color, size) of each of the regions of a partition; a *merging order* that defines the order in which two neighboring regions will be used as candidates for merging; and a *merging criterion* that defines whether the selected regions should be merged or not [8]. The merging criterion includes the definition of a termination criteria for the merging process. A typical example of a region merging segmentation algorithm is shown in Fig. 1(c). Given the original image f and an initial partition P, a region merging algorithm has been applied to generate the partition $P_m = \psi_m(P)$ in Fig. 1(c). In this algorithm, the region model consists in the median gray-level of each region, the merging order is given by the minimum size of any two adjacent regions times their region model difference, and the merging criterion consists in merging two regions whenever either of their size is less than a threshold (refer to [8] for details). On the other hand, region classification algorithms [3], [2], [9], [11] assign a class label to the regions of a partition, given statistical criteria. For their specification, such algorithms also require three elements: *reference information*, that defines the number of classes and their attributes; a *region model*, that specifies the corresponding features for each zone in a partition; and a *classification criterion*, that establishes the rule for class label assignment to each of the regions. An example of a region classification segmentation algorithm is shown in Fig. 1(d). The partition $P_c = \psi_c(P)$ has been generated assuming two classes C_0, C_1, and a Gaussian mixture intensity model for each class, that have been learned using training data. The region model remains the same as before, and the regional classification criterion is maximum *a posteriori* (MAP)

$$C^*(x) = \arg\max_j \Pr(C_j \,|\, \theta(x)), \quad j \in \{0, 1\}$$

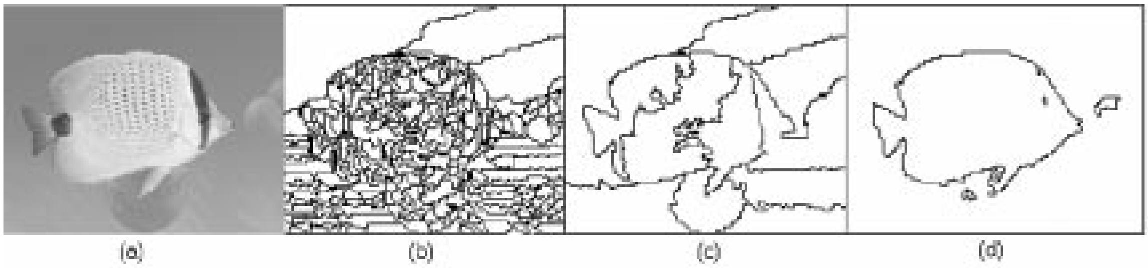

Fig. 1. (a) *Bream*, (b) initial partition of the image support, (c) partition generated by region merging, and (d) partition generated by region classification.

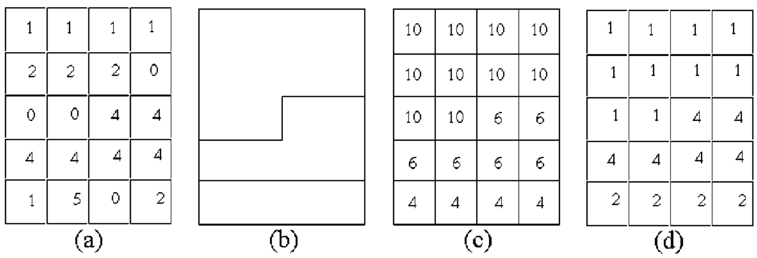

Fig. 2. (a) A digital image f and (b) a partition P generate two components for a region model: one (Θ_A) that reflects properties of the partition, and another (Θ_G) that characterizes the gray-level properties and hence depends both of f and P. In particular, $\Theta_A(P)$ represents (c) the cardinality (area) of each region, and $\Theta_G(f, P)$ represents (d) their median gray-level.

where $\theta(x)$ denotes the model of $P(x)$, the region of P that contains x. Note that in this operation each region is classified independently.

One could ask whether these two very different conceptions for segmentation have anything in common, and if so, how to formalize such coincidences from a lattice-theoretic perspective. To this end, we start in this section by defining a segmentation model in a complete lattice framework, and then conceive a generic region merging/classification segmentation algorithm as an extensive operator in a complete product lattice. Such formalism not only will allow for the study of theoretical properties of segmentation algorithms, but also will be the clue to establish their links with gray-level connected operators.

A. Segmentation Models in a Lattice-Theoretic Formulation

From the previous description and Fig. 1, we notice that the region model constitutes a fundamental common point in region merging/classification algorithms. In general, given an image $f \in \mathrm{Fun}(\mathcal{Z}^2)$ and a partition P, a region model can be characterized by two main components. The first one consists of a set of attributes that is *solely* related to characteristics of the partition, like the cardinality of each of the regions (related to its size or area), the length of its convex hull (related to their shape), and so on [1]. The second component consists of a set of features that model the gray-level/color value of each region being the mean or median region value two typical representations [8]. This paper will analyze the case of single partition attributes and gray-level images; the case of multiple attributes (color images, multiple shape/size features) represents a problem of multivalued morphology [27], [10] and will be analyzed elsewhere. The region model of a segmentation can therefore be formally described as composed of the following two operators.

Definition 6: Let $f \in \mathrm{Fun}(\mathcal{Z}^2)$ and $P \in \Pi$ be a gray-level image and a partition of the image support, respectively. Then,

1) an **attribute region model** is a mapping $\theta_A : \mathcal{P}(E) \rightarrow \mathcal{Z}$, that can be extended as a mapping $\Theta_A : \Pi \rightarrow \mathrm{Fun}(\mathcal{Z}^2)$, so that $\Theta_A(P)(x) = \theta_A(P(x))$. The image $\Theta_A(P)$ is called **partition attribute image**.
2) a **gray-level region model** is a mapping $\Theta_G : \mathrm{Fun}(\mathcal{Z}^2) \times \Pi \rightarrow \mathrm{Fun}(\mathcal{Z}^2)$. The image $\Theta_G(f, P)$ is referred to as **gray-level model image**.

Fig. 2 illustrate the model definitions.

Clearly, the mappings Θ_A are not injective in general when size or shape are used as attributes. This situation is illustrated in Fig. 3. The same holds for $\Theta_G(f, P)$, as shown in Fig. 4: given a partition, several images map to the same gray-level model image. It will be convenient to define the equivalence relation in Fun( ) induced by Θ_G.

Definition 7: The **equivalence relation** Q_{Θ_G} in $\mathrm{Fun}(\mathcal{Z}^2)$ induced by Θ_G is given by

$$\forall f_1, f_2, \quad f_1 \equiv_{Q_{\Theta_G}} f_2 \leftrightarrow \Theta_G(f_1, P) = \Theta_G(f_2, P) \qquad (1)$$

where $f_1 \equiv_{Q_{\Theta_G}} f_2$ denotes the equivalence of f_1 and f_2 under the relation Q_{Θ_G}. Note that P is fixed in this definition. It is easy to see that Q_{Θ_G} is indeed an equivalence relation (reflexive, symmetric, and transitive). Let $[f_i]_{Q_{\Theta_G}} = \{f_j \in \mathrm{Fun}(\cdot) \,|\, f_j \equiv_{Q_{\Theta_G}} f_i\}$ denote the i-th equivalence class. For each of them, let $f_{i_{\Theta_G}}$ be the element such that for every f_j in the class, $f_{i_{\Theta_G}} = \Theta_G(f_j, P)$. By definition, $f_{i_{\Theta_G}}$ is an invariant or fixpoint of Θ_G $(f_{i_{\Theta_G}} = \Theta_G(f_{i_{\Theta_G}}, P))$ [15]. Fig. 4(b) illustrates this situation.

The previous mappings are the pieces needed to properly define a segmentation model for a partition.

Definition 8: Let $\mathcal{T} = \Pi \times \mathrm{Fun}(\mathcal{Z}^2)^2$, and let t_i denote an arbitrary element of $\mathcal{T}$, with components indexed by k, i.e., $t_i = \{t_{ik}\}, k \in \{0, 1, 2\}$. Given an image f, a **segmentation**

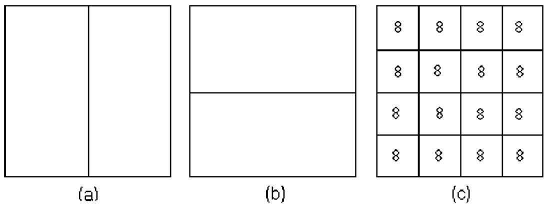

Fig. 3. When region size (cardinality) is used as the partition attribute $(\Theta_A(P)(x) = \mathrm{card}(P(x)))$ two partitions (a) P_1 and (b) P_2 can be mapped to the same partition attribute image (c) $\Theta_A(P_1)$.

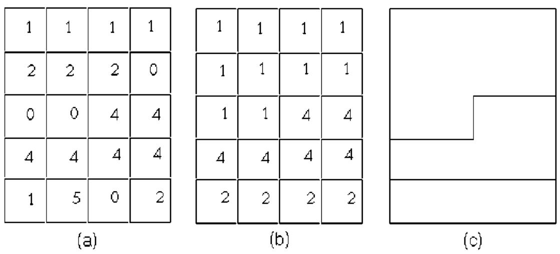

Fig. 4. When the truncated mean is used as the gray-level region model, two images (a) f_1, and (b) f_2, given P in (c), can be mapped to the same the gray-level model image: $\Theta_G(f_1, P) = \Theta_G(f_2, P) = f_2$.

model generator of a partition P is a mapping $\sigma^f : \Pi \to \mathcal{T}$ defined by

$$\sigma^f(P) \triangleq (P, \Theta_A(P), \Theta_G(f, P)), \quad \forall P \in \Pi. \tag{2}$$

The three-tuple $(P, \Theta_A(P), \Theta_G(f, P))$, composed of the partition, its partition attribute image, and its gray-level model image, is called **segmentation model**.

It is clear that each mapping $\sigma^f = (\mathrm{id}_\Pi, \Theta_A, \Theta_G)$ is an injection. However, it is not a surjection; its range $\mathrm{Ran}(\sigma^f)$ is given only by a set of three-tuples of the form $\{t_i : t_i = (P_i, \Theta_A(P_i), \Theta_G(f, P_i))\}$ which is only a proper subset of $\mathcal{T}$.

In order to define the product set $\mathcal{T}$ as a lattice, we need to impose a partial order on it. Additionally, we would like that such partial order could define the segmentation model generators σ^f as order-preserving, so that the ordering properties on Π—the lattice of partitions—are inherited on $\mathcal{T}$—the lattice that include all possible segmentation models. We discuss two important cases.

1) A *marginal* ordering in a product lattice [28]

$$t_i \le t_j \leftrightarrow t_{ik} \le t_{jk} \quad \forall t_i, t_j \in \mathcal{T}, \quad k \in \{0, 1, 2\} \tag{3}$$

defines $\mathcal{T}$ as a complete lattice, with the infimum and supremum obtained component-wise, $(\wedge t_i)_k = \wedge t_{ik}; (\vee t_i)_k = \vee t_{ik}$. However, it can be shown that the ordering properties in Π are lost when expanding the space to $\mathcal{T}$ under this marginal ordering.

2) A *conditional* ordering imposes the highest priority on the first component of any two elements $t_i, t_j \in \mathcal{T}$, and the lowest priority on their last component [28]

$$t_i \le_P t_j \leftrightarrow \begin{cases} t_{i1} < t_{j1} \\ t_{i1} = t_{j1}, \quad t_{i2} \le t_{j2} \\ t_{i1} = t_{j1}, \quad t_{i2} = t_{j2}, \quad t_{i3} \le t_{j3}, \end{cases} \tag{4}$$

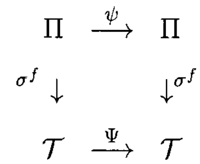

Fig. 5. Schematic representation of the relations between the lattice of partitions Π and the lattice $\mathcal{T}$ that contains all segmentation models as a proper subset. ψ denotes an arbitrary partition operator, σ^f represents a segmentation model generator, and Ψ corresponds to an operator on $\mathcal{T}$. If ψ is extensive, and (7) is satisfied, Ψ is a generic region merging operator and is denoted by Ψ_R.

where the subscript in the ordering relation indicates that the priority belongs to the first component of the three-tuples (P). Equation (4) corresponds to a lexicographic order in a vector space. It is simple to show that this is indeed a partial order relation and that, by defining the infimum and supremum component-wise, $\mathcal{T}$ constitutes a complete lattice [27].

Given that Π and $\mathcal{T}$ are both complete lattices, the set of all possible mappings $\Gamma = \{\gamma : \Pi \to \mathcal{T}\}$ also constitutes a complete lattice, with partial ordering defined by $\gamma_i \le \gamma_j \leftrightarrow \gamma_i(P) \le \gamma_j(P), \forall P \in \Pi$. However, the collection of all segmentation model generators $\Sigma = \{\sigma^f : \Pi \to \mathcal{T}\}$, constitutes only a proper subset of Γ, and is not a complete lattice [15].

Note that the mappings σ^f are clearly order-preserving under $\le_P$. Furthermore, as every σ^f is an injection, it has left inverses $(\sigma^f)^{\leftarrow}$, which correspond to a projection of $t_i \in \mathcal{T}$ onto its first component, and therefore are surjective. In fact, the left inverses are increasing; if

$$t_i \le t_j \to P_i = (\sigma^f)^{\leftarrow}(t_i) \le (\sigma^f)^{\leftarrow}(t_j) = P_j. \tag{5}$$

This property can be summarized as follows.

Property 1: The **set of segmentation model generators** $\Sigma = \{\sigma^f\}$, and the associated left inverses, given the conditional partial ordering $\le_P$, are *order-preserving*.

B. Generic Region Merging Operators

Using the conditional partial ordering for $\mathcal{T}$—that now shares ordering properties with Π—we can define segmentation algorithms as a class of operators on this lattice. Let $\mathcal{U}$ denote the complete lattice of all operators from $\mathcal{T}$ into itself, and let ψ be any partition operator. As illustrated in Fig. 5, it is possible to define a corresponding operator $\Psi \in \mathcal{U}$ by

$$\Psi = \sigma^f \psi (\sigma^f)^{\leftarrow}. \tag{6}$$

From the class of operators $\{\Psi\} \subset \mathcal{U}$ that model segmentation algorithms, we are interested in those that correspond to region merging/classification. This concept can be defined as follows.

Definition 9: Given an image f, a **generic region merging operator** is an extensive mapping $\Psi_R : \mathcal{T} \to \mathcal{T}$ defined by $\Psi_R = \sigma^f \psi (\sigma^f)^{\leftarrow}$, where $\psi : \Pi \to \Pi$ is an extensive operator in a lattice of partitions, such that for all the three-tuples $t_i \in \mathrm{Ran}(\sigma^f)$

$$\begin{aligned}\Psi_R(&P_i, \Theta_A(P_i), \Theta_G(f, P_i)) \\ &= (\psi(P_i), \Theta_A(\psi(P_i)), \Theta_G(f, \psi(P_i))).\end{aligned} \tag{7}$$

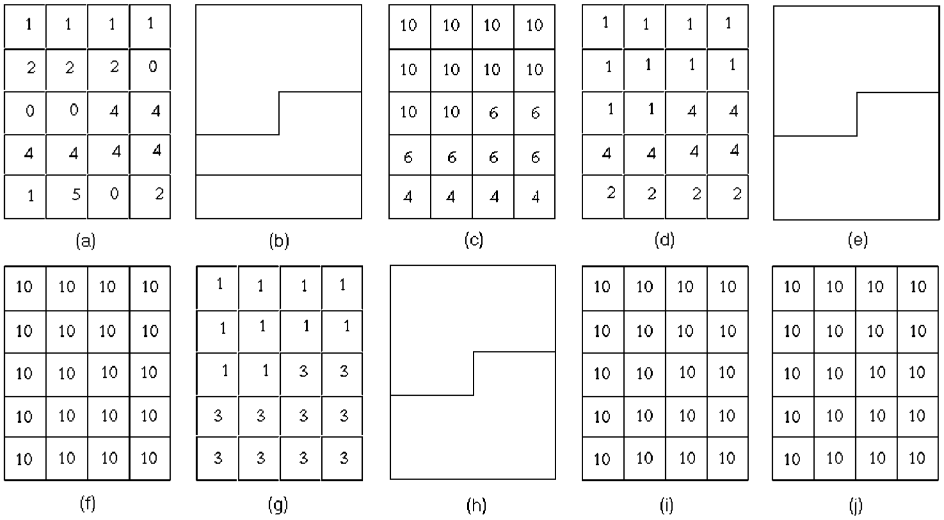

Fig. 6. A region merging operator Ψ_R, that fuses the smallest region in the partition with the smallest neighboring region, is applied on the image f (a). The input (t_1) and output $(t_2 = \Psi_R(t_1))$ segmentation models, are shown in (b)–(d) and (e)–(g), respectively. The partition attribute images $\Theta_A(P_1)$ (c) and $\Theta_A(P_2)$ (f), represent the size of each region in the corresponding partition. The gray-level model images $\Theta_G(f, P_1)$ (d) and $\Theta_G(f, P_2)$ (g), represent the truncated region mean. In (h)–(j), we show an "invalid" output three-tuple if $\Psi_R(\,)$ is a generic region merging operator: given the function f in (a), the gray-level model image in (j) cannot be obtained from the partition P in (h).

The class of generic region merging operators $\mathcal{R} = \{\Psi_R\} \subset \mathcal{U}$ consists of all the operators from $\mathcal{T}$ to $\mathcal{T}$ associated to extensive operators in Π, and that map three-tuples $t_i \in \mathrm{Ran}(\sigma^f)$ into other "valid" three-tuples $t_j \in \mathrm{Ran}(\sigma^f)$. Note that the previous definition does not specify the action of the operators on those $t_i \notin \mathrm{Ran}(\sigma^f)$, other than $t_i \leq \Psi_R(t_i)$. Fig. 6 illustrates these facts.

The analysis of some properties of a general region merging operator under this formulation is delayed until Section VI. In the next section, we first review the notion of connected operators, and based on their similarities with region merging segmentation algorithms [5], [8], we propose an extended representation for gray-level functions.

IV. Connected Operators and Extended Representation for Gray-Level Functions

A. Connected Operators

Connected operators are based on the notion of connectivity. The following definitions introduce the main concepts, and are borrowed from [28] and [16].

Definition 10: Let $E \subset \mathcal{Z}^2$ equipped with the usual 8-connectivity. A **connected component** or **grain** of a set $Y \subset \mathcal{Z}^2$ is a set $C^Y \subseteq Y$ in which for each pair of pixels there exists a path (a set of points, all included in the set) that joins them.

Definition 11: Let C_x^Y denote the grain of $Y \subset \mathcal{Z}^2$ that contains the point x, and let $f \in \mathrm{Fun}(\cdot)$. A **flat zone** of f at level t is a grain $C^{Y_f^t}$ of the level set $Y_f^t = \{y \in \mathcal{Z}^2 \mid f(y) = t\}$, i.e., it is a maximal connected component of $\mathcal{Z}^2$ where the image f is constant.

It is easy to see that the flat zones of f induce a partition of the image support.

Definition 12: A **flat-zone partition** is a mapping $P_{\mathrm{FZ}} : \mathrm{Fun}(\cdot) \to \Pi$, such that

$$P_{\mathrm{FZ}}(f)(x) = C_x^{Y_f^{f(x)}}. \tag{8}$$

Definition 13: An operator $\phi : \mathrm{Fun}(\cdot) \to \mathrm{Fun}(\cdot)$ is called **connected** if

$$P_{\mathrm{FZ}}(f) \leq P_{\mathrm{FZ}}(\phi(f)), \quad \forall f \in \mathrm{Fun}(\cdot). \tag{9}$$

In words, the partition of flat zones of f is said to be *finer* than the partition of flat zones of $\phi(f)$. $\phi(f)$ is also called a planing of f [18].

A simple example that illustrates the previous concepts is shown in Fig. 7 for a four-tone image. Figure 7(b) shows the corresponding flat-zone partition, $P_{\mathrm{FZ}}(f)$.

As connected operators operate at the flat zone level, they preserve contours in an image: no new discontinuties can be introduced, but only changes of regional (flat-zone) grey-level intensity. In a few words, the effect of a connected operator in an image consists on 1) *merging* and 2) *recoloring* of its flat zones [16]. This is illustrated in Fig. 7(c)–(d), that present the result of applying a hypothetical connected operator ϕ on f, and its corresponding flat-zone partition, $P_{\mathrm{FZ}}(\phi(f))$.

General as it is, the definition of a connected operator does not specify the mechanisms by which the colors of an image should be transformed. Different specifications have generated a variety of classes of binary and gray-level connected operators, obtained either by imposing some theoretical properties (extensiveness, idempotence, increasingness), or by designing them to perform a particular function (e.g., elimination of small, dark/bright or low-contrast areas). Furthermore, some classes of gray-level operators are important extensions of their binary counterparts (flat connected operators). The reader may refer to [28], [4], [21], [16], [18], [17] for a detailed exposition on the subject. In particular, anti-extensive filters have been commonly

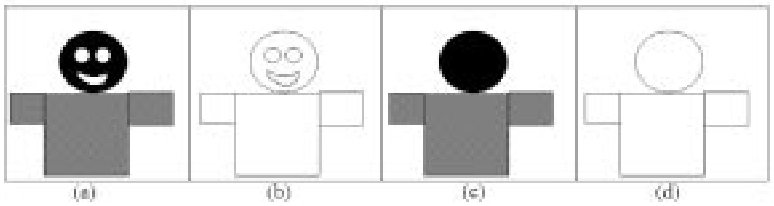

Fig. 7. Gray-level connected operator. (a) Original image f; (b) flat-zone partition $P_{\rm FZ}(f)$; (c) $\phi(f)$, where ϕ is a connected operator in which the grains with area smaller than a certain threshold have their gray-level value substituted by the value of the neighboring grain with the largest area; and (d) flat-zone partition $P_{\rm FZ}(\phi(f))$.

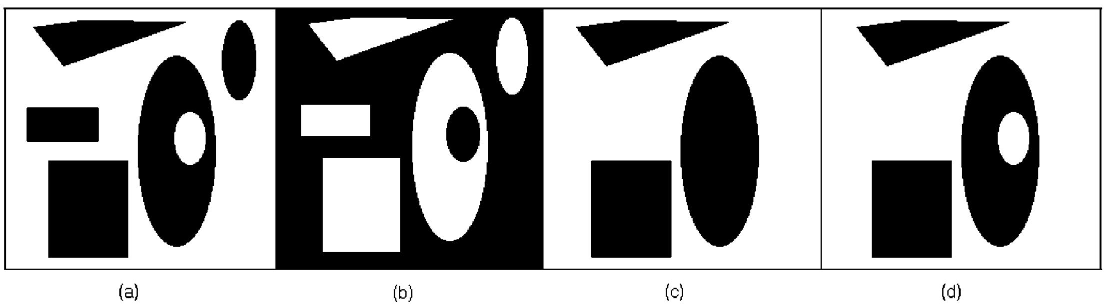

Fig. 8. Examples of binary connected operators to explain the extended gray-level representation. (a) Original binary image, (b) switching operator, (c) area switching operator, and (d) area opening.

used in binary morphology (opening/closing by reconstruction, area opening) [31]. For gray-level imagery, their gray-level extensions [28], [31], [4], and some more recent approaches [8], [18], [17] are frequently applied for image preprocessing due to their simplification capabilities, and for marker extraction when used in combination with the watershed algorithm in segmentation. Further extensions to image sequences and video make them useful in several color still and video processing and analysis applications [7], [22], [23].

The definition of a connected operator relates both a gray-level function and its associated partition. Obviously, the effect of any of these operations (flat zone merging and recoloring) affect both components. With the objective of formally analyzing similarities between connected operators and region merging/classification algorithms, it appears natural to extend the representation of a gray-level function to explicitly model the flat-zone partition properties. We propose such a representation in the next subsection.

B. Extended Representation of Gray-Level Functions

We start by establishing a fundamental property of the the flat-zone mapping $P_{\rm FZ}$, the key concept in defining a connected operator.

Property 2: $P_{\rm FZ}$ is neither an injection nor preserves infima and suprema.

This property establishes that a whole collection of gray-level functions correspond to the same flat-zone partition, and that in this mapping the ordering relation in ${\rm Fun}(\cdot)$ is lost. Note that $P_{\rm FZ}$ is surjective, and therefore it has right inverse mappings $P_{\rm FZ}^{\rightarrow} : \Pi \rightarrow {\rm Fun}(\cdot)$ such that, as defined in Section II, the composition $P_{\rm FZ}P_{\rm FZ}^{\rightarrow} = {\rm id}_{\Pi}$.

As we mentioned before, any connected operator can be thought of as accomplishing a double function: it both *merges* the flat zones of an image (that is essentially an operation at the partition level) and *recolors* them (that is a procedure that acts on the gray-level values of the image, according to a given rule). The specific procedures might be very different, but the underlying idea is the same: a connected operator is based on operations that depend both of properties of the function itself, and of properties of its induced flat-zone partition.

To point out the links between connected operators and segmentation algorithms, it is convenient to think of an *extended* representation of a gray-level function, as a three-tuple

$$(f, P_{\rm FZ}(f), \Theta_A(P_{\rm FZ}(f))), \quad \forall f \in {\rm Fun}(\cdot),$$

in which each component is generated from the original image f. $\Theta_A(P_{\rm FZ}(f))$ denotes, as in Section III, a partition attribute image (in this case of the flat-zone partition).

The convenience of explicitly specifying the second and third components of the three-tuple will be illustrated with three simple examples of grain binary connected operators [16]: a trivial one, an idempotent area operator, and the binary area opening. These examples are illustrated in Fig. 8.

Example 1: Let $\phi : {\rm Fun}(\mathcal{Z}^2, \{0,1\}) \rightarrow {\rm Fun}(\mathcal{Z}^2, \{0,1\})$ be the nonlinear binary operator[1] defined by $\phi(f) = 1 - f$, where $f(x) = 0$ if x is in the background, and 1 otherwise (in Fig. 8, white corresponds to $f(x) = 0$, black to $f(x) = 1$). It is evident that $P_{\rm FZ}(f) = P_{\rm FZ}(\phi(f))$ and therefore ϕ is connected. This is an operator that depends solely on the function values, in the

[1]We express these examples of binary connected operators as special cases of gray-level operators to maintain a consistent notation in the paper.

sense that it does not require of any attributes of the original flat-zone partition, for its operation.

Example 2: Consider another case to illustrate the opposite extreme. Such an operator switches the color of all those grains that have an area lower than a certain number λ. In other words, for any $f \in \mathrm{Fun}(\mathcal{Z}^2, \{0,1\})$

$$\phi(f)(x) = \begin{cases} 1 - f(x), & \text{if } \Theta_A(P_{\mathrm{FZ}}(f))(x) \leq \lambda \\ f(x), & \text{otherwise,} \end{cases}$$

where $\Theta_A(\cdot)$ is the operator that extracts the cardinality (size) of each region of $P_{\mathrm{FZ}}(f)$. It is trivial to see that ϕ is connected [16]. In this case, the action of the operator only relies on the properties of the flat-zone partition, and merges background and foreground based only on the area attribute, in the sense that the decision of changing or not the gray-level of each flat zone is independent of the gray-level itself.

Example 3: A final example is the binary area opening, that preserves all the connected components of the foreground that have an area greater or equal than λ shown in the equation at the bottom of the page.

For this example, it is clear that the operator uses both attributes of the partition (size of the region) and attributes of the gray-level function (gray-level value of the flat-zone). In fact, every connected operator that has been proposed with some utility has this general characteristic, and that is why that the extended representation can be useful to visualize the concept of connected operator, and also to understand the links with region merging segmentation algorithms.

The extended representation for gray-level images will be defined as follows.

Definition 14: Let $\mathcal{S} = \mathrm{Fun}(\cdot) \times \Pi \times \mathrm{Fun}(\cdot)$, and let s_i denote an arbitrary element of $\mathcal{S}$, with components indexed by $k, s_i = \{s_{ik}\}, k \in \{0,1,2\}$. An **extended representation generator** of a function $f \in \mathrm{Fun}(\mathcal{Z}^2)$ is a mapping $\upsilon : \mathrm{Fun}(\mathcal{Z}^2) \rightarrow \mathcal{S}$ defined by

$$\upsilon(f) = (f, P_{\mathrm{FZ}}(f), \Theta_A(P_{\mathrm{FZ}}(f))), \quad \forall f \in \mathrm{Fun}(\cdot). \tag{10}$$

From this representation, several analogies with the segmentation model presented in the previous section are evident. The mappings υ are injective with left inverses $\upsilon^{\leftarrow}$ (projections of $\mathcal{S}$ onto the first component, $\mathrm{Fun}(\cdot)$), but not surjective, with their range $\mathrm{Ran}(\upsilon)$ given only by all the three-tuples of the form specified (10), which is a proper subset of $\mathcal{S}$.

The decision about what order relation to impose on the product lattice $\mathcal{S}$ is virtually identical to the one discussed in Section III. A marginal order causes that the ordering properties in $\mathrm{Fun}(\cdot)$ are lost when expanding the gray-level representation to $\mathcal{S}$. In contrast, a conditional ordering $\leq_f$ (identical to (4), where now the ordering takes priority on f), defines $\mathcal{S}$ as a complete lattice. Additionally, the collection of all extended representation generators $\Upsilon = \{\upsilon : \mathrm{Fun}(\cdot) \rightarrow \mathcal{S}\}$ (which

$$\begin{array}{ccc} Fun(\cdot) & \xrightarrow{\phi} & Fun(\cdot) \\ {}^{\upsilon}\downarrow & & \downarrow^{\upsilon} \\ \mathcal{S} & \xrightarrow{\Phi} & \mathcal{S} \end{array}$$

Fig. 9. Schematic representation of the relations between the lattice of gray-level functions $\mathrm{Fun}(\cdot)$ and the lattice $\mathcal{S}$ that contains the set of all extended representations of gray-level functions. ϕ represents an arbitrary operator on $\mathrm{Fun}(\cdot)$, υ denotes an extended representation generator, and Φ corresponds to an operator on $\mathcal{S}$. If ϕ is connected, Φ is a extended connected operator and is denoted by Φ_C.

represent a proper subset of the complete lattice of all operators in $\mathcal{S}$, denoted by $\Delta = \{\delta : \mathrm{Fun}(\cdot) \rightarrow \mathcal{S}\}$), and their left inverses $\{\upsilon^{\leftarrow}\}$ preserve the ordering properties of $\mathrm{Fun}(\cdot)$ on $\mathcal{S}$. We summarize this result as follows.

Property 3: The **set of extended representation generators** $\Upsilon = \{\upsilon : \mathrm{Fun}(\cdot) \rightarrow \mathcal{S}\}$, and the associated left inverses, given the conditional ordering $\leq_f$, are *order-preserving*.

C. Extended Connected Operators

With a similar approach to the one used in Section III, we can define extended connected operators as a class of operators on $\mathcal{S}$. Let $\mathcal{W}$ denote the complete lattice of all operators from $\mathcal{S}$ into itself. For any arbitrary gray-level operator $\phi : \mathrm{Fun}(\cdot) \rightarrow \mathrm{Fun}(\cdot)$, we define $\Phi \in \mathcal{W}$ by

$$\Phi = \upsilon \phi \upsilon^{\leftarrow} \tag{11}$$

as shown in the diagram in Fig. 9.

The class of operators $\mathcal{K} = \{\Phi_C\} \subset \mathcal{W}$ that correspond to connected operators is defined as follows.

Definition 15: An *extended connected operator* is a mapping $\Phi_C : \mathcal{S} \rightarrow \mathcal{S}$ defined by $\Phi_C = \upsilon \phi \upsilon^{\leftarrow}$, where $\phi : \mathrm{Fun}(\cdot) \rightarrow \mathrm{Fun}(\cdot)$ is a connected operator in the lattice of gray-level functions, such that for all the three-tuples $s_i \in \mathrm{Ran}(\upsilon)$,

$$\begin{aligned} &\Phi_C(f, P_{\mathrm{FZ}}(f), \Theta_A(P_{\mathrm{FZ}}(f))) \\ &\quad = (\phi(f), P_{\mathrm{FZ}}(\phi(f)), \Theta_A(P_{\mathrm{FZ}}(\phi(f)))). \end{aligned} \tag{12}$$

The class $\mathcal{K}$ is composed of all the operators in $\mathcal{S}$ associated to connected operators in $\mathrm{Fun}(\cdot)$. Note again that this definition does not specify the operation of any $\Phi_C \in \mathcal{K}$ on those $s_i \in \mathcal{S}$ that are not three-tuples of the form $(f, P_{\mathrm{FZ}}(f), \Theta_A(P_{\mathrm{FZ}}(f)))$. Indeed, such operation can be arbitrary.

V. Links Between Connected Operators, Extensive Partition Operators and Generic Region Merging Operators

We now proceed to develop the main result in this paper regarding the links between connected operators, extensive partition operators and region merging/classification segmentation

$$\phi(f)(x) = \begin{cases} 1 - f(x), & \text{if } f(x) = 0, \quad \text{and} \quad \Theta_A(P_{\mathrm{FZ}}(f))(x) \leq \lambda \\ f(x), & \text{otherwise.} \end{cases}$$

algorithms. We start by summarizing some fundamental points from the previous development (also see Figs. 5 and 9).

1) There does not exist a one-to-one relation between gray-level functions and partitions. This holds both for segmentation models [Equation (1)], and for extended representations of gray-level functions from the connected-operator perspective (Property 2).
2) Except for a permutation, segmentation models (elements of $\mathcal{T}$) and extended representation of gray-level functions (elements of $\mathcal{S}$) are members of the same product set. However, $\mathcal{T}$ and $\mathcal{S}$ are distinct complete product lattices, not only because of the permutation, but because the ordering relation in each of them is different.
3) The order properties in Π are preserved in $\mathcal{T}$ under the priority ordering $\leq_P$. Similarly, the order properties in $\mathrm{Fun}(\cdot)$ are preserved in $\mathcal{S}$ under a different priority ordering $\leq_f$. However, regardless of the selected partial ordering, an operator in the product lattice ($\mathcal{S}$ or $\mathcal{T}$) *cannot* inherit the ordering properties of both a partition operator and a function operator.

We now formally establish the links between the different domains.

Definition 16: Let $\mathcal{T}$ and $\mathcal{S}$ the complete lattices defined as in Def. 8 and Def. 14, respectively. A **circular permutation mapping** $\alpha : \mathcal{S} \rightarrow \mathcal{T}$ is defined by

$$\alpha(s_{i1}, s_{i2}, s_{i3}) = (s_{i2}, s_{i3}, s_{i1}), \quad \forall s_i \in \mathcal{S}.$$

Note that the components of the three-tuple in $\mathcal{S}$ are shifted one position to the left. Evidently, α is a bijection, with inverse α^{-1}. This definition is the last piece we need to prove the connections between segmentation algorithms and connected operators.

Property 4: Every generic region merging operator corresponds to a connected operator. However, the converse is not true in general.

The first part of this property states that for every generic region merging operator $\Psi_R : \mathcal{T} \rightarrow \mathcal{T}$, we can build an operator in the lattice of gray-level functions $\phi : \mathrm{Fun}(\cdot) \rightarrow \mathrm{Fun}(\cdot)$ by composing

$$\phi = v^{\leftarrow}\alpha^{-1}\Psi_R\alpha v \tag{13}$$

that is connected (see diagram in Fig. 10). Equivalently, the second part establishes that, for every connected operator ϕ, we can define an extensive operator in $\mathcal{T}$ by

$$\Psi = \alpha v \phi v^{\leftarrow}\alpha^{-1}. \tag{14}$$

However, not always will Ψ represent a region merging segmentation operation of an image f as we have defined it, as we show in the following examples.

Example 4: The identity operator in the lattice of functions $\phi = \mathrm{id}_{\mathrm{Fun}(\cdot)}$ is obviously connected, and corresponds to a valid generic region merging operator $\Psi_R = (\mathrm{id}_\Pi, \mathrm{id}_{\mathrm{Fun}(\cdot)}, \mathrm{id}_{\mathrm{Fun}(\cdot)})$, for any image f.

Example 5: In contrast, the connected operator defined by $\phi(g) = g + 1, g \in \mathrm{Fun}(\cdot)$, only adds a constant to the pixel values, leaving the same flat-zone partitions for input and output images. In particular, the application of ϕ on any gray-level model image generated by an image f and a partition P_i produces $\phi(\Theta_G(f, P_i)) = \Theta_G(f, P_i) + 1$. Furthermore, the flat-zone partitions of input and output are the same when applying ϕ, i.e., $P_{\mathrm{FZ}}(\Theta_G(f, P_i)) = P_{\mathrm{FZ}}(\phi(\Theta_G(f, P_i)))$. But at the same time, $P_{\mathrm{FZ}}(\Theta_G(f, P_i)) = P_i$, and the same holds for the output partition $P_j = P_{\mathrm{FZ}}(\phi(\Theta_G(f, P_i))) = P_{\mathrm{FZ}}(\Theta_G(f, P_i) + 1) = P_i$, which implies that $\Theta_G(f, P_j) = \Theta_G(f, P_i)$. This represents a contradiction as $\phi(\Theta_G(f, P_i)) \neq \Theta_G(f, P_j)$, therefore the corresponding operator $\Psi \in \mathcal{T}$ cannot represent a generic region merging operator.

$$\begin{array}{ccc}
Fun(\cdot) & \xrightarrow{\phi} & Fun(\cdot) \\
{}^{v}\downarrow & & \uparrow^{v^{\leftarrow}} \\
\mathcal{S} & \xrightarrow{\Phi} & \mathcal{S} \\
{}^{\alpha}\downarrow & & \uparrow^{\alpha^{-1}} \\
\mathcal{T} & \xrightarrow{\Psi} & \mathcal{T} \\
{}^{(\sigma^f)^{\leftarrow}}\downarrow & & \uparrow^{\sigma^f} \\
\Pi & \xrightarrow{\psi} & \Pi
\end{array}$$

Fig. 10. Final schematic representation of the operations between lattices of partitions and gray-level functions. If Ψ_R is a generic region merging operator, the corresponding $\phi = v^{\leftarrow}\alpha^{-1}\Psi_R\alpha v$ is connected. However. if ϕ is connected, then $\Psi = \alpha v \phi v^{\leftarrow}\alpha^{-1}$ is extensive but not always a region merging operator.

Furthermore, as every extensive operator in Π corresponds to an operator in $\mathcal{T}$, we can proceed in a similar way to conclude the following.

Property 5: Every connected operator corresponds to an extensive operator in the lattice of partitions.

Finally, with the above analysis we can also express the flat-zone partition operator P_{FZ} and their right inverses by

$$P_{\mathrm{FZ}} = (\sigma^f)^{\leftarrow}\alpha v, \quad P_{\mathrm{FZ}}^{\rightarrow} = v^{\leftarrow}\alpha^{-1}\sigma^f. \tag{15}$$

Recapitulating, we have shown that the lattice-theoretic approach proposed in this paper allows for an algebraic formulation of the connection between region merging/classification algorithms and connected operators, namely, that every generic region merging operator corresponds to a connected operator, and that the converse situation is not true in general. This result provides a basis to formally explain results previously proposed, and for the specification of new algorithms. We revisit in the next section some of the properties of a general region merging algorithm, under the complete lattice perspective. The analysis of some region classification operators and their morphological properties are reported in [9].

VI. Analysis of Generic Region Merging Operators

In this section, we analyze the general model for region merging proposed in [8]. Let $\Psi_{\mathrm{Rm}} \in \mathcal{R}$ denote a generic region merging operator that merges only one region of a partition P

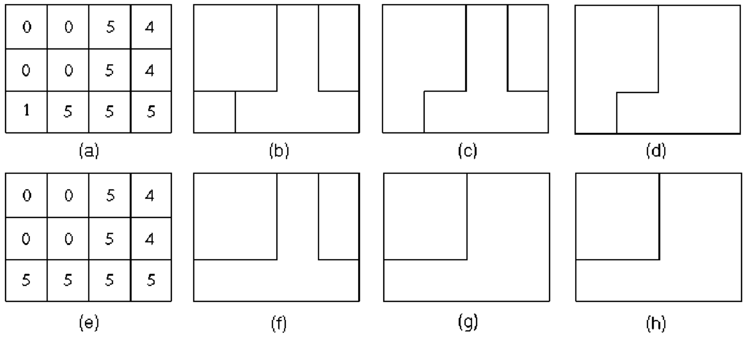

Fig. 11. Counterexample to show that basic region merging operators are usually neither increasing nor idempotent (a), (e) Gray-level images f_1 and f_2. (b), (f) Associated partitions in the segmentation model, $P_{\rm FZ}(f_1), P_{\rm FZ}(f_2)$. (c), (g) Associated partitions $\psi_m(P_{\rm FZ}(f_1)), \psi_m(P_{\rm FZ}(f_2))$. These partitions are not comparable, so both ψ_m, and $\Psi_{\rm Rm}$ are not increasing. (d), (h) By reapplying the operators, idempotence fails to hold by comparing (c) and (d).

with one of its neighboring regions, and ψ_m denote the corresponding operator in Π, given by $\psi_m = \sigma^f \Psi_{\rm Rm}(\sigma^f)^{\leftarrow}$. Let $\{\Psi_{\rm Rm}\}$ denote the set of all such operators. As we established in Section III, a general region merging method needs three elements for its specification [8]: the region model, that has been already explicitly decoupled and included in the lattice $\mathcal{T}$ by Θ_A and Θ_G, a merging order, and a merging criterion. The latter includes the definition of a termination criteria TC for the merging process, typically represented by either the number of zones in the partition or a measure of error between the original image and a reconstructed image based on the partition and its zone models.

Let $P(x_{c1})$ and $P(x_{c2})$ denote the two zones specified as candidates by the merging order. In addition, let $[\rm TC]$ represents the Boolean value of the statement TC (1 if it is true, 0 otherwise). With this notation, a general formulation for the partition operator ψ_m can be specified as follows:

$$\psi_m(P)(x) = \begin{cases} P(x_{c1}) \cup P(x_{c2}), & [TC] = 0 \quad \text{and} \\ & x \in P(x_{c1}) \cup P(x_{c2}) \\ P(x), & \text{otherwise} \end{cases} \tag{16}$$

and the corresponding expression for the generic region merging operator is

$$\Psi_{\rm Rm}(P) = (\psi_m(P), \Theta_A(\psi_m(P)), \Theta_G(f, \psi_m(P))) \tag{17}$$

or in a pseudocode-like description

```
if ([TC] = 0)
{merge  P(x_c1) and P(x_c2), and
  update the model of P(x_c1) ∪ P(x_c2)}.
```

A first result can be established.

Property 6: Every $\Psi_{\rm Rm}$ is extensive. However, in general they are neither increasing nor idempotent.

Fig. 11 illustrates this statement.

We can now define a complete segmentation algorithm based on region merging.

Definition 17: Let f and P_0 be an image and an initial partition (usually equal to the finest partition), respectively, and let t_0 denote the segmentation model generated by f and P_0. A **complete region merging segmentation algorithm** is an operator $\Psi_{\rm RM}$ that consists on the iterative composition of $\Psi_{\rm Rm}$ until convergence

$$\Psi_{\rm RM}(t_0) = \lim_{n\to\infty} \Psi^n_{\rm Rm}(t_0) \triangleq \Psi^\infty_{\rm Rm}(t_0) \tag{18}$$

where $\Psi^n_{\rm Rm} = \overbrace{\Psi_{\rm Rm}\Psi_{\rm Rm}\dots\Psi_{\rm Rm}}^{n}$, and $\Psi^\infty_{\rm Rm}$ is the limit of the sequence $\Psi_{\rm Rm}, \Psi^2_{\rm Rm}, \Psi^3_{\rm Rm}, \dots$. In fact, convergence of this sequence is always assured in a finite number of steps K, so

$$\Psi_{\rm RM}(t_0) = \Psi^K_{\rm Rm}(t_0) \tag{19}$$

because $\Psi^K_{\rm Rm} = \Psi^\infty_{\rm Rm}$. The criteria for convergence corresponds to the termination criterion TC in $\Psi_{\rm Rm}$.

Let $\{\Psi_{\rm RM}\}$ denote the class of region merging operators. The fact that any $\Psi_{\rm RM}$ is extensive is obvious. One more result can be specified.

Property 7: Every $\Psi_{\rm RM}$ is idempotent, and its invariance domain ${\rm Inv}(\Psi_{\rm RM}) = {\rm Inv}(\Psi_{\rm Rm})$.

However, the operators $\{\Psi_{\rm RM}\}$ are not increasing in general. This can be seen using the previous example, in which Fig. 11(d) and (h) coincide with the application of $\Psi_{\rm RM}$ until convergence, and in which the associated partitions are not comparable by the partial order relation. The fact that the operators $\{\Psi_{\rm RM}\}$ are generally not order-preserving has important consequences, as the general theory of mathematical morphology mainly deals with order-preserving mappings. Additionally, the elements in $\{\Psi_{\rm RM}\}$ are generally not injective. If the two last conditions hold, the next result is obvious.

Property 8: If a generic region merging operator $\Psi_{\rm RM}$ is neither order-preserving nor injective, then it is neither of the following: 1) invertible, 2) lattice isomorphism, and 3) morphological filter.

It is important to point out that the theoretical description of segmentation models and extended function representations presented in this paper do not constitute the most efficient data structure for actual algorithm implementation. In practice, a segmentation model (partitions and region models) can be efficiently represented by a region adjacency graph. Additionally, an efficient structure for storing the sequence of steps in a region merging operator is a *binary partition tree* [24]. Indeed, when region merging is performed, beginning from the flat-zone partition of the original image and until convergence, the binary par-

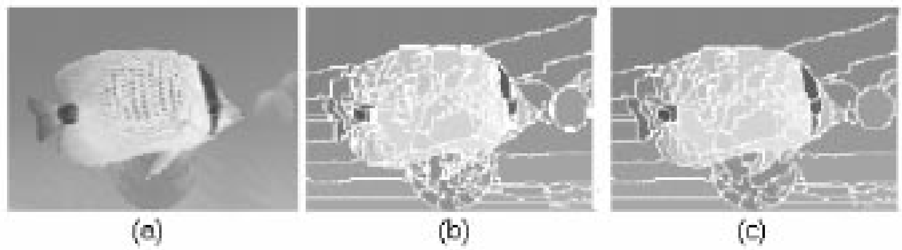

Fig. 12. Complete region merging segmentation algorithm Ψ_{RM}. (a) Original image f. Two different steps in the merging sequence, $\Psi^{k}_{\mathrm{Rm}}(t_0)$ (b), and $\Psi^{l}_{\mathrm{Rm}}(t_0), k < l$ (c), for which the partitions have been superimposed to the gray-level model image. A binary partition tree (not shown) is used to keep track of the merging sequence.

tition tree represents a way of storing each of the outcomes of the sequence of operators $\Psi_{\mathrm{Rm}}(t_0), \Psi^2_{\mathrm{Rm}}(t_0), \ldots$, also called *merging sequence*. Due to the algorithmic equivalence with connected operators, tree representations have also demonstrated its usefulness for connected operators [1], [16], [17], [24]. An example of such a sequence is shown in Fig. 12. The reader may refer to those references for implementation details.

VII. Concluding Remarks

This paper addressed a question pointed out several times in recent mathematical morphology literature: the connection between region merging/classification segmentation algorithms and gray-level connected operators. We have proposed that such a connection can be formally obtained by defining a complete product lattice representation for both gray-level images and segmentation models, that allows for the explicit modeling of partition and intensity attributes. We have shown that every region merging/classification segmentation algorithm corresponds to a connected operator. This result explains the algorithmic equivalence in terms of implementation, that has been recently exploited by several authors. Additionally, we have also provided an analysis of some important theoretical properties of a general region merging algorithms, from the proposed lattice-theoretic framework.

It is important to note that, although both region merging and region classification algorithms are linked to connected operators, the latter class does not make explicit use of the usual 4- or 8-connectivity. There has been an increasing study of other connectivities [29]. We are interested in studying the connectivity properties of region classification algorithms. Additionally, given the connections analyzed in this paper, we are also interested in the development of novel strategies for image segmentation and filtering.

Appendix

A. Glossary of Symbols

See the glossary of symbols at the bottom of the next page.

B. Proofs of the Properties

Property 2: P_{FZ} is neither an injection nor preserves infima and suprema.

Proof: The first part is obvious. The gray-level functions f_1 and $f_2 = f_1 + 1$ induce the same partition $P_{\mathrm{FZ}} \in \Pi$. The second part means that $P_{\mathrm{FZ}}(f_1 \wedge f_2) = P_{\mathrm{FZ}}(f_1) \wedge P_{\mathrm{FZ}}(f_2), \forall f_1, f_2 \in \mathrm{Fun}(\mathcal{Z}^2)$, and dually for $\vee$. However, this is not the case, as it can be shown with another counterexample. Let $E = \{0,1\}^2$, and f_1 and f_2 be the following 2×2 gray-level functions,

$$f_1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \quad f_2 = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \quad f_1 \wedge f_2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Clearly $P_{\mathrm{FZ}}(f_1) = P_{\mathrm{FZ}}(f_2) = P_{\mathrm{FZ}}(f_1) \wedge P_{\mathrm{FZ}}(f_2)$ but this is different from $P_{\mathrm{FZ}}(f_1 \wedge f_2) = \vee \Pi$, therefore P_{FZ} does not preserve infima. A similar example can be used to show that P_{FZ} does not preserve suprema. □

Property 4: Every generic region merging operator corresponds to a connected operator. However, the converse is not true in general.

Proof: For the first part, let $\Psi_R : \mathcal{T} \rightarrow \mathcal{T}$ be a generic region merging operator, so that for any $t_i \in \mathcal{T}$ of the form $(P_i, \Theta_A(P_i), \Theta_G(f, P_i))$, there exists a $t_j \in \mathcal{T}$ of the form $(P_j, \Theta_A(P_j), \Theta_G(f, P_j))$ (for some f) such that $t_j = \Psi_R(t_i)$. We construct an operator $\phi : \mathrm{Fun}(\cdot) \rightarrow \mathrm{Fun}(\cdot)$ by composing

$$\phi = \upsilon^{\leftarrow} \alpha^{-1} \Psi_R \alpha \upsilon.$$

The diagram in Fig. 10 illustrates such definition. Now, let $f_i \in \mathrm{Fun}(\cdot)$, and $P_i = P_{\mathrm{FZ}}(f_i)$. Then

$$\begin{aligned}
\phi(f_i) &= \upsilon^{\leftarrow} \alpha^{-1} \Psi_R \alpha \upsilon(f_i) \\
&= \upsilon^{\leftarrow} \alpha^{-1} \Psi_R(P_{\mathrm{FZ}}(f_i), \Theta_A(P_{\mathrm{FZ}}(f_i)), f_i) \\
&= \upsilon^{\leftarrow} \alpha^{-1} \Psi_R(P_i, \Theta_A(P_i), \Theta_G(f, P_i)), \quad \text{for some } f \\
&= \upsilon^{\leftarrow} \alpha^{-1} (P_j, \Theta_A(P_j), \Theta_G(f, P_j)) \\
&\triangleq \upsilon^{\leftarrow} \alpha^{-1} (P_j, \Theta_A(P_j), f_j) \\
&= \upsilon^{\leftarrow} \alpha^{-1} (P_{\mathrm{FZ}}(f_j), \Theta_A(P_{\mathrm{FZ}}(f_j)), f_j) \\
&\triangleq f_j.
\end{aligned}$$

But by hypothesis, Ψ_R is a generic region merging operator, i.e., $P_i \leq P_j$, which means that $P_{\mathrm{FZ}}(f_i) \leq P_{\mathrm{FZ}}(\phi(f_i))$. Therefore, the operator ϕ is connected.This proves the first part of the proposition.

For the second part, assume that ϕ is connected. We can obtain an expression for an operator in $\mathcal{T}$ as

$$\Psi = \alpha \upsilon \phi \upsilon^{\leftarrow} \alpha^{-1}.$$

Note that we cannot not write the subscript R in Ψ yet, because we do not know at this point if it represents a generic region merging operator. Now, let $t_i \in \mathcal{T}$ be of the form $(P_i, \Theta_A(P_i), \Theta_G(f, P_i))$ for some f. Therefore,

$$\begin{aligned}\Psi(t_i) &= \Psi(P_i, \Theta_A(P_i), \Theta_G(f, P_i)) \\ &= \alpha\upsilon\phi\upsilon^{\leftarrow}\alpha^{-1}(P_i, \Theta_A(P_i), \Theta_G(f, P_i)) \\ &= \alpha\upsilon\phi(\Theta_G(f, P_i)) \\ &\triangleq \alpha\upsilon(f_j) \\ &= (P_{\mathrm{FZ}}(f_j), \Theta_A(P_{\mathrm{FZ}}(f_j)), f_j) \\ &\triangleq t_j.\end{aligned}$$

Symbol	Meaning	Reference
$\mathcal{L}$	complete lattice	Def. 1
$\mathrm{id}_{\mathcal{L}}$	identity operator on $\mathcal{L}$	Def. 2
E	arbitrary nonempty set	Def. 3
x	element of E	Def. 3
f	gray-level image	Def. 3
$\mathrm{Fun}(\mathcal{Z}^2)$	complete lattice of gray-level images	Def. 3
$\mathcal{P}(E)$	powerset of E	Def. 4
P	partition	Def. 4
$P(x)$	zone or region that contains x	Def. 4
Π	complete lattice of partitions	Def. 4
ψ	partition lattice operator	Def. 5
Θ_A	attribute region model operator	Def. 6
$\Theta_A(P)$	partition attribute image	Def. 6
Θ_G	gray-level region model operator	Def. 6
$\Theta_G(f, P)$	gray-level model image	Def. 6
Q_{Θ_G}	gray-level model equivalence relation	Def. 7
$f_{i_{\Theta_G}}$	i-th fixpoint of Θ_G	Def. 7
σ^f	segmentation model generator	Def. 8
$\sigma^f(P)$	segmentation model	Def. 8
$\mathcal{T}$	complete lattice that includes all segmentation models	Def. 8
t	element of $\mathcal{T}$ (three-tuple)	Def. 8
Γ	complete lattice of mappings between Π and $\mathcal{T}$	Section III-A
Σ	set of segmentation model generators	Prop. 1
$\mathcal{U}$	complete lattice of mappings on $\mathcal{S}$	Section III-B
Ψ	operator on $\mathcal{T}$	Eq. (6)
Ψ_R	generic region merging operator	Def. 9
$\mathcal{R}$	set of generic region merging operators	Def. 9
C_x^Y	grain of $Y \subset \mathcal{Z}^2$ that contains $x \in \mathcal{Z}^2$	Def. 11
Y_f^t	level set	Def. 11
$P_{\mathrm{FZ}}(f)$	flat-zone partition of f	Def. 12
$P_{\mathrm{FZ}}(f)(x)$	flat zone	Def. 12
ϕ	connected operator	Def. 13
υ	extended representation generator	Def. 14
$\upsilon(f)$	gray-level extended representation of f	Def. 14
$\mathcal{S}$	complete lattice that includes all gray-level extended representations	Def. 14
s	element of $\mathcal{S}$ (three-tuple)	Def. 14
Δ	complete lattice of mappings between $\mathrm{Fun}(\cdot)$ and $\mathcal{S}$	Section IV-B
Υ	set of extended representation generators	Prop. 3
$\mathcal{W}$	complete lattice of mappings on $\mathcal{T}$	Section IV-C
Φ	operator on $\mathcal{S}$	Eq. (11)
Φ_C	extended connected operator	Def. 15
$\mathcal{K}$	set of extended connected operators	Def. 15
α	circular permutation mapping	Def. 16
Ψ_{Rm}	region merging operator that merge two regions	Section VI
ψ_m	partition operator that merge two regions	Section VI
$P(x_{\mathrm{ci}})$	candidate regions for merging	Section VI
Ψ_{RM}	complete region merging segmentation algorithm	Def. 17

But by hypothesis, ϕ is connected, i.e., $P_{\mathrm{FZ}}(\Theta_G(f, P_i)) \leq P_{\mathrm{FZ}}(\phi(\Theta_G(f, P_j)))$. Furthermore, as $P_{\mathrm{FZ}}(\Theta_G(f, P_i)) = P_i$ and $P_{\mathrm{FZ}}(\phi(\Theta_G(f, P_j))) = P_j$, it follows that Ψ is extensive. However, only when $f_j = \phi(\Theta_G(f, P_i))$ is equal to the (unique) valid $\Theta_G(f, P_j)$ (see Fig. 6(h)–(j)), will Ψ correspond to a generic region merging operator. Otherwise, Eq. (20) is still valid, but Ψ does not represent a region merging segmentation operation of an image f as we define it. □

Property 6: Every Ψ_{Rm} is extensive. However, in general they are neither increasing nor idempotent.

Proof: The first property follows by definition. The second one can be proved with a typical counterexample. Assume the median gray-level of the zone as region model, $[\mathrm{TC}] = [\text{number of regions} \geq 2], P(x_{c1})$ as the zone of smallest cardinality in P, and $P(x_{c2})$ as the neighboring region of $P(x_{c1})$ that has the most similar region model. Now assume f_1, f_2, the two 3×4 images as shown in Fig. 11(a) and (e), and define their segmentation models in $\mathcal{T}$ as $(P_{\mathrm{FZ}}(f_i), \Theta_A(P_{\mathrm{FZ}}(f_i)), \Theta_A(f_i, P_{\mathrm{FZ}}(f_i)))$. It is clear, as shown in Fig. 11(b) and (f), that $P_{\mathrm{FZ}}(f_1) < P_{\mathrm{FZ}}(f_2)$. However, $\psi_m(P_{\mathrm{FZ}}(f_1))$ and $\psi_m(P_{\mathrm{FZ}}(f_1))$, shown in Fig. 11(c) and (g), can not be compared in Π, and neither can $\Psi_{\mathrm{Rm}}(t_i)$ and $\Psi_{\mathrm{Rm}}(t_j)$ in $\mathcal{T}$. Therefore, ψ_m and Ψ_{Rm} are not increasing. It is also evident that $\psi_m(P_{\mathrm{FZ}}(f_1)) < \psi_m(\psi_m(P_{\mathrm{FZ}}(f_1)))$ (Fig. 11(d)); therefore, ψ_m and Ψ_{Rm} are not idempotent. □

Property 7: Every Ψ_{RM} is idempotent, and its invariance domain $\mathrm{Inv}(\Psi_{\mathrm{RM}}) = \mathrm{Inv}(\Psi_{\mathrm{Rm}})$.

Proof: To prove this, we just need to show that $\Psi_{\mathrm{Rm}}^n \to \Psi_{\mathrm{Rm}}^\infty$ and $\Psi_{\mathrm{Rm}}\Psi_{\mathrm{Rm}}^\infty = \Psi_{\mathrm{Rm}}^\infty$, and apply a known result ([15], pp. 452). The first condition is true by hypothesis. For the second one, we know that Ψ_{Rm} only takes an action when the termination criterion is false. But Ψ_{RM} is precisely defined by the iterative merging until the termination criterion is true (after k steps), so $\Psi_{\mathrm{Rm}}\Psi_{\mathrm{Rm}}^\infty = \Psi_{\mathrm{Rm}}^\infty$. □

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their effort to improve the quality of this paper. They are also thankful to Prof. Philippe Salembier for his comments.

REFERENCES

[1] E. J. Breen and R. Jones, "Attribute openings, thinnings, and granulometries," *Comput. Vis. Image Understand.*, vol. 64, no. 3, pp. 377–389, Nov. 1996.

[2] R. Castagno, T. Ebrahimi, and M. Kunt, "Video segmentation based on multiple features for interactive multimedia applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 562–571, Sept. 1998.

[3] E. Chalom and V. M. Bove, "Segmentation of an image sequence using multi-dimensional image attributes," *Proc. IEEE Int. Conf. Image Processing*, pp. 643–647, Oct. 1996.

[4] J. Crespo, J. Serra, and R. W. Schafer, "Theoretical aspects of morphological filters by reconstruction," *Signal Process.*, vol. 47, no. 2, pp. 201–225, 1995.

[5] J. Crespo, R. W. Schafer, J. Serra, C. Gratin, and F. Meyer, "The flat zone approach: A general low-level region merging segmentation method," *Signal Process.*, vol. 62, no. 1, pp. 37–60, 1997.

[6] B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[7] L. Garrido, A. Oliveras, and P. Salembier, "Motion analysis of image sequences using connected operators," in *Proc. SPIE Visual Communications Image Processing (VCIP'97)*, vol. 3024, San Jose, CA, Feb. 1997, pp. 546–557.

[8] L. Garrido, P. Salembier, and D. Garcia, "Extensive operators in partition lattices for image sequence analysis," *Signal Process.*, vol. 66, pp. 157–180, Apr. 1998.

[9] D. Gatica-Perez, M. T. Sun, and C. Gu, "Partition lattice operators for extraction of semantic video objects," in *Proc. Int. Symp. Mathematical Morphology 2000*, Palo Alto, CA, June 2000.

[10] J. Goutsias, H. J. A. M. Heijmans, and K. Sivakumar, "Morphological operators for image sequences," *Comput. Vis. Image Understand.*, vol. 62, no. 3, pp. 326–346, 1995.

[11] C. Gu and M.-C. Lee, "Semantic video object tracking using region-based classification," *Proc. IEEE Int. Conf. Image Processing*, pp. 643–647, Oct. 1998.

[12] R. Haralick and L. Shapiro, *Computer and Robot Vision*. Reading, MA: Addison-Wesley, 1992, vol. I.

[13] K. Haris, S. Efstratiadis, N. Maglaveras, and A. K. Katsaggelos, "A hybrid image segmentation algorithm using watersheds and hierarchical region merging," *IEEE Trans. Image Processing*, vol. 7, pp. 1684–1699, Dec. 1998.

[14] H. J. A. M. Heijmans and J. Serra, "Convergence, continuity and iteration in mathematical morphology," *J. Vis. Commun. Image Represent.*, vol. 3, no. 1, pp. 84–104, Mar. 1992.

[15] H. J. A. M. Heijmans, *Morphological Image Operators*. New York: Academic, 1994.

[16] ——, "Connected morphological operator for binary images," CWI, Amsterdam, The Netherlands, Report PNA-R9708, Apr. 1997.

[17] R. Jones, "Connected filtering and segmentation using component trees," *Comput. Vis. Image Understand.*, vol. 75, no. 3, pp. 215–228, Sept. 1999.

[18] F. Meyer, "From connected operators to levelings," in *Proc. ISMM '98*, 1998, pp. 191–198.

[19] S. MacLane and G. Birkhoff, *Algebra*, 3rd ed. New York: AMS Chelsea, 1999.

[20] L. Najman and M. Schmitt, "Geodesic saliency of watershed contours and hierarchical segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 1163–1173, Dec. 1996.

[21] P. Salembier and J. Serra, "Flat zones filtering, connected operators, and filters by reconstruction," *IEEE Trans. Image Processing*, vol. 4, pp. 1153–1160, Aug. 1995.

[22] P. Salembier, P. Brigger, J. R. Casas, and M. Pardas, "Morphological operators for image and video compression," *IEEE Trans. Image Processing*, vol. 5, pp. 881–898, June 1996.

[23] P. Salembier, A. Oliveras, and L. Garrido, "Anti-extensive connected operators for image and sequence processing," *IEEE Trans. Image Processing*, vol. 7, pp. 555–570, Apr. 1998.

[24] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for filtering, segmentation and information retrieval," in *Proc. IEEE Int. Conf. Image Processing (ICIP'98)*, Chicago, IL, 1998.

[25] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic, 1982.

[26] ——, *Image Analysis and Mathematical Morphology, Vol. II: Theoretical Advances*. New York: Academic, 1988.

[27] J. Serra, "Anamorphoses and function lattices (multivalued morphology)," in *Mathematical Morphology in Image Processing*, E. R. Dougherty, Ed., 1993, pp. 483–523.

[28] J. Serra and P. Salembier, "Connected operators and pyramids," in *Proc. SPIE Image Algebra Morphological Image Processing IV*, vol. 2030, San Diego, CA, 1993, pp. 65–76.

[29] J. Serra, "Connectivity on complete lattices," *J. Math. Imag. Vis.*, vol. 9, pp. 231–251, 1998.

[30] J. C. Simon, *Patterns and Operators: The Foundations of Data Representation*. Oxford, U.K.: North Oxford, 1986.

[31] L. Vincent, "Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms," *IEEE Trans. Image Processing*, vol. 2, pp. 176–201, Apr. 1993.

[32] ——, "Grayscale area openings and closings, their efficient implementation and applications," in *Mathematical Morphology and Its Applications to Signal Processing*, J. Serra and P. Soille, Eds. New York: UPC, May 1993, pp. 22–27.

[33] D. Wang, "A multiscale gradient algorithm for image segmentation using watersheds," *Pattern Recognit.*, vol. 30, no. 12, 1997.

Daniel Gatica-Perez (S'01) received the B.S. degree in electronic engineering from the University of Puebla, Mexico, in 1993, and the M.S. degree in electrical engineering from the National University of Mexico, Mexico City, in 1996. He is currently pursuing the Ph.D. degree in electrical engineering at the University of Washington, Seattle.

His research interests include video analysis, machine learning, and mathematical morphology.

Chuang Gu (M'97) received the B.S. and M.S. degrees in computer science from Fudan University, Shanghai, China, in 1986 and 1989, respectively. He received the Ph.D. degree in electrical engineering from Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1995. His Ph.D. dissertation was on multivalued morphology, segmentation-based coding, and tracking.

From 1989 to 1990, he was an Assistant Director at the CAD Center of Fudan University. He was an Associate Research Fellow at the European Organization for High Energy Physics (CERN) in 1991. From 1992 to 1995, he was with Signal Processing Laboratory, EPFL, as a Research Assistant. Since 1995, he has been with Microsoft Corporation, Redmond, WA, carrying out research and development on video processing, analysis and coding for Windows Media Technology (http://www.microsoft.com/windows/windowsmedia/). He has published more than 40 papers and has six patents on these subjects. Since 1999, he has been an Affiliate Professor at the University of Washington.

Dr. Gu received the Young Investigator Award from SPIE Visual Communication and Image Processing in 1999. He is an associate editor of the IEEE TRANSACTIONS ON MULTIMEDIA.

Ming-Ting Sun (S'79–M'81–SM'89–F'96) received the B.S. degree from National Taiwan University, Taipei, Taiwan, R.O.C., in 1976, the M.S. degree from University of Texas at Arlington in 1981, and the Ph.D. degree from University of California, Los Angeles in 1985, all in electrical engineering.

He joined the University of Washington, Seattle, in August 1996, where he is now a Professor. Previously, he was the Director of the Video Signal Processing Research Group at Bellcore. He has seven patents and has published more than 100 technical papers including ten book chapters in the area of video technology.

Dr. Sun is Editor-in-Chief of the IEEE TRANSACTIONS ON MULTIMEDIA. He was a conference general co-chair of Visual Communications and Image Processing 2000. He received a Golden Jubilee Medal from the IEEE Circuits and Systems Society in 2000. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS for Video Technology (TCSVT) from 1995 to 1997. He was a co-recipient of the TCSVT Best Paper Award in 1993. From 1988 to 1991, he served as the Chairman of the IEEE Circuits and Systems Standards Committee and established an IEEE Inverse Discrete Cosine Transform Standard. He received an Award of Excellence from Bellcore in 1987 for the work on Digital Subscriber Line.

Salvador Ruiz-Correa received the B.S. and M.S. degrees in electrical engineering from the National University of Mexico, Mexico City, in 1990 and 1994, respectively. He is currently pursuing the Ph.D. degree in electrical engineering at the University of Washington, Seattle.

His research interests include computer vision and machine learning.