Toward the Automatic Generation of Mathematical Morphology Procedures Using Predicate Logic

*Hyonam Joo, Robert M. Haralick, and Linda G. Shapiro**
Intelligence Systems Laboratory
Department of Electrical Engineering, FT-10
University of Washington
Seattle, WA 98195

ABSTRACT

This paper discusses the design of a system that can input a vision task specification and use its knowledge of the operations of mathematical morphology to automatically construct a procedure that can execute the task. To do this, we develop a predicate calculus representation to describe the essence of the states of all the images that are created during the execution of the morphological procedure and the states of the relationships among them. We translate the English descriptions of morphological procedures into predicate logic. In so doing we gain an understanding of the goal of each procedure and the exact conditions under which a procedure achieves its goal. With this knowledge of the operations of mathematical morphology represented in predicate logic, a search procedure can be used to automatically produce vision procedures. The search begins with the desired image described in predicate logic. If the existence of the desired image can be proved, then a morphological processing sequence to produce the desired image exists. Hence to automatically determine the morphological vision procedure, a search can be set up to constructively prove a theorem in predicate logic about the existence of the desired image. In this way, each step of the proof is associated with a morphological operation. The execution of the morphological operation in the order given in the proof then constitutes the desired morphological processing sequence.

1. Introduction

Mathematical morphology has been widely used in machine vision since it was introduced by Serra and Sternberg [Serra 82, Sternberg 86]. Many useful image processing procedures have been developed using basic operators of mathematical morphology. Vision algorithm developers have spent a lot of time finding meaningful sequences of basic morphological operators which can execute the vision tasks they are trying to solve. When a vision expert is given a task, he or she first analyzes the task, makes a plan, produces a procedure, tries it, evaluates the test results, and refines or updates the procedure. In doing so, the expert uses knowledge of the given problem domain and knowledge of the available morphological operations to determine a reasonable vision procedure. Their effort has been successful in producing many morphological procedures in a variety of application fields [Joo and Haralick, 89]. It is natural for us to ask now how a human vision expert can develop the morphological procedures that solve the vision tasks. Is it possible to create a system that can mimic what a vision expert does? The goal of our research is to answer this question by designing a system that can input a vision task specification and use its knowledge of the operations of mathematical morphology to automatically construct a procedure that can execute the vision task. In this paper, we establish the theoretical groundwork with which to pursue our research toward the solution of this problem.

In a related philosophical paper, Serra [Serra 86] discusses a definitive irreversible operation and poses a question regarding how one can spread out successive losses among a series made up of dozens of morphological transformations, so that the series of operations can produce the desired result. In selecting the series of transformations, he points out that we need to have a set of reference properties and perform the assessment with respect to these properties. Since there is a series of operations, he advises us to apply the first criteria to the choice of the comparison between inputs and outputs, and to direct our interest to the mappings which may have special concepts such as extensive or anti-extensive, over- or underpotent, idempotent, and isotropic. In addition to this, he suggests several rules for organizing a morphological processing sequence.

A morphological algorithm consists of a sequence of basic morphological operators and each morphological operator requires some parameters which are commonly called structuring elements. Gillies, working on the line of searching for the right parameters, proposed a learning system that uses genetic search to generate feature

CH2934-8/90/0000/0156$01.00 © 1990 IEEE

detectors which cooperate in the classification of image samples [Gillies 85]. In his hybrid human/machine adaptive system, the user provides the logical components of the feature detector, namely the sequence of morphological operators, and the machine provides the spatial component, namely the structuring elements. The system generates structuring elements and tests the resultant feature detectors using images from the training set. The results of the test are used to direct the search for new structuring elements which will lead to better feature detectors. The entire adaptive process constitutes a search in which breadth and depth are balanced according to the observed performance of feature detectors evaluated so far.

Research on finding a best vision algorithm among a small set of predetermined sequences of operators can be seen in several papers [Hasegawa *et al* 86, Ikeuchi and Kanade 88, Sakaue and Tamura 85, Goad 83, Matsuyama 88]. Most of the systems proposed in this category are designed to present the basic framework for the automatic generation of vision algorithms. Even with their reported success, their algorithm search space is limited, the description of the image is not quite extensive, and their use of knowledge in the selection of operators and the parameters is limited. However, the idea of inferencing the expected image in [Hasegawa *et al* 86] and the use of an algorithm graph in [Matsuyama 88] are interesting.

A search for both the best morphological sequence of operators and optimal parameters for each operator can be found in the work of Vogt [Vogt 88]. Vogt implemented a system called REM, which is able to generate some vision algorithms for binary images. In REM, the basic morphological operators are categorized in terms of both their form and their properties. Given accept and reject masks, the system finds a single algorithm which distinguishes the corresponding accept pixels from the reject pixels. A blackboard-based control strategy improves the efficiency of the search. REM currently solves one- or two-step problems that have perfect solutions, for a relatively small number of different band-pass operators. However, the search performed by REM is guided by the false positive and false negative error rate that the generated algorithm achieves, not by the intrinsic morphological properties of the objects to be detected. The categorization of basic morphological operators is quite coarse; the knowledge of what each operator achieves is not well represented in REM.

We believe that automatic procedure generation requires a careful, concrete formalization of the mechanism of the algorithm development cycle performed by human vision experts. We attempt to do this by uncovering and explicitly representing the knowledge that vision experts possess and use. We show how morphological knowledge and vision tasks can be represented in a predicate calculus form and how an algorithm can be created by a reasoning process which can be implemented in the form of a predicate logic theorem prover.

2. Notation and basic definitions

Let $\mathbf{E}$ be set of integers. Let upper case letters $A, B, C, \cdots$ be finite subsets of $\mathbf{E}^2$ and let boldface lower case letters $\mathbf{a}, \mathbf{b}, \mathbf{c}, \cdots$ denote elements of $\mathbf{E}^2$, which represent the pixel locations in an image.

The following are the definitions for the basic binary morphological operators used in this paper. For any set $A \subseteq \mathbf{E}^2$ and $\mathbf{x} \in \mathbf{E}^2$, let $A_{\mathbf{x}}$ denote the *translation* of A by $\mathbf{x}$ which is defined by

$$A_{\mathbf{x}} = \{\mathbf{y} \mid \text{for some } \mathbf{a} \in A, \mathbf{y} = \mathbf{a} + \mathbf{x}\}.$$

For any set $A \in \mathbf{E}^2$, let $\check{A}$ denote the *reflection* of A about the origin which is defined by

$$\check{A} = \{\mathbf{x} \mid \text{ for some } \mathbf{a} \in A, \mathbf{x} = -\mathbf{a}\}.$$

The *dilation* of a set A by a set B is denoted by $A \oplus B$ and is defined by

$$A \oplus B = \{\mathbf{x} \mid \text{for some } \mathbf{a} \in A \text{ and } \mathbf{b} \in B, \mathbf{x} = \mathbf{a} + \mathbf{b}\}.$$

The *erosion* of a set A by a set B is denoted by $A \ominus B$ and is defined by

$$A \ominus B = \{\mathbf{x} \mid \text{for every } \mathbf{b} \in B, \mathbf{x} + \mathbf{b} \in A\}.$$

Let J and K be two sets satisfying $J \cap K = \phi$, then the *hit and miss transform* of a set A by a pair of sets (J, K) is denoted by $A \circledast (J, K)$ and is defined by

$$A \circledast (J, K) = (A \ominus J) \cap (A^c \ominus K).$$

where A^c is the complementation of the set A with respect to E^2. The *opening* of a set A by a set B is denoted by $A \circ B$ and is defined by

$$A \circ B = (A \ominus B) \oplus B.$$

The *closing* of a set A by a set B is denoted by $A \bullet B$ and is defined by

$$A \bullet B = (A \oplus B) \ominus B.$$

There are many relationships among these basic morphological operators. For a complete discussion on this topic, readers are referred to [Haralick *et al* 87].

One of the important properties of a set is its connectedness. The following are the definitions for a distance function and a binary relation required to define the eight-way connectivity relations used in our discussion.

Definition: The chessboard distance $\rho_c(A, B)$ between two non-empty sets A and B of $\mathbf{E}^2$ is defined by

$$\rho_c(A, B) = \min_{(x,y) \in A} \min_{(x',y') \in B} \max\{|x - x'|, |y - y'|\}$$

The chessboard distance between an empty set and any other set is defined to be infinity.

Definition: The separation relation Γ_s is the binary relation defined by

$$\Gamma_s = \{(A,B) \mid \rho_c(A,B) > 1\}$$

All the connectivity relations discussed in the following sections are defined with respect to the separation relation Γ_s [Haralick and Shapiro, 90].

Definition: Two sets A and B are *separated* or *not connected to each other* if $(A,B) \in \Gamma_s$. If $(A,B) \notin \Gamma_s$, then A and B are *connected to each other.*

Definition: A set A is called *connected* iff every 2-celled partition $\pi = \{\pi_1, \pi_2\}$ of A satisfying $\pi_1 \neq \phi$ and $\pi_2 \neq \phi$, also satisfies $(\pi_1, \pi_2) \notin \Gamma_s$.

According to the above definitions, the empty set is connected but is separated from all other sets. The separation relation Γ_s has the following four properties that are used in some of the proofs of the theorems in Section 4 [Haralick and Shapiro, 90].

(1) Symmetric: $(X,Y) \in \Gamma_s$ implies $(Y,X) \in \Gamma$.
(2) Exclusive: $(X,Y) \in \Gamma_s$ implies $X \cap Y = \phi$.
(3) Hereditary: $(X,Y) \in \Gamma_s$ implies $(X',Y') \in \Gamma_s$ for each $X' \subseteq X$ and $Y' \subseteq Y$.
(4) Extensive: $(\{\mathbf{x}\},\{\mathbf{y}\}) \in \Gamma_s$ for every $\mathbf{x} \in X$ and $\mathbf{y} \in Y$ implies $(X,Y) \in \Gamma_s$.

Having defined the notation needed to describe the binary morphological algorithms, we start discussing our representation scheme.

3. Representation Scheme

Given a morphological algorithm and a set of input images used in the algorithm, vision experts can explain what the algorithm achieves in plain English. If there is a difficulty in describing it, they can also point to the regions in an image that should be detected by the algorithm. They can describe what each step of the algorithm does and why each step is necessary. However, for a machine to perform like a vision expert, the knowledge used in the algorithm development cycle must be made explicit. In this section we discuss a representation scheme that allows us to precisely state the facts about mathematical morphology. As logic is the study of reasoning and because of the simplicity and expressive power of mathematical logic, we choose to design our representation scheme in a first-order predicate logic setting.

We begin by defining the symbols of our representation. We have already defined the symbols used for subsets and elements of $\mathbf{E}^2$. We add to that list the symbols for binary images and scalar constants. Let calligraphic capital letters $\mathcal{A}, \mathcal{B}, \cdots, \mathcal{I}, \mathcal{J}, \cdots$ represent binary images. We define the binary image as follows:

Definition: A binary image $\mathcal{I}$ is a tuple $\mathcal{I} = (D,F,B)$ where $D \subseteq \mathbf{E}^2$ is the domain of the image $\mathcal{I}$, $F \subseteq D$ is the set of its binary 1 pixels (foreground pixels), and $B \subseteq D$ is the set of its border pixels. The border pixels are the pixels in D which have, among their eight neighboring pixels, at least one pixel not belonging to D.

The structuring elements used in morphological operators are finite non-empty subsets of $\mathbf{E}^2$ and are written in upper case letters. To represent scalar quantities such as the radius of a disk structuring element, the length of a line, etc, we use lower case letters which are elements of $\mathbf{E}$.

Throughout this section, we omit the explicit typing of terms used in logic expressions for reasons of simplicity. The types of terms should be clear by the choice of characters used to name them. When explicit type checking is necessary, we will use the set of predicates, "image", "2D-set", "structuring-element", "integer", and "2D-element".

Functions are used to represent both the image operators and the structuring elements used in morphological operators. Each image operator, whether it is a morphological operator or a set operator, returns an image which is the output image produced by the operator. The names of functions are written in italics to distinguish them from predicates. For example, the image complement operator and the morphological erosion operator are defined as follows.

Definition: *I_complement*($\mathcal{I}$). The complement image $\mathcal{J}$ of a binary image $\mathcal{I} = (D,F,B)$ is defined by $\mathcal{J} = (D, D-F, B)$ and we write $\mathcal{J} = \textit{I\_complement}(\mathcal{I})$.

Definition: *B_erode*($\mathcal{I}, S$). The morphological erosion image $\mathcal{J}$ of a binary image $\mathcal{I} = (D,F,B)$ by a structuring element S is defined by $\mathcal{J} = (D, (F \ominus S) \cap D, B)$ and we write $\mathcal{J} = \textit{B\_erode}(\mathcal{I}, S)$.

The structuring elements of special geometric shapes are defined in terms of functions returning the subsets of $\mathbf{E}^2$ corresponding to their shapes. For example, disks of positive integer radius and boxes (rectangles) of positive integer widths and heights are defined as follows:

$$\textit{disk}(r) = \{(x,y) \in \mathbf{E}^2 \mid (x^2 + y^2) < (r+0.5)^2\}$$
$$\textit{box}(w,h) = \{(x,y) \in \mathbf{E}^2 \mid -(w-1)/2 \leq x \leq w/2, \; -(h-1)/2 \leq y \leq h/2\}$$

We use predicates to represent concepts and relations satisfied among terms in our representation. For example the following three predicates can be used to test if a set $A \subseteq \mathbf{E}^2$ is one of the three components of the binary image tuple.

- I-domain($A, \mathcal{I}$) is true iff the set A is the domain of the image $\mathcal{I}$.
- I-foreground($A, \mathcal{I}$) is true iff the set A is the set of all the binary 1 pixels of the image $\mathcal{I}$.
- I-border($A, \mathcal{I}$) is true iff the set A is the set of all the border pixels of the image $\mathcal{I}$.

Likewise, we use a predicate "connected-to(A, B)" to test if two sets A and B are connected to each other and a predicate "connected(A)" to test if a set A is a connected set.

The following are some of the simple predicates used to describe properties of sets in $\mathbf{E}^2$ and relationships among them.

Definition: inImage($A,\mathcal{I}$). A set A is *in image* $I = (D, F, B)$ iff $A \subseteq F$.

Definition: subset(A, B). A set A is a *subset* of B iff $A \subseteq B$.

Definition: set-equal(A, B). A set A is *equal* to a set B iff $A = B$.

Definition: empty(A). A set A is an *empty* set iff $A = \phi$.

Definition: set-diff(A, B, C). A set A is a *set-difference* of a set B and a set C iff $A = B - C$.

If an important property or a relationship corresponds to a rather complicated expression in terms of simple predicates, it may be desirable to introduce a new predicate to abbreviate the expression. It will make certain expressions shorter or more readable. An example of such a property is maximal-connectedness of a set in a binary image, which can be defined as follows:

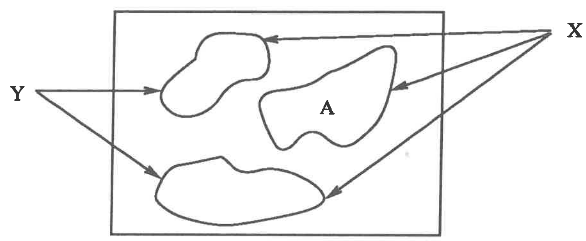


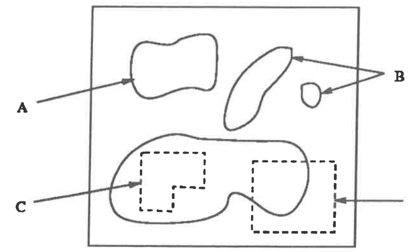
Figure 3–1 Illustration of the predicate "max-connected($A,\mathcal{I}$)".

Definition: max-connected($A,\mathcal{I}$). A set A is *maximally-connected* in image $\mathcal{I} = (D, F, B)$ iff A is a connected set in image $\mathcal{I}$, and A and $(F - A)$ are separated.

With the simple predicates defined so far, we can define the predicate "max-connected" more precisely as follows:

$$\forall A, \forall \mathcal{I}, \left(\begin{array}{l} \text{max-connected}(A,\mathcal{I}) \Leftrightarrow \\ \left(\begin{array}{l} \text{connected}(A), \text{inImage}(A,\mathcal{I}),\ \forall X, \forall Y, \\ \left(\begin{array}{l} (\text{I-foreground}(X,\mathcal{I}), \text{set-diff}(Y, X, A)) \\ \quad \Rightarrow \neg\text{connected-to}(Y, A) \end{array} \right) \end{array} \right) \end{array} \right)$$

Figure 3–1 illustrates the above logic statement. Illustration of some of the properties defined in this section are shown in figure 3–2.



inImage($A,\mathcal{I}$)	inImage($B,\mathcal{I}$)
inImage($C,\mathcal{I}$)	¬inImage($D,\mathcal{I}$)
connected(A)	¬connected(B)
connected(C)	connected(D)
max-connected($A,\mathcal{I}$)	¬max-connected($B,\mathcal{I}$)
¬max-connected($C,\mathcal{I}$)	¬max-connected($D,\mathcal{I}$)

Figure 3–2 Illustration of sets in a binary image and the predicates they satisfy.

There are two main kinds of knowledge: (1) knowledge about the state of an image or relationships among several images and (2) knowledge about the effects of both set–theoretic and morphological operators. Both kinds of knowledge are represented in terms of rules of inference. For example, it is obvious from the definition that a set A being maximally connected in an image $\mathcal{I}$ implies that the set A is in the image $\mathcal{I}$. We represent this rule by the following predicate logic statement.

$$\forall A, \forall \mathcal{I}, (\text{max-connected}(A,\mathcal{I}) \Rightarrow \text{inImage}(A,\mathcal{I}))$$

On the other hand, if a set A is maximally connected in image $\mathcal{I}$, it is not in the complemented image of $\mathcal{I}$. This fact can be represented by the following statement.

$$\forall A, \forall \mathcal{I}, \left(\begin{array}{l} \text{max-connected}(A,\mathcal{I}) \\ \quad \Rightarrow \neg\text{inImage}(A, \textit{I\_complement}(\mathcal{I})) \end{array} \right)$$

These rules of inference will enable us to conclude the existence of an algorithm to achieve the desired task.

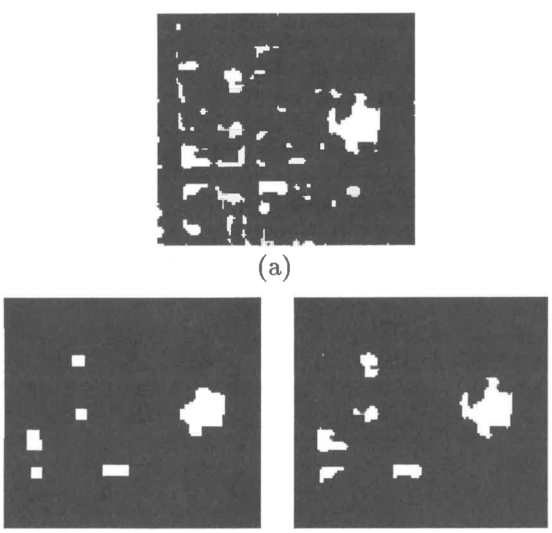
In order to better explain the representation scheme being used to describe binary morphological algorithms, we give a simple example algorithm in the next section and show how it can be represented and created in our representation.

4. Algorithm Creation

In this section, we analyze an example morphological algorithm precisely in terms of the representation scheme developed in the previous section and show how such an algorithm can be created. We describe the essence of the states of all the images that are being created during the execution of the algorithm and the states of the relationships among them. The description needs to be complete enough to explain the whole process of the algorithm which achieves the required vision task, i.e., the goal of the algorithm. To do this we must give predicate calculus definitions for concepts such as morphological opening, conditional dilation, disjoint sets, one set being larger than another, and connectedness. Then we must develop theorems concerning some relationships between connectedness and opening, connectedness and dilation, separation and dilation, maximally connected sets and conditional dilation. These definitions and theorems are on the one hand very technical. On the other hand they uncover and reveal the association between a variety of spatial and topological concepts and some operations of mathematical morphology. Although none of what we do is interesting in and of itself mathematically, it does constitute the knowledge base of a reasoning system for the automatic construction of vision procedures. From this point of view it may be interesting and justify the effort we have put in to develop it.

We begin by a statement of the goal of the vision procedure needed in the algorithm design process for the determination of a vision procedure and then give all the definitions and theorems. We conclude the section with the algorithm design.

Let us suppose that we want to develop an algorithm that removes small objects that cannot contain a rectangle 5 pixels high and 5 pixels wide in a binary image. The objects are sets of foreground pixels in the image. We can achieve this vision task by first marking the larger objects using a binary opening operator, and then recovering the original maximally connected objects using a conditional dilation operator. Figure 4–1 illustrates this algorithm. We begin our explanation by defining the two morphological operators, binary opening (*B_open*) and conditional dilation (*cdil*).



(a) (b) (c)

Figure 4–1 Illustration of the example morphological algorithm which removes small objects in an image. (a) shows the original binary image $\mathcal{I}_0$; (b) shows the result $\mathcal{I}_a$ of the opening of the original image by a rectangle 5 pixels high and 5 pixels wide; (c) shows the result of the conditional dilation of $\mathcal{I}_a$ with respect to $\mathcal{I}_0$.

Definition: $B\_open(\mathcal{I}, S)$. A morphologically opened image $\mathcal{J}$ of a binary image $\mathcal{I} = (D, F, B)$ by a structuring element S is defined by $\mathcal{J} = (D, F \circ S, B)$ and we write $\mathcal{J} = B\_open(\mathcal{I}, S)$.

Definition: $cdil(\mathcal{J}, \mathcal{I})$. A conditionally dilated image $\mathcal{K}$ of $\mathcal{J} = (D, G, B)$ with respect to $\mathcal{I} = (D, F, B)$ by a rectangle of width 3 and height 3 pixels is defined by $\mathcal{K} = (D, H^m, B)$ where the index m is the smallest index satisfying $H^m = H^{m-1}$ when $H^0 = G$ and $H^n = (H^{n-1} \oplus box(3,3)) \cap F$ and we write $\mathcal{K} = cdil(\mathcal{J}, \mathcal{I})$. Figure 4–2 illustrates this operator.

Now, we define a set of predicates required to describe the example morphological algorithm. A set of simple, easily proved rules are also stated without proof. Each rule is preceded by the definition of the predicate to which the rule is related.

Definition: disjoint(A, B). A set A is *disjoint* from B iff $A \cap B = \phi$.

The following logic statement states that if a set is not disjoint from a non-empty set, it is also a non-empty set.

$$\forall A, \forall B, ((\neg \text{empty}(A), \neg \text{disjoint}(A, B)) \Rightarrow \neg \text{empty}(B))$$

Definition: isopened-by(A, B, S). Let S be a structuring element and A and B be sets. Then isopened-by(A, B, S) iff $A = B \circ S$.

The following three logic statements state that (1) for any set there exists its opened set, (2) the opened set is unique, (3) $(A \circ S) \subseteq A$, for any structuring element S; this is one of the basic relationships satisfied by the morphological opening operator.

$$\forall B, \forall S, (\exists A, \text{isopened-by}(A, B, S))$$

$$\forall A, \forall B, \forall C, \forall S, \left(\begin{pmatrix} \text{isopened-by}(A, C, S), \\ \text{isopened-by}(B, C, S) \end{pmatrix} \Rightarrow \text{equal}(A, B) \right)$$

$$\forall A, \forall B, \forall S, (\text{isopened-by}(A, B, S) \Rightarrow \text{subset}(A, B))$$

Definition: larger(A, S). Let S be a structuring element. A set A is *larger* than S iff $A \circ S \neq \phi$.

The following logic statement characterizes this predicate.

$$\forall A, \forall S, \begin{pmatrix} \text{larger}(A, S) \Leftrightarrow \exists X, \\ (\text{isopened-by}(X, A, S), \neg \text{empty}(X)) \end{pmatrix}$$

The following are the set of rules of inference needed to generate the example algorithm. These rules constitute the information data base of our system. They are stated as theorems and the proofs are provided. Most rules encode the information regarding the morphological operators which vision experts utilize when designing algorithms. The first two theorems describe the morphological opening operator. Theorem 4–1 is used to show that if a set is maximally connected in an input image $\mathcal{I}_0$ and is larger than a structuring element S, its opening is non–empty and is in the image $\mathcal{I}_0$ opened by S. Theorem 4–2 is needed to show that the marker of a large object A found by the opening operation must intersect with $A \circ S$.

Theorem 4–1: If A is a set in image $\mathcal{I}$, then its opening by a structuring element S is in image $B\_open(\mathcal{I}, S)$. It can be stated as the following logic statement.

$$\forall A, \forall \mathcal{I},$$
$$\left(\begin{array}{l} \text{inImage}(A, \mathcal{I}) \Rightarrow \forall S, \forall X, (\text{isopened-by}(X, A, S) \\ \qquad \Rightarrow \text{inImage}(X, B\_open(\mathcal{I}, S))) \end{array} \right)$$

Proof: Let $\mathcal{I}$ be (D, F, B), then $B\_open(\mathcal{I}, S) = (D, F \circ S, B)$. Let $A \subseteq F$, then $(A \circ S) \subseteq (F \circ S)$, which implies that $(A \circ S)$ is in image $B\_open(\mathcal{I}, S)$. Since $(A \circ S)$ is unique, the second logic statement is also true. ∎

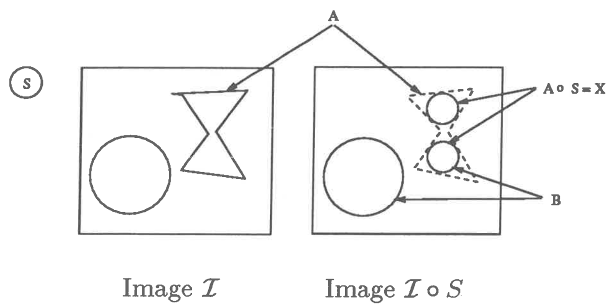


Figure 4–3 Illustration of theorem 4–2.

Theorem 4–2: If S is a connected structuring element, A is a maximally connected set in $\mathcal{I}$, and B is a set in image $B\_open(\mathcal{I}, S)$ such that A and B are connected to each other, then $B \cap (A \circ S) \neq \phi$. In predicate logic, we write the following.

$$\forall A, \forall B, \forall S, \forall \mathcal{I},$$
$$\left(\begin{array}{l} \left(\begin{array}{l} \text{connected}(S), \text{inImage}(B, B\_open(\mathcal{I}, S)), \\ \text{max-connected}(A, \mathcal{I}), \text{connected-to}(B, A) \end{array} \right) \\ \Rightarrow \exists X, (\text{isopened-by}(X, A, S), \neg\text{disjoint}(B, X)) \end{array} \right)$$

An illustration of this theorem is given in figure 4–3.

Proof: Let $\mathcal{I} = (D, F, B)$ and $\mathcal{J} = (D, G, B)$ where $G = F \circ S$. Since A is a maximally connected set in $\mathcal{I}$, A and $(F - A)$ are separated. Therefore, $G = F \circ S = (A \circ S) \cup ((F - A) \circ S)$. Since $((F - A) \circ S) \subseteq (F - A)$, $(A, (F - A)) \in \Gamma_s$, which implies $(A, ((F - A) \circ S)) \in \Gamma_s$. Thus, A is not connected to $((F - A) \circ S)$. Let a non-empty set $X \subseteq G$ and A be connected to each other and $X \cap (A \circ S) = \phi$. Then $X \subseteq ((F - A) \circ S)$ or $(A \circ S) = \phi$. If $(A \circ S) = \phi$, then $X \subseteq G$ implies $X \subseteq ((F - A) \circ S)$. If $X \subseteq ((F - A) \circ S)$, X and A are separated, which is a contradiction. ∎

The following are a set of rules related to the connectivity property. We need them to prove theorem 4–5, which encodes an important property of the morphological conditional dilation operator.

Theorem 4–3: A set A and a set B are connected to each other iff there exists a subset X of A such that X and B are connected to each other. Figure 4–4 illustrates this theorem.

$$\forall A, \forall B, \left(\begin{array}{l} \text{connected-to}(A, B) \Leftrightarrow \exists X, \\ (\text{subset}(X, A), \text{connected-to}(X, B)) \end{array} \right)$$

Proof: If A and B are connected to each other, there must exist $(x_A, y_A) \in A$ and $(x_B, y_B) \in B$ such that $\max\{|x_A - x_B|, |y_A - y_B|\} \leq 1$. Therefore, $\rho_c(\{(x_A, y_A)\}, B) \leq 1$, which implies that $\{(x_A, y_A)\} \subseteq A$ and B are connected to each other. If there exists a subset X of A such that X and B are connected to each other, $\rho_c(X, B) \leq 1$. But,

$$\begin{aligned} \rho_c(A, B) &= \min_{(x,y) \in A} \min_{(x',y') \in B} \max\{|x - x'|, |y - y'|\} \\ &\leq \min_{(x,y) \in X} \min_{(x',y') \in B} \max\{|x - x'|, |y - y'|\} \\ &\leq 1 \end{aligned}$$

Therefore, A and B are connected to each other. ∎

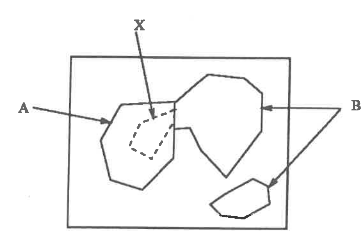


Figure 4–4 Illustration of theorem 4–3.

Theorem 4–4: A set A and a set B are connected to each other iff $(A \oplus box(3,3)) \cap B \neq \phi$.

Proof: Since an empty set is separated from any subset of $\mathbf{E}^2$, $A \neq \phi$ and $B \neq \phi$. Suppose $A \cap B \neq \phi$. Since $(0,0) \in box(3,3)$ implies $A \subseteq (A \oplus box(3,3))$, $(A \oplus box(3,3)) \cap B \neq \phi$. Suppose $A \cap B = \phi$ but $(A \oplus box(3,3)) \cap B = \phi$. Since A and B are connected to each other, there must exist $(x_A, y_A) \in A$ and $(x_B, y_B) \in B$ such that $\max\{|x_A - x_B|, |y_A - y_B|\} \leq 1$. Hence, $x_B \in \{x_A - 1, x_A, x_A + 1\}$ and $y_B \in \{y_A - 1, y_A, y_A + 1\}$, which implies that $(x_B, y_B) \in [\{(x_A, y_A)\} \oplus box(3,3)]$. But, $(x_A, y_A) \in A$ implies that $(x_B, y_B) \in (A \oplus box(3,3))$. Since $(x_B, y_B) \in B$, $(x_B, y_B) \in [(A \oplus box(3,3)) \cap B]$, which is a contradiction to $(A \oplus box(3,3)) \cap B = \phi$. Therefore, $(A \oplus box(3,3)) \cap B \neq \phi$.

Suppose that $(A \oplus box(3,3)) \cap B \neq \phi$. Then, there must exist $(x, y) \in (A \oplus box(3,3))$ and $(x, y) \in B$. Suppose $(x, y) \in A$, then $(x, y) \in B$ implies $\rho_c(A, B) = 0$. Suppose $(x, y) \in (A \oplus box(3,3) - A)$, then there must exist $(x_A, y_A) \in A$ such that $x \in \{x_A - 1, x_A + 1\}$ and $y \in \{y_A - 1, y_A + 1\}$. Therefore, $\rho_c(A, B) = \max\{|x_A - x|, |y_A - y|\} = 1$. Hence, A and B are connected to each other. ∎

Corollary 4–1: A set A and a set B are separated iff $(A \oplus box(3,3)) \cap B = \phi$.

The following theorem encodes the information on the conditional dilation operator, which states that if an image $\mathcal{J}$ is conditionally dilated by a $box(3,3)$ structuring element with respect to an image $\mathcal{I}$, the output image consists of all the maximally connected objects in image $\mathcal{I}$ that are connected to the foreground pixels of the image $\mathcal{J}$. The proof of this theorem is given in [Joo, Haralick, and Shapiro, 90].

Theorem 4–5: A set A is a maximally connected set in image $\mathcal{I}$ and there exists a set B in image $\mathcal{J}$ such that A and B are connected to each other if and only if A is a non-empty maximally connected set in the image obtained by $\mathcal{J}$ conditionally dilated by a $box(3,3)$ with respect to $\mathcal{I}$.

$$\forall A, \forall \mathcal{I}, \forall \mathcal{J}, \left(\begin{pmatrix} \text{max-connected}(A, \mathcal{I}), \exists X, \\ (\text{inImage}(X, \mathcal{J}), \text{connected-to}(X, A)) \end{pmatrix} \Leftrightarrow \\ (\neg\text{empty}(A), \text{max-connected}(A, \mathit{cdil}(\mathcal{J}, \mathcal{I}))) \right)$$

The following is a set of additional rules of inference related to the predicates defined so far and required to prove the existence of an algorithm for this example. Each one of them can be easily proved using the theorems of set theory and mathematical morphology and the proofs are omitted here.

- A set is non-empty iff there exists a non-empty subset of it.

$$\forall A, (\neg\text{empty}(A) \Leftrightarrow \exists X, (\text{subset}(X, A), \neg\text{empty}(X)))$$

- The connectivity relation is symmetric.

$$\forall A, \forall B, (\text{connected-to}(A, B) \Leftrightarrow \text{connected-to}(B, A))$$

- If $B \neq \phi$ and $B \subseteq A$, then B and A are connected to each other.

$$\forall A, \forall B, \left(\begin{pmatrix} \neg\text{empty}(B), \\ \text{subset}(B, A) \end{pmatrix} \Rightarrow \text{connected-to}(B, A) \right)$$

- If a set A and a set B are connected to each other, $A \neq \phi$ and $B \neq \phi$.

$$\forall A, \forall B, \left(\text{connected-to}(A, B) \Rightarrow \begin{pmatrix} \neg\text{empty}(A), \\ \neg\text{empty}(B) \end{pmatrix} \right)$$

We have defined all the predicates and functions and have stated rules of inference sufficient to be able to describe the example algorithm. We will show in the next subsection how the example algorithm can be designed with all the information explicitly stated so far.

Algorithm design

We first define two constants, one for the input image and the other for the box shaped structuring element.

- $\mathcal{I}_0$: input binary image.
- $B_5 = box(5,5)$: 5×5 box.

Since a box of 5 pixels high and 5 pixels wide is a connected set, we have connected(B_5) as one of the initially satisfied logic statements. We want to express the vision task of this example algorithm in our representation. When an algorithm executes a vision task successfully, it should produce an output image (the goal image) which satisfies the vision task. Thus, by proving the existence of the goal image, we should be able to construct an algorithm. The vision task of this example algorithm can be stated more precisely as follows: Prove that there exists an image $\mathcal{G}$ where all the maximally connected sets in image $\mathcal{I}_0$ larger than the structuring element $box(5,5)$ are also maximally connected sets in $\mathcal{G}$, and every non-empty maximally connected set in $\mathcal{G}$ is also a maximally connected set in $\mathcal{I}_0$ and is larger than the structuring element $box(5,5)$. We can represent this vision task by the following logic statement, called the *goal statement.*

$$\exists \mathcal{G}, \forall A, \left[\begin{array}{l} (\text{max-connected}(A, \mathcal{I}_0), \text{larger}(A, B_5)) \\ \Leftrightarrow (\neg\text{empty}(A), \text{max-connected}(A, \mathcal{G})) \end{array} \right]$$

We will prove the goal statement using the logic statements initially satisfied and the rules of inference in the information data base. We approach them in a constructive manner. To prove that a goal image with a certain property exists, we will actually *construct* such an image, not merely show that the nonexistence of such an image would lead to a contradiction. Note that each image operator is defined as a function which returns an image. Since an algorithm is a sequence of image operators, it is a concatenation of functions in our representation and the output image returned by this concatenation of functions is the output image produced by the algorithm. Thus, by constructively proving the existence of the goal image we can create an algorithm which achieves the vision task. We will do so for this example in such a way that when the goal statement is proved, the goal image $\mathcal{G}$ should exist and should be unified with the concatenation of two functions, "$cdil(B\_open(\mathcal{I}_0, B_5), \mathcal{I}_0)$".

We first prove the only if, ($\Rightarrow$), part of the goal statement and construct the goal image. To prove this, we arbitrarily choose a set which satisfies the left hand side of "$\Rightarrow$" and then prove that it also satisfies the right hand side. Assume that a set A is given which is a maximally connected set in image $\mathcal{I}_0$ and is larger than the structuring element $box(5,5)$. Then, larger(A, B_5) implies $\exists X$, (isopened-by(X, A, B_5), $\neg$empty(X)). Now let B be a set satisfying (isopened-by(B, A, B_5), $\neg$empty(B)), then

$$\text{isopened-by}(B, A, B_5) \Rightarrow \text{subset}(B, A),$$

$$\begin{pmatrix} \text{subset}(B, A), \\ \neg\text{empty}(B) \end{pmatrix} \Rightarrow \begin{pmatrix} \text{connected-to}(B, A), \\ \neg\text{empty}(A) \end{pmatrix}.$$

The set A being maximally connected in image $\mathcal{I}_0$ implies inImage$(A, \mathcal{I}_0)$, which implies, by theorem 4–1,

$$\forall X, \begin{pmatrix} \text{isopened-by}(X, A, B_5) \\ \Rightarrow \text{inImage}(X, B\_open(\mathcal{I}_0, B_5)) \end{pmatrix}.$$

Since isopened-by(B, A, B_5) is true, inImage$(B, B_open(\mathcal{I}_0, B_5))$ must be true. Hence:

$$\exists B, \begin{pmatrix} \text{connected-to}(B, A), \\ \text{inImage}(B, B\_open(\mathcal{I}_0, B_5)) \end{pmatrix}$$

By theorem 4–5 and the fact that A is a maximally connected set in image $\mathcal{G} = cdil(B\_open(\mathcal{I}_0, B_5), \mathcal{I}_0)$, we have proved that there exists an image $\mathcal{G}$ which satisfies the ($\Rightarrow$) part of the goal statement.

To prove the if, ($\Leftarrow$), part of the goal statement, let $\mathcal{G}$ be $cdil(B\_open(\mathcal{I}_0, B_5), \mathcal{I}_0)$ and a non-empty set A be given which satisfies max-connected$(A, \mathcal{G})$. Then, by theorem 4–5,

$$\begin{pmatrix} \text{max-connected}(A, \mathcal{I}_0), \exists X, \\ (\text{connected-to}(X, A), \text{inImage}(X, B\_open(\mathcal{I}_0, B_5))) \end{pmatrix}.$$

Let B be a set satisfying

$$(\text{connected-to}(B, A), \text{inImage}(B, B\_open(\mathcal{I}_0, B_5)))\,;$$

then $B \neq \phi$. Since connected(B_5) is true, by theorem 4–2, if A and B are sets satisfying the following

$$\begin{pmatrix} \text{max-connected}(A, \mathcal{I}_0), \\ (\text{connected-to}(B, A), \text{inImage}(B, B\_open(\mathcal{I}_0, B_5))) \end{pmatrix},$$

then

$$\exists X, (\text{isopened-by}(X, A, B_5), \neg\text{disjoint}(B, X))\,.$$

Let C be the set satisfying the above logic statement. Since $C \cap B \neq \phi$ and B is not empty, we have $\neg$empty(C). Hence, we have proved that

$$\exists C, (\text{isopened-by}(C, A, B_5), \neg\text{empty}(C))\,,$$

which is equivalent to larger(A, B_5). Thus, we have proved the ($\Leftarrow$) part of the goal statement. Therefore, we have proved the goal statement and have created the desired algorithm to produce the goal image $\mathcal{G}$. The goal image is given by

$$\mathcal{G} = cdil(B\_open(\mathcal{I}_0, B_5), \mathcal{I}_0)$$

and can be obtained by first applying an opening to image $\mathcal{I}_0$ and then conditionally dilating the result.

5. More Examples

We succesfully used our representation scheme to analyze the process of creating a set of morphological algorithms. In this section, we list without proof two such examples. We will state the goal statement and the solution which can come out of the theorem proving. We will also give the definitions of the predicates and the operators appearing in the goal statement and the solution.

5.1 Particle marking

This algorithm finds cells with round shaped and large nucleus from images taken through microscope. In this example we have a pair of input images. $\mathcal{I}_0$ represents an image of cytoplasms and nuclei while $\mathcal{J}_0$ represents an image of nuclei only. We need definitions for the following five concepts: sub-body, round, *B_close*, *B_close_res*, and *I_diff*.

sub-body$(C, A, \mathcal{I})$:

A set C is a ***sub-body*** of A in Image $\mathcal{I} = (D, F, B)$ iff $C = A \cap F$.

round(A, K, L):

A set A is ***round*** with respect to a structuring element pair (K, L) iff $((A \bullet K) - A) \circ L = \phi$.

$B\_close(\mathcal{I}, S)$:

A morphologically closed image $\mathcal{J}$ of a binary image $\mathcal{I} = (D, F, B)$ by a structuring element S is defined by $\mathcal{J} = (D, (F \bullet S) \cap D, B)$.

$B\_close\_res(\mathcal{I}, S)$:

A morphologically closed residue image $\mathcal{J}$ of a binary image $\mathcal{I} = (D, F, B)$ by a structuring element S is defined by $\mathcal{J} = (D, ((F \bullet S) \cap D) - F, B)$.

$I\_diff(\mathcal{I}, \mathcal{J})$:

Let the domains of two images $\mathcal{I}$ and $\mathcal{J}$ are same. Then, a difference image $\mathcal{K}$ of the two binary images $\mathcal{I} = (D, F, B)$ and $\mathcal{J} = (D, G, B)$ is defined by $\mathcal{K} = (D, F - G, B)$.

Goal Statement:

$\exists \mathcal{G}, \forall A,$

$$\left(\begin{pmatrix} \text{max-connected}(A, \mathcal{I}_0), \\ \exists X, \begin{pmatrix} \text{subset}(X, A), \text{inImage}(X, \mathcal{J}_0), \\ \text{larger}(X, D_{10}) \end{pmatrix}, \\ \forall Y, (\text{sub-body}(Y, A, \mathcal{J}_0) \Rightarrow \text{round}(Y, D_5, D_2)) \end{pmatrix} \\ \Leftrightarrow (\neg\text{empty}(A), \text{max-connected}(A, \mathcal{G}))\right)$$

Solution:

$$\mathcal{G} = I\_diff\begin{pmatrix} B\_open(B\_close\_res(\mathcal{J}_0, D_5), D_2), \\ cdil(B\_open(\mathcal{J}_0, D_{10}), \mathcal{I}_0) \end{pmatrix}$$

5.2 Finding missing gear tooth space

This algorithm finds missing or broken teeth spaces of watch gears. We need a bunch of definitions.

transof(A, B):

A set A is a *translation* of a set B iff $A = B_{\mathbf{x}}$ for some $\mathbf{x} \in \mathbf{E} \times \mathbf{E}$.

holeof$(H, S, \mathcal{I})$:

A set H is a *hole of* $S \subseteq F$ in image $\mathcal{I} = (D, F, B)$ iff H is a maximally connected set of $D - S$ in image $(D, D - S, B)$ and is disjoint from B.

isa-hole$(H, \mathcal{I})$:

A set H *is a hole* of image $\mathcal{I} = (D, F, B)$ iff H is a hole of F in image $\mathcal{I}$.

isolated-hole$(H, S, \mathcal{I})$:

Let S be a structuring element. A set H is an *isolated hole* with respect to S in image $\mathcal{I}$ iff it is a hole and for some translation $\mathbf{x} \in \mathbf{E} \times \mathbf{E}$, $H \subseteq S_{\mathbf{x}}$, and the translated structuring element $S_{\mathbf{x}}$ is separated from all the other holes of image $\mathcal{I}$ disjoint from $S_{\mathbf{x}}$.

c-union(A, Ω):

The *collection union* A of a collection of sets Ω is the union of all sets in Ω and can be characterized by: $\mathbf{a} \in A \Rightarrow \exists C \in \Omega, \mathbf{a} \in C$, and $C \in \Omega \Rightarrow C \subseteq A$.

filled-hull$(Z, S, \mathcal{I})$:

The *filled-hull* Z of a set S in image $\mathcal{I}$ is the union of S with the collection union of the collection of all holes of S in image $\mathcal{I}$.

trans-collection$(\Omega, S, \mathcal{I})$:

Let S be a structuring element. The *trans-collection* Ω of a structuring element S in image $\mathcal{I} = (D, F, B)$ is the collection of all sets $S_{\mathbf{x}} \subseteq F$ for some translation $\mathbf{x} \in \mathbf{E} \times \mathbf{E}$.

trans-body$(A, S, \mathcal{I})$:

If Ω is a trans-collection of a structuring element S in image $\mathcal{I}$, a set A is a *trans-body* of a structuring element S in image $\mathcal{I}$ iff A is the collection union of Ω.

isopen-under(A, S):

A set A is open under a structuring element S iff $A = A \circ S$.

shape-body$(A, S, \mathcal{I})$:

Let S be a structuring element. A set A is a *shape-body* of S in image $\mathcal{I} = (D, F, B)$ iff A is the maximal subset of F whose filled-hull is open under S.

inDomain$(A, \mathcal{I})$:

A set A is *in the domain* of an image $\mathcal{I} = (D, F, B)$ iff $A \subseteq D$.

dist-farther(A, d, B):

Let d be a positive integer. A set A is *distance farther* than d pixels from a set B iff $(B \oplus \text{disk}(d)) \cap A = \phi$.

dist-closer(A, d, B):

Let d be a positive integer. A set A is *distance closer* than d pixels from a set B iff $(B \oplus \text{disk}(d)) \cap A = A$.

dist-between(A, a, b, B):

Let a and b be positive integers satisfying $a < b$. A set A is *distance between* a and b pixels from a set B iff A is distance farther than a pixels from B and A is distance closer than b pixels from B.

dist-outer-between$(A, a, b, B, \mathcal{I})$:

Let a and b be positive integers satisfying $a < b$. A set A is *distance between* a and b pixels from the *outer* boundary of a set B in image $\mathcal{I}$ iff A is between a and b pixels from the filled-hull of B in image $\mathcal{I}$.

gear-body$(B, r, \mathcal{I})$:

Let r be a positive integer. Let Σ be the set of all application images in this example. A set B is a *gear body* of size r of the image $\mathcal{I} \in \Sigma$ iff B is the shape-body of a disk of radius r in image $\mathcal{I}$.

gear-tooth$(T, a, b, c, \mathcal{I})$:

Let a and b be positive integers satisfying $a < b$ and c be a positive integer. Let Σ be the set of all application images in this example. A set T is a *gear tooth* of the image $\mathcal{I} \in \Sigma$ with parameter a, b, and c iff it is in $\mathcal{I}$ and it is distance between a and b pixels from the outer boundary of the gear body of size c in $\mathcal{I}$.

is-missing-tooth-space$(A, a, b, c, d, \mathcal{I})$:

Let a, b, c, and d be positive integers and $a < b$. Let Σ be the set of all application images in this example. A set A is a *missing tooth space* of the image $\mathcal{I} \in \Sigma$ with parameters a, b, c, and d iff it is in the domain of $\mathcal{I}$, it is distance between a and b pixels from the outer boundary all the gear bodies of size d of $\mathcal{I}$, and distance farther than c pixels from all the gear teeth of parameter a, b, and d of $\mathcal{I}$.

B_dilate$(\mathcal{I}, S)$:

A morphologically dilated image $\mathcal{J}$ of a binary image $\mathcal{I} = (D, F, B)$ by a structuring element S is defined by $\mathcal{J} = (D, (F \oplus S) \cap D, B)$.

$I\_and(\mathcal{I}, \mathcal{J})$:

Let the domains of two images $\mathcal{I}$ and $\mathcal{J}$ are same. Then, an ANDed image $\mathcal{K}$ of the two binary images $\mathcal{I} = (D, F, B)$ and $\mathcal{J} = (D, G, B)$ is defined by $\mathcal{K} = (D, F \cap G, B)$.

$ring\_fill(\mathcal{I}, r)$:

Let C be the collection union of the collection of isolated-holes with respect to a disk(r), $(r > 0)$, in a binary image $\mathcal{I} = (D, F, B)$. Then the output image $\mathcal{J}$ of the ring_fill morphological operator is defined by $\mathcal{J} = (D, F \cup C, B)$.

Goal Statement:

$$\exists \mathcal{G}, \forall A, \left(\begin{array}{c} \text{is-missing-tooth-space}(A, 3, 7, 8, 80, \mathcal{I}_0) \\ \Leftrightarrow \text{inImage}(A, \mathcal{G}) \end{array} \right)$$

Solution:

$\mathcal{G} = I\_and(\mathcal{I}_b, \mathcal{I}_c)$

where

$\mathcal{I}_a = B\_open(ring\_fill(\mathcal{I}_0, 20), D_{80})$

$$\mathcal{I}_b = I\_and \left(\begin{array}{l} I\_complement(B\_dilate(\mathcal{I}_a, D_7)), \\ B\_dilate(\mathcal{I}_a, D_3) \end{array} \right)$$

$$\mathcal{I}_c = I\_complement\Big(B\_dilate\big(I\_and(\mathcal{I}_0, \mathcal{I}_b), D_8\big)\Big)$$

6. Summary

Our goal is to begin to develop a method by which machine vision algorithms could be automatically generated for given tasks. We begin with some binary image tasks. We designed an algorithm development representation scheme based on first order predicate logic that can be used to explicitly represent the information involved in the development of morphological algorithms. We developed definitions and theorems expressing the relationships between morphological operations and a variety of spatial relationships. We showed that important spatial properties of a set of pixels in a binary image can be precisely described in our representation scheme. We have shown by example how a morphological algorithm can be created by a constructive proof in predicate logic which proves the existence of an image that satisfies a vision task specified by a logic statement. We listed without proof two other example problems. A dozen other such examples have been worked out. The next step of our work is to design and implement a computer program that employs a search procedure to construct such a proof and that, therefore, can generate computer vision algorithms.

References

1. Gillies, A.M., "Machine learning procedures for generating image domain feature detectors", *Doctoral Dissertation*, Department of Computer and Communication Science, University of Michigan, 1985.
2. Goad, C., "Special Purpose Automatic Programming for 3D Model-Based Vision", *Proceed. of the Image Understanding Workshop*, pp. 94–104, June 1983.
3. Haralick, R.M., S.R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-9, No. 4, pp. 532–550, 1987.
4. Haralick, R.M. and L.G. Shapiro, "Computer and Robot Vision", Book in preparation, 1990.
5. Hasegawa, J., H. Kubota, and J. Toriwaki, "Automated construction of image processing procedures by sample–figure presentation", *Proc. 8th ICPR*, pp. 586–588, Oct., 1986.
6. Ikeuchi, K. and T. Kanade, "Automatic generation of object recognition programs", *Proc. of IEEE*, Nov, 1988.
7. Joo, H. and R.M. Haralick, "A Compendium of Morphological Image Processing Algorithms", *Proceedings of the DAGM Symphosium*, Hamburg, Germany, September, 1989 (page unassigned).
8. Joo, H., R.M. Haralick, and L.G. Shapiro, "Toward the Automatic Generation of Mathematical Morphology Procedures Using Predicate Logic," *Technical Report # EE-ISL-90-07*, Intelligent Systems Laboratory, University of Washington, 1990.
9. Matsuyama, T., "Expert systems for image processing: Knowledge–based composition of image analysis process", *Technical report*, Dept. of Information Engineering, Tokohu University, Sendai, Miyagi 980, 1988.
10. Sakaue, K. and H. Tamura, "Automatic generation of image processing programs by knowledge–based verification", *IEEE CH2145*, pp. 189–192, 1985.
11. Serra, J., "From mathematical morphology to artificial intellegence", *Proceedings, 8th International Conference on Pattern Recognition*, Paris, France, pp. 1336–1343, Oct., 1986.
12. Serra, J., "Image analysis and mathematical morphology", *Academic Press Inc.*, 1982.
13. Sternberg, S.R., "Grayscale Morphology", *Computer Vision, Graphics, and Image Processing*, Vol. 35, pp. 333-355, 1986.
14. Vogt, R.C., "Automatic generation of morphological set recognition algorithms", *Doctoral Dissertation*, Electrical Engineering and Computer Science Department, University of Michigan, 1988. Also published by *Springer–Verlag*, 1990.