

PATTERN CLASSIFICATION AND SCENE ANALYSIS

RICHARD O. DUDA

PETER E. HART

**Stanford Research Institute,
Menlo Park, California**

A WILEY-INTERSCIENCE PUBLICATION

JOHN WILEY & SONS

New York • Chichester • Brisbane • Toronto • Singapore

Copyright © 1973, by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

Reproduction or translation of any part of this work beyond that permitted by Sections 107 or 108 of the 1976 United States Copyright Act without the permission of the copyright owner is unlawful. Requests for permission or further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.

Library of Congress Cataloging in Publication Data

Duda, Richard O.

Pattern classification and scene analysis.

"A Wiley-interscience publication."

Includes bibliographical references.

1. Perceptrons. 2. Statistical decision.

I. Hart, Peter E., joint author. II. Title.

Q327.D83 001.53'3 72-7008

ISBN 0-471-22361-1

Printed in the United States of America

30 29 28 27 26 25 24 23 22

noise, which is the occasional random complementing of binary picture values. A logical smoothing operator using a 3×3 window and having the required properties is the following, where for notational simplicity we will let the picture elements in the window be as shown below:

<i>a</i>	<i>b</i>	<i>c</i>
<i>d</i>	<i>e</i>	<i>f</i>
<i>g</i>	<i>h</i>	<i>i</i>

We define e' , the new picture value at the center cell of the window, by

$$e' = \bar{e}(abcd fghi) \vee e(a \vee b \vee c \vee d \vee f \vee g \vee h \vee i),$$

where $a \vee b$ means “ a or b ”, $\bar{a}$ means “not a ”, and ab means “ a and b ”. It is easy to see that this logical function has the desired properties. The new value e' is 1 if and only if e is 0 and all its neighbors are 1, or if e itself is 1 and at least one of its neighbors is 1. The advantage of logical smoothing over regularization, then, is that it allows us to specify more elaborate conditions under which a cell value is to be changed. In fact, regularization (of binary pictures) is a special case of logical smoothing, since the numerical operations of regularization can be defined by a boolean expression. Other boolean operators can be constructed to combat special noise idiosyncracies or to take advantage of prior knowledge about the figures likely to be encountered.

7.5 TEMPLATE MATCHING

7.5.1 Template Matching—Metric Interpretation

In a great variety of scene analysis problems the analyzer is confronted with one variation or another of a single simple question: Does the scene contain within it a picture of a previously specified object? The technique classically employed to answer this class of questions is called *template matching*. Let us consider template matching first within the simple context of binary pictures, and then extend our discussion to the general case. We proceed by means of an example.

Suppose we have a gradient picture, such as Figure 7.3, that shows elementary geometrical solids in outline form, and we would like to ascertain whether or not a triangle appears. (A triangle indicates the presence of a wedge.) A very simple approach to the problem might involve making a template, or mask, like the one shown in Figure 7.12 and scanning it systematically across the entire picture. If we found a position for which the “hole”

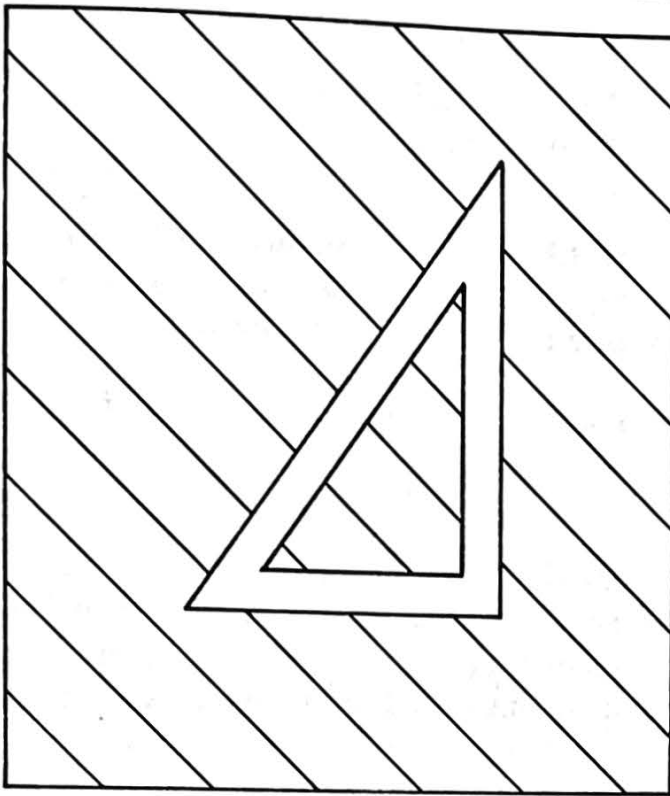


FIGURE 7.12. A simple template.

in the template was filled with white, we would declare that a triangle had been found at that position. An immediate objection to this procedure is that any sufficiently large region of solid white would result in our erroneously declaring that a triangle had been found.

This objection could be overcome if we were to look not merely for a white region that filled the template, but for a white triangular region surrounded by black regions. In Figure 7.13 we have shown diagrammatically how this can be done with a template. Using the new template, we announce that a triangle has been found only if each template region covers an area of the picture whose gray values corresponds to the template label. In other words, the 0 regions of the template must "show" only gray values of 0, and the 1 region must show only gray values of 1. Notice that the template in Figure 7.13 is itself a binary picture. (For simplicity, we have not shown the quantization of the template onto a quadrupled grid.) The size of the template, however, is typically smaller than the size of the original picture, since our aim is to discover the presence of a "sub-picture" within the given picture. Mathematically, we say that the domain of definition of the template is smaller than the domain of definition of the original picture.

In most practical applications, one would not expect to find the perfect template match contemplated above. A more realistic approach would be to define some measure of how well a portion of a picture matched a template.

One such definition is the following:

Let $g(i, j)$ be our digital picture, let
 $t(i, j)$ be the template, and let
 D be the domain of definition of the template.

(D , for example, might be a 20×20 square, whereas the domain of definition of the picture might be a 500×500 square.) Then a measure of how well a portion of the picture matches the template can be defined as

$$M(m, n) = \sum_{\substack{i, j \text{ such} \\ \text{that} \\ (i-m, j-n) \\ \text{is in } D.}} |g(i, j) - t(i - m, j - n)|. \quad (1)$$

Notice that this definition amounts to translating the template $t(i, j)$ to position (m, n) in the picture and setting $M(m, n)$ equal to the number of cells for which the picture and the translated template differ in gray value. Since we are presumably interested in finding a match to the template anywhere in the picture, we would have to compute $M(m, n)$ for all template translations (m, n) and note those translations for which $M(m, n)$ is small.

Let us abstract from the previous discussion the essential elements of template matching. In terms of picture functions, we are looking for a region

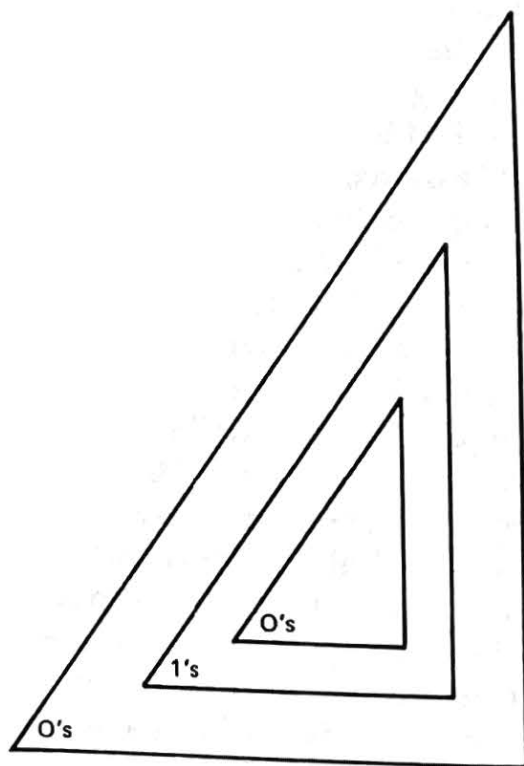


FIGURE 7.13. A more refined template.

of the picture plane such that the picture function in that region is similar to a previously specified picture function called the template. Hence, we need in general a means for measuring the similarity, or distance, between two picture functions, and the notion of a metric becomes useful. For the moment we need not be concerned with the formal definition of the class of functions called metrics, except to note that it is the standard abstraction of the notion of Euclidean distance. Our definition of $M(m, n)$ given in (1) satisfies the definition and is sometimes called the metric based on the L^1 norm. Two other metrics are

$$E(m, n) = \left\{ \sum_i \sum_j [g(i, j) - t(i - m, j - n)]^2 \right\}^{1/2} \quad (2)$$

and

$$S(m, n) = \max_{i, j} |g(i, j) - t(i - m, j - n)|, \quad (3)$$

where in each case the range of i and j is the same as in (1). The definition in (2) is the standard Euclidean distance between two vectors. The definition in (3) is sometimes called the metric based on the L^∞ norm. Note that these definitions are not restricted to binary pictures although, as we saw above, a definition may have a particularly simple interpretation when restricted to binary pictures.

Let us examine definition (2) in a little more detail. It is often convenient to remove the square root by considering the measure of distance to be $E^2(m, n)$. If we do this, and multiply out, we obtain

$$E^2(m, n) = \sum_i \sum_j [g^2(i, j) - 2g(i, j)t(i - m, j - n) + t^2(i - m, j - n)],$$

where, as usual we sum over all (i, j) such that the arguments of t remain within its domain of definition. Now as we move the template around the picture by varying m and n the summation of the last term is unchanged, since for any m and n the range of the arguments of t^2 is the domain of definition of t . The summation of the first term—the *picture energy* in the window—in general varies with (m, n) , since the range of i and j vary with m and n . Suppose for a moment that this variation in picture energy is small enough to be ignored. Then $E^2(m, n)$ is small when $\sum_i \sum_j g(i, j)t(i - m, j - n)$ is large. Accordingly, we define $R_{gt}(m, n)$, the *cross-correlation* between the two functions g and t , by

$$R_{gt}(m, n) = \sum_i \sum_j g(i, j)t(i - m, j - n), \quad (4)$$

where, as always, we sum over all i and j within the domain of the translated template. We can use this definition as a measure of the similarity between the template and the region of the picture in the vicinity of (m, n) ; the template and the picture are declared similar when the cross-correlation is

large. Of course, if we translated the template to a region of the picture that was all white then the cross-correlation would have a high value. This is just another way of saying that our initial supposition (that $\sum_i \sum_j g^2(i, j)$ is independent of (m, n)) is grossly unjustified. One alternative to computing E^2 is to compute the *normalized cross-correlation* $N_{gt}(m, n)$ given by

$$N_{gt}(m, n) = \frac{1}{\left[\sum_i \sum_j g(i, j)^2 \right]^{1/2}} R_{gt}(m, n),$$

where we place the usual restriction on the range of i and j . By the Cauchy-Schwarz inequality, we have

$$N_{gt}(m, n) \leq \left[\sum_i \sum_j t(i - m, j - n)^2 \right]^{1/2} \quad (5)$$

with equality if and only if the picture function in the region of interest is a multiple of the template function. Hence, the normalized cross-correlation has a maximum value when the match with the picture function is perfect (up to a scale factor). On the other hand, in some circumstances our supposition about $\sum \sum g^2$ is satisfied precisely. For example, suppose we have a binary picture and we let white be 1 but let black be -1 instead of the usual zero. Then the sum of the squares of the picture function values over any domain of fixed size is constant, and the cross-correlation has a maximum exactly when E^2 has a minimum. The use of cross-correlation as a measure of similarity has its roots in certain analogous, but not identical, problems in pattern classification and signal detection. It is still often used, partly because there exist a number of efficient implementations. As we have seen, however, using cross-correlation in template matching involves a tacit assumption that the picture energy in every window is roughly the same.

Let us reconsider our introductory example: the problem of locating a triangle in Figure 7.3. Assuming that we had settled upon one of the distance measures as our measure of similarity, we still must design the template itself. A little reflection will lead us to the conclusion that the problem as stated can only be solved by having a whole family of templates. The wedge, in the absence of any explicit constraints, can appear anywhere in the scene and can be viewed from any angle. Its triangular face, then, can appear in a wide range of sizes and can have a wide range of apparent interior angles. For each such wedge orientation we evidently need a separate template and, since each template must be scanned across the scene, the computational burden is likely to be large. In such situations a common approach is to replace the *global* template by a set of *local* templates. The local templates are designed to match various parts of the object of interest, the rationale being that the individual parts vary in appearance less than the entire object.

In our triangle example, we might first look for three separate lines and then worry about devising a method for recognizing when three lines form a triangle. This approach is reasonable in a large class of problems whose hallmark is wide variability in the appearance of the object of interest. On the other hand, there are some pattern classification problems that can be sufficiently constrained or stylized to insure that each pattern will always appear in isolation and at the same size and orientation. In these simple problems it is sometimes practical to construct a single template matched to the pattern as a whole. In other words, the choice of whether to use local or global templates is primarily dictated by the anticipated variability in the pictures to be processed.

Template matching appears in a variety of superficially different, but essentially equivalent, forms. The choice of forms ultimately becomes one of convenience and efficiency. As a simple example, suppose we wanted to detect the presence of a vertical edge. We could use a "binary" template (like the one in Figure 7.13, but shaped like a single vertical line) and apply it to a gradient picture. Alternatively, we could easily devise a template matched to the shape of a vertical line in the original picture. (Actually, if we operated on the original picture we would probably want to use a pair of templates, one for dark-light transitions and one for light-dark transitions. This symmetry problem is avoided if we use the gradient picture because we usually take the magnitude of the gradient.) Figure 7.14 shows a template that will detect a left to right transition of dark to light along a vertical line if it is used with the similarity definition in (1). The template region labelled "low" would have values near the dark end of the gray scale, and conversely for the region labelled "high." A possible problem with this template is that it is not invariant to the absolute values of the gray scale; adding a constant to the gray scale will change the degree of match. This consideration might lead us

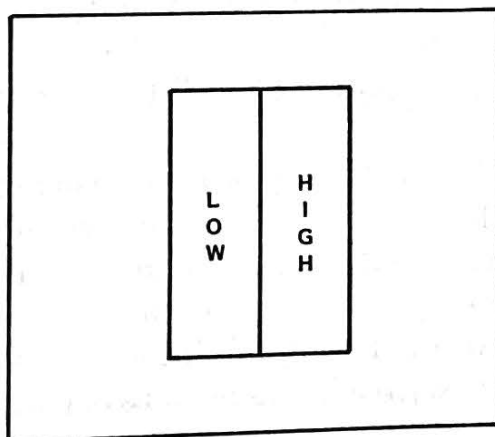


FIGURE 7.14. A template for vertical edges.

to a procedure in which the template regions labelled "low" and "high" were respectively given values of -1 and $+1$, and we used correlation as our definition of similarity (4). This procedure is equivalent to subtracting adjacent columns and adding up the differences. In other words, the procedure looks for differences in gray scale along a vertical line, and is therefore almost the same thing as using a binary template on a gradient picture.

We should note that an important aspect of template matching, whether performed on the original picture or a processed version of the original picture, is that the operation makes use of only local information. If we are trying to decide, say, whether a vertical line is present in a given region of the picture, the only thing that affects our decision is the set of intensity values in that region. This local aspect is both the reason for the appealing simplicity of template matching and the source of its most stringent limitations.

7.5.2 Template Matching—Statistical Interpretation

Certain forms of template matching can be interpreted within the framework of statistical decision theory developed in Part I. Let us illustrate this interpretation with a simple example.

Suppose we have a binary picture $g(i, j)$ and suppose further that the picture is known to depict one of two possible ideal scenes, say $r(i, j)$ or $s(i, j)$. The actual picture g , however, is only an imperfect representation of an ideal scene. Let us assume that with probability $p > 1/2$ a given picture cell in g has the value specified by the ideal scene that g represents. If these cell probabilities are independent then we can easily write the probabilities of observing a picture g conditioned on either r or s being the ideal scene. Specifically, if we let ω_r and ω_s be respectively the events that $r(i, j)$ and $s(i, j)$ are the ideal scenes, we have

$$P(g \mid \omega_r) = \prod_{i,j} p^{1-|g(i,j)-r(i,j)|} (1-p)^{|g(i,j)-r(i,j)|} \quad (6)$$

and

$$P(g \mid \omega_s) = \prod_{i,j} p^{1-|g(i,j)-s(i,j)|} (1-p)^{|g(i,j)-s(i,j)|} \quad (7)$$

Notice in these equations that the exponents of p are one if the ideal picture and the actual picture have the same value for a given (i, j) , and zero otherwise; conversely, the exponents of $(1-p)$ are one if the ideal and actual pictures are different for a given (i, j) .

Suppose now that we want to determine whether the picture g depicts scene r or scene s . We know from the discussions of Chapter 2 that a minimum probability of error classification rule computes the a posteriori probabilities $P(\omega_r \mid g)$ and $P(\omega_s \mid g)$ and selects either ω_r or ω_s according to which probability is larger. If we assume for simplicity that the a priori probabilities

of the two scenes are equal, then we know that this rule is equivalent to a rule that selects the class for which the conditional probability is larger. As in previous chapters, we can simplify things by selecting the class for which the logarithm of the conditional probability is maximized. Thus, from (6) and (7), it suffices to compute

$$\log P(g \mid \omega_r) = \left[\log \left(\frac{1-p}{p} \right) \right] \sum_{i,j} |g(i,j) - r(i,j)| + \sum_{i,j} \log p \quad (8)$$

and

$$\log P(g \mid \omega_s) = \left[\log \left(\frac{1-p}{p} \right) \right] \sum_{i,j} |g(i,j) - s(i,j)| + \sum_{i,j} \log p. \quad (9)$$

Now notice that $\log[(1-p)/p]$ is negative, since we have specified $p > 1/2$; furthermore, the last terms of (8) and (9) are the same. Hence, the minimum probability of error rule for classifying picture g selects ω_r if

$$\sum_{i,j} |g(i,j) - r(i,j)| < \sum_{i,j} |g(i,j) - s(i,j)| \quad (10)$$

and selects ω_s otherwise. Thus, we see that in this example our statistical assumptions lead to a minimum probability of error rule that is a template match using the similarity measure defined by Eq. (1); the ideal pictures r and s , as we would anticipate on intuitive grounds, are the templates.

There are several points about the preceding example worth noting. First, suppose we know that the picture g may in fact show neither r nor s ; hence we want to allow the alternative of rejecting g in addition to the alternatives of classifying it as scene r or scene s . This refinement can also be analyzed in statistical terms. As a matter of practice, however, it is more common to set a threshold on the template similarity measure. If the best match achieved fails to satisfy the threshold, then g is rejected; that is, g is declared not to depict any of the ideal scenes.

A different typical situation was discussed earlier. Often the ideal scene of interest is only a subpart of g . In this case, the templates are defined on domains smaller than the domain of g and are scanned across the picture. Each template position provides us with a new classification problem that, in principle at least, can be analyzed along the lines of our illustrative problem.

The application of formal statistical classification methods to problems of detecting objects in pictures has proved to be difficult in practice. One of the chief difficulties has been in establishing useful statistical assumptions. As an example of this difficulty, consider the problem just mentioned of scanning a template across a picture to detect an object. Suppose good template matches occurred at two template locations only a single picture cell apart. Certainly, we would not want to say that two separate objects had been detected; much

more probably, the same object in the picture is responsible for both matches. Formally, this would be represented statistically by nonindependent probabilities. In practice, however, it is usually far easier to develop ad hoc procedures for circumventing such difficulties than it is to overcome them by a formal analytical assault. Nevertheless, the statistical viewpoint provides worthwhile guidance, if not universal prescriptions, for the design of template-matching procedures.

7.6 REGION ANALYSIS

7.6.1 Basic Concepts

In a previous section we discussed means for simplifying a picture by enhancing or extracting edges. We now turn our attention to a technique that is in a sense complementary. This technique, called *region analysis*, attempts to simplify a digital picture by partitioning it into a set of disjoint regions. In the simplest case, each region is composed of picture cells having the same gray value and connected to each other. Before proceeding any further, therefore, we must make precise the notion of when two picture cells are connected. In particular, we must define whether a cell is connected to all eight surrounding cells (8-connected), or whether it is connected only to those four cells sharing a common edge (4-connected).

To illustrate the subtleties of this question, consider the simple binary picture in Figure 7.15, where the unmarked cells are background 0's. Suppose we initially define connectedness to be 8-connectedness. Then, since cells sharing only a common vertex are connected, the figure is connected. However, by the same token the background is also connected—the “hole” in the figure is not separated from the “surrounding” background. Hence, we are in the uncomfortable position of having on our hands a connected, shell-like figure with no inside. Suppose, therefore, that we use 4-connectedness as our definition. Then the figure is not connected. Neither, however, is the background. Hence, we are in the equally embarrassing position of having a disconnected figure that has a distinct interior. This state of affairs reflects a

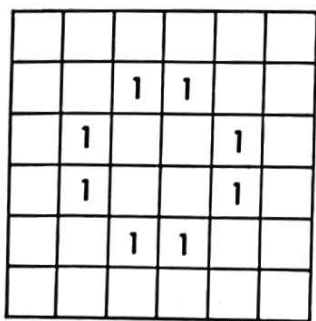


FIGURE 7.15. Connectivity on a quadruled grid.