Morphological Approach for Template Matching

GERALD JEAN FRANCIS BANON
SERGIO DONIZETE FARIA

INPE/DPI – Instituto Nacional de Pesquisas Espaciais / Divisão de Processamento de Imagens
Caixa Postal 515
12201–970 – São José dos Campos, SP, Brazil
banon@dpi.inpe.br
faria@dpi.inpe.br

Abstract. In this work, a morphological representation of a template matching algorithm for gray–scale images is presented. The algorithm is the composition of the so–called template matching operator with the maximum gray–level location operator, which can be both expressed in terms of the following classes of mathematical morphology elementary operators: dilations, erosions and anti–dilations. Moreover, the algorithm is applied to remote sensing images.

Keywords: Image processing, mathematical morphology, template matching, remote sensing.

1 Introduction
The template matching problem consists of finding out in a search image the location of a template (pattern). In some applications, for example in image registration, this template is extracted from a reference image.

In literature, we can find many techniques to perform template matching (Ballard & Brown, 1982; Gosh- tasby, 1985; Li & Dubes, 1985; Lemmens, 1988). These techniques can be classified in three groups: area–based (Barnea & Silverman, 1972; Moik, 1980), feature–based (Medioni & Nevatia, 1984; Hannah, 1988; Toth & Schenk, 1992; Flusser & Suk, 1994) and structural (Bins, 1988; Ventura et al., 1990; Haala & Vosselman, 1992).

The classical technique, among the area–based ones, derives from the normed vector space theory (Luenberger, 1969). In this approach, the images or templates are seen as vectors.

When the Euclidian norm is used, we get the well known correlation method (Barnea & Silverman, 1972) and the best fit is defined in terms of maximal correlation between two vectors. In this approach, the correlation is a sum of products.

When the so called $l_1$ norm is used, Maragos (1988) showed that minimizing it is equivalent to maximizing a nonlinear correlation, a sum of minima, which he called morphological correlation.

Following the nonlinear approach, Ronse (1996) characterized the template extraction by three requirements: the so called “overcondensation”, anti–extensivity and idempotence. It is shown that an operator having these properties is the composition of an inf–separable operator (Banon & Barrera, 1993) followed by a dilation.

Moreover, Khosravi and Schafer (1996) used the sum of a gray–scale erosion and a gray–scale anti–dilation (both vertically invariant) with the same structuring function to perform a template matching within a signal corrupted by Gaussian or impulsive noises.

Our main objective in this paper is to introduce a simple template matching algorithm based on area and defined within the framework of the lattice theory and, more specifically, the mathematical morphology. In our approach the correlation is a sum of intersections of thresholdings.

More precisely, we use some elementary morphological operators (dilations, erosions and anti–dilations) to construct a template matching algorithm. This is an important issue because it can unify the way we process images.

Different from Maragos, we do not assume the vector space structure and different from Ronse, we do not include a dilation for template reconstruction at the end of the matching procedure. Finally, different from Khosravi and Schafer we use pairs of erosion and anti–dilation with distinct structuring functions.

In Section 2, we recall some basic definitions from mathematical morphology. In Section 3, we present our template matching algorithm in terms of some elementary morphological operators and, finally, in Section 4 we show an application to remote sensing images.

2 Mathematical Morphology
Originally, mathematical morphology was developed in the context of image analysis (Serra, 1982). Nowadays, an important part of it belongs to the lattice theory (Serra, 1988; Heijmans & Ronse, 1990).

A lattice is a partially ordered set in which the supremum and the infimum of two elements exist (Birkhoff, 1967). We denote as $a \lor b$ and $a \land b$, respectively, the supremum and the infimum of two elements $a$ and $b$ in the
lattice. If the lattice is finite then it has a least and a greatest element that we denote, respectively, as $o$ and $i$.

Let $(L_1, \leq)$ and $(L_2, \leq)$ be two finite lattices. Let $\psi$ be a mapping from $(L_1, \leq)$ to $(L_2, \leq)$. We will call $\psi$ an operator.

By definition,

- $\psi$ is a dilation iff $\psi(a \lor b) = \psi(a) \lor \psi(b)$ and $\psi(o) = o$;
- $\psi$ is an erosion iff $\psi(a \land b) = \psi(a) \land \psi(b)$ and $\psi(i) = i$;
- $\psi$ is an anti-dilation iff $\psi(a \lor b) = \psi(a) \lor \psi(b)$ and $\psi(o) = i$;
- $\psi$ is an anti-erosion iff $\psi(a \land b) = \psi(a) \land \psi(b)$ and $\psi(i) = o$;

for any $a$ and $b$ in $L_1$. For the general case of complete lattice see Serra (1988), Heijmans & Ronse (1990), Banon & Barrera (1993).

These operators are called elementary operators of the mathematical morphology because from them, it is possible to represent any operators from $(L_1, \leq)$ to $(L_2, \leq)$ (Banon & Barrera, 1991, 1993).

3 Template Matching Algorithm

Exact template matching is useless in most of the practical applications. For example, in remote sensing two images of the same scene at different times or from different sensors are almost never equal. A template matching algorithm must be based on inexact matching (Shapiro & Haralick, 1981; Bins, 1988; Lemmens, 1988). From the mathematical morphology point of view, inexact matching can be achieved by defining intervals around reference values and by testing if all the values of interest are falling into them. Nevertheless, the requirement that all the conditions are satisfied is not reasonable in practice, and what we suggest here is to look for the situations where a maximum number of conditions are satisfied.

The template matching algorithm proposed is shown in Figure 1. In this figure, $f$ is the reference image and $g$ the search image. The template matching algorithm is composed of several operators, one in each block. In order to present each of them, we are now introducing some mathematical notations.

Let $Z$ be the set of integer numbers, we denote as $Z^2$ the Cartesian product $Z \times Z$. Let $E$ be a finite rectangle of $Z^2$, i.e., $E \triangleq I_1 \times I_2$, where $I_1$ and $I_2$ are two intervals of $Z$, and let $K_m$ be the interval $[0, m]$ of $Z$. We denote the set of mappings from $E$ to $K_m$ as $K_m^E$. Actually, $K_m^E$ represents the set of images with domain $E$ and range $K_m$. In this work, we will need to deal with images having different domains, for this reason, we introduce one more finite rectangle of $Z^2$ that we call $D$.

In Figure 1, the so-called measure $\mu$ is a mapping from $E$ (or $D$) to $Z$, defined by,

$$\mu(f) \triangleq \frac{1}{2} + \left(1/\#E\right) \left(\sum_{x \in E} f(x)\right),$$

for any $f$ in $K_m^E$, where $\#E$ is the number of elements of $E$. Actually, $\mu(f)$ is an approximation of the average of the image $f$.

The template matching algorithm is aimed at searching for a given template $f_w$ in the search image $g$. The template $f_w$ is a subimage of $f$ in the sense that $W$ is a rectangle of $Z^2$ called window.

In this section, for the sake of simplicity and without lose of generality we assume that $W$ has a center (in other words, its number of rows and columns is odd) and this center is located at the origin of $Z^2$, i.e., the point $(0, 0)$.
The template \( f_w \) can be chosen manually or through an automatic procedure. In Figure 1, this procedure is represented by the operator \( \psi_1 \). In this work, we will not go into details about it.

The operator \( \psi_2 \) produces a pair of templates \((f_w, f_w')\) from the template \( f_w \) in \( K_m^w \). This pair of templates is defined in the following way:

\[
\begin{align*}
    f_w(x) &\triangleq \max \{ 0, \min \{ m, f_w(x) + c_1 \} \} \quad (2) \\
    f_w'(x) &\triangleq \max \{ 0, \min \{ m, f_w(x) + c_2 \} \} \\
\end{align*}
\]

for any \( x \) in \( W \), where \( c_1 \) and \( c_2 (c_1 \leq c_2) \) are two integers depending on the averages \( \mu(f) \) and \( \mu(g) \), and a parameter \( F \).

The two integers \( c_1 \) and \( c_2 \) are calculated in the following way:

\[
c_1 = d\mu - F/2 \quad \text{and} \quad c_2 = d\mu + F/2 \quad (4)
\]

where,

\[
d\mu = \mu(g) - \mu(f). \quad (5)
\]

The parameter \( F \) defines the length of the interval \([c_1, c_2]\) centered at \( d\mu \).

The parameter \( d\mu \) is intended for reducing the image brightness difference. If \( f \) and \( g \) have the same average, then \( d\mu \) is zero and no first order adjust is needed. Furthermore, we assume that both images have the same standard deviation, so no second order adjust is needed.

The search image \( g \) is looked over in order to find out the location of the best fit with the template \( f_w \). This is done through the so called template matching operator \( \phi \) (Figure 1). At this stage, we need to introduce some new notations.

Let \( G \) be the subset of \( \mathbb{Z}^2 \), given by \( G = D \ominus W \), where the symbol \( \ominus \) represents the Minkowski subtraction (Banon & Barrera, 1994), and let \( l \) be an integer number between 0 and \( m \). The subset \( G \), i.e., the difference of \( D \) and \( W \), is a smaller rectangle compared to \( D \), as shown in Figure 2. It will correspond to a reduce domain for further processed images.

For any integer number \( i \) in \( n \), we denote as \( \epsilon_i \) and \( \delta_i \) the operators from \( K_m^D \) to \( K_1^G \) defined by:

\[
\begin{align*}
    \epsilon_i(g)(x) &= \begin{cases} 1 & \text{if } g(x + w_i) \geq l \\ 0 & \text{otherwise} \end{cases} \quad (6a) \\
    \delta_i(g)(x) &= \begin{cases} 1 & \text{if } g(x + w_i) \leq l \\ 0 & \text{otherwise} \end{cases} \quad (6b)
\end{align*}
\]

for any \( g \in K_m^D \) and \( x \in G \), where \( i \mapsto w_i \) is a bijection from \( n \) to \( W \), just for numbering the elements of \( W \).

The pixel of the binary image \( \epsilon_i(g) \) (the transformed of \( g \in K_m^D \) through \( \epsilon_i \)), at position \( x \) has value 1 whenever the pixel of \( g \) at the neighborhood position \( x + w_i \) has a value greater than or equal to \( l \). For the operator \( \delta_i \), we just reverse the binary relation \( \geq \).

By construction, the thresholding operators \( \epsilon_i \) and \( \delta_i \) are, respectively, erosions and anti–dilations from \( (K_m^D, \leq) \) to \( (K_1^G, \leq) \), where \( \leq \) is the pointwise ordering derived from the ordering between gray–levels (Banon & Barrera, 1993; Faria, 1997).

We are now ready to introduce the template matching operator. We call template matching operator the operator \( \phi \) from \( K_m^D \) to \( K_1^G \), given by:

\[
\phi \triangleq \sum_{i \in n} \lambda^i \quad (7)
\]

where the \( \lambda^i \)’s are \( n \) operators from \( K_m^D \) to \( K_1^G \), given by:

\[
\lambda^i \triangleq \epsilon_i(g_{f_w}(w_i)) \cotr \delta_i(g_{f_w}(w_i)) \quad (8)
\]

Actually, the operators, like the \( \lambda^i \)’s, which are the intersections of an erosion and an anti–dilation, are important in operator decomposition and are called sup–generating operators (Banon & Barrera, 1993). These operators for binary images are similar to the well known Hit–or–Miss operators (Serra, 1982).

Figure 3 shows the operator \( \phi \).
The pixel of the image \( \lambda'(g) \) (the transformed of \( g \in K_n^G \) through \( \lambda' \)), at position \( x \in G \), has the value \( \lambda'(g)(x) \). This value is called the matching condition.

From Equations (6) and (7), the matching condition is 1 (i.e., satisfied) whenever \( g(x + w_i) \) falls in the interval \([f_u(w_i), f_u'(w_i)]\) and 0 (i.e., not satisfied) otherwise. Hence, we can see that \( \phi(g)(x) \) simply indicates the number of satisfied conditions among \( n \) possible ones.

The operation of sum, appearing in the definition of \( \phi \), is an extension to operators of the sum of intersections of thresholding operators.

In order to present it we need to introduce some more notation.

For any \( x \in G \) or 1 which returns simply the number of 1’s.

In other words, \( \phi \) can be defined in an equivalent way by:

\[
\phi(g)(x) = \# \{ i \in n : g(x + w_i) \in [f_u(w_i), f_u'(w_i)] \},
\]

for any \( g \in K_n^G \) and \( x \in G \). We can see that “behind” this expression we have a sum of \( n \) sup-generating operators.

Furthermore, we observe that the operator \( \phi \) is the sum of intersections of thresholding operators.

In the next step, the image \( g' \) produced by \( \phi \) is looked over through an operator \( \psi \) which locates its maximum gray-level value and produces the image \( h \) (Figure 1). In order to present it we need to introduce some more notations.

For any integer number \( l \) between 0 and \( n \), we denote as \( \delta_l \) and \( \epsilon_l \) the operators from \( K_n^G \) to \( K_1^G \) defined by:

\[
\delta_l(f)(x) = \begin{cases} 1 & \text{if } f(x) > l \\
0 & \text{otherwise}
\end{cases}
\]

\[
\epsilon_l(f)(x) = \begin{cases} 1 & \text{if } f(x) \geq l \\
0 & \text{otherwise}
\end{cases}
\]

for any \( f \in K_n^G \) and \( x \in G \).

Like the operators of Equations (6), \( \delta_l \) and \( \epsilon_l \) are thresholding operators, but on the other hand, they are not neighborhood operators.

Furthermore, we denote as \( \delta^*_G \) the operator from \( K_1^G \) to \( K_1^G \) defined by:

\[
\delta^*_G(f) = \begin{cases} 1_g & \text{if } f = 0_g \\
0_g & \text{otherwise}
\end{cases}
\]

for any \( f \in K_1^G \), where \( 0_g \) and \( 1_g \) are mappings from \( G \) to \( K_1 \), assuming, respectively, the constant values 0 and 1.

Finally, for any integer number \( l \) between 0 and \( n \), we denote as \( \delta^*_l \) the operator from \( K_n^G \) to \( K_1^G \) defined by the composition:

\[
\delta^*_l = \delta^*_G \circ \delta_l.
\]

By construction, the operators \( \delta_n \), \( \epsilon_l \), and \( \delta^*_l \) are, respectively, dilations, erosions and anti-dilations from \( (K_n^G, \leq) \) to \( (K_1^G, \leq) \), and \( \delta^*_G \) is an anti-dilation from \( (K_1^G, \leq) \) to \( (K_1^G, \leq) \) (Banon & Barrera, 1993; Faria, 1997).

We are now ready to introduce the operator for maximum gray–level location. We call maximum gray–level location operator the operator \( \psi \) from \( K_n^G \) to \( K_1^G \), given by:

\[
\psi = \bigvee_{l = 0, \ldots, n} \lambda_l
\]

where the \( \lambda_l \)'s are \( n + 1 \) operators from \( K_n^G \) to \( K_1^G \), given by:

\[
\lambda_l = \epsilon_l \wedge \delta^*_l.
\]

For the same reason that we have seen above, the operators \( \lambda_l \)'s are sup–generating operators. Figure 4 shows the operator \( \psi \).

![Fig. 4 – Maximum gray–level location operator.](image-url)

The pixel of the binary image \( \bar{\psi}(g') \) (the transformed of \( g' \in K_n^G \) through \( \bar{\psi} \)), at position \( x \in G \), has value 1 whenever the pixel of \( g' \) at this position has the greatest value among all the pixel values, i.e., has the value \( \max g'(G) \).

In other words, \( \bar{\psi} \) can be defined in an equivalent way by:

\[
\bar{\psi}(g')(x) = \begin{cases} 1 & \text{if } g'(y) \leq g'(x) \ (y \in G) \\
0 & \text{otherwise}
\end{cases}
\]

for any \( g' \in K_n^G \) and \( x \in G \).

Actually, in the above representation of the operator for maximum gray–level location, there is an implicit thresholding decomposition of the input image (Banon, 1997).

The domain of the resulting image \( h = \bar{\psi}(g') \) is \( G \). In order to return to the original domain \( D \) of the search image, which is larger than \( G \) (actually, \( D = G \oplus W \)), we need one more step. Processing \( h \) through the last operator \( \delta_n \), we get the image \( h' \), showing where the template matching occurs in the proper domain of the search.
image (Figure 1). To achieve this expansion effect, while preserving the information contained in $h$, the operator $\delta_{\rho}$ from $K \downarrow$ to $K \uparrow$ must be define in this way:

$$\delta_{\rho}(h)(x) = \begin{cases} h(x) & \text{if } x \in G \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

for any $h \in K \downarrow$ and $x \in D$. Furthermore, we observe that $\delta_{\rho}$ distributes over the union, i.e., it is a dilation from $(K \downarrow \subseteq) \rightarrow (K \uparrow \subseteq)$.

Figure 5 shows a numerical example where the images and the template are reduced to one–dimensional signals. In this example, $w_2$ coincides with the origin of $Z$.

### 4 Application to Remote Sensing Images

The template matching algorithm of Section 3 was used on remote sensing images (Faria, 1997). We present now the result of its application to a scene acquired from the Thematic Mapper (TM) sensor of the LANDSAT–5 satellite (path=220, row=77, quadrant=A, band=5), of a region near Itapeva in São Paulo State. We have used two TM subimages of size 360 by 360, one taken in September 09, 1990, as the reference image, and the other taken in July 18, 1994, as the search image.

In our experiment, we have chosen three different templates of size 51 × 51 in the reference image. The first column in Table 1 shows the position of the center of these templates. For each one, we have used three different values for the parameter $F$. For each of these values, Table 1 shows the maximum gray–level in $g'(G)$ and its position in the search image domain. The last column of Table 1 shows for each template its visual location obtained by a human operator.

<table>
<thead>
<tr>
<th>Table 1 – Experimental Results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters of the Experiment</strong></td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td>Template Center in the Reference Image $f$ (col., row)</td>
</tr>
<tr>
<td>(241,37)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>(284,120)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>(37,243)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

In this experiment, the maximum gray–level in $g'(G)$ is unique. This is generally the case in remote sensing images. If it is not unique, some decision should be made depending on the final objective of the template matching.

Although the correct matching depends on the parameter $F$, we have observed that it can be achieved within a fairly large interval of values. For example, from Table 1, we observe that the result is not affected so much by a change in $F$.

Figure 6 shows the reference image $f$ and the search image $g$ used in the experiment. The small square in the domain of the reference image contains the second template of Table 1, i.e., the template of size 51 × 51 centered at position (284,120) as indicated by the small cross. The small square in the domain of the search image indicates the template visual location. Moreover, Figure 6 shows the images $g'$ and $h'$ obtained by running the algorithm with $F$ equal to 15. The first one is a gray–level image which displays at each position the number of satisfied conditions in the matching process, and the last
We observe that the output image $g'$ of the template matching operator is very much like a correlation image, the brighter the pixel, the better the fit.

In this experiment, we see that the template matching algorithm was able to find out, in the search image, the right location of the template.

5 Conclusion
The template matching algorithm presented in this paper is the composition of three operators which have been defined in terms of dilations, erosions, and anti-dilations,
that is, in terms of some mathematical morphology elementary operators.

Different from other techniques, our template matching operator is the sum of intersections of thresholding operators.

The algorithm has been successfully applied to a pair of remote sensing images of the same scene. Three different templates from the reference images have been encountered at the right location in the search image.

In this way, we have shown that a simple template matching operator can be defined in terms of the elementary morphological operators.

**Acknowledgment**

During the period of this work, the authors have obtained support from CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) through the AnMoMat project (contract 680067/94–9).

**References**


