HIGH PERFORMANCE ALGORITHMS FOR OBJECT RECOGNITION PROBLEM BY MULTIRESOLUTION TEMPLATE MATCHING

L. Prasad
Theoretical Physics Division
Los Alamos National Laboratory
Los Alamos, New Mexico

S.S. Iyengar
Department of Computer Science
Louisiana State University
Baton Rouge, LA 70803-4020

Abstract

Template matching is a fundamental method of detecting the presence or the absence of objects and identifying them in an image. A template is itself an image that contains a feature or an object or a part of a bigger image, and is used to search a given image for the presence or the absence of the contents of the template. This search is carried out by translating the template systematically pixel-by-pixel all over the image, and at each position of the template the closeness of the template to the area covered by it is measured. The location at which the maximum degree of closeness is achieved is declared to be the location of the object detected.

The problem of object/shape recognition in image is addressed in this paper in a multiresolutional setting using pyramidal decomposition of images with respect to an orthonormal wavelet basis. A new approach to efficient template matching to detect objects using computational geometric methods is put forward. An efficient paradigm for object recognition is described in detail with a complexity analysis.

1 Introduction

The problem addressed here is that of object recognition using template matching. It is assumed that an input image contains finitely many objects from some superset S of objects. A database of templates is stored, wherein each template τ contains an object from S and is an $M \times M$ image. The input image $\mathcal{I}$ is to be searched for presence or absence of objects from S and the types, location and apparent sizes of those objects present in $\mathcal{I}$ are to be reported, using the template database of objects in S.

Since in the above described search process a minimum is always attained in the distance between the template and the area covered by it, a predetermined proximity bound is set, as an upper bound for the minimum distance observed. If the minimum observed in the search process is less than this bound, then the object is declared to be present at the location at which the minimum is attained, else it is declared to be absent.

More precisely, let

$$\mathcal{I} = \{\mathcal{I}(i,j) \mid 0 \leq i,j \leq N-1\}$$

and

$$\tau = \{\tau(i,j) \mid 0 \leq i,j \leq M-1\}$$

be a template of a given object $\mathcal{O}$. A measure of the distance of the template τ from a region of the image $\mathcal{I}$ covered by τ, is given by

$$\mathcal{D}(m,n) = \left[\sum_{j=n}^{M-1+n} \sum_{i=m}^{M-1+m} [\mathcal{I}(i,j) - \tau(i-m, j-n)]^2\right]^{\frac{1}{2}}$$

This is the discrete version of the distance formula

$$\mathcal{D}(u,v) = \left[\int \int_{\mathcal{D}} [\mathcal{I}(x,y) - \tau(x-u, y-v)]^2 \, dx dy\right]^{\frac{1}{2}}$$

where, the image $\mathcal{I}$ and the template τ are considered to vary continuously and $\mathcal{D}$ is the domain of definition of the template.

$$\mathcal{D}^2(m,n) = \sum_{j=n}^{M-1+n} \sum_{i=m}^{M-1+m} \mathcal{I}^2(i,j) + \sum_{j=n}^{M-1+n} \sum_{i=m}^{M-1+m} \tau^2(i-m, j
- 2 \sum_{j=n}^{M-1+n} \sum_{i=m}^{M-1+m} \mathcal{I}(i,j)\tau(i-m, j-n)$$

$$= \sum_{j=n}^{M-1+n} \sum_{i=m}^{M-1+m} \mathcal{I}^2(i,j) + \sum_{j=n}^{M-1+n} \sum_{i=n}^{M-1+m} \tau^2(i,j)$$

1082-3409/95 $04.00 © 1995 IEEE

$$-2\sum_{j=n}^{M-1+n}\sum_{i=m}^{M-1+m}\mathcal{I}(i,j)\tau(i-m,j-n)$$

The second term on the r.h.s of the above equation is a constant independent of m and n, being the square of the energy of the template. If it can be assumed that the energy of the image $\mathcal{I}$ over any window of the size of the template remains approximately constant, then $\mathcal{D}(m,n)$ is minimum when

$$\mathcal{R}_{\mathcal{I},\tau}(m,n) \doteq \sum_{j=n}^{M-1+n}\sum_{i=m}^{M-1+m}\mathcal{I}^2(i,j)\tau(i-m,j-n)$$

is a maximum.

$\mathcal{R}_{\mathcal{I},\tau}$ is called the *cross-correlation coefficient* of $\mathcal{I}$ and τ at (m,n).

In general however, it is not true that $\sum_{j=n}^{M-1+n}\sum_{i=m}^{M-1+m}\mathcal{I}^2(i,j)$ is approximately constant for all $(m,n)\in\{0,1,\ldots,N-1\}\times\{0,1,\ldots,N-1\}$. If we define the *normalised* cross-correlation coefficient of $\mathcal{I}$ and τ at each point (m,n) by

$$\mathcal{N}_{\mathcal{I},\tau}(m,n)=\frac{\sum_{j=n}^{M-1+n}\sum_{i=m}^{M-1+m}\mathcal{I}(i,j)\tau(i-m,j-n)}{\left[\sum_{j=n}^{M-1+n}\sum_{i=m}^{M-1+m}\mathcal{I}^2(i,j)\right]^{\frac{1}{2}}},$$

then by the Cauchy-Schwartz inequality,

$$\mathcal{N}_{\mathcal{I},\tau}(m,n)\leq\sum_{j=n}^{M-1+n}\sum_{i=m}^{M-1+m}\tau^2(i,j) \quad ,$$

with equality iff

$$\mathcal{I}(i,j)=k\tau(i-m,j-n) \quad \forall\, m\leq i\leq M-1+m, n\leq j\leq M-1+n$$

for some scalar $k\neq 0$. As a simple example of a template matching problem, consider the $M\times M$ binary image of an "L-shaped" diagram (figure 1) as a template τ.

Given any $N\times N(N>M)$ binary image $\mathcal{I}$, in order to detect the presence (or absence) of the L-shaped object in $\mathcal{I}$, the template τ is translated pixel-by-pixel over the image $\mathcal{I}$, and at each position the correlation coefficient is calculated to check whether it lies within acceptable bounds to declare the presence of the object at that position. Alternatively, the location of the peak value of the correlation coefficients computed over the entire image field $\mathcal{I}$ is taken to be the position of location of the object in the iamge. Clearly, for sufficiently large images $\mathcal{I}$, this method of template matching is computationally intensive. Moreover, scaled and rotated versions of the object in the

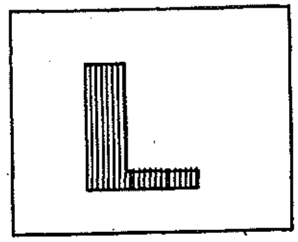

Fig 1: Template of an L shaped object

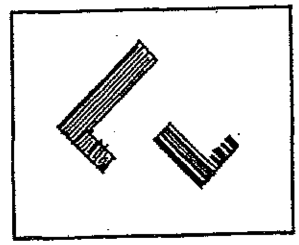

Fig 2: Image with scaled 'L's

template ocurring in the image (figure 2) may go undetected. In order to accomodate variations in scale and rotation of the object being searched in the image with respect to the object in the template, one would have to perform a range of rotation and scale transformations on the template, and each of these transformed versions of the template correlated with the image at each pixel, leading to an explosive search. Nevertheless, the fundamental necessity of template matching for object, shape and feature recognition in the areas of image analysis, image understanding and robotic and computer vision, make it an indispensable tool. Several efficient implementations of variations of correlation template matching and other alternatives exist with varying degrees and aspects of efficiency, and are constantly being put forward. A brief survey of some of the more important and recent advances in this area is described in [3]. Details on the template database, interclass migration and pose-invariant template matching can be found in [3].

2 The object recognition algorithm

Input: $\mathcal{I}$(image of size $N\times N$);T: Template database

1. Decompose $\mathcal{I}$ into an image pyramid (with respect to a wavelet basis) of L levels $\mathcal{I}\rightarrow\{\mathcal{I}_j^H,\mathcal{I}_j^V,\mathcal{I}_j^D,\mathcal{I}_L\}_{j=1}^L$ $\mathcal{I}_o=\oplus_{j=1}^L[\mathcal{I}_j^H\oplus\mathcal{I}_j^V\oplus\mathcal{I}_j^D]\oplus\mathcal{I}_L$. This takes $O(N^2)$ time.
2. Preprocess the low-pass image $\mathcal{I}_L$ at level L by segmenting and identifying the boundary points of various objects present in by a seed-fill algo-

rithm.

3. Compute the convex hulls of the boundary points of the objects; this takes $O(\sum_{i=1}^{k} n_i \log n_i)$ time where n_i is the number of border points of object O_i and k is the number of objects in $\mathcal{I}$. Also $\sum_{i=1}^{k} n_i \ll N^2$, so this computational time is very small compared to the image size.

4. Compute the MERs of the convex hulls obtained from the previous step. This takes $O(\sum_{i=1}^{k} m_i)$ time, where $m_i \leq n_i$ is the number of points in the convex hull of O_i, and again $\sum m_i \ll N^2$.

5. Compute the aspect ratios $\rho_A(O_i)$ and the pose vectors $(l(O_i), c_x(O_i), c_y(O_i), \theta(O_I))$. This takes $O(k)$ time, where k is the number of objects in $\mathcal{I}$.

6. For each object O_i do

   (a) For each template in the aspect ratio class $\rho_A(O_i)$, transform the template τ by scaling it using $l(O_i)$ and by rotating it by $\theta(O_i)$. (note: each template is stored in a L-level pyramid)

   (b) set $j = L$

   (c) Apply the transformed template τ_j' at level j to the location $(c_x^j(O_i), c_y^j(O_i))$ in $\mathcal{I}_j$ and compute the correlation coefficient. If the correlation coefficient is above the threshold t_j for level j, accept the template as a promising template at level j; else, reject it.

   (d) Reconstruct the image and the selected template pyramids by one step from the level j to obtain $\mathcal{I}_{j-1}$ and τ_{j-1} at level $j-1$. Scale τ_{j-1} to obtain τ_{j-1}' by doubling its size with relation to τ_j.

   (e) set j to $j-1$; goto step (c)

7. Output the positions, scale, angle and description of objects corresponding to the unrejected templates.

One can associate many kinds of confidence measures describing the degree of match obtained for the various objects in the image, and there are several ways of defining threshold values at various levels of the pyramidal matching to determine whether to accept or reject a match for comparison at the next higher resolution. The choice of these measures is dependent on the application at hand, and what is acceptable (or not acceptable) for a specific application. For instance, often it may not be useful or necessary to match promising templates at all levels for a given application. For details on the advantages of orthonormal wavelet pyramid decomposition, see [3].

3 Novelties and improvements

The algorithm has been implemented and is shown as follows :
The figures are presented at the end of the text.

Salient features :

- In this method, the number of locations in an image at which template matching is done depends only on the number of objects detected in the image and not on the size of the image.
- This method identifies the templates in the template database T that are likely to find a match in a given image, thus making the selection of templates deterministic and more efficient. This is a significant improvement over existing methods, since these methods do not specify any criteria for the choice or rejection of a template. In real applications the number of templates in T is large, and hence an efficient method of narrowing down the possible choices significantly affects the computational overhead of the matching procedure.
- Apart from reducing the number of choices of candidate templates, this method also identifies the correct scaling and rotational transformations to be performed on a candidate template, as well as the location in the image at which the match should be computed. This completely eliminates the search in the pose space for the right pose of the template. In other methods, the template pose is methodically or randomly searched in the pose space, involving repeated matching of templates with incrementally corrected poses based on previous matches. This wastes a lot of computations on searching for the pose alone. Thus the proposed method minimizes the number of matches to be performed for identification.

Bibliography

1. R.Y. Wong and E.L. Hall, "Sequential Hierarchial Scene Matching", IEEE Trans. Comput., 27, 4, pp. 359-366.

2. L. Prasad, S.S. Iyengar, R.L. Rao and R.L. Kashyap, "Fault-tolerant sensor integration using multiresolution decomposition", Physical Review E, 49, 4, pp. 3452-3461, April 1994.

3. L. Prasad "Multiresolutional Fault-Tolerant Sensor Integration and Object Recognition in Images" Ph.D Dissertation, Louisiana State University, May 1995.

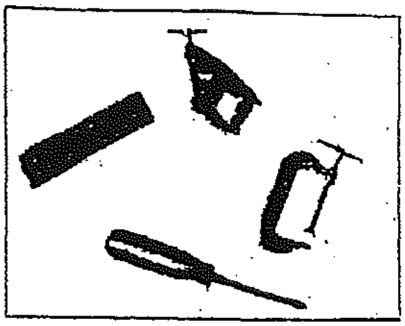

Figure 3: Tool Scene

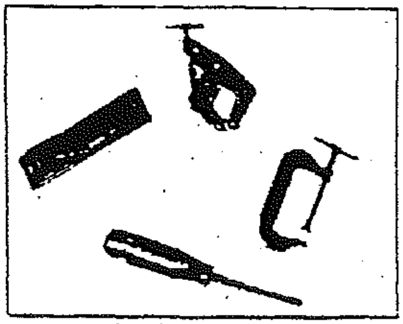

Figure 4: Segmented Image of Tool Scene

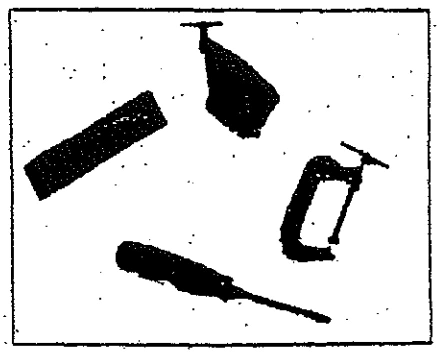

Figure 5: Morphological closure of figure-ground image

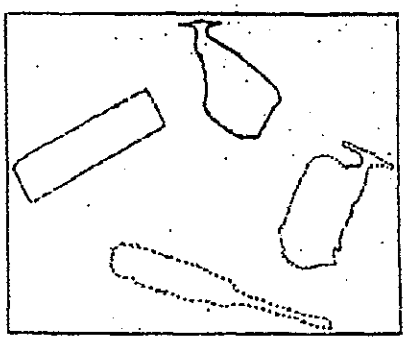

Figure 6: Contour Image from boundary tracing the silhoutte image

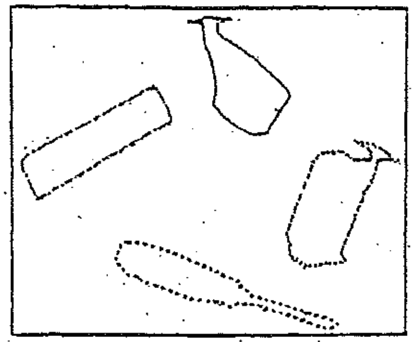

Figure 7: Polygonal approximation of contour image

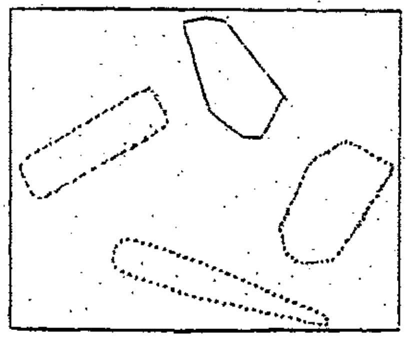

Figure 8: Convex hull image obtained from the polygonal image

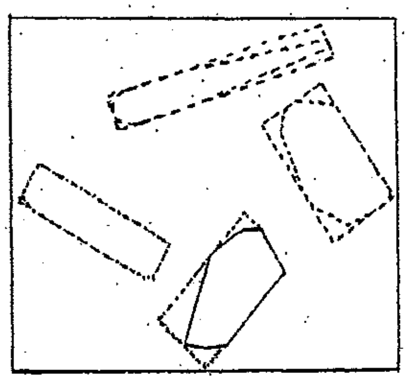

Figure 9: The MER image obtained from the convex hull image