3-2001

Locating target at high speed using image decimation decomposition processing

Zikuan Chen
University of Tennessee - Knoxville

Mohammad Rezaul Karim
University of Tennessee - Knoxville

Majeed M. Hayat
Marquette University, majeed.hayat@marquette.edu

PERGAMON

Pattern Recognition 34 (2001) 685–694

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Locating target at high speed using image decimation decomposition processing

Zikuan Chen[a], Mohammad A. Karim[a,*], Majeed M. Hayat[b]

[a] *Department of Electrical and Computer Engineering, The University of Tennessee, 414 Ferris Hall, Knoxville, TN 37996-2100, USA*
[b] *Electro-Optics Program, The University of Dayton, 300 College Park Ave, Dayton, OH 45469-0245, USA*

Received 17 December 1999; received in revised form 22 December 1999; accepted 31 January 2000

Abstract

We develop a decimation-decomposition processing technique that consists of judiciously selecting certain decimation-decomposed components of an image and then performing inter-component processing. For a (k_x, k_y)-decimation decomposition, there may be up to $k_x k_y$ decimation-decomposed components. The minimal surviving and maximal non-surviving lengths associated with inter-component processing algorithm allows for clutter suppression. By removing detection redundancies, one can locate the target at high speed. A "where-then-what" model is proposed for target tracking and recognition. It locates the target by-image decimation-decomposition processing first and then recognizes the target in question using a suitable image recognition technique.

Keywords: Image decimation; Target location; "Where-then-what" model; Clutter suppression; Target tracking

1. Introduction

When searching or tracking a target, it is desirable to have a wide field of view. Consequently, targets often occupy a relatively small area in the acquired image. On the other hand, high spatial resolution for both image sensor and frame grabber is preferred for successful recognition [1]. Obviously, it is undesirable to identify a target in a lower spatial resolution image. Typically, more accurate recognition of an object is obtained with detailed image. However in target tracking and recognition, it is not necessary to find the target and identify it simultaneously. Since there is less computation need for locating a target in the lower spatial resolution image, one can locate an object in coarser resolution first, and then recognize the object by focusing on the local region in its finer resolution versions. Accordingly, we propose a "where-then-what" model for target tracking and recognition which consists of two steps: (i) locate a candidate target in the scene as an answer to "where is the target"; and, (ii) perform recognition by focusing on the localized area as an answer to "what is the target".

Correlation-based and template matching techniques have been found to be effective in target tracking and recognition; however, they are highly computation-intensive and not suitable for real-time applications [2]. Optical correlation technique [3] may provide alternate solutions but typically the processes need bulky optical systems. In this paper, we develop a novel technique called decimation-decomposition processing for locating the target in a digital image at high speed.

Decimation is a sampling-rate conversion technique that is based on selecting equi-spaced elements in a digital signal to generate its down-sampled representation [4–6]. It is widely used in subband image coding [5], multirate signal processing [6–9], and multiresolution image processing [10–14]. These techniques involve a tree data structure representation at various resolution levels, and the processing pertains primarily to multiresolution linkage from parent-level to children-level or

* Corresponding author. Tel.: + 1-865-974-3461; fax: + 1-423-974-5483.
E-mail address: karim@utk.edu (M.A. Karim).

0031-3203/01/$20.00
PII: S0031-3203(00)00021-2

vice versa. In this paper, instead of pursuing parent–children processing between different levels, we employ an inter-children processing approach at a low spatial resolution level, called image decimation-decomposition processing, for locating target at high speed.

For a Nyquist-sampled image, decimation may result in information loss and aliasing due to undersampling. Since the goal of image detection is different from that of image representation, the computation-intensive low-pass anti-aliasing prefiltering [5] may be avoided as long as the target data survives the image decimation-decomposition process. For a single-object image, the target (corresponding to the largest patch) presence will outlive all other smaller patches with decreasing spatial resolution.

Moreover, since any two points determine a line and any three non-collinear points determine a plane, additional points are considered as redundancies. In digital image processing and recognition, a line or a region consists of more pixels than the minimum required by the geometrical axioms, which means that high redundancies are associated with detecting the presence of a line or region. To detect the presence of a target without an accurate recognition, it seems reasonable to rapidly locate a candidate target position with the least redundancies. We show in this paper that this clever approach reduces the complexity drastically.

The paper is organized as follows. Section 2 introduces decimation decomposition which yields multiple polyphase components, and inter-component processing using simple logic operations. Section 3 provides estimations of line length and target area, and target location in a scene. In Section 4, we show that our technique reduces the computation cost significantly when the detection redundancies are removed. Based on this decimation-decomposition processing, we develop in Section 5 the "where-then-what" pattern recognition model. Finally, in Sections 6 and 7, we provide an illustration for fast locating a target and a summary.

2. Image decimation decomposition

2.1. Definition

Let $f(m,n)$ represents a binary quantization of an $M \times N$ digital image. Assume that the pixel index assignment is in accordance with the left–right and top–down convention. The top-left pixel at $(m,n)=(1,1)$ is treated as the reference or origin pixel. For an input image $f(m,n)$, the decimation with a decimator factor k_x in x direction and a decimator factor k_y in y direction is given by

$$g(p,q;i,j) = f(pk_x + i, qk_y + j), 0 \leqslant p < P, 0 \leqslant q < Q, \quad 1 \leqslant i \leqslant k_x, 1 \leqslant j \leqslant k_y, \tag{1a}$$

where

$$P = \text{ceil}\left(\frac{M}{k_x}\right) + 1, \tag{1b}$$

$$Q = \text{ceil}\left(\frac{N}{k_y}\right) + 1, \tag{1c}$$

ceil (x) denotes the ceiling function which takes the largest integer less than or equal to x, and (p,q) are the indices of the decimation-decomposed image $g(p,q)$ which is represented by a $P \times Q$ matrix in a reduced resolution. The parameters (i,j) are the indices of the top-left pixel of children $g(p,q;i,j)$ inherited from the indices of the same pixel in the top-left corner block (with size $k_x \times k_y$) of the parent $f(m,n)$. There are up to $k_x k_y$ possible children resulting from a (k_x, k_y)-decimation decomposition. In accordance with notations used by Vetterli [5] and Suter [6], each child $g(p,q;i,j)$ is a polyphase representation component of the original parent image. We hereby refer to it as component (i,j) for short. Obviously, each component is a collection of equally-spaced pixels in the original digital image. The element indices (p,q) of component (i,j) are defined in the range: $pk_x + i \leqslant M$, and $qk_y + j \leqslant N$; otherwise, it is possible for some elements to outstretch the original matrix beyond the right and bottom margins. Such marginal problems are overcome by appending zeros as follows:

$$g(p,q;i,j) = \begin{cases} f(pk_x + i, qk_y + j), & 1 \leqslant pk_x + i \leqslant M, 1 \leqslant qk_y + j \leqslant N, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Zero-appending, however, is not required when

$$\text{mod}\,(M, k_x) = k_x - 1, \tag{3a}$$

$$\text{mod}\,(M, k_y) = k_y - 1, \tag{3b}$$

where mod (a,b) is the remainder of a divided by b.

To understand image decimation decomposition, we illustrate a (4, 3)-decimation as shown in Fig. 1. The reference component (1, 1) is the collection of all boxed elements of the original image, component (2, 1) is the collection of elements marked by single strike-through, and component (3, 2) is the collection of elements marked by double strike-through. In accordance to Eq. (2), the right-most two columns in the original image and, correspondingly the last column of components (3, 2) are augmented by zeros (shown in bold).

2.2. Component selection and intra-set processing

A maximum of $k_x k_y$ components are associated with a (k_x, k_y)-decimation decomposition. To reduce computation time, in practice, only some of the components are used. We adopt the set-theoretic notation to simplify the manipulation of selected components. The set of all

Original image f(m,n)

0	~~1~~	2	3	4	~~5~~	6	7	8	~~9~~	10	11	12	~~13~~	14	51	16	~~17~~	18	19	20	21	**0**	**0**
10	11	12	~~13~~	14	15	16	~~17~~	18	19	20	~~21~~	22	23	24	~~25~~	26	27	28	~~29~~	30	31	**0**	~~**0**~~
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	**0**	**0**
30	~~31~~	32	33	34	~~35~~	36	37	38	~~39~~	40	41	42	~~43~~	44	45	46	~~47~~	48	49	50	51	**0**	**0**
40	41	42	~~43~~	44	45	46	~~47~~	48	49	50	~~51~~	52	53	54	~~55~~	56	57	58	~~59~~	60	61	**0**	~~**0**~~
50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	**0**	**0**
60	~~61~~	62	63	64	~~65~~	66	67	68	~~69~~	70	71	72	~~73~~	74	75	76	~~77~~	78	79	80	81	**0**	**0**
70	71	72	~~73~~	74	75	76	~~77~~	78	79	80	~~81~~	82	83	84	~~85~~	86	87	88	~~89~~	90	91	**0**	~~**0**~~

component (1,1)

g(p,q;1,1)

0	4	8	12	16	20
30	34	38	42	46	50
60	64	68	72	76	80

component (2,1)

g(p,q;2,1)

1	5	9	13	17	21
31	35	39	43	47	51
61	65	69	73	77	81

component (3,2).

g(p,q;3,2)

13	17	21	25	29	**0**
43	47	51	55	59	**0**
73	77	81	85	89	**0**

Fig. 1. Illustration of a (4, 3)-decimation decomposition showing three components and their elements.

component labels is denoted by $\Omega = \{(i, j)|\, i = 1, 2, \ldots, k_x, j = 1, 2, \ldots, k_y\}$ whereas a collection of selected components is represented by a subset $\Omega' \subset \Omega$. There are different ways to select components from this universal set of labels. For simplicity, the reference component (1, 1) is included in each such subset as common reference. We can identify a collection of all possible selections by a collection C of subsets of Ω. An admissible of selections C must obey the following rule:

$$\underset{\Omega' \in C}{\mathrm{Y}}\ \Omega' = \Omega, \tag{4}$$

where Y denotes a union. Note that for Ω' and $\Omega'' \in C$, Ω' and Ω'' are not necessarily disjoint.

There are a variety of algorithms that can perform inter-component processing within the set. This is referred to as intra-set processing. The point-to-point inter-frame processing requires the operands to have the same frame size as given by Eq. (2). In addition, an inter-set postprocessing which is based on the intra-set results may also be used to yield desirable results. For simplicity, we consider herein only the case of single-object binary image. We show in that the target can be located at high speed by simple logic operations for two types of inter-component processing: the intra-set processing and inter-set processing. The intra-set inter-component processing is associated with only one set while the inter-set processing involves multiple sets.

An inter-component logical AND operation for a subset $\Omega_s \subset \Omega$, is defined by

$$G(p, q) = \bigwedge_{(i,\ j) \in \Omega_s} g(p, q; i, j), \tag{5}$$

where $\wedge$ denotes AND logic. Fig. 2 shows an example of the inter-component processing. Decimation decomposition is followed by intra-set processing and then by inter-set postprocessing.

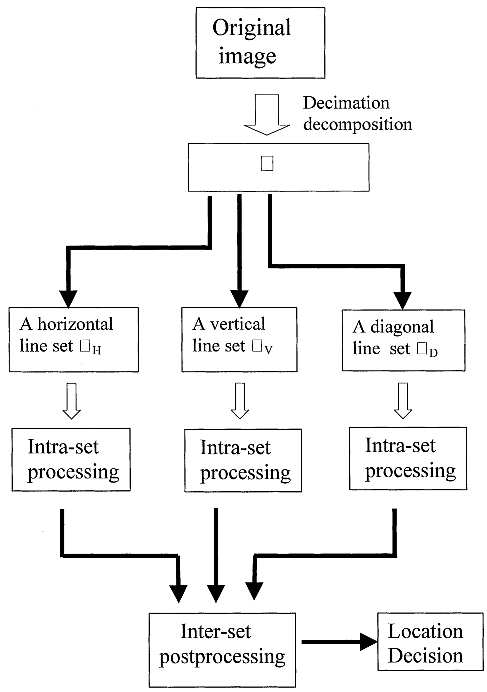

Fig. 2. An inter-component processing where intra-set processing is followed by inter-set postprocessing.

In the appendix, we list some pertinent inter-component processing algorithms suitable for detecting the presence of a line. If a line is detected by more than two points, we say the line detection has redundancy. Accordingly, detecting a line by two points and detecting a rectangular or triangular region by three non-collinear points are said to have no redundancy. In practice, more points are preferred for improving accuracy and robustness. The redundancy-reduced detection strategy provides an efficient way to detect a target's presence. This strategy is realized by decimation-decomposition processing. There are more degrees of freedom to select components from the plenary for redundancy-reduced detection. In the case of (k_x, k_y)-decimation decomposition processing, it is possible to detect a horizontal, vertical and diagonal lines, respectively, with set $\Omega_h = \{(1,1),(k_{x1},1)\}$, $\Omega_v = \{(1,1),(1,k_{y1})\}$, and $\Omega_d = \{(1,1),(k,k)\}$ such that $k_{x1} \leqslant k_x$, $k_{y1} \leqslant k_y$, and $k \leqslant \min(k_x, k_y)$. The corresponding inter-component processing algorithm are given by

$$h(p,q) = \bigwedge_{(i,j)\in\Omega_h} g(p,q;i,j) = g(p,q;1,1) \wedge g(p,q;,k_{x1},1), \tag{6a}$$

$$v(p,q) = \bigwedge_{(i,j)\in\Omega_v} g(p,q;i,j) = g(p,q;1,1) \wedge g(p,q;1,k_{y1}), \tag{6b}$$

$$d(p,q) = \bigwedge_{(i,j)\in\Omega_d} g(p,q;i,j) = g(p,q;1,1) \wedge g(p,q;,k,k). \tag{6c}$$

Since there are no redundancies in Eqs. (6a)–(6c), the algorithms are optimally efficient in detecting lines. The redundancies for the detection of line are graphically shown in Fig. 3 for the cases of (a) full redundancy as given in the appendix, by Eqs. (A.1)–(A.3); (b) reduced redundancies; and (c) no redundancy using Eqs. (6a)–(6c). All other component set selection strategy and inter-component processing algorithm lies in between the two extremes.

2.3. Inter-set postprocessing

We have pointed out already that horizontal, vertical, and diagonal lines can be detected by inter-component processing with a selected component set. It is possible to further combine intra-set processing results together for improving effectiveness. The general combination algorithm for an admissible components collection C is given by

$$\hat{G}(p,q) = \left[\bigwedge_{(i,j)\in\Omega_1} g(p,q;i,j)\right]\Theta_1 \left[\bigwedge_{(i,j)\in\Omega_2} g(p,q;i,j)\right]\Theta_2 \wedge\left[\wedge_{(i,j)\in\Omega_s} g(p,q;i,j)\right]\Theta_s \wedge, \tag{7}$$

where for each $1 \leqslant s \leqslant \text{card}\{C\}$, (card$\{C\}$ represents cardinality of the set C, which is equal to the number of its elements), and Ω_s denotes an inter-set operation. Also note that for $1 \leqslant s \leqslant \text{card}\{C\}$, $\Omega_s \in C$. For example, $\Theta_s = \wedge$ and $\Theta_s = \vee$ refer, respectively, to logical AND and OR operators. The $\Theta_s' s$ can be the same or different. It is understood in Eq. (7) that the intra-set operations are performed prior to the inter-set operations.

It is possible to detect a convex region by subjecting the results of line detection to inter-set postprocessing. If the two-dimensional region of interest is of diamond shape, for example, then one may choose two diagonal lines to detect the region. Herein we provide the corresponding inter-set postprocessing algorithms as follows:

$$\hat{G}_1(p,q) = H(p,q)\,\Theta_1\,V(p,q)\,\Theta_2\,D(p,q) \tag{8}$$

and

$$\hat{G}_2(p,q) = h(p,q)\,\Theta_1\,v(p,q)\,\Theta_2\,d(p,q). \tag{9}$$

In some cases, effectiveness can be improved by a mixed processing algorithm such as

$$\hat{G}_3(p,q) = H(p,q)\,\Theta_1\,v(p,q)\,\Theta_2\,d(p,q), \tag{10}$$

where the horizontal line detection has full redundancy and the vertical and diagonal line detection has no redundancy for (k_x, k_y)-decimation decomposition processing. The results indicate that more considerations are given to the horizontal direction. When designing the inter-set postprocessing algorithms, it is instructive to keep in mind that the logical AND operation provides high discrimination but is sensitive to salt-and-pepper noise, and that the OR logic operation contributes to robustness [15].

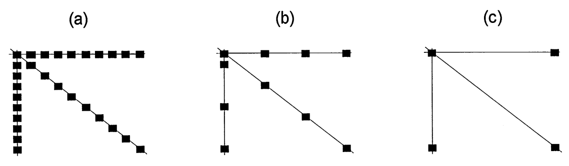

Fig. 3. Line detection with (a) full redundancy, (b) reduced redundancy, and (c) no redundancy.

3. Target size and location estimation

Due to the logical AND operations in Eqs. (A.1)–(A.3) and Eqs. (6a)–(6c), one-dimensional decimation-decomposition processing is associated with a minimal length that survives inter-component processing. This is referred to as the minimal surviving length. Meanwhile, there is a maximal length that could be missed and is referred to as the maximal non-surviving length. The minimal surviving and maximal non-surviving lengths

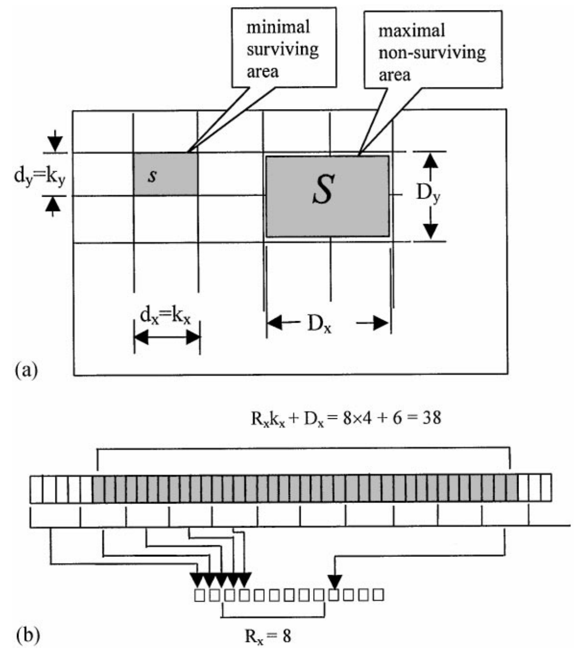

Fig. 4. Illustrations of (a) minimal surviving length (area) and maximal non-surviving length (area) as defined in Eqs. (11a), (11b), (14a), (14b) and (15); and (b) implementation of Eq. (13a) and (13b).

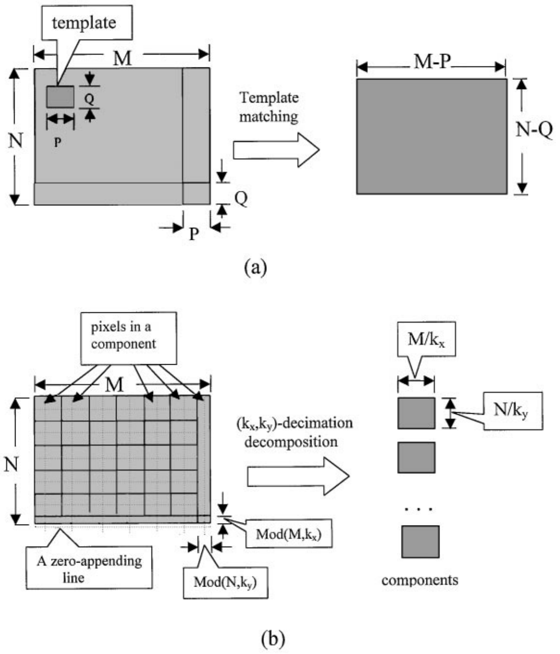

Fig. 5. Digital image reduction using (a) template matching, and (b) decimation decomposition.

are associated with the (k_x, k_y)-decimation decomposition processing and are

$$d_x = k_x, \quad D_x = 2(k_x - 1), \tag{11a}$$

$$d_y = k_y, \quad D_y = 2(k_y - 1), \tag{11b}$$

where d_x, and d_y respectively, denote the minimal surviving length and D_x, and D_y denote maximal non-surviving length along x- and y-directions, as shown in Fig. 4(a).

For the case of no-redundancy, the minimal surviving and maximal non-surviving lengths follow the relationships

$$d_{x1} = k_{x1} < k_x, \quad D_{x1} = D_x, \tag{12a}$$

$$d_{y1} = k_{y1} < k_y, \quad D_{y1} = D_y. \tag{12b}$$

With the minimal surviving and maximal non-surviving lengths, the estimate for line length along x and y, i.e., L_x and L_y, can be obtained:

$$R_x k_x \leqslant L_x \leqslant R_x k_x + D_x, \tag{13a}$$

$$R_y k_y \leqslant L_y \leqslant R_y k_y + D_y, \tag{13b}$$

where R_x and R_y, respectively, represent the run lengths of non-zero pixels in an inter-component-processed image (at a low resolution), along x- and a y-directed lines. The uncertainty of the estimate is given by the maximal non-surviving length. Fig. 4(b) shows an example pertaining to Eq. (13a) where $R_x = 8$, $k_x = 4$, and $D_x = 6$ for decimation decomposition processing as given in Eq. (A.1). Correspondingly, the minimal surviving area s and the maximal non-surviving area S associated with the (k_x, k_y)-decimation decomposition-processing shown in Fig. 5(a) are given by

$$s = k_x k_y = d_x d_y, \tag{14a}$$

$$S = 2(k_x - 1)2(k_y - 1) = D_x D_y. \tag{14b}$$

Since a plane is specified uniquely by two non-collinear lines, one can detect a two-dimensional region by a combination of two one-dimensional line detection algorithms. Accordingly, the area of a region of interest is estimated by combining two line estimates given in Eqs. (13a) and (13b) wherein the uncertainty $D_x D_y$ is given by

$$R_x k_x R_y k_y \leqslant a \leqslant R_x k_x R_y k_y + D_x D_y. \tag{15}$$

According to Eq. (1a), the relationship between the pixel indices (m, n) and (p, q) associated with (k_x, k_y)-decimation decomposition is given by

$$m = pk_x, \tag{16a}$$

$$n = qk_y. \tag{16b}$$

Using Eqs. (13a) and (13b), one can estimate the window size for location identification by

$$w_x = L_x + \Delta w_x, \tag{17a}$$

$$w_y = L_y + \Delta w_y, \tag{17b}$$

where $w_x \times w_y$ denotes the window size with border allowances Δw_x along x and Δw_y along y. It is safe to consider border allowances for segmenting the target region in the original image by

$$\Delta w_x \geqslant D_x, \tag{18a}$$

$$\Delta w_y \geqslant D_y. \tag{18b}$$

4. Computation cost

To appreciate the computational saving in the image decimation decomposition processing, it is instructive to make a comparison with the conventional template matching technique. For an $M \times N$ binary image $f(m, n)$ and a $P \times Q$ template $t(p, q)$, the template matching is given by

$$f_1(m_1, n_1) = \sum_{p=1}^{P} \sum_{q=1}^{Q} f(m_1 + p, n_1 + q)t(p, q),$$

$$m_1 = 0, 1, \wedge, M - P, n_1 = 0, 1, \wedge, N - Q, \tag{19}$$

where the number of multiplication and addition operations are, respectively, given by

$$C_\times = (M - P)(N - Q)PQ, \tag{20a}$$

$$C_+ = (M - P)(N - Q)PQ. \tag{20b}$$

As shown in Fig. 5, the template matching yields an $(M - P) \times (N - Q)$ matrix with a reduction in matrix order resulting from the exclusion of the right and bottom margins in the template. To find the maximum element a loop with $C_>$ comparisons is required, where

$$C_> = (M - P)(N - Q). \tag{21}$$

The decimation decomposition is realized by data selection manipulation that only involves accessing data from the memory. The corresponding time cost is negligible. It corresponds to the time for point-to-point inter-component processing and locating the maximum within a component-size image.

For an $M \times N$ binary image, the (k_x, k_y)-decimation decomposition produces components with a reduced matrix order $M/k_x \times N/k_y$ and, the number of point-to-point inter-component AND operations is given by

$$C_\wedge = \frac{MN}{k_x k_y}. \tag{22}$$

In comparison, the multiplication-related computation cost is considerably small. In particular, the search area associated with decimation technique is smaller than that for the brute-force template matching by a factor of $1/(k_x k_y)$. Table 1 lists the computation costs for both template matching and decimation decomposition processing. The algorithms of Eqs. (6a) and (6b) and Eqs. (A.1) and (A.2), respectively, represents the no-redundancy and full-redundancy cases dealing with both horizontal and vertical lines. The proposed image-decimation-decomposition processing cuts down computation requirements significantly, especially in the cases of redundancy-reduced detection. Strictly speaking, the computation costs listed in Table 1 account only for the detection of target. Additional computation cost should be considered to account for the recognition of target following the detection of target location.

5. "Where-then-what" target recognition model

We propose a "where-then-what" target recognition model as shown in Fig. 6 where a CCD camera grabs the scene and produces a digital image, namely, the original image. The "where-then-what" model is characterized by the following steps:

Step 1: The "where firmware" estimates the target location in a low spatial resolution image using image-decimation decomposition processing. While accessing the original gray image, a binarization is employed to generate its binary version. The "where" firmware is implemented by either software, hardware, or both. The firmware generates parameters such as target location indices (m, n), horizontal width L_x, and vertical height L_y to locate the candidate target.

Step 2: The "what recognizer" identifies the target by focusing on a subimage, in full-spatial resolution, at the location of target already pointed out by the "where firmware."

Table 1
Cost comparisons

	Multiplication #	Addition #	Comparisons #
Template matching	$C_\times = (M - P)(N - Q)PQ$	$C_+ = (M - P)(N - Q)PQ$	$C_> = (M - P)(N - Q)$
Algorithm of Eqs. (A.1) and (A.2)	$C = MN(k_x + k_y)/(k_y k_y)$	$C_+ = 0$	$C_> = MN/(k_x k_y)$
Algorithm of Eqs. (6a) and (6b)	$C = 2MN/(k_y k_y)$	$C_+ = 0$	$C_> = MN/(k_x k_y)$

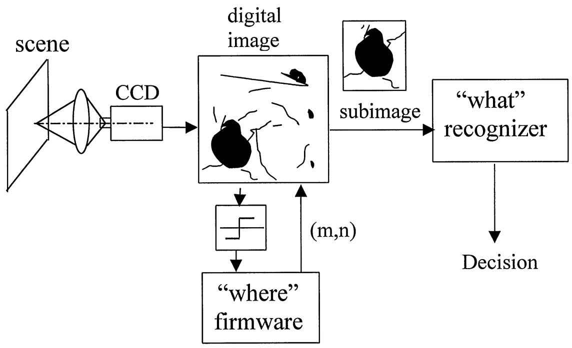

Fig. 6. "Where-then-what" model for tracking target.

As identified in Section 2, there are different implementation algorithms for (k_x, k_y)-decimation decomposition processing. These involve judiciously selecting components and designing inter-component processing. For adaptive applications, the process becomes sophisticated since the decimation factors k_x and k_y and inter-component processing algorithms may be determined dynamically. Further, it is possible to develop the decimation-decomposition technique for multistage decimation processing by adopting a pyramid transformation [16].

6. Simulation

To exhibit the effectiveness of image-decimation-decomposition processing, we consider two examples. Figs. 7–9 show the results pertaining to vertical line detection. Fig. 9 shows the original image which is a 64×64 binary image. For (6, 6)-decimation decomposition, we select a set $\Omega_s = \{(1,1),(2,1),(3,1),(5,1)\}$ as shown in Fig. 8 (a1–a4). By employing inter-component processing algorithms given by

$$v_1(p,q) = g(p,q;1,1) \wedge g(p,q;2,1) \wedge g(p,q;3,1) \wedge g(p,q;5,1), \quad (23a)$$

$$v_2(p,q) = g(p,q;1,1) \wedge g(p,q;5,1), \quad (23b)$$

we obtain the results shown in Figs. 8(b) and (c), respectively. Note that Eq. (23a) has redundancies while Eq. (23b) has no redundancy. In this case, both results are the same. Fig. 9 illustrates the vertical line detection result pertaining to Eqs. (23a) and (23b), in full spatial resolution. All clutters smaller than the minimal surviving length are suppressed as well as a horizontal line is suppressed as a clutter.

Figs. 10–13 illustrate the results of a more comprehensive decimation decomposition processing. In Fig. 10, the

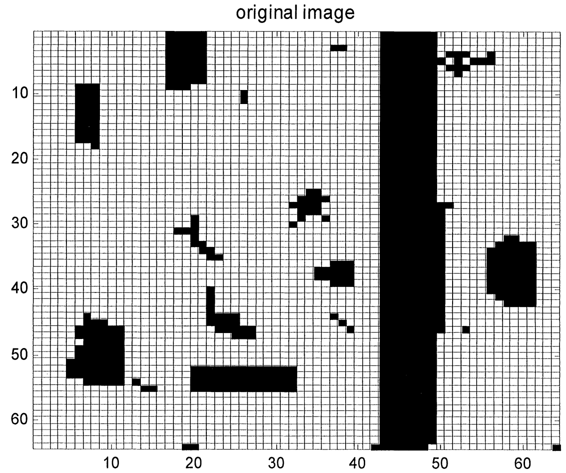

Fig. 7. A candidate original image for the detection of vertical line.

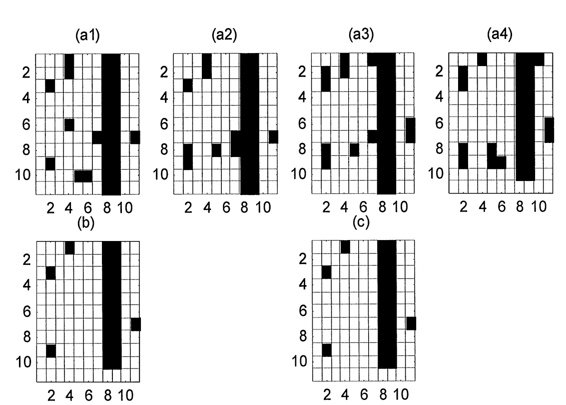

Fig. 8. (a1)–(a4) Examples of selected components associated with (6, 6)-decimation decomposition of the original shown in Fig. 7; (b) the result of inter-component processing (using Eq. (23a)); and (c) the result of inter-component processing (using Eq. (23b)).

target plane is present in the scene along with clutter. Fig. 11 shows examples of selected components associated with (6, 6)-decimation decomposition. The results shown in Fig. 12 are obtained by implementing the following inter-component processing algorithms:

$$G_{a1}(p,q) = g(p,q;1,1) \wedge g(p,q;2,1) \wedge g(p,q;3,1) \wedge g(p,q;5,1), \quad (24a)$$

$$G_{a2}(p,q) = g(p,q;1,1) \wedge g(p,q;5,1), \quad (24b)$$

$$G_{b1}(p,q) = g(p,q;3,1) \wedge g(p,q;3,2)\ AND\ g(p,q;3,4) \wedge g(p,q;3,5), \quad (24c)$$

$$G_{b2}(p,q) = g(p,q;3,1) \wedge g(p,q;5,1), \quad (24d)$$

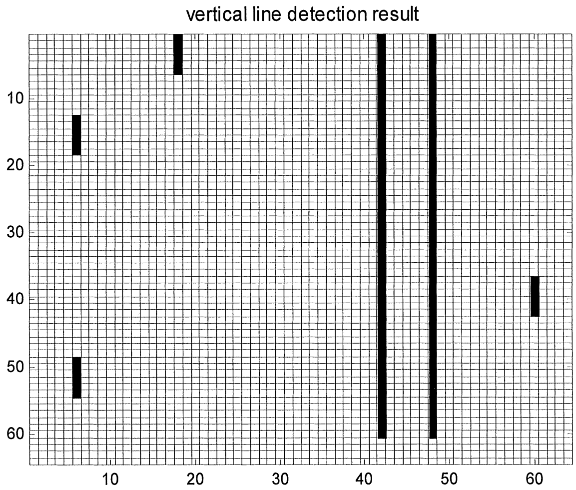

Fig. 9. Results corresponding to Figs. 8(b) or (c) in the same resolution as original image.

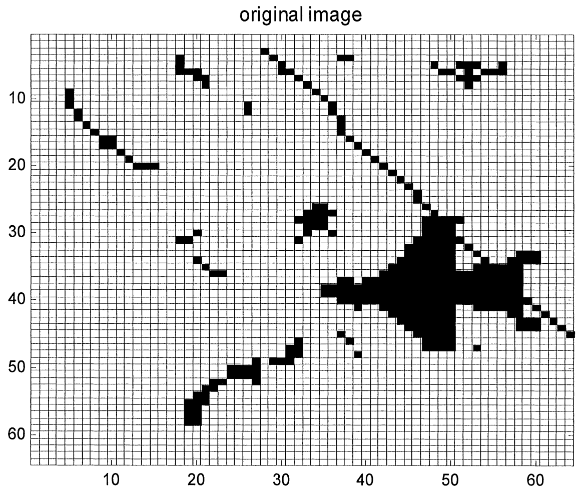

Fig. 10. An image for a comprehensive study of decimation-decomposition processing.

$$G_{d1}(p,q) = g(p,q;2,2) \wedge g(p,q;3,3)\ AND\ g(p,q;4,4) \wedge g(p,q;5,5), \tag{24e}$$

$$G_{d2}(p,q) = g(p,q;2,2) \wedge g(p,q;5,5), \tag{24f}$$

$$G_{s1}(p,q) = G_{a1}(p,q) \wedge G_{b1}(p,q), \tag{24g}$$

$$G_{s2}(p,q) = G_{a2}(p,q) \wedge G_{b2}(p,q), \tag{24h}$$

Figs. 12(a1), (b1) and (c1) show line detection results with redundancies as defined by Eqs. (24a), (24c) and (24e), respectively. Figs. 12(a2), (b2), and (c2) show alternative

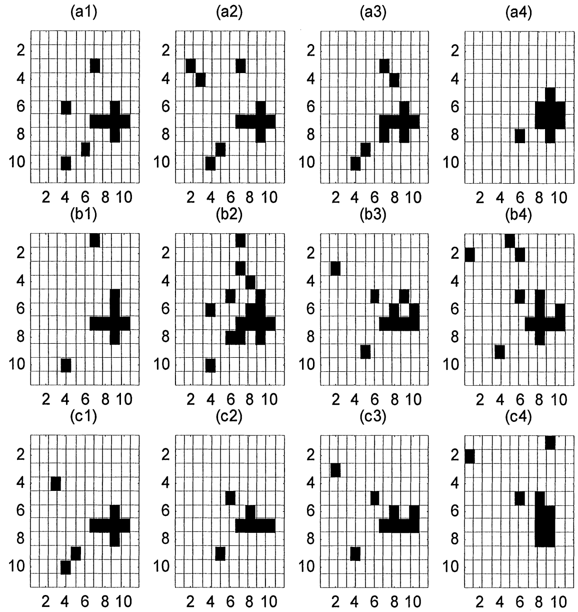

Fig. 11. Representative components associated with (6, 6)-decimation decomposition of the original image shown in Fig. 10.

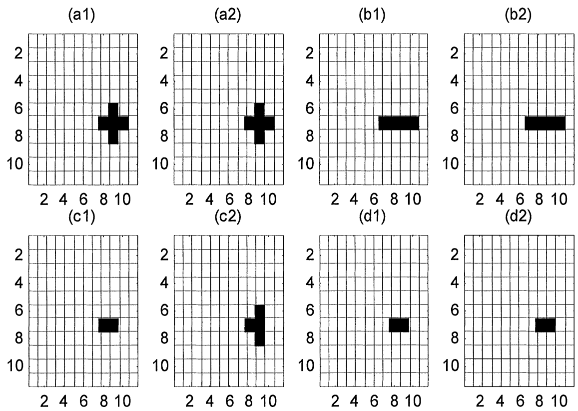

Fig. 12. Results generated by the inter-component processing using Eq. (24a)–(24h) of the image of Fig. 11.

results with no redundancy as defined by Eqs. (24b), (24d) and (24f). Figs. 12(d1) and (d2) show the region detection results as obtained using Eqs. (24g) and (24h). It is seen that the no-redundancy case, as in Eq. (24h), produces the same result as that of the redundancy case (as in Eq. (24g)). The results are the same again. A good and robust algorithm often leads to more cases of identical results. Fig. 13 shows the localized region for $(\Delta w_x, \Delta w_y) = (3k_x, 3k_y)$ with the top-left allowance of $(2k_x, 2k_y)$ and the bottom-right allowance of (k_x, k_y).

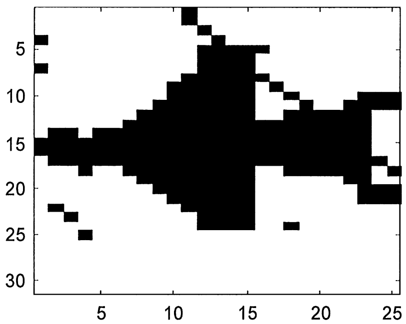

Fig. 13. Subimage location using Eq. (17a) and (17b) with $(\Delta w_x, \Delta w_y) = (3k_x, 3k_y)$.

7. Summary

The proposed image-decimation-decomposition processing exploits interframe processing among the decimation-decomposed subimages (components) thus rendering clutter suppression with reduced computation cost. The process consists in resampling down a digital image without low-pass filtering as required otherwise in the conventional decimation processing [5]. The algorithm involves selecting decimation-decomposed components and then rendering inter-component processing. The component selection could have full redundancy, reduced redundancy, or no redundancy depending on the number of the selected components. The inter-component processing includes logical AND, OR, and their combination.

The proposed method can be used to detect the presence of a target without anti-aliasing prefiltering. In the extreme case of decimation-decomposition processing, the detection of presence of a line or a region is realized with minimum computation in the case of no redundancy. By using decimated data, removing detection redundancy, and executing simple logic inter-component operations, a target can be located at high speed. The uncertainty associated with the target estimates can be overcome by designing ad hoc inter-component processing algorithms.

Although the technique is immune to stationary noise (clutter), it is susceptible to spot noise as in the multiplication or AND operations. To remove spot noise, thus, it will be necessary to preprocess the binary image using hole filling and morphology-based techniques [17]. Sophisticated algorithms involving logical AND can also be used to combat this noise sensitivity.

Sophisticated algorithms are needed to reduce the maximal non-surviving length and, thus, improve accuracy and robustness. To be more general and complete, the sensor detection theory based on statistics and sensor fusion [18,19] may be considered. Multistage decimation processing can provide for more efficient solutions. To develop a general "where-then-what" target tracking and recognition system, the decimation factors (k_x, k_y) should be adaptively determined. This may be developed in the same way as the pyramid transformation [16] for multistage decimation processing.

Appendix A

Selected inter-component processing algorithms for line detection (with full-redundancy) are listed below.

(1) *Horizontal line detection*:

$$H(p,q) = \bigwedge_{(i,j)\in\Omega_H} g(p,q;i,j) = g(p,q;1,1) \wedge g(p,q;2,1) \wedge g(p,q;k_x,1), \tag{A.1}$$

where the set Ω_H is given by $\{(i,j)|\, i=1,2,\ldots,k_x, j=1\}$. This algorithm involves an 1-D k_x-decimation-decomposition processing. It is equivalent to moving a non-overlapping window of size equal to the decimation factor k_x.

(2) *Vertical line detection*:

$$V(p,q) = \bigwedge_{(i,j)\in\Omega_V} g(p,q;i,j) = g(p,q;1,1) \wedge g(p,q;1,2) \Lambda \wedge g(p,q;1,k_y), \tag{A.2}$$

where the set Ω_V is given by $\{(i,j)|\, i=1, j=1,2,\ldots,k_y\}$.

(3) *Diagonal line detection*:

$$D(p,q) = \bigwedge_{(i,j)\in\Omega_D} g(p,q;i,j) = g(p,q;1,1) \wedge g(p,q;2,2) \Lambda \wedge g(p,q;\min(k_x,k_y),\min(k_x,k_y)), \tag{A.3}$$

where the set Ω_D is given by $\{(i,j)|\, i=j=1, 2,\ldots,\min(k_x,k_y)\}$, and $\min(a,b)$ denotes the minimum of a and b.

Eqs. (A.1)–(A.3) are special cases of the general formula given by Eq. (5) with $\Omega_s=\Omega_H$, $\Omega_s=\Omega_V$, and $\Omega_s=\Omega_D$, respectively. They are obtained by a judicious selection of the components.

References

[1] Y. Bar-Shalom, T.E. Fortman, Tracking and Data Association, Academic Press, New York, 1988.

[2] R.M. Haralick, L.G. Shapiro, Computer and Robot Vision, Addison-Wesley Publishing Company, Reading, MA, 1992, pp. 157–253.

[3] M.A. Karim, Electro-Optical Devices and Systems, PWS-Kent Publishers, Boston, MA, 1990.

[4] R.E. Crochiere, Interpolation and decimation, in: Digital Signal Processing Committee (Ed.), Programs for Digital Signal Processing, IEEE Press, New York, 1979.

[5] M. Vetterli, Multirate filter banks for subband coding, in: J.W. Woods (Ed.), Subband Image Coding, Kluwer Academic Publishers, Dordrecht, 1991, pp. 43–100.
[6] B.W. Suter, in: Multirate and Wavelet Signal Processing, Academic Press, New York, 1998, pp. 1–143.
[7] C.R. Dyer, Multirate image understanding, in: L. Uhr (Ed.), Computer Vision, Academic Press, New York, 1987, pp. 171–213.
[8] V. Cantoni, L. Cinque, C. Guerra, S. Levialdi, L. Lombardi, 2-D object recognition by multiscale tree matching, Pattern Recognition 31 (1998) 1443–1454.
[9] F. Satter, L. Floreby, G. Salomonson, B. Lovstrom, Image enhancement based on a nonlinear multiscale method, IEEE Trans. Image Process. 6 (1997) 888–895.
[10] P.J. Burt, The pyramid as a structure for efficient computation, in: A. Rosenfeld (Ed.), Multiresolution Image Processing and Analysis, Springer, Berlin, 1984, pp. 6–35.
[11] S.H. Yoon, J.H. Kim, W.E. Alexander, S.M. Park, K.H. Sohn, An optimum solution for scale-invariant object recognition based on the multiresolution approximation, Pattern Recognition 31 (1998) 889–908.
[12] V. Concepcion, H. Wechsler, Detection and localization of objects in time-varying imaging using attention, representation and memory pyramids, Pattern Recognition 29 (1996) 1543–1557.
[13] S.G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Trans. Pattern Anal. Mach. Intell. 11 (1989) 674–693.
[14] G. Bongiovanni, L. Cinque, S. Levialdi, A. Rosenfeld, Image segmentation by a multiresolution approach, Pattern Recognition 26 (1993) 1845–1854.
[15] Z. Chen, G. Mu, Pattern identification by combination of Foley-Sammon transform and synthetic discriminant function, in: International Conference on Neural Network and Signal Processing, Vol. 1, Nanjing, China, 1995, pp. 659–662.
[16] S.L. Tanimoto, From pixels to predicates in pyramid machine, in: J.C. Simon (Ed.), From Pixel to Features, North-Holland, Amsterdam, 1989, pp. 384–392.
[17] R.M. Haralick, L.G. Shapiro, in: Computer and Robot Vision, Addison-Wesley, Reading, MA, 1992, pp. 157–253.
[18] D.H. Johnson, D.E. Dudgeon, Array Signal Processing: Concepts and Techniques, PTR Prentice-Hall, Englewood Cliffs, NJ, 1993, pp. 424–467.
[19] J.R. Maheshkumar, New computational technique for complementary sensor integration in detection-localization systems, Opt. Eng. 35 (1996) 674–684.

About the Author—ZIKUAN CHEN received his MS in physics from Yunnan University in 1988 and his Dr. Eng. in optical information processing from Nankai University in 1993. He is an associate professor with the Institute of Modern Optics, Nankai University and had been a postdoctoral fellow at both the University of Dayton and the University of Tennessee. Dr. Chen is currently a research associate at the University of Arkansas, Fayetteville. His research interests include hybrid optical/digital processing, imaging process, pattern recognition, and machine vision applications in agricultural engineering.

About the Author—MOHAMMAD A. KARIM is Professor and Head of the Electrical and Computer Engineering Department at the University of Tennessee. He received his BS in physics in 1976 from the University of Dacca, Bangladesh, and MS degrees in both physics and electrical engineering, and a Ph.D. in electrical engineering from the University of Alabama, respectively, in 1978, 1979, and 1981. He is active in research in the areas of information processing, pattern recognition, optical computing, displays, and EO systems. Dr. Karim has authored 4 textbooks, 8 book chapters and over 260 papers. He is the North American Editor of *Optics and Laser Technology* (United Kingdom), an Associate Editor of the *IEEE Transactions on Education* and serves on the Editorial board of *Microwave and Optical Technology Letters*. He has served as invited guest editor for nine journal special issues. He is a Fellow of both the Optical Society of America (OSA) and Society of Photo-Instrumentation Engineers (SPIE), a senior member of IEEE, and a member of American Society of Engineering Education (ASEE).

About the Author—MAJEED M. HAYAT was born in Kuwait in 1963. He received his B.S. degree (summa cum laude) in 1985 in Electrical Engineering from the University of the Pacific, Stockton, CA. He received his M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Wisconsin-Madison in 1988 and 1992, respectively. From 1993 to 1996, he worked at the University of Wisconsin-Madison as the co-principal investigator of a project on statistical modeling and detection of minefields, which was funded by the Office of Naval Research. Since 1996, he has been an assistant professor of Electro-Optics and Electrical and Computer Engineering at the University of Dayton, Dayton, Ohio. His research interests include noise in optoelectronic devices, statistical communication theory, optical and digital communications, image and signal processing, quantum imaging, photon statistics of non-classical light, and applied point processes. Dr. Hayat is a recipient of the 1997 National Science Foundation CAREER Award. He is a member of IEEE, OSA, and SPIE.