We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Small Object Recognition Techniques Based on Structured Template Matching for High-Resolution Satellite Images

Toshio Modegi[1], Tomoaki Inazawa[2], Tsugio Chiba[2] and Chiaki Kobayashi[3]
[1]Media Technology Research Center, Dai Nippon Printing Co., Ltd.
[2]Infoserve Inc.
[3]Earth Remote Sensing Data Analysis Center
Japan

1. Introduction

We are developing infrastructure tools of wide-area monitoring for disaster damaged areas or traffic conditions, using earth observation satellite images. In these days, resolution of optical sensors installed in earth observation satellites has been highly improved. In case of the panchromatic image captured by the QuickBird (DigitalGlobe, 2008), the ground-level resolution is about 0.6 [m], which makes possible to recognize each automobile on roads or parking lots.

The previous satellite image analysis works have been mainly focused on area segmentation and classification problems (Giovanni Poggi et al., 2005), and object recognition targets have been limited to large objects such as traffic roads or buildings (Qi-Ming Qin et al., 2005), using high-resolution panchromatic satellite images. Whereas, there have been a lot of works on recognizing automobiles in aerial images including the paper (Tao Zhao et al., 2001), however, there have been almost any works trying to recognize such small objects as automobiles in satellite images excluding (So Hee Jeon et al., 2005). This previous work (So Hee Jeon et al., 2005) has been applying template matching methods to recognizing small objects but its recognition rate has been very poor because of insufficient pixel information for pattern matching.

In the previous paper (Modegi T., 2008), we proposed an interactive high-precision template matching tool for satellite images, but it took amount of calculation times and object searching area was limited and far from practical uses. In order to overcome this problem, we apply a structured identification approach similar to the work (Qu Jishuang et al., 2003). In this paper, we propose a three-layered structured template matching method, which enables recognizing small objects such as automobiles in satellite images at very little calculation load. The first layer is focusing on extracting candidate areas, which have metallic-reflection optical characteristics, where any types of transportation objects are included. The second layer is identification and removal of excessively extracted candidate areas such as roads and buildings, which have the similar optical characteristics but do not include our targets. The third layer is identifying each automobile object within the focusing area using our proposing conceptual templates (Modegi T., 2008), which are learned

patterns based on user's operations, based on our improved high-speed template-matching algorithm. The following sections describe specific algorithms of each layer.
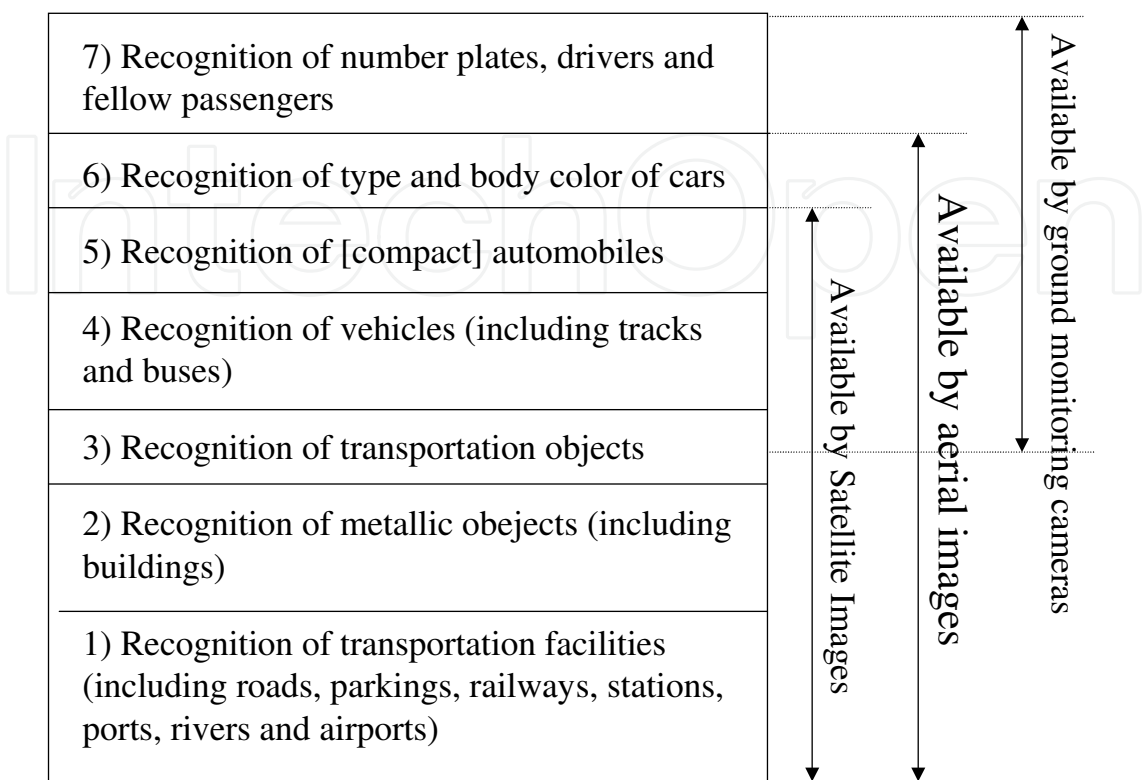

Fig. 1. 7-layered structured recognition model for automobiles on the ground.

2. Proposing structured template matching method

Figure 1 shows our proposing structured recognition model for automobiles on the ground (Modegi T., 2009), which resembles 7-layered communication protocols called as OSI (Open System Interconnection) designed by the ISO (International Standard Organization). The highest recognition level has been already operated by the Japanese police, known as "N-System", by installing a lot of monitoring cameras along highways. The current available high-resolution earth observation satellites cover up to the fifth level in Fig.1.

Figure 2 shows our proposing structured template matching method for recognizing small objects in satellite images. The first matching is called as a micro-template matching, and it extracts candidate areas thoroughly including target small objects by non-linear heuristic filtering with micro-block templates. This candidate means pixel blocks indicating metallic reflection characteristics, which any types of transport objects including automobiles have in common. This process can be made at very little calculation load and also decrease the following third matching calculation load.

The second matching is called as a clustered micro-template matching, and it removes excessively matched relatively large candidate areas, which do not include target small objects with multiple 8-neighbored connected micro-block templates called as a clustered micro-template. This process also can be made at very little calculation load and decrease more the following third matching calculation load.

(1) Given image

(3) Filtered Segmented areas

Segmentation of candidate regions including targets by abstract matching with micro templates (Fast Processing)

Filtering out of large area patterns from the segmented areas with clustered micro-templates. (Fast Processing)

Identification of target objects in the candidate regions by specific matching with macro templates (Slow Processing)

(2) Segmented areas

[macro-templates]   (4) Identified objects

[micro-templates]
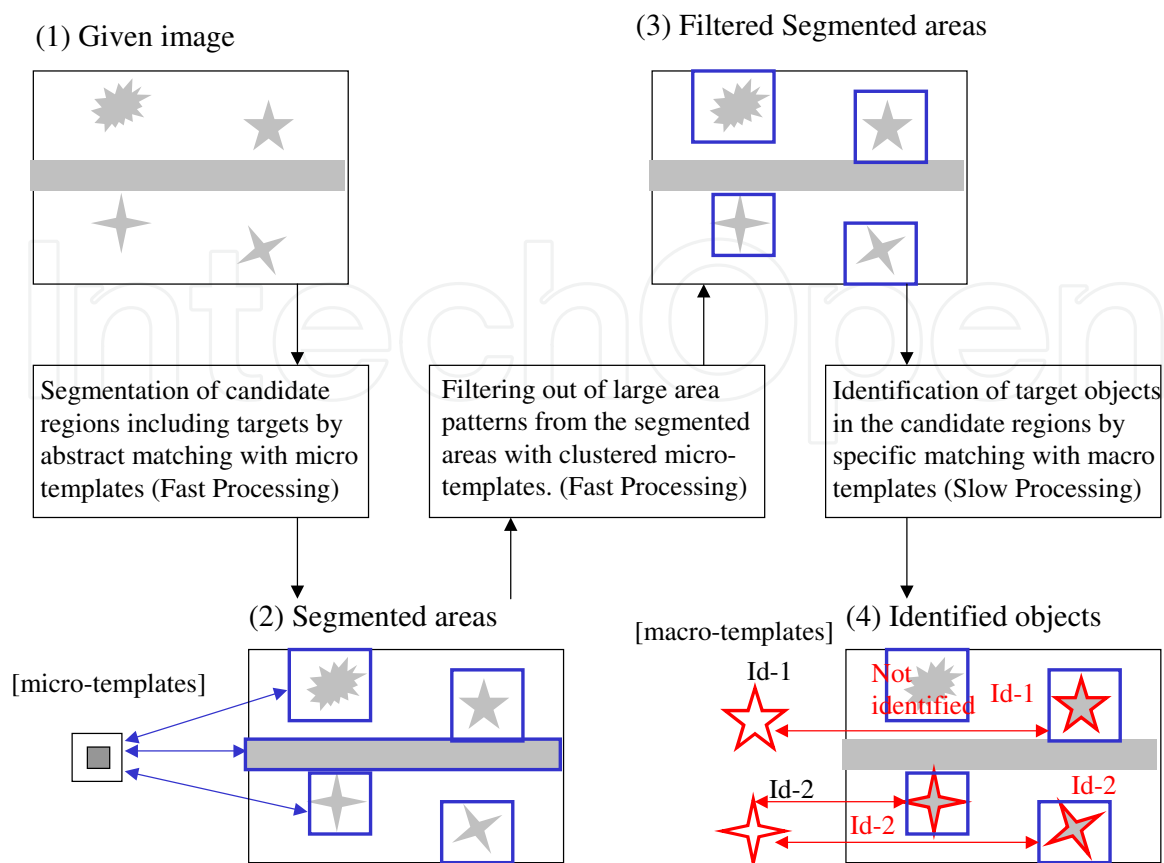
Id-1

Not identified   Id-1

Id-2

Id-2   Id-2

Fig. 2. Concept of structured template matching method for recognizing small objects in satellite images.

The third matching is called as a macro-template matching whose size is almost the same as the target object, and it identifies each object in the segmented candidate areas by the pixel-value correlation based pattern matching shown in the paper (Modegi T., 2008). This process needs a lot of calculation times but its calculation areas will be shrunken by the first and second matching processes.

3. Algorithms of proposing template matching methods

3.1 Micro-template matching

The Figure 3 shows a concept of micro-template matching, which defines binary mask value $M(x,y)$ for given 256 gray-scale image $I(x,y)$ ($0 \leq x \leq N_x$-1, $0 \leq y \leq N_y$-1). This process determines whether optical reflection values of each pixel block have metallic characteristics or not, by a heuristic filtering process. In other words, we determine each tiny area as shown in Fig.3 would be a part of transportation materials which we are searching. As this determination logic, we can use our following described heuristic rules defined between some pixels in the micro-template.

In case of 4×4 pixel micro-template shown in Fig. 3, we separate 2×2 inside pixels V_{ik} (k=1,4) from the other 12 outside pixels V_{ok} (k=1,12). Using these pixel values, we calculate the following 7 statistical parameters: the minimum value of outside pixels V_{omin}, the maximum value of outside pixels V_{omax}, the minimum value of inside pixels V_{imin}, the maximum value of inside pixels V_{imax}, the average value of outside pixels V_{oave}, the average value of inside pixels V_{iave}, the standard deviation value of outside pixels V_{dir}.
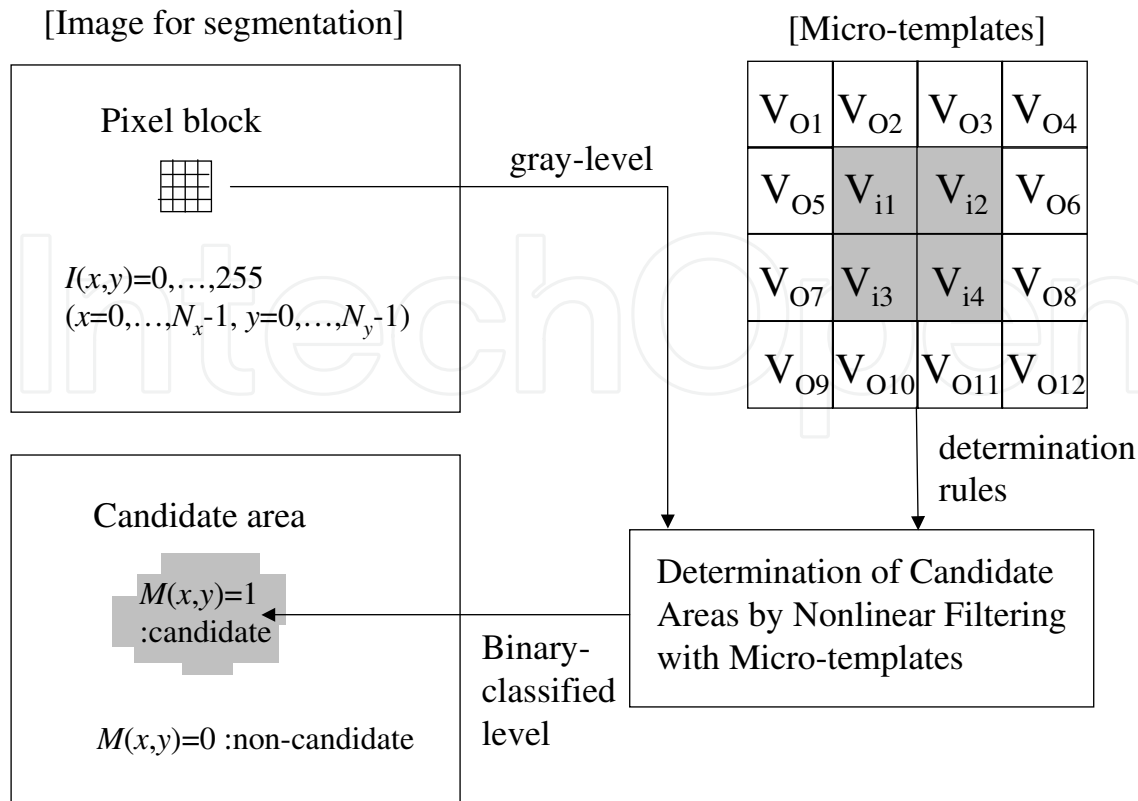
[Image for segmentation]   [Micro-templates]

Pixel block

$I(x,y)=0,\ldots,255$
$(x=0,\ldots,N_x\text{-}1, y=0,\ldots,N_y\text{-}1)$

gray-level

V_{O1}	V_{O2}	V_{O3}	V_{O4}
V_{O5}	V_{i1}	V_{i2}	V_{O6}
V_{O7}	V_{i3}	V_{i4}	V_{O8}
V_{O9}	V_{O10}	V_{O11}	V_{O12}

determination rules

Candidate area

$M(x,y)=1$
:candidate

$M(x,y)=0$:non-candidate

Binary-classified level

Determination of Candidate Areas by Nonlinear Filtering with Micro-templates

Fig. 3. Concept of micro-template matching for extracting candidate areas.

$$V_{omin}= Min_{k=1}^{12} [V_{ok}] ,$$
$$V_{omax}= Max_{k=1}^{12} [V_{ok}] ,$$
$$V_{imin}= Min_{k=1}^{4} [V_{ik}] ,$$
$$V_{imax}= Max_{k=1}^{4} [V_{ik}] ,$$
$$V_{oave}=(\Sigma_{k=1}^{12} V_{ok})/12 ,$$
$$V_{iave}=(\Sigma_{k=1}^{4} V_{ik})/4 ,$$
$$V_{dir}=\{ \Sigma_{k=1}^{12} (V_{ok}- V_{oave})^2/12 \}^{1/2} .$$

(1)

We apply the above template to its nearest 4×4 pixel block values of each pixel of given satellite panchromatic image $I(x,y)$, and determine it would be included in candidate areas or not by the following rules. We have to consider both two kinds of candidate patterns, one is the inside part is brighter than the outside and the other is its negative pattern. In order to determine some pixel (x,y) included in candidate $M(x,y)=1$, the following 5 conditions should be satisfied using the 7 predetermined slice levels: S_{aoi}, S_{doi}, S_{omin}, S_{omax}, S_{imin}, S_{imax}, and S_{dir}.

These 7 kinds of slice levels can be defined interactively by indicating areas where target objects are definitely included on the image displayed screen. For each pixel in our indicated areas, we calculate the 7 statistical parameters based on the equation (1), and using either the minimum or maximum statistical parameters, we can define each of the slice levels as the following.

$$1) \ | V_{oave} - V_{iave} \ | > S_{aoi}.$$
$$(2) \ V_{omax} - V_{imin} > S_{doi}, \ if \ V_{omax} - V_{imin} > V_{imax} - V_{omin}.$$
$$Or, \ V_{imax} - V_{omin} > S_{doi}, \ if \ V_{imax} - V_{omin} > V_{omax} - V_{imin}.$$
$$3) \ V_{oave} > S_{omin} \ and \ V_{oave} < S_{omax}. \qquad (2)$$
$$4) \ V_{iave} < S_{imin}, \ if \ V_{oave} > V_{iave}.$$
$$Or, \ V_{iave} > S_{imax}, \ if \ V_{iave} > V_{oave}.$$
$$5) \ V_{dir} > S_{dir}.$$

$$1) \ S_{aoi} = MIN[\ | V_{oave} - V_{iave} \ |] \bullet 0.9.$$
$$2) \ S_{doi} = MIN[S_1, S_2] \bullet 0.9.$$
$$S_1 = V_{omax} - V_{imin}, \ if \ V_{omax} - V_{imin} > V_{imax} - V_{omin}.$$
$$S_2 = V_{imax} - V_{omin}, \ if \ V_{imax} - V_{omin} > V_{omax} - V_{imin}.$$
$$3) \ S_{omin} = MIN[V_{oave}] \bullet 0.9, \qquad (3)$$
$$S_{omax} = MAX[V_{oave}] \bullet 1.1.$$
$$4) \ S_{imin} = MAX[V_{iave}] \bullet 1.1, \ if \ V_{oave} > V_{iave}.$$
$$S_{imax} = MIN[V_{iave}] \bullet 0.9, \ if \ V_{iave} > V_{oave}.$$
$$5) \ S_{dir} = MIN[V_{dir}] \bullet 0.9.$$

In case of the pixel value of given monochrome image is between 0 and 255, we give typically these slice levels as S_{aoi}=15, S_{doi}=80, S_{omin}=100, S_{omax}=160, S_{imin}=35, S_{imax}=245, and S_{dir}=10 .

3.2 Clustered micro-template matching

In the first micro-template matching, all of metallic reflection characteristic areas are selected as candidates, but these characteristics are not limited to transportation materials. In general, edge parts of buildings and roads are the same characteristics and selected as candidates. Ironically these non-transport areas are larger than our target transport objects, and increase searching load of the next object identification processes. Therefore, in this section we provide an identifying and removing process of excessively selected candidate areas.

In general, incorrectly selected candidate areas such as edge parts of buildings and roads are long slender shape, which can be detected as multiple 8-neighbor connected micro-template blocks called as a clustered micro-template. However, some large areas such as parking lots may include target objects. Therefore, we have to distinguish these patterns from correctly selected large areas where multiple target objects are located closely, with their pixel value characteristics in the detected clustered area.

Figure 4 shows an algorithm of recognizing these incorrectly selected areas to be removed.

Fig.4-(1) shows 8×8 pixel parts of selected candidate areas, where painted pixels are determined as candidates based on the equation (2). On this image, we will search long candidate pixel clusters, whose height or width is larger than S_N-pixel length. In order to find these clusters, we will track 8-neighbored connected candidate pixels from the top-left pixel shown in Fig.4-(2) and Fig.4-(3). Supposing the previous candidate pixel $[i, j]$, in the horizontal direction we will find next candidate pixel from the three pixels $[i+1, j-1]$, $[i+1, j]$ and $[i+1, j+1]$ as shown in Fig.4-(2). Similarly, in the vertical direction we will find next candidate pixel from the three pixels $[i-1, j+1]$, $[i, j+1]$ and $[i+1, j+1]$ as shown in Fig.4-(3).
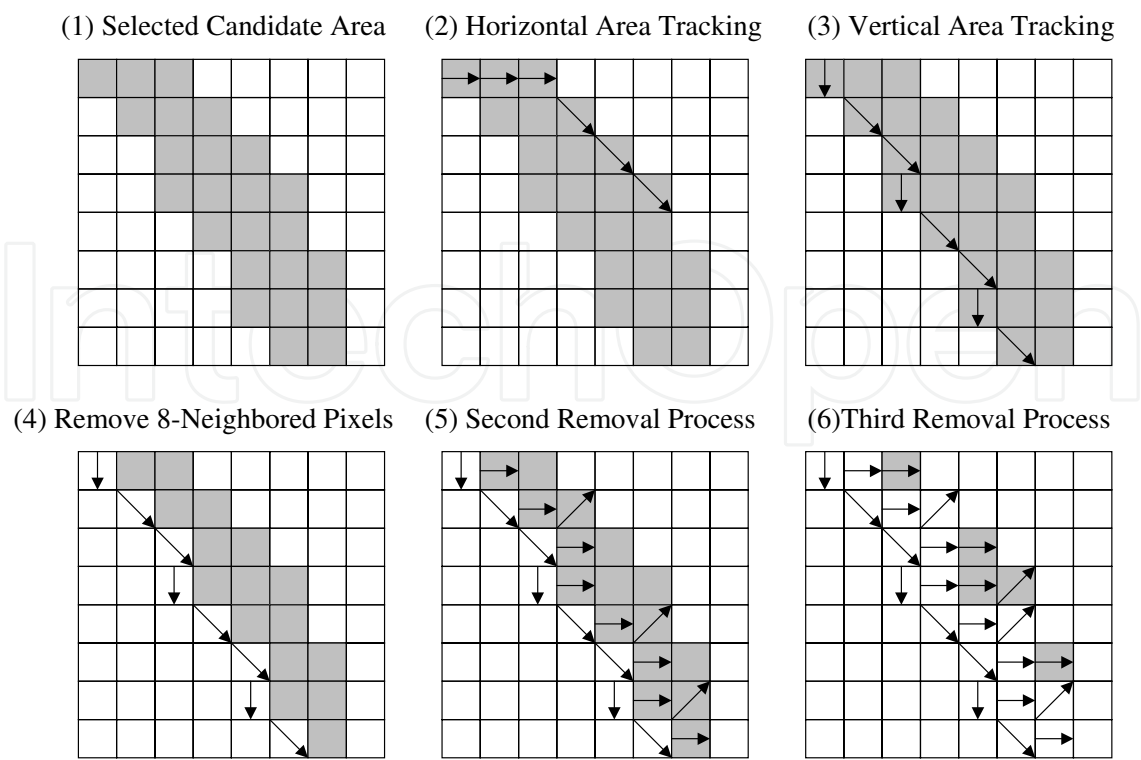
(1) Selected Candidate Area   (2) Horizontal Area Tracking   (3) Vertical Area Tracking

(4) Remove 8-Neighbored Pixels   (5) Second Removal Process   (6)Third Removal Process

Fig. 4. Algorithm of clustered micro-template matching for removing excessively selected large candidate areas.

(1) Source 256-level Image   (2) Two-classified Binaty Mask
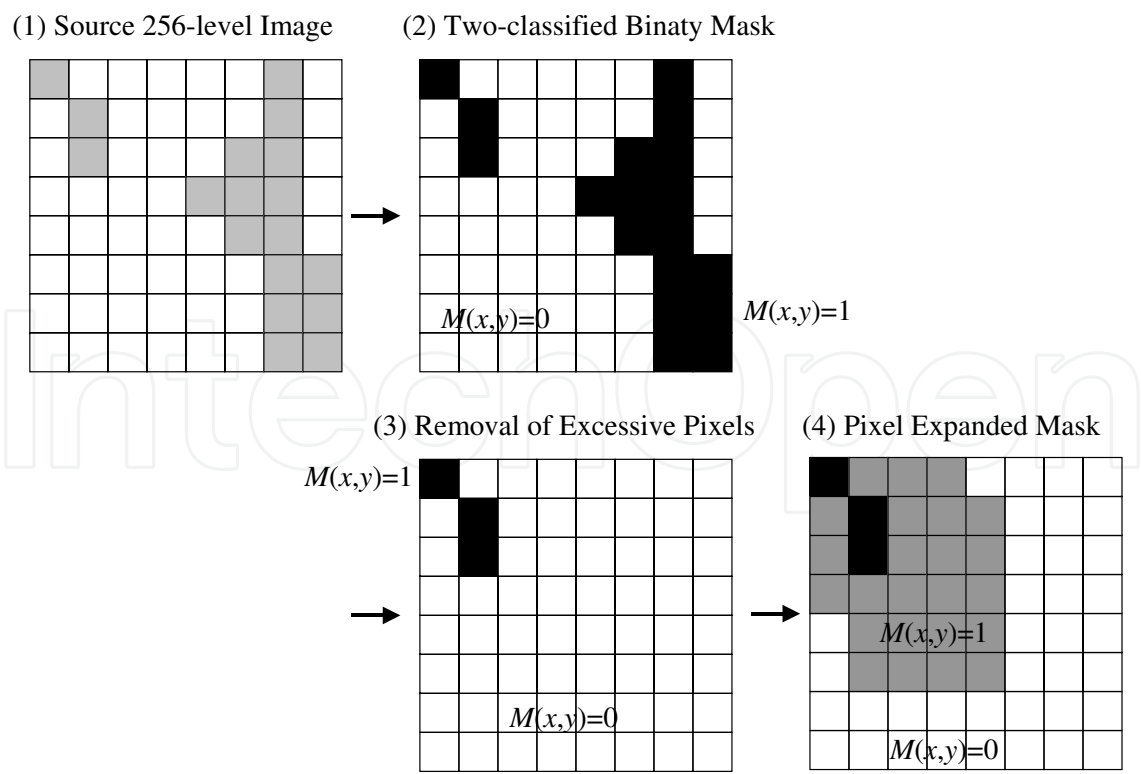
$M(x,y)=0$ $M(x,y)=1$

(3) Removal of Excessive Pixels   (4) Pixel Expanded Mask

$M(x,y)=1$

$M(x,y)=1$

$M(x,y)=0$ $M(x,y)=0$

Fig. 5. Example of both micro-template matching and clustered micro-template matching processes.

In the either horizontal or vertical direction, if we can find successfully a S_N-pixel length cluster, we will calculate the minimum, maximum and average pixel value of this cluster as C_{min}, C_{max} and C_{ave}. Defining two slice levels as S_{cave} and S_{cmax}, if the following conditions are satisfied, we will extend the cluster by finding another 8-neighbored connected candidate pixels around previously found clustered pixels. Typically, we give to these slice levels as S_N =13, S_{cave}=50 and S_{cmax}=50, in case of the pixel value of given monochrome image is between 0 and 255.

$$C_{ave} > S_{cave} \text{ and } C_{max} - C_{min} > S_{cmax} . \tag{4}$$

Then we will reset all of tracked pixels in the extended cluster to non-candidate pixels as $M(x,y)$=0, whereas each of left candidate pixels $[x,y]$ will be extended to 4×4 candidate pixel block as $M(x+i,y+j)$=1 for $0 \le i \le 3$ and $0 \le j \le 3$.

Figure 4-(3) shows in the vertical direction we have found a 8-pixel length cluster (S_N =8), then we will reset all of tracked pixels in the cluster to non-candidate pixels as shown in Fig.4-(4). Furthermore, we extend removing areas around the removed cluster. Fig.4-(5) shows its second extended removed cluster, and Fig.4-(6) shows its third extended removed cluster.

Figure 5 shows a series of both micro-template matching and clustered micro-template matching processes. In Fig.5-(2), we found a small 3-pixel area and a large 15-pixel area with a micro-template. Then the larger area has been removed as $M(x,y)$=0 with a clustered micro-template as shown in Fig.5-(3). Finally, we extend each of left 3 candidate pixels to 4×4 candidate pixel block. Therefore, the size of final candidate area $M(x,y)$=1 becomes 27-pixel size.
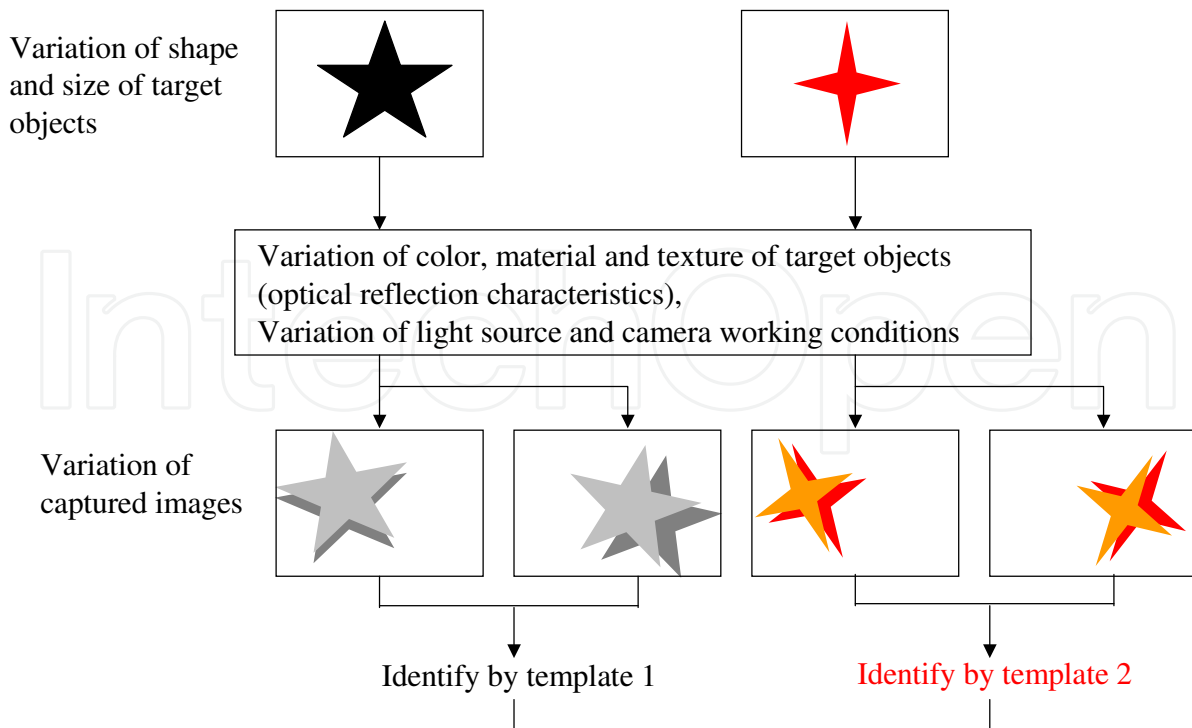
3.3 Macro-template matching

As a final process, we identify each target object within the selected candidate areas using macro-template whose size is almost the same as that of searching objects. In order to execute this process efficiently, we propose using conceptual templates.

Figure 6 shows a concept of our proposing macro-template matching. The upper two objects have different shape, size and color each other. If these two objects are captured by a camera with some light sources, we can obtain tremendous kinds of images including 4 images shown in the middle part of Fig.6. In order to identify these objects, we have to prepare amounts of templates, at least two kinds of templates in this example. Our proposing macro-template matching makes possible identify the objects on these various kinds of captured images with small amounts of templates called as conceptual templates.

Our proposing macro-template matching process consists of two kinds of matching processes, the angle-independent and angle-dependent processes, based on the previous work (Modegi T., 2008). The first angle-independent process is mainly comparing a gray-level histogram of each determining block with that of a template. If the first matching process is successful, the second angle dependent process will be made. This is mainly comparing a normalized correlation coefficient of each determining block with those of several angle rotated image blocks of a template. If one of rotated image is fitted, the determining block will be identified as that template pattern.

Figure 7 shows how to define templates in our proposing macro-template matching algorithm. A template image $I_t(a,x,y)$ is originally an extracted block of pixels in some sample image, which is not necessarily this working image for search $I(x,y)$. Then this image

We have to prepare amounts of templates meeting with these variations.

Robust template matching enables identifying with a little kinds of templates.

Fig. 6. Concept of our proposing template matching using conceptual templates.

is rotated to 8 kinds of angle for angle-dependent matching, and 8 kinds of template images are defined. In each defined $N \times N$ pixel template $I_t(a,x,y)$, two kinds of quadangle-shape outlines are defined for making mask image data $M_t(a,x,y)$. The inner outlines indicate the actual outline pattern of a target and the nearest patterns outside these inner outlines will be considered for another closely located object identification. The common area of the inner areas in all of angles shown the bottom of Fig.7 is used for the angle-independent template matching as the following algorithm (b) and (c), whereas the outer outlined area is used for the angle-dependent template matching as the following algorithm (d) and (e).

The following describes a specific algorithm of our proposing macro-template matching.

a. Check the inside pixels of inner mask are candidate. If $D_{can} \geq N^2/2$, proceed to the next.

$$D_{can} = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} M(x+X,y+Y) \bullet M_t(0,x,y)/3 . \tag{5}$$

b. Calculate a 16-step brightness histogram value difference D_{his} between the angle-0 template $H_t(v)$ ($v=0,..,15$) and its corresponding working image $H(v)$, based on $I_t(0,x,y)/16$ and $I(x+X,y+Y)/16$, where $M_t(0,x,y) \geq 3$, $x=0,…,N-1$ and $y=0,…,N-1$.

$$D_{his} = 1000 \bullet \sum_{v=0}^{15} |H(v)-H_t(v)| / \sum_{v=0}^{15} \{H(v)+H_t(v)\} . \tag{6}$$

c. Calculate a dispersion value difference D_{dis} between the angle-0 template and its corresponding working image, where $M_t(0,x,y) \geq 3$.
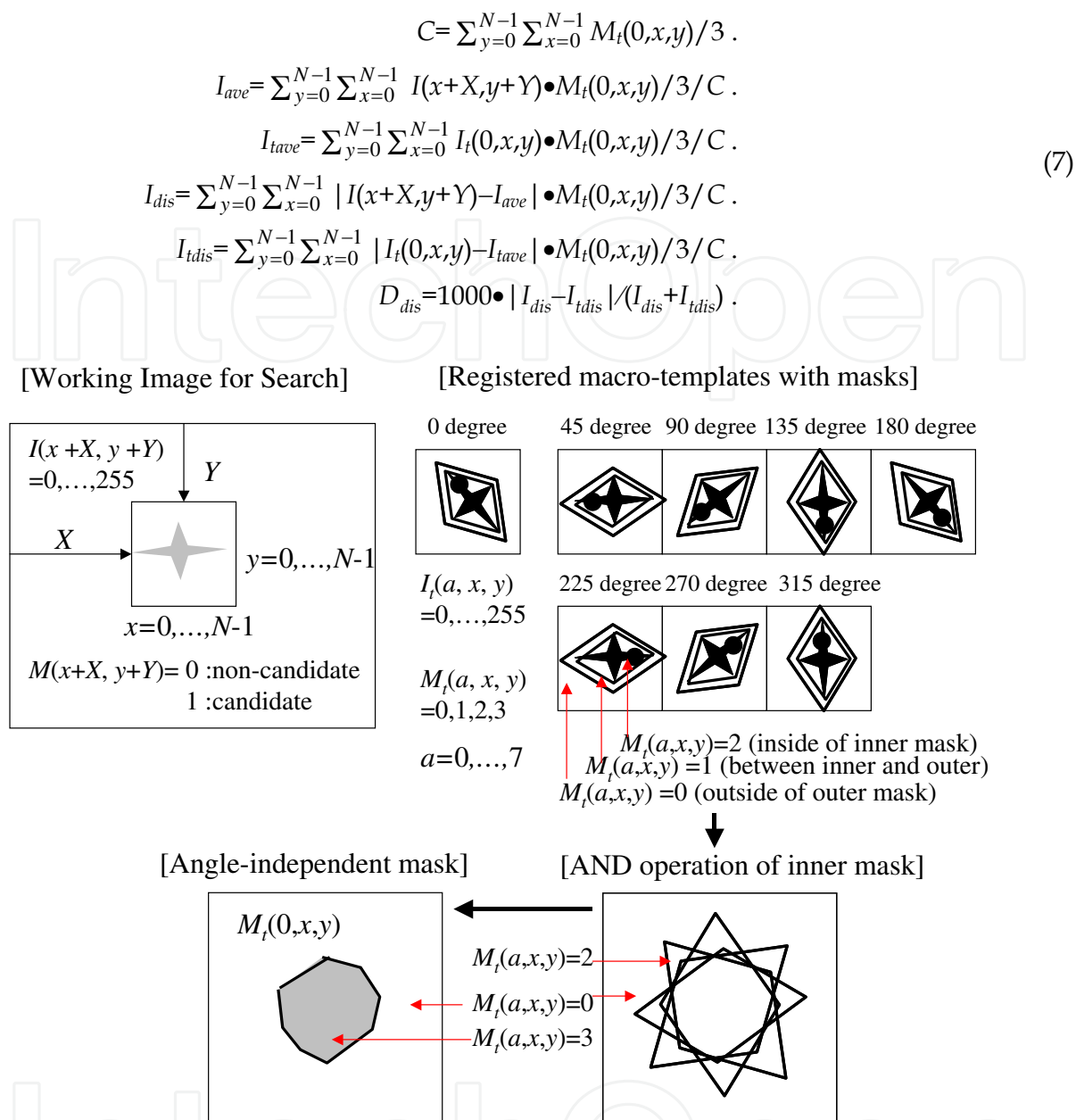
$$C = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} M_t(0,x,y)/3 \ .$$

$$I_{ave} = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} I(x+X,y+Y) \bullet M_t(0,x,y)/3/C \ .$$

$$I_{tave} = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} I_t(0,x,y) \bullet M_t(0,x,y)/3/C \ .$$

$$I_{dis} = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} |I(x+X,y+Y) - I_{ave}| \bullet M_t(0,x,y)/3/C \ .$$

$$I_{tdis} = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} |I_t(0,x,y) - I_{tave}| \bullet M_t(0,x,y)/3/C \ .$$

$$D_{dis} = 1000 \bullet |I_{dis} - I_{tdis}| / (I_{dis} + I_{tdis}) \ .$$

(7)

[Working Image for Search]   [Registered macro-templates with masks]

Fig. 7. Template definitions of our proposing macro-template matching algorithm.

d. Calculate a pixel value difference summation $D_{sub}(a)$ between all angles ($a=0,\ldots,7$) of template and its corresponding working image, where $M_t(a,x,y) \geq 1$.

$$D_{sub}(a) = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} |I(x+X,y+Y) - I_t(a,x,y)|$$
$$\bullet M_t(a,x,y) \qquad (8)$$
$$/ \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} \{I(x+X,y+Y) + I_t(a,x,y)\} \bullet M_t(a,x,y).$$

e. Calculate a normalized correlation coefficient value $D_{cor}(a)$ between all angles ($a=0,\ldots,7$) of the template and its corresponding working image, where $M_t(a,x,y) \geq 1$. Determine the fitted angle a_{max} which makes the value of $D_{cor}(a_{max})$ be the maximum.

$$C = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} M_t(a,x,y) \ .$$

$$I_{ave}(a) = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} I(x+X,y+Y) \bullet M_t(a,x,y)/C \ .$$

$$I_{tave}(a) = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} I_t(a,x,y) \bullet M_t(a,x,y) / C \ .$$

$$D_{cor}(a) = 1000 \bullet \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} \{I(x+X,y+Y) - I_{ave}(a)\}$$

$$\bullet \{I_t(a,x,y) - I_{tave}(a)\} \bullet M_t(a,x,y)$$

$$/ [\sum_{y=0}^{N-1} \sum_{x=0}^{N-1} \{I(x+X,y+Y) - I_{ave}(a)\}^2 \bullet M_t(a,x,y)$$

$$\bullet \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} \{I_t(a,x,y) - I_{tave}(a)\}^2 \bullet M_t(a,x,y)]^{1/2} \ .$$

(9)

Figure 8 shows a construction of total macro-template matching processes for identifying each small object included in candidate areas. The first and second processes are based on our proposing template matching processes, which calculate 4 kinds of matching parameters D_{his}, D_{dis}, $D_{sub}(a_{max})$ and $D_{cor}(a_{max})$ with the fitted angle parameter a_{max} for some position. The first process (1) finds a matching position (X_c, Y_c) where all of matching parameters are satisfied with the predefined matching conditions as $D_{his} \leq S_{his}$, $D_{dis} \leq S_{dis}$, $D_{sub}(a_{max}) \leq S_{sub}$ and $D_{cor}(a_{max}) \geq S_{cor}$. The second process (2) corrects the matching position to the
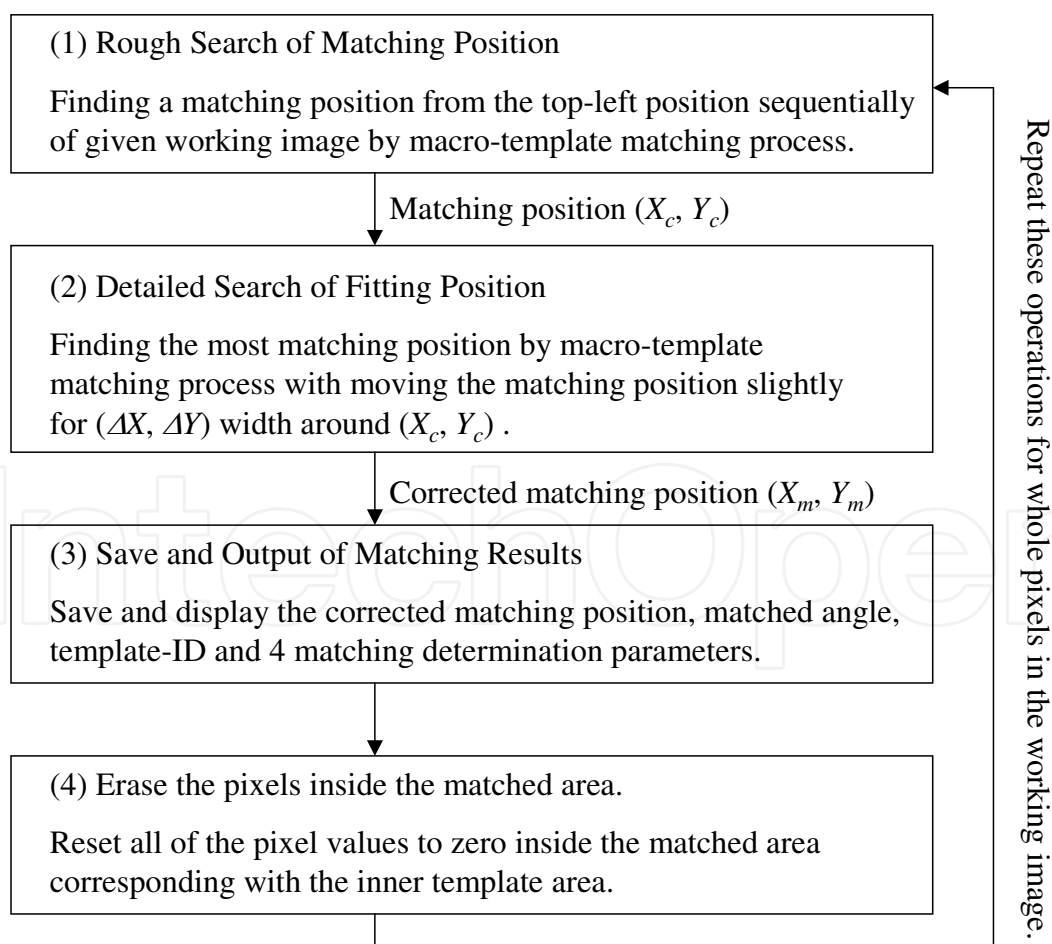

Fig. 8. Construction of our proposing total macro-template matching processes.

most fitted position (X_m, Y_m) where D_{his}, D_{dis} and $D_{sub}(a_{max})$ to be the minimum, and $D_{cor}(a_{max})$ to be the maximum. The third process (3) saves and displays this matched position (X_m, Y_m), matching parameters and the matched angle parameter. The fourth process clears values of the pixel block to zero, corresponding with this matched area where $M_t(x,y) \geq 2$, in the working image $I(x,y)$. This process prevents picking up the already matched area in duplicate.

Figure 9 shows interactive definition processes of macro-template matching conditions: 4 kinds of slice levels as S_{his}, S_{dis}, S_{sub} and S_{cor}, and conceptual updated templates. The first process (1) defines a position (X_c, Y_c) to be matched interactively by a user where the target area should be identified as an object. The second process (2) corrects the defined position to the most fitted position (X_m, Y_m), where D_{his}, D_{dis} and $D_{sub}(a_{max})$ will be the minimum and $D_{cor}(a_{max})$ will be the maximum. The third process calculates 4 kinds of slice levels as: $S_{his}=D_{his} \bullet 1.1$, $S_{dis}=D_{dis} \bullet 1.1$, $S_{sub}=D_{sub} \bullet 1.1$ and $S_{cor}=D_{cor} \bullet 0.9$. The fourth process updates the template image $I_t(x,y)$ by mixing it with the matched pixel block in the working image $I(x,y)$ in order half of pixels to become those of the working image as shown in Fig.10. This process makes the template match with more types of patterns and become a more conceptual pattern.
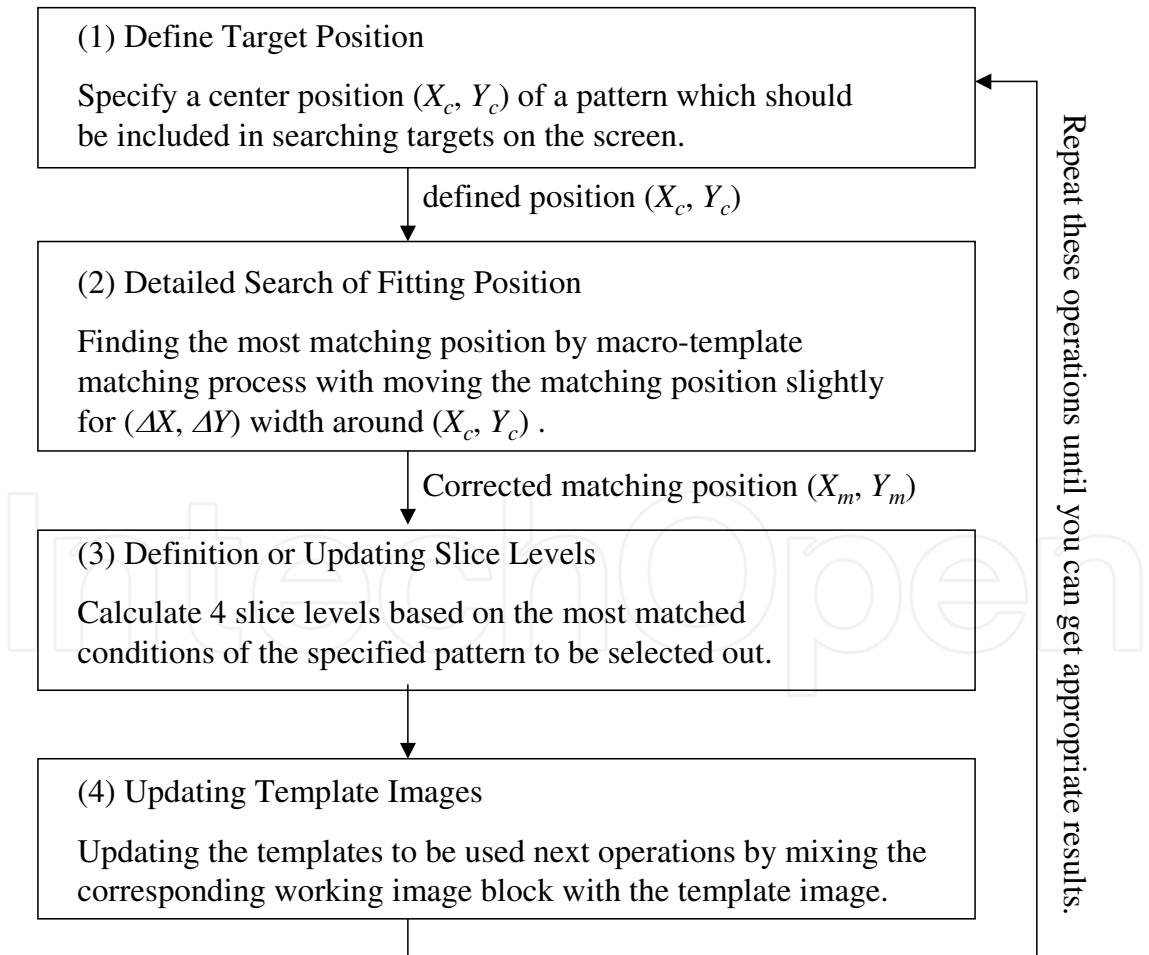
(1) Define Target Position

Specify a center position (X_c, Y_c) of a pattern which should be included in searching targets on the screen.

defined position (X_c, Y_c)

(2) Detailed Search of Fitting Position

Finding the most matching position by macro-template matching process with moving the matching position slightly for $(\Delta X, \Delta Y)$ width around (X_c, Y_c) .

Corrected matching position (X_m, Y_m)

(3) Definition or Updating Slice Levels

Calculate 4 slice levels based on the most matched conditions of the specified pattern to be selected out.

(4) Updating Template Images

Updating the templates to be used next operations by mixing the corresponding working image block with the template image.

Repeat these operations until you can get appropriate results.

Fig. 9. Interactive definition processes of macro-template matching conditions.

[Identifying Target Block]

W	W	W	W	W
W	W	W	W	W
W	W	W	W	W
W	W	W	W	W
W	W	W	W	W

[Macro-template Image]

Masking Pixels

			T	T
		T	T	T
	T	T	T	T
T	T	T	T	T
	T	T	T	T

Either pixel is randomly picked up

[Mixed Image for Updated Template]

			T	W
		T	W	T
	W	T	W	T
W	T	T	W	W
		W	T	T

Fig. 10. Updating process of template images.

4. Experimental results

The Figure 11 shows an example of experimental results using a QuickBird (DigitalGlobe, 2008) panchromatic 11-bit monochrome 13032 x 13028 pixel satellite image Fig. 11-(a) shows 664 x 655 trimmed area after the first segmentation process has been made with a 4 x 4 pixel micro-template. Extracted candidate areas are indicated in red. Fig.11-(b) shows modified area after the clustered template-matching process has been made. The large candidate areas such as roads, which do not include automobiles, have been removed from the candidate. Fig.11-(c) shows the three-time zoomed area 591 x 468 (a parking lot image) of the center-left 197 x 156 after the third identification process has been made with 52 x 52 pixels, 4 kinds of angle macro-templates, whose initial definition and updated patterns are shown in Fig.12. Then the slice levels are defined as: S_{his}=603, S_{dis}=600, S_{sub}=130 and S_{cor}=412.

Fig. 11. Example of automobile recognition experiments by our proposing template matching using a 0.6 [m]-resolution QuickBird panchromatic image (13032 x 13028 pixels).

In this final stage, 61 automobile objects out of 85 automobiles in this picture could be correctly identified, whereas 19 objects out of 558 non-target patterns were excessively identified. This incorrect identification rate has been improved, compared with that of without a micro-template matching or a clustered micro-template matching as shown in Table 1, which shows identification precision rates in three kinds of experiments whether adding a micro-template matching or a clustered micro-template matching. However, correct identification rate has been slightly worse, which should be improved in the future.
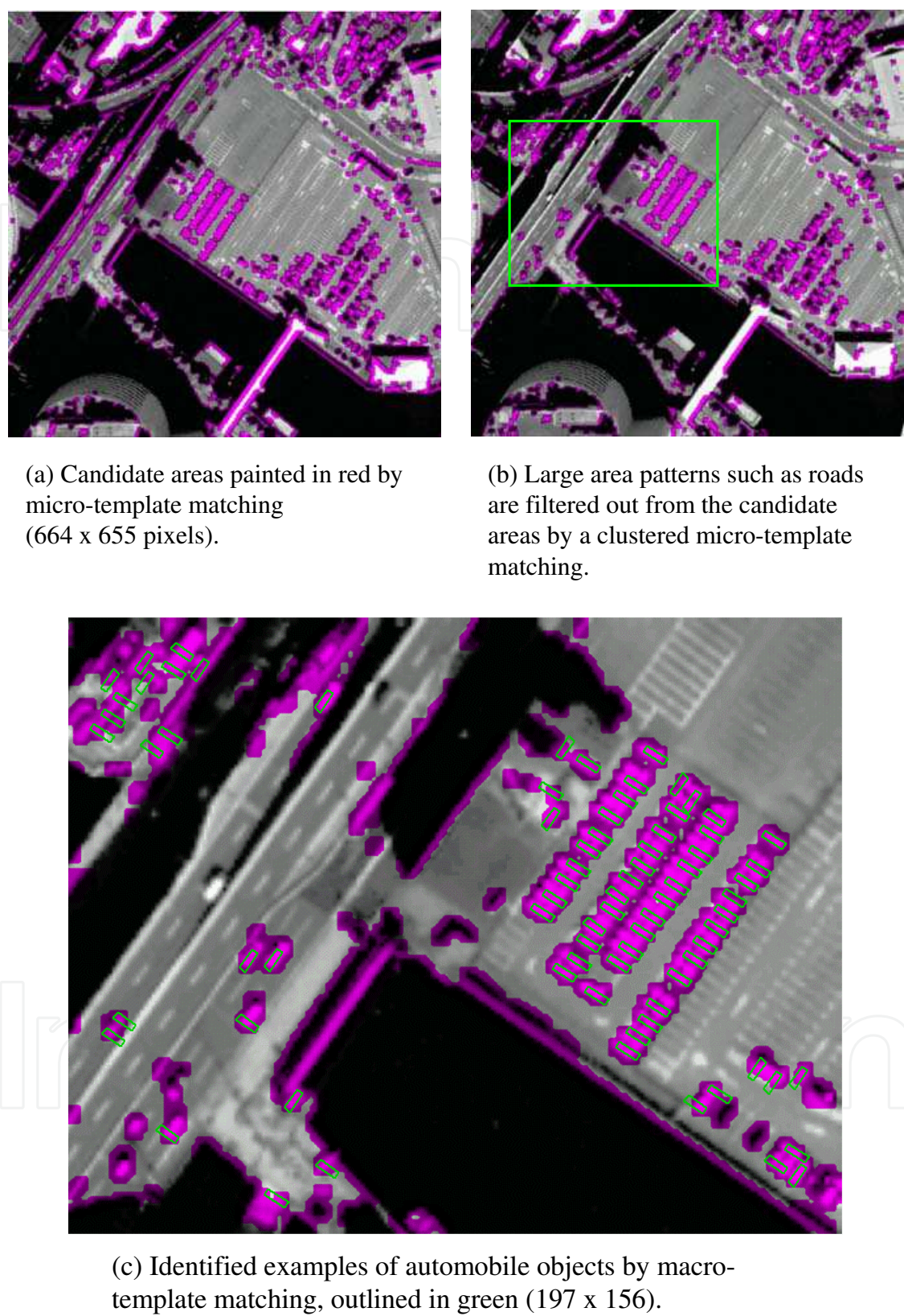
(a) Candidate areas painted in red by
micro-template matching
(664 x 655 pixels).

(b) Large area patterns such as roads
are filtered out from the candidate
areas by a clustered micro-template
matching.

(c) Identified examples of automobile objects by macro-
template matching, outlined in green (197 x 156).

Fig. 11. (Continued) Example of automobile recognition experiments by our proposing
template matching using a 0.6 [m]-resolution QuickBird panchromatic image (13032 x 13028
pixels).
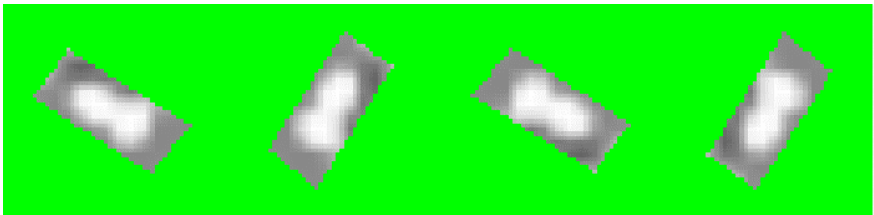
(a) Initial macro-template images 4 kinds of angle (52 × 52)
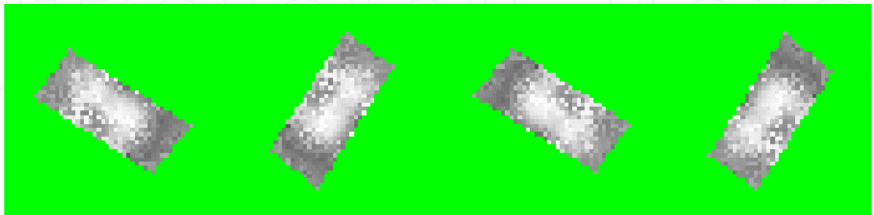
(b) Updated macro-template images used for matching (52 × 52)

Fig. 12. Template definition and updated macro-template image examples (52×52).

Method	true patterns sensitivity (false negative)	false patterns specificity (false positive)
(1) Macro-template matching only	63 74.1% (22/85)	39 93.0% (39/558)
(2) Micro-template matching & Macro-template matching	62 72.9% (23/85)	22 96.0% (22/558)

(3) Micro-template matching
   with Clustered micro-templates
   & Macro-template matching
	61
	71.7%
	(24/85)
		19
		96.5%
		(19/558)

Table 1. Automobile pattern recognition precision results in three kinds of template matching experiments.

5. Texture analysis application of our proposed method

Our proposing micro-template matching method can be extended to indentify the other kinds of objects or areas other than transportation objects. For example, we are trying to extract rice field areas in several kinds of satellite images including low-resolution SAR images by designing micro templates for detecting some spatial frequency feature patterns. In this section, we present an application of our proposing micro-template matching method to texture analysis of satellite images.

5.1 Proposing texture analysis method

Our proposing texture analysis method proposed is based on the micro-template matching method proposed in this paper. This makes binary determinations whether the target pixel $I(x,y)$ is included in metallic texture areas or not, using micro-templates defined with multiple binary determination rules around nearest $N \times N$ pixel blocks. Applying the micro-tmeplates around the target pixel $I(x,y)$ $(=0-255)$, we can create a binary image $B(x,y)$ $(=0$ or $1)$ which indicates metallic areas. In this paper, we propose extending this micro-template to the filter matrices $M(x,y)$ $(= -1$ or $+1)$ which have spatial frequency analysis functions as shown in Fig.13 and Fig.14.

For $N \times N$ block pixels $(N=8)$ around the pixel $I(x,y)$ in given source image data, we calculate the following 4 parameters, and determine $B(x,y)$ $(=0$ or $1)$ whether the target pixel should have predefined texture characteristics or not, by all of the calculated parameter values are included in predefined ranges or not. More specifically, we define 8 kinds of slice values as L_{dis}, H_{dis}, L_{f1}, H_{f1}, L_{f2}, H_{f2}, L_{f4}, H_{f4}; if $L_{dis} \le D_{dis} \le H_{dis}$, $L_{f1} \le F_1 \le H_{f1}$, $L_{f2} \le F_2 \le H_{f2}$, and $L_{f4} \le F_4 \le H_{f4}$, we set as $B(x,y)=1$ $(x=1,8; y=1,8)$.

(1) Pixel Dispersion Value: D_{dis}

$$I_{ave}: \text{Average values of } N \times N \text{ block pixels,}$$

$$\{ \textstyle\sum_{y=1}^{8} \sum_{x=1}^{8} I(x,y) \}/64 \tag{10}$$

$$D_{dis} = \{ \textstyle\sum_{y=1}^{8} \sum_{x=1}^{8} (I(x,y) - I_{ave})^2 \}^{1/2}/8.$$

(2) First Spatial Frequency Components: F_1

We defined 4 kinds of matrices from (1-1-1) to (1-1-4) shown in Fig.13 as $M_{x11}(x,y)$ to $M_{x14}(x,y)$, and 4 kinds of matrices from (1-2-1) to (1-2-4) as $M_{y11}(x,y)$ to $M_{y14}(x,y)$. Using these matrices, we calculate the following F_{x1i} and F_{y1i} $(i=1,4)$, define the maximum values F_{x1} to F_{y1} among each of 4 values F_{x1i} and F_{y1i} $(i=1,4)$, and define a square root average value of these two as F_1.
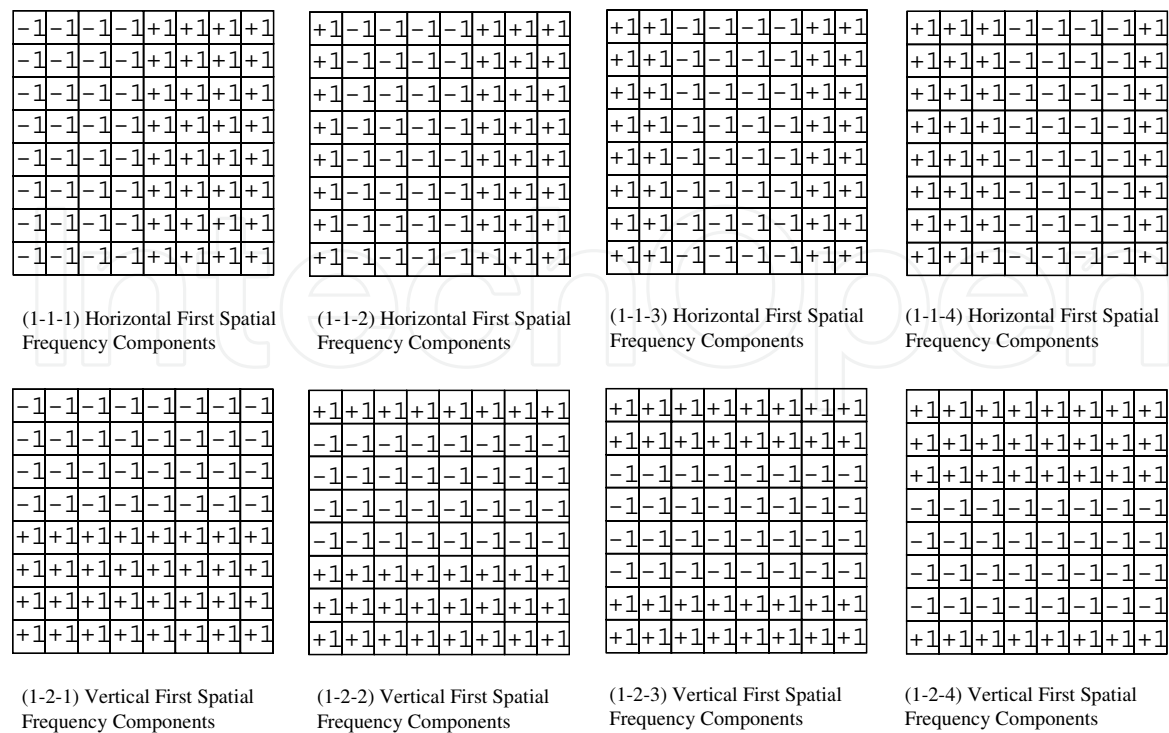
$$F_{x1i} = | \textstyle\sum_{y=1}^{8} \sum_{x=1}^{8} \{ I(x,y) - I_{ave} \} \bullet M_{x1i}(x,y) | / 64, \ (i=1,4)$$

$$F_{y1i} = | \textstyle\sum_{y=1}^{8} \sum_{x=1}^{8} \{ I(x,y) - I_{ave} \} \bullet M_{y1i}(x,y) | / 64, \ (i=1,4)$$

$$F_{x1} = \text{MAX}_{i=1,4} F_{x1i} \tag{11}$$

$$F_{y1} = \text{MAX}_{i=1,4} F_{y1i}$$

$$F_1 = (F_x^2 + F_y^2)^{1/2}.$$

(3) Second Spatial Frequency Components: F_2

We defined 2 matrices (2-1-1) and (2-1-2) shown in Fig.14 as $M_{x21}(x,y)$ and $M_{x22}(x,y)$, and 2 matrices (2-2-1) and (2-2-2) as $M_{y21}(x,y)$ and $M_{y22}(x,y)$. Using these matrices, we calculate the following F_{x2i} and F_{y2i} $(i=1,2)$, define the maximum values F_{x2} to F_{y2} among each of 2 values F_{x2i} and F_{y2i} $(i=1,2)$, and define a square root average value of these two as F_2.

(1-1-1) Horizontal First Spatial Frequency Components

(1-1-2) Horizontal First Spatial Frequency Components

(1-1-3) Horizontal First Spatial Frequency Components

(1-1-4) Horizontal First Spatial Frequency Components

(1-2-1) Vertical First Spatial Frequency Components

(1-2-2) Vertical First Spatial Frequency Components

(1-2-3) Vertical First Spatial Frequency Components

(1-2-4) Vertical First Spatial Frequency Components

Fig. 13. Proposing filter matrices for texture analysis (1).

(2-1-1) Horizontal Second Spatial Frequency Components

(2-1-2) Horizontal Second Spatial Frequency Components

(2-2-1) Vertical Second Spatial Frequency Components

(2-2-2) Vertical Second Spatial Frequency Components

(3-1) Horizontal Fourth Spatial Frequency Components

(3-2) Vertical Fourth Spatial Frequency Components
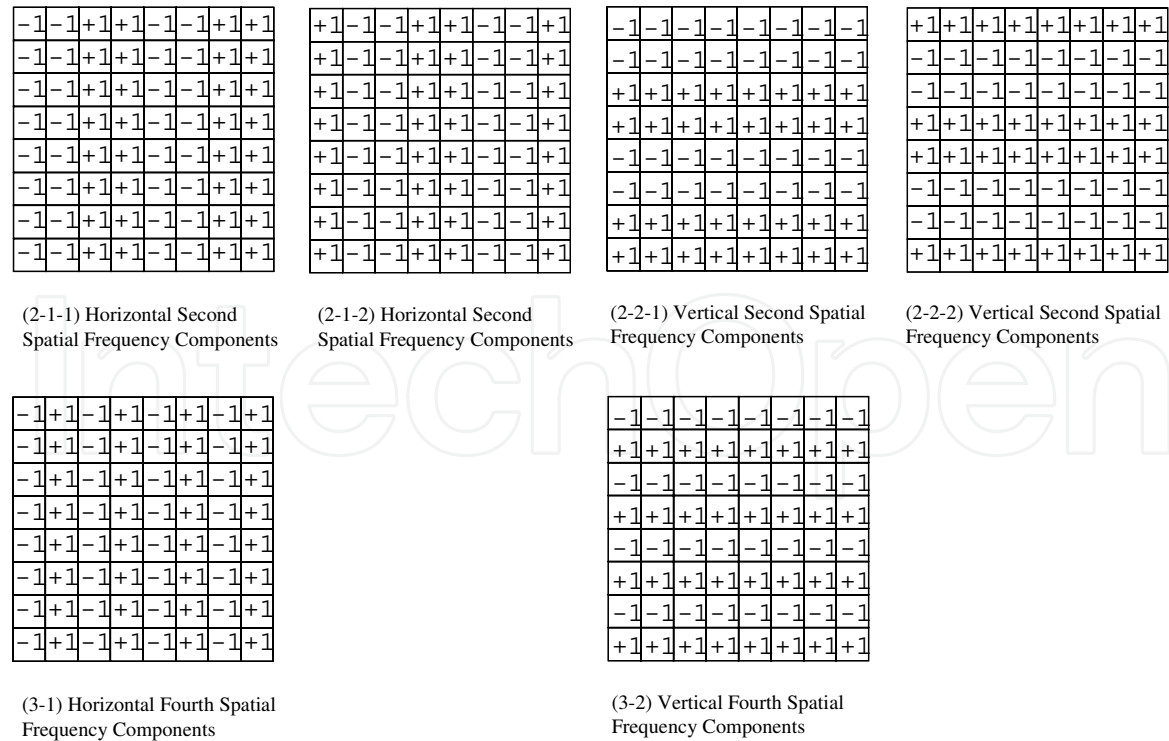
Fig. 14. Proposing filter matrices for texture analysis (2).

$$F_{x2i}=|\ \sum_{y=1}^{8}\sum_{x=1}^{8}\{I(x,y)-I_{ave}\}^{\bullet}M_{x2i}(x,y)\ |\ /64,\ (i=1,2)$$

$$F_{y2i}=|\ \sum_{y=1}^{8}\sum_{x=1}^{8}\{I(x,y)-I_{ave}\}^{\bullet}M_{y2i}(x,y)\ |\ /64,\ (i=1,2)$$

$$F_{x2}=MAX_{i=1,2}F_{x2i} \tag{12}$$

$$F_{y2}=MAX_{i=1,2}F_{y2i}$$

$$F_{2}=(F_{x2}^{2}+F_{y2}^{2})^{1/2}.$$

(4) Fourth Spatial Frequency Components: F_4

We defined 2 matrices (3-1) and (3-2) shown in Fig.14 as $M_{x4}(x,y)$ and $M_{y4}(x,y)$. Using these matrices, we calculate the following F_{x4i} and F_{y4}, and define a square root average value of these two as F_4.

$$F_{x4}=|\ \sum_{y=1}^{8}\sum_{x=1}^{8}\{I(x,y)-I_{ave}\}^{\bullet}M_{x4}(x,y)\ |\ /64$$

$$F_{y4}=|\ \sum_{y=1}^{8}\sum_{x=1}^{8}\{I(x,y)-I_{ave}\}^{\bullet}M_{y4}(x,y)\ |\ /64 \tag{13}$$

$$F_{4}=(F_{x4}^{2}+F_{y4}^{2})^{1/2}.$$

Then we describe how to define 8 kinds of slice values as L_{dis}, H_{dis}, L_{f1}, H_{f1}, L_{f2}, H_{f2}, L_{f4}, H_{f4} interactively. We indicate one of target texture areas to be extracted, area-O, and also indicate two reverse feature areas not to be extracted: area-A and area-B. We calculate average values of 4 parameters D_{dis}, F_1, F_2, F_4 based on the equations described above for each of three selected areas. We define average values in the area-A as A_{dis}, A_{f1}, A_{f2}, A_{f4}, average values in the area-B as B_{dis}, B_{f1}, B_{f2}, B_{f4}, average values in the area-O as O_{dis}, O_{f1}, O_{f2}, O_{f4}. In case of $A_{dis}<B_{dis}$, we can set the values as follows:

$$\text{If } A_{dis}<O_{dis}<B_{dis} \text{ then } L_{dis}=(A_{dis}+O_{dis})/2,$$

$$H_{dis}=(B_{dis}+O_{dis})/2.$$

$$\text{If } A_{dis}<B_{dis}<O_{dis} \text{ then } L_{dis}=(B_{dis}+O_{dis})/2,\ H_{dis}=\infty. \tag{14}$$

$$\text{If } O_{dis}<A_{dis}<B_{dis} \text{ then } L_{dis}=0,\ H_{dis}=(A_{dis}+O_{dis})/2.$$

In case of $A_{dis}>B_{dis}$, we can apply the above by changing values of A_{dis} and B_{dis}.

5.2 Example of rice field area extraction

Applying the previously described texture analysis method to several feature areas in satellite images, we have obtained the specific average values of 4 parameters D_{dis}, F_1, F_2, F_4. As experiment images, we used two kinds of QuickBird (DigitalGlobe, 2008) panchromatic images (urban area:13032×13028 pixels, rural area:29195×34498 pixels), we have extracted 252×182 pixel area and 1676×1563 pixels area from each image and converted depth of brightness level from 11-bits to 8-bits. We selected three areas from the first image, which were a road area without cars existed, a parking area without cars existed and a parking area with multiple cars parked in parallel. And we also selected three areas form the second image, which were a town and housing area, a rice field area and a river area. For each of these selected 6 areas, we calculated average values of 4 parameters D_{dis}, F_1, F_2, F_4, as plotted in Fig.15.
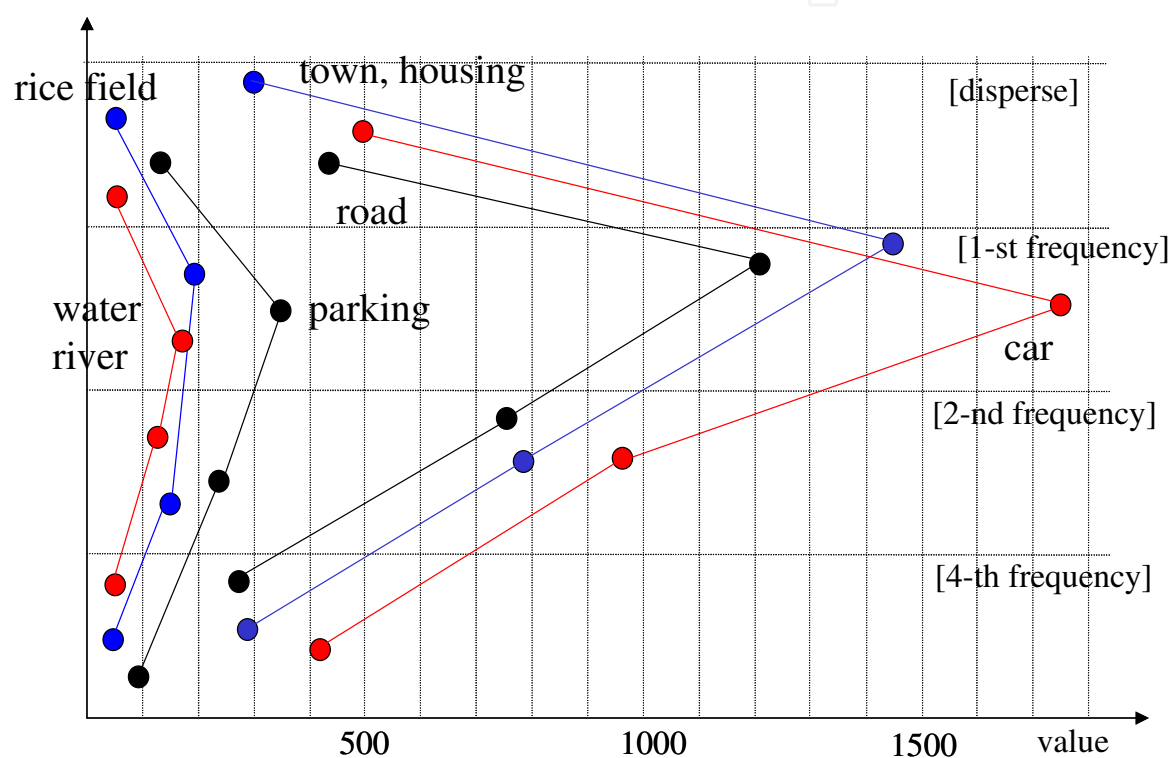
Fig. 15. Specific parameter values of texture analysys in 6 kinds of areas in QuickBird (DigitalGlobe, 2008) satellite images.
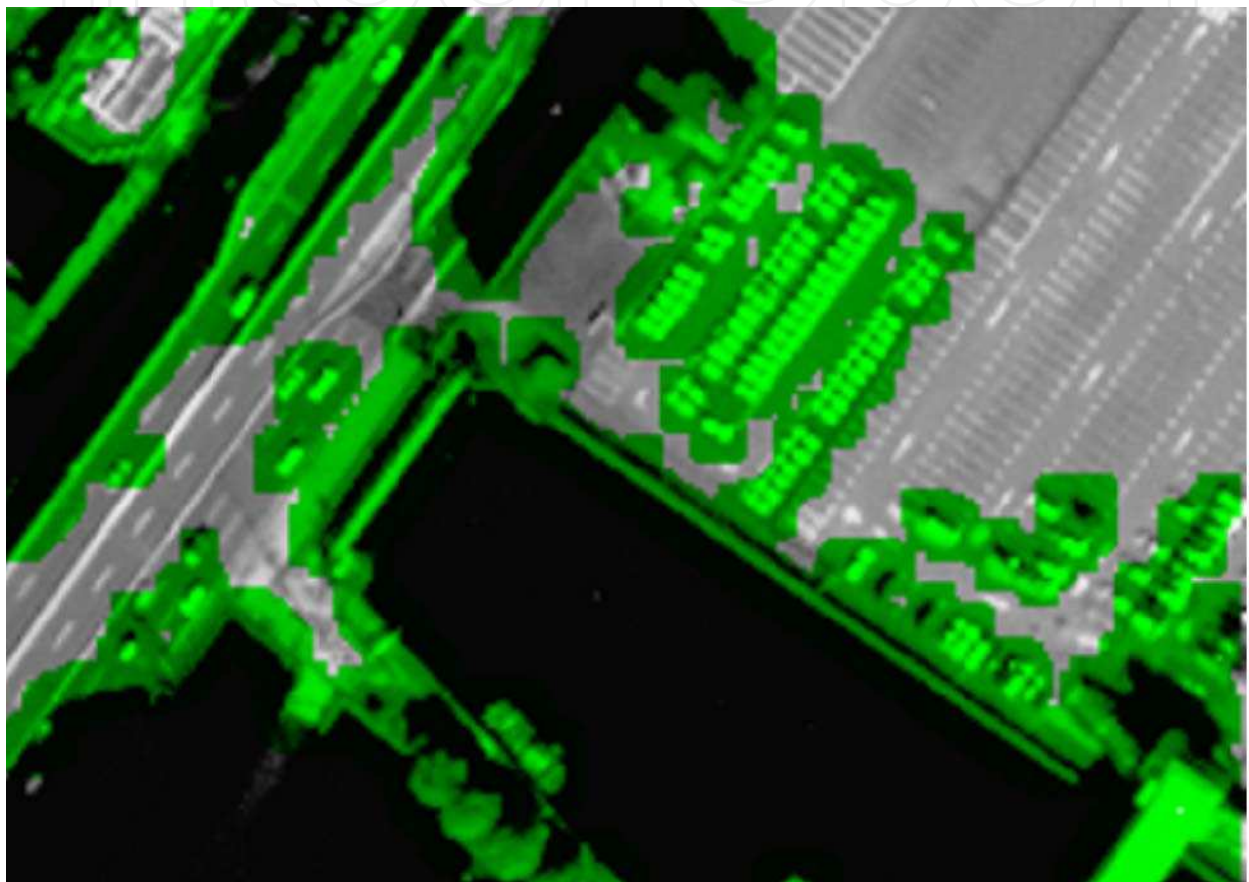
Fig. 16. Example of automobile extraction experiment from road and parking areas in
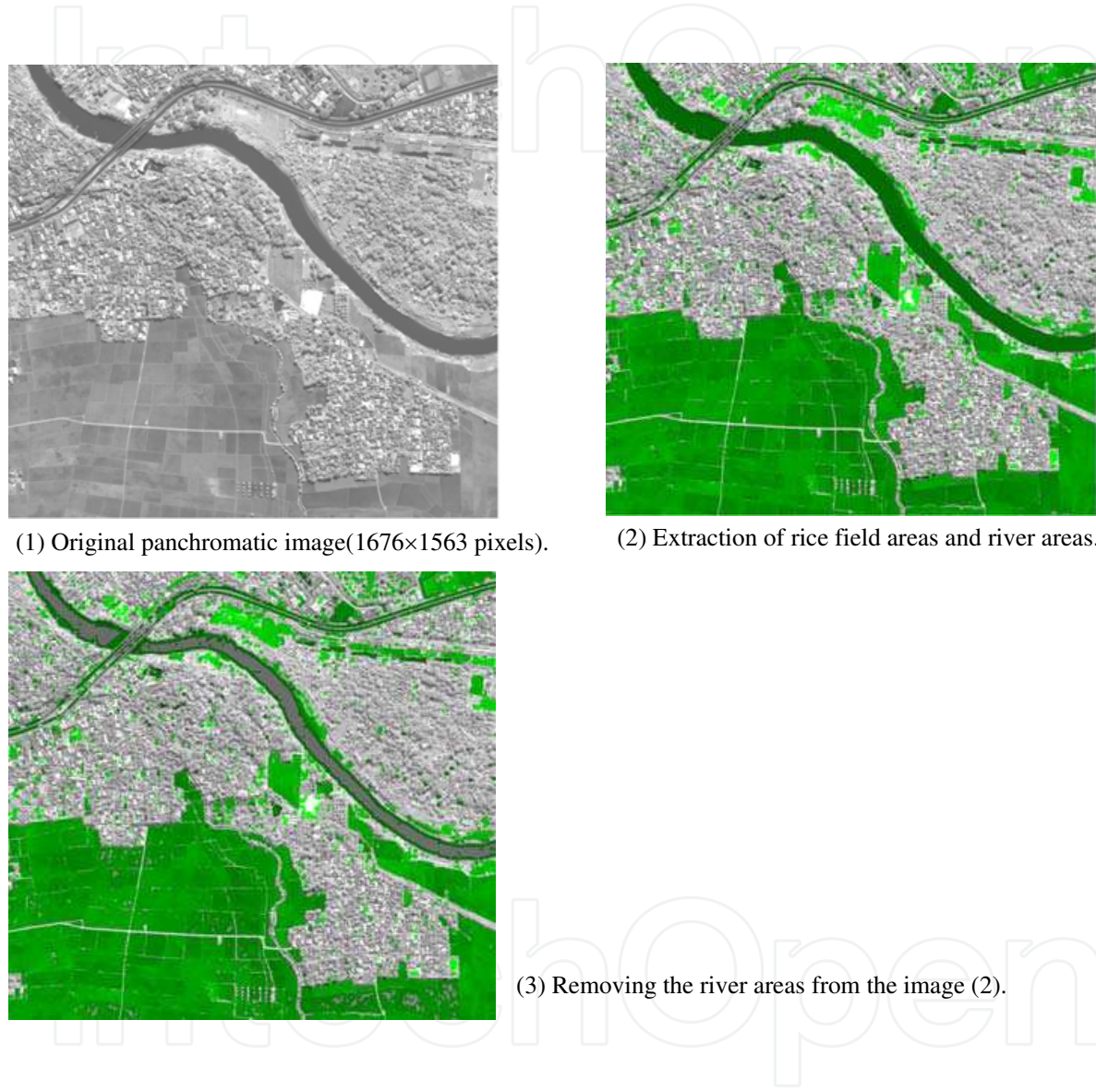QuickBird (DigitalGlobe, 2008) image.

(1) Original panchromatic image(1676×1563 pixels).

(2) Extraction of rice field areas and river areas.

(3) Removing the river areas from the image (2).

Fig. 17. Three kinds of texture separation experiments of rice field areas, river areas, town and housing areas in QuickBird (DigitalGlobe, 2008) image.

In Fig.15, We have divided the vertical direction to four parts, and plotted 4 average parameters for each of the 6 selected areas plotted from the top to bottom parts: pixel dispersion values, first spatial frequency components, second spatial frequency components, and fourth spatial frequency components. The horizontal dimension is 100-time integer values of calculated values by the equation (1) to (4). For example, Figure 16 shows a result of extraction experiment for the first urban area image, setting slice values as L_{dis}=520, H_{dis}=10000, L_{f1}=2200, H_{f1}=10000, L_{f2}=770, H_{f2}=10000, L_{f4}=300, H_{f4}=10000.

As shown in Fig.15, parameter values of the rice field area and the river area are nearer compared with the values of the town and housing area. This indicates, it is difficult to separate the two areas of the rice field area and the river area than separating them from the town and housing area. Therefore, we have made separation experiments of these 3 kinds of texture in the same image, as shown in Fig.17.

Using the source image Fig.17-(1), the result of extraction is shown in Fig.17-(2), setting parameters as L_{dis}=0, H_{dis}=96, L_{f1}=0, H_{f1}=396, L_{f2}=0, H_{f2}=280, L_{f4}=0, H_{f4}=96. Then from the processed image Fig.17-(2), the result of extraction is shown in Fig.17-(3), setting parameters as L_{dis}=21, H_{dis}=96, L_{f1}=0, H_{f1}=396, L_{f2}=80, H_{f2}=280, L_{f4}=0, H_{f4}=96. We can almost separate the rice filed areas from the river areas except the several edge parts of river areas are incorrectly extracted.

6. Conclusions

In this paper, we have proposed a three-layered structured template matching method for decreasing calculation loads and improving identification precisions against the conventional template-matching algorithm. The first layer called as a micro-template matching is focusing on extracting candidate areas, which have metallic-reflection optical characteristics, where any types of transportation objects are included. The second layer called as a clustered micro-template matching is identification and removal of excessively extracted candidate areas such as roads and buildings, which have the similar optical characteristics but do not include our targets. The third layer called as macro-template matching is identifying each automobile object within the focusing area using our proposing conceptual templates, which are learned patterns based on user's operations, based on our improved high-speed template-matching algorithm. In our experiments using our proposed method, we could extract about 70% automobile objects correctly in a single scene of a QuickBird panchromatic image.

Moreover, we have proposed giving a texture analysis function to the first micro-template matching process, by adding independent spatial frequency component as parameters. The previously proposed micro-template matching has been using only a mixed parameter of pixel dispersion value and first spatial frequency component in this proposing texture analysis. It has been difficult to distinguish similar texture characteristic areas shown in such as Fig.17-(2) and Fig.17-(3). We have found overcoming this problem by giving independent spatial frequency component parameters separating from pixel dispersion parameters and adding the second spatial frequency component parameter. As future works, we are going to evaluate functions of the other first and fourth spatial frequency components in several kinds of satellite images, and redesign matrices size or analysis parameters.

We suppose our proposing technology creates new industrial applications and business opportunities of high-cost earth observation satellite images such as a wide-area traffic monitoring, which compensates for the blind areas of the conventional terrestrial traffic

monitoring. Especially, this is effective for planning of city constructions and next-generation traffic networks.

7. References

Tao Zhao & Nevatia R. (2001). Car detection in low resolution aerial image, *Proceedings of Eighth IEEE International Conference on Computer Vision ICCV 2001*, Vol.1, pp. 710-717, ISBN 0-7695-1143-0, Vancouver, BC, July 2001, IEEE, NJ US

Qu Jishuang; Wang Chao & Wang Zhengzhi (2003). Structure-context based fuzzy neural network approach for automatic target detection, *Proceedings of IEEE International Conference on Geoscience and Remote Sensing Symposium IGARSS '03*, Vol.2, pp. 767-769, ISBN 0-7803-7929-2, July 2003, IEEE, NJ US

So Hee Jeon ; Kiwon Lee & Byung-Doo Kwon (2005). Application of template matching method to traffic feature detection using KOMPSAT EOC imagery, *Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGRSS'05)*, pp. 441-443, ISBN 0-7803-9050-4, conference location, July 2005, IEEE, NJ US

Qi-Ming Qin; Si-Jin Chen; Wen-Jun Wang; De-Zhi Chen & Lin Wang (2005). The building recognition of high resolution satellite remote sensing image based on wavelet analysis, *Proceedings of IEEE International Conference on Machine Learning and Cybernetics*, Vol.7, pp. 4533-4538, ISBN 0-7803-9091-1, August 2005, IEEE, NJ US

Giovanni Poggi; Giuseppe Scarpa & Josiane B. Zerubia (2005). Supervised segmentation of remote sensing images based on a tree-structured MRF model, *IEEE Transactions on Geoscience and Remote Sensing*, Vol.43, No.8, (August 2005) pp. 1901-1911, ISSN 0196-2892

DigitalGlobe (2008). QuickBird Sattellite Images by DigitalGlobe, provided by Hitachi Software Engineering Co.,Ltd. (http://www.hitachisoft.jp/products/hgiis/index.html) 2008, Japan.

Modegi T. (2008). Development of Pattern Searching Tool for Satellite Images Using Conceptual Template Patterns, *Proceedings of the 70-th Information Processing Society of Japan*, Demo-2, pp. 1-2, Tukuba Japan, March 2008, IPSJ, Tokyo Japan

Modegi T. (2009). Target Detection Tool for High Resolution Images: "TD-HRS", Enable to Recognize Vehicle Objects in Satellite Images, *Proceedings of the 71-th Information Processing Society of Japan*, Demo-4, pp. 1-2, Kusatsu Japan, March 2009, IPSJ, Tokyo Japan
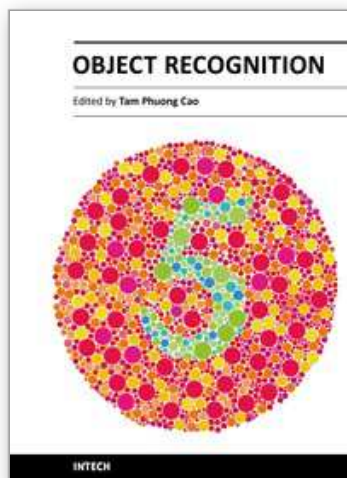
Object Recognition

Edited by Dr. Tam Phuong Cao

Vision-based object recognition tasks are very familiar in our everyday activities, such as driving our car in the correct lane. We do these tasks effortlessly in real-time. In the last decades, with the advancement of computer technology, researchers and application developers are trying to mimic the human's capability of visually recognising. Such capability will allow machine to free human from boring or dangerous jobs.

INTECH
open science | open minds