1-D fast normalized cross-correlation using additions

Jae-Chern Yoo, Byoung Deog Choi, Hyoung-Kee Choi *

School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Gyeonggi-Do, 440-746, Republic of Korea

A R T I C L E   I N F O

A B S T R A C T

A new fast algorithm for the computation of the normalized cross-correlation (NCC) is presented. For a search window of size M and a template of size N, our fast NCC requires only approximately $3N \cdot (M - N + 1)$ additions/subtractions without multiplications. Numerical results with 100,000 test signals show that the use of the fast NCC instead of the traditional approaches for the determination of the degree of similarity between a test signal and a reference signal (template) brings about a significant improvement in terms of false negative rate, identification rate and computational cost without a significant increase in false positive rate, especially when signal to noise ratio (SNR) is higher than 4 dB.

1. Introduction

Template matching by normalized correlations has been commonly used as a metric to evaluate the degree of similarity between two signals, and can be traced back to very early research in pattern recognition [1]. One-dimensional correlation-based methods have been used extensively for many applications such as radar target identification [2–5], Electrocardiogram (ECG) signal processing [6], audio/speech signal processing [7–9], database search [10], watermarking electronic text documents [11,12], etc. In these kinds of applications, usually the signal profiles exist as one-dimensional inherently or even are transformed to one-dimension domain for the facilities of characteristic computations like cross-correlation. Unfortunately the normalized form of correlation preferred in template matching does not have a correspondingly simple and efficient frequency domain expression. For this reason normalized cross-correlation (NCC) has been computed in the spatial domain [2–5]. The main advantage of the NCC over the cross-correlation is that it is less sensitive to linear changes in the amplitude of signal in the two compared signals. Furthermore, the NCC is confined in the range between -1 and 1, and the setting of detection threshold value is much easier than the cross-correlation [14]. The normalized cross-correlation is a reasonable choice in many cases. Nevertheless, it is computationally expensive and its computation time increases dramatically as the size of the template gets larger. Therefore a fast correlation algorithm that requires fewer calculations than the basic version is of interest. The emphasis and contribution of this work is to propose a new fast NCC algorithm only using additions, of which benefit is more obvious when implemented in hardware for real-time applications. Up to now there has not been any study on NCC algorithm only using additions.

2. Fast NCC algorithm

The problem treated in this paper is to determine the position of a given pattern t in a one-dimensional signal f. A common way to calculate the position of the pattern t in the signal f is to evaluate the normalized cross-correlation value (coefficient) at each point (u), which has been shifted by u steps in the x direction. The maximum values or peaks

---

* Corresponding author. Fax: +82 31 299 4909.
E-mail address: hkchoi@ece.skku.ac.kr (H.-K. Choi).

of the computed correlation values indicate the best matches between the template (pattern) t and the signal f. The normalized cross-correlation used for finding matches of a template $t(x)$ of size N in a signal $f(x)$ of size M is defined as

$$\gamma_u(f, t) = \frac{\sum_x [f_u(x) - \overline{f}_u] \cdot [t(x - u) - \overline{t}]}{\sqrt{\sum_x [f_u(x) - \overline{f}_u]^2 \cdot \sum_x [t(x - u) - \overline{t}]^2}}, \tag{1}$$

where the summations are over all template coordinates, $f_u(x)$ is $f(x)$ in the region under the template (positioned at the place u), $\overline{t}$ is the mean of the template and $\overline{f}_u$ is the mean of $f_u(x)$.

Unfortunately, the calculation of (1) is computationally too expensive. The number of computations required to calculate the NCC coefficients is $2N \cdot (M - N + 1)$ multiplications and $3N \cdot (M - N + 1)$ additions/subtractions.

2.1. Derivation of the fast NCC coefficient

Let us assume that:

$$f_{u_0}(x) = \alpha \cdot t(x - u_0) \quad \text{for a given } u_0, \tag{2}$$

where α is a positive constant that represents a scaling factor and the lower suffix letter u_0 denotes a location in which the NCC coefficient is 1. Under the assumption of (2), we can rewrite (1) as (3):

$$\begin{aligned}
\gamma_{u_0}(f, t) &= \frac{\sum_x [f_{u_0}(x) - \overline{f}_{u_0}] \cdot [t(x - u_0) - \overline{t}]}{\sqrt{\sum_x [f_{u_0}(x) - \overline{f}_{u_0}]^2 \cdot \sum_x [t(x - u_0) - \overline{t}]^2}} \\
&= \frac{\sum_x [f_{u_0}(x) - \overline{f}_{u_0}]^2}{\sum_x [f_{u_0}(x) - \overline{f}_{u_0}]^2} \quad \left(f_{u_0}(x) = \alpha \cdot t(x - u_0) \right) \\
&= \frac{1}{N} \sum_x \frac{[f_{u_0}(x) - \overline{f}_{u_0}]^2}{[f_{u_0}(x) - \overline{f}_{u_0}]^2} \\
&= \frac{1}{N} \sum_x \frac{[f_{u_0}(x) - \overline{f}_{u_0}] \cdot [f_{u_0}(x) - \overline{f}_{u_0}]}{|f_{u_0}(x) - \overline{f}_{u_0}| \cdot |f_{u_0}(x) - \overline{f}_{u_0}|}.
\end{aligned} \tag{3}$$

As long as $\gamma_{u_0}(\cdot) = 1$, it is possible to replace $\overline{f}_{u_0}$ with any real value and then (3) gives numerically satisfactory results. In this paper, we chose the values $f_{u_0}^+(x)$ and $f_{u_0}^-(x)$ such that $f_{u_0}^+(x) \geqslant f_{u_0}(x)$ and $f_{u_0}^-(x) \leqslant f_{u_0}(x)$. Therefore,

$$\gamma_{u_0}(f, t) = \frac{1}{N} \sum_x \frac{[f_{u_0}^+(x) - f_{u_0}(x)] \cdot [f_{u_0}(x) - f_{u_0}^-(x)]}{|f_{u_0}^+(x) - f_{u_0}(x)| \cdot |f_{u_0}(x) - f_{u_0}^-(x)|}. \tag{4}$$

For a given SNR (Signal to Noise Ratio), we can rewrite the assumption (2) as

$$\left| f_{u_0}(x) - \alpha \cdot t(x - u_0) \right| \leqslant \alpha \eta, \tag{5}$$

where the value η was chosen as a positive real number such that the signal $f_{u_0}(x)$ is bounded between $f_u^-(x) = \alpha \cdot [t(x - u) - \eta]$ and $f_u^+(x) = \alpha \cdot [t(x - u) + \eta]$.

Note that the worse the SNR is, the larger the value η will be. The SNR is the power ratio between a template $t(x)$ and the background noise $n(x)$:

$$SNR = 10 \log_{10} \frac{\sum_x |t(x)|^2}{\sum_x |n(x)|^2}. \tag{6}$$

From (6), we obtain

$$\sum_x |n(x)|^2 \propto \frac{1}{10^{(SNR/10)}}. \tag{7}$$

Here, assuming that the amplitude of $n(x)$ is approximately uniform with the value η over the region under the template gives

$$\sum_x |n(x)|^2 \approx N \cdot \eta^2. \tag{8}$$

By substituting (8) into (7), we can approximate the value η as

Table 1
The values of parameter k as a function of SNR.

SNR (dB)	−3	−2	−1	0	1	2	3	4	5	6	7	8	9	10
k	0.6	0.7	0.73	0.75	0.78	0.80	0.83	0.87	0.89	0.91	0.93	0.95	0.97	0.98

$$\eta \approx \lambda \cdot \sqrt{\frac{1}{N \cdot 10^{(SNR/10)}}}, \tag{9}$$

where the parameter λ is a proportional constant with a positive value.

If the formula (4) is used even when the assumption (2) may not be valid or when $u \neq u_0$, the result is an approximation to the NCC coefficient. We will refer to this approximation as the fast NCC coefficient denoted as $\Lambda_u(\cdot)$, which is given by

$$\Lambda_u(f, t) \triangleq \frac{1}{N} \sum_x F_u^+(x) \cdot F_u^-(x) \quad \text{for a given shift } u, \text{ where } F_u^+(x) = \frac{f_u^+(x) - f_u(x)}{|f_u^+(x) - f_u(x)|}, \ F_u^-(x) = \frac{f_u(x) - f_u^-(x)}{|f_u(x) - f_u^-(x)|}. \tag{10}$$

The values $F_u^+(x)$ and $F_u^-(x)$ are given as follows:

$$F_u^+(x) = \begin{cases} +1 & \text{if } f_u(x) - \alpha \cdot t(x-u) < \alpha \cdot \eta, \\ -1 & \text{otherwise,} \end{cases} \tag{11}$$

$$F_u^-(x) = \begin{cases} +1 & \text{if } \alpha \cdot t(x-u) - f_u(x) < \alpha \cdot \eta, \\ -1 & \text{otherwise.} \end{cases} \tag{12}$$

Here, we can have the observations that if the signal $f_u(x)$ is bounded between $f_u^-(x)$ and $f_u^+(x)$, then the values $F_u^+(x)$ and $F_u^-(x)$ will be both 1, resulting in $\Lambda_u(\cdot) = 1$.

Notice that the larger the proportional constant λ in (9), the larger the value η will be, and then it will lead to more robust $F_u^+(x)$ and $F_u^-(x)$ to noise. However, it may cause more false alarms. Therefore, there is an inevitable trade-off between robustness to noise and performance in choosing λ. Thus, the proportional constant λ were experimentally chosen to satisfy the rate of overall identification as high as possible with minimum number of false alarms for the given SNR and finally set to 3.2.

2.2. Proposed fast NCC algorithm

We assume that (i) the template t is pre-normalized to be between 0 and 1, (ii) the SNR is known a priori, (iii) the scaling factor α is taken in the range of [0.1, 10]. Given two signals, $f(x)$ and $t(x)$, the steps of the proposed fast NCC algorithm include the following:

Step 1: Estimate the scaling factor α.
We used the ratio of the average between $f_u(x)$ and $t(x)$, as the estimator $\hat{\alpha}(u)$ of the original scaling factor α:

$$\hat{\alpha}(u) = k \cdot \frac{\delta_f(u)}{\delta_t}, \tag{13}$$

where $\delta_f(u) = \sum_{x=u}^{u+N-1} f(x)$, $\delta_t = \sum_{x=0}^{N-1} t(x)$ and the parameter k is a real number.
After testing, we chose the value of k that gave the best estimation of the scaling factor for a given SNR.

Table 1 shows the experimental values of k under different SNR for estimating the best scaling factor α. Notice that the better the SNR is, the closer to 1 the value of k is.

Also, the value $\delta_f(u)$ at each shifting instant is based on that at the previous instant, u. Therefore,

$$\delta_f(u+1) = \delta_f(u) + f(u+N) - f(u). \tag{14}$$

Thus the computation for $\delta_f(u)$ during entire matching process involves approximately $2M - N$ additions.

Step 2: Next, compute the values $F_u^+(x)$ and $F_u^-(x)$ by using $\hat{\alpha}(u)$ $(= k \cdot \delta_f(u)/\delta_t)$ obtained from Step 1.
The inequality condition of (11) can be rearranged as follows:

$$f_u(x) - \hat{\alpha}(u) \cdot t(x-u) < \hat{\alpha}(u) \cdot \eta \tag{15}$$

which reduces to

$$\big(f_u(x) - \Delta_u(x)\big) < \theta_u, \tag{16}$$

where $\Delta_u(x) = \delta_f(u) \cdot k \cdot t(x-u)/\delta_t$ and $\theta_u = \delta_f(u) \cdot k \cdot \eta/\delta_t$.

Table 2
Number of computations for calculating $\Lambda_u(\cdot)$.

	Addition/subtraction	Multiplication	
Step 1	$2M - N$	0	(14)
Step 2	$2N \cdot (M - N + 1)$	0	(18)
Step 3[*]	$N \cdot (M - N + 1)$	0	(10)

[*] The division for the term $(1/N)$ in (10) is not included if $\Lambda_u(f, t_i) \geqslant N \cdot TH_{roi}^{\Lambda}$ instead of (23) is used. "Logic AND" operations performed in (10) are not included due to their triviality. Here, it is assumed that an AND-accumulator circuit for these logic computations is built.

Table 3
The comparison of arithmetic operations among the proposed method and the traditional normalized correlation techniques.

	Addition/subtraction		Multiplication	
	Numerator	Denominator	Numerator	Denominator
Direct NCC	$N \cdot (M - N + 1)$	$2N \cdot (M - N + 1)$	$N \cdot (M - N + 1)$	$N \cdot (M - N + 1)$
FFT	$9M \log_2 M$	$2N \cdot (M - N + 1)$	$6M \log_2 M$	$N \cdot (M - N + 1)$
	(FFT)	(Direct)	(FFT)	(Direct)
Lewis[*]	$9M \log_2 M$	$3 \cdot (M - N + 1)$	$6M \log_2 M$	0
	(FFT)	(Lewis)	(FFT)	(Lewis)
Walsh Transforms	$(N - 1) \cdot (M - N + 1)$		$N \cdot (M - N + 1)$	
Fast NCC	$3N \cdot (M - N) + 2M - N$		0	

[*] The extra computations required to set up the sum tables are not included.

Similarly, the inequality condition of (12) is

$$\big(f_u(x) - \Delta_u(x)\big) > -\theta_u. \tag{17}$$

Here, the values $k \cdot t(x - u)/\delta_t$ and $k \cdot \eta/\delta_t$ can be precomputed, and $\Delta_u(x)$ and θ_u can be obtained from a precomputed look-up table. That is, $\Delta_u(x)$ is obtained from the table indexed by $\delta_f(u)$ and $k \cdot t(x - u)/\delta_t$ while θ_u by $\delta_f(u)$ and $k \cdot \eta/\delta_t$.

In practice, the amount of memory required for storing such look-up tables is less than 110 Kbytes when it is assumed that the scaling factor $\hat{\alpha}(u)$ is defined as values in the range $[0.1, 10]$ with a precision of 0.1 and the template $t(x - u)$ in the range $[0, 1]$ with a precision of 0.001.

Finally, we have from (16) and (17)

$$F_u^+(x) \cdot F_u^-(x) = \begin{cases} +1 & \text{if } |f_u(x) - \Delta_u(x)| < \theta_u, \\ -1 & \text{otherwise.} \end{cases} \tag{18}$$

Step 3: Finally, compute the fast NCC coefficient $\Lambda_u(\cdot)$ of (10).

Given $F_u^+(x) \cdot F_u^-(x)$, the computation (10) for $\Lambda_u(\cdot)$ during entire matching process involves approximately $N \cdot (M - N + 1)$ additions. Notice that the division for the term $(1/N)$ in (10) is not included if $\Lambda_u(f, t_i) \geqslant N \cdot TH_{roi}^{\Lambda}$ instead of (23) in Section 4.3 is used.

Repeat the above Steps 1–3 with $u = u + 1$ until u reaches $(M - N)$.

3. Complexity analysis

For a signal of size M and a template of size N the approximate number of computations required at each step in Section 2 is given in Table 2.

The proposed method belongs to the class of exhaustive search algorithms. Therefore, in order to evaluate its performance gain, we compare the fast NCC algorithm to the traditional normalized correlation techniques such as the direct NCC, cross-correlation using Fourier transform, Lewis algorithm [15], fast block matching using Walsh Transforms [16], fast template-matching algorithm using adaptive skipping [17–19].

Table 3 shows the comparison of arithmetic operations between the proposed method and these traditional normalized correlation techniques. For a search window of size M and a feature of size N the fast NCC algorithm requires approximately $3N \cdot (M - N + 1)$ additions/subtractions without multiplications, which opens up many new time-critical applications. In contrast to this, the direct NCC requires $2N \cdot (M - N + 1)$ multiplications and $3N \cdot (M - N + 1)$ additions/subtractions. Cross-correlation using FFT is obtained by calculating only the numerator of (1) in the frequency domain, resulting in being not normalized. In addition, the complexity of the FFT depends on the size of the template $t(x)$ and the signal $f(x)$ [20]. When M is much larger than N, it may even exceed the number of computations required by the direct method. Lewis' sum-table approach can be used to efficiently calculate the denominator of (1). However, it cannot be applied to compute the numerator of (1), and thus the number of computations required to calculate (1) is still comparatively high even if the computation of the numerator of (1) is done in the frequency range with FFT algorithm. Furthermore, Lewis' algorithm requires approximately $3 \cdot M$ operations for the construction of the sum tables [15]. Fast block matching with normalized

cross-correlation using Walsh Transforms [16] is faster than the direct NCC, but requires more operations than those needed by the fast NCC algorithm.

The fast template-matching algorithms using adaptive skipping method [17–19] skip unnecessary calculations for achieving efficient search. However, these approaches can make lots of false negatives under noisy environment because of the skipping process and requires multiplications during the computation of the NCC as well.

To meet real-time requirements for time-critical applications such as surveillance radar system, collision avoidance and cruise missile [21,22] it is crucial to use a hardware based template matching [23–25].

From a hardware viewpoint, the addition operation needs much less than a tenth of hardware complexity and processing time compared to a multiplication circuit [26]. Therefore, as the bit-widths of signal get larger, the computational advantages of the fast NCC over the traditional ones are more obvious.

4. Simulation results

A simulation was conducted to test the fast NCC algorithm proposed in this paper. The database was structured with 1000 templates denoted by $t_i(x)$ where the indexes $i = \{1, 2, \ldots, 1000\}$. These templates were used to examine the performance of the presented approach.

4.1. Generating the template $t_i(x)$ and the signal $f_i(x)$

J. Ben-Aire and K.R. Rao proposed a method for one-dimensional signal decomposition by Gaussian basis functions and showed that the one-dimensional signal can be approximated by a set of Gaussian basis functions [13]. In this paper, to generate various templates, a Gaussian pulse was considered:

$$g_n(x) = A_n \cdot e^{-(x/\sigma_n)^2} \quad \text{for } 0 \leqslant x \leqslant N-1, \tag{19}$$

where A_n is the maximum amplitude of the pulse and σ_n is the pulse half-duration at the $1/e$ point.

The parameter A_n between 0 and 1 was chosen at random, while the σ_n was chosen arbitrary in $(0.001N, N/32)$. By choosing random values, the shape and amplitude of the Gaussian pulse $g_n(x)$ generated could be randomly altered. The template $t_i(x)$ can be obtained by superposition of P Gaussian pulses as follows:

$$
\begin{aligned}
&\text{Initialization: } t_i(x) = 0 \\
&\text{For } n = 1:1:P \\
&\quad temp = t_i(x) + g_n(x - \tau) \\
&\quad t_i(x) = \frac{temp}{\max(temp)} \\
&\text{End}
\end{aligned}
\tag{20}
$$

where the τ is a random displacement which was chosen between 0 and $N-1$ with Gaussian distribution of mean $N/2$ and variance $(N/8)^2$, and the arbitrary P was chosen in $(0.2N, N)$.

By these random parameters the shape and amplitude of the templates $(t_i(x))$ are various enough to test the performance of the algorithm presented in this paper. However, the template $t_j(x)$ such that $\gamma_u(t_i, t_j \mid i \neq j) \geqslant TH^\gamma_{roi}$ was excluded from consideration, where the value TH^γ_{roi} is a threshold for identification. Some examples of these templates with various profiles are shown in the left column in Fig. 1. Also, the corresponding signals $f_i(x)$ to the templates are shown in the right column in the same row in Fig. 1.

The signals $f_i(x)$ with various SNR from -3 to 10 dB were generated from the templates as follows:

$$f_i(x) = \alpha \cdot t_i(x) + AWGN, \tag{21}$$

where $f_i(x)$ was labeled with the corresponding template index i for verification, the scaling factor α was taken in the range of $[0.1, 10]$ with increment of 0.1.

The signal $f_i(x)$ for each database entry to check the correlation performance was created, and thus there are total of 100,000 $f_i(x)$ for a given SNR. A test signal $f(x)$ of size M can be generated by AWGN of size M in (21), where $M > N$ is assumed.

In this simulation, the size N between 70 and 90 was chosen at random, while M was set to $4 \cdot N$. Fig. 2 shows a typical test signal $f(x)$ which includes $f_i(x)$ together with background noise depending on SNR.

4.2. The choice of TH_{roi}

Fig. 3 shows $\gamma_u(f_i, t_i)$ and $\Lambda_u(f_i, t_i)$, averaged over i as a function of SNR, in the matching position. It can be seen that the normalized correlation coefficient $\gamma_u(f_i, t_i)$ decreases as the SNR (Signal to Noise Ratio) decreases. This figure provides information on how to choose the suitable threshold TH_{roi}, which is given by

$$
\begin{aligned}
TH^\gamma_{roi} &= \overline{\gamma_u(f_i, t_i)} - \sigma_\gamma \quad \text{for the direct NCC}, \\
TH^\Lambda_{roi} &= \overline{\Lambda_u(f_i, t_i)} - \sigma_\Lambda \quad \text{for the fast NCC},
\end{aligned}
\tag{22}
$$
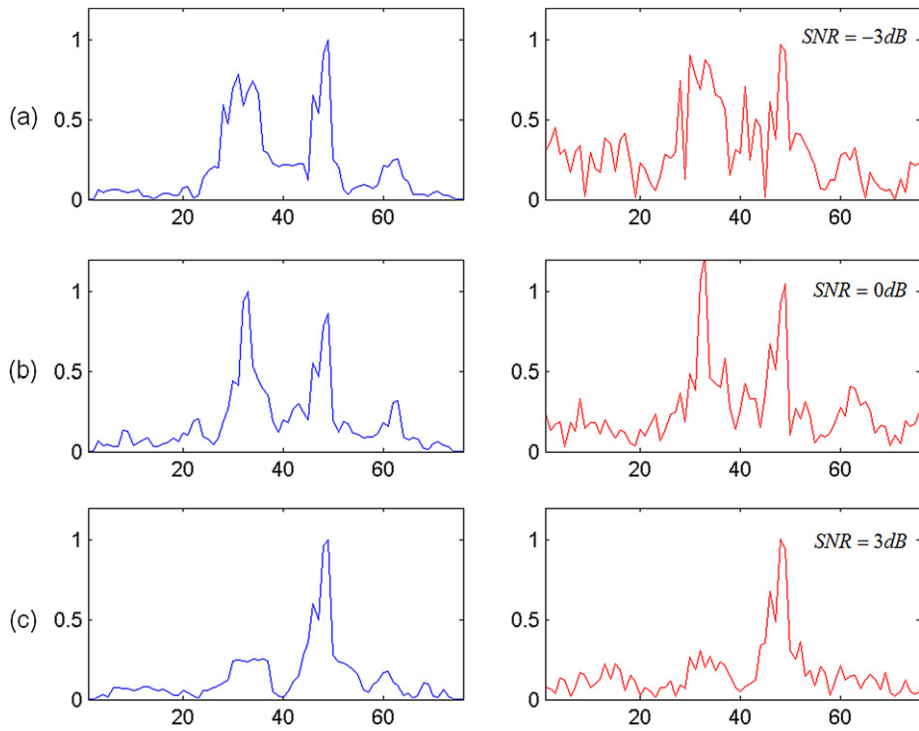
Fig. 1. Some examples of $t_i(x)$ and the corresponding $f_i(x)$ when $\alpha = 1$. Y-axis indicates $t_i(x)$ (left column) and the corresponding $f_i(x)$ (right column).
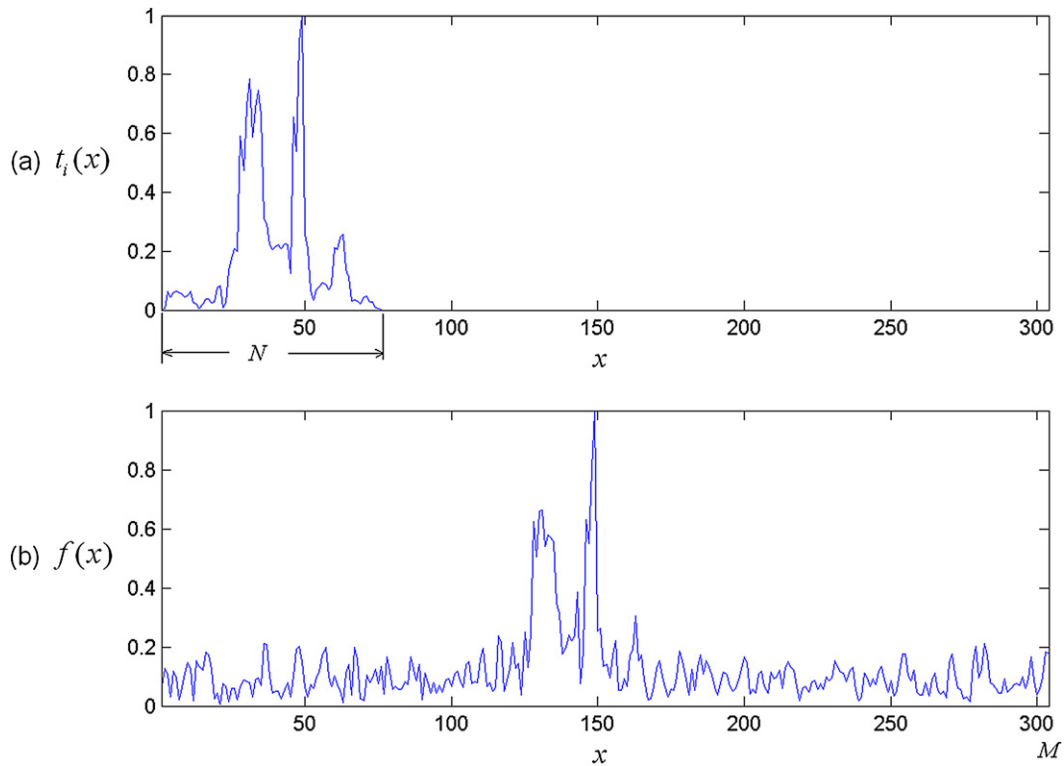
Fig. 2. An example of a test signal $f(x)$ which includes $f_i(x)$ together with background noise. (a) A template $t_i(x)$ and (b) the corresponding test signal $f(x)$ (3 dB AWGN).
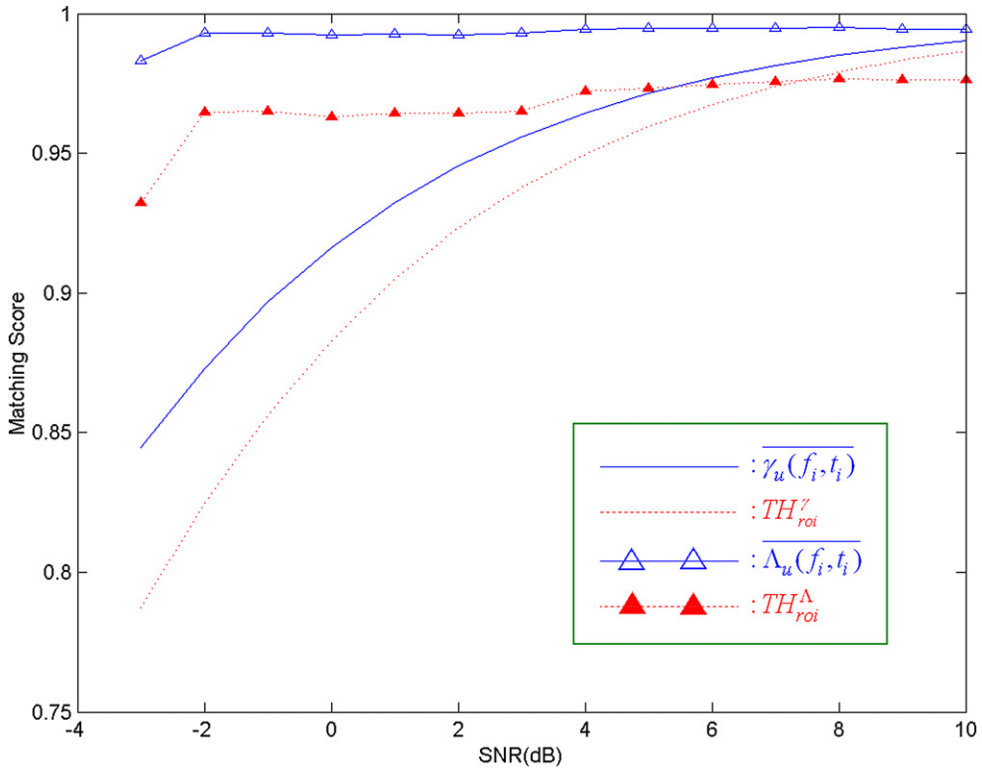
Fig. 3. Normalized cross-correlation coefficient versus signal to noise ratio.

where $\overline{(\cdot)}$ and $\sigma_{(\cdot)}$ are the mean and standard deviation of the normalized correlation coefficient between $t_i(x)$ and $f_i(x)$, and the upper suffix letter "γ" denotes the threshold TH_{roi} for the direct NCC while "Λ" denotes the threshold for the fast NCC.

The lower the SNR is, the lower TH_{roi}^{γ} is. However, TH_{roi}^{Λ} is almost constant for various SNR levels. This is because the value $\Lambda_u(f_i, t_i)$ under the optimal value η for the given SNR is evaluated (see (9)). The values of TH_{roi} versus SNR are stored in the form of look-up table and are used when (23) and (24) are evaluated.

4.3. Measures of performance

The algorithm described in Section 2.2 was applied to the test signal $f(x)$ to obtain the coefficient $\Lambda_u(\cdot)$. For a given test signal $f(x)$, all entries in the database are scanned through and then the indices of database whose matching score is higher than TH_{roi}^{Λ} are listed as candidates.

Let us denote by $\mathbb{R}$ the set of all possible index pairs (i, u) such that

$$\Lambda_u(f, t_i) \geqslant TH_{roi}^{\Lambda} \tag{23}$$

and then take the set $\mathbb{R}$ as a set of candidates for the test signal $f(x)$.

If there is more than one index that matches the criteria in (23) the results will be listed in descending order of matching score. The higher the score, the more likely it is the correct index. However, if the number of candidates with a high matching score is large, they cannot easily be identified only by listing in descending order of matching score. The number of candidates will be increased rapidly by false positives as the SNR decreases. In this paper, the direct NCC is used to filter out the unsuitable candidates from the set $\mathbb{R}$ and in particular such a filtering process can considerably reduce the false positives caused by the approximation ((9) and (10)) used in deriving the coefficient $\Lambda_u(\cdot)$.

Let us denote by $\Omega_u(\cdot)$ the NCC coefficients that meet the following filtering process:

$$\gamma_{u \in \mathbb{R}}(f, t_i \mid i \in \mathbb{R}) \geqslant TH_{roi}^{\gamma}. \tag{24}$$

Note that the computational burden in obtaining the $\Omega_u(\cdot)$ entirely depends on the size of the set $\mathbb{R}$ (i.e., the signal to noise ratio).

Some comparisons were made between the algorithms based on the three coefficients ($\gamma_u(\cdot)$, $\Lambda(\cdot)$ and $\Omega_u(\cdot)$) with the test signal $f(x)$ and then the performance of each algorithm in terms of false positive/negative rate and identification rate was evaluated. The false positive rate is the proportion of negative cases that were incorrectly classified as positive while
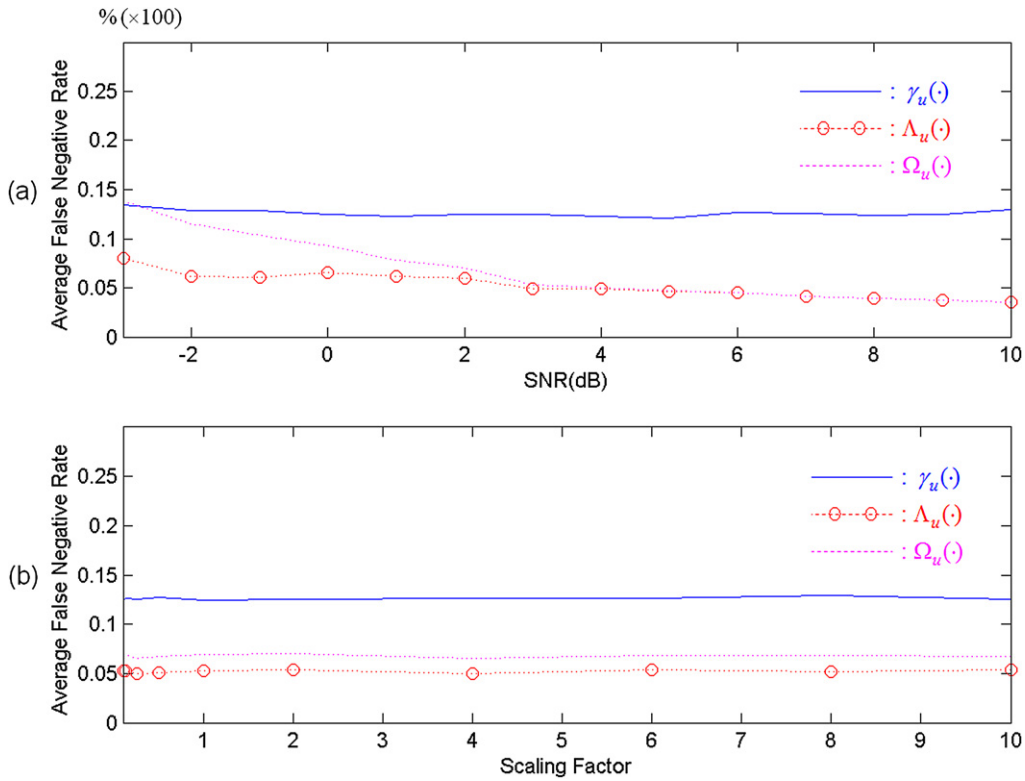
Fig. 4. Comparison of average FN rate between the three coefficients. (a) FN rate versus SNR, averaged over scaling factor between 0.1–10. (b) FN rate versus scaling factor, averaged over SNR between −3 dB–10 dB.

the false negative rate is the proportion of positive cases that were incorrectly identified as negative. The identification rate is defined as the percentage of successful detections for all trials. Fig. 4 shows the average false negative (FN) rate while Fig. 6 the average false positive (FP) rate. It can be seen that $\Lambda_u(\cdot)$ gives relatively better FN rate than $\gamma_u(\cdot)$. Generally, there is a trade-off between the FN and FP rate. As shown in Figs. 4 and 6, the FN rate of $\Lambda_u(\cdot)$ is better than that of $\gamma_u(\cdot)$ while the FP rate of $\Lambda_u(\cdot)$ is worse than that of $\gamma_u(\cdot)$. That is, we obtained a lower FN rate at the sacrifice of the FP rate and thus the identification rate of $\Lambda_u(\cdot)$ is better than that by $\gamma_u(\cdot)$ (see Fig. 8). However, such a high FP rate of $\Lambda_u(\cdot)$ can be reduced to as low as that of $\gamma_u(\cdot)$ by the filtering process given in (24). This is because the filtering process is done by $\gamma_u(\cdot)$. As a result of the filtering process, the FP rate of $\Omega_u(\cdot)$ become almost the same as those of $\gamma_u(\cdot)$, as shown in Figs. 6 and 7(c).

That is, we can have the observations that the filtering process in (24) gives significant improvement in FP rate and yields almost the same FP rate as in the direct NCC for all SNR conditions. Furthermore, even though we obtained a lower FN rate than $\gamma_u(\cdot)$ at the sacrifice of the FP rate there was no significant difference in FP rate between $\Lambda_u(\cdot)$ and $\gamma_u(\cdot)$ for high SNR (> 4 dB). This is because the number of FP itself decreases around high SNR regions. Fig. 8(a) shows a graph that plot the average identification rate as a function of SNR. As shown in this figure, the identification results by $\Lambda_u(\cdot)$ is higher than that by $\gamma_u(\cdot)$ while keeping almost the same FP rate as and less FN rate than in the direct NCC, especially when the SNR is greater than about 4 dB. Also, it is seen from Figs. 5, 7 and 9 that the fast NCC is very stable to the scaling factor α.

The simulation results can be summarized as follows:

(1) The identification rate of the fast NCC is better than that of the direct NCC, especially when the signal to noise ratio (SNR) is higher than 4 dB. Nevertheless, (i) there is no significant difference in FP rate between the fast NCC and the direct NCC, (ii) since any multiplications are not used during the evaluation of the NCC, the computational cost is significantly reduced in comparison with other traditional approaches, in particular when the function of NCC is implemented by hardware, and (iii) the FN rate is lower than that in the direct NCC.
(2) For low SNR (< 4 dB), the filtering process given in (24) to suppress false positives is necessary and then the amount of computation required for the process depends primarily on the SNR. This is because the size of the set $\mathbb{R}$ rapidly increases as SNR decreases.
(3) The fast NCC can be used as a coarse identifier for rough locating a template when SNR is low.
(4) The fast NCC can be used as a fine identifier without using the filtering process when SNR is high.

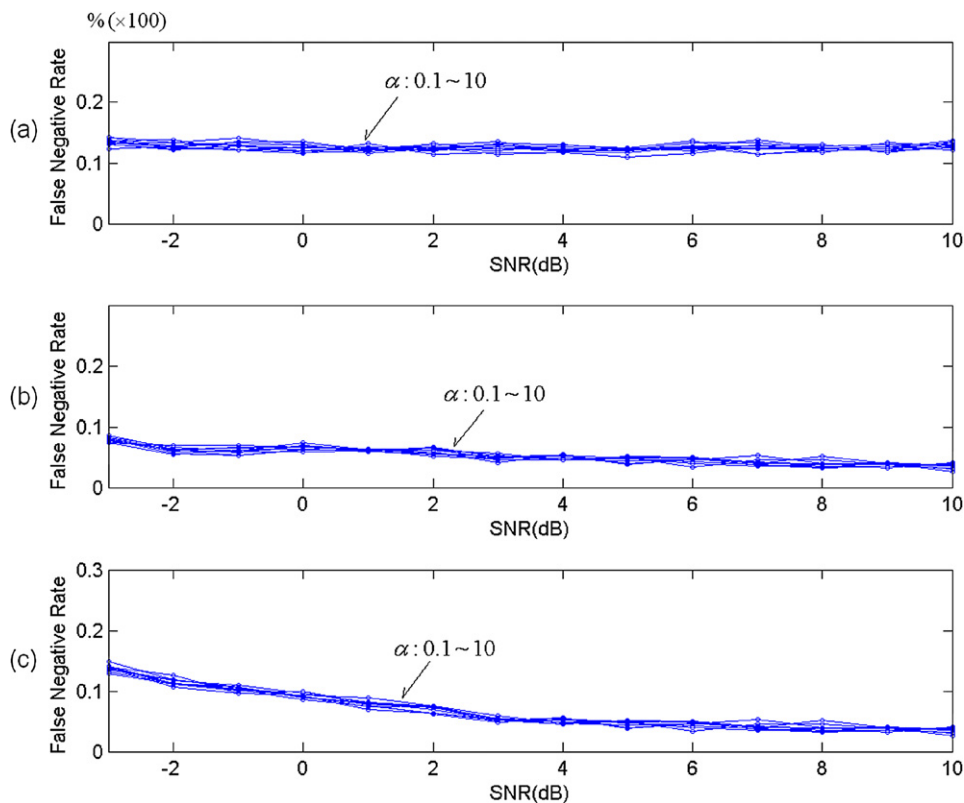These results clearly show the advantage of using the fast NCC algorithm.

Fig. 5. Plots of FN rate versus SNR with various scaling factor α. (a) $\gamma_u(\cdot)$, (b) $\Lambda_u(\cdot)$, (c) $\Omega_u(\cdot)$.
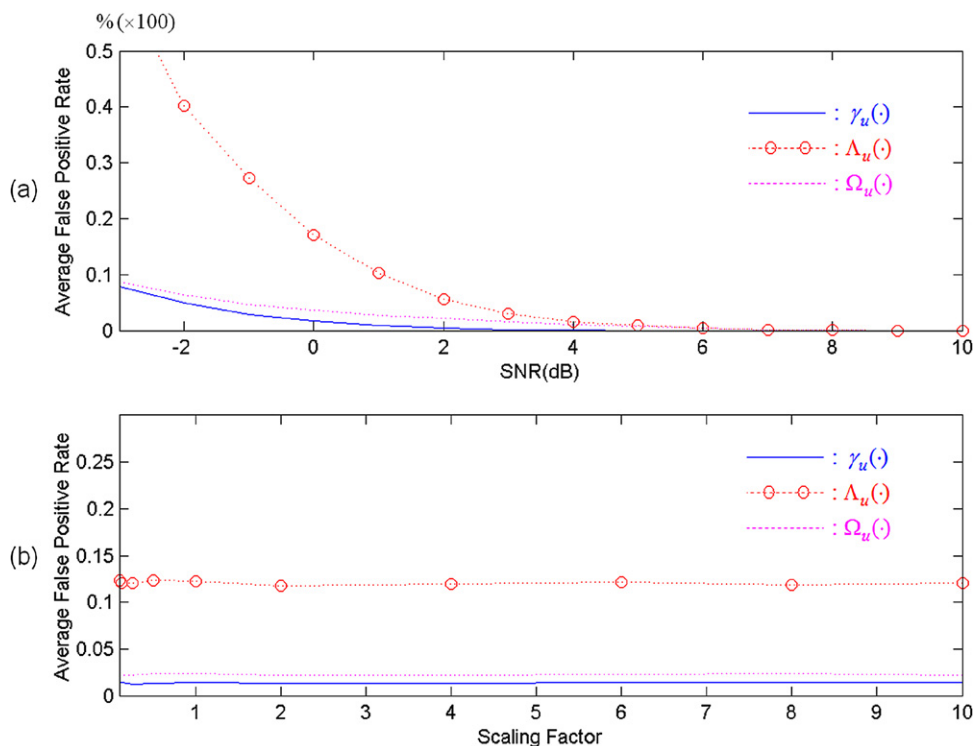

Fig. 6. Comparison of average FP rate between the three coefficients. (a) FP rate versus SNR, averaged over scaling factor between 0.1–10. (b) FP rate versus scaling factor, averaged over SNR between −3 dB–10 dB.
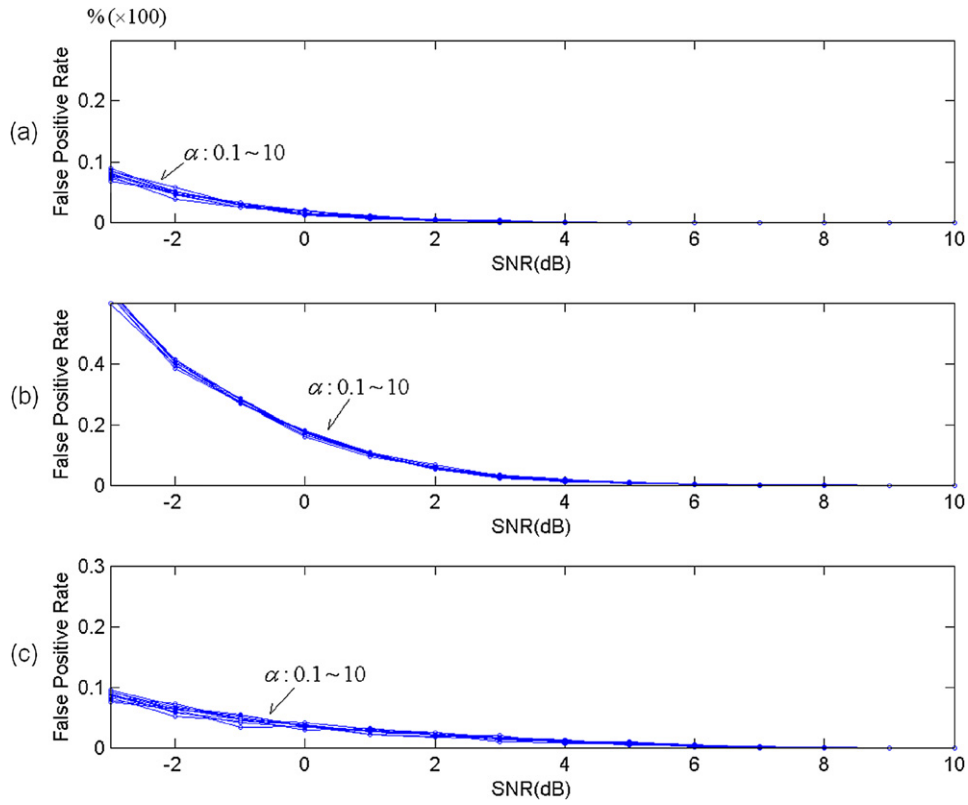
Fig. 7. Plots of FP rate versus SNR as a function of scaling factor. (a) $\gamma_u(\cdot)$, (b) $\Lambda_u(\cdot)$, (c) $\Omega_u(\cdot)$.
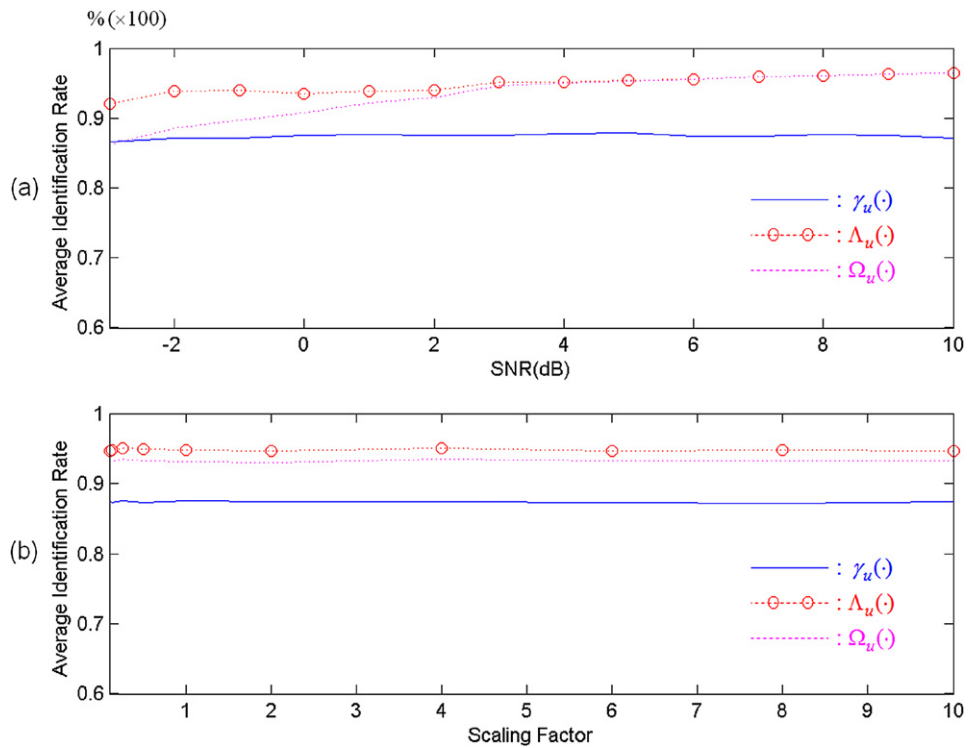

Fig. 8. Comparison of average identification rate between the three coefficients. (a) Identification rate versus SNR, averaged over scaling factor between 0.1–10. (b) Identification rate versus scaling factor, averaged over SNR between −3 dB–10 dB.
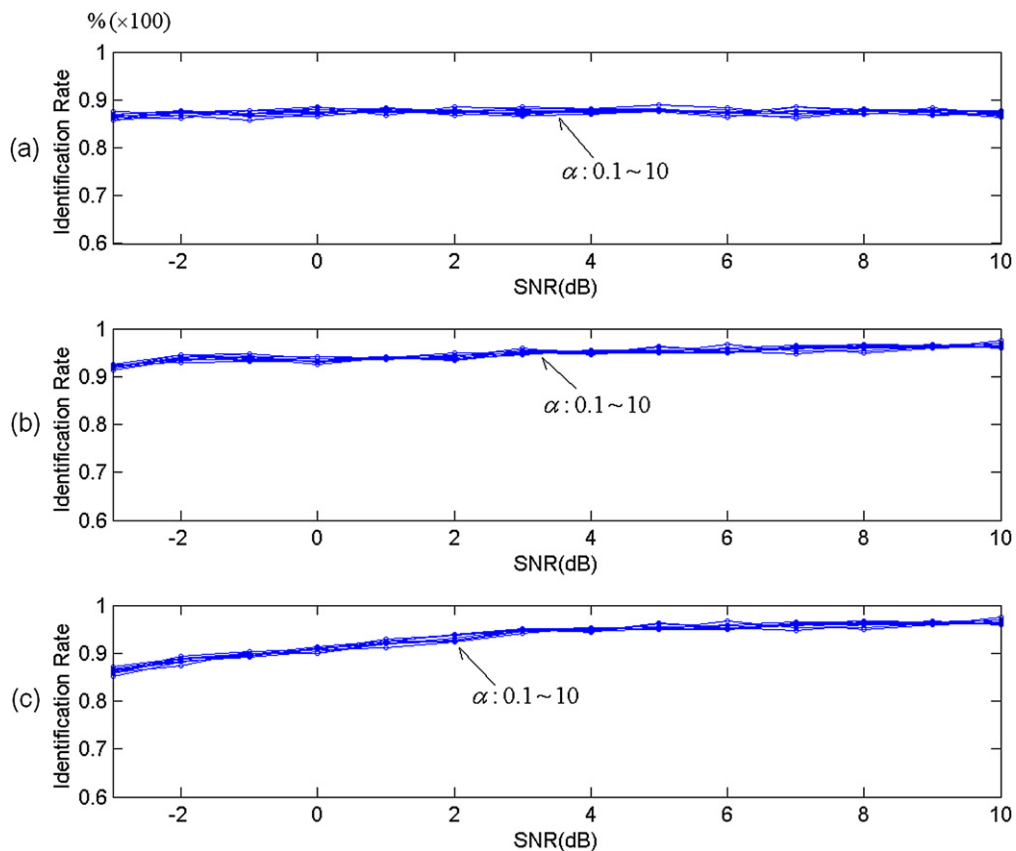
Fig. 9. Plots of identification rate versus SNR as a function of scaling factor. (a) $\gamma_u(\cdot)$, (b) $\Lambda_u(\cdot)$, (c) $\Omega_u(\cdot)$.

5. Conclusions

We have developed a novel template-matching algorithm for 1-D signal, called a fast NCC algorithm, which achieves a significant computation reduction with no loss in identification performance and without increasing FP and FN. The fast NCC is fast compared with the direct NCC and Lewis' algorithm. This is because most of its calculations consist of only additions, which make them computationally efficient. Our simulation results with 100,000 test signals show that the proposed algorithm can achieve better performance in terms of FN and identification rate without a significant increase in FP rate while keeping considerably less computational cost as compared to the conventional algorithms, especially when the SNR is higher than 4 dB. In addition, it was seen that the fast NCC can be used as either a coarse identifier or a fine identifier, depending on the SNR. We are confident that the fast NCC will be widely used for time-critical applications such as object recognition in image and word spotting in speech.

Acknowledgment

References

[1] Richard O. Duda, Peter E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, 1973.
[2] Hsueh-Jyh Li, Yung-Deh Wang, Long-Huai Wang, Matching score properties between range profiles of high resolution radar targets, IEEE Trans. Antennas Propagation 44 (4) (1996) 444–452.
[3] Long-Huai Wang, Hsueh-Jyh Li, Using range profiles for data Association in multiple-target tracking, IEEE Trans. Aerospace Electron. Syst. 32 (1) (January 1996) 445–450.
[4] Scott Hudson, Demetri Psaltis, Correlation filters for aircraft identification from radar range profiles, IEEE Trans. Aerospace Electron. Syst. 29 (3) (July 1993) 741–748.
[5] Rob Williams, John Westerkamp, Automatic target recognition of time critical moving targets using 1D high range resolution (HRR) radar, IEEE Aerospace Electron. Syst. Mag. 15 (4) (April 2000) 37–43.
[6] Chuang-Chien Chiu, Tong-Hong Lin, Ben-Yi Liau, Using correlation coefficient in ECG waveform for arrhythmia detection, Biomed. Eng. Appl. Basis Commun. (June 2005) 147–152.
[7] Ronald M. Aarts, Roy Irwan, A.J.E.M. Janssen, Efficient tracking of the cross-correlation coefficient, IEEE Trans. Speech Audio Process. 10 (6) (September 2002) 391–402.

[8] G. Jacovitti, R. Cusani, An efficient technique for high correlation estimation, IEEE Trans. Acoust. Speech Signal Process. ASSP-35 (May 1987) 654–660.
[9] A. Craciun, M. Gabrea, Correlation coefficient-based voice activity detector algorithm, in: Proc. IEEE Canadian Conference on Electrical and Computer Engineering, May 2004, pp. 1789–1792.
[10] Yiping Ke, James Cheng, Wilfred Ng, Correlation search in graph databases, in: Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 390–399.
[11] A. Tefas, A. Giannoula, N. Nikolaidis, I. Pitas, Enhanced transform-domain correlation-based audio watermarking, in: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, March 2005, pp. 1049–1052.
[12] A.M. Alattar, O.M. Alattar, Watermarking electronic text documents containing justified paragraphs and irregular line spacing, in: Proc. of the SPIE, vol. 5306, 2004, pp. 685–695.
[13] J. Ben-Aire, K.R. Rao, A lattice network for signal representation using Gaussian basis functions and max-energy paradigm, in: Circuits and Systems, Proceedings of the 34th Midwest Symposium, vol. 1, 14–17 May, 1991, pp. 76–79.
[14] Du-Ming Tsai, Chien-Ta Lin, Fast normalized cross correlation for defect detection, Pattern Recognit. Lett. 24 (15) (2003) 2625–2631.
[15] J.P. Lewis, Fast normalized cross-correlation, in: Proceedings of Vision Interface, 1995, pp. 120–123.
[16] Peter Nillius, Jan-Olof Eklundh, Fast block matching with normalized cross correlation using Walsh Transforms, Technical Report ISRN KTH/NA/P-02/11-SE, September 2002.
[17] Takahito Kawanishi, Takayuki Kurozumi, Kunio Kashino, Shigeru Takagi, A fast template matching algorithm with adaptive skipping using inner-subtemplates' distances, in: 17th International Conference on Pattern Recognition (ICPR'04), vol. 3, 2004, pp. 654–657.
[18] Sakata Shun'ichi, Toyama Fubito, Shoji Kenji, Miyamichi Juichi, A fast template matching using adaptive window skipping method considering position of reference template, IPSJ SIG Technical Reports, vol. 2007, no. 1 (CVIM-157), 2007, pp. 201–208.
[19] Wei Shou-Der, Lai Shang-Hong, Fast template matching based on normalized cross correlation with adaptive multilevel winner update, IEEE Trans. Image Process. 17 (11) (2008) 2227–2235.
[20] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, 2nd ed., ISBN 0-13-094650-8, Prentice Hall, 2002, pp. 701–704.
[21] Xuanke Zhao, Qingwu Zhao, Hongxia Wang, Junfa He, Real-time images matching location technology with joint-transform-correlator, Proc. SPIE 5642 (2005) 451–456.
[22] S. Le Beux, V. Gagne, E. Aboulhamid, P. Marquet, J.L. Dekeyser, Hardware/software exploration for an anti-collision radar system, in: 49th IEEE International Midwest Symposium on Circuits and Systems, San Juan, Puerto Rico, vol. 1, August 2006, pp. 385–389.
[23] Tyler J. Moeller, David R. Martinez, Field programmable gate array based radar front-end digital signal processing, in: 7th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 1999, p. 178.
[24] R. Tate, J. Northern, Fast template matching system using VHDL, in: Region 5 Conference, 2008, IEEE, April 2008, pp. 1–5.
[25] S. Le Beux, P. Marquet, O. Labbani, J.L. Dekeyser, FPGA implementation of embedded cruise control and anti-collision radar, in: Proceedings of the 9th EUROMICRO Conference on Digital Systems Design, 2006, pp. 280–287.
[26] J.M. Rabaey, A. Chandrakasan, B. Nikolic, Digital Integrated Circuits, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2002.

Jae-Chern Yoo received the B.S. degree in electronics from Sungkyunkwan University in 1986, and the M.S. and Ph.D. degrees in information & communication engineering and electronics from KAIST and POSTECH, Korea, in 1996 and 2001, respectively. Since March 2008, he has been in Sungkyunkwan University, Korea as an assistant professor. His researches of interest are in the area of pattern recognition, neural network, bio signal processing and image processing.

B.D. Choi received the B.S. and M.S. degrees in physics from Kyung Hee University, Seoul, S. Korea in 1988 and 1990, respectively, and the M.S. and Ph.D. degrees in electrical engineering from Arizona State University, Tempe, Arizona in 1998 and 2001, respectively. He spent 6 years at Samsung in S. Korea. In 2008, he joined School of Information and Communication Engineering, Sungkyunkwan University in S. Korea, where he is currently an assistant professor. His research involves MOS devices, defects in semiconductor, carrier lifetime measurement, and gate oxide integrity.

Hyoung-Kee Choi received the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology in 2001. He is an assistant professor and a director of the Education Center for Mobile Communications in Sungkyunkwan University, Korea. He joined Lancope in 2001 and remained until 2004, where he guided and contributed to research in Internet security. His research interests span network security and Internet traffic modeling. He serves as an Associate Editor for ACM Transactions on Internet Technology.