Circuits Syst Signal Process (2009) 28: 819–843
DOI 10.1007/s00034-009-9130-7

Fast Normalized Cross-Correlation

Jae-Chern Yoo · Tae Hee Han

Received: 29 August 2007 / Revised: 10 December 2008 / Published online: 22 August 2009

Abstract Normalized cross-correlation has been used extensively for many signal processing applications, but the traditional normalized correlation operation does not meet speed requirements for time-critical applications. In this paper, a new fast algorithm for the computation of the normalized cross-correlation (NCC) without using multiplications is presented. For a search window of size M and a template of size N the fast NCC requires only approximately $2N \cdot (M - N + 1)$ additions/subtractions without multiplications. Simulation 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 the signal-to-noise ratio (SNR) is higher than 3 dB.

Keywords Normalized cross-correlation · Correlation coefficient · FFT · Signal-to-noise ratio (SNR)

1 Introduction

Searching and locating patterns of interest in one- (1-D) and two-dimensional (2-D) signals is one fundamental task occurring in countless signal analysis applications. The matched filter (cross-correlator) has been used in communication systems to detect and classify 1-D signals in a noisy environment. The matched filter is the optimal

J.-C. Yoo · T.H. Han (✉)
School of Information and Communication Engineering, Sungkyunkwan University, 300 Cheoncheon-dong, Jangan-gu, Gyeonggi-do, Suwon 440-746, Republic of Korea
e-mail: than@skku.edu

J.-C. Yoo
e-mail: yoojc@skku.edu

linear filter for maximizing the SNR in the presence of additive stochastic noise. From a statistical point of view, the matched filter actually assumes additive white Gaussian noise which is a bad model for objects embedded in signals with various intensities. Also, the main drawback of the matched filter is that the correlation values depend on signal brightness. A bright signal would therefore generate larger correlation values than dim signals and this will definitely cause false positives (FP) and false negatives (FN). A better method is to use a template matching by normalized correlations which have 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 [4]. Correlation-based methods have been used extensively for many applications such as object recognition [2], face recognition [11], radar target identification [9, 10], industrial inspections of printed-circuit boards [5], defect detection [12], etc. 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 [5, 9, 10]. The main advantage of the NCC over the cross correlation is that it is less sensitive to linear changes in the amplitudes of the two compared signals. Furthermore, the NCC is confined in the range between -1 and 1. The setting of a detection threshold value is much easier than the cross-correlation. The NCC 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 algorithm that computes the NCC without using multiplications.

In Sect. 2, the problem treated in this paper is defined and a brief summary of the NCC algorithm is given. Section 3 introduces a new, fast algorithm that computes the NCC without using multiplications. We refer to it as the "fast NCC algorithm". The performance of the new algorithm is compared to that of traditional algorithms such as the NCC, cross-correlation using the Fourier transform and Lewis's algorithm [8]. In Sect. 4, an example is presented, in which the proposed algorithm is applied for template matching. Section 5 presents simulation results.

2 NCC Algorithm

The problem treated in this paper is to determine the position of a given pattern t in a one-dimensional (1-D) signal f. A common way to calculate the position of the pattern t in the signal f is to evaluate the NCC value (coefficient) at each point (u), which has been shifted by u steps in the x direction. The maximum values or peaks of the computed correlation values indicate the matches between the template (pattern) t and the signal f. The NCC used for finding matches of a template $t(x)$ of size N in a signal $f(x)$ of size M is defined as

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

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

We refer to (1) as "direct NCC" and the value $\gamma_u(\cdot)$ as the "NCC coefficient". Due to the normalization by the denominator in (1), the direct NCC is more robust than other similarity measures, like simple covariance or the sum of absolute differences (SAD). Nevertheless the main drawback is that the calculation of (1) is computationally expensive. The number of computations required to calculate the NCC coefficients is $2N(M-N+1)$ multiplications and $3N(M-N+1)$ additions/subtractions. To overcome these complexity problems, an efficient method to calculate the denominator of (1) was proposed by Lewis [8]. He presented an algorithm for fast calculation of the normalized correlation using two pre-computed sum tables over $f(x)$ and $f^2(x)$. Lewis's sum-table approach can be used to efficiently calculate the denominator of (1). However, it cannot be directly applied to compute the numerator of (1). The numerator of (1) can be calculated in the frequency domain using the well-known Fourier transform and its computational complexity is $6M\log_2 M$ multiplications and $9M\log_2 M$ additions/subtractions. Even though Lewis's algorithm provides an order of magnitude speedup over the spatial domain computation of NCC, the number of computations is still comparatively high. Therefore, further simplification of this calculation is required. In Sect. 3, the key idea that allows a very efficient calculation of the NCC is explained in detail.

3 Fast NCC Algorithm

In this study, it is assumed that all signals are real valued and the template $t(x)$ is pre-normalized to be between 0 and 1.

3.1 Derivation of the Fast NCC Coefficient

Let us assume that

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

where α is a positive constant that represents a scaling factor.

Under the assumption of (2), we get $\gamma_u(\cdot) = 1$ and $\overline{f_u} = \alpha \cdot \bar{t}$ and then the NCC coefficient in (1) can be computed as follows:

$$\begin{aligned}
\gamma_u(f,t) &= \frac{\sum_x [f(x) - \overline{f}_u][t(x-u) - \bar{t}]}{\sqrt{\sum_x [f(x) - \overline{f}_u]^2 \sum_x [t(x-u) - \bar{t}]^2}} \\
&= \frac{\sum_x [\alpha \cdot t(x-u) - \alpha \cdot \bar{t}][t(x-u) - \bar{t}]}{\sqrt{\sum_x [\alpha \cdot t(x-u) - \alpha \cdot \bar{t}]^2 \sum_x [t(x-u) - \bar{t}]^2}} \\
&= \frac{\alpha \cdot \sum_x [t(x-u) - \bar{t}][t(x-u) - \bar{t}]}{\alpha \cdot \sum_x [t(x-u) - \bar{t}]^2} \\
&= \frac{\sum_x [t(x-u) - \bar{t}]^2}{\sum_x [t(x-u) - \bar{t}]^2}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{N}\sum_x \frac{[t(x-u)-\bar{t}]^2}{[t(x-u)-\bar{t}]^2} \\
&= \frac{1}{N}\sum_x \frac{[t(x-u)-\bar{t}]\cdot[t(x-u)-\bar{t}]}{|t(x-u)-\bar{t}|\cdot|t(x-u)-\bar{t}|}.
\end{aligned}
\tag{3}
$$

By using the assumption of (2), (3) can be written as follows:

$$
\begin{aligned}
\gamma_u(f,t) &= \frac{1}{N}\sum_x \frac{[f(x)/\alpha-\overline{f}_u/\alpha][t(x-u)-\bar{t}]}{\left|f(x)/\alpha-\overline{f}_u/\alpha\right|\left|t(x-u)-\bar{t}\right|} \\
&= \frac{1}{N}\sum_x \frac{[f(x)-\overline{f}_u][t(x-u)-\bar{t}]}{|f(x)-\overline{f}_u||t(x-u)-\bar{t}|} \\
&= \frac{1}{N}\left[\frac{[f(x_1)-\overline{f}_u]\cdot[t(x_1-u)-\bar{t}]}{|f(x_1)-\overline{f}_u|\cdot|t(x_1-u)-\bar{t}|}+\cdots\right. \\
&\quad \left.+\frac{[f(x_N)-\overline{f}_u]\cdot[t(x_N-u)-\bar{t}]}{|f(x_N)-\overline{f}_u|\cdot|t(x_N-u)-\bar{t}|}\right],
\end{aligned}
\tag{4}
$$

where $x_1, x_2, \ldots$ and x_N denote x coordinates in the region under the template.

(4) can be rewritten as

$$
\gamma_u(f,t) = \frac{1}{N}\sum_x \lambda_u(x) \tag{5}
$$

where the following definitions have been used:

$$
\lambda_u(x) \triangleq \hat{f}_u(x)\cdot\hat{t}_u(x), \tag{6a}
$$

$$
\hat{f}_u(x) \triangleq \frac{a_u(x)}{|a_u(x)|}, \tag{6b}
$$

$$
\hat{t}_u(x) \triangleq \frac{b_u(x)}{|b_u(x)|}, \tag{6c}
$$

$$
a_u(x) \triangleq f(x)-\overline{f}_u, \tag{6d}
$$

$$
b_u(x) \triangleq t(x-u)-\bar{t}. \tag{6e}
$$

If the formula in (5) is used even when the assumption (2) may not be valid, the result is an approximation to the NCC coefficient. We will refer to this approximation as "the fast NCC coefficient" denoted as $\Lambda_u(f,t)$, which is given by:

$$
\Lambda_u(f,t) \triangleq \frac{1}{N}\sum_x \lambda_u(x). \tag{7}
$$

Note from (7) that:

(i) $\hat{f}_u(x)$ and $\hat{t}_u(x)$ have only the values of either $+1$ or -1. Therefore, we can use the following expressions instead of (6b) and (6c).

$$\hat{f}_u(x) = \begin{cases} +1 & \text{if } a_u \geq 0, \\ -1 & \text{otherwise,} \end{cases} \tag{8}$$

$$\hat{t}_u(x) = \begin{cases} +1 & \text{if } b_u \geq 0, \\ -1 & \text{otherwise.} \end{cases} \tag{9}$$

(ii) Since $\lambda_u(x)$ can be obtained by performing the logic "AND" operation between $\hat{f}_u(x)$ and $\hat{t}_u(x)$, its computation is trivial.

(iii) If $f(x) = \alpha \cdot t(x-u)$ and thus $\hat{f}_u(x) = \hat{t}_u(x)$, then $\lambda_u(x)$ has only the value of $+1$, resulting in $\Lambda_u(\cdot) = 1$.

(iv) The fast NCC coefficient satisfies the constraint that $-1 \leq \Lambda_u(\cdot) \leq 1$, and $\Lambda_u(\cdot) = 1$ if and only if $f(x) = \alpha \cdot t(x-u)$. Note that this constraint is the same as that of $\gamma_u(\cdot)$.

3.2 Noise Sensitivity of the Fast NCC Coefficient

The main drawback of the fast NCC coefficient is that it is very sensitive to noise. As you can see from (6d) and (8), $\hat{f}_u(x)$ has transitions from $+1$ to -1 or from -1 to $+1$ around the value $f(x) = \hat{f}_u$. For example, let us consider a signal $f(x)$ given by $f(x) = t(x-u) + n(x)$ where $n(x)$ is additive white Gaussian noise (AWGN) with mean zero; then $\hat{f}_u(x)$ is given by

$$\hat{f}_u(x) = \frac{t(x-u) + n(x) - \bar{t}}{|t(x-u) + n(x) - \bar{t}|}. \tag{10}$$

The signal $\hat{f}_u(x)$ is very sensitive to noise $n(x)$ around the transition regions where $t(x-u) \approx \bar{t}$. That is, $\hat{f}_u(x)$ can be either $+1$ or -1, depending on $n(x)$. As a result, the coefficient $\Lambda_u(\cdot)$ can be quite different from the coefficient when the noise $n(x)$ is absent.

One method of decreasing the noise sensitivity is to use a noise threshold η_t around the transition regions in calculating the signal $\hat{f}_u(x)$.

Here, consider $f(x)$ scaled by a scaling factor α, then $\hat{f}_u(x)$ is given by

$$\begin{aligned} \hat{f}_u(x) &= \frac{\alpha \cdot t(x-u) + \alpha \cdot n(x) - \alpha \cdot \bar{t}}{|\alpha \cdot t(x-u) + \alpha \cdot n(x) - \alpha \cdot \bar{t}|} \\ &= \frac{k(x-u) + \alpha \cdot n(x) - \bar{k}}{|k(x-u) + \alpha \cdot n(x) - \bar{k}|} \end{aligned} \tag{11}$$

where $k(x-u) \overset{\Delta}{=} \alpha \cdot t(x-u)$ and $\bar{k} \overset{\Delta}{=} \alpha \cdot \bar{t}$.

Note that the larger the scaling factor α is, the more sensitive to $n(x)$ the signal $\hat{f}_u(x)$ is. This is because the noise $n(x)$ is random and thus the signal $\hat{f}_u(x)$ is changeable with the change of $n(x)$ even though $k(x-u)$ and $\bar{k}$ are constant. By comparing (10) with (11), it is seen that the value $\hat{f}_u(x)$ in (11) is about α times more sensitive to noise $n(x)$ than that in (10). Therefore, the noise threshold η_t should be decided according to the scaling factor α.

3.3 Proposed Fast NCC Algorithm

Given two signals, $f(x)$ and $t(x)$, the steps of the proposed fast NCC algorithm include the following:

Step 1: First, compute $\hat{f}_u(x)$ and $\hat{t}_u(x)$ and initialize $F^o(x)$ and $T^o(x)$.

$$\hat{f}_u(x) = \begin{cases} +1 & \text{if } a_u(x) \geq 0, \\ -1 & \text{otherwise,} \end{cases} \tag{12}$$

$$\hat{t}_u(x) = \begin{cases} +1 & \text{if } b_u(x) \geq 0, \\ -1 & \text{otherwise,} \end{cases} \tag{13}$$

$$F^o(x) = \hat{f}_u(x); \qquad T^o(x) = \hat{t}_u(x). \tag{14}$$

Note that $\hat{t}_u(x)$ can be pre-computed. The following steps 2 through 5 include the processes to obtain a robust $\hat{f}_u$ to noise. In step 2, the scaling factor α is estimated based on the average absolute deviation (AAD). In step 3, the noise threshold based on the scaling factor α is calculated. Step 4 finds the transition regions of $\hat{f}_u(x)$ and $\hat{t}_u(x)$ by obtaining the waveforms $T^-(x)$, $T^+(x)$, $F^-(x)$ and $F^+(x)$. In step 5, a robust $F^o(x)$ to noise is obtained by considering the noise threshold at the transition regions.

Step 2: Estimate the scaling factor α by calculating the estimator α_{AAD}.
First, compute the AAD from the mean of the signal

$$\delta_f = \frac{1}{N}\sum_x \left|a_u(x)\right|, \tag{15}$$

$$\delta_t = \frac{1}{N}\sum_x \left|b_u(x)\right| \tag{16}$$

where δ_f and δ_t, respectively, denote the AAD of $f(x)$ and $t(x)$.

Next, compute the ratio (α_{AAD}) between δ_f and δ_t.

$$\alpha_{\text{AAD}} = \delta_f/\delta_t \tag{17}$$

which reduces to

$$\alpha_{\text{AAD}} = \frac{\sum_x |a_u(x)|}{\sum_x |b_u(x)|}. \tag{18}$$

In this paper, α_{AAD} is used as the estimator of the scaling factor α.

Note that $\sum_x |b_u(x)|$ can be pre-computed. As an alternative, a pre-computed look-up table, indexed by $\sum_x |a_u(x)|$ and $\sum_x |b_u(x)|$, for (18) can be used by which the computations required can be trivial.

Step 3: Compute the noise threshold η_t whose value is proportional to the scaling factor α to decrease the noise sensitivity, as early stated in (11):

$$\eta_t = \begin{cases} k_1 - k_2 \cdot (1 - \alpha_{\text{AAD}}) & \text{if } \delta_f < \delta_t, \\ k_1 + k_2 \cdot (\alpha_{\text{AAD}} - 1) & \text{otherwise} \end{cases} \tag{19}$$

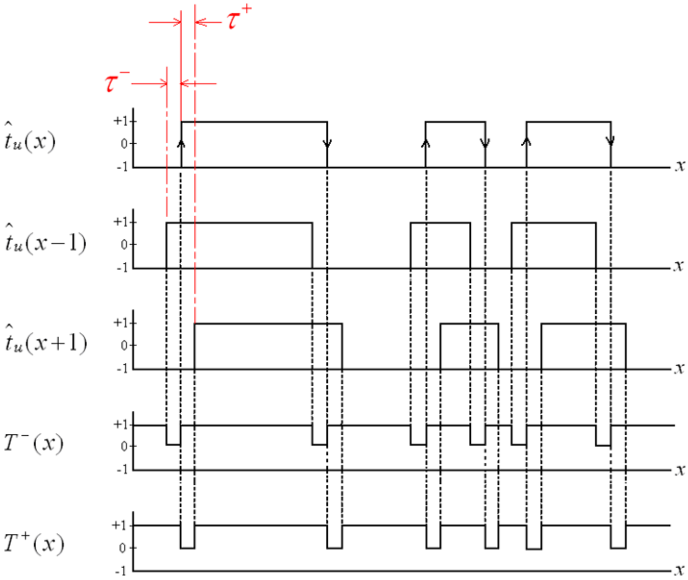


Fig. 1 An example explaining how to compute the waveforms $T^-(x)$ and $T^+(x)$ with $\hat{t}_u(x)$

where k_1 indicates the value of the noise threshold when $\alpha_{\text{ADD}} = 1$ and k_2 is the proportional constant.

Provided that k_1 and k_2 are chosen properly, a pre-computed look-up table, indexed by α_{AAD}, for (19) can be used by which the computations required can be trivial.

Step 4: Find the transition regions of $\hat{f}_u(x)$ and $\hat{t}_u(x)$ by obtaining the waveforms $T^-(x)$, $T^+(x)$, $F^-(x)$ and $F^+(x)$, which are given by

$$\begin{aligned} T^-(x) &= \hat{t}_u(x-1) \otimes \hat{t}_u(x); \qquad & T^+(x) &= \hat{t}_u(x) \otimes \hat{t}_u(x+1), \\ F^-(x) &= \hat{f}_u(x-1) \otimes \hat{f}_u(x); \qquad & F^+(x) &= \hat{f}_u(x) \otimes \hat{f}_u(x+1) \end{aligned} \tag{20}$$

where the operator $\otimes$ denotes the logic "Exclusive-NOR" operation, and the upper suffix symbol "$-$" means the waveforms for finding pre-transitions and "$+$" for post-transitions.

Figure 1 shows an example of how to compute $T^-(x)$ and $T^+(x)$ with a given $\hat{t}_u(x)$. The transition regions consist of pre- and post-transitions in the regions before and after transition edges. From Fig. 1, it can be noted that: $T^-(x)$ has the value zero at the pre-transitions(τ^-) of $\hat{t}_u(x)$ while $T^+(x)$ has the value zero at the post-transitions (τ^+) of $\hat{t}_u(x)$. Similarly, $F^-(x)$ has the value zero at the pre-transitions (τ^-) of $\hat{f}_u(x)$ while $F^+(x)$ has the value zero at the post-transitions (τ^+) of $\hat{f}_u(x)$.

Step 5: Find a robust $F^o(x)$ to noise by the following rules:

$$F^o(x) = \begin{cases} \hat{t}_u(x) & \text{if } (A \vee B) == 1, \\ \hat{f}_u(x) & \text{otherwise} \end{cases} \tag{21}$$

where the parameters A and B are logical ones which are given by

$$\begin{aligned} A &= \bigl(\lvert b_u(x)\rvert < \eta_t\bigr) \wedge \bigl(T^+(x) == 0 \vee T^-(x) == 0\bigr), \\ B &= \bigl(\lvert a_u(x)\rvert < \eta_t\bigr) \wedge \bigl(\lvert b_u(x)\rvert < \eta_t\bigr) \wedge \bigl(F^+(x) == 0 \vee F^-(x) == 0\bigr) \end{aligned} \tag{22}$$

and the operators $\wedge$ and $\vee$ denote the logic "AND" and logic "OR" operation, respectively.

In this paper, the logic parameters A and B that satisfy the constraints such that $|b_u(x)| < \eta_t$ and $(|a_u(x)| < \eta_t) \odot (|b_u(x)| < \eta_t)$ at the pre-transitions or post-transitions are used to decrease the noise sensitivity.

Step 6: Finally, compute the fast NCC coefficient:

$$\Lambda_u(f,t) = \frac{1}{N}\sum_x T^o(x) \cdot F^o(x) \tag{23}$$

where its computation is trivial since $T^o(x)$ and $F^o(x)$consist only of the values $+1$ and -1.

Repeat the above steps 1–6 with $u = u + 1$ until u reaches $(M - N)$.

3.4 The Number of Computations Required to Calculate the Fast NCC Coefficient

For a signal of size M and a template of size N the approximate number of computations required at each step in Sect. 3.3 is given in Table 1.

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 the Fourier transform and the Lewis algorithm [8]. Table 2 shows the comparison of arithmetic operations between the proposed method and the traditional normalized correlation techniques. For a search window of size M and a template of size N the fast NCC algorithm requires approximately $2N \cdot (M - N + 1)$ additions/subtractions and zero 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 the fast Fourier transform (FFT) is obtained by calculating only the numerator of (1) in the frequency domain. Therefore, the cross-correlation is not normalized. In addition, this approach can be less efficient than the direct implementation of correlation in the

Table 1 Number of computations for calculating $\Lambda_u(\cdot)$

	Addition/Subtraction	Multiplication	
Step 1$^{(*)}$	$N \cdot (M - N + 1)$	0	Equation (12)
Step 2$^{(*)}$	$N \cdot (M - N + 1)$	0	Equation (18)
Step 3	0	0	Equation (19)
Step 4$^{(*)}$	0	0	Equation (20)
Step 5$^{(*)}$	0	0	Equation (21)
Step 6	N	0	Equation (23)

(*): It is assumed that $\hat{t}_u(x)$ and $\sum_x |b_u(x)|$ can be pre-computed, and the logic computations performed due to their triviality are not included

Table 2 The comparison of arithmetic operations between the proposed method and the traditional normalized correlation techniques

	Addition/Subtraction		Multiplication	
	Numerator	Denominator	Numerator	Denominator
Direct NCC	$N(M-N+1)$	$2N(M-N+1)$	$N(M-N+1)$	$N(M-N+1)$
FFT	$9M\log_2 M$ (FFT)	$2N(M-N+1)$ (Direct)	$6M\log_2 M$ (FFT)	$N(M-N+1)$ (Direct)
Lewis(*)	$9M\log_2 M$ (FFT)	$3(M-N+1)$ (Two sum tables)	$6M\log_2 M$ (FFT)	0 (Two sum tables)
Fast NCC	$2N\cdot(M-N+1)$		0	

(*): The extra computations required to set up the sum tables are not included

spatial domain especially when the duration of the template $t(x)$ is usually much shorter than that of the signal $f(x)$ [5].

Lewis's 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 domain with the FFT algorithm. Furthermore, Lewis's algorithm requires approximately $3M$ operations for the construction of the two sum tables [8].

4 Example

This section shows an explanation of the procedure for calculating the fast NCC coefficient. Consider a signal $f(x)$ such that

$$f(x)=t(x)+n(x). \tag{24}$$

An example of a template $t(x)$ is shown in Fig. 2 where the template $t(x)$ is corrupted by AWGN of $SNR=10$ dB, resulting in $f(x)$. Given the two signals, $f(x)$ and $t(x)$, the steps of the proposed fast NCC algorithm include the following:

(i) First, compute the values $F^o(x)$ and $T^o(x)$ given in (14). These values are shown in the left column of Fig. 3(a) and (b).
(ii) Compute the noise threshold, η_t, given in (19). In Fig. 2, the solid lines indicate $\overline{t}$ and $\overline{f}_u$ while the dotted lines mark $\overline{t}\pm\eta_t$ and $\overline{f}_u\pm\eta_t$.
(iii) Find the transition regions by calculating (20).
(iv) Next, find a robust $F^o(x)$ to noise by considering the noise thresholds at the transition regions. The signal $F^o(x)$ based on (21) is shown in the right column of Fig. 3(b).
(v) Finally, compute the fast NCC coefficient given in (23).

The resulting $T^o\cdot F^o$ is shown in Fig. 3(c). By comparing left column with right column in Fig. 3(c), we can see how the noise sensitivity can be reduced by using

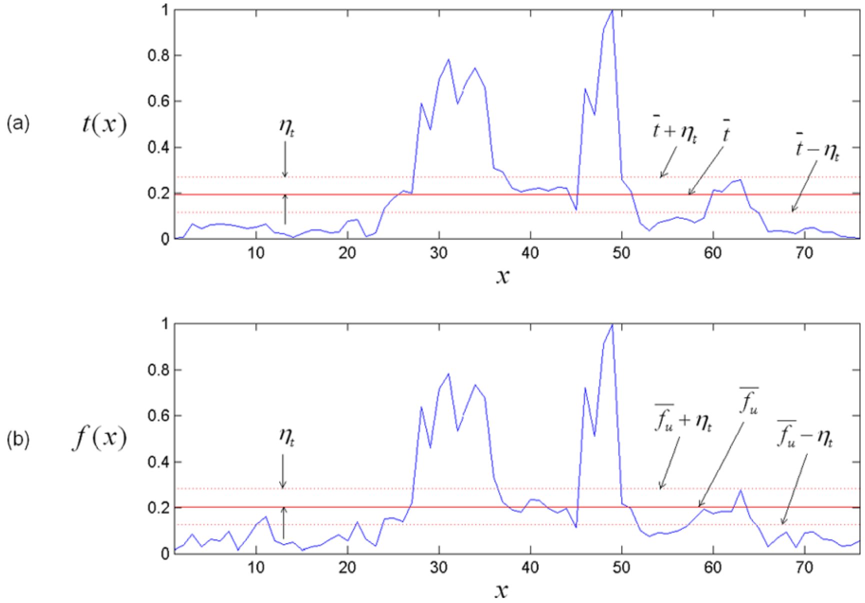


Fig. 2 An example of a template $t(x)$ corrupted by AWGN of $SNR = 10$ dB, resulting in $f(x)$. The NCC coefficient $\gamma_u(f,t)$ in the matching position is 0.99. (**a**) A template $t(x)$ where the *upper dotted line* indicates $\bar{t} + n_t$ and the *lower one* indicates $\bar{t} - n_t$. (**b**) The corresponding signal $f(x)$, where the *upper dotted line* indicates $\bar{f}_u + n_t$ and the *lower one* indicates $\overline{f}_u - \eta_t$

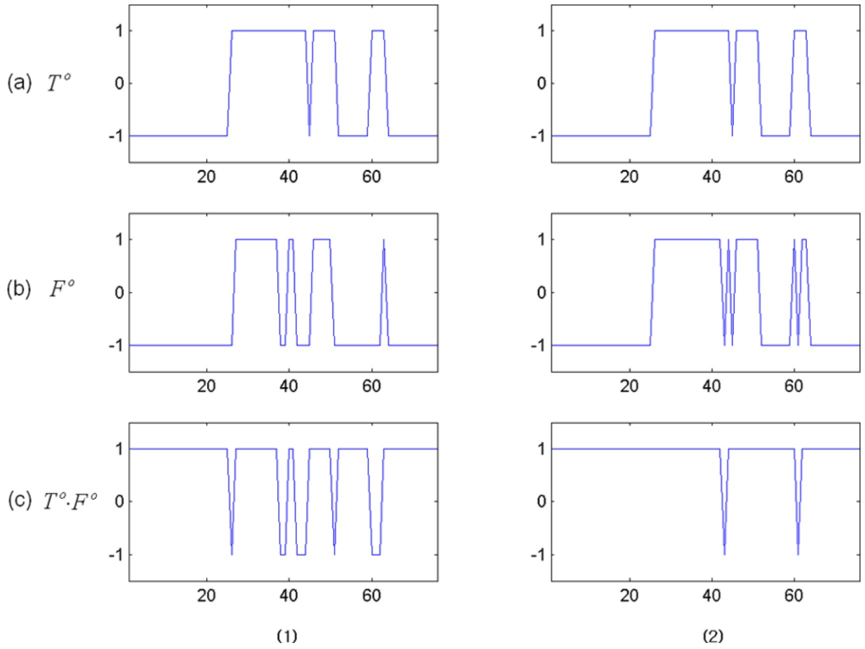


Fig. 3 An example illustrating reduction of noise sensitivity

(21), resulting in $\Lambda_u(\cdot)$, defined in (23) whose value is very close to that of $\gamma_u(\cdot)$, defined in (1).

(1) The figures in the left column indicate the results when steps 2 through 5 in Sect. 3.3 are skipped. In this case, F^o is sensitive to noise. As a result, the value $\Lambda_u(\cdot)$ is far away from the value of $\gamma_u(\cdot)$. (i.e., $\Lambda_u(\cdot) = 0.74, \gamma_u(\cdot) = 0.99$): (a) result of T^o using (14), (b) result of F^o using (14), (c) result of $T^o \cdot F^o$.

(2) The figures in the right column indicate the results when steps 2 through 5 in Sect. 3.3 are performed. In this case, F^o is insensitive to noise and thus the value $\Lambda_u(\cdot)$ is close to the value of $\gamma_u(\cdot)$. (i.e., $\Lambda_u(\cdot) = 0.95, \gamma_u(\cdot) = 0.99$): (a) result of T^o using (14), (b) result of F^o using (21), (c) result of $T^o \cdot F^o$.

5 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 index $i = \{1, 2, \ldots, 1000\}$. These templates were used to examine the performance of the presented approach.

5.1 Generating the Template $t_i(x)$ and the Signal $f_i(x)$

Ben-Aire and Rao proposed a method for 1-D signal decomposition by Gaussian basis functions and thereby the 1-D signal can be approximated by a set of Gaussian basis functions [1]. In this paper, to generate various templates, a Gaussian pulse was considered:

$$g_n(x) = A_n \cdot e^{-(x/\sigma_n)^2} \quad \text{for } 0 \le x \le N-1 \tag{25}$$

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:

```
Initialization : $t_i(x) = 0$

For $n = 1 : 1 : P$
    $temp = t_i(x) + g_n(x - \tau)$
    $t_i(x) = \frac{temp}{\max(temp)}$
End
```
(26)

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 $\gamma_u(t_i, t_j | i \ne j.) \ge TH_{\text{roi}}$ was excluded from consideration, where the value TH_{roi} is a threshold for identification. Some examples of these templates with various profiles are shown in the left column in Fig. 4. Also, the corresponding signals $f_i(x)$ to the templates are shown in the right column in the same row in Fig. 4. The signals $f_i(x)$ with various SNRs from -3 to 10 dB were generated from the templates as follows:

$$f_i(x) = \alpha \cdot t_i(x) + AWGN \tag{27}$$

where $f_i(x)$ was labeled with the corresponding template index i for verification, and the scaling factor α was taken in the range of [0.1, 10] with increments of 0.1. The signal $f_i(x)$ for each database entry to check the correlation performance was created, and thus there are a total of 100,000 $f_i(x)$ for a given SNR. A test signal $f(x)$ of size M can be generated by using AWGN of size M from (27), where $M > N$ is assumed.

In this simulation, the size N between 70 and 90 was chosen at random, while M was set to $4 \cdot N$. Figure 5 shows a typical test signal $f(x)$ which includes $f_i(x)$ together with background noise depending on SNR.

5.2 The choice of TH_{roi}

Figure 6 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 coefficients decrease as the SNR decreases. This figure provides information on how to choose the suitable threshold TH_{roi}, which is given by:

$$\begin{aligned} TH^{\gamma}_{\mathrm{roi}} &= \overline{\gamma_u(f_i, t_i)} - \sigma_\gamma \quad \text{for the direct NCC,} \\ TH^{\Lambda}_{\mathrm{roi}} &= \overline{\Lambda_u(f_i, t_i)} - \sigma_\Lambda \quad \text{for the fast NCC,} \end{aligned} \tag{28}$$

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 the TH_{roi} is. The values of TH_{roi} versus the *SNR* are stored in the form of a look-up table and are used when (29) and (30) are evaluated.

5.3 Measures of Performance

The algorithm described in Sect. 3.3 was applied to the test signal $f(x)$ to obtain the coefficient $\Lambda_u(\cdot)$. Figure 7 shows the comparison between $\gamma_u(\cdot)$ and $\Lambda_u(\cdot)$ at various SNR levels. Figure 7(a) indicates the correlation coefficients between $t_i(x)$ and $f_i(x)$ while Fig. 7(b) indicates those between $t_i(x)$ and $f_j(x)$ for an index i such that $i \neq j$. From the observation of Fig. 7, it is seen that $\Lambda_u(\cdot)$ follows $\gamma_u(\cdot)$ with fidelity and thus the proposed coefficient $\Lambda_u(\cdot)$ can be used as the similarity metric between two signals.

For a given test signal $f(x)$, all entries in the database are scanned through and then the indices of the database whose matching score is higher than $TH^{\Lambda}_{\mathrm{roi}}$ are listed as candidates.

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

$$\Lambda_u(f, t_i) \geq TH^{\Lambda}_{\mathrm{roi}} \tag{29}$$

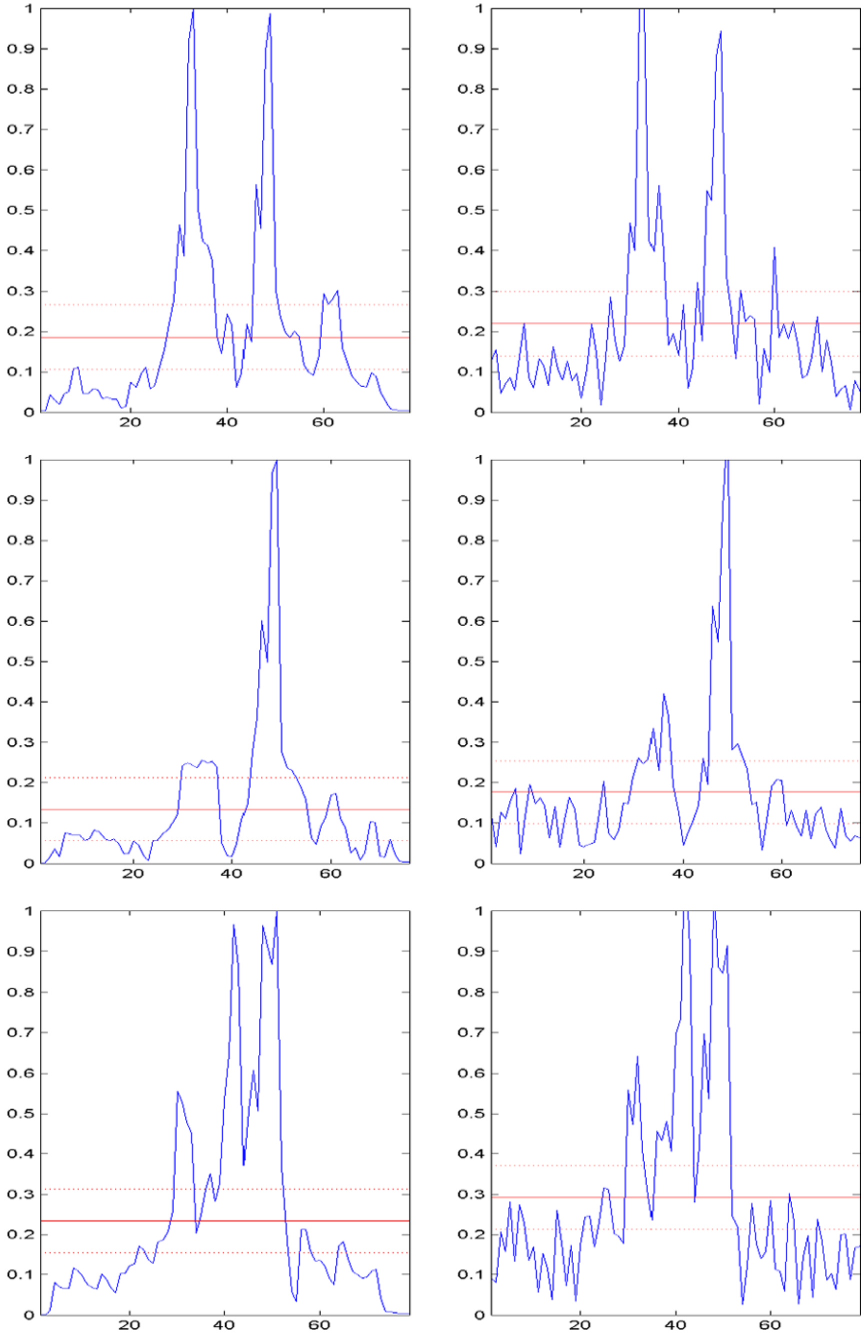


Fig. 4 Some examples of $t_i(x)$ (*left column*) and the corresponding $f_i(x)$ (*right column*: *SNR* $= 3$ dB, $\alpha = 1$). The *dotted lines* and *solid lines* have the same meaning as that of Fig. 2. y-axis indicates $t_i(x)$ (*left column*) and the corresponding $f_i(x)$ (*right column*)

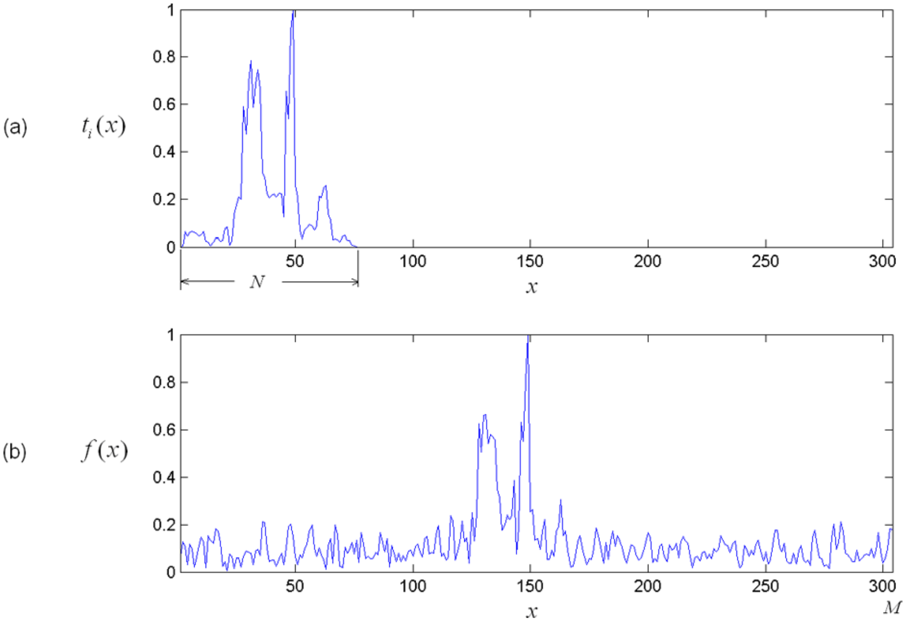


Fig. 5 An example of a test signal $f(x)$ which includes $f_i(x)$ together with background noise (3 dB AWGN). (**a**) a template $t_i(x)$ and (**b**) the corresponding test signal $f(x)$

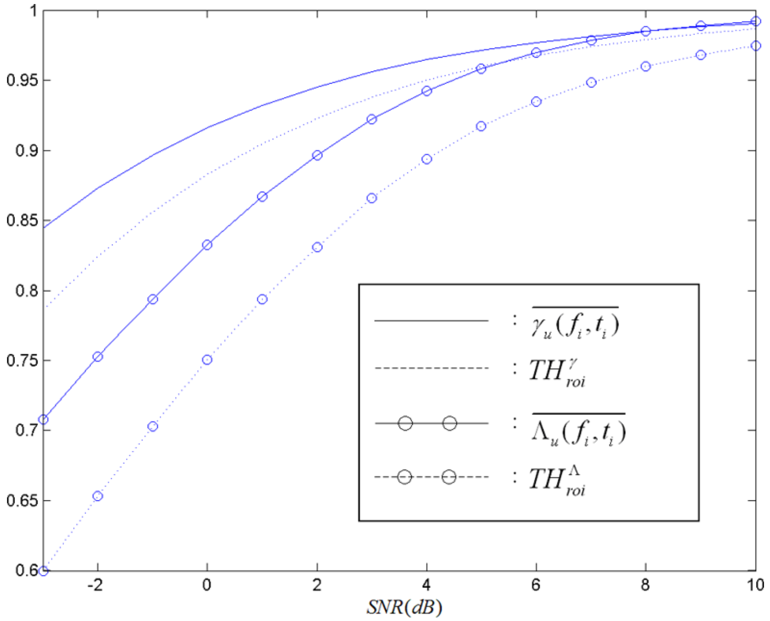


Fig. 6 NCC coefficient versus SNR

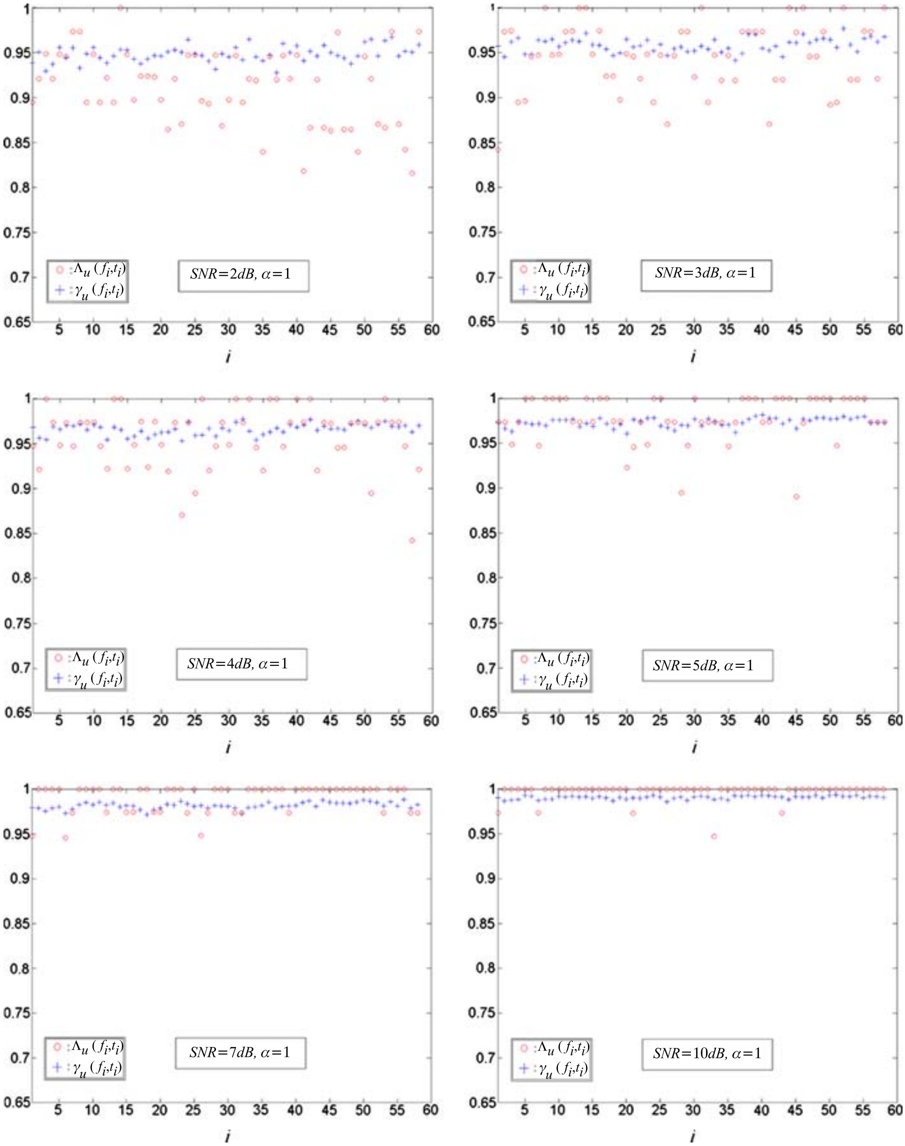


Fig. 7 Comparison of the coefficients $\gamma_u(\cdot)$ and $\Lambda_u(\cdot)$. (**a**) Normalized correlation coefficients between $t_i(x)$ and $f_i(x)$. (**b**) Normalized correlation coefficients between $t_i(x)$ and $f_i(x)$ for a given i such that $i \neq j$

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 (29) the results will be listed in descending order of matching score. The higher the score, the more likely it is to be 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

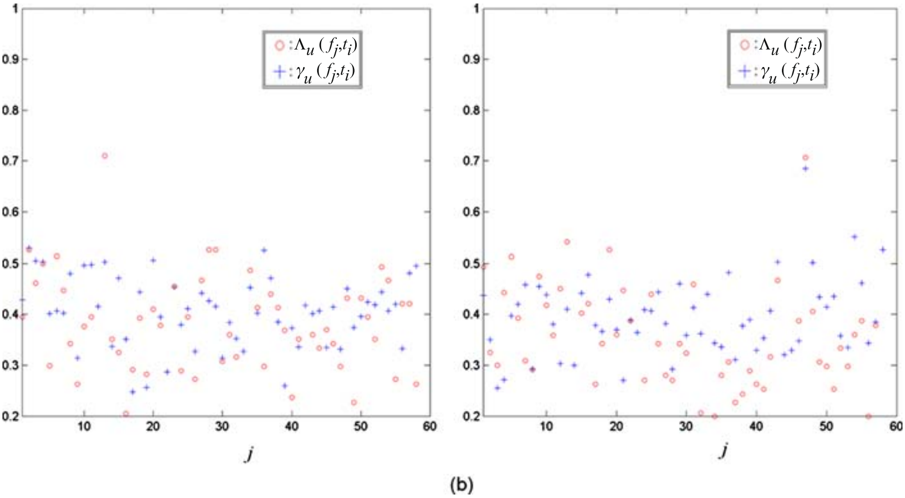


Fig. 7 (*Continued*)

matching score. The number of candidates will be increased rapidly by FPs as the SNR decreases. This increase is because the lower the SNR is, the smaller value the threshold $TH_{\mathrm{roi}}^{\Lambda}$ should be set to (see Fig. 6). In this paper, the direct NCC is used to filter out the unsuitable candidates from the set $\mathbb{R}$ and such a filtering process can considerably reduce the FPs because of $TH_{\mathrm{roi}}^{\gamma} > TH_{\mathrm{roi}}^{\Lambda}$.

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

$$\gamma_{u\in\mathbb{R}}(f, t_i \mid i \in \mathbb{R}) \geq TH_{\Omega 1} \tag{30}$$

where $TH_{\Omega 1} = TH_{\mathrm{roi}}^{\gamma}$.

Note that the computational burden in obtaining $\Omega_u(\cdot)$ the entirely depends on the size of the set $\mathbb{R}$ (i.e., the SNR). In this paper, in order to satisfy the rate of overall identification as high as possible with the minimum number of false alarms, (29) and (30) were experimentally adjusted with an ad hoc method (refer to Appendix) as follows:

$$A_u(f, t_i) \geq TH_{\mathrm{roi}}^{\mathrm{A}} - (s/500), \tag{31}$$

$$\gamma_{u\in\mathbb{R}}(f, t_i \mid i \in \mathbb{R}) \geq TH_{\Omega 2} \tag{32}$$

where $TH_{\Omega 2} = TH_{\mathrm{roi}}^{\gamma} - (s/200)$ and the value s is an integer number which is obtained by one-to-one mapping between SNR (dB) $= -3, -2, \ldots, 10$ and the integer numbers $1, 2, 3 \ldots, 14$.

Some comparisons were made between the algorithms based on the three coefficients $\gamma_u(\cdot)$, $\Lambda_u(\cdot)$ and $\Omega_u(\cdot)$ with the test signal $f(x)$ and then the performance of each algorithm in terms of FP/FN rate and identification rate was evaluated. The FP rate is the proportion of negatives cases that were incorrectly classified as positive while the FN rate is the proportion of positives cases that were incorrectly identified

Table 3 Comparison of the performance, averaged over scaling factor, between the three coefficients when $SNR = 10$ dB

	Coefficient	Identification rate, see Fig. 14(a)	FN rate, see Fig. 8(a)	FP rate, see Fig. 11(a)
Direct NCC	$\gamma_u(\cdot)$	87.4%	12.6%	0.0%
Fast NCC	$\Lambda_u(\cdot)$	96.32%	3.68%	5.57%
Fast NCC using the filtering process	$\Omega_u(\cdot)$	96.32%	3.68%	0.74%

as negative. The identification rate is defined as the percentage of successful detections for all trials.

Figure 8 shows the average FN rate while Fig. 11 shows the average FP rate. Figure 9 shows FN rate versus SNR as a function of scaling factor while Fig. 10 shows FN rate versus scaling factor as a function of SNR. It can be seen from Fig. 8(a) that $\Lambda_u(\cdot)$ and $\Omega_u(\cdot)$ give relatively better FN rates than $\gamma_u(\cdot)$ for high *SNR* (>3 dB).

The reason that $\Lambda_u(\cdot)$ gives better FN rates than $\gamma_u(\cdot)$ is because $TH_{\text{roi}}^{\Lambda}$ is less than TH_{roi}^{γ}. Also, the reason that $\Omega_u(\cdot)$ gives better FN rates than $\gamma_u(\cdot)$ is because the threshold, $TH_{\Omega 2}$, used in the filtering process (32) is less than the threshold, TH_{roi}^{γ}, used for identification in the direct NCC (28).

Figure 12 shows FP rate versus SNR as a function of scaling factor while Fig. 13 shows FP rate versus scaling factor as a function of SNR. From Fig. 11 we have the observations that the filtering process in (32) gives significant improvement in FP rate and yields almost the same FP rate as in the direct NCC for all SNR conditions and various scaling factors. Note that there was no significant difference in FP rate between the $\Lambda_u(\cdot)$ and $\Omega_u(\cdot)$ for high SNR, compared to that in FP rate for low *SNR*.

Figure 14(a) and (b) are graphs that plot the average identification rate as a function of SNR and scaling factor, respectively. Figure 15 shows identification rate versus SNR as a function of scaling factor while Fig. 16 shows identification rate versus scaling factor as a function of SNR. As could be expected, it is noted that when the SNR is higher, the templates can be better identified. It can be seen from Figs. 14, 15 and 16 that the fast NCC algorithm based on $\Lambda_u(\cdot)$ and $\Omega_u(\cdot)$ can outperform the direct NCC algorithm in identification performance, especially when the SNR is greater than 3 dB. Table 3 compares the performance between the three coefficients when $SNR = 10$ dB. The identification rate by $\Omega_u(\cdot)$ is about 96.32% as opposed to 87.4% by $\gamma_u(\cdot)$. As shown in this table, the identification results using $\Omega_u(\cdot)$ are higher than those obtained by $\gamma_u(\cdot)$ while keeping almost the same FP rate and a lower FN rate than in the direct NCC. This is because the threshold, $TH_{\Omega 2}$, used in the filtering process (see (32)) is less than that used for identification in the direct NCC (see (28)). Nevertheless, there was no significant difference in FP rate between them.

The simulation results can be summarized as follows:

(1) It can be seen that the identification rate of the fast NCC, based on $\Omega_u(\cdot)$, is better than that of the direct NCC for various scaling factors, especially when the SNR is higher than 3 dB. Nevertheless, (i) there is no significant difference in FP rate between the fast NCC and the direct NCC, (ii) the computational burden is

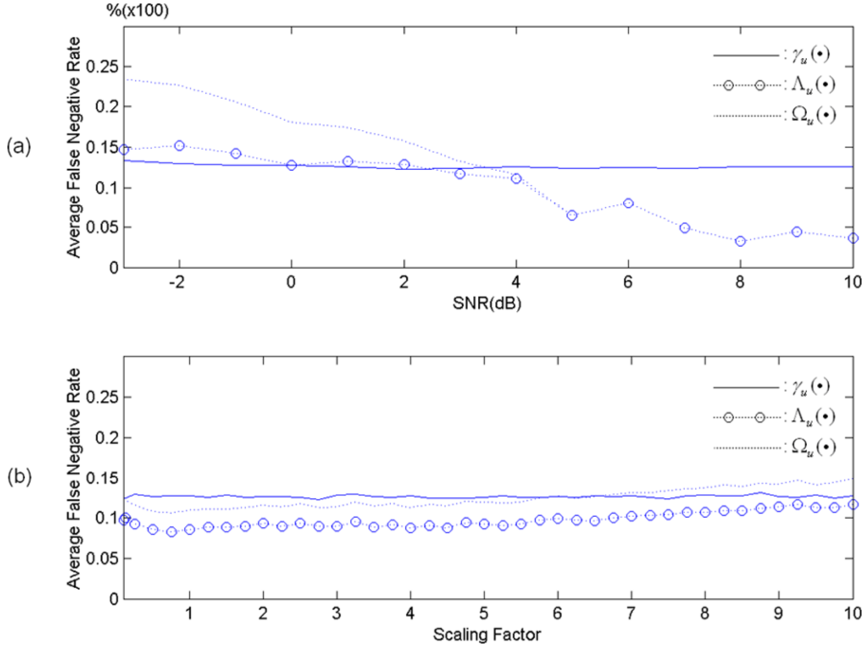


Fig. 8 Comparison of average FN rate between the three coefficients. (**a**) FN rate versus SNR, averaged over scaling factor between 0.1–10. (**b**) FN rate versus scaling factor, averaged over SNR between −3–10 dB

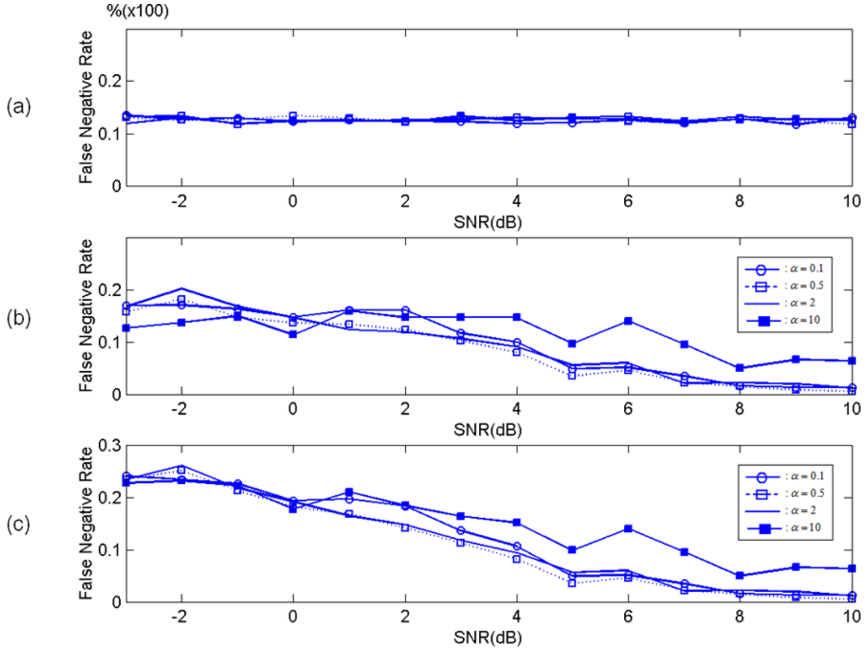


Fig. 9 Plots of FN rate versus SNR with various scaling factor α. (**a**) $\gamma_u(\cdot)$, (**b**) $\Lambda_u(\cdot)$, (**c**) $\Omega_u(\cdot)$

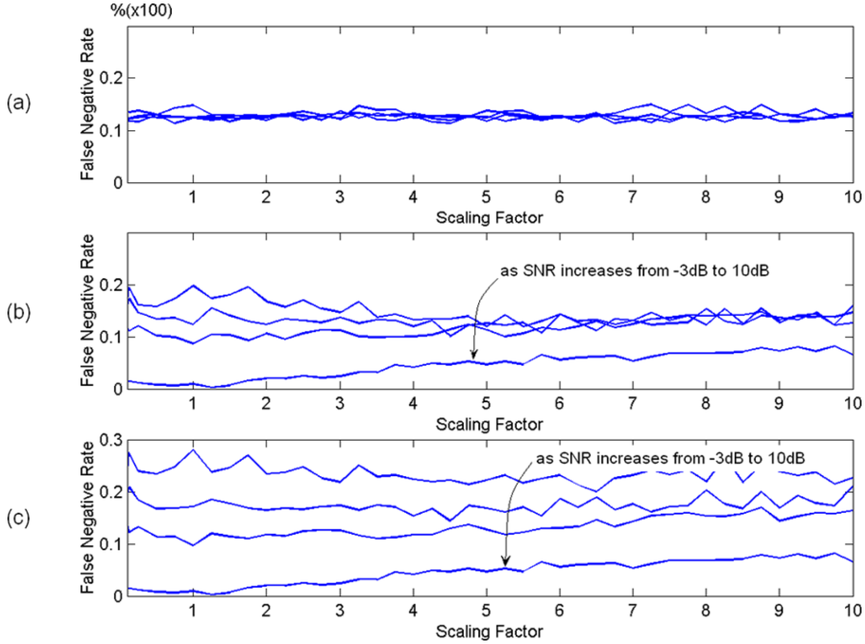


Fig. 10 Plots of FN rate versus scaling factor as a function of SNR. (**a**) $\gamma_u(\cdot)$, (**b**) $\Lambda_u(\cdot)$, (**c**) $\Omega_u(\cdot)$

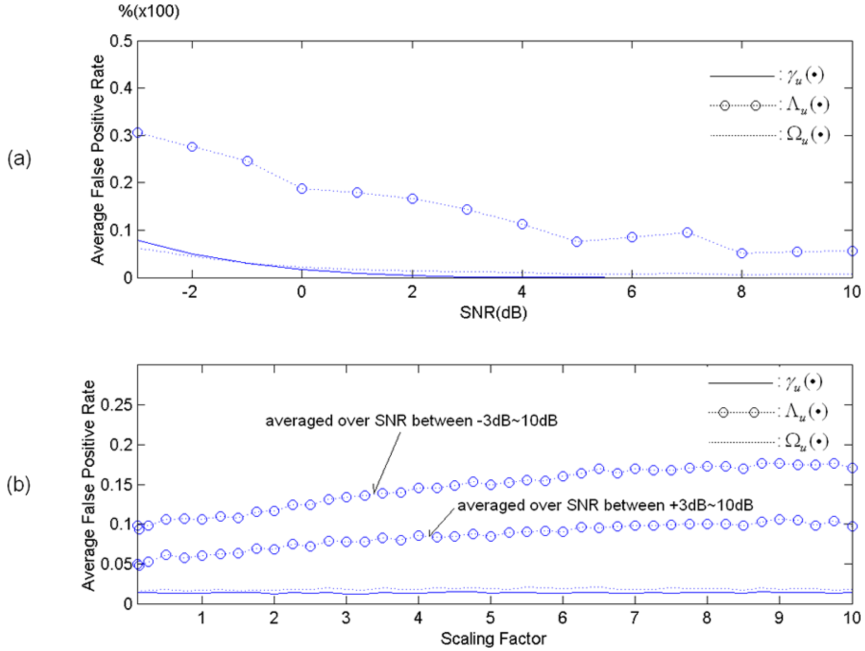


Fig. 11 Comparison of average FP rate between the three coefficients. (**a**) FP rate versus SNR, averaged over scaling factor between 0.1–10. (**b**) FP rate versus scaling factor, averaged over SNR

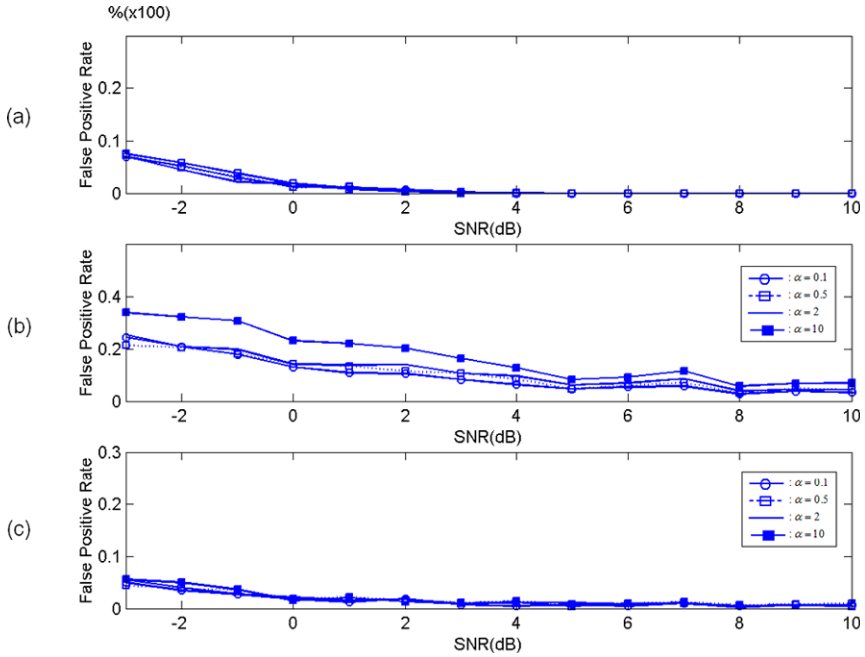


Fig. 12 Plots of FP rate versus SNR as a function of scaling factor. (**a**) $\gamma_u(\cdot)$, (**b**) $\Lambda_u(\cdot)$ (**c**) $\Omega_u(\cdot)$

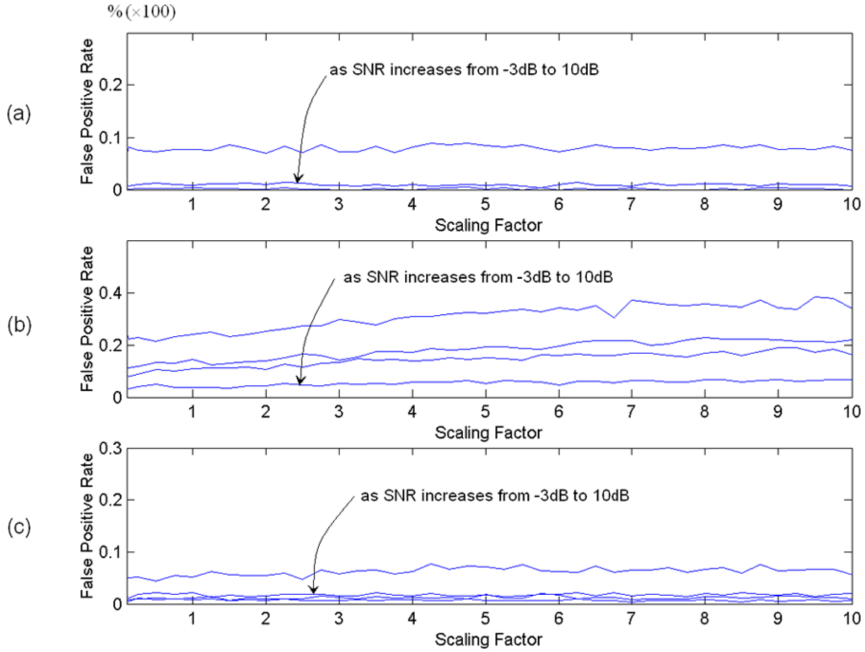


Fig. 13 Plots of FP rate versus scaling factor as a function of SNR. (**a**) $\gamma_u(\cdot)$, (**b**) $\Lambda_u(\cdot)$ (**c**) $\Omega_u(\cdot)$

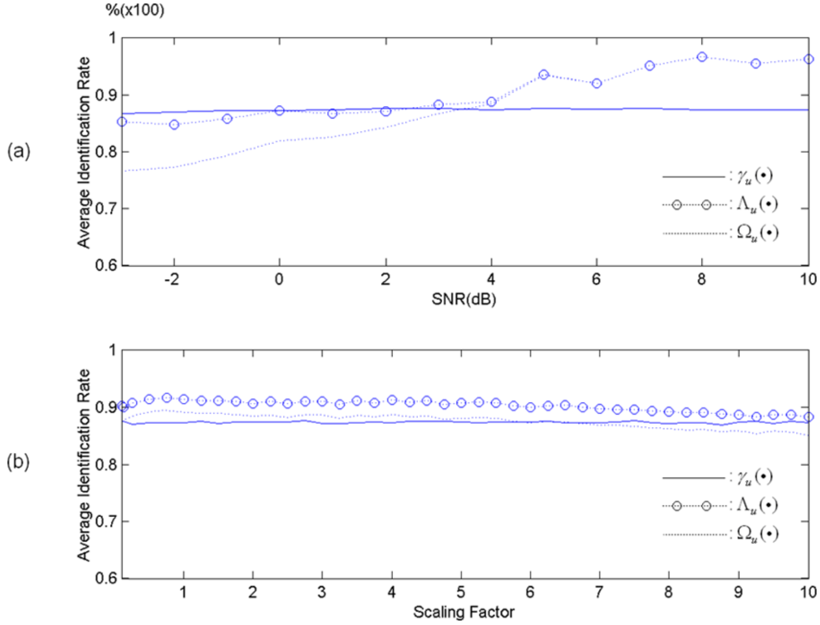


Fig. 14 Comparison of average identification rate between the three coefficients. (**a**) Identification rate versus SNR, averaged over scaling factor between 0.1–10. (**b**) Identification rate versus scaling factor, averaged over SNR between −3–10 dB

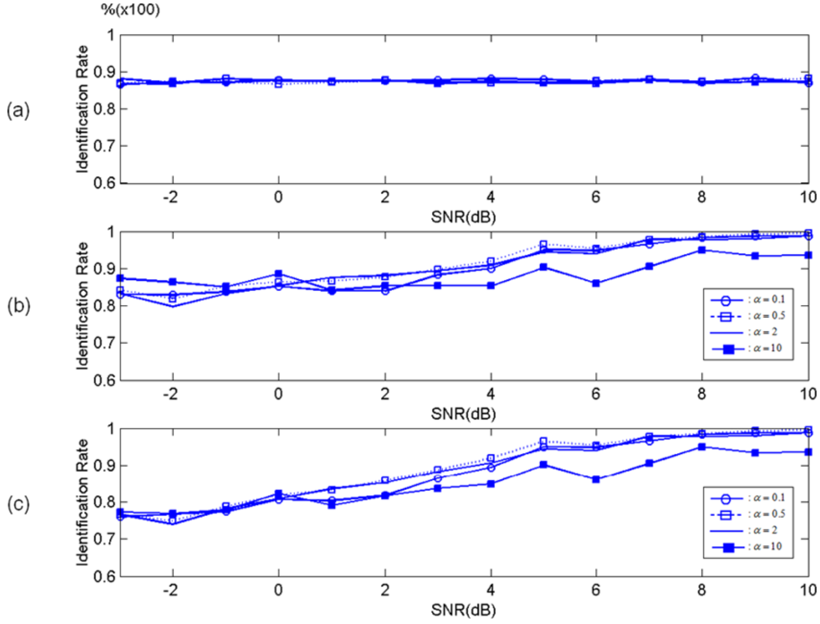


Fig. 15 Plots of identification rate versus SNR as a function of scaling factor. (**a**) $\gamma_u(\cdot)$, (**b**) $\Lambda_u(\cdot)$, (**c**) $\Omega_u(\cdot)$

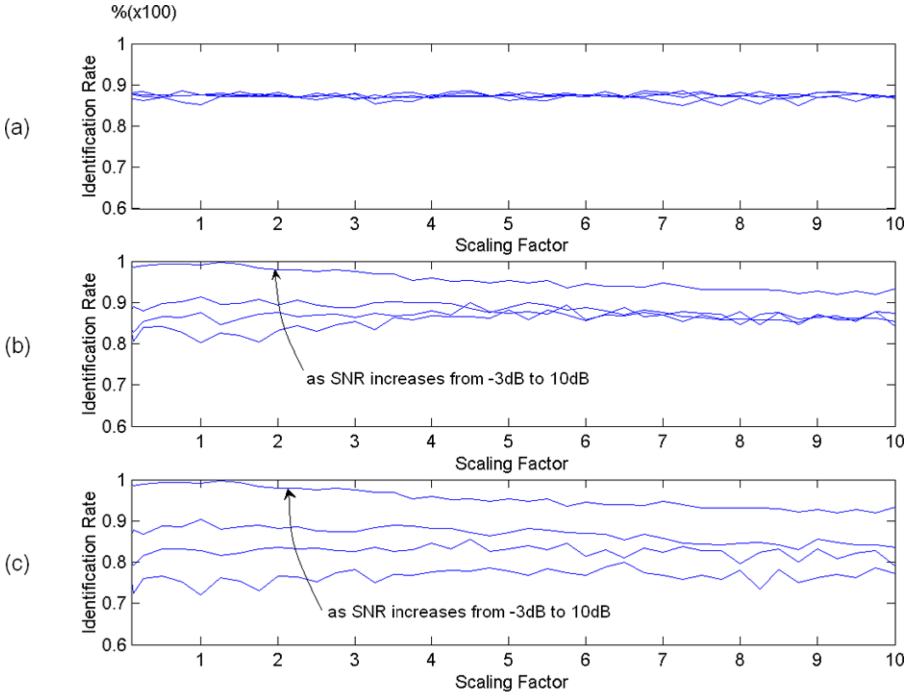


Fig. 16 Plots of identification rate versus scaling factor as a function of SNR. (**a**) $\gamma_u(\cdot)$, (**b**) $\Lambda_u(\cdot)$, (**c**) $\Omega_u(\cdot)$

significantly reduced in comparison with other traditional approaches and (iii) the FN rate is lower than that in the direct NCC.

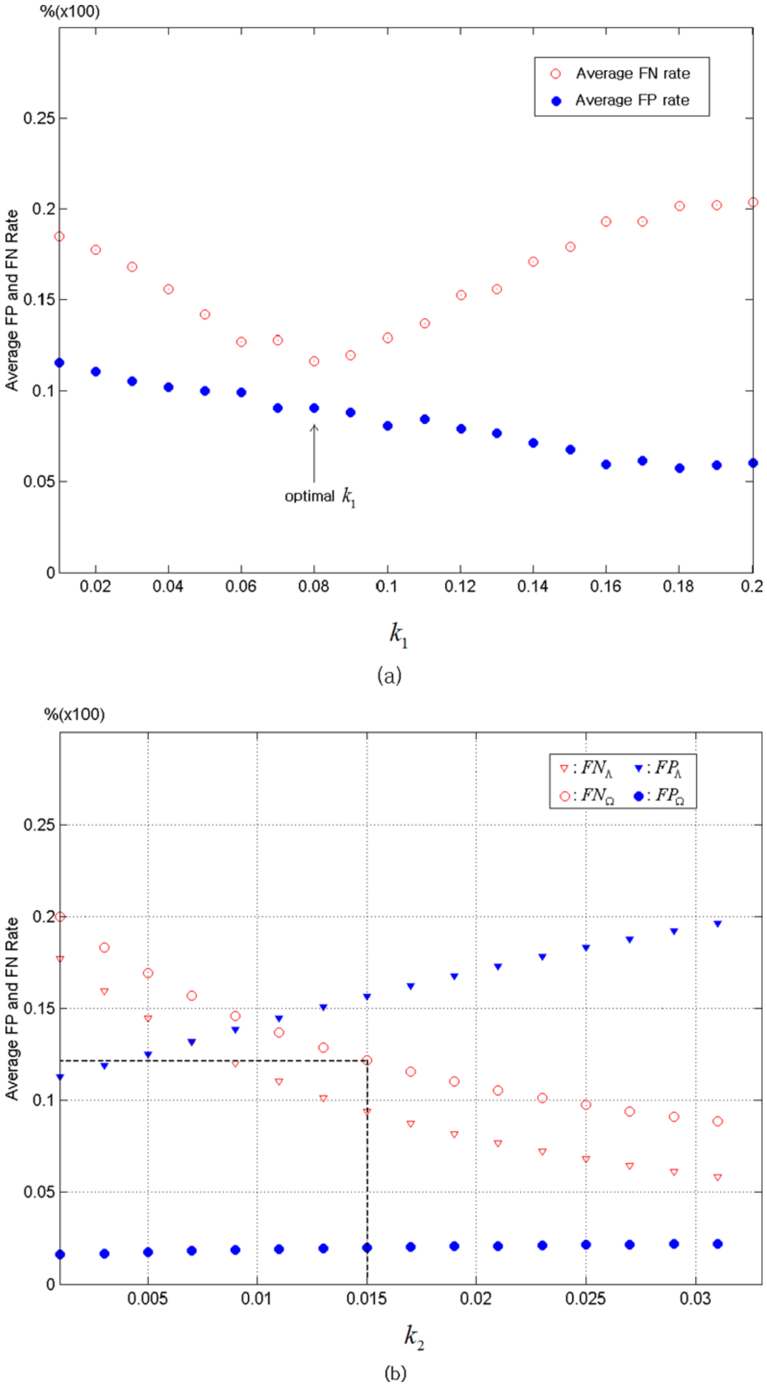
(2) The computations required for the filtering process given in (32) are trivial especially when the SNR is higher than 3 dB. This is because the size of the set $\mathbb{R}$ rapidly decreases as the SNR increases.

(3) The fast NCC based on $\Lambda_u(\cdot)$ can be used as a coarse identifier for roughly locating a template when the SNR is low and then the amount of computation required for the filtering process (see (32)) depends primarily on the *SNR*.

The fast NCC based on $\Lambda_u(\cdot)$ can be used as a fine identifier without using the filtering process when the *SNR* is high. These results clearly show the advantage of using the fast NCC algorithm.

5.4 The choice of parameters k_1 and k_2

The parameters, k_1 and k_2, in (19) were experimentally chosen to satisfy the rate of overall identification as high as possible with the minimum number of false alarms for the given SNR and the scaling factor α and finally set to 0.08 and 0.015, respectively. Figure 17(a) shows the FN and FP rate, averaged over SNR, as a function of the parameter k_1. This figure provides information on how to choose a suitable k_1. We can find the optimal point of the parameter k_1 at 0.08 from Fig. 17(a).



(a)

(b)

Fig. 17 The choice of parameters k_1 and k_2. (**a**) The parameter k_1 versus the average false alarm rate over SNR when $\alpha = 1$. (**b**) The parameter k_2 versus the average false alarm rate under the optimal value k_1

Figure 17(b) shows FN and FP rate, averaged over SNR and the scaling factor α, as a function of the parameter k_2. The symbols FN_Λ and FP_Λ denote, respectively, the average FN and FP rate by $\Lambda_u(\cdot)$ while FN_Ω and FP_Ω represent the average FN and FP rate by $\Omega_u(\cdot)$. It can be seen that the size of the set $\mathbb{R}$ increases as k_2 increases while the rate FN_Ω rate decreases. Therefore, there is a trade-off between the rate FN_Ω and the computational cost for the filtering process. In this paper, the parameter k_2 was set to 0.015 for achieving an FN rate of almost 13%.

6 Conclusions

The direct NCC is one of the most effective and commonly used similarity metrics in digital signal processing. However, it does not meet speed requirements for time-critical applications. In this paper, a fast NCC algorithm, which achieves a significant computation reduction with no loss in identification performance and without increasing FP and FN, was proposed. The fast NCC is very fast compared with the direct NCC and Lewis's algorithm. This is because most of its calculations consist of the logic operations between the values $+1$ and -1, which make them computationally efficient. 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 3 dB. In addition, it was seen that the fast NCC based on $\Lambda_u(\cdot)$ 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 [3, 6] in images and word spotting [7] in speech.

Appendix

Here, let us assume that $\Lambda_u(\cdot)$ follows $\gamma_u(\cdot)$ with fidelity.

(1) Fast NCC process using the threshold TH_Λ (see (29)).
Since $TH^{\gamma}_{\mathrm{roi}} > TH^{\Lambda}_{\mathrm{roi}}$ (see Fig. 6), $FP_\gamma < FP_\Lambda$ and $FN_\gamma > FN_\Lambda$.

(2) After (1), by filtering process (30) using the threshold $TH_{\Omega 1} = TH^{\gamma}_{\mathrm{roi}}$, the number of FP ($FP_{\Omega 1}$) decreases and thus:

$$FP_{\Omega 1} \le FP_\gamma < FP_\Lambda.$$

Also, the number of FN ($FN_{\Omega 1}$) increases and thus:

$$FN_{\Omega 1} > FN_\Lambda \quad \text{(and also } FN_\gamma > FN_\Lambda).$$

If $TH_{\Omega 2} = TH^{\gamma}_{\mathrm{roi}} - (s/200)$ (see (32)) instead of $TH_{\Omega 1} = TH^{\gamma}_{\mathrm{roi}}$ during the filtering process (30) is used then:

(i) since $TH_{\Omega 2} \approx TH_{\Omega 1}$ at Low SNR (i.e., small s), $FP_{\Omega 1} \approx FP_{\Omega 2}$ and $FN_{\Omega 1} \approx FN_{\Omega 2}$.

(ii) since $TH_{\Omega 2} < TH_{\Omega 1}$ at high SNR (i.e., large s),
   (a) $FN_{\Omega 1} > FN_{\Omega 2}$,
   (b) $FN_{\gamma} > FN_{\Omega 2} (\because TH_{\Omega 2} < TH^{\gamma}_{\text{roi}})$ and
   (c) $FP_{\Omega 1} \approx FP_{\Omega 2}$ ($\because$ the number of FP itself decreases at high SNR region).

From (a), (b) and(c), we obtain

$$FN_{\gamma} > FN_{\Omega 2} \quad \text{and} \quad FP_{\Omega 2} \leq FP_{\gamma} \quad \text{at high } SNR.$$

References

1. J. Ben-Aire, K.R. Rao, A lattice network for signal representation using Gaussian basis functions and max-energy paradigm, in *Proceedings of the 34th Midwest Symposium on Circuits and Systems*, vol. 1, May 1991, pp. 76–79
2. M. Betke, N.C. Makris, Fast object recognition in noisy images using simulated annealing, in *Proceedings of the Fifth International Conference on Computer Vision*, June 1995, pp. 523–530
3. J.S. Boland et al., Design of a correlator for real-time video comparisons. IEEE Trans. Aerosp. Electron. Syst. **1**, 11–20 (1979)
4. R.O. Duda, P.E. Hart, *Pattern Classification and Scene Analysis* (Wiley, New York, 1973)
5. R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, 2nd edn. (Prentice Hall, Upper Saddle River, 2002), pp. 701–704
6. O. Guillaume, R. Loic, M. Philippe, Correlation and similarity measures for SAR image matching. Proc. SPIE **5236**, 182–189 (2004)
7. E. Huang, H. Wang, F.K. Soong, A fast algorithm for large vocabulary keyword spotting application. IEEE Trans. Speech Audio Process. **2**(3), 449–452 (1994)
8. J.P. Lewis, Fast normalized cross-correlation, in *Proceedings of Vision Interface*, 1995, pp. 120–123
9. H. Li, Y. Wang, L. Wang, Matching score properties between range profiles of high resolution radar targets. IEEE Trans. Antennas Propag. **44**(4), 444–452 (1996)
10. H. Li, S. Yang, Using range profiles as feature vectors to identify aerospace objects. IEEE Trans. Antennas Propag. **41**(3), 261–268 (1993)
11. R. Tjahyadi, W. Liu et al., Face recognition based on ordinal correlation. Int. J. Intell. Syst. Technol. Appl. **3**(3/4), 226–240 (2007)
12. D. Tsai, C. Lin, Fast normalized cross correlation for defect detection. Pattern Recognit. Lett. **24**(15), 2625–2631 (2003)