Contributed article

Fractional Fourier transform pre-processing for neural networks and its application to object recognition

Billur Barshan*, Birsel Ayrulu

Department of Electrical Engineering, Bilkent University, Bilkent, TR-06533 Ankara, Turkey

Abstract

This study investigates fractional Fourier transform pre-processing of input signals to neural networks. The fractional Fourier transform is a generalization of the ordinary Fourier transform with an order parameter a. Judicious choice of this parameter can lead to overall improvement of the neural network performance. As an illustrative example, we consider recognition and position estimation of different types of objects based on their sonar returns. Raw amplitude and time-of-flight patterns acquired from a real sonar system are processed, demonstrating reduced error in both recognition and position estimation of objects. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Fractional Fourier transform; Neural networks; Input pre-processing; Object recognition; Position estimation; Sonar; Acoustic signal processing

1. Introduction

The fractional Fourier transform has received considerable interest in the past decade resulting in hundreds of papers dealing with its fundamental properties and its applications to optics and wave propagation, and signal analysis and processing. Its relationship to a wide range of concepts has been established and it has been employed in conjunction with a variety of techniques. Although two papers (Lee & Szu, 1994; Shin, Jin, Shin & Lee, 1998), one built upon the other, claim to discuss the relationship between the fractional Fourier transform and neural networks, it is debatable whether the networks in these papers can be properly called neural networks. In Lee and Szu (1994), an analogy is drawn between a new optical architecture and neural networks. The authors suggest that so-called fractional Fourier domain filtering configurations (Ozaktas, Barshan, Mendlovic & Onural, 1994b) can be interpreted as neural networks. In Shin et al. (1998), the authors implement a similar structure, and also consider reducing the mean-square error by log-likelihood and the use of parallel networks. It is shown that the 'neural network' using the fractional Fourier transform and the mean-square error classifies patterns much better than the one using the ordinary Fourier transform and the mean-square error. To speed up the learning convergence and, thus, improve the classification performance of the network, the mean-square error is replaced with the log-likelihood, and parallelism is introduced. It is shown that the combination of fractional Fourier transformation, log-likelihood and parallelism improves the learning convergence and recall rate of the network.

However, the network architectures employed in these works are actually nothing more than fractional Fourier domain filtering configurations. These static networks have linear input–output relations with the exception of point nonlinearities at the output; the weights are fixed and learning takes place only by adjustment of the filter coefficients, not the connection weights. In the present paper, we combine, to the best of our knowledge for the first time, fractional Fourier transforms and true neural networks with adjustment of the weights with a learning algorithm. We show how the use of fractional Fourier transformation as a pre-processing stage for a neural network classifier can result in increased performance and reduced error in classification. The improvement in the error is obtained by virtue of the order parameter a of this transform, which can be optimized to yield the best performance. Unlike neural networks, to the best of our knowledge, the fractional Fourier transform has not been applied to sonar signals and sensing before. While the illustrative application explored in our laboratory and reported in this paper is the differentiation and localization of targets using sonar, the use of fractional Fourier transform pre-processing should be of general applicability to a variety of problems where neural networks are employed.

* Corresponding author. Tel.: +90-312-290-2161; fax: +90-312-266-4192.

E-mail address: billur@ee.bilkent.edu.tr (B. Barshan).

Section 2.1 gives an overview of the fractional Fourier transform. In Section 2.2, background information related to object recognition with neural networks is presented. In Section 3, the use of the fractional Fourier transform for the pre-processing of inputs to neural networks is proposed and the system is described. Experimental results are presented in Section 5. In the final section, concluding remarks are made and directions for future work are discussed.

2. Background

2.1. The fractional Fourier transform

Fourier analysis is widely used in signal processing as well as many other branches of science and engineering (Bracewell, 1986). The a-th order fractional Fourier transform is a generalization of the ordinary Fourier transform such that the first order fractional Fourier transform is the ordinary Fourier transform and the zeroth order fractional Fourier transform corresponds to the function itself (Ozaktas, Kutay & Mendlovic, 1999; Ozaktas, Zalevsky & Kutay, 2001). Thus, the fractional Fourier transform includes the Fourier transform as a special case. Because of the additional parameter a, whose optimal value will in general be other than $a = 1$, the fractional transform is much more flexible and will in general offer better performance except in the special case when the optimal value of a coincidentally turns out to be precisely equal to 1. The transform has been studied extensively since the early 1990s with a view to applications in wave propagation, optics and optical signal processing (Mendlovic & Ozaktas, 1993; Ozaktas & Mendlovic, 1993a,b, 1995), time- and space-frequency analysis (Almeida, 1994; Kutay, Erden, Ozaktas, Arıkan, Güleryüz & Candan, 1998), pattern recognition (Mendlovic, Zalevsky & Ozaktas, 1998), digital signal processing (Kutay, Ozaktas, Arıkan & Onural, 1997; Kutay, Özaktaş, Ozaktas & Arıkan, 1999; Ozaktas et al, 1994a,b), and image processing (Barshan, Kutay & Ozaktas, 1997; Kutay & Ozaktas, 1998; Yetik, Ozaktas, Barshan & Onural, 2000) and other areas. Most applications are based on replacing the ordinary Fourier transform with the fractional Fourier transform. Since the latter has an additional degree of freedom (the order parameter a), it is often possible to generalize and improve upon previous results.

The a-th order fractional Fourier transform $f_a(u)$ of the function $f(u)$ is defined for $0 < |a| < 2$ as (McBride & Kerr, 1987; Ozaktas et al., 1994b)

$$f_a(u) \triangleq \int_{-\infty}^{\infty} K_a(u, u') f(u') \mathrm{d}u'$$

$$K_a(u, u') \triangleq A_\phi \exp\left[j\pi\left(u^2 \cot\phi - 2uu' \csc\phi + u'^2 \cot\phi \right) \right] \tag{1}$$

where

$$A_\phi = \frac{\exp[-j(\pi \mathrm{sgn}(\phi)/4 - \phi/2)]}{|\sin\phi|^{1/2}}$$

and $\phi = a\pi/2$.

The kernel $K_a(u, u')$ approaches $\delta(u - u')$ and $\delta(u + u')$ as a approaches 0 and ± 2, respectively, and are defined as such at these values. The fractional Fourier transform reduces to the ordinary Fourier transform when $a = 1$. The transform is linear and index additive; that is, the a_1-th order fractional Fourier transform of the a_2-th fractional Fourier transform of a function is equal to the $(a_1 + a_2)$-th order fractional Fourier transform. An important property of the fractional Fourier transform relating it to time-frequency (or space-frequency) concepts is its close relationship to the Wigner distribution (Cohen, 1995). The a-th order fractional Fourier transform of a function corresponds to a rotation of the Wigner distribution of the function by an angle $a\pi/2$ in the time-frequency plane. Moreover, digital implementation of the fractional Fourier transform is as efficient as that of the ordinary Fourier transform in the sense that it can also be computed in the order of $N \log N$ time with a fast algorithm, where N is the number of sample points or the signal length (Ozaktas, Arıkan, Kutay & Bozdaği, 1996). Therefore, the proposed technique does not introduce substantial overload.

The ordinary (unitary) discrete Fourier transform (DFT) of a signal $f(n)$ is defined as:

$$f_1(k) = \mathscr{F}\{f(n)\} \triangleq \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f(n) \mathrm{e}^{-(2\pi/N)jnk} \tag{2}$$

Here, the $1/N$ factor has been distributed between the forward and inverse transform expressions so as to result in a unitary definition of the DFT. The DFT can be represented in matrix notation as

$$\mathbf{f_1} = \mathbf{F}\mathbf{f} \tag{3}$$

where $\mathbf{f}$ is an $N \times 1$ column vector, $\mathbf{F}$ is the $N \times N$ DFT matrix, and $\mathbf{f_1}$ is the DFT of $\mathbf{f}$.

With a similar notation, the a-th order discrete fractional Fourier transform (DFRT) of $\mathbf{f}$, denoted $\mathbf{f_a}$, can be expressed as

$$\mathbf{f_a} = \mathbf{F^a}\mathbf{f} \tag{4}$$

where $\mathbf{F^a}$ is the $N \times N$ DFRT matrix which corresponds to the a-th power of the ordinary DFT matrix $\mathbf{F}$. However, it should be noted that there are certain subtleties and ambiguities in defining the power function, for which we refer the reader to Candan, Kutay and Ozaktas (2000).

The DFRT can be used to approximately compute the continuous fractional Fourier transform. That is, it can be used to approximately map the samples of the original function into the samples of its fractional Fourier transform. As with the ordinary DFT, the value of N should be chosen at
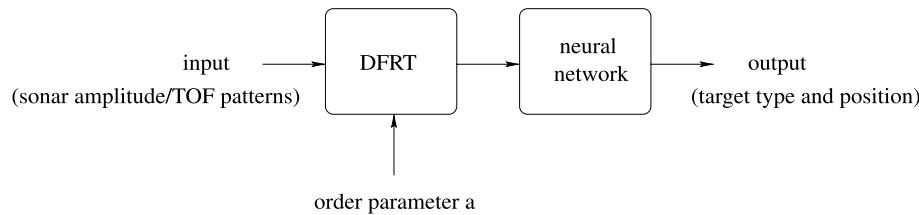
Fig. 1. The system block diagram.

least as large as the time- or space-bandwidth product of the signals in question.

2.2. Neural networks for object recognition

Neural networks are non-parametric and make weaker assumptions on the shape of the underlying distributions of input data than traditional statistical classifiers. Therefore, they can prove more robust when the underlying statistics are unknown or the data are generated by a nonlinear system. These networks are trained to compute the boundaries of decision regions in the form of connection weights and biases by using training algorithms. The most frequently used training algorithm for neural networks is the back-propagation algorithm (Werbos, 1990) which is also used to train the networks in this study. The stopping criterion used is as follows: the training is stopped either when the average error is reduced to 0.001 or if a maximum of 10,000 epochs is reached, whichever occurs earlier. The second case occurs very rarely.

The use of neural networks in sonar systems has been reviewed in Ayrulu and Barshan (2001). The successful use of neural networks for target differentiation and localization was first proposed in Barshan, Ayrulu and Utete (2000) and further developed in Ayrulu and Barshan (2001). There, it was shown that neural networks are superior to other pattern classification methods such as evidential reasoning, majority voting, and statistical differentiation algorithms.

An important issue in target differentiation with neural networks is the selection of input signals such that the network can differentiate all target types. Input signals resulting in a minimal network configuration (in terms of the number of layers and the number of neurons in these layers) with minimum classification error are preferable. In the next section, we investigate the effect of DFRT pre-processing of the input signals to the neural networks.

3. The proposed model

A simple block diagram of the system is given in Fig. 1. The input to the system is fractional Fourier transformed before being presented to the neural network for target type recognition and position estimation. The order parameter a of the DFRT represents a degree of freedom which we can optimize in order to transform the input

signals to a form in which the best classification performance is obtained. The spaces to which the signals are transformed have been referred to as 'fractional Fourier domains' (Ozaktas & Aytür, 1995).

To determine the optimal value of a, the training procedure is repeated for values of a varying from 0 to 1 with 0.05 increments. These networks are tested using three different test data sets which are: test set I, test data obtained at the training positions with the training targets; test set II, test data obtained at positions not in the set of training positions with the training targets; and test set III, test data obtained at the training positions with modified targets. Acquisition of these test data sets are described in more detail later in Section 5. For each value of a, and for each test set, error of classification, error of range and azimuth estimation, averaged over all target types, are calculated. The value of a to be employed is chosen as that which results in the smallest error.

In the next section, we will demonstrate through a concrete application example that by choosing a in this manner, it is possible to obtain improved results.

4. Application

We will illustrate the proposed method with the problem of differentiation and localization of targets using sonar signals. More concretely, an ultrasonic sensor pair transmitting and receiving ultrasonic pulses will be used to collect data from an unknown target, to be processed to reveal the type of target and its position.

The basic target types to be differentiated in this study are *plane*, *corner*, *acute corner*, *edge* and *cylinder* (Fig. 2). In particular, we have employed a planar target, a corner of $\theta_c = 90°$, an acute corner of $\theta_c = 60°$, an edge of $\theta_e = 90°$, and cylinders with radii $r_c = 2.5$, 5.0 and 7.5 cm, all made of wood. Detailed reflection models of these target primitives are provided in Ayrulu and Barshan (1998).

The most common sonar ranging system is based on *time-of-flight* (TOF) which is the time elapsed between the transmission and the reception of a pulse. In the commonly used TOF systems, an echo is produced when the transmitted pulse encounters an object and a range measurement $r = ct_o/2$ is obtained (Fig. 3) by
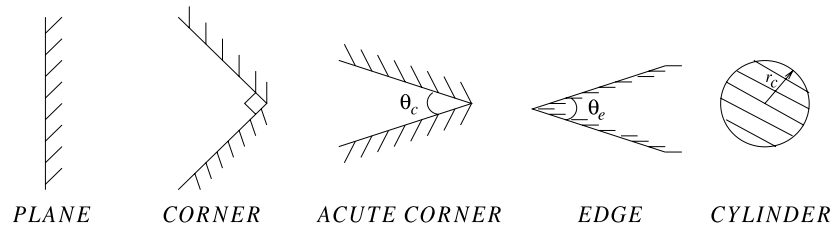
Fig. 2. Horizontal cross sections of the target primitives differentiated in this study. Reprinted from Barshan et al. (2000) with permission. © 2000 IEEE.

simple thresholding. Here, t_o is the TOF and c is the speed of sound in air (at room temperature, $c = 343.3$ m/s). The transducers can function both as transmitter and receiver. In simple thresholding systems, a constant threshold value τ is set according to the noise level of the sonar signals and the instance at which the signal amplitude exceeds this threshold value for the first time is marked as the TOF.

The major limitation of ultrasonic transducers comes from their large beamwidth. Although these devices return accurate range data, they cannot provide direct information on the angular position of the object from which the reflection was obtained. Sensory information from a *single* sonar sensor has poor angular resolution and is usually not sufficient to differentiate more than a small number of target primitives (Barshan & Kuc, 1990). Improved target classification can be achieved by using multi-transducer pulse/echo systems and by employing both amplitude and TOF information. In the present paper, amplitude and TOF information from a pair of identical ultrasonic transducers a and b with center-to-center separation $d = 25$ cm is employed to improve the angular resolution of sonar sensors (Barshan et al., 2000).

The transducers used in our experimental setup are Panasonic transducers (Panasonic Corporation, 1989). The aperture radius of the transducers is $a = 0.65$ cm, their resonance frequency is $f_o = 40$ kHz, and their
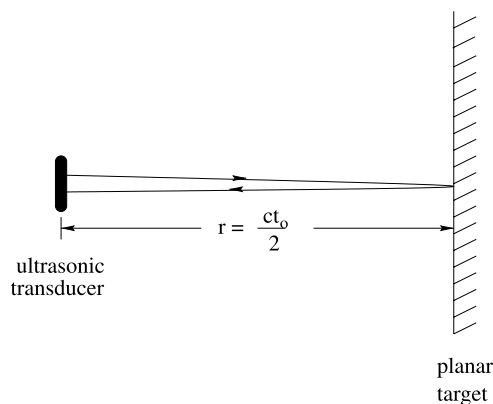
beamwidth angle is 54°. The entire sensing unit is mounted on a small 6 V computer-controlled stepper motor with step size 1.8°. Data acquisition from the sonars is through a PC A/D card with 12-bit resolution and 1 MHz sampling frequency. Starting at the transmit time, 10,000 samples of each echo signal are collected and thresholded to extract the TOF information. The amplitude information is obtained by finding the maximum value of the signal after the threshold value is exceeded.

Amplitude and TOF patterns of the targets are collected in this manner at 25 different locations (r, θ) for each target from $\theta = -20$ to $20°$ in 10° increments, and from $r = 35$ to 55 cm in 5 cm increments (Fig. 4). The target primitive located at range r and azimuth θ is scanned by the rotating sensing unit for scan angles $-52° \leq \alpha \leq 52°$ with 1.8° increments (determined by the step size of the motor). The reason for using a wider range for the scan angle is the possibility that a target may still generate returns outside of the range of θ. The angle α is always measured with respect to $\theta = 0°$ regardless of target location (r, θ), as

Fig. 3. Reflection of ultrasonic echoes from a planar target.
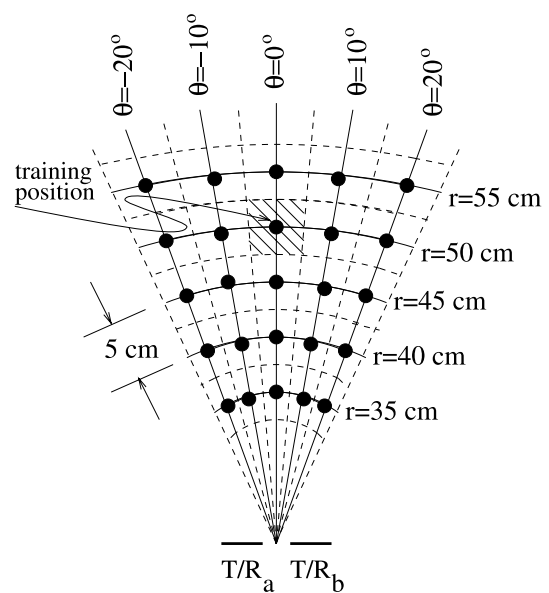

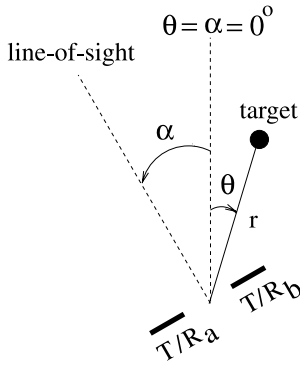Fig. 4. Network training positions. Reprinted from Barshan et al. (2000) with permission. © 2000 IEEE.

Fig. 5. The scan angle α and the target azimuth θ.

shown in Fig. 5. In other words, $\theta = 0°$ and $\alpha = 0°$ always coincide.

At each step of the scan, four sonar echo signals are acquired as a function of time (Fig. 6). In the figure, A_{aa}, A_{bb}, A_{ab}, and A_{ba} denote the maximum values of the echo signals, and t_{aa}, t_{bb}, t_{ab}, and t_{ba} denote the TOF readings extracted from the same signals by simple thresholding. The first index in the subscript indicates the transmitting transducer, the second index denotes the receiver. At each step of the scan, only these eight amplitude and TOF values extracted from the

four echo signals are recorded. For the given scan range and motor step size, 58 ($= 2 \times 52°/1.8°$) angular samples of each of the amplitude and TOF patterns $A_{aa}(\alpha)$, $A_{bb}(\alpha)$, $A_{ab}(\alpha)$, $A_{ba}(\alpha)$, $t_{aa}(\alpha)$, $t_{bb}(\alpha)$, $t_{ab}(\alpha)$ and $t_{ba}(\alpha)$ are acquired at each target location. These scans can be considered as acoustic signatures embodying shape and position information of the objects.

Since the cross terms $A_{ab}(\alpha)$ and $A_{ba}(\alpha)$ (or $t_{ab}(\alpha)$ and $t_{ba}(\alpha)$) should be equal under ideal conditions due to reciprocity, it is more representative to employ their average. Thus, 58 samples each of the following six functions are taken collectively as the input to the overall system:

$$A_{aa}(\alpha), A_{bb}(\alpha), \frac{A_{ab}(\alpha) + A_{ba}(\alpha)}{2}, t_{aa}(\alpha), t_{bb}(\alpha),$$

and

$$\frac{t_{ab}(\alpha) + t_{ba}(\alpha)}{2}$$

Scans are collected with 4-fold redundancy for each target primitive at each location, resulting in 700 ($= 4$-fold redundancy $\times 25$ locations $\times 7$ target types) sets of scans to be used for training.
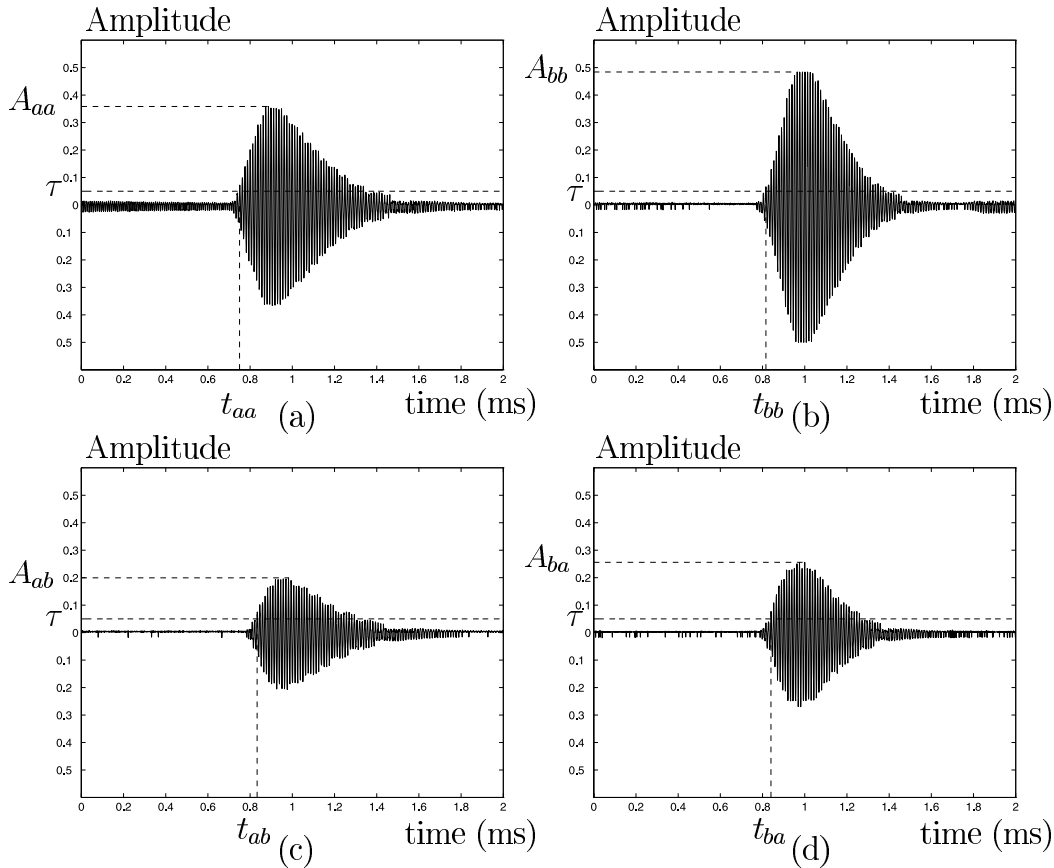

Fig. 6. Real sonar echoes from a planar target located at $r = 60$ cm and $\theta = 0°$ when: (a) transducer a transmits and transducer a receives; (b) transducer b transmits and b receives; (c) transducer a transmits and b receives; and (d) transducer b transmits and a receives. Reprinted from Ayrulu and Barshan (2001).
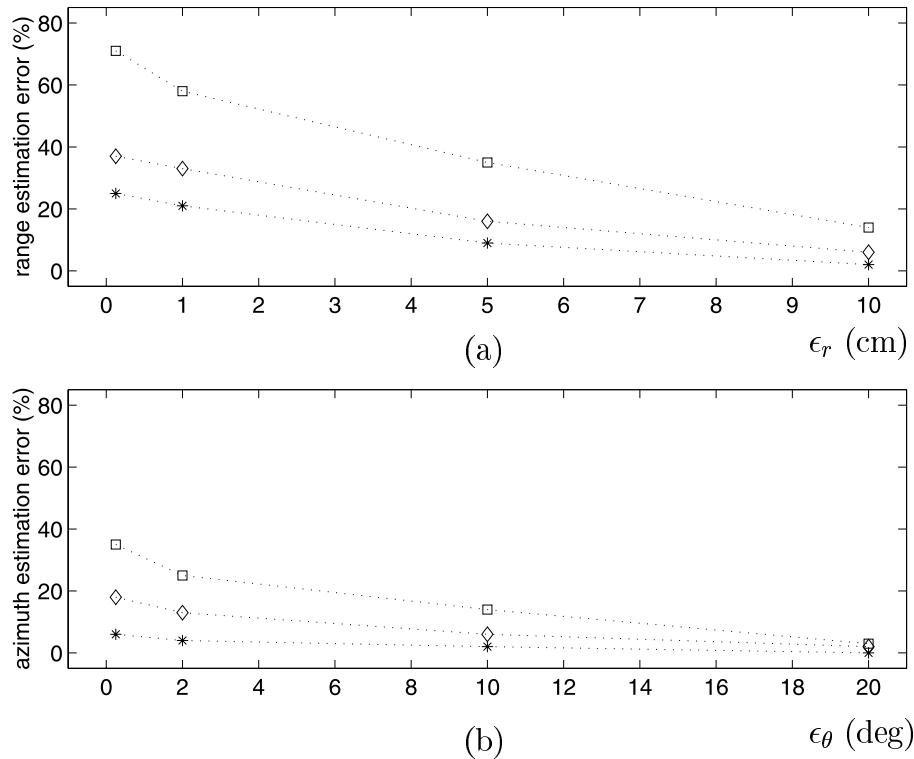
Fig. 7. The range and azimuth estimation error versus the tolerance levels ϵ_r and ϵ_θ when the grid locations are used as test positions (test set I). The data points correspond to $\epsilon_r = 0.125, 1, 5, 10$ cm and $\epsilon_\theta = 0.25, 2, 10, 20°$. The squares, diamonds and stars correspond to $a = 0$ (no pre-processing), $a = 1$ (DFT pre-processing), $a = a_{opt}$ (DFRT pre-processing), respectively.

5. Experiments and results

The neural networks constructed consist of one input, one hidden, and one output layer. The number of input-layer neurons is determined by the total number of samples of the amplitude and TOF patterns of the input signal, described above. After averaging the cross terms of the raw amplitude and TOF patterns, there are six patterns each with 58 samples; therefore, $348 \ (= 6 \times 58)$ input units are used.

Two well-known methods for determining the number of hidden-layer neurons in feed-forward neural networks are *pruning* and *enlarging* (Haykin, 1994). Pruning begins with a relatively large number of hidden-layer neurons and eliminates unused neurons according to some criterion. Enlarging begins with a relatively small number of hidden-layer neurons and gradually increases their number until learning occurs. In this study, the number of hidden-layer neurons is determined by enlarging. On average, 79 units are used at the hidden layer for the different networks constructed. The number of output-layer neurons is 21. The first seven neurons encode the target type. The next seven represent the target range r which is binary coded with a resolution of 0.25 cm. The last seven neurons represent the azimuth θ of the target, which is also binary coded with resolution 0.5°. To ensure that the initial weight values do not affect the results, we averaged the results

of 10 networks with different randomly chosen initial weights.

The networks have been first tested with test set I. Each target primitive is placed in turn in each of the 25 training positions shown in Fig. 4. Four sets of scans are collected for each combination of target type and location, again resulting in 700 sets of experimentally acquired scans. Based on these data, the trained neural networks estimate the target type, range, and azimuth. The test data are collected independently from the training data and the targets are presented to the neural network in a randomized order. These ensure that systematic biases do not result in overstatement of the performance of the method.

The average classification error obtained without pre-processing of the amplitude and TOF patterns is 14%, whereas it is 5% with the ordinary discrete Fourier transform (DFT) pre-processing and 0% with discrete fractional Fourier transform (DFRT) pre-processing. A range or azimuth estimate is considered correct if it is within an error tolerance of ϵ_r or ϵ_θ of the actual range or azimuth, respectively. Average range and azimuth estimation errors for different values of ϵ_r and ϵ_θ are given in Fig. 7. As expected, the average errors decrease with increasing tolerance. We observe that DFRT pre-processing offers significant reduction in localization error compared to no pre-processing (on average, 72% or 3.6-fold for range
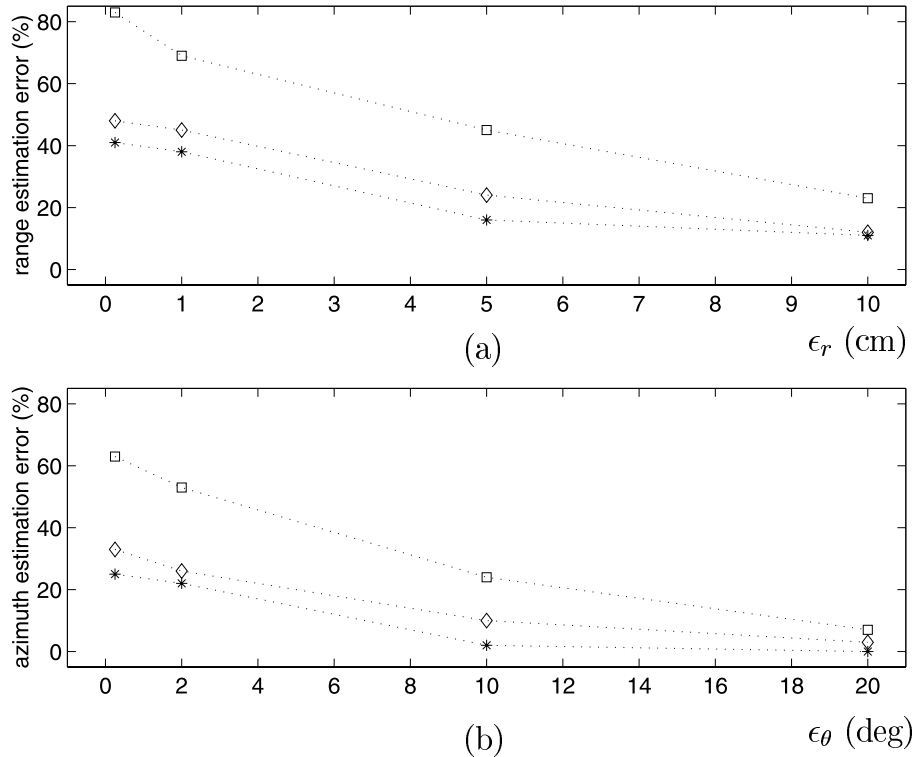
Fig. 8. The range and azimuth estimation error versus the tolerance levels ϵ_r and ϵ_θ when the test points are randomly chosen from a continuum (test set II). The data points correspond to $\epsilon_r = 0.125, 1, 5, 10$ cm and $\epsilon_\theta = 0.25, 2, 10, 20°$. The squares, diamonds and stars correspond to $a = 0$ (no pre-processing), $a = 1$ (DFT pre-processing), $a = a_{opt}$ (DFRT pre-processing), respectively.

estimation and 88% or 8.3-fold for azimuth estimation). Since the DFRT has a fast algorithm, these improvements do not introduce substantial overhead once the system is trained. The DFT also results in considerable reduction of the error (on average, 50% or 2-fold for range estimation and 47% or 1.9-fold for azimuth estimation); however, since the cost of computing the DFRT and DFT are the same, the additional improvement coming from the use of the DFRT brings no additional cost after training.

The networks have been also tested for targets situated arbitrarily in the continuous estimation space (test set II) and not necessarily confined to the 25 training locations of Fig. 4. This second set of test data was also acquired independently after collecting the training data. Randomly generated locations within the area shown in Fig. 4, not necessarily corresponding to one of the 25 grid locations, are used as target positions. The r, θ values corresponding to these locations are generated by using the uniform random number generator in MATLAB. The range for r is [32.5 cm, 57.5 cm] and that for θ is [−25°, 25°]. In this case, the average classification error obtained without pre-processing is 15%, whereas it is 5% with the DFT and 0% with DFRT pre-processing. The localization results are given in Fig. 8. We see that the DFRT results in comparable improvements as in the case where the test targets were located at the grid positions. The reduction in error is on

average 57% for range estimation and 77% for azimuth estimation. Similar comments as we have made for Fig. 7 apply. As expected, the errors for the non-grid test positions can be higher than those for the grid test positions (compare the corresponding data points in Figs. 7 and 8).

Noting that the networks were trained only at 25 locations and at grid spacings of 5 cm and 10°, it can be concluded from the localization errors at tolerances of $\epsilon_r = 0.125$ and 1 cm and $\epsilon_\theta = 0.25$ and 2°, that the networks demonstrate the ability to interpolate between the training grid locations. Thus, the neural network maintains a certain spatial continuity between its input and output and does not haphazardly map positions which are not drawn from the 25 locations of Fig. 4. Correct target recognition percentages are quite high and the accuracy of the range/azimuth estimates would be acceptable in many applications. If better estimates are required, they can be achieved by reducing the training grid spacing in Fig. 4.

It is instructive to examine the confusion matrices associated with the classification process. We consider test sets I and II. In the following matrices, the (i, j)-th element denotes the percentage of classification of actual target i as target j, where $i, j = 1, \dots, 7$ corresponds, respectively, to a planar target, a corner of $\theta_c = 90°$, an acute corner of $\theta_c = 60°$, an edge of $\theta_e = 90°$, and cylinders with radii $r_c = 2.5, 5.0$ and

7.5 cm. By definition, the row sums are equal to 100. For no pre-processing:

$$C_{none}^{I} = \begin{bmatrix} 90 & 0 & 5 & 0 & 5 & 0 & 0 \\ 0 & 91 & 0 & 9 & 0 & 0 & 0 \\ 3 & 2 & 91 & 0 & 4 & 0 & 0 \\ 1 & 12 & 0 & 87 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 84 & 12 & 2 \\ 1 & 0 & 1 & 0 & 9 & 82 & 7 \\ 4 & 0 & 1 & 0 & 9 & 11 & 75 \end{bmatrix} \qquad C_{none}^{II} = \begin{bmatrix} 87 & 0 & 5 & 0 & 5 & 0 & 3 \\ 0 & 90 & 2 & 8 & 0 & 0 & 0 \\ 0 & 0 & 96 & 0 & 4 & 0 & 0 \\ 4 & 11 & 0 & 85 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 81 & 10 & 6 \\ 1 & 0 & 1 & 0 & 12 & 80 & 6 \\ 4 & 0 & 1 & 0 & 10 & 12 & 73 \end{bmatrix}$$

for DFT pre-processing:

$$C_{DFT}^{I} = \begin{bmatrix} 97 & 0 & 1 & 0 & 2 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 98 & 0 & 2 & 0 & 0 \\ 0 & 3 & 0 & 96 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 95 & 2 & 1 \\ 1 & 0 & 1 & 0 & 3 & 93 & 2 \\ 3 & 0 & 0 & 0 & 4 & 4 & 89 \end{bmatrix} \qquad C_{DFT}^{II} = \begin{bmatrix} 97 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 98 & 0 & 2 & 0 & 0 \\ 1 & 3 & 0 & 96 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 94 & 3 & 2 \\ 1 & 0 & 1 & 0 & 4 & 92 & 2 \\ 3 & 0 & 2 & 0 & 4 & 6 & 85 \end{bmatrix}$$

and for DFRT pre-processing:

$$C_{DFRT}^{I} = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 99 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 99 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 99 \end{bmatrix} \qquad C_{DFRT}^{II} = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 99 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 99 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 98 & 1 \\ 0 & 0 & 0 & 0 & 0 & 4 & 96 \end{bmatrix}$$

By studying these matrices, it is possible to deduce which targets are most similar under the different pre-processing feature spaces and how these similarities are resolved with DFRT pre-processing. For instance, for test set I with no pre-processing, a plane is confused 5% of the time as an edge and 5% of the time as a cylinder. Cylinders with different radii are often confused with each other. We observe that DFRT pre-processing does a much better job of discriminating these targets.

We have carried out further tests with the same system using targets not scanned during training, which are slightly different in size, shape, or roughness than the targets used for training (test set III). These are two smooth cylinders of radii 4 and 10 cm, a cylinder of radius 7.5 cm covered with bubbled packing material, a 60° smooth edge, and a plane covered with bubbled packing material. The packing material with bubbles has a honeycomb pattern of uniformly distributed circular bubbles of diameter 1.0 cm and height 0.3 cm, with a center-to-center separation of 1.2 cm. The test data are collected at the 25 grid locations used for training. In this case, the average classification error obtained with unpre-processed patterns is 19%, whereas it is 18% with DFT pre-processing and 9% with DFRT pre-processing. The corresponding range and
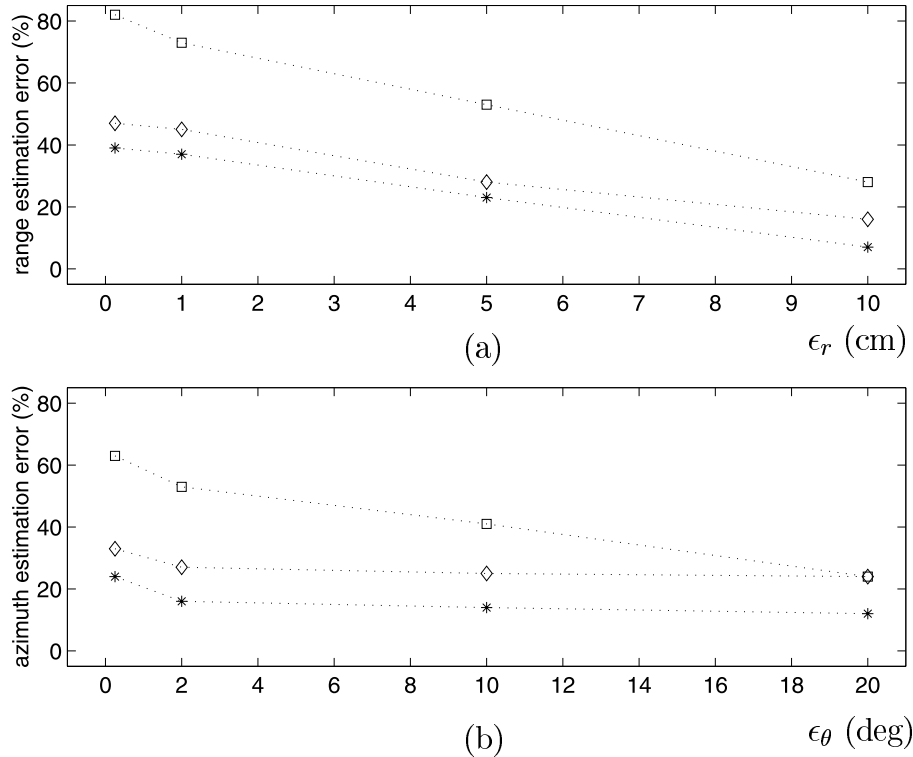
Fig. 9. The range and azimuth estimation error versus the tolerance levels ϵ_r and ϵ_θ when targets which are not scanned during training are used for testing at the discrete positions (test set III). The data points correspond to $\epsilon_r = 0.125, 1, 5, 10$ cm and $\epsilon_\theta = 0.25, 2, 10, 20°$. The squares, diamonds and stars correspond to $a = 0$ (no pre-processing), $a = 1$ (DFT pre-processing), $a = a_{opt}$ (DFRT pre-processing), respectively.

azimuth estimation errors are presented in Fig. 9. Once again, we observe substantial improvements with the use of DFRT pre-processing. On average, there is 58% reduction in range estimation error and 62% reduction in azimuth estimation error compared to no pre-processing. When the network is tested with these modified targets, there is some increase in localization errors compared to the results obtained with targets identical to those used for training (compare the corresponding data points in Figs. 7 and 9). Overall, we can conclude that the network exhibits some degree of robustness to variations in target shape, size, and roughness.

The percentage errors of correct classification, correct range (r) and azimuth (θ) estimation for no pre-processing, DFT, and DFRT pre-processing are summarized in Table 1 for the three test sets. The error tolerances corresponding to r and θ estimation are selected equal to the grid intervals which are 5 cm and 10°, respectively. It can be clearly seen

that only the optimal DFRT achieves 100% correct classification for test set I, and it still achieves the best performance for test sets II and III.

No pre-processing corresponds to the space domain and DFT pre-processing corresponds to the Fourier domain. Both no pre-processing and DFT pre-processing have the same status as being limiting special cases of DFRT pre-processing, with $a = 0$ and 1, respectively. Therefore, while DFRT pre-processing will by definition always be at least equal to or better than both, there was no a priori reason to expect DFT pre-processing to give better results than no pre-processing. Nevertheless, in our results, we observe that DFT pre-processing consistently gives better results.

6. Conclusions

In this paper, we considered the use of the discrete

Table 1
The percentage errors of correct classification, correct range (r) and azimuth (θ) estimation for no pre-processing, DFT, and DFRT pre-processing for test sets I, II and III. The error tolerances corresponding to r and θ estimation were selected equal to the grid intervals which are 5 cm and 10°, respectively

	Classification			r estimation			θ estimation		
	I	II	III	I	II	III	I	II	III
No pre-processing	14	15	19	35	45	53	14	24	41
DFT	5	5	18	16	24	28	6	10	25
DFRT	0	1	9	9	16	23	2	2	14

fractional Fourier transform (DFRT) pre-processing for neural network input signals with the purpose of increasing the overall performance. We have illustrated the method with a specific application example using experimental data: target classification and localization with sonar. The order parameter of the DFRT provides an additional degree of flexibility for optimization to reduce errors. The DFT also results in considerable reduction of the error; however, since the cost of computing the DFRT and DFT are the same, the additional improvement coming from the use of the DFRT brings no additional cost once the system is trained.

The method presented can be further generalized by employing the three-parameter family of linear canonical transforms instead of the one-parameter family of fractional Fourier transforms (Barshan et al., 1997; Ozaktas et al., 2001).

Although trained on a discrete and relatively coarse grid, the networks are able to interpolate between the grid locations and offer higher resolution than that implied by the grid size. The correct estimation rates for target type, range and azimuth can be further increased by employing a finer grid for training.

In conclusion, the use of fractional Fourier transform pre-processing results in increased performance compared to both no pre-processing and ordinary Fourier transform pre-processing with substantial reduction in classification and localization error. While use of the fractional Fourier transform increases the cost of the training procedure, the improvements achieved with its use come at no additional routine operating cost.

References

Almeida, L. B. (1994). The fractional Fourier transform and time-frequency representations. *IEEE Transactions on Signal Processing*, *42* (11), 3084–3091.

Ayrulu, B., & Barshan, B. (1998). Identification of target primitives with multiple decision-making sonars using evidential reasoning. *International Journal of Robotics Research*, *17* (6), 598–623.

Ayrulu, B., & Barshan, B. (2001). Neural networks for improved target differentiation and localization with sonar. *Neural Networks*, *14* (3), 355–373.

Barshan, B., & Kuc, R. (1990). Differentiating sonar reflections from corners and planes by employing an intelligent sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12* (6), 560–569.

Barshan, B., Ayrulu, B., & Utete, S. W. (2000). Neural network based target differentiation using sonar for robotics applications. *IEEE Transactions on Robotics and Automation*, *16* (4), 435–442.

Barshan, B., Kutay, M. A., & Ozaktas, H. M. (1997). Optimal filtering with linear canonical transformations. *Optics Communications*, *135* (1–3), 32–36.

Bracewell, R. N. (1986). *The Fourier transform and its applications*, New York: McGraw-Hill.

Candan, Ç., Kutay, M. A., & Ozaktas, H. M. (2000). The discrete fractional Fourier transform. *IEEE Transactions on Signal Processing*, *48* (5), 1329–1337.

Cohen, S. (1995). *Time-frequency analysis*, New Jersey: Prentice-Hall.

Haykin, S. (1994). *Neural networks: a comprehensive foundation*, New Jersey: Prentice-Hall.

Kutay, M. A., & Ozaktas, H. M. (1998). Optimal image restoration with the fractional Fourier transform. *The Journal of the Optical Society of America A*, *15* (4), 825–833.

Kutay, M. A., Erden, M. F., Ozaktas, H. M., Arıkan, O., Güleryüz, Ö., & Candan, Ç. (1998). Space-bandwidth-efficient realizations of linear systems. *Optics Letters*, *23* (14), 1069–1071.

Kutay, M. A., Ozaktas, H. M., Arıkan, O., & Onural, L. (1997). Optimal filtering in fractional Fourier domains. *IEEE Transactions on Signal Processing*, *45* (5), 1129–1143.

Kutay, M. A., Özaktaş, H., Ozaktas, H. M., & Arıkan, O. (1999). The fractional Fourier domain decomposition. *Signal Processing*, *77* (1), 105–109.

Lee, S. Y., & Szu, H. H. (1994). Fractional Fourier transform, wavelet transforms and adaptive neural networks. *Optical Engineering*, *33* (7), 2326–2330.

McBride, A. C., & Kerr, F. H. (1987). On Namias's fractional Fourier transform. *IMA Journal of Applied Mathematics*, *39* (2), 159–175.

Mendlovic, D., & Ozaktas, H. M. (1993). Fourier transforms and their optical implementation: I. *The Journal of the Optical Society of America A*, *10* (9), 1875–1881.

Mendlovic, D., Zalevsky, Z., & Ozaktas, H. M. (1998). Applications of the fractional Fourier transform to optical pattern recognition. In F. T. S. Yu & S. Jutamulia, *Optical pattern recognition* (pp. 89–125). Cambridge: Cambridge University Press.

Ozaktas, H. M., & Aytür, O. (1995). Fractional Fourier domains. *Signal Processing*, *46*, 119–124.

Ozaktas, H. M., & Mendlovic, D. (1993a). Fourier transforms of fractional order and their optical interpretation. *Optics Communications*, *101* (3,4), 163–169.

Ozaktas, H. M., & Mendlovic, D. (1993b). Fractional Fourier transforms and their optical implementation: II. *The Journal of the Optical Society of America A*, *10* (12), 2522–2531.

Ozaktas, H. M., & Mendlovic, D. (1995). Fractional Fourier optics. *The Journal of the Optical Society of America A*, *12* (4), 743–751.

Ozaktas, H. M., Arıkan, O., Kutay, M. A., & Bozdaği, G. (1996). Digital computation of the fractional Fourier transform. *IEEE Transactions on Signal Processing*, *44* (9), 2141–2150.

Ozaktas, H. M., Barshan, B., & Mendlovic, D. (1994a). Convolution and filtering in fractional Fourier domains. *Optical Review*, *1* (1), 15–16.

Ozaktas, H. M., Barshan, B., Mendlovic, D., & Onural, L. (1994b). Convolution, filtering, and multiplexing in fractional Fourier domains and their relation to chirp and wavelet transforms. *The Journal of the Optical Society of America A*, *11* (2), 547–559.

Ozaktas, H. M., Kutay, M. A., & Mendlovic, D. (1999). Introduction to the fractional Fourier transform and its applications. In P. W. Hawkes, *Advances in imaging and electron physics* (pp. 239–291). San Diego, CA: Academic Press.

Ozaktas, H. M., Zalevsky, Z., & Kutay, M. A. (2001). *The fractional Fourier transform with applications in optics and signal processing*, New York: John Wiley.

Panasonic Corporation (1989). *Ultrasonic ceramic microphones*, 12 Blanchard Road, Burlington, MA, USA: Panasonic Corporation.

Shin, S. G., Jin, S. I., Shin, S. Y., & Lee, S. Y. (1998). Optical neural networks using fractional Fourier transform: log likelihood and parallelism. *Optics Communications*, *153* (4–6), 218–222.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, *78*, 1550–1560.

Yetik, I.Ş., Ozaktas, H. M., Barshan, B., & Onural, L. (2000). Perspective projections in the space-frequency plane and fractional Fourier transforms. *The Journal of the Optical Society of America A*, *17* (12), 2382–2390.