Information Retrieval 2, 207–226 (2000)

Word Spotting in Bitmapped Fax Documents

WILLIAM J. WILLIAMS
EUGENE J. ZALUBAS
ALFRED O. HERO, III
Electrical Engineering and Computer Science Dept., University of Michigan, Ann Arbor MI 48109, USA

Received April 1, 1999; Revised November 1, 1999; Accepted December 17, 1999

Abstract. Images and signals may be represented by forms invariant to time shifts, spatial shifts, frequency shifts, and scale changes. Advances in time-frequency analysis and scale transform techniques have made this possible. However, factors such as noise contamination and "style" differences complicate this. An example is found in text, where letters and words may vary in size and position. Examples of complicating variations include the font used, corruption during facsimile (fax) transmission, and printer characteristics. The solution advanced in this paper is to cast the desired invariants into separate subspaces for each extraneous factor or group of factors. The first goal is to have minimal overlap between these subspaces and the second goal is to be able to identify each subspace accurately. Concepts borrowed from high-resolution spectral analysis, but adapted uniquely to this problem have been found to be useful in this context. Once the pertinent subspace is identified, the recognition of a particular invariant form within this subspace is relatively simple using well-known singular value decomposition (SVD) techniques. The basic elements of the approach can be applied to a variety of pattern recognition problems. The specific application covered in this paper is word spotting in bitmapped fax documents.

Keywords: word spotting, facsimile, scale, position, invariant

1. Introduction

The recognition of specific signatures in images and signals has long been of interest. Powerful techniques exist for their detection and classification, but these techniques are often defeated by changes or variations in the signature. These variations often include translation and scale changes. Methods exist for transforming the signal/image so that the result is invariant to these disturbances. Translation and scaling are well understood in a mathematical sense, so it is fairly straightforward to design methods which yield a transformed form of the data wherein these effects are removed. There are other variations which are not well described mathematically or are not mathematically tractable in terms of reasonable transformations. This paper describes a combination of techniques which allow scale and translation invariant transformations to be used as one step of the signature recognition process. This is followed by an approach which separates the entities to be classified into a number of subsets characterized by additional variations. A new method is introduced to identify the subset to which the specific entity at hand belongs so that classifiers specific to that subset can be used. A two dimensional image is the basic starting point for the technique. This may be the actual image of an object or the two dimensional form of a signal representation such as a time-frequency distribution. We have had some success in representing images in terms of two spatial dimensions and two spatial frequency dimensions. It is clear that this representation captures some unique features of the image, but computation times and memory presently make this approach impractical.

Classification of words appearing in different fonts and sizes serves to illustrate the methods developed. The specific goal set by the sponsor of the research was to spot facsimile corrupted words in bitmapped representations for a variety of fonts and font sizes. However, the approach is quite general and may be applied to a variety of problems and signals.

A representation termed the Scale and Translation Invariant Representation (STIR) is utilized here (Williams et al. 1998). It has desirable properties for pattern recognition under certain conditions. The object to be recognized must have consistent shape and appear on a constant intensity background. Using autocorrelation and the scale transform, one may produce STIRs which are identical for examples that have been translated on the background or scaled (compressed or dilated) along one or more axes.

Concepts borrowed from high-resolution spectral analysis, but adapted uniquely to the problem of classifying these STIRs have been found to be useful in this context. In high resolution frequency estimation, the noise subspace eigenvectors of the autocorrelation matrix are used. Pisarenko harmonic decomposition (Pisarenko 1973) employs the orthogonality of the noise subspace to the signal vectors to estimate sinusoidal frequencies. This idea is used in the classification of signals following STIR processing.

A standard approach is to use the training data to generate templates for each class. A similarity measure, such as correlation coefficient, between the processed test data and each template is calculated and the test data is declared to be in the class corresponding to the largest similarity measure. In contrast, in the noise subspace approach, an orthogonal subspace is created for each class of training data. A measure of the projection of the test data onto each of these subspaces is calculated. Test data matching a given class should be orthogonal to the noise subspace for that class and yield a small projection value.

The STIR and noise subspace classification method are applied to the example of word spotting in bitmapped documents. For a bitmapped word input, the data are represented invariantly to translation and size, then categorized by font, and finally classified by word. This combination of methods is applicable to many pattern recognition problems of any dimension.

2. Background

The approach presented in this paper appears to be quite novel for this area of application. There is not a lot of previous work that needs to be cited in order to build up to the approach. However, a few words on related approaches are appropriate. In addition, the time-frequency motivations behind the present work reveal the progression of the concepts and how they came to be applied to the word spotting problem.

2.1. *Character and word spotting*

Word and character spotting or recognition in documents have been a topic of interest for many years. A comprehensive review by Mori, Suen and Yamamoto covers the field up to 1982 (Mori et al. 1982). More recently, Kahan et al. (1987) described a system that recognizes text of various fonts and sizes for the Roman alphabet. Shape extraction was performed on the graph of a run-length encoding of a binary image. A shape clustering algorithm was used to map the results into binary features which were fed into a Baysian classifier. Classification was better than 97% on mixtures of six dissimilar fonts and over 99% on single fonts over a range of font sizes.

2.2. Hidden Markov model approaches

The directions being taken now depart from historical approaches which depended on template matching and edge tracing. Many of these approaches have historically involved the "direct approach" wherein one tries to capture such obvious features. An approach based on hidden Markov processes (HMM) is more in line with the statistical approaches that we wish to employ, however. Recent reports (Agazzi and Kuo 1993, Chen et al. 1993, Ho et al. 1990, Kuo and Agazzi 1994) might be considered to reflect the present success using this technique in word spotting in documents. The work of Chen et al. (1993) represents the use of HMM techniques wherein the results were presented in the form of an ROC curve. They achieved a 95% detection rate for a false alarm rate of 25% using text in eight fonts from the table of contents of five journals and conference proceedings. This work is of particular interest, since ROC results for our method will be presented later in this paper. Kuo and Agazzi (1994) have carried out an ambitious evaluation of an HMM based system. The system was evaluated on a synthetically created database that contains about 26 000 words. They achieve a recognition accuracy using a 2-D HMM of 99% when words in testing and training sets are of the same font size, and 96% when they are in different sizes. In the latter case, the conventional 1-D HMM achieves only a 70% accuracy rate. The work of Ho et al. (1990) specifically addresses degraded words which are also the subject of the present paper.

Word spotting in speech has also been accomplished using HMM techniques (Yen and Kuo 1995). These studies were carried out using radio dialog. Both the HMM document studies and the HMM radio speech studies form an appropriate benchmark for our research. It seems that HMM techniques are widely regarded to be the best present approach available.

2.3. The scale transform

The scale transform has been described by Cohen (1993) to be:

$$D(c) = \frac{1}{\sqrt{2\pi}} \int_0^\infty f(t) \frac{e^{-jc\ln t}}{\sqrt{t}}\, dt \tag{1}$$

The scale transform is of interest to this paper because it is able to remove the effect of scale in the images. There is an analogy to the Fourier transform. The Fourier transform of a signal, $x(t)$ and the Fourier transform of a shifted version of that signal, $x(t - t_o)$ differ only by a phase factor.

$$F[x(t - t_o)] = X_o(\omega) = X(\omega)\, e^{-j\omega t_o} \tag{2}$$

so that

$$|X(\omega)| = |X_o(\omega)| \tag{3}$$

In a like manner, the scale transform of $x(at)$ differs from the scale transform of $x(t)$ only by a phase factor, so that the magnitudes of the scale transform of $x(t)$ and $\sqrt{a}\,x(at)$ are identical. Thus the effect of size changes can be removed by using only the magnitude of the scale transform.

$$|D(c)| = |D_a(c)| \tag{4}$$

We have developed a discrete form of the scale transform (Williams and Zalubas 1996, Zalubas and Williams 1995) which can be computed efficiently. One might question the use of the scale transform rather than the more well-known Mellin transform. One reason is that the standard Mellin transform weights signal components in lower time more than in higher time. A second reason is the relationship of scale to wavelet concepts and the insights it brings in this light. We have adapted Mellin transform ideas from the paper by Zwicke and Kiss (1983) for our scale transform applications.

2.4. Scale transform applications to speech

Cohen, in his theory of scale, treats scale as a physical variable just like frequency. The kind of scale desribed by Cohen is related to the wavelet type of scale, but is not the same. Marinovich et al. (1995), have shown that the concept of scale can be profitably used to understand the nature of the speech signal. To illustrate the main issue that motivated the work consider the example of a grown person and a child making the sound “ah”. The time-frequency structure of these two sounds is different. The frequency bands, the formants, are at different locations; the frequency spacings between the formants are different. However, one interprets the sounds as the same. How is that possible? If the two sounds had spectra that were scaled versions of each other and the auditory system unscaled them, they would be perceived the same. Indeed, it has been experimentally shown for vowels that sounds which one categorizes as the same, but produced by different size vocal tracts have spectra that are scaled versions of each other, where the scaling occurs in the frequency domain.

It clear that we are able to cope with the various types of transformations performed on the signals and images which occur in our environment—time shifts, frequency shifts, translations in space and scale changes. In order to devise systems which will perform under these conditions it is neccessary to cope with these changes in the design of the algorithms.

2.5. Tools for invariant image representation

Several tools have been developed for representation of the images we seek to classify. In this section we will confine ourselves to 2-D images. These results were obtained when we decided to “back-off” from the full 4-D representations mentioned previously for a time due to their representational and computational complexities. The idea was to gain insight in a simpler setting and then return to the more complex representations as experience and advances in computor technology dictate. The more complicated 4-D functions required for the 4-D representations were thus temporarily replaced by 2-D autocorrelations. The steps in the image processing algorithm are:

- Autocorrelation of the 2-D representations to remove translational effects.
- 2-D scale transformations of the the autocorrelation result to remove scaling effects.
- Partition of the results into subsets which reflect extraneous variations of the data.

Classification of the image involves two steps. These are:

- Determine the subset to which the unknown image belongs.
- Use the classifier designed for that specific subset to classify the image.

2.6. Computation of the 2-D autocorrelation

The autocorrelation function of the signal provides the stable origin needed by the scale transform. Since the autocorrelation simultaneously sums over all points of a function, shifting of a signal over the plane does not affect the values for each lag. It is well known that autocorrelation removes translational effects in images and specifically in optical character recognition (OCR) methods (Mori et al. 1982).

The 2-D discrete autocorrelation may be carried out as follows:[1]

$$A(k_1, k_2) = \sum_{n_1} \sum_{n_2} a(n_1, n_2) a(n_1 - k_1, n_2 - k_2) \tag{5}$$

where $a(n_1, n_2)$ is the image. The image need not be centered within the bitmap representation, which has finite support in n_1, n_2. The bitmap is assumed zero outside of the specific bitmap support region chosen. Autocorrelations for characters 'a' and 'b' in three different fonts are shown in figure 1.

The 0,0 lag point provides an origin from which the autocorrelation function scales. Another feature of the 2-D autocorrelation function is the symmetry $A(k_1, k_2) = A(-k_1, -k_2)$. Hence, the first and fourth quadrants together contain complete information about the entire autocorrelation lag plane. This attribute will be used in applying the scale transform. For pattern recognition purposes, one must be aware of the loss of information which results from obtaining the autocorrelation of the signal. The goal here is to remove only translation effects. Unfortunately, due to the symmetry of the autocorrelation function, an ambiguity in the orientation of the original image is introduced. The autocorrelation of an image is indistinguishable from the autocorrelation of a 180 degree rotated version of the image. This is due to the masking of phase information when the autocorrelation is applied to a signal.

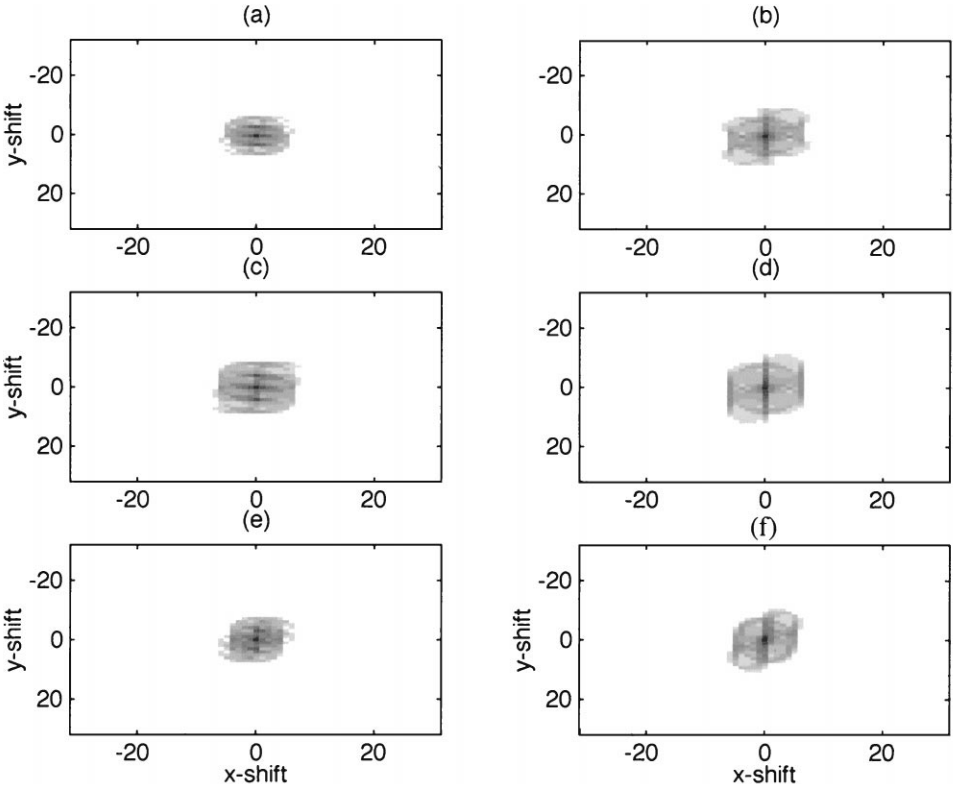


Figure 1. Typical 2-D autocorrelation result for a 63 × 63 pixel bitmap of two lowercase letters. (a) **‘a’** in Courier (12pt), (b) **‘b’** in Courier (12pt), (c) **‘a’** in Helvetica (12pt), (d) **‘b’** in Helvetica (12pt), (e) **‘a’** in Times (12pt), (f) **‘b’** in Times (12pt).

2.7. 2-D direct scale transform

Since the scale transform is based on exponential sampling relative to the origin, the entire autocorrelation plane cannot be dealt with at once. Since both lag values in the first quadrant index from zero in the first quadrant, the scale transform may be directly applied. The lag axes in the fourth quadrant, however, aren’t both positive, so reindexing is necessary. For each quadrant the axes must be included, since the scale transform indexing is based relative to the origin.

Hence, define two discrete quadrant functions of the Autocorrelation plane as follows:

$$Q_1(k_1, k_2) = A(k_1, k_2) \qquad \text{for } k_1, k_2 \geq 0 \tag{6}$$

$$Q_2(k_1, k_2) = A(k_1, -k_2) \qquad \text{for } k_1, k_2 \geq 0 \tag{7}$$

A 2-D scale transform approximation is implemented by applying the 1D scale transform algorithm in Eq. (1) first to the rows then to the columns of a matrix of values. Applying such a 2-D scale transform to Q_1 and Q_2 and taking the magnitude of the result yields two 2-D matrices of scale coefficients. We call this the Autocorrelation Function Scale Transform (ACFSX). The size of these matrices is determined by the number of row and column scale values selected.

Since the autocorrelation function input was not energy normalized, normalization of the scale magnitudes is required for a scale invariant representation. Since the scale transform is a linear transform, normalization may be done by a variety of methods to generate an appropriate result.

The normalized scale transformed quadrant functions represent a STIR of the original 2-D input. Since it is not possible to calculate the scale coefficient $D(c)$ for every c, a set of scales is chosen for computation of the scale transform coefficients. Hence, the transform is not invertible. In addition to providing a scale invariant representation, other signal information is lost. The usefulness of the STIR is dependent on its implementation and application. For the very common case of a 2-D function sampled into a matrix of discrete values, we have developed a classification scheme which can be used with STIRs as the inputs.

The novel image classification approach involves two steps. These are:

- Determine the subset to which the unknown image belongs.
- Use the classifier designed for that specific subset to classify the image.

2.8. *Designing the classifier*

The next task is to design a classifier. Suppose that the invariant form is characterized by a two dimensional representation $\Delta(p, q)$. This 2-D representation may be decomposed using eigensystem techniques as

$$\Delta(p, q) = \sum_j a_j \beta_j(p, q) \tag{8}$$

where the $\beta_j(p, q)$ are eigenimages and the a_j are the eigenvalues of the decomposition. The eigensystem decomposition is carried out on a collection of $\Delta(p, q)$ examples coming from the classes of objects (signals or images) that are of interest. The eigensystem decomposition then provides an ordered set of eigenimages ordered according to their eigenvalues. Although the eventual goal is to use true two dimensional eigenimage analysis, suitable algorithms to accomplish this have not been identified. One may utilize a simpler one dimensional approach which lends itself to readily available algorithms.

2.9. *Classification of patterns*

Our technique for pattern classification uses STIR images decomposed into an orthonormal set of descriptors, using a concept borrowed from Pisarenko's harmonic decomposition (Pisarenko 1973, Williams and Zalubas 1996). The Karhonen-Loève transform is a means of accomplishing this. The singular value decomposition (SVD) provides equivalent results. The STIRs of each exemplar in a class are shaped into a row vector by concatenating rows

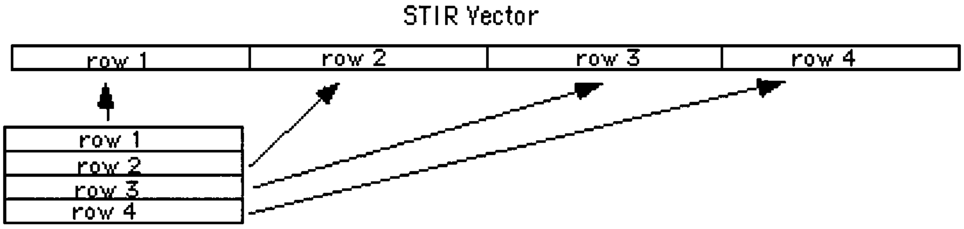


Figure 2. Forming one of the components of the STIR vector from one of the ACFSX matrices.

of the two STIR matrices (See figure 2). These row vectors are stacked to form a matrix representing the class. The SVD is then applied to extract essential features of the set of vectors. Provided that a sufficient number of scale coefficients are calculated, singular values of zero will result. Right singular vectors corresponding to zero singular value define a subspace orthogonal to the class of STIR vector representations.

In classifying a test signal, generate its STIR vector. Compute for each class the sum of inner product magnitudes of the STIR vector with the orthogonal subspace vectors. If the sum is zero, then the test signal must be a member of the corresponding class. In practice, one does not obtain a zero sum with the proper subspace class, but the sum resulting from the proper class has the smallest magnitude relative to sums from calculation with other class subspaces.

In addition to the invariances, STIRs have the desirable property that for a fixed set of row and column scales the sizes of all STIR matrices are identical, regardless of the size of the input matrices. Hence, inputs from different sources may be treated identically once processed into STIR images.

2.10. Classification of characters

The initial approach which was taken was to decompose the STIR images via singular value eigendecomposition (SVD) in order to provide an orthonormal set of descriptors. In order to accomplish this, the STIR images of the characters were reshaped into a single vector by concatonating the rows of the STIR matrix to form a STIR vector. A new matrix consisting of all of the characters of interest for a range of sizes and the three fonts was formed from these vectors. The SVD was applied to extract essential features of the of the set of vectors. Several singular vectors with the largest singular values were chosen as features. Unfortunately, classification results were not impressive. The font variations were sufficient to reduce classification accuracy below acceptable levels. In order to combat this problem, the results were separated into subsets, with one subset representing each font type. Then, a novel orthogonal noise subspace method was used to identify the specific font used to produce the unknown character.

Almost all of the undesired variation due to shift and scale may be squeezed out of the final invariant form. There may still be some residual effects due to discretization and computation.

The $N \times N$ STIR matrices may be converted into vectors of length $N \times N$ by either concatenating the rows or columns. There are two unique quadrants matrices as defined in Eqs. (6) and (7). Each yields a vector. These vectors are then concatenated to form an $N \times 2N$ or $2N^2$ element **STIR vector**. Then, readily available Singular Value Decomposition (SVD) techniques may be applied to the vectorized set of images. If there are M exemplars each with $2N^2$ elements, one could stack them to form an $M(\text{rows}) \times 2N^2(\text{columns})$ matrix. The SVD would be applied to this matrix. Suppose there are several different extraneous variations in the "invariant representations" caused by a variety of factors. Representation by a variety of font types and pixelation errors as well as FAX noise are examples of these extraneous variations. Conversion of the STIR matrices into STIR vectors is illustrated in figure 2. The SVD has the property of isolating common features of such a set of vectors and this property can be used very well in this application.

3. Noise subspace method details

Suppose that there are M bitmapped images of the training set of words. This would yield an $M(\text{rows}) \times 2N^2(\text{columns})$ matrix of STIR vectors. The STIR vectors have a large number of elements. Usually, for classification methods to work, one wishes to have a considerably greater number of representations of the signal vectors than there are elements in those representations. Here, we have exactly the opposite. There are many more elements in the vectorized 2-D forms than there are vectorized 2-D forms. This is usually a statistical nightmare. However, suppose there are M examples ($M \ll 2N^2$). Then the SVD produces $2N^2$ orthogonal eigenvectors, the first M of which form a complete orthonormal basis for the set of STIR vectors. The remaining SVD eigenvectors (the noise eigenvectors) must be orthogonal to all of the original STIR vectors. Suppose that we now obtain a new example. Convert it into the STIR matrix form and then, vectorize it to form the STIR vector. If it belongs to the set of STIR vectors used to produce the SVD results, then it should be *orthogonal* to all of the noise eigenvectors produced by the SVD. Therefore, its projection on any of the noise eigenvectors should be zero. If we have carried out the whole process through the SVD for a number of different sets of signals, we should find the projection of the STIR vector of the unknown signal on the noise eigenvectors of each set of signals. The smallest result will be theoretically obtained when this is done using the noise eigenvectors of the set to which the signal belongs. This idea may be expressed more formally. The SVD is performed on the set of matrices formed by STIR vectors for each font. Denote this matrix to be Q_k, where k is the kth font. The variety of the STIR vectors which form the rows of this matrix should cover a full range of variations of all of the characters represented by that font. Suppose there are N_c characters of interest. This could be all letters, special characters and numbers. This number (N_c) would be multiplied by the number of font sizes of interest to yield N_{tot}. Application of the SVD yields

$$U_k S_k V_k' = Q_k \tag{9}$$

The N_{tot} columns of V form a basis for the rows of Q_k. If there are columns left over they will be orthogonal to all of the rows of Q_k. This may be accomplished by designing the STIR representations such that the STIR vectors meet this condition by having more elements than there are rows in the Q_k matrix. For illustration, suppose one picks a "noise vector" Z from this set of orthogonal columns, call it Z_k. It is a column vector. Then,

$$Q_k Z_k = O \tag{10}$$

Where "O" is a row vector of zeros. However, for another font representation, Q_r,

$$Q_r Z_k \neq 0 \tag{11}$$

This provides a powerful means of determining whether or not a STIR vector belongs to the subspace of interest. Find

$$s_u = STIR_u Z_k \tag{12}$$

If the selection value, s_u, belongs to the kth subset, its projection will be zero. If not, its projection will most likely be large. Thus, one may detect the font representation by this means.

3.1. *Partition into font subsets and detecting font*

The difficulties due to font differences have been solved by first detecting the font in which the unknown character is represented. Ideally, the subspaces represented by the different fonts would be disjoint, so that one may discover which font the unknown is in and then chose a font specific classifier to home in on the character. This is not quite true, but the subspaces for the different fonts are sufficiently distinct to provide good font detection.

3.2. *Font specific classification of characters and words*

Suppose one has the bitmap of an unknown character. The STIR representation of the character is projected onto the subspace of each font. We can thus find a selection value that determines the font class membership of the unknown character. Next, the classifier designed for the specific font subspace is used to classify the character. This subspace is built from all of the characters in the search set represented over a number of sizes. We have used a wide range of font sizes from 10 to 50 in building these font specific subspaces. This method works very well using a number of classification methods. We use the most important features extracted from the SVD decomposition of the STIR vector to determine the font. Next, we use a font specific subspace method as described in Eqs. (9)–(12) to identify the character. Typically, we obtain 100 percent correct character classification, even with fax corrupted bitmaps (Williams and Zalubas 1996). On rare occasion, incorrect font subspaces are selected. Nevertheless, the correct character classification still results. This is the ultimate goal, after all. Also, on rare occasions, "b" is confused with "p" and "u" is confused with "n" in some fonts represented by noisy bitmaps. This is due to strong symmetry for such characters in the STIR representation. It should be easy to confirm the correct result by other simple means in such cases.

4. Application: Wordspotting

As mentioned in the Introduction, the specific goal of the project was to develop a new method for wordspotting for use with bitmapped representations of faxed documents. Many research techniques and commercially available techniques perform well when the document is available as a clean bitmapped representation of text. Performance substantially deteriorates when the bitmapped representation is from a faxed document, however. In fact, the result is often useless due to the large number of errors.

An example which shows how STIR and the SVD noise subspace index are combined to perform as a size independent word recognition classifier is given here. The method is also robust under fax corruption. A complete document identification system incorporates much more than the pattern classifier presented here. This application shows the viability of the method for pattern classification under fax corruption. Performance on multiple fonts is a straightforward extension of the single font wordspotter.

STIRs and noise subspace methodology are used to spot a word in text independent of size or translation. Omitted is the task of segmentating an image into individual word bitmaps. Given the additional white space between words in a document, the segmentation task is much easier than character segmentation. For many documents, this may be simply performed by breaking text at intervals of white space which exceed a given distance.

Each segmented bitmap is considered as an isolated recognition task. Contextual information such as positioning within a line, height/width ratio, and pixel density is not used.

A preliminary data set was produced by faxing documents and subjecting the clean bitmapped result and the bitmapped result after faxing to analysis. It was confirmed that faxing produces a peculiar type of noise that is unlike additive noise. Broken and touching characters are often seen (Chen et al. 1993, Ho et al. 1990). Examples of fax corruption can be seen in some of the examples presented later in this paper. Preliminary experiments showed that words consisting of letters with slanted parts such as v, w and x were more troublesome than letters with horizontal or vertical parts. ‘van’ and ‘vax’ were found to be more difficult to distinguish than other pairs examined. Therefore, the decision was made to concentrate on these words, since other words would be considerably easier to spot in comparison. Various words using combinations of v, a, n and x with other letters were constructed in order to further concentrate this worst case scenario. Three letter words were chosen for two reasons. First, it was possible to creat a reasonable set of three letter words that were one or two characters different from ‘van’ and ‘vax’. Second, longer words create bitmaps of increasing size with resulting increases in time to carry out test runs. This is a particular concern for future plans which involve expanding the method to higher dimensions.

Obtaining good sets of fax corrupted words is not a trivial task. An undergraduate summer student spent a number of weeks constructing, printing, faxing and bitmap converting the word sets used in our studies.

The word set result to be presented consists of three letter words, all in lowercase. The word to spot was ‘van’. Figure 3 shows the bitmapped image of ‘van’. Figure 4 shows its autocorrelation. 75 words other than ‘van’ were generated by altering one letter of the three. Helvetica was the font examined. Text in sizes 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, and 20

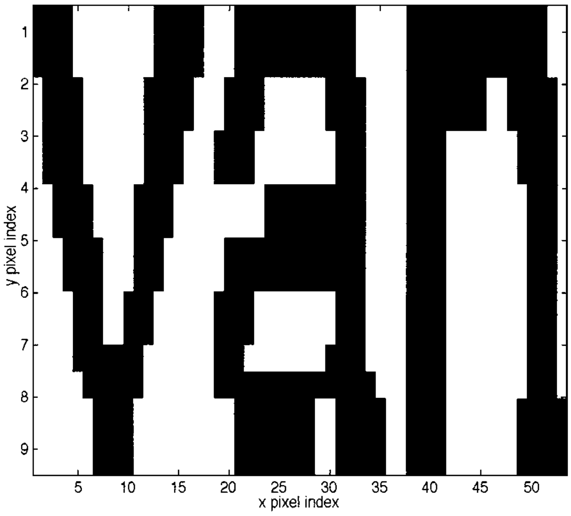


Figure 3. Bitmapped representation of van.

point was used for training. Bitmaps from faxed versions of clean printed copy were used as the input signals. The training text consisted of five instances of the word to be spotted in each size. An example of the bitmaps of the word 'van' in 10 point appear in figure 5. These were dubbed the "Placebo Words" by the summer intern and she called the set of words which differed from 'van' plus some instances of 'van' the "Waldo Words". This clever allusion to the cartoon character Waldo, where the task is to find Waldo in a complex picture, was adopted to describe the word sets. The "Waldo Words" are shown in figure 6.

The classification methodology was tested on 10 point faxed 'words' in each of the fonts. Hence, the recognition tool is being tested on a size of text different from any size used in training. In this character recognizer, Font is determined first. For each font, exemplars in the four training sizes are available for each of the 26 characters, a total of 104 training characters.

Every STIR row vector is generated by the steps of autocorrelation, scale transform, and reshaping to a vector. To illustrate, consider a 9 point bitmap of the desired word 'van'.

The first and fourth quadrants are scale transformed using an interval distance $T = 1$ with row and column scale values of 0.1, 0.4, 0.7, 1.0, 1.3, 1.6, 1.9, 2.2, 2.5, 2.8. Figure 7 shows the matrices of magnitudes of these scale transform coefficients, the STIR values. Note that the difference in scale values between quadrants is very small. This similarity is exhibited in the scale coefficient magnitudes for most data encountered. Another notable feature is that the scale magnitudes generally show a roughly exponential drop off. These coefficient magnitudes reformed as an STIR row vector give the appearance shown in figure 8.

Since, in this example, only one word is to be spotted, STIR training vectors formed only one matrix. This implies an SVD on 55 STIR vectors since we are using 5 instances of 'van' in each point size 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20. The length of each row

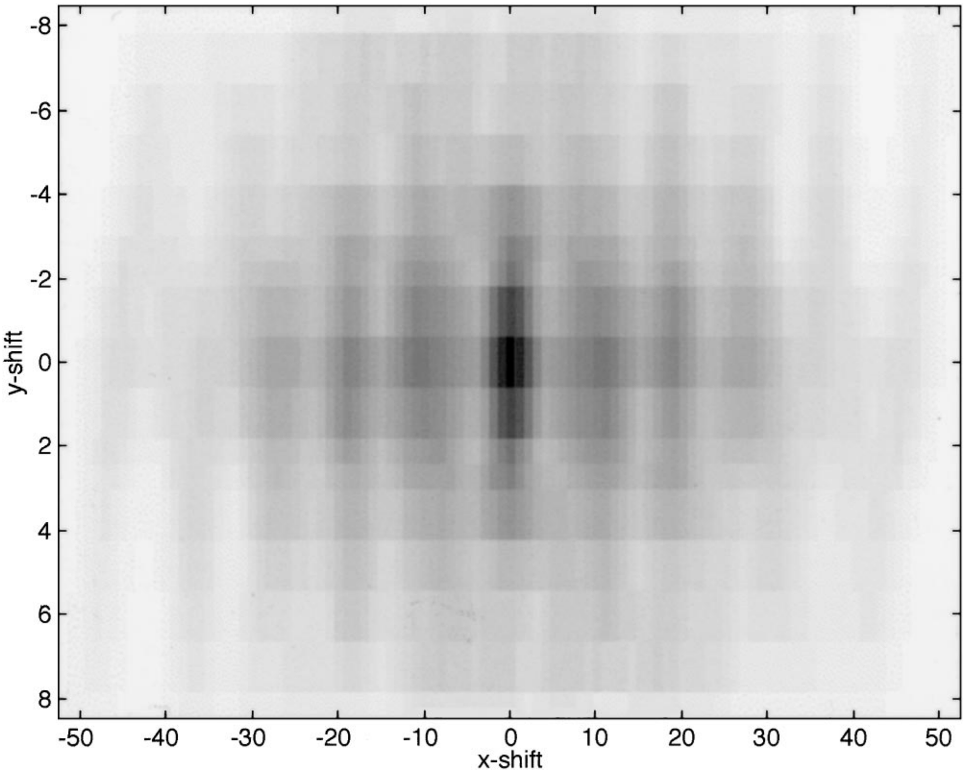
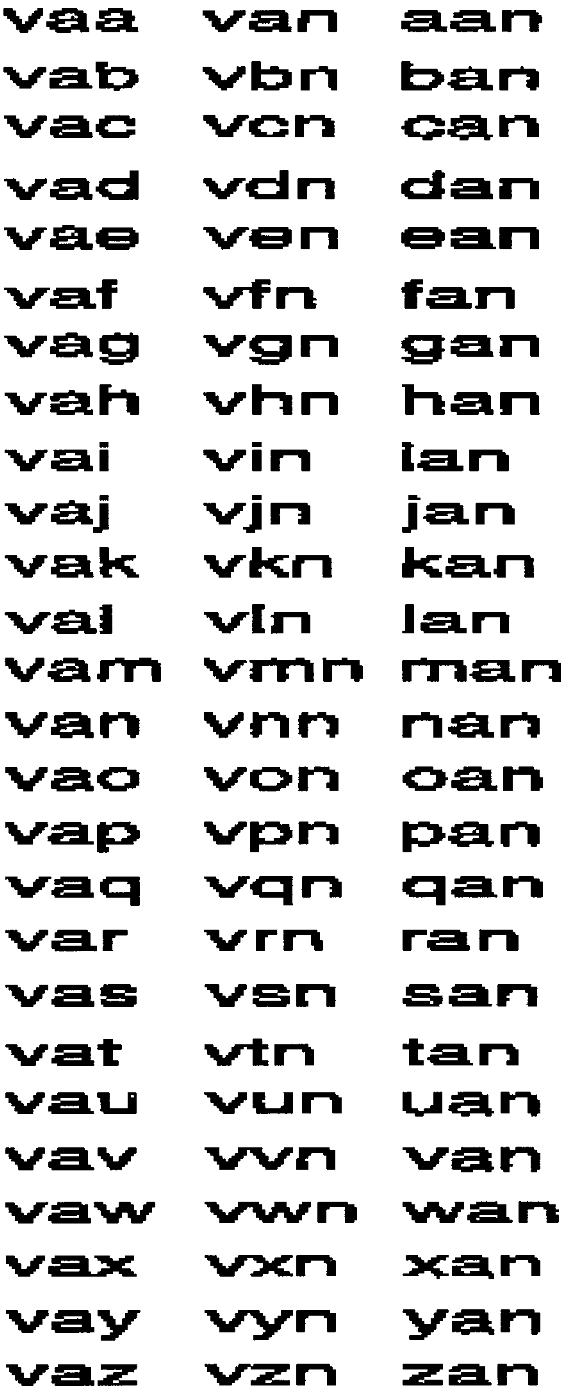


Figure 4. 2-D Autocorrelation of ‘van’.

is determined by the number of row and column scales chosen for calculation. The STIR row vectors each have 200 elements because, choice of row and column scales in the scale transform dictates a 10 by 10 matrix output for each autocorrelation quadrant, regardless of the size of each autocorrelation quadrant. Thus, the SVD for each font will yield noise vectors corresponding to $200 - 55 = 145$ singular values with zero magnitude. Calculating the sum of inner product magnitudes between these orthogonal vectors and a test STIR vector yields a selection value for each font. If the result is zero, then the unknown character must be represented in that font. In practice, one does not obtain a zero inner product with the correct font noise vectors. However, the correct font should correspond to the matrix generating the smallest selection value.

The similar words in 10 point, an untrained size, were processed into STIR vectors and selection value, (SV) were calculated. The SVs of the instances of ‘van’ character sequence did indeed show low values as shown in figure 9. The ROC curve is shown in figure 10.

Two sets of words were used the “Waldo Words”, those that were close to ‘van’ and the “Placebo Words”, those consisting of variations of ‘van’. These words are shown in figures 5 and 6. These words were constructed from faxed images. One can readily see the fax corruption.



vaa	van	aan
vab	vbn	ban
vac	vcn	can
vad	vdn	dan
vae	ven	ean
vaf	vfn	fan
vag	vgn	gan
vah	vhn	han
vai	vin	ian
vaj	vjn	jan
vak	vkn	kan
val	vln	lan
vam	vmn	man
van	vnn	nan
vao	von	oan
vap	vpn	pan
vaq	vqn	qan
var	vrn	ran
vas	vsn	san
vat	vtn	tan
vau	vun	uan
vav	vvn	van
vaw	vwn	wan
vax	vxn	xan
vay	vyn	yan
vaz	vzn	zan

Figure 5. Placebo Words in 10pt representation.

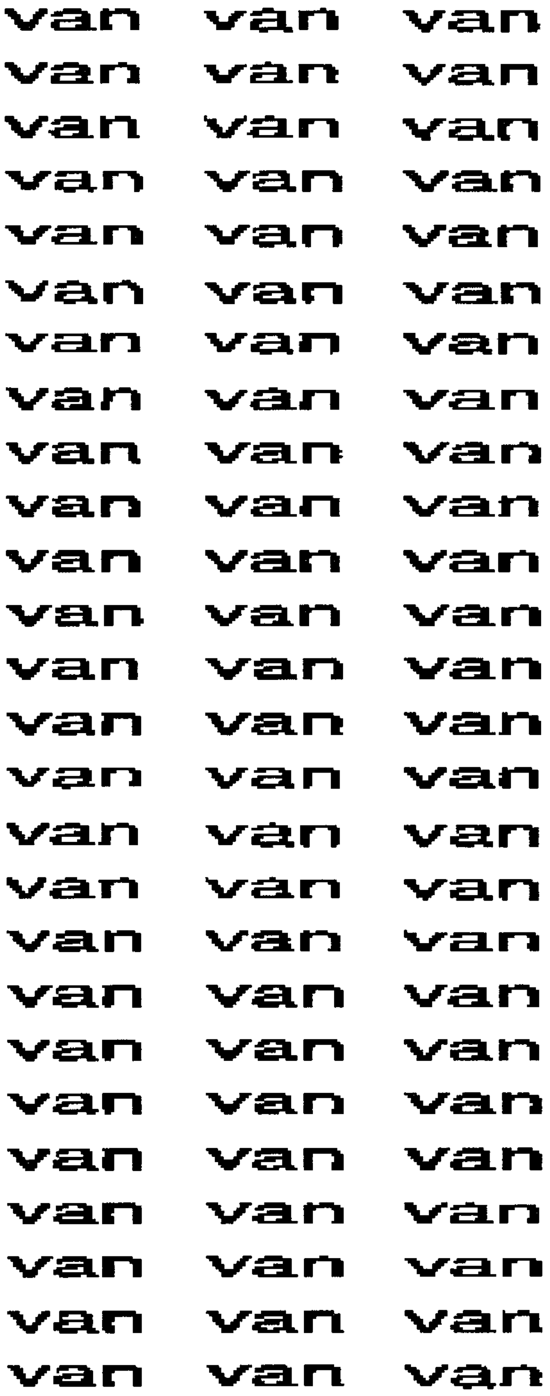


Figure 6. Waldo words in 10pt representation.

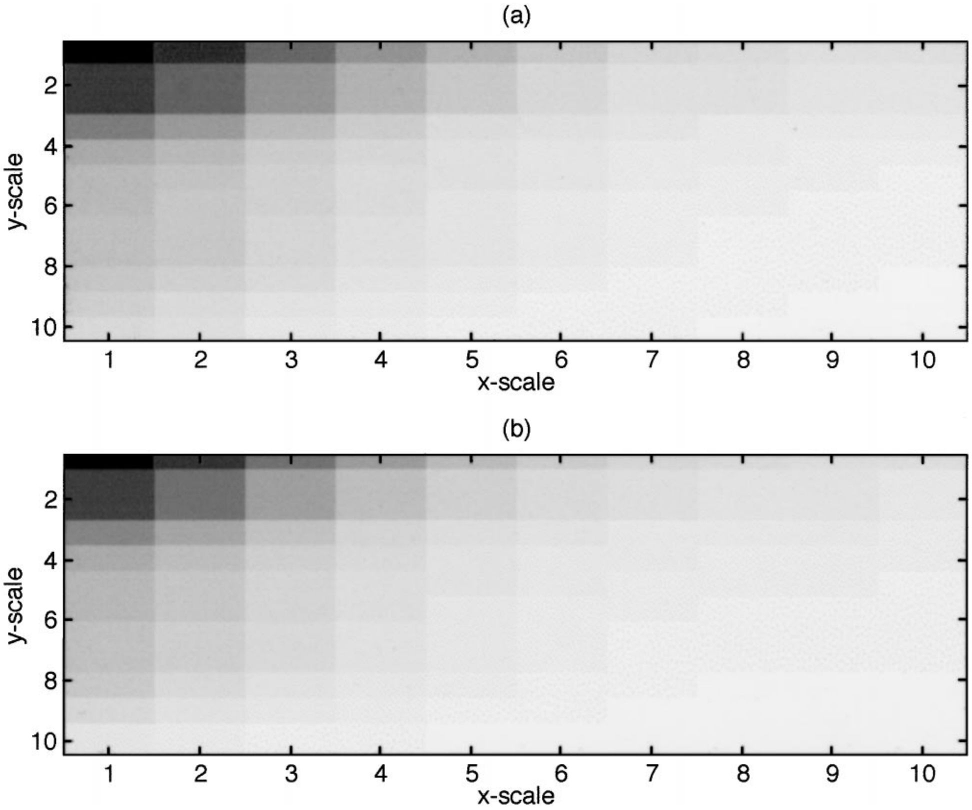


Figure 7. 2-D Scale Transforms from the two unique quadrants (a) and (b).

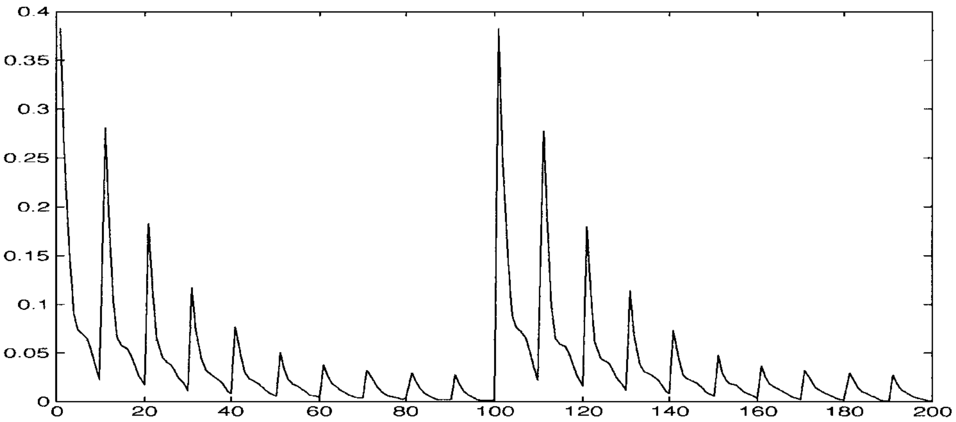


Figure 8. The STIR vector for ‘van’.

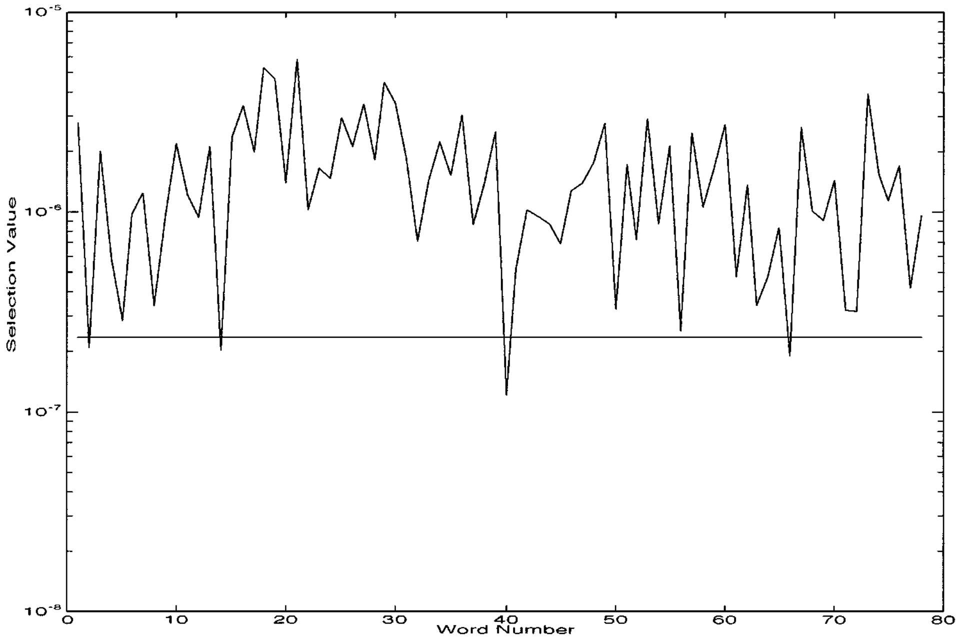


Figure 9. Selection value results. The upper plot represents the Waldo word selection values. The circles represent the Placebo word results. The horizontal line represents the threshold chosen.

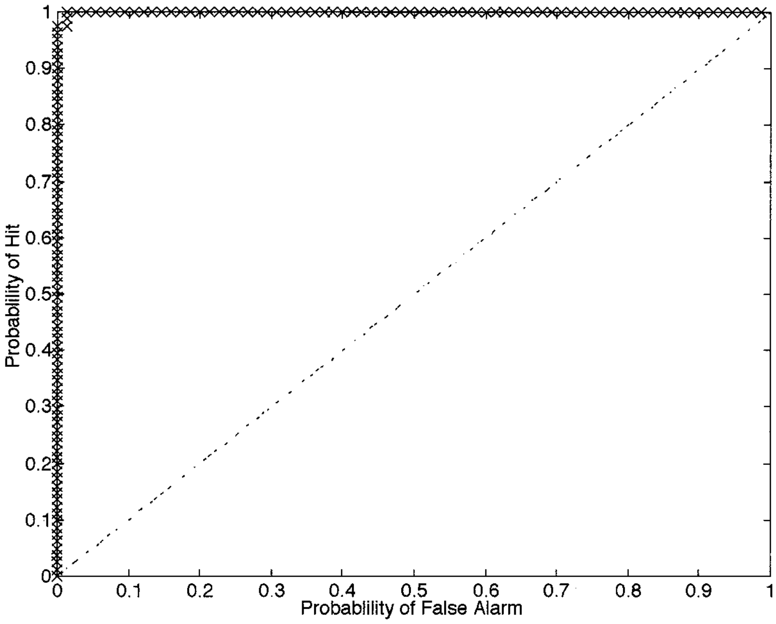


Figure 10. ROC results.The word results are x's. The dotted line represents chance performance.

The 2-D autocorrelation of ‘van’ is shown in figure 4. The 2-D scale transforms of two unique quadrants of the 2-D autocorrelation are shown in figure 7. The STIR vector produced by concatenating the rows of the 2-D scale transforms is shown in figure 8.

It was assumed that an effective word segmentation algorithm had been applied to the bitmapped page. This is not a trivial problem, but some fairly straightforward software was developed to accomplish this task quite well. Sophisticated methods are available which should be considered in a fully developed system (Etemad et al. 1994a, Etemad et al. 1994b, Reed and Wechsler 1991). In order to increase the difficulty of classification the word set used with ‘van’ and ‘vax’ was constructed such that the other words were very close to ‘van’ and ‘vax’, differing only by a single letter.

Results were quite good. ‘van’ appeared three times in a series of 78 words. The algorithm found ‘van’ each time (3 Hits) and mistakenly identified another word as ‘van’ one time (1 False Alarm) for the threshold shown. See figure 9. The word which was incorrectly identified as ‘van’ was ‘ven’ with a corrupted ‘e’. This word can be found as the third word in the middle column of the ‘Waldo Words’ in figure 6. In order to assess the method more fully, the threshold was varied over a range of values to generate a Receiver Operating Characteristic (ROC) shown in figure 10.

5. Discussion

The work statement for the project which generated these results was focussed on developing a new method for spotting words in faxed documents. The need for scanning such documents for certain words has created a crisis situation in many areas. Recent declassification legislation has provided a strong impetus for improved techniques. Some documents are available after multiple faxing or multiple copying and are thus considerably degraded. It was not the purpose of this project to provide a finished method which had been tested for a very large number of words with a large number of types and levels of corruption. The intent of the project was to provide promising concepts which could be brought into governmental evaluation programs such that a number of techniques could be compared using common data. Comparisons with other techniques are difficult to orchestrate due to lack of details and the requisite expertise in “tweaking” the parameters of the method in order to obtain the best result. Such comparisons are sorely needed, however and it is hoped that ongoing governmental programs will provide these comparisons.

Generating the data sets for this project took quite a lot of time. In order to show the efficacy of the method we decided to do an exploratory test to identify troublesome words and then show that the method could do well even in these difficult situations. It is expected that results may even be much better with a broad range of less troublesome words. However, proof of this conjecture awaits a much longer term and more ambitious project effort than our time and resources allowed.

The examples provided show the potential for application of the STIR and noise subspace discrimination methods to character recognition. A selection value threshold could be added to reject symbols which are not among the valid set of characters. In addition, the detection method might be improved considerably. Some impressive results have been obtained by Warke and Orsak (1996) in classifying faces using an information theoretic method.

A perusal of such problems as face recognition suggests that the methods described in this paper need not be confined to character and word spotting. Any bitmapped representation may be brought into this methodology. Logos, Chinese characters, images of ships and planes, faces, sounds (via time-frequency representations) (Williams et al. 1998) and many other types of signals and images might be profitably identified using this approach. The methods may also be readily extended to higher dimensional spatial-temporal-frequency-wavenumber represention wherein complex objects may be identified. A major problem in doing this is the size of the representations produced. However, with rapily increasing RAM allocations in computers, this too may be possible.

The nature of some of the errors seen when using this method has prompted some of our psychologist and neurologist colleagues to comment that the errors are similar to some of the phenomena observed in human perception. Dyslexia, for example, is a condition wherein words and letters are perceived as being reversed. We have observed exactly this type of phenomenon with characters using the STIR method. This does not occur when the letters are uncorrupted, but does begin to occur when small details of the character are altered by faxing. Presumably, this could also occur with words as well. This is due to the symmetry characteristics of the ACFSX matrix in both the horizontal and vertical directions. There are several simple fixes for this which include a secondary check for right-left and up-down orientation should it become a problem in large scale applications.

The algorithms for the method were written in Matlab© which is an interpretive language. Still, the example given in this paper runs quickly enough to quite satisfactorily demonstrate on a laptop computer in a seminar setting. With increasing computer speeds and memory, coupled with good C coded routines, the speed should be quite reasonable for larger scale application.

5.1. Conclusions

The methods described in this paper provide a very robust means of identifying target words in a bitmapped document. The various parts of the algorithm are not unique in and of themselves. The scale transform (or Mellin transform) is an obvious way of removing scale effects. Autocorrelation has been used in many applications to dispense with absolute time or displacement. The noise subspace concept is used in many modern methods for high resolution spectral analysis. However, taken together, in the form suggested in this paper, they yield a powerful new method for character and word spotting in bitmapped documents. There are many points of refinement which remain to be investigated. Each step in the process offers considerable opportunity for refinement. Large scale testing needs to be carried out to fully confirm the method and identify fixes to some of the problems which may occur.

Acknowledgments

We gratefully acknowledge the support of this research via ONR Grant N000014-97-1-0072 and NSA Contract no. MDA904-95-C-2157. We would also like to thank Ronit Reger for her dedicated work preparing the faxed data sets.

Note

[1] The reader will probably recognize that this computation can be done in the frequency domain by taking the squared magnitude of the 2-D Fourier transform of the image with subsequent 2-D inverse Fourier transforming.

References

Agazzi OE and Kuo S (1993) Pseudo two-dimensional hidden Markov models for document recognition. *AT&T Technical Journal*, 72:60–72.

Chen FR, Wilcox LD and Bloomberg DS (1993) Word spotting in scanned images using hidden Markov models. In: Proc. of the IEEE Int. Conf. on Acoust., Speech, and Signal Processing. IEEE, Vol. 5, pp. 1–4.

Cohen L (1993) The scale representation. IEEE Trans. on Signal Processing, 41(12):3275–3292.

Etemad K, Chellappa R and Doermann D (1994a) Document page segmentation by integrating distributed soft decisions. In: Proc. of the IEEE International Conference on Neural Networks, Vol. 6, pp. 4022–4027.

Etemad K, Doermann D and Chellappa R (1994b) Page segmentation using decision integration and wavelet packets. In: Proc. of the 12th IAPR International Conference on Pattern Recognition, Vol. 2, pp. 345–349.

Ho TK, Hull JJ and Srihari SN (1990) A word shape analysis approach to recognition of degraded word images. In: Proc. of the USPS Advanced Technology Conference. United States Postal Service, pp. 217–231.

Kahan S, Pavlidis T and Baird HS (1987) On the recognition of printed characters of any font and size. *PAMI*, 9(2):274–288.

Kuo S and Agazzi OE (1994) Keyword spotting in poorly printed documents using pseudo 2-D hidden Markov models. IEEE Trans. on Pattern Analysis and Machine Intelligence, 16(8):842–848.

Marinovich N, Cohen L, Umesh S and Nelson D (1995) Scale-invariant speech analysis via joint time-frequency-scale processing. Proc. Int. Soc. Opt. Eng., 2569:522–537.

Mori S, Suen CY and Yamamoto K (1982) Historical review of OCR research and development. Proc. IEEE, 80:1029–1092.

Pisarenko VF (1973) The retrieval of harmonics from a covariance function. Geophys. J. Royal Astron. Soc., 33:347–366.

Reed T and Wechsler H (1991) Spatial/spatial-frequency representations for image segmentation and grouping. Image and Vision Computing, 9(3):175–193.

Warke N and Orsak GC (1996) An information theoretic methodology for noisy image classification with application to face recognition. In: Proc. Conf. on Information Science and Systems, Princeton, NJ.

Williams WJ and Zalubas EJ (1996) Separating desired image and signal invariant components from extraneous variations. SPIE: Advanced Signal Processing Algorithms, Architectures and Implementations, 2846:262–272.

Williams WJ, Zalubas EJ, Nickel RM and Hero AO III (1998) Scale and translation invariant methods for enhanced time-frequency pattern recognition. Multidimensional Systems and Signal Processing, 9(4):465–473.

Yen C and Shiaw Kuo S (1995) Degraded gray-scale text recognition using pseudo–2D hidden Markov models and N-best hypotheses. Computer Speech and Language, 9:381–405.

Zalubas EJ and Williams WJ (1995) Discrete scale transform for signal analysis. In: Proc. of the IEEE Int. Conf. on Acoust., Speech, and Signal Processing, Vol. 3, pp. 1557–1561.

Zwicke PE and Kiss JI (1983) A new implementation of the Mellin transform and its application to radar classification of ships. IEEE Trans. on Pattern Analysis and Machine Intelligence, 5(2):191–199.