Fabric Defect Detection Using a GA Tuned Wavelet Filter

Warren Jasper, Jeffrey A. Joines and Joe Brenzovich
Textile Engineering Chemical and Science
North Carolina State University
Raleigh, NC, 27695

Abstract

Many textures such as woven fabrics and composites have a regular and repeating texture. This paper presents a new method to capture the texture information using adaptive wavelet bases. Wavelets are compact functions which can be used to generate a multiresolution analysis. Texture constraints are used to adapt the wavelets to better characterize specific textures. An adapted wavelet basis has very high sensitivity to the abrupt changes in the texture structure caused by defects. This paper will demonstrate how to formulate the problem of solving for the wavelet coefficients as an optimization problem with non-convex constraints using a hybridized genetic algorithm.

1 Introduction

Many textures, such as those in knitted fabrics, exhibit a regular or repeating pattern. For fairly simple repeating patterns, researchers [6, 14] have shown that wavelets, or multiresolution analysis can be used to filter out texture and reveal underlying defects in a fabric. A wavelet based texture characterization and defect detection system has applications in robust on-line inspection systems for woven fabrics and composites. Wavelet functions have seen many applications in image compression following Daubechies work in wavelet theory [3]. There has been limited work in the area of adapted wavelets using different wavelet functions such as Coifman and Wickerhauser [2].

Our work, presented in this paper, derives the wavelet basis directly from the texture data of the image. As was done by Jasper et al. [6], the wavelet coefficients are couched as an optimization problem with non-convex constraints. Then, a genetic algorithm is used to find a global set of wavelet coefficients which minimizes the high frequency components of the filtered image. The effects of shifting the image on the optimized set of coefficients is also explored.

2 Theory

A computationally efficient yet flexible method to develop a multiresolution analysis is by means of the wavelet transform. Wavelet functions are finite duration, orthonormal signals that form a basis for the signal subspace. Wavelets-based MRA decomposes a signal into a smooth subspace (low frequency information) and a detail subspace (high frequency). Current texture analysis tools do not take local texture variations into account and are only able to provide a global description of the texture. By using the adaptive wavelet filter which localizes in space as well as in frequency, we can obtain global texture descriptions along with information about local phenomena.

The objective of this paper is to develop a method to find an “optimal wavelet basis” to represent image texture. This section focuses on finding the optimal adaptive basis for a given texture. The adaptation of the wavelet basis can be accomplished by dynamically altering the wavelet coefficients, and to reject waveforms that do not fit the texture pattern (such as defects). The problem of finding an optimal wavelet basis can be written as a least-squares minimization problem with non-convex constraints, which can be solved using a hybridized genetic algorithm.

2.1 The Daubechies Wavelet Transform

A wavelet transform can be specified by high and low pass filtering a signal using a particular set of numbers called wavelet filter coefficients. A class of compact wavelets (functions which are nonzero over a finite range) was discovered by Daubechies, and includes functions which range from being highly localized (high fractal dimension) to being highly smooth (low fractal dimension) [3]. The simplest case involves only four coefficients. Daubechies [11] wavelets can be formed with an even number of coefficients which satisfy certain orthogonality conditions and approximation conditions of order p. Consider the follow-
ing transformation matrix $W \in \mathbb{R}^{2^n \times 2^n}$ in Equations 1 and 2 where the $c_i$’s represent the coefficients and $H$ and $G$ represent the low and high pass filter respectively. 

$$W = \begin{bmatrix}
c_0 & c_1 & c_2 & c_3 \\
-1 & 0 & -c_2 & -c_3 \\
-c_1 & c_0 & -c_2 & -c_3 \\
... & ... & ... & ...
\end{bmatrix}$$

(1)

$$= \begin{bmatrix}
H \\
G
\end{bmatrix}$$

(2)

The condition that the matrix in Equation (1) is orthogonal is to impose the condition that its inverse is just its transpose. $W^T = W^{-1}$

$$W^T = \begin{bmatrix}
c_0 & c_1 & c_2 & c_3 \\
-1 & 0 & -c_2 & -c_3 \\
-c_1 & c_0 & -c_2 & -c_3 \\
... & ... & ... & ...
\end{bmatrix}$$

(3)

For the matrix $W$ to be orthogonal, $WW^T = I$, the following two equations must hold [15],

$$c_0^2 + c_1^2 + c_2^2 + c_3^2 = 1,$$

$$c_2c_0 + c_3c_1 = 0.$$  

(4)

(5)

This leaves some freedom in the choice of coefficients. If we additionally require the approximation condition of order $p = 2$, then

$$c_3 - c_2 + c_1 - c_0 = 0,$$

$$0c_3 - 1c_2 + 2c_1 - 3c_0 = 0.$$  

(6)

(7)

This condition states that low frequency signals (starting with DC values, ramps, etc) will be filtered by the high frequency filter $G$. Equations (5) and (7) represent 4 nonlinear equations for the four unknown coefficients. The following values represent the Daubechies solution.

$$c_0 = (1 + \sqrt{3})/4\sqrt{2} \quad c_1 = (3 + \sqrt{3})/4\sqrt{2}$$

$$c_2 = (3 - \sqrt{3})/4\sqrt{2} \quad c_3 = (1 - \sqrt{3})/4\sqrt{2}$$

### 2.2 Properties of the Wavelet Transform

In some sense, orthogonal transformations of a function do not alter the function itself, but rather the basis by which we choose to represent the function. The fast Fourier Transform (FFT) of a function changes the basis functions to sines and cosines, which are not compact.

The usefulness of transforms is that they project a function onto a new set of basis functions. If one or more basis functions represent a feature, and all the other basis functions are orthogonal to it, then one can quickly determine if a feature exists in a signal by projecting the signal function onto the new basis.

The texture of woven fabric can be described by periodic functions while a defect such as a missing yarn (a mispick) can be described by a high frequency event in one direction (warp direction) and a low frequency in the other (weft direction). The power spectrum of an image loses its spatial information, since the FFT is nonlocal in space. However, to detect mispicks, we need to detect high frequency events over a small spatial dimension. Thus, we would like to have both frequency and time (space) information.

The wavelet transform has the property of giving both frequency and spatial information about an image. It also can be computed very quickly, of order $n$, for one pass of the wavelet transform which is sufficient for defect detection. The FFT is order $n \log n$.

The advantage of the wavelet transform in particular, and Multi-Resolution Analysis (MRA) in general, is that the signal is decomposed over many scales. A fusion of the features extracted at each scale [9] gives a more robust description of the signal than features extracted at only one scale. Also, in the case of images, wavelet analysis separates the edge information (horizontal and vertical) from the smooth part of the image. Let $c$ denote the coefficients of $G$, the detail filter, in Equation 1 and let $P$ be the matrix of texture information. $P$ is formed from a section of the texture and is assumed to represent the texture from a digitized image. $P$ should not be too small in case some texture variations are lost. In general, it is better to choose a larger $P$ since we want all long term effects of the texture to die down in the detail signal.

The matrix $P$ has the form in Equation 8 where $n$ is the number of wavelet coefficients, and $m$ is the number of points in the texture image, which should be a power of 2. Recall that a wavelet is a convolution of coefficients $(c_i)$ with $P$ matrix. Having the problem in this form causes the problem to be very difficult in determining the coefficients. By using the $P$ matrix in Equation 8 allows the reverse order of $P$ with $W$. 

346
subject to the orthogonality constraints

\[ C_j(c) = \sum_{k=0}^{n-1} c_k c_{k-2j} - \delta_{0j} = 0 \quad j = 1 \ldots n/2, \]

where \( C_m \) is the \( m \)th element of the function vector \( C \). Adjoining the constraint to the cost function yields:

\[ J = c^T P_2^T P_2 c + \lambda^T C(c). \quad (9) \]

A necessary condition to minimize Equation 9 is that

\[
\begin{bmatrix}
\frac{\partial J}{\partial c} \\
\frac{\partial J}{\partial \lambda}
\end{bmatrix} = 0.
\]

where \( \lambda \) is the Lagrange multiplier. Explicitly, this turns out to be the following.

\[
\frac{\partial J}{\partial \lambda_k} = \sum_{i=0}^{n-1} c_k c_{i+2k} - \delta_{0k} \quad (10)
\]

\[
\frac{\partial J}{\partial c_k} = 2 \sum_{j=0}^{n-1} P_{kj} c_j + 2 \sum_{j=0}^{n/2-1} \lambda_j \left(c_{k+2j} + c_{k-2j}\right) \quad (11)
\]

### 2.3 Solution Methodology

Equation (9) represents a quadratic cost function with \( n/2 \) non-linear and non-convex constraints. To solve for the coefficients, a hybridized form of a modified float genetic algorithm (GA) was used. Genetic algorithms (GAs) are a powerful set of stochastic global search techniques that have been shown to produce very good results for a wide class of problems. GAs can find good solutions to nonlinear problems by simultaneously exploring multiple regions of the solution space and exponentially exploiting promising areas through mutation, crossover and selection operations [10]. In general, the fittest individuals of any population are more likely to reproduce and survive to the next generation, therefore improving successive generations. However, some of the inferior individuals can, by chance, survive and also reproduce. Unlike many other optimization techniques, GAs do not make strong assumptions about the form of the objective function [10]. Whereas traditional search techniques use characteristics of the problem (objective function) to determine the next sampling point (e.g., gradients, Hessians, linearity, and continuity), the next sampled points in genetic algorithms are determined based on stochastic sampling/decision rules, rather than a set of deterministic decision rules. Therefore, evaluation functions of many forms can be used, subject to the minimal requirement that the function can map the population into a totally ordered set. A more complete discussion of GAs, including extensions to the general algorithm and related topics, can be found in Michalewicz [10].

Local improvement procedures (LIPs), e.g., two-opt switching for combinatorial problems and gradient descent for unconstrained nonlinear problems, quickly find the local optimum of a small region of the search space, but are typically poor global searchers. Because these procedures do not guarantee optimality, in practice, several random starting points are generated and used as input into the local search technique and the best solution is recorded. This global optimization technique (multistart) was also used for this problem [6], but it is a blind search technique since it does not take account past information [5, 13]. GAs, unlike multistart, utilize past information in the search process. Therefore, LIPs have been incorporated into GAs in order to improve their performance through what could be termed “learning.” Such hybrid GAs have been used successfully to solve a wide variety of problems [1, 5, 4, 8, 10, 12]. Houck, Joines, and Kay [5] showed that for the continuous location-allocation problem, a GA that incorporated a LIP outperformed multistart and a two-way switching procedure, where both methods utilized the same LIP as the hybrid GA.
The GA used in these experiments is of a form advocated by Michalewicz [10] and uses a floating point (real-valued) representation with three real-valued crossovers (simple, heuristic, and arithmetic) and five real-valued mutations (boundary, uniform, multi-uniform, non-uniform, and multi-non-uniform). A normalized geometric ranking scheme is used as the selection procedure [4]. The GA uses a LIP (the nonlinear optimization technique (fmincon) that is part of the Matlab Optimization toolbox) for its evaluation function. In general, GAs do not include contraints beyond variable bounds. Since this problem contains several nonlinear constraints, a dynamic Bean’s penalty method is used to penalize infeasible points returned from the LIP. The methodology was implemented in Matlab using the Genetic Algorithm Optimization Toolbox (GAOT) developed by Joines et al. [7]. The GA algorithm (see Figure 1) that was used is quite simple with the exception of the penalty term that is updated in Step 4 and the evaluation function in Figure 2.

1. Set generation counter \( i \leftarrow 0 \).

2. Create the initial population, \( P_0 \), by randomly generating \( N \) individuals.

3. Determine the fitness of each individual in the population by applying the objective function to the individual and recording the value found.

4. Increment to the next generation, \( i \leftarrow i + 1 \) and determine whether to update the dynamic penalty term(s) \( (\beta) \).

5. Create the new population, \( P_i \), by selecting \( N \) individuals stochastically based on the fitness from the previous population, \( P_{i-1} \).
   
   (a) Randomly select \( R \) parents from the new population to form the new children by application of the genetic operators.
   
   (b) Evaluate the fitness of the newly formed children by applying the objective function.

6. If \( i < \) the maximum number of generations to be considered, go to Step 4.

7. Print out the best solution found.

Figure 1: A Simple Genetic Algorithm

The evaluation function in Figure 2 uses a Lamarckian learning scheme [5, 4, 8] since the individual is updated to represent the point that was found by the gradient descent method as seen in Steps 3a and 4a. Also, we limit the number of iterations that the local search will perform to find the right balance of local exploitation versus global exploration. If the point that is returned by the local search is infeasible (i.e., the local search does not converge to a feasible point), then we have to calculate the penalized fitness value. Bean’s dynamic penalty method allows the algorithm to dance along the feasible domain. The penalty term is increased if the best point has been infeasible for the last \( k \) generations putting more pressure on driving the solutions to the feasible region. If the best solution has been feasible over the past \( k \) generations, then the penalty term is decreased allowing it to move toward the boundary. Otherwise, the penalty term remains the same.

3 Experimental Results

Several textures (both synthetic and real), were tested with the hybridized GA method described above. The first image is shown in Figure 3(a). The repeat for this simple image is two black pixes and 2 white pixles in a plain weave pattern. As the image repeated on 4 pixels, four wavelet coefficients were used. Figure 3(b) shows the result of filtering the im-
age through a set of optimized coefficients. Note that for this simple image, the filter was able to completely filter out the texture in the HH, LH, and HL quadrants, and showed a smoothed version of the image in the LL quadrant. As a comparison, using Daubechies coefficients does not perform as well in filtering out the high frequency components of the texture as is shown in Figure 3(c). In Figure 4(a), a horizontal defect was introduced into the image, and it’s resulting transformed image using the same coefficients is shown in Figure 4(b). Note that the horizontal defect only shows up in the lower left quadrant (LH), whereas vertical defects would show up in the upper right quadrant as expected.

What is of greater interest is the effect of shifting of the tested image from that of the trained image. It is known that these types of wavelet filters, or quadrature-mirror filters (QMF) are not shift invariant. Figure 4(c) shows the result of shifting the image down by one pixel. There is a rather dramatic change in the output of the filtered image, if one compares Figure 3(b) with Figure 4(c). The information that was in the upper left quadrant now appears in the lower left quadrant. In all our studies on shifting, there was no bleeding of information among the quadrants, but rather a moving of all the information for these highly optimized filters which is quite promising once the filters are combined with data fusion techniques.

Figures 5(a) and 5(b) show a GA wavelet of a real fabric. The filter does a pretty good job at filtering out the texture in the HH quadrant, as it is designed to do, but still lets a fair amount of texture information pass into the LH and HL quadrants. Using the method described in this paper, it should be possible to design different filters for the LH and HL quadrants (lower left and upper right) which look for line defects, or cross-correlate these quadrants with the trained image and subtract out the texture.

4 Conclusions

A method was presented that locates defects in a repeating texture by use of wavelets. The problem of finding the wavelet coefficients was couched as a optimization problem with non-convex constraints. This problem was solved using a hybridized genetic algorithm to find a global minima to the cost function, which represented the amount of energy not filtered by the high-pass filter portion of the wavelet filter. For the images shown, the hybrid GA was quite effective at solving the problem as compared to the multistart procedure.

References


Figure 3: GA Wavelet Transformation versus the Daubechie Wavelet Transformation

Figure 4: 2x2 Plain Weave with Defects as well as Shift.

Figure 5: GA Wavelet Transformation of real Fabric with and without Defects