

COMPUTATIONALLY EFFICIENT WAVELET AFFINE INVARIANT FUNCTIONS FOR SHAPE RECOGNITION

Erdem Bala,

Dept. of Electrical and Computer Engineering,
University of Delaware, 140 Evans Hall, Newark, DE, 19716

A. Enis Cetin,

Dept. of Electrical and Electronics Engineering,
Bilkent University, Ankara, 06533, Turkey

ABSTRACT

An affine invariant function for object recognition is constructed from wavelet coefficients of the object boundary. In previous works, undecimated wavelet transform was used for affine invariant functions. In this paper, an algorithm based on decimated wavelet transform is developed to compute the same affine invariant functions. As a result computational complexity is reduced without decreasing recognition performance. Experimental results are presented.

1 INTRODUCTION

Object recognition is an important problem in computer vision and pattern analysis [1-6]. In this paper, recognition of objects from their boundaries that are subject to affine transformations is considered. The affine transformation includes rotation, scaling, skewing and translation. It preserves parallel lines and equispaced points along a line. In some cases, the affine transformation can also be used to approximate the perspective transformation [1].

Several features that are linear under an affine transformation were developed in the literature. The most commonly used ones are affine arc length [7], affine invariant Fourier descriptors [2], and moment invariants [3]. Recently, dyadic wavelet transform was also used to develop several affine invariant functions [5,10]. These functions are constructed from undecimated wavelet coefficients which are produced after computing the wavelet transform of a curve corresponding to the boundary of the object. In this paper, an algorithm based on decimated wavelet transform is developed to compute the affine invariant functions proposed in [5]. This leads to a computationally efficient object recognition scheme.

The paper is organized as follows: In Section 2, some background information on affine invariant functions is presented. In Section 3, the computationally efficient algorithm is presented. In Section 4, experimental results are presented. In addition, a new object recognition scheme based on linear combination of affine invariant functions constructed from multiple resolution wavelet coefficients is presented. It is observed that recognition performance is comparable to other wavelet based schemes.

2 BACKGROUND

Consider a parametric curve $\{x(t), y(t)\}$ with parameter t on a plane. A point on the curve under an affine transformation becomes

$$\tilde{x}(t) = a_0 + a_1x(t) + a_2y(t) \tag{1}$$

and

$$\tilde{y}(t) = b_0 + b_1x(t) + b_2y(t) \tag{2}$$

Equations (1) and (2) can be rewritten in matrix form as follows:

$$\begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} = A \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + B \quad (3)$$

where the nonsingular matrix A represents the scaling, rotating, and skewing transformation and the vector B corresponds to the translation. Jacobean, J , of the transformation is $J = a_1 b_2 - a_2 b_1 = \det(A)$.

Let $I(t)$ be an affine invariant function and $\tilde{I}(t)$ be the same invariant function calculated using the points that are subject to the affine transformation. The relation between the two invariant functions can be formulated as:

$$\tilde{I} = IJ^\omega \quad (4)$$

The exponent ω is called the weight of the invariance. If $\omega = 0$, then I is called an absolute invariant, else it is called a relative invariant.

3 AFFINE INVARIANT FUNCTIONS USING DECIMATED WAVELET COEFFICIENTS

Wavelet transform was used to recognize planar objects under the similarity transformation in [8, 9]. Affine invariant functions using the dyadic wavelet transform was derived by Tieng and Boles [10] and Khalil and Bayoumi [5]. The main difference between [10] and [5] is that, in [10] two dyadic levels were used, whereas in [5], a

wavelet-based conic equation was introduced. This leads to an affine invariant function of six or more dyadic levels.

Discrete dyadic wavelet transform (DWT) of a signal is implemented using halfband lowpass and highpass filters forming a filterbank together with downsamplers [11]. The filterbank produces two sets of coefficients: orthogonal detail (or wavelet) coefficients which are the even outputs of the highpass filter, and the approximation coefficients which are the even outputs of the lowpass filter. Samples with odd indices are dropped by the downsamplers in decimated implementation. Due to downsampling computational cost of implementing DWT drops to $O(N \log N)$ (even to $O(N)$ for some wavelets).

Let us denote the wavelet transform of the signal $x(t)$ at the resolution level (or scale) i as $W_i x(t)$, then the wavelet transform of (1) and (2) will be

$$W_i \tilde{x}(t) = a_1 W_i x(t) + a_2 W_i y(t) \quad (5)$$

$$W_i \tilde{y}(t) = b_1 W_i x(t) + b_2 W_i y(t) \quad (6)$$

Note that $W_i a_0 = W_i b_0 = 0$. because of the highpass filter.

Let the signal pair $x(t)$ and $y(t)$ represent the boundary of an object. An affine invariant function for an object using the wavelet coefficients of signals $x(t)$ and $y(t)$ for two i, j ($i \neq j$) can be defined as

$$f_{ij}(t) = W_i x(t) W_j y(t) - W_i y(t) W_j x(t) \quad (7)$$

It can be easily shown that

$$\tilde{f}_{ij}(t) = W_i \tilde{x}(t) W_j \tilde{y}(t) - W_i \tilde{y}(t) W_j \tilde{x}(t) = \det(A) f_{ij}(t) \quad (8)$$

This invariant function $f_{ij}(t)$ defined in [5] uses only the detail coefficients calculated at two different levels. In [10] another affine invariant function using both the detail and approximation coefficients of the same dyadic level is defined. In [5] Equation (7) is also used to construct a wavelet-based conic equation leading to an affine invariant function based on six dyadic levels.

All of the invariant functions defined in [5, 10] are computed using the undecimated implementation of the wavelet transform (WT) which does not use downsampling operation after filtering. This dramatically increases the computational cost of the wavelet transform. If the length of the original signal is N , then for the undecimated wavelet transform, length- N signals are filtered at each level. However, in the decimated implementation of the wavelet transform, the signal length is halved due to downsampling operation performed after each filtering step. In this paper, we develop an algorithm to compute the affine invariant function defined in (7) using the orthogonal decimated wavelet transform scheme. The wavelet signal $W_i x(t)$, at resolution scale $i = 1$ can be expressed as

$$W_i x(t) = \sum d_k w(t - k), \quad i = 1 \quad (9)$$

where d_k are the wavelet coefficients computed using a decimated filterbank at resolution scale $i=1$ and $w(t)$ is the so-called mother wavelet. If the length of the data is N ($N=512$ is chosen in this paper) then the limits of summation in (9) go from $k=0$ to $k=N$ assuming a circular computation of the WT. Similarly, $W_j y(t)$ can be expressed for $j=2$ as follows

$$W_j y(t) = \sum e_l w(t/2 - l) \quad (10)$$

where e_l are the wavelet coefficients at resolution scale $j=2$. In this case the limits of the summation go from $l=0$ to $l=N/2$ due to downsampling. Let us assume that $w(t)$ is the Haar wavelet, i.e.,

$$w(t) = 1 \text{ for } 0 < t < 0.5, \quad w(t) = -1 \text{ for } 0.5 < t < 1, \quad w(t) = 0, \text{ otherwise} \quad (11)$$

The first term of (7) can be expressed as

$$W_i x(t) W_j y(t) = \sum \sum d_k e_l w(t-k) w(t/2 - l) \quad \text{for } i=1, j=2 \quad (12)$$

Direct computation of (12) and the affine invariant function defined in (7) requires $N \times N/2$ and $N \times N$ multiplications, respectively. However, notice that $w(t)w(t/2) = w(t)$, $w(t)w(t/2 - k) = 0$, for $k > 1$, since the Haar wavelet has a compact support with length 2. Similarly, $w(t-2)w(t/2 - 1) = w(t-2)$

$w(t-3)w(t/2-1) = -w(t-3)$, etc. By taking advantage of these relations the double sum in (12) can be reduced to a single summation as follows:

$$W_i x(t) W_j y(t) = \sum_{k=0, \text{even}}^N d_k e_{k/2} w(t-k) - \sum_{k=1, \text{odd}}^N d_k e_{(k-1)/2} w(t-k), \quad \text{for } i=1, j=2 \quad (13)$$

Computation of the right hand side of (13) requires only N multiplications. The affine invariant function, $f_{ij}(t)$ for $i = j + 1$, can be expressed as

$$f_{ij}(t) = \sum_{k, \text{even}} d_k^i e_{k/2}^{i+1} w_i(t-k) - \sum_{k, \text{odd}} d_k^i e_{(k-1)/2}^{i+1} w_i(t-k) + \sum_{k, \text{even}} e_k^i d_{k/2}^{i+1} w_i(t-k) - \sum_{k, \text{odd}} e_k^i d_{(k-1)/2}^{i+1} w_i(t-k) \quad (14)$$

where $w_i(t) = w(t/2^i)$ is the wavelet of the resolution scale i , d_k^i , and e_k^i are the wavelet coefficients of the signals x and y at resolution level i , respectively. An important feature of this equation is that it can be computed using the computationally efficient orthogonal wavelet transform as the wavelet coefficients d_k^i , and e_k^i can be computed using a filterbank having downsamplers. Equations (13) and (14) are developed for the specific case of $i = 1, j = i + 1$. However similar equations with $O(N)$ complexity can be easily developed to any i, j values because $w(t)w(t/2^j) = w(t), \dots, w(t-j)w(t/2^j) = -w(t-j)$, and 0, otherwise etc due to the fact that $w(t)$ has a compact support. Since all the affine invariant functions developed in [5] are based on $f_{ij}(t)$ they can be computed using decimated wavelet transform. As a result significant amount of computational savings can be achieved. In the undecimated WT implementation, length- N signals are filtered at each level whereas in decimated

implementation length - $N/2^i$ signals are filtered at resolution level i and the final stage of constructing $f_{ij}(t)$ requires only $O(N)$ arithmetic.

Although the decimated wavelet coefficients are translation variant Equation (14) is translation invariant as the continuous-time function $f_{ij}(t)$ can be computed for all t values using the right hand side of (14). In practice $f_{ij}(t)$ is computed for uniformly spaced $N = 512$ points $t = 0, 1, \dots, 511$ in [10] and in this paper.

Equation (14) is obtained by taking advantage of the fact Haar wavelet has compact support. Some computationally efficient signal reconstruction algorithms from WT also take advantage of this fact [12]. In fact, all wavelets constructed from FIR filters have compact support. Therefore the double summation in (7) can be reduced to a set of single summations as in (13) for all compactly supported wavelets and equations similar to (14) can be obtained as well. For example, widely used Daubechies-4 wavelet has a compact support of length 6, i.e., $w(t) = 0$, for $t > 6$, and $t < 0$. In the case of Daubechies-4 wavelet $w(t)w(t/2 - k) = 0$, for $k > 3$. This leads to a slightly higher computational cost than Haar wavelet but longer wavelets are more robust to noise compared to Haar wavelet. In general the length of data N (e.g., $N=512$) is much higher than the support length of most wavelets. Therefore computational savings are significant.

4 EXPERIMENTAL RESULTS

Since a computationally efficient algorithm is developed in the previous section for the affine invariant functions developed in [5] it is natural that we get the same simulation results. In [5] simulation results are obtained by using a conic equation based affine invariant function using six dyadic resolution levels. In addition, we also present a new

practical object recognition scheme using multiple resolution wavelet coefficients in this section.

In this scheme, k invariant functions $f_{ij}(t)$ for a given test object are calculated by using consecutive pairs of resolution levels $(i_1, i_{1+1}), (i_2, i_{2+1}), \dots, (i_k, i_{k+1})$. Corresponding k invariant functions for each model object are kept in a database. The correlations between the k invariant functions of the test object and each model object are calculated to get correlation values R_1, R_2, \dots, R_k , which are defined as

$$R(I_1(t), I_2(t)) = \frac{\int I_1(t) I_2(t) dt}{\|I_1\| \|I_2\|} \quad (15)$$

where $I_1(t)$ and $I_2(t)$ represent the invariant functions. The final decision function between the test object and any model object is found by linearly combining the k correlation values as follows:

$$R_{final} = \nu_1 R_1 + \nu_2 R_2 + \dots + \nu_k R_k \quad (16)$$

where $\nu_1 + \nu_2 + \dots + \nu_k = 1$. As a rule of thumb more weight should be given to resolution levels containing more signal energy to obtain robustness against noise. This approach gives us also the flexibility of sampling $f_{ij}(t)$ in a nonuniform manner, i.e., at the resolution level pair (i_1, i_{1+1}) , $f_{(i_1, i_{1+1})}(t)$ can be computed at $N=512$ points but at the next resolution level pair (i_2, i_{2+1}) , $f_{(i_2, i_{2+1})}(t)$ can be computed at $N=256$

points etc. to achieve computational savings in computing the correlation functions defined in (15).

The experiments to test the effectiveness of the proposed object recognition method are carried out with airplane images that were also used in [5]. The same type of wavelet used in [5] are used in the experiments. There are 20 model images in the database. 10 test images are constructed by applying random affine transformations to randomly chosen 10 of the model images. The boundary signals of all the objects are normalized to length 512. The correlation values between the test image and the model images are calculated and the result is determined according to the model producing the highest correlation value. The experiments are carried out with two different levels of uniformly distributed random noise which is added to the boundaries of the test images. The signal to noise ratio (SNR) is defined as in [5]. In the first set of experiments the SNR is about 50 dB, and in the second set of experiments the SNR is about 20 dB. Table 1 gives the highest five correlation values for each test image with SNR 50 dB, and Table 2 gives the highest five correlation values for each test image with SNR 20 dB. In all experiments, correct recognition results are obtained as in [5]. In both cases of high and low noise power, the highest correlation value is produced with the model image from which the test image is constructed by applying a random affine transformation.

In all experiments summarized in Tables 1 and 2, resolution level pairs (4,5), (5,6) and (6,7) are used to calculate the invariant functions $f_{ij}(t)$ and the corresponding weights are chosen as $\nu_1 = 0.4, \nu_2 = 0.3, \nu_3 = 0.3$, respectively. In these experiments, low and high noise levels are used, and the recognition success rate is 100%. In another set of experiments the correlations of affine invariant functions of resolution level pairs (2,3),

(3,4), (4,5), (5,6) and (6,7) are linearly combined with weights 0.2, 0.3, 0.3, 0.1, and 0.1, respectively. Perfect recognition results are achieved at low noise case. However at high noise power, it turns out that the wavelet coefficients of levels 2 and 3 are very noisy and objects are not recognized correctly.

5 CONCLUSION

The problem of 2D object recognition using affine invariant functions is considered. In previous works, undecimated wavelet transform was used for constructing affine invariant functions. In this paper, an algorithm based on decimated wavelet transform is developed to compute the same affine invariant functions. As a result computational complexity is reduced without decreasing recognition performance. It is experimentally shown that the invariant function detects the affine transformed objects with high accuracy.

ACKNOWLEDGMENTS

We would like to thank Dr. Mahmoud I. Khalil who provided the set of plane images used in this paper.

REFERENCES

- [1] J.L. Mundy and A. Zisserman, Geometric Invariance in Computer Vision. Cambridge, Mass.: MIT Press, 1992.
- [2] K. Arbter, W.E. Synder, H. Burkhardt, and G. Hirzinger, "Application of Affine-Invariant Fourier Descriptors to Recognition 3-D Objects," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 7, pp. 640-647, July 1990.
- [3] T.H. Reiss, "The Revised Fundamental Theorem of Moment Invariants", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 13, no. 8, pp. 830-834, Aug. 1991.

- [4] H. Freeman, "Shape Description via the Use of Critical Points", *Pattern Recognition*, vol. 10, no. 3, pp. 159-166, 1978.
- [5] M. I. Khalil and M. M. Bayoumi, "A Dyadic Wavelet Affine Invariant Function for 2D Shape Recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1152-1164, Oct. 2001.
- [6] I. Weiss, "Geometric Invariants and Object Recognition", *Int'l J. Computer Vision*, vol. 10, no. 3, pp. 207-231, 1993.
- [7] H.W. Guggenheimer, *Differential Geometry*. New York: McGraw Hill, 1963.
- [8] Q.M. Tieng and W.W. Boles, "Recognition of 2D Object Contours Using the Wavelet Transform Zero-Crossing Representation", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 8, pp. 910-916, Aug. 1997.
- [9] M. Khalil and M. Bayoumi, "Invariant 2D Object Recognition Using the Wavelet Modulus Maxima", *Pattern Recognition Letters*, vol. 21, no. 9, pp. 863-872, 2000.
- [10] Q.M. Tieng and W.W. Boles, "Wavelet-Based Affine Invariant Representation: A Tool for Recognizing Planar Objects in 3D Space", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 8, pp. 846-857, Aug. 1997.
- [11] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693, July 1989.
- [12] A. E. Cetin, R. Ansari, 'Signal recovery from wavelet transform maxima,' *IEEE Trans. Signal Processing*, vol. 42, No.1, pp. 194-196, Jan. 1994.

Table 1
The Best Five Matches Between the Test Images and the Model Images
for Small Noise Level

Test Image	1.	2.	3.	4.	5.
1	0.9997	0.9225	0.8837	0.7937	0.6210
2	0.9986	0.6248	0.5005	0.4521	0.4304
3	0.9996	0.7174	0.6474	0.5942	0.3490
4	0.9998	0.9185	0.8219	0.6925	0.6255
5	0.9996	0.6564	0.5654	0.5276	0.4760
6	0.9995	0.6432	0.4841	0.4209	0.3631
7	0.9992	0.6437	0.5695	0.5087	0.4406
8	0.9994	0.7603	0.6855	0.6197	0.5014
9	0.9976	0.5770	0.4906	0.3837	0.3126
10	0.9985	0.6445	0.5886	0.5655	0.4765

Table 2
The Best Five Matches Between the Test Images and the Model Images
for High Noise Level

Test Image	1.	2.	3.	4.	5.
1	0.9617	0.8870	0.8407	0.7635	0.5971
2	0.8602	0.5658	0.4822	0.4378	0.3891
3	0.9534	0.6801	0.6109	0.5557	0.3504
4	0.9750	0.8935	0.8010	0.6790	0.6034
5	0.9512	0.6459	0.5484	0.5107	0.4628
6	0.9376	0.6265	0.4883	0.4008	0.3642
7	0.9089	0.6144	0.5457	0.4818	0.3965
8	0.9148	0.6858	0.6280	0.5524	0.4859
9	0.7869	0.5612	0.4625	0.4074	0.3401
10	0.8552	0.6181	0.5478	0.4986	0.4324