

THE FAST CORRENTROPY MACE FILTER

Kyu-Hwa Jeong, Seungju Han, Jose C. Principe

Computational NeuroEngineering Laboratory
Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32611 U.S.A.
E-mail: {khjeong, han, principe}@cnel.ufl.edu

ABSTRACT

In this paper, we implement the newly introduced correntropy MACE filter using the Fast Gauss Transform (FGT). The Correntropy MACE filter is a nonlinear extension to the MACE filter using the correntropy function in a feature space nonlinearly related to the input. The correntropy MACE outperforms the traditional linear MACE in both generalization and rejection abilities. However, in practice, the drawback of the correntropy MACE filter is its computation complexity. This paper presents a fast version of the correntropy MACE by using the FGT idea and validates the approximation with results in Synthetic Aperture Radar (SAR) image recognition.

Index Terms— Correntropy MACE, Fast Gauss Transform (FGT), Synthetic Aperture Radar (SAR) image recognition.

1. INTRODUCTION

Cross-correlating an input data with a synthesized template is a central concept to determine whether an object of interest is present or not. The advantages of correlation filter are simplicity and shift invariance property [1][2]. The best known correlation filter is the minimum average correlation energy (MACE) filter [3]. The MACE minimizes the average correlation energy of the output over the training images subject to a peak constraint in the center of the space. The MACE filter is generally known to be sensitive to distortions and has been shown to have poor generalization properties, that is, images in the recognition class but not in the training data set are not recognized well. In order to overcome the problem of the MACE, there have been several approaches for advanced correlation filters[4]. The Correntropy MACE filter, recently proposed in [5] is a new approach to utilize higher order moment information of the image structure in a new reproducing kernel Hilbert space (VRKHS) induced by correntropy [6]. Correntropy is a positive definite function, which measures a nonlinear similarity between random variables (or stochastic

processes) produced by the inclusion of the high-order statistics of the input images. The correntropy MACE filter outperforms the traditional linear MACE in both generalization and rejection abilities in object recognition. In practice, the drawback of the proposed correntropy MACE filter is its computation complexity. The output of the correntropy MACE filter is obtained by computing the product of two matrices, whose computation time depends on the image size and the number of training images. Each element involves a double summation of weighted kernel functions. Therefore, each element of the matrix requires $O(d^2)$ computations, where d is the number of pixels for one image. When the number of training images is N , the total computation complexity of one test output is $O(d^2 N(N + 1))$. This quickly becomes too complex in practical settings.

This paper presents a way to reduce the computation by utilizing clustering and a Hermite expansion of the Gaussian function which has been called the Fast Gauss Transform (FGT) [7]. In the correntropy MACE case, we cannot use the FGT directly, however, this paper shows how the complexity can be reduced to $O(pk d N(N + 1))$, where p is the order of the Hermite approximation and k is the number of clusters utilized in FGT.

The organization of the paper is as follows. We present the correntropy MACE filter in section 2. In section 3, we review the FGT briefly and modify the basic algorithm to the Fast correntropy MACE. In section 4, we present simulation results for SAR image recognition using the MSTAR public release data [8] and section 5 summarizes and points out some further research.

2. THE CORRENTROPY MACE FILTER

Correntropy is a generalized similarity measure between two arbitrary random variables X and Y defined by

$$V(X, Y) = E[\langle \Phi(X), \Phi(Y) \rangle] = E[k(X, Y)], \quad (1)$$

where E is the mathematical expectation and k is a kernel function that obeys the Mercer's conditions. In this paper, we

This work was partially supported by NSF grant ECS-0601271 and Lockheed Martin Corporation.

use the Gaussian kernel, which is the most widely used Mercer kernel. Correntropy is a positive function therefore it induces a new reproducing kernel Hilbert space called VRKHS. Given data samples $\{x_i\}_{i=1}^d$, the correntropy kernel creates another data set $\{f(x_i)\}_{i=1}^d$ preserving the similarity measure as [9]

$$V(i, j) = E[k(x_i - x_j)] = E[f(x_i)f(x_j)]. \quad (2)$$

According to (2), there exists a nonlinear mapping f which makes the correntropy of x_i the correlation of $f(x_i)$. As the conventional MACE filter is derived by controlling correlation output, the correntropy MACE is formulated by minimizing average correntropy energy in the d dimensional VRKHS.

Let the i th image vector be $\mathbf{x}^i = [x_1^i \ x_2^i \ \cdots \ x_d^i]^T$ and the filter be $h = [h_1 \ h_2 \ \cdots \ h_d]^T$, where T denotes transpose. We denote the transformed training image matrix and filter vector whose size are $d \times N$ and $d \times 1$, respectively, be

$$\mathbf{F}_X = [\mathbf{f}_{x_1}, \mathbf{f}_{x_2}, \cdots, \mathbf{f}_{x_N}], \quad (3)$$

$$\mathbf{f}_h = [f(h_1) \ f(h_2) \ \cdots \ f(h_d)]^T, \quad (4)$$

where $\mathbf{f}_{x_i} = [f(x_1^i) \ f(x_2^i) \ \cdots \ f(x_d^i)]^T$ for $i = 1, \cdots, N$. Given data samples, we can estimate the cross correntropy between i th training image vector and the filter as

$$v_{oi}[m] = \frac{1}{d} \sum_{n=1}^d f(h_n) f(x_{n-m}^i), \quad (5)$$

for all the lags $m = -d + 1, \cdots, d - 1$. Then the correntropy energy of the i th image is given by

$$E_i = \mathbf{v}_{oi}^T \mathbf{v}_{oi} = \mathbf{f}_h^T \mathbf{V}_i \mathbf{f}_h, \quad (6)$$

where, \mathbf{V}_i is the $d \times d$ correntropy matrix as

$$\mathbf{V}_i = \begin{pmatrix} v_i(0) & v_i(1) & \cdots & v_i(d-1) \\ v_i(1) & v_i(0) & \cdots & v_i(d-2) \\ \vdots & \vdots & \ddots & \vdots \\ v_i(d-1) & \cdots & v_i(1) & v_i(0) \end{pmatrix}. \quad (7)$$

Each element of the correntropy matrix is computed without explicitly knowing the mapping function f by

$$v_i(l) = \sum_{n=1}^d k(x_n^i - x_{n+l}^i), \quad (8)$$

for $l = 0, \cdots, d - 1$. Since our objective is to minimize the average correntropy energy in feature space, we can formulate the optimization problem by

$$\min \mathbf{f}_h^T \mathbf{V}_X \mathbf{f}_h \quad \text{subject to } \mathbf{F}_X^T \mathbf{f}_h = \mathbf{c}, \quad (9)$$

where, $\mathbf{V}_X = \frac{1}{N} \sum_{i=1}^N \mathbf{V}_i$ and \mathbf{c} is the desired vector for all the training images. Since the correntropy matrix \mathbf{V}_X is positive definite, there exists an analytic solution to the optimization problem in the new finite dimensional VRKHS. Then the correntropy MACE filter in feature space becomes

$$\mathbf{f}_h = \mathbf{V}_X^{-1} \mathbf{F}_X (\mathbf{F}_X^T \mathbf{V}_X^{-1} \mathbf{F}_X)^{-1} \mathbf{c}. \quad (10)$$

In order to test this filter, let \mathbf{Z} be the matrix of L vector testing images and \mathbf{F}_Z be the transformed matrix of \mathbf{Z} , then the $L \times 1$ output vector is given by

$$\mathbf{y} = \mathbf{K}_{ZX} (\mathbf{K}_{XX})^{-1} \mathbf{c}, \quad (11)$$

where $\mathbf{K}_{ZX} = \mathbf{F}_Z^T \mathbf{V}_X^{-1} \mathbf{F}_X$ and $\mathbf{K}_{XX} = (\mathbf{F}_X^T \mathbf{V}_X^{-1} \mathbf{F}_X)^{-1}$. \mathbf{K}_{XX} is $N \times N$ symmetric matrix and \mathbf{K}_{ZX} is $L \times N$ matrix, whose (i, j) th element is computed by

$$\begin{aligned} (\mathbf{K}_{XX})_{ij} &= \sum_{l=1}^d \sum_{k=1}^d w_{lk} f(x_l^i) f(x_k^j) \\ &\cong \sum_{l=1}^d \sum_{k=1}^d w_{lk} k(x_l^i - x_k^j), \end{aligned} \quad (12)$$

$$\begin{aligned} (\mathbf{K}_{ZX})_{ij} &= \sum_{l=1}^d \sum_{k=1}^d w_{lk} f(z_l^i) f(x_k^j) \\ &\cong \sum_{l=1}^d \sum_{k=1}^d w_{lk} k(z_l^i - x_k^j), \end{aligned} \quad (13)$$

where w_{lk} is the (l, k) th element of \mathbf{V}_X^{-1} .

The final output expressions in (12) and (13) are obtained by approximating $f(x_l)f(x_k)$ and $f(z_l)f(x_k)$ by $k(x_l - x_k)$ and $k(z_l - x_k)$, respectively, which holds good on an average because of (2). Hence, we do not need to find the transformation $f(\cdot)$ as expected by the "kernel trick". Applying an appropriate threshold to the output of (11), one can detect and recognize the testing data without generating the composite filter in feature space.

3. IMPLEMENTATION USING THE FAST GAUSS TRANSFORM

3.1. The Fast Gauss Transform

In many problems in mathematics and engineering, the function of interest can be decomposed into sums of pairwise interactions among a set of sources. In particular, this type of problem is found in nonparametric probability density estimation as

$$G(z) = \sum_{j=1}^d q_j k(z - x_j), \quad (14)$$

where k is a kernel function centered at the source points x_j and q_j are scalar weighting coefficients. With the Gaussian kernel, (14) can be interpreted as a "Gaussian" potential field due to sources of strengths q_j at the points x_j , evaluated at the target point z . Suppose that we have M evaluation target points, then the computation of (14) requires $O(dM)$ calculations, which constrains the computation bandwidth for large data sets d and M in real world applications. The Fast Gauss Transform (FGT) can reduce the complexity to $O(d+M)$ for

(14). The FGT is one of a class of very interesting and important families of fast evaluation algorithms that have been developed over the past decades to enable rapid calculation of approximations at arbitrary accuracy. The basic idea is to cluster the sources and target points using appropriate data structures and the Hermite expansion, and then reduce the number of summations with a given level of precision.

3.2. The Fast Correntropy MACE

The major part of the computation burden in the correntropy MACE filter is given by

$$\mathbf{K} = \sum_{i=1}^d \sum_{j=1}^d w_{ij} e^{-(z_i - x_j)^2 / 2\sigma^2}. \quad (15)$$

This is very similar to the density estimation problem that evaluates at d targets z_i with given d source samples x_j . However, the weighting factor w_{ij} in (15) are dependent on both target and source, which is different from the original FGT applications, where the weights $\mathbf{q} = [q_1, \dots, q_d]^T$ is always the same at every evaluation target points. In our case, the weight vector $\mathbf{w}_i = [w_{i1}, \dots, w_{id}]^T$ is varying on every evaluation point z_i . We can say that (15) is a more general expression than the FGT that can be written as

$$\mathbf{K} = \sum_{i=1}^d G_i(z), \quad (16)$$

where

$$G_i(z) = \sum_{j=1}^d w_{ij} k(z_i - x_j). \quad (17)$$

This means that clustering and the Hermite expansion should be performed at every target z_i with a different weight vector \mathbf{w}_i , which causes an extra computation for clustering. However, since the sources are clustered in the FGT, if one expresses the clustered sources about its center into the Hermite expansion, then there is no need to do clustering and the Hermite expansion at every evaluation. The only thing that is necessary is to use different weight vectors at every evaluation point. This process does not require additional complexity compare to the original FGT formulation except that more storage is required to keep the weight vectors. By using the Hermite expansion around the target s , the Gaussian centered at x_j evaluated at z_i can be obtained by

$$\exp\left\{-\frac{(z_i - x_j)^2}{2\sigma^2}\right\} = \sum_{n=0}^{p-1} \frac{1}{n!} \left(\frac{x_j - s}{\sqrt{2}\sigma}\right)^n h_n\left(\frac{z_i - s}{\sqrt{2}\sigma}\right) + \varepsilon(p), \quad (18)$$

where the Hermite function $h_n(x)$ is defined by

$$h_n(x) = (-1)^n \frac{d^n}{dx^n} (\exp(-x^2)). \quad (19)$$

Also, in this paper, we use a simple greedy algorithm for clustering [10], which computes a data partition with a maximum radius at most twice the optimum. This clustering method and the Hermite expansion with order p requires $O(pd)$. In the case of (12) and (13), since the number of sources and targets are the same, they can be interchanged, that is, the test image can be the source so that the clustering and Hermite expansion can be done only one time per test. Thus \mathbf{K} in (16) can be approximated by

$$\mathbf{K} \approx \sum_{i=1}^d \sum_B \sum_{n=0}^{p-1} \frac{1}{n!} h_n\left(\frac{x_i - s_B}{\sqrt{2}\sigma}\right) C_n(B), \quad (20)$$

where B represents a cluster with a center s_B and $C_n(B)$ is given by

$$C_n(B) = \sum_{z_j, w_{ij} \in B} w_{ij}^T \left(\frac{z_j - s_B}{\sqrt{2}\sigma}\right)^n. \quad (21)$$

From (20), we can see that evaluation at k expansions at all the target costs $O(pkd)$ work, so the total number of operations is $O(pd(k+1))$ per computation of each element in (12) and (13). The final aim is to obtain the output of the correntropy MACE filter with N training images and L test images. In order to compute the output of one test image, the original direct method requires $O(d^2N(N+1))$ operations and we can reduce the operation count to $O(pd(k+1)N(N+1))$ by applying this enhanced FGT. Typically p and k are much more small values than d and N in our application. Additionally, clustering with the test image is performed only one time per test which reduces the computation time even more.

4. SIMULATIONS

In this paper, we illustrate the computation speed improvement and the effect on accuracy of the fast correntropy MACE filter in SAR image recognition using the MSTAR (Moving and Stationary Target Acquisition and Recognition) data which is a standard dataset in the SAR ATR community, allowing researchers to fairly test and compare their ATR algorithms[11]. In this simulation, we selected the BTR60 (Armored personal carrier) as a target (true class) and the T62 (Tank) as a confuser (false class). The goal is to design a filter which will recognize the BTR60 without being confused by the T62. In this paper, we use as training images aspect angles of 0 to 179 degree at a 17 degree depression angle. Training images were selected at every odd index number from a total of 120 exemplar images for each vehicles (so the number of training images, N , is 60). Testing is conducted with all of 120 exemplar images for each vehicle. All the images are cropped to be centered with the size of 64×64 pixels ($d=4096$). In this simulation, computation time is clocked with MATLAB version 7.0 on a 2.8GHz Pentium 4 processor with 2GByte of RAM.

Table 1 shows the comparison of computation time for (12) and (13) between the original method (direct) and the

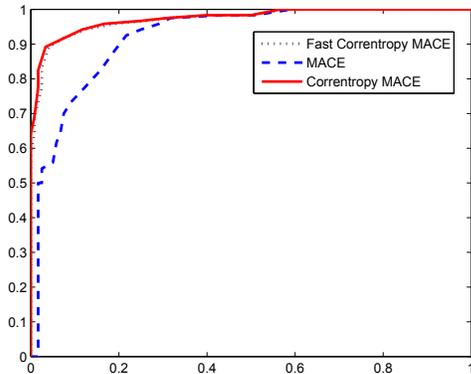


Fig. 1. Comparison of ROC curves

Table 1. Comparison of computation time and error for one test image between the direct method and the FGT method with $p = 4$ and $k = 4$

	Direct (sec)	FGT (sec)	Error
Train : \mathbf{K}_{XX}	7622.8	68.31	9.9668e-06
\mathbf{K}_{ZX}	122.8	1.15	8.7575e-06
Test(true)output			2.8225e-03
\mathbf{K}_{ZX}	128.6	1.18	3.8844e-05
Test(false)output			8.4377e-03

proposed fast method with a Hermite approximation order $p = 4$ and the number of cluster $k = 4$. The test time and absolute error were obtained by averaging 120 test images results for one test image. The FGT method is 100 times faster than the direct method with a reasonable error precision. Fig. 1 presents the comparison in terms of ROC curves of the MACE, the correntropy MACE and the fast correntropy MACE. Form the ROC curve we can observe that the approximation with $p = 4$ and $k = 4$ is very close to the original ROC curve and it also tells us that detecting ability of the proposed fast correntropy MACE filter is much better than that of the MACE. Table 2 shows the effect of different orders(p) and clusters(k). We can conclude that the computation time increases roughly proportional to p and k , while the absolute error linearly decreases.

Table 2. Comparison of computation time and error for one test image in the FGT method with a different number of orders and clusters

Order	Time (sec)	Error	Cluster	Time (sec)	Error
2	0.8116	1.48e-02	2	0.7181	5.61e-02
6	1.5140	8.23e-04	6	1.6693	3.87e-04
10	2.2119	8.58e-06	10	2.5595	4.71e-05
14	2.8533	4.16e-07	14	3.5660	6.93e-06
20	3.8097	1.25e-09	20	5.3067	1.14e-06

5. CONCLUSIONS

In this paper, we have implemented and evaluated the fast version of the correntropy MACE filter for object recognition using a modified Fast Gauss Transform (FGT). We presented experimental results for the SAR object recognition using the MSTAR public release data. Using the recently introduced correntropy idea, the conventional MACE filter can be implemented in a new RKHS that has the same dimension as the input data and it outperforms the conventional MACE. In practice, the drawback of the proposed correntropy MACE filter is the computation complexity due to the double summation of weighted pairwise interactions with Gaussian kernels that requires $O(d^2 N(N + 1))$ computation. With the generalized FGT, we can implement the fast correntropy MACE filter in only $O(pkdN(N + 1))$ for one test image. Simulation results shows that the fast correntropy MACE is much faster than the direct method and preserves the good performance.

6. REFERENCES

- [1] B.V.K. Vijaya Kumar, "Tutorial survey of composite filter designs for optical correlators," vol. 31, pp. 4773–4801, 1992.
- [2] B.V.K. Vijaya Kumar, M. Savvides, C. Xie, and K. Venkataramani, "Biometric verification with correlation filters," *Applied Optics*, vol. 43, no. 2, pp. 391–402, Jan 2004.
- [3] A. Mahalanobis, B.V.K. Vijaya Kumar, and D. Casasent, "Minimum average correlation energy filters," *Appl. Opt.*, vol. 26, no. 17, pp. 3633–3640, 1987.
- [4] B.V.K. Vijaya Kumar, "Recent advances in distortion-invariant correlation filter design," in *Proc. SPIE*, 1995, vol. 2490, pp. 2–13.
- [5] K.H. Jeong, P.P. Pokharel, J. Xu, S. Han, and J.C. Principe, "The correntropy mace filter for image recognition," in *Proc. IEEE Int. Workshop on Machine Learning for signal Processing (MLSP)*, France, July 2006, pp. 9–14.
- [6] I. Santamaría, P.P. Pokharel, and J.C. Principe, "Generalized correlation function: Definition, properties and application to blind equalization," *IEEE Trans. Signal Processing*, vol. 54, no. 6, pp. 2187–2197, June 2006.
- [7] L. Greengard and J. Strain, "The fast gauss transform," *SIAM J. Sci. Statist. Comput.*, vol. 12, no. 1, pp. 79–94, Jan. 1991.
- [8] "Moving and stationary target acquisition and recognition (MSTAR/IU) public mixed target dataset," .
- [9] P.P. Pokharel, J.W. Xu, D. Erdogmus, and J.C. Principe, "A closed form solution for a nonlinear wiener filter," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, France, May 2006, vol. 3, pp. 720–723.
- [10] T. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, pp. 293–306, 1985.
- [11] T. Ross, S. Worrell, V. Velten, J. Mossing, and M. Bryant, "Standard SAR ATR evaluation experiments using the mstar public release data set," in *Proc. SPIE*, April 1998, vol. 3370, pp. 566–573.