2010 Second International Workshop on Education Technology and Computer Science

A correlation coefficient based affine parameter estimation algorithm for image watermark detection

Xiaojun Tang, Qingxiu Du, Shuwu Zhang
Institute of Automation, Chinese Academy of Science, Beijing 100190, China
(E-mail: {xiaojun.tang, qingxiu.du, shuwu.zhang}@ia.ac.cn)

***Abstract*—In digital image watermark applications, most of watermark algorithms are vulnerable to geometrical affine transformation attacks. In order to resist affine transformation attacks, we propose a correlation coefficient based algorithm, which uses the original image to estimate affine parameters and recover the affine transformed image before watermark detection. Compared with existing affine parameter estimation algorithms, the correlation coefficient based algorithm is more precise, and is able to resist various image processing attacks, including cropping and darken (luminance modification). Therefore, the proposed algorithm is particularly suitable for affine transformation recovery in digital watermark applications.**

Keywords-digital watermark; affine transformation; parameter estimation; fast search algorithm.

I. INTRODUCTION

Nowadays, the problem of copyright violation of digital images and digital videos has become very serious, as digital storage and transmission make it trivial to quickly and inexpensively construct exact copies. Digital watermark, however, is a useful method of protecting multimedia data from intellectual piracy. Unfortunately, most of watermark algorithms, including the well known spread spectrum watermark [1], can't resist geometrical affine transformation attacks (translation, rotation and scaling are special cases of affine transformations). A feasible solution [2] is to estimate the affine parameters by comparing the original image and the affined image, and use the affine parameters to recover the affine transformed image before digital watermark detections. In [3], Zhang et al proposed a moment based algorithm, which use geometric moment to calculate affine parameters. In [4], Lucchese et al estimated affine parameters by the relationships between companion stretched slices of the Fourier transforms magnitudes of the two images. In [5], Kadyrov et al proposed a trace transform based algorithm, which calculates affine parameters by trace transforms. However, the above algorithms are not robust to cropping and darken attacks, and the estimation precision is not accurate enough for digital watermark applications. In order to accurately estimate the affine parameters, we propose a correlation coefficient based affine parameter search algorithm, which is straightforward and is composed of a coarse search algorithm, a precise search algorithm and a translation parameters estimation algorithm. Owing to the immunity of correlation coefficients to noise and malicious disturbances, the proposed search algorithm is especially impervious to cropping, darken and other attacks. Experiments verify the effectiveness of the proposed algorithm.

This work has been supported by the National Key Technology R&D Program (No.2008BAH26B02-3, No.2008BAH21B03-04 and No.2008BAH26B03), the National 863 Program (No. 2006AA01Z130) and the National Natural Science Foundation of PRC (No. 60773038).

II. EXISTING AFFINE PARAMETER ESTIMATION ALGORITHMS

We briefly introduce two typical affine parameter estimation algorithms in this section.

A. A Moment based algorithm

Zhang et al proposed a moment based RST (Rotation, Scaling and Translation) parameter estimation algorithm [3]. The moment M_{pq} of image I is defined by (1).

$$M_{pq} = \sum_m \sum_n m^p \cdot n^q \cdot I(m,n) \tag{1}$$

The moment based algorithm is simple and fast, but it's not suitable for general affine parameters estimation, and it's not robust to cropping and darken attacks. We denote the moments of the original image by M_{pq}, and denote the moments of the affine transformed image by M'_{pq}. The moment based algorithm is illustrated as follows:

1) Rotation parameter estimation:
The rotation angle θ can be calculated by resolving (2).

$$M'_{10} \cdot \cos\theta = M_{10} - M'_{01} \cdot \sin\theta \tag{2}$$

2) Scaling parameters estimation:
The scaling parameters λ_x and λ_y can be calculated by (3,4) (Note that (3,4) are slightly modified and are more concise than the formulas in [3]).

$$\lambda_x = \sqrt[3]{(M'_{10}/M_{10})^2/(M'_{01}/M_{01})} \tag{3}$$

$$\lambda_y = \sqrt[3]{(M'_{01}/M_{01})^2/(M'_{10}/M_{10})} \tag{4}$$

3) Translation parameters estimation:
The translation parameters m_0 and n_0 can be calculated by solving (5, 6).

$$M'_{01} = M_{01} + m_0 \cdot M_{00} \tag{5}$$

$$M'_{10} = M_{10} + n_0 \cdot M_{00} \tag{6}$$

B. A trace transform based algorithm

In [5], Kadyrov et al proposed a trace transform affine parameter estimation algorithm . The trace transformation is

978-0-7695-3987-4/10 $26.00 © 2010 IEEE
DOI 10.1109/ETCS.2010.237

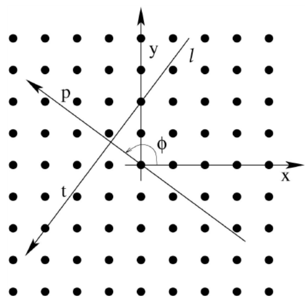

Figure 1. Trace transformation

illustrated in Fig.1 [5], in which a set of lines criss-cross an image. Each line l is defined by parameters ϕ and p. The trace transform calculates a function T(t) called the trace function, over parameter t along each line (ϕ, p).

Thus, one computes a number for each pair of values (ϕ, p). Plotting these numbers in a Cartesian coordinate system (ϕ, p) constitutes the trace transform of the image. One may then calculate another function, P(p), called the diametrical function, along the columns of the trace transform that is applied to parameter p to produce the so-called circus function of the image, $h(\phi)$. Note that the columns of the trace transform correspond to specific values of angle ϕ, so computing a function along each column is equivalent to making a calculation over batches of parallel lines. What function one computes is immaterial at this stage. The properties that these functions have to have in order to produce invariant features were discussed in [6]. The authors then use the circus function $h(\phi)$ to recover the parameters of the affine transform.

The trace transform based algorithm is able to estimate parameters of general affine transformations. Experiment results illustrate that this algorithm is robust to noise, JPEG compression and most image processing attacks. However, it is not robust to cropping and darken attacks.

III. FAST COARSE SEARCH ALGORITHM

In this section, given the original image I (luminance component) and the affine transformed image I_A, we propose a fast correlation coefficient based coarse search algorithm for affine parameter estimation, which uses the correlation coefficient as a criterion function in optimal parameter searches.

A. Correlation Coefficients

The correlation coefficient of vector X and Y is defined by (7) ($\overline{X}$ and $\overline{Y}$ denote the mean values of X and Y).

$$R=\frac{<X-\overline{X},Y-\overline{Y}>}{\sqrt{<X-\overline{X},X-\overline{X}><Y-\overline{Y},Y-\overline{Y}>}} \tag{7}$$

B. Affine Transformation Property of DFT

A general affine transformation is illustrated by (8) (note m and n are horizontal and vertical coordinates). It has four affine matrix parameters (a_{11}, a_{12}, a_{21} and a_{22}) and two translation parameters (m_0 and n_0).

$$\begin{bmatrix} m' \\ n' \end{bmatrix} = A\begin{bmatrix} m \\ n \end{bmatrix} + \begin{bmatrix} m_0 \\ n_0 \end{bmatrix}, \quad \left(A=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \right) \tag{8}$$

Affine transforming a 2D image in the pixel domain causes an affine transformation in the DFT (Discrete Fourier Transformation) domain [4]. In (9), F represents DFT coefficients. The translation parameters don't affect the magnitudes of DFT coefficients, and the affine matrix in DFT domain is $\left(A^{-1}\right)^T$ (the inverse and transpose of matrix A).

$$\left| F\left(\left(A^{-1}\right)^T \begin{bmatrix} u \\ v \end{bmatrix} \right) \right| \leftrightarrow I\left(A\begin{bmatrix} m \\ n \end{bmatrix} + \begin{bmatrix} m_0 \\ n_0 \end{bmatrix} \right) \tag{9}$$

Utilizing the above DFT property, we can first estimate the affine matrix $\left(A^{-1}\right)^T$ in DFT domain [4], and then calculate A in pixel domain by inverting and transposing $\left(A^{-1}\right)^T$. Finally, we search the translation parameters.

C. Exhaustive Search the Affine Matrix in DFT Domain

For convenience, we denote the inverse and transpose of matrix A by B ($B=\left(A^{-1}\right)^T$). We suppose that I and I_A are of size 256×256 (If images of different sizes are used, we can scale them to 256×256 before affine parameter estimation). In order to make the DFT coefficients continuous and smooth, we pad I and I_A with zeros to a size of 1024×1024. The DFT transformation is then applied and DFT coefficients F, and F_A are calculated correspondingly. In the DFT domain, the number of affine parameters is reduced to 4, so exhaustive searches of the 4 affine parameters are feasible. We first set the value ranges of elements of matrix B (for example: $0.5\le b_{11},b_{22}\le1.6$; $-0.8\le b_{12},b_{21}\le0.8$), and set the search step δ_1 as 0.02. Note that larger value ranges and smaller search steps are also allowable, however, they will lead to slower search speed. The exhaustive search algorithm is illustrated by the following steps:

1) Sequentially select a group of affine parameters b_{11},b_{12},b_{21} and b_{22} from the pre-defined value ranges. Apply the corresponding affine transformation on F (Formula (10, 11)), and calculate the affine transformed DFT coefficients $|F'|$.

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} \tag{10}$$

$$|F'(u',v')| = |F(u,v)| \tag{11}$$

2) We then calculate the correlation coefficient of $|F'|$ and $|F_A|$ (Formula (12)), and use it as a criterion function to check whether the selected affine parameters are right. In this formula, $\overline{|F'|}$ denotes the mean value of $|F'|$, and $\overline{|F_A|}$ denotes the mean value of $|F_A|$. Note that frequency f1 and f2 should be chosen to occupy a middle frequency range (for example f1=100, f2=150). The low frequency coefficients contain strong energy, and are not suitable for calculating correlation coefficients. In addition, high frequency coefficients are also

avoided since they are significantly modified during lossy compression such as JPEG.

$$R(B)=\frac{\sum_{u=f1}^{f2}\sum_{v=f1}^{f2}\left[|F'(u,v)|-\overline{|F'|}\right]\cdot\left[|F_A(u,v)|-\overline{|F_A|}\right]}{\sqrt{\sum_{u=f1}^{f2}\sum_{v=f1}^{f2}\left[|F'(u,v)|-\overline{|F'|}\right]^2\cdot\sum_{u=f1}^{f2}\sum_{v=f1}^{f2}\left[|F_A(u,v)|-\overline{|F_A|}\right]^2}} \quad (12)$$

3) If $R(B)$ is larger than a threshold T_1 ($T_1=0.5$), we successfully find a coarse estimation of the affine matrix B, and the search process can be stopped. Otherwise we return to Step 1, and sequentially select the next group of affine parameters.

D. *Speed Up the Exhaustive Search Algorithm*

In the above search algorithm, when the matrix B equals to $(A^{-1})^T$, the correlation coefficient $R(B)$ reaches the maximum value. General contour figures of $R(B)$ are illustrated in Fig.2. In each contour figure, we keep two affine matrix parameters fixed, and change the other two affine matrix parameters. In the first two contour figures, the contour lines are obviously elliptical, while in the other four contour figures the contour lines are relatively rounded. We conclude that $R(B)$ decreases steeply with ($b_{11}+b_{12}$), while deceases slowly with ($b_{11}-b_{12}$). Similarly, $R(B)$ is sensitive to ($b_{22}+b_{21}$), and is insensitive to ($b_{22}-b_{21}$). This property is independent of test images and affine parameters, and might be explained by the fact that magnitudes of DFT coefficients generally decrease as frequencies (only positive frequencies are used) increase. Making use of this property, we are able to speed up the search algorithm. We define a matrix B' by (13, 14).

$$\begin{cases} b'_{11}=(b_{11}+b_{12})/\sqrt{2}, & b'_{12}=(b_{11}-b_{12})/\sqrt{2} \\ b'_{21}=(b_{22}-b_{21})/\sqrt{2}, & b'_{22}=(b_{22}+b_{21})/\sqrt{2} \end{cases} \quad (13)$$

$$\begin{cases} b_{11}=(b'_{11}+b'_{12})/\sqrt{2}, & b_{12}=(b'_{11}-b'_{12})/\sqrt{2} \\ b_{21}=(b'_{22}-b'_{21})/\sqrt{2}, & b_{22}=(b'_{22}+b'_{21})/\sqrt{2} \end{cases} \quad (14)$$

Since $R(B)$ is not sensitive to b'_{12} and b'_{21}, we can increase the search step of b'_{12} and b'_{21}. For example, we set the search step of b'_{12} and b'_{21} as $10\times\delta$, and the speed of the search algorithm increases by 100 times.

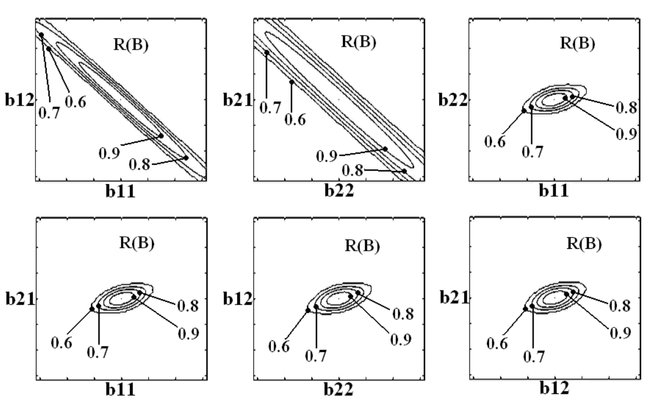

Figure 2. Typical contour figures of $R(B)$

IV. FAST PRECISE SEARCH ALGORITHM

In section III, we use the exhaustive search to get a coarse estimation which is near the optimal estimation. The coarse estimation is not precise enough for watermark detection, and need to be further refined. Since the correlation coefficient $R(B)$ is single peaked and smooth in the neighborhood of the optimal estimation, we can use a Quasi-Newton Method [7] to refine the coarse estimation. The applied Quasi-Newton Method is illustrated by the following steps.

1) Initialization: get a coarse estimation B_0 by the search algorithm in Section III, define a matrix H_0 to be a four-dimensional idendity matrix, and set variable k as 0.

2) If $\|\nabla R(B_k)\|$ is smaller than a threshold ε (for example $\varepsilon=0.001$), terminate the search process, otherwise calculate $d_k=-H_k\cdot\nabla R(B_k)$.

3) Choose an optimal α_k by one-dimensional search, and then calculate $B_{k+1}=B_k+\alpha_k\cdot d_k$.

4) Update the matrix H_{k+1} by the DFP method [7] (Formula (15)). In this formula, $\Delta B=B_{k+1}-B_k$ and $y_k=\nabla R(B_{k+1})-\nabla R(B_k)$.

$$H_{k+1}=H_k+\frac{\Delta B_k\Delta B_k^{\,T}}{y_k^{\,T}\Delta B_k}-\frac{H_k y_k y_k^{\,T} H_k^{\,T}}{y_k^{\,T} H_k y_k} \quad (15)$$

5) Increase k by 1, and return to Step 2.

V. TRANSLATION PARAMETERS ESTIMATION

After applying the precise search algorithm, we get a precise estimation of the affine matrix B, then the pixel domain affine matrix A can be calculated by inverting and transposing B ($A=(B^{-1})^T$). In this section, based on the estimated affine matrix A, we continue to estimate the translation parameters m_0 and n_0. We first ignore the translation parameters, and use the affine matrix A to recover the affine transformed image I_A. The recovered image is denoted by I_R. Note the translation parameters between I_R and I are also transformed by A^{-1} ($\begin{bmatrix} m'_0 \\ n'_0 \end{bmatrix}=A^{-1}\cdot\begin{bmatrix} m_0 \\ n_0 \end{bmatrix}$), so we can first search the parameter m'_0 and n'_0, and then calculate m_0 and n_0 by inverse transform. We set the value ranges of m'_0 and n'_0 as $-50\le m'_0,n'_0\le 50$, and set the search step δ_2 as 1, then m'_0 and n'_0 can be estimated by the following steps.

1) Sequentially select a group of parameter m'_0 and n'_0 from the pre-defined value ranges, and reversely shift the image I_R (the shifted image is denoted by I'_R).

2) Calculate the correlation coefficient of I'_R and I (Formula (16)).

$$R(m'_0,n'_0)=\frac{\sum_{m=1}^{256}\sum_{n=1}^{256}\left[I'_R(m,n)-\overline{I'_R}\right]\cdot\left[I(m,n)-\overline{I}\right]}{\sqrt{\sum_{m=1}^{256}\sum_{n=1}^{256}\left[I'_R(m,n)-\overline{I'_R}\right]^2\cdot\sum_{m=1}^{256}\sum_{n=1}^{256}\left[I(m,n)-\overline{I}\right]^2}} \quad (16)$$

3) If $R(m'_0, n'_0)$ is larger than a threshold T_2 (for example $T_2 = 0.5$), we successfully find a coarse estimation of m'_0 and n'_0, and the search process can be stopped. Otherwise we return to Step 1, and sequentially select the next group of translation parameters.

Similarly, the coarse estimation of m'_0 and n'_0 can be refined by a Quasi-Newton Method (similar to the algorithm in Section IV). Finally, the parameter m_0 and n_0 can be calculated by the formula: $\begin{bmatrix} m_0 \\ n_0 \end{bmatrix} = A \cdot \begin{bmatrix} m'_0 \\ n'_0 \end{bmatrix}$.

VI. EXPERIMENT RESULTS

In this section, we perform experiments to verify the effectiveness of the proposed affine parameter search algorithm. We choose a 256×256 Lena image as the test image. Fig.3 illustrates different kinds of affine transformations, including translation, rotation, scaling (scale to 250×220) and general affine transformation. Fig.4 illustrates different kinds of image processing attacks, including cropping (60% remains), noise (5%), JPEG (20% quality), Darken (nonlinear) and Darken plus Histogram Equalization. We also test the moment based algorithm [3] and the trace transform based algorithm [5] for comparison. In the trace transform based algorithm, the selected trace function and diagonal function are illustrated in (17, 18) (t_1 is defined by $\int_{-\infty}^{t1} f(t)dt = \int_{t1}^{\infty} f(t)dt$).

$$T(t) = \int_{-\infty}^{\infty} \left| t - t_1 \right|^2 f(t)dt \quad (17)$$

$$P(p) = \int_{-\infty}^{\infty} f(p)dp \quad (18)$$

The experiment results are displayed in Table I. In this table, Mom represents the moment based algorithm, Trace represents the trace transform based algorithm, and Prop represents the proposed algorithm (Section III, IV and V). Since the moment based algorithm is not suitable for general affine parameters estimation, the corresponding test results are not displayed. We conclude from the test results that both the moment and the trace transform based algorithms can't resist cropping and darken attacks, while the proposed algorithm is robust to various attacks. In addition, the proposed algorithm is generally more precise than the other two algorithms. Note that in the noise attack, the translation parameter estimation error of the proposed algorithm is 0.32 (higher than other attacks). This problem can be solved by low pass filtering the noise image (the estimation error can be reduced to 0.01 after filtering).

Figure 3. Different kinds of affine transformations

Figure 4. Different kinds of attacks

TABLE I. AFFINE PARAMETER ESTIMATION

Attacks	**Errors of matrix A**			**Errors of m0 and n0**		
	Mom	***Trace***	***Prop***	***Mom***	***Trace***	***Prop***
Translation	0	0	0.0017	0	0	0.013
Rotation	0.0066	0.012	0.0019	—	0.21	0.041
Scaling	0.0038	0.016	0.0017	—	0.15	0.052
Affine	—	0.05	0.0012	—	0.19	0.037
Cropping	0.21	0.25	0.0058	8.4	6.6	0.031
Noise (5%)	0.0012	0.0002	0.0034	0.033	0.12	0.32
JPEG (20%)	0.0003	0.02	0.0013	0.87	0.10	0.016
Darken	0.078	0.13	0.001	3.1	5.7	0.021
Darken + Histogram equalization	0.074	0.11	0.0007	2.9	4.5	0.032
Affine + Cropping	—	0.29	0.005	—	7.2	0.07
Affine + Noise (5%)	—	0.051	0.0044	—	0.2	0.094
Affine + JPEG (20%)	—	0.06	0.0053	—	0.18	0.082
Affine + Darken	—	0.12	0.0024	—	6.1	0.023
Affine + Darken + Histogram equalization	—	0.13	0.003	—	6.9	0.033

VII. CONCLUSIONS

In this paper, we propose a correlation coefficient based affine parameter search algorithm, which is especially suitable for affine transformation recovery in digital image watermark applications. Compared with the moment based algorithm and the trace transform based algorithm, the proposed algorithm is more accurate and is robust to cropping and darken attacks. In addition, the speed of the coarse search algorithm (Section III) needs to be improved in future work (it takes about 30 seconds for successfully finding a group of general affine parameters).

REFERENCES

[1] I.J. Cox, J. Kilian, F.T. Leighton, T. Shamoon, "Secure spread spectrum watermarking for multimedia," IEEE Transactions on Image Processing 6 (12) (1997) 1673–1687.

[2] Dong Zheng, Yan Liu, Jiying Zhao, Abdulmotaleb El-Saddik: A survey of RST invariant image watermarking algorithms. ACM Comput. Surv. 39(2): (2007)

[3] Li Zhang, Weiwei Xiao, Jihong Zhang, "Affine Invariant Watermarking Transform Based on Geometric Moment," Journal of Shenzhen University (Science & Engineering), vol. 20, Jun. 2003. (in Chinese)

[4] L. Lucchese, G. M. Cortelazzo, and C. Monti, "Estimation of affine transformations between image pairs via Fourier transform," Proc. Int. Conf. Image Processing, Switzerland, Sep. 1996, vol. 3, pp. 715–718.

[5] Alexander Kadyrov, Maria Petrou: Affine Parameter Estimation from the Trace Transform. IEEE Trans. Pattern Anal. Mach. Intell. 28(10): 1631-1645 (2006)

[6] M. Petrou and A. Kadyrov, "Affine Invariant Features from the Trace Transform," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, no. 1, pp. 30-44, Jan. 2004.

[7] Yaxiang Yuan, Wenyu Sun, "Optimization theory and methods", Chinese Science Press, pp 219-225, 1997. (in Chinese)