

# Evolutionary Multi-objective Image Feature Extraction in the Presence of Noise

Wissam A. Albukhanajer, *Student Member, IEEE*, Johann A. Briffa, *Member, IEEE*, and Yaochu Jin, *Senior Member, IEEE*

**Abstract**—A Pareto-based evolutionary multi-objective approach is adopted to optimize the functionals in the Trace Transform for extracting image features that are robust to noise and invariant to geometric deformations such as rotation, scale and translation (RST). To this end, sample images with noise and with RST distortion are employed in the evolutionary optimization of the Trace Transform, which is termed evolutionary Trace Transform with noise (ETTN). Experimental studies on a fish image database and the Columbia COIL-20 image database show that the ETTN optimized on a few low-resolution images from the fish database can extract robust and RST invariant features from the standard images in the fish database as well as in the COIL-20 database. These results demonstrate that the proposed ETTN is very promising in that it is computationally efficient, invariant to RST deformation, robust to noise and generalizable.

**Index Terms**—Trace transform, invariant feature extraction, evolutionary algorithms, multi-objective optimization, image identification.

## I. INTRODUCTION

IN multi-class machine learning, invariant image identification is a challenging problem especially in the presence of noise and geometric deformations such as rotation, scale and translation (RST). In contrast to image classification, image identification focuses on finding out the geometrically transformed versions of an image from a large number of different images. For example, a rotated version of a shark fish image is still considered as the same image, although it may look different from the original fish image. On the other hand, image identification is methodologically similar to image classification in that RST invariant image identification also consists of two main stages: feature extraction and classification. Naturally, robust and efficient feature extraction techniques can substantially enhance the identification performance.

Improving image feature extraction techniques has attracted increasing interest over the past few decades. For instance, the image moments method was developed for achieving

geometric invariance [1]. Early work on image moments was undertaken by Hu [2] in 1962 and much work has been geared towards improving the performance of image moments [3]–[5]. On the other hand, the Radon transform [6] was widely investigated for its ability in capturing the directional features of an image and robustness to zero mean white noise [7]. Tabbone *et al.* [8] presented a computationally efficient region-based shape descriptor based on Radon transform and Fourier transform to construct a one-dimensional descriptor invariant to RST transformations. Furthermore, Hasegawa and Tabbone [9] presented a shape descriptor based on amplitude extraction and log-mapping to project shapes scaling and rotation onto a translation in the radial coordinate of the Radon space.

Trace transform can be seen as a more general case of Radon transform, which has been successfully applied to many image processing tasks such as image database retrieval [10], texture classification [11], insect footprint recognition [12] and character recognition [13]. Trace transform involves calculating functionals along straight lines crossing the image in different directions. It can be constructed in the same way as constructing Radon transform but using various functionals rather than using the line integral function only.

Trace transform has received much attention in the field of image analysis and invariant feature extraction due to its effectiveness in producing multiple features that describe an image [10], [11]. Kadyrov and Petrou [10], [14]–[18] proposed the Trace transform and a theory for invariant Triple feature extraction. They derived different functionals for image database search invariant to RST transformation and robust to noise [10]. The same authors also constructed features invariant to affine distortions by choosing appropriate functionals in the transform [15]. Later on, in [16] they dealt with occlusion and illumination changes by considering properties of other methods such as Fourier transform and median filter when constructing functionals in Trace transform. In [17], [18] they extended their earlier work for image retrieval to achieve robustness to affine distortion as well as robustness to noise, occlusion and illumination changes. The method was evaluated on a gray level fish image database and it was shown that the proposed Trace transform outperforms the moment invariants developed by Flusser and Suk [19]. Srisuk *et al.* [20] reported a framework for invariant image retrieval by defining a weight matrix from the differences of the Trace transforms using several Trace functionals. This approach introduces better performance in dealing with local texture as well as geometric distortions of planar objects.

Manuscript received January 1, 2014; revised July 25, 2014 and September 19, 2014; accepted September 20, 2014. This work was supported in part by the EPSRC Industrial CASE Studentship under Grant 10001560, in part by the Joint Research Fund for Overseas Chinese, Hong Kong, and in part by the Macao Scholars of the National Natural Science Foundation of China under Grant 61428302. This paper was recommended by Associate Editor Y. Tan.

The authors are with the Department of Computing, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom (e-mail: w.albukhanajer@surrey.ac.uk; j.briffa@surrey.ac.uk; yaochu.jin@surrey.ac.uk). Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>. Digital Object Identifier 10.1109/TCYB.2014.2360074

However, it is computationally expensive since multiple Trace matrices need to be constructed for calculating the weight matrix. Siskovicova *et al.* [21] considered the autocorrelation function when constructing invariant Triple features using Trace transform. The combination of the Trace functionals was selected empirically and applied to the original images to produce the invariant Triple features for comparison with the distorted images using the Euclidean classifier. The recognition performance on gray-level images is poorer than that of the binary images with translation-distortion images and not robust to noise. A similar framework is developed by Turan *et al.* [22] for road sign recognition. Brasnett and Bober [23] proposed a multi-resolution Trace transform by subsampling the original transform to build an image identifier invariant to general image deformation and robust to blur, noise and flip. The binary identifier is constructed from the Trace transform by using two functionals and the magnitude of Fourier transform is obtained from each function. Then, the difference between the magnitude of the neighbor coefficients of each function is obtained to form two binary identifiers. At the end, the complete binary identifier is constructed and the distance between different identifiers is measured by a normalized Hamming distance for recognition.

From all previously mentioned work, the choice of proper Trace transform functionals is central to the efficiency and robustness of the transform. Most recently, research effort has been dedicated to selecting the optimal combinations of functionals in Trace transform. In an earlier work [24], [25], we used evolutionary algorithms to find the best combinations of the classical Trace functionals, thereby reducing the computational cost while maintaining high identification performance for RST invariant image identification. Frias-Velazquez *et al.* [26] proposed a feature selection methodology based on Laguerre polynomials by minimizing the dependency among signatures from functionals of the classical Trace transform and tested the method on vehicle identification. Although there was no significant increase in the identification rate compared to the classical Trace functional [17], the computational cost was reduced because only eight signatures were used instead of 22 in [17].

This paper presents a substantial extension of our preliminary work reported in [24], [25]. New contributions of the paper are as follows. First, we demonstrate that noise must be deliberately injected into the sample images for evolutionary optimization of the Trace transform to achieve image features that are robust not only to RST deformation, but also to noise. To empirically show the necessity of adding noise in the sample images to achieve robustness to noise, a separate set of experiments has been performed on evolutionary Trace transform using sample images without noise, ETT for short. Our comparative results clearly indicate that image features extracted by ETT are sensitive to noise, although they are invariant to RST deformation, whilst those extracted by ETTN are robust to both noise and RST distortion. Second, we examine the generalization ability of the ETTN, which is critical for the ETTN to be applicable to a wide range of images without re-performing the evolutionary optimization. This can be considered as a kind of transfer learning [27],

which has become popular in recent years and shown to be very promising in improving learning performance when there is a lack of training data. We empirically prove that Trace transforms optimized on a few low-resolution images from a fish database are able to extract robust and RST invariant features for standard images (having a higher resolution) in the fish database, as well as those in a completely different database, i.e., the Columbia COIL-20 database. These results make it evident that ETTN has a strong generalization ability, which is essential for the applicability of ETTN to real-world image feature extraction. Finally, we analyze the computational complexity of the canonical Trace transform and the ETTN, which indicates that ETTN is computationally more efficient than the canonical Trace transform.

It should be mentioned that a large amount of work has been reported on extracting image features using evolutionary algorithms. One research methodology that is most relevant to this work is evolutionary image feature extraction, typically using genetic programming (GP). Neshatian *et al.* [28] proposed a GP-based system to construct multiple features. Good features are selected according to the evaluation of its power to discriminate between the instances of different classes. In [29], GP is used to extract domain-independent image features. These features are not necessarily similar to the human experts with domain knowledge, but can still perform similarly well. Generally, features extracted using evolutionary algorithms are more efficient to compute and perform reasonably well compared to manually constructed features, see [30] and [31] for example and references cited therein.

For complex image analysis in particular in the presence of noise, it remains a big challenge to build a robust image identification system. Note that noisy data-driven modeling has undergone fast developments in computational intelligence due to its importance to tackle problems in various real world applications [32]. As an example from real world applications, in [33], [34] noisy data is constructed from an industrial iron blast furnace to train a neural network model using the model error and the network size as two objectives to be minimized. In [35] a fuzzy neural model is constructed for turbo generator identification utilizing noisy data injected in the training pattern to learn the dynamic system behavior in the turbo generator model. These research findings might be instructive for dealing with noise in image processing using computational intelligence techniques.

Another challenge in image analysis is the requirement for a large number of training samples to build a good image identification model [36]. To address this problem, *one-shot learning* [36], [37] has been suggested, aiming to replicate the ability of the human vision system to learn new objects from one or a few examples only. In [37], the one-shot learning approach is adopted by using only two instances of each class to evolve image classifier via genetic programming.

The remainder of this paper is organized as follows. A brief overview of Trace transform algorithm is given in Section II. In Section III, the proposed evolutionary multi-objective Trace transform will be described, including the main components of the evolutionary algorithm used in ETTN. In Section IV, experimental results are presented on two benchmark databases

(Fish-94 database and COIL-20 database), followed by analysis of the computational complexity of ETTN and the canonical Trace transform in Section V. Conclusions and future work are discussed in VI.

## II. FEATURE CONSTRUCTION BY TRACE TRANSFORM

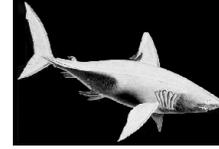
Among several support region feature extraction approaches, i.e., local, region and global [38], global support region feature extraction methods have been found to be most promising and efficient due to their low computational complexity and strong robustness [39]. The Trace transform [14] can represent an image in a different form by applying a finite functional along lines crossing the image pixels in different directions [10], see Fig. 1(b). Let  $I$  denotes an  $N \times N$  image of real valued pixels in the range  $[0, 1]$ , then a real number called a *Triple feature* can characterize image  $I$  using Trace transform as follows.

- 1) Trace transform of image  $I$  is obtained by defining a number of straight lines, each line is characterized by a distance  $\rho$  and a direction  $\theta$ . By applying a functional  $T$  called *Trace*, over image pixels along parameter  $t$  defined on the tracing line, refer to Fig.1(b), a two-dimensional matrix is obtained as a function of the variables  $\rho$  and  $\theta$ , as shown in Fig. 1(c), where  $\theta$  is the orientation of the tracing line,  $\rho$  is the line distance from the origin. Each pixel in the transform domain corresponds to a result obtained by applying a Trace functional along a line in the original image. This step consumes most of the computational cost in extracting the Triple features. The calculations are made at a sub-pixel accuracy by defining a fine grid over the source image with one pixel sampling step for  $\theta$ ,  $\rho$  and  $t$  parameters. This captures the fine details on the image including specific object shapes such as the sharp corners. The reader is referred to [40] for details about the practical implementation of Trace transform and defining the beginning and the end points of each tracing line.
- 2) Apply another functional  $D$  called *Diametric* along columns of Trace matrix obtained from step 1, a sequence of numbers of length  $n_\theta$  is obtained, where  $n_\theta$  is the number of orientations considered.
- 3) Finally, a Triple feature is obtained by applying a third functional  $C$  called *Circus* over the final sequence of numbers in step 2 (over parameter  $\theta$ ). A Triple feature is a scalar value (real number) denoted by  $\Xi$ , which can be used to form a unique identifier (feature) for the image [18].

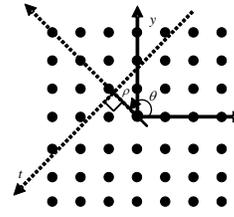
Different Triple features can be obtained by using different Trace functionals  $T$ , Diametric functionals  $D$  and Circus functionals  $C$ . Figure 2 summarizes the main steps for Triple feature construction.

Tables I to III depict a set of Trace, Diametric and Circus functionals, respectively. The functionals used in the Trace transform and their combination will directly affect the robustness of the extracted features and the computational complexity of Trace transform. Our preliminary work [24],

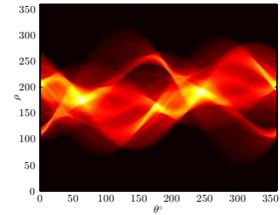
[25] has already shown that finding an optimal combination of functional used in Trace transform can significantly enhance the invariance to RST distortions for image identification. In the present work, we will investigate the robustness of the extracted Trace features to noise and the applicability of the optimized Trace transforms to different image databases. Therefore, in the next section we will introduce the evolutionary multi-objective algorithm used for optimizing the Trace transform.



(a) Fish image.



(b) Trace parameters [10].



(c) Trace transform of image (a).

Fig. 1. The Trace transform.

## III. EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION OF TRACE TRANSFORM

### A. Multi-objective optimization

Most real-world optimization problems have more than one objective to be optimized. More often than not, these objectives are conflicting with each other. Consequently, there is a set of solutions to multi-objective optimization problems rather than one single optimal solution.

Consider a multi-objective minimization problem having  $N_f$  objectives,  $f_1, f_2, \dots, f_{N_f}$ . Let  $s_1$  and  $s_2$  are two solutions in a feasible solution set  $S$  consisting of  $N_s$  solutions. Then solution  $s_1$  is said to dominate solution  $s_2$  if  $\forall i, f_i(s_1) \leq f_i(s_2)$ ,  $i = 1, 2, \dots, N_f$ , and  $\exists j \in 1, 2, \dots, N_f$  such that  $f_j(s_1) < f_j(s_2)$ . A solution  $s$  is called *Pareto optimal* if there is no other solutions that dominates  $s$  and the set of all Pareto optimal solutions is called the Pareto set. The image formed by the Pareto optimal solutions in the objective space is termed *Pareto front* [41]. Note that in solving many real-world problems, it is very hard, if not impossible to verify whether the non-dominated solution set achieved by a multi-objective optimization algorithm is Pareto optimal or not. For the sake of simplicity, the non-dominated solution set is often called Pareto set.

The Pareto-based approach has witnessed great success in evolutionary multi-objective optimization [42], [43] as well as in evolutionary multi-objective learning [44]. Thus, it is natural to adopt the Pareto-based evolutionary approach to optimize the combination of functionals in the Trace transform for extracting robust image features.

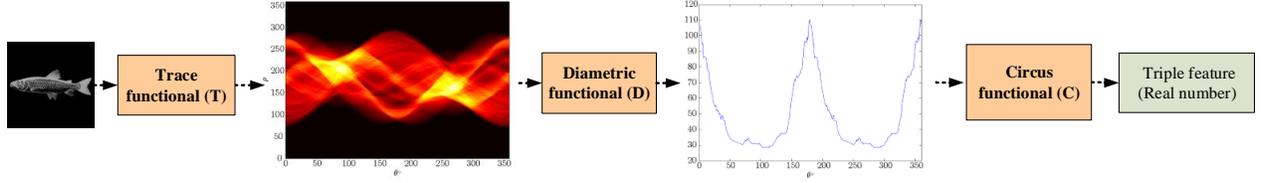


Fig. 2. Triple feature construction

TABLE I

LIST OF TRACE FUNCTIONAL  $T$ .  $n_t$  IS THE TOTAL NUMBER OF IMAGE PIXELS ALONG THE TRACING LINE (I.E. ALONG PARAMETER  $t$  IN FIG 1(B)), AND  $\tau_i$  IS THE VALUE OF THE  $i^{\text{TH}}$  PIXEL ALONG THE TRACING LINE. FUNCTIONALS ARE APPLIED ALONG EACH LINE DEFINED BY  $\rho$  AND  $\theta$ )

| No.      | Functional   |
|----------|--|
| $T_0$    | $\sum_{i=1}^{n_t} \tau_i$  |
| $T_1$    | $\max_{i=1}^{n_t} \tau_i - \min_{i=1}^{n_t} \tau_i$  |
| $T_2$    | $\left(\sum_{i=1}^{n_t}  \tau_i ^{\frac{1}{2}}\right)^2$   |
| $T_3$    | $\left(\sum_{i=1}^{n_t}  \tau_i ^4\right)^{\frac{1}{4}}$   |
| $T_4$    | $\sum_{i=1}^{n_t}  \tau'_i $   |
| $T_5$    | $\sqrt{\frac{1}{n_t} \sum_{i=1}^{n_t} (\tau_i - M)^2}$ , $M = \frac{1}{n_t} \sum_{i=1}^{n_t} \tau_i$   |
| $T_6$    | $\sum_{i=1}^{n_t} \sqrt{ \tau_i }$   |
| $T_7$    | $\max_{i=1}^{n_t}  \tau_i $  |
| $T_8$    | $\sum_{i=c}^{n_t} (i-c)^2 \tau_i$ , $c = \frac{\sum_{i=1}^{n_t} i  \tau_i }{\sum_{i=1}^{n_t}  \tau_i }$  |
| $T_9$    | $\sum_{i=c}^{n_t} (i)^2 \tau_i$ , $c = \frac{\sum_{i=1}^{n_t} i  \tau_i }{\sum_{i=1}^{n_t}  \tau_i }$  |
| $T_{10}$ | $\sum_{i=c}^{n_t} (i)^3 \tau_i$ , $c = \frac{\sum_{i=1}^{n_t} i  \tau_i }{\sum_{i=1}^{n_t}  \tau_i }$  |
| $T_{11}$ | $\sum_{i=c}^{n_t} (r)^{0.5} \tau_i$ , $c = \frac{\sum_{i=1}^{n_t} l  \tau_i }{\sum_{i=1}^{n_t}  \tau_i }$ , $r =  l - c $ , $l = 1, 2, \dots, n_t$ |
| $T_{12}$ | $\sum_{i=c}^{n_t} (r) \tau_i$ , $c = \frac{\sum_{i=1}^{n_t} l  \tau_i }{\sum_{i=1}^{n_t}  \tau_i }$ , $r =  l - c $ , $l = 1, 2, \dots, n_t$       |
| $T_{13}$ | $\sum_{i=c}^{n_t} (r)^2 \tau_i$ , $c = \frac{\sum_{i=1}^{n_t} l  \tau_i }{\sum_{i=1}^{n_t}  \tau_i }$ , $r =  l - c $ , $l = 1, 2, \dots, n_t$     |

### B. Multi-objective Trace Transform

In image identification, discrimination between different image classes is important for classification. More specifically, the variations of the features extracted from variants of the same image, measured by within-class variance, should be as small as possible. On the other hand, features extracted from different images, denoted by between-class variance, should be as large as possible to make it easy for classifiers to distinguish one image from others. Therefore, optimization of Trace transform is concerned with two objectives, namely, maximization of the between-class variance and minimization of the within-class variance. This is a typical bi-objective optimization problem with two objectives being conflicting with each other.

Note that additional objectives can be considered. For example, for applications where the computational time is critical,

TABLE II

LIST OF DIAMETRIC FUNCTIONAL  $D$ .  $n_\rho$  IS THE TOTAL NUMBER OF ELEMENTS ALONG THE COLUMNS OF TRACE MATRIX (I.E. ALONG PARAMETER  $\rho$  IN FIG. 1(C)), AND  $\delta_i$  IS THE VALUE OF THE  $i^{\text{TH}}$  SAMPLE ALONG THE COLUMNS OF THE TRACE MATRIX. FUNCTIONALS ARE APPLIED FOR EACH  $\theta$

| No.   | Functional   |
|-------|--|
| $D_0$ | $\sum_{i=1}^{n_\rho} \delta_i$   |
| $D_1$ | $\max_{i=1}^{n_\rho} \delta_i$   |
| $D_2$ | $\left(\sum_{i=1}^{n_\rho}  \delta_i ^{\frac{1}{2}}\right)^2$  |
| $D_3$ | $\left(\sum_{i=1}^{n_\rho}  \delta_i ^4\right)^{\frac{1}{4}}$  |
| $D_4$ | $\sqrt{\sum_{i=1}^{n_\rho} \delta_i^2}$  |
| $D_5$ | $\max_{i=1}^{n_\rho} \delta_i - \min_{i=1}^{n_\rho} \delta_i$  |
| $D_6$ | $\sum_{i=1}^{n_\rho}  \delta'_i $  |
| $D_7$ | $\sum_{i=c}^{n_\rho} (i-c)^2 \delta_i$ , $c = \frac{\sum_{i=1}^{n_\rho} i  \delta_i }{\sum_{i=1}^{n_\rho}  \delta_i }$ |

TABLE III

LIST OF CIRCUS FUNCTIONAL  $C$ .  $n_\theta$  IS THE TOTAL NUMBER OF ELEMENTS IN THE ROW DIRECTION OF TRACE MATRIX (I.E. ALONG PARAMETER  $\theta$ ) GENERATED PREVIOUSLY BY APPLYING  $D$  FUNCTIONALS, AND  $\xi_i$  IS THE VALUE OF THE  $i^{\text{TH}}$  SAMPLE IN THE ROW DIRECTION OF TRACE MATRIX

| No.   | Functional   |
|-------|--|
| $C_0$ | $\sum_{i=1}^{n_\theta} \xi_i$  |
| $C_1$ | $\text{median}_{i=1}^{n_\theta} \xi_i$   |
| $C_2$ | $\sqrt{\frac{1}{n_\theta} \sum_{x=1}^{n_\theta} (\xi_i - M)^2}$ , $M = \frac{1}{n_\theta} \sum_{i=1}^{n_\theta} \xi_i$ |
| $C_3$ | $\sum_{i=1}^{n_\theta}  \xi'_i $   |
| $C_4$ | $\max_{i=1}^{n_\theta} \xi_i$  |
| $C_5$ | $\max_{i=1}^{n_\theta} \xi_i - \min_{i=1}^{n_\theta} \xi_i$  |

an objective can be included to find out the functionals that minimize the computational time. Another possible objective is robustness to noise. Although noise is considered in this work, it is not explicitly handled as an objective. If robustness to noise is most important for some specific applications, an objective for robustness to noise can be included.

As discussed above, the two objectives in optimizing the Trace transform are the minimization of the within-class variance ( $S_w$ ) and the maximization of the between-class variance ( $S_b$ ). For convenience, both objectives are formulated

as minimization problems. To this end,  $f_2$  is defined to be the inverse of  $S_b$  with an addition of a small positive constant to avoid numerical problems. It is worth mentioning that this is not the only possible choice to convert a maximization problem into a minimization one. For example, we can also formulate the second objective as  $f_2 = C - S_b$ , where  $C$  is a big positive constant.

Based on the above discussions, the optimization of the Trace transform can now be formulated as a bi-objective optimization problem:

$$\begin{aligned} \min\{f_1, f_2\}, \\ f_1 = S_w, \\ f_2 = \frac{1}{(S_b + \epsilon)}, \end{aligned} \quad (1)$$

where  $\epsilon$  is a small positive number to avoid division by zero, and the within-class variance  $S_w$  and the between-class  $S_b$  are defined by:

$$S_w = \sum_{k=1}^K \sum_{j=1}^{C_k} (\Xi_{jk} - \mu_k^{\Xi})^2 \quad (2a)$$

$$S_b = \sum_{k=1}^K (\mu_k^{\Xi} - \mu^{\Xi})^2 \quad (2b)$$

where

$$\mu_k^{\Xi} = \frac{1}{C_k} \sum_{j=1}^{C_k} \Xi_{jk}, \quad \mu^{\Xi} = \frac{1}{K} \sum_{k=1}^K \mu_k^{\Xi},$$

where  $K$  is the number of classes,  $C_k$  is the number of samples in class  $k$ ,  $\mu_k^{\Xi}$  is the mean of class  $k$  of  $\Xi$  Triple features,  $\Xi_{jk}$  is the  $j^{\text{th}}$  Triple features of class  $k$ , and  $\mu^{\Xi}$  is mean of all classes of  $\Xi$  Triple features.

Many multi-objective evolutionary algorithms (MOEAs) have been developed over the past two decades [42]. Here, we adopt the Elitist Non-dominated Sorting Genetic Algorithm (termed NSGA-II) [45]. NSGA-II has been adopted as it has been shown to be a very powerful and robust evolutionary algorithm for a wide range of multi-objective optimization problems, in particular when the number of objectives is lower than three. In addition, NSGA-II is computationally relatively efficient compared to many other counterparts such as the strength Pareto evolutionary algorithm (SPEA-2) [46]. Due to the above reasons, NSGA-II has been adopted for solving the bi-objective optimization problem studied in this work.

In the following, the main components of the evolutionary multi-objective optimization of Trace transform using NSGA-II are listed:

- The *Chromosome* encodes three integer parameters representing different Trace functionals, namely, Trace  $T$ , Diametric  $D$  and Circus  $C$ .
- The *Population* is randomly initialized taking into account the range of each design variable. For instance, there are 14 Trace functionals in Table I,  $T_0$  to  $T_{13}$ . Therefore,  $T$  varies from 0 to 13. Similarly, there are eight Diametric functionals and six Circus functionals, see Tables II and III. Consequently,  $D$  and  $C$  take the value from 0 to 7 and from 0 to 5, respectively. Therefore,

there are a total 672 possible configurations. It should be pointed out that many more functions can be used, which will lead to a much larger number of combinations. In addition, parameters in the Trace transform such as sampling steps can also be optimized.

- The *Fitness* function consists of two objectives,  $f_1$  and  $f_2$  as defined in (1), which aim to minimize the within-class variance  $S_w$  and maximize the between-class variance  $S_b$ , as defined in (2) [47].
- The *Selection* operation is the same as the selection strategy suggested in NSGA-II [42], which is composed of four main steps. First, combine the parent population with the offspring. Second, all individuals in the combined population are assigned a Pareto front number and a crowding distance [45]. Third, all individuals are then sorted according to the assigned Pareto front number in an ascending order and individuals having the same Pareto front number are sorted according to the crowding distance in a descending order. Finally, the top  $N_p$  individuals, where  $N_p$  is the population size, are selected and passed to the next generation.
- *Crossover* and *Mutation* operations are applied on the selected individuals to generate offspring. In this work we adopt *uniform crossover* and *uniform mutation* which allows parent chromosome to change at gene level instead of segment level. Crossover and mutation operations occur during the evolution with *crossover probability*  $P_c$  and a *mutation probability*  $P_m$ , respectively.

#### IV. EXPERIMENTAL RESULTS

The experiments presented in this section aim to examine the performance of the proposed evolutionary Trace transform (ETT) with respect to its robustness to RST distortions and noise in images. To verify the generalization ability of the ETT, we are also keen to know whether the optimized ETT using low-resolution sample images from one database can perform well on images of higher-resolution from the same database and from other databases. To this end, we designed two sets of experiments based on two image databases, the Fish-94 database [10] and the Columbia COIL-20 database [48]. In the first setup, the sample images used in evolutionary optimization consist of five different images randomly chosen from the Fish-94 database, each class containing four images generated as follows:

- *Sample 1*: A low-resolution image ( $64 \times 64$ ) generated from a randomly chosen original image ( $256 \times 256$ );
- *Sample 2*: Random rotation [ $1-359^\circ$ ] of Sample 1;
- *Sample 3*: Random translation of Sample 1 (objects remain within image boundaries);
- *Sample 4*: Random scale (0.1-0.9) of Sample 1.

In the second setup, sample images include three different classes, each containing five different types of changes. The major difference here is that Gaussian noise is added to the sample images apart from RST deformations:

- *Sample 1*: A low-resolution image from ( $64 \times 64$ ) generated from a randomly chosen original image ( $256 \times 256$ );

TABLE IV  
NSGA-II PARAMETERS SET-UP

| Parameter             | Value     |
|-----------------------|-----------|
| $N_p$                 | 150       |
| $P_m$                 | 0.125     |
| $P_c$                 | 0.9       |
| Number of generations | 200       |
| $\epsilon$            | $10^{-5}$ |

TABLE V  
TRIPLE FEATURES COMBINATIONS FROM ETT AND ETTN

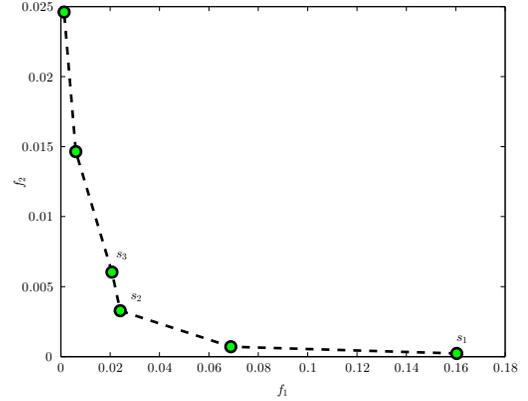
| Solution No. | ETT            | ETTN        |
|--------------|----------------|-------------|
| $s_1$        | $T_{12}D_4C_2$ | $T_0D_5C_5$ |
| $s_2$        | $T_6D_3C_5$    | $T_0D_3C_2$ |
| $s_3$        | $T_0D_3C_1$    | $T_0D_1C_2$ |

- *Sample 2*: Random rotation, scale and translation of Sample 1 with Gaussian noise (standard deviation=4);
- *Sample 3*: Random rotation, scale and translation of Sample 1 with Gaussian noise (standard deviation=6);
- *Sample 4*: Random rotation of Sample 1;
- *Sample 5*: Random scale of Sample 1.

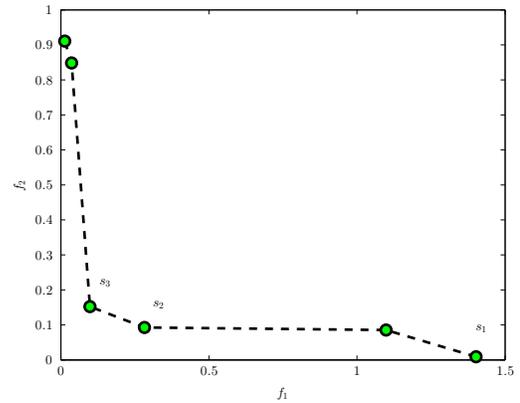
For convenience, we call the evolutionary Trace transform using images in the first setup ETT, and the second setup ETTN. Therefore, there are 20 sample images in ETT and 15 images in ETTN. The NSGA-II is implemented using the Shark Machine Learning library [49]. Table IV lists the parameter setup for the evolutionary algorithm.

After 200 generations of evolution, a set of non-dominated solutions are obtained from the two experiments, as depicted in Fig. 3 (a) and (b), respectively. As seen from the figure, there are six solutions in each Pareto-front of ETT and ETTN (dominated and infeasible solutions were eliminated). To test the performance of optimized Trace transforms obtained by ETT and ETTN, we selected three solutions (marked by the numbers) from the Pareto front for each setup for image identification. As suggested in [44], solutions around the knee point of the Pareto-front, for example, solutions 2 and 3 in Fig. 3 are selected for test as these solutions achieve the best trade-off between two objectives. Intuitively, solutions having a larger between-class variance ( $f_2$ ) are preferred if their within-class variance is adequately small to construct features that ease classification. Therefore, we selected an additional solution with a minimum  $f_2$ , i.e., solution 1. In general, it is the user's preference to choose solutions from the Pareto-front, which is one of the main advantages of using multi-objective optimization over a single objective optimization. Solutions from ETT and ETTN corresponding to Triple features are detailed in Table V.

In the following, features extracted by ETT and ETTN will be applied to image identification.



(a) ETT.



(b) ETTN.

Fig. 3. First non-dominated fronts of ETT and ETTN with corresponding solutions representing Triple features.

TABLE VI  
PAIRS OF TRIPLE FEATURES COMBINATIONS FROM ETT AND ETTN

| Triple feature pair   | ETT                       |
|-----------------------|---------------------------|
| $\Xi_1^{\text{ETT}}$  | $T_{12}D_4C_2, T_6D_3C_5$ |
| $\Xi_2^{\text{ETT}}$  | $T_{12}D_4C_2, T_0D_3C_1$ |
| $\Xi_3^{\text{ETT}}$  | $T_6D_3C_5, T_0D_3C_1$    |
| Triple feature pair   | ETTN                      |
| $\Xi_1^{\text{ETTN}}$ | $T_0D_5C_5, T_0D_3C_2$    |
| $\Xi_2^{\text{ETTN}}$ | $T_0D_5C_5, T_0D_1C_2$    |
| $\Xi_3^{\text{ETTN}}$ | $T_0D_3C_2, T_0D_1C_2$    |

#### A. Fish-94 Database

Fish database consists of 94 fish images [10], which are shown in Fig. 4. Several test objects are produced from each object in the database by performing a random rotation, scale and translation, with and without noise being added. To compare ETT and ETTN with the traditional Trace transform (TT), we perform another set of experiments using a setup

similar to Kadyrov and Petrou [10]. It should be noted that the authors in [10] use TT for image database retrieval with the best five results as query results. We shall take only the first best results in [10], which correspond to correct classification. Furthermore, features from the traditional TT used in [10] are normalized features that consist of a thousand Triple features denoted by  $\Xi$ . In our work, we create three pairs of features from the original Triple features in Table V so as to form three different 2D vectors tested individually and denoted by the form of  $\Xi_i^{method}$ , where  $i = 1, 2$  or  $3$  and  $method$  could be ETT or ETTN, as shown in Table VI. Henceforth, Triple features of the traditional Trace transform will be denoted by  $\Xi^{TT}$ .

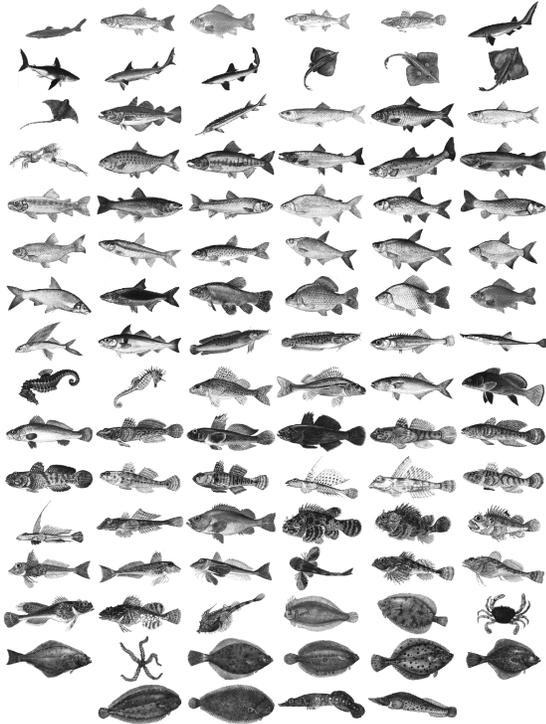


Fig. 4. Fish-94 database [10].

In our work we use a simple  $k$ -Nearest Neighbour ( $k$ -NN) classifier with  $k = 1$  and the Euclidean distance measure. For training purpose, a set of training images are used with a five-fold cross-validation. We generate for each image 44 RST transformations, i.e. one original image (of size 256 by 256), 24 rotation only [15-360°], 9 scale only [0.9-0.1] and 10 random translations. No images are generated using a combination of different transformation in the training samples.

Note that a separate classifier for each features ( $\Xi_1^{ETT}$  to  $\Xi_3^{ETT}$  and  $\Xi_1^{ETTN}$  to  $\Xi_3^{ETTN}$ ) will be trained. In this work, we investigated all features  $\Xi_1$  to  $\Xi_3$  of both ETT and ETTN. We will present results from  $\Xi_3^{ETT}$  and  $\Xi_3^{ETTN}$ , which correspond to the solutions around the knee point of the Pareto front.

The similarity measure used by Kadyrov and Petrou [10] involves scaling features obtained by the traditional TT so that it varies within the same range. Then, the absolute differences between the test sample and each of the objects in the database

TABLE VII  
CLASSIFICATION ACCURACY AND ROBUSTNESS TO SCALE ONLY  
(FISH-94 DATABASE)

| Scale factor | $\Xi^{TT}$ | $\Xi_3^{ETT}$ | $\Xi_3^{ETTN}$ |
|--------------|------------|---------------|----------------|
| 0.9          | 100.0      | 100.0         | 100.0          |
| 0.8          | 100.0      | 100.0         | 100.0          |
| 0.7          | 96.81      | 97.87         | 100.0          |
| 0.6          | 88.30      | 100.0         | 100.0          |
| 0.5          | 86.17      | 100.0         | 100.0          |
| 0.4          | 74.47      | 100.0         | 100.0          |
| 0.3          | 65.96      | 100.0         | 100.0          |
| 0.2          | 50.00      | 100.0         | 100.0          |
| 0.1          | 19.15      | 100.0         | 100.0          |

were computed.

1) *Robustness to Scale (Fish-94 Database)*: First, we test the robustness to scale using different scaling factors for each image in the Fish-94 database without involving other geometric transformation. We computed the classification accuracy for each approach according to the following equation:

$$\text{Accuracy}(\%) = \frac{\text{Number of correctly classified samples}}{\text{Number of all test samples}} \times 100$$

Classification results from TT, ETT and ETTN are shown in Table VII. We can see that TT performs reasonably well up to a scale level of 0.3, whereas both ETT and ETTN remain more stable.

2) *Robustness to Additive Noise (Fish-94 Database)*: To verify the robustness of the ETT and ETTN to additive noise, we considered two types of noise; Gaussian noise and salt & pepper noise. In both cases, noise is added to the whole image (i.e. on the object and the background), combined with random rotations and translations in addition to specific scaling factors from 1.0 to 0.3. The standard deviation of the zero-mean Gaussian varies from 2 to 10 with an increment by 2. We plot the classification accuracy over various scaling factors. Fig. 5 shows the percentage of the accuracy when the objects have been rotated and translated randomly and scaled in addition to the added Gaussian noise.

From Fig. 5(a), we can see that the classification accuracy degrades rapidly when the scaling factor is larger than 0.8 using the features extracted by the traditional TT. Fig. 5(b) indicates that the classification accuracy deteriorates using the features extracted by ETT, even when the noise level is low. By contrast, the classification accuracy shows a strong robustness to noise using the features extracted by ETTN, refer to Fig. 5(c). Note that, although features extracted by TT and ETT exhibit a good accuracy when the images are not contaminated by noise, the speed of losing robustness becomes very quick when noise level is increased. For instance, Fig. 5(b) indicates that the features perform very well in response to RST deformation without noise, the classification accuracy rapidly falls to about 20% when a small level of noise was added. In contrast, features from ETTN demonstrate a better accuracy as high as 97.9% when the scaling factor is low.

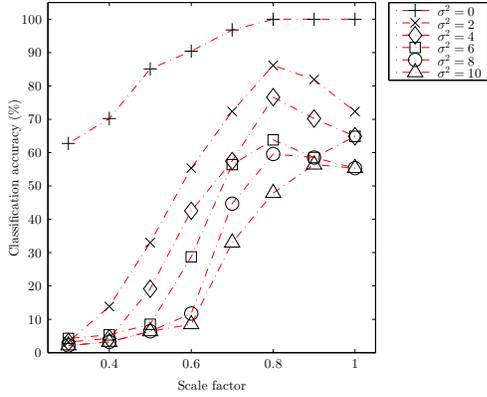
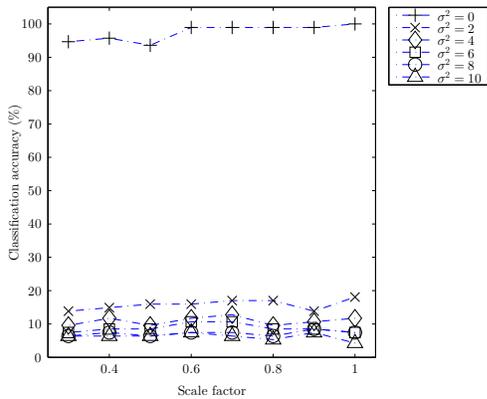
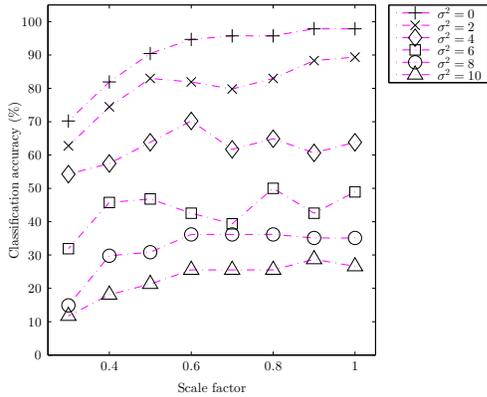
(a)  $\Xi^{TT}$  [10].(b)  $\Xi_3^{ETT}$ .(c)  $\Xi_3^{ETT}$ .

Fig. 5. Robustness of each approach to additive Gaussian noise of zero mean and variance  $\sigma^2$ , and random rotation, translation and specific scaling, (Fish-94 database).

Similarly, we test the robustness to salt & pepper noise where the noise is added to the whole image combined with random rotation and translation, and with a specific scaling factor from 1.0 to 0.3. We follow the same setup and noise level used in [10]. In particular, noise levels were chosen in such a way that when no RST transformations present, all objects could be identified correctly using the traditional TT.

Fig. 6 shows sample images when salt & pepper noise is added to the whole image with noise density 1%, 3%, 5% and 6%. It is obvious that adding noise to the whole image will result in a major distortion in the gray information as well as in the shape information of the object in addition to the fact that noise dominates many pixels along each tracing line computed in the transform while the object occupies a small area in the image.

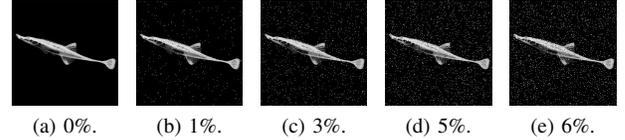


Fig. 6. Sample image from Fish-94 database with additive salt & pepper noise added to the whole image with noise density  $p = 0\%$ , 1%, 3%, 5% and 6%.

Fig. 7 shows the accuracy in percentage when the objects have been rotated and translated randomly, and scaled with a specific scaling factor from 1.0 to 0.3 with additive salt & pepper noise. Obviously, features extracted by the traditional TT show a relatively stable accuracy when no noise is present, see Fig. 7(a). However, the performance decreases rapidly when the scaling factor decrease below 0.7, although the noise level is as low as 1%, see Fig. 7(a). Also, the accuracy deteriorates more rapidly when the noise level increases. On the other hand, ETT features exhibit stronger robustness to RST without noise, see Fig. 7(b), with a slightly better robustness to noise up to level 2% with a classification accuracy around 60%, then it shows a rapid decrease of accuracy. Although features extracted by TT and ETT exhibit a very good accuracy without noise, their performance drastically degrades when the level of noise increases. For example, the top curve in Fig. 7(b) shows that the classification performance is robust to RST deformation when noise level is zero. The accuracy starts to drop once noise is added. For ETTN, Fig. 7(c) shows that features extracted by ETTN are less sensitive to the added noise.

### B. COIL-20 Database

The Columbia COIL-20 database [48] consists of 20 distinctive objects, each having 72 3D multi-views ( $5^\circ$  displacement). In this work, we also use the original 20 front view only images to generate 2D RST transformations and noisy images. Fig. 8 shows 20 original objects in the COIL-20 database used in the experiment.

Note that the Trace transforms used in this set of experiments are directly taken from Table VI, which are optimized using ETT and ETTN based on the low-resolution sample images generated from the fish database. The purpose of doing this is to check the generalization ability of the ETT and ETTN from one database to another without re-performing evolutionary optimization. To train new  $k$ -NN classifiers for the COIL-20 images, we generate samples in the same way as we did in the Fish-94 database, i.e., 24 rotated images, 9 scaled images and 10 random translated images for each class with a five-fold cross-validation.

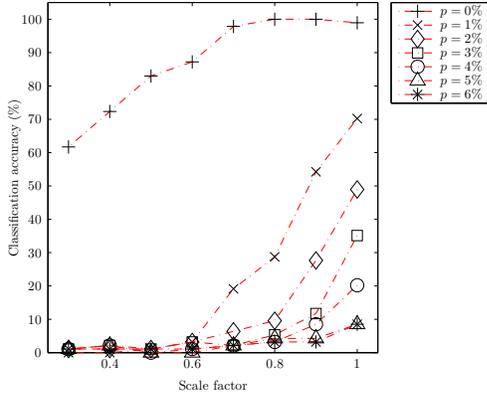
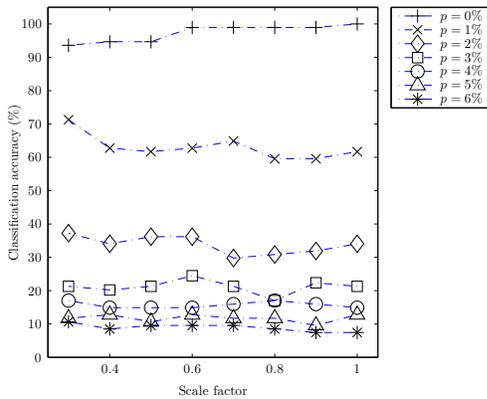
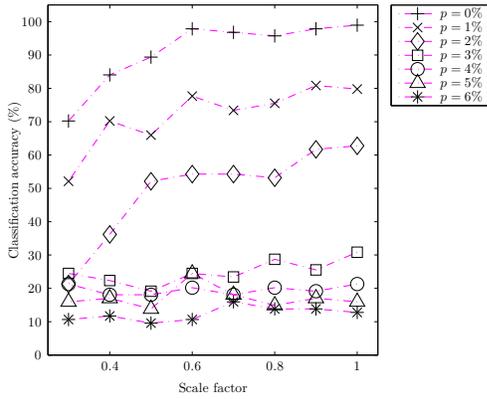

 (a)  $\Xi^{\text{TT}}$  [10].

 (b)  $\Xi_3^{\text{ETT}}$ .

 (c)  $\Xi_3^{\text{ETT}}$ .

Fig. 7. Robustness of each approach to additive salt & pepper noise with noise density  $p$  of affected pixels, and random rotation, translation and specific scaling, (Fish-94 database).

1) *Robustness to Scale (COIL-20 Database)*: In this experiment, images from COIL-20 database are transformed using a scaling factor varying from 1.0 to 0.3. Classification results listed in Table VIII indicate that, as expected, the Triple features from ETT and ETTN exhibit excellent discrimination ability despite the fact that these features are extracted by the Trace transforms optimized using low-resolution images in the



Fig. 8. COIL-20 database [48].

TABLE VIII  
CLASSIFICATION ACCURACY AND ROBUSTNESS TO SCALE ONLY  
(COIL-20 DATABASE)

| Scale factor | $\Xi_3^{\text{ETT}}$ | $\Xi_3^{\text{ETT}}$ |
|--------------|----------------------|----------------------|
| 0.9          | 100.0                | 100.0                |
| 0.8          | 100.0                | 100.0                |
| 0.7          | 100.0                | 100.0                |
| 0.6          | 100.0                | 100.0                |
| 0.5          | 100.0                | 100.0                |
| 0.4          | 100.0                | 100.0                |
| 0.3          | 100.0                | 100.0                |
| 0.2          | 100.0                | 100.0                |
| 0.1          | 100.0                | 100.0                |

fish database.

2) *Robustness to Additive Noise (COIL-20 Database)*: To verify the robustness to noise, similar experiments were performed on the COIL-20 images. Both Gaussian noise and salt & pepper noise are added to the whole image, to which RST transformations are also performed using a random rotation and translation, and a predefined scaling factor. Fig. 9 depicts classification results in percentage on the RST transformed images with Gaussian noise being added using the features extracted by ETT and ETTN, respectively. It can be seen that ETTN exhibits better robustness to noise compared to ETT. Likewise, the robustness to salt & pepper noise is shown in Fig. 10, where ETTN shows a better performance.

Taking a closer look at Trace functionals ( $T$ ) in ETTN, one can notice that this functional is  $T_0$  in all  $\Xi_i^{\text{ETT}}$ , which is actually the Radon transform. This is consistent with the observations previous findings suggesting that Radon transform is robust to noise.

## V. COMPUTATIONAL COMPLEXITY OF TRACE TRANSFORM

The computational complexity of constructing Triple features depends on several parameters. Among them, the number of different functionals utilized is dominating in the computation time although some functionals can be executed in parallel. The sampling frequency of pixels along each

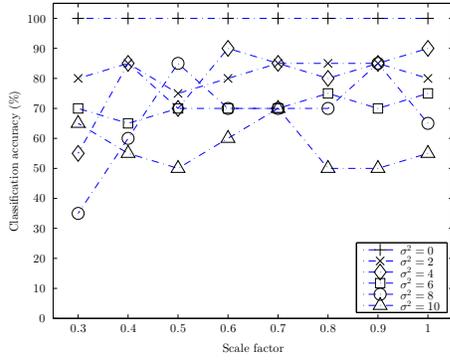
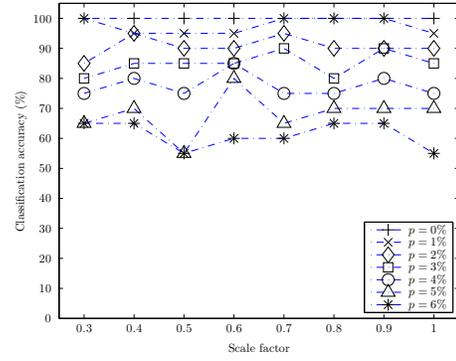
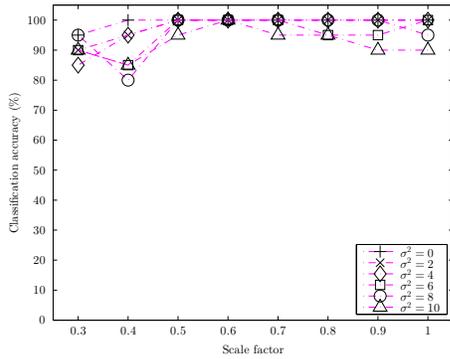
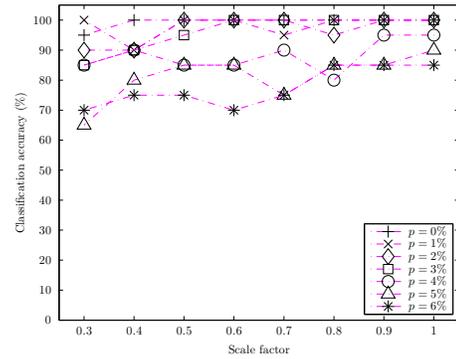
(a)  $\Xi_3^{\text{ETT}}$ .(a)  $\Xi_3^{\text{ETT}}$ (b)  $\Xi_3^{\text{ETTn}}$ .(b)  $\Xi_3^{\text{ETTn}}$ 

Fig. 9. Robustness of ETT and ETTN to Gaussian noise of zero mean and variance  $\sigma^2$ , and random rotation, translation and specific scaling, (COIL-20 database).

Fig. 10. Robustness of ETT and ETTN to additive salt & pepper noise with noise density  $p$ , and random rotation, translation and specific scaling, (COIL-20 Database).

tracing line would also affect the computational complexity for construction of Triple features. Table IX shows the parameters that are involved in calculating the computational cost of Triple features using Trace transform [10], refer also to Figs. 1 and 2. In our work, the number of rotation angles  $n_\theta$  is set to 180 with a fixed step angle of  $1^\circ$ . A more coarse sampling may be considered at the cost of the amount of information extracted from the image, i.e. the accuracy of the transform. Some functionals do not necessarily require  $360^\circ$  as the tracing lines up to  $180^\circ$  rotation is sufficient. Considering an image of  $N \times N$  pixels, the largest number of sampling lines per rotation angle equals the number of pixels on the image diagonal, i.e., the maximum value of  $n_\rho$  is equal to  $\sqrt{2}N$  with a step of a single pixel on the each line. This results in a Trace matrix of size  $n_\theta \times n_\rho$ . Also, we consider that each pixel along the tracing line is sampled with a maximum number of points  $n_t$  equal to the number of pixels on the diagonal of the image, i.e.  $\sqrt{2}N$ . Therefore, we can calculate the sampling rates ( $n_\theta, n_\rho$  and  $n_t$ ) for a given image. Note that the pixel locations on each tracing line are computed in advance and stored in a file.

To compute the Trace transform, there are  $n_t$  samples on each  $n_\rho$  lines for every  $n_\theta$  rotation, refer to Table IX. Therefore,  $O_T N_T n_t n_\rho n_\theta$  operations are required to compute  $N_T$  transforms. To construct Triple features using  $N_D$  Diametric functionals,  $O_D N_D N_T n_\rho n_\theta$  operations are needed to con-

struct  $N_D N_T$  Diametric of size  $n_\theta$ . Finally,  $O_C N_C N_D N_T n_\theta$  operations are required to produce  $N_C N_D N_T$  Triple features. Hence, the total number of operations equals to  $(O_T N_T n_t n_\rho n_\theta / 2 + O_D N_D N_T n_\rho n_\theta + O_C N_C N_D N_T n_\theta)$ . For  $n_\theta = 180$  and  $n_\rho = n_t = \sqrt{2}N$  (pixel increment by 1), then the number of operations equals to  $(O_T 180 N^2 N_T + O_D 180 \sqrt{2} N N_D N_T + O_C 180 N_C N_D N_T)$  operations. From the above analysis, we can conclude that calculating the Trace functionals has a computational complexity of at most  $180 N^2 N_T$ . In our approach, only  $T_0$  as the Trace functional is used in ETTN, which is the summation line of image pixels (i.e. Radon Transform). The complexity of Radon transform is  $O(N^2 n_\theta)$  [8], [50]. Therefore  $N_T = 1$ , and the computational complexity of ETTN equals to  $O(N^2 n_\theta)$ . Moreover, there are only a few Triple features in ETTN, while TT needs a thousand Triple features calculated using  $n_t = n_\rho = n_\theta = 10^2$  ( $\theta$  increments by 5 for  $360^\circ$ ) and  $N_T = N_D = N_C = 10$  [10]. Therefore TT requires approximately  $10^7$  operations for images of a size  $200 \times 400$  pixels in the Fish-94 database. The computational efficiency of the evolutionary optimization of the Trace transform was enhanced using three strategies in this work. First, the sampling parameters and pixel locations that form each tracing line in the image were computed offline, which significantly reduces the computational time. Second, due to the symmetry of the Trace functionals, the transform is performed over the range of  $0 \leq \theta \leq \pi$ , instead

TABLE IX  
COMPUTATIONAL PARAMETERS OF TRACE TRANSFORM

| Parameter  | Description  |
|------------|--|
| $N_T$      | no. of Trace functionals                                 |
| $N_D$      | no. of Diametrical functionals                           |
| $N_C$      | no. of Circus functionals                                |
| $n_t$      | no. of points along the tracing line                     |
| $n_\rho$   | no. of points along parameter $\rho$                     |
| $n_\theta$ | no. of rotations (angles) to consider for $\theta$       |
| $O_T$      | no. of operations per sample for Trace functionals       |
| $O_D$      | no. of operations per sample for Diametrical functionals |
| $O_C$      | no. of operations per sample for Circus functionals      |

TABLE X  
COMPUTATIONAL TIME IN SEC. OF SOLUTIONS FROM ETT AND ETTN

| Solution No. | ETT   | ETTN  |
|--------------|-------|-------|
| $s_1$        | 2.646 | 0.962 |
| $s_2$        | 1.438 | 0.999 |
| $s_3$        | 1.878 | 0.995 |

of from  $0 \leq \theta \leq 2\pi$ . Finally, while the traditional Trace transform needs a large number of Triple features, which must be manually chosen, the proposed method requires only a very small number of Triple features owing to the optimized Trace functions that can extract distinctive features that are robust to a wide range of geometrical transformations as well as different types and levels of additive noise.

The estimate the computational time for obtaining each Triple feature in ETT and ETTN, e.g. solutions  $s_1$ ,  $s_2$ , and  $s_3$  in Table V, we performed 10 runs for each solution using a standard PC with Intel Core 2 Duo E8500 3.1GHz processor and 3GB of RAM. Table X lists the average computational time in seconds for the three solutions. As we can see, all the Triple feature used in this work take about one to three seconds to calculate.

## VI. CONCLUSIONS

In this paper, we exploited the Pareto optimality approach to optimize the functionals used in the Trace transform for extracting image features that are robust to RST distortions in the presence of noise. The between-class variance and within-class variance in feature distribution in a 2D space are used as two objectives to be optimized using the elitist non-dominated sorting genetic algorithm (NSGA-II). To investigate the effectiveness of generating features that are robust not only to RST transformations but also to noise, two slightly different approaches have been adopted. The first approach performs evolutionary optimization of the Trace transforms using low-resolution images with RST distortions only (termed ETT), whilst the other uses low-resolution RST images with additive noise (ETTN). Experimental results on two image databases demonstrated that both ETT and ETTN can extract features robust to RST deformation. However, robustness to noise can

be gleaned only if the Trace transforms are optimized when the sample images used in the evolutionary optimization are also subject to noise. This is consistent with the findings in evolutionary robust optimization [51], which suggest that robust optimal solutions can be obtained by injecting noise in evaluating the fitness of candidate designs. It was also shown that the ETT and ETTN exhibit good generalization ability, as the Trace transforms optimized using low-resolution images from one image database can perform well on images of a higher resolution from the same database, and from a different database having very different objects in the images as well.

Future work includes the test of the ETTN on more image datasets to verify its generalization ability. In the present work, the  $k$ -NN classifier was used for comparing the performance of the extracted Trace features, which performed fairly poorly when the noise level increases. We will design more effective classifiers, e.g., classifier ensembles that can take advantage of the multiple features extracted by ETTN. We expect that such classifier ensembles will significantly improve the classification performance in the presence of strong noise.

## REFERENCES

- [1] M. Alghoniemy and A. H. Tewfik, "Geometric Invariance in image watermarking," *IEEE Transactions on Image Processing*, vol. 13, no. 2, pp. 145–153, 2004.
- [2] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.
- [3] S. Li, M.-C. Lee, and C.-M. Pun, "Complex Zernike Moments Features for Shape-Based Image Retrieval," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 39, no. 1, pp. 227–237, Jan. 2009.
- [4] A. Amanatiadis, V. Kaburlasos, A. Gasteratos, and S. Papadakis, "Evaluation of shape descriptors for shape-based image retrieval," *Image Processing, IET*, vol. 5, no. 5, pp. 493–499, August 2011.
- [5] H. Rai, X. Shen, K. Deepak, and P. Krishna, "Hybrid feature to encode shape and texture for content based image retrieval," in *2011 International Conference on Image Information Processing (ICIIP)*, Nov. 2011, pp. 1–6.
- [6] S. R. Deans, *The Radon Transform and Some of Its Applications*. Krieger Publishing Company, 1983.
- [7] G. Y. Chen and B. Kegl, "Feature extraction using Radon, wavelet and fourier transform," in *IEEE International Conference on Systems, Man and Cybernetics, ISIC.*, 2007, pp. 1020–1025.
- [8] S. Tabbone, L. Wendling, and J.-P. Salmon, "A new shape descriptor defined on the radon transform," *Computer Vision and Image Understanding*, vol. 102, no. 1, pp. 42–51, 2006.
- [9] M. Hasegawa and S. Tabbone, "Amplitude-only log radon transform for geometric invariant shape descriptor," *Pattern Recognition*, no. 0, pp. –, 2013.
- [10] A. Kadyrov and M. Petrou, "The trace transform and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 811–828, 2001.
- [11] A. Kadyrov, A. Talepbour, and M. Petrou, "Texture Classification With Thousands of Features," in *13th British Machine Vision Conference. BMVC, 2-5 September 2002*, pp. 656–665.
- [12] B.-S. Shin, E.-Y. Cha, K.-B. Kim, K.-W. Cho, R. Klette, and Y. W. Woo, "Effective feature extraction by trace transform for insect footprint recognition," in *3rd International Conference on Bio-Inspired Computing: Theories and Applications, 2008. BICTA 2008.*, 2008, pp. 97–102.
- [13] M. F. Nasrudin, M. Petrou, and L. Kotoulas, "Jawi Character Recognition Using the Trace Transform," in *Seventh International Conference on Computer Graphics, Imaging and Visualization (CGIV)*, 2010, pp. 151–156.
- [14] A. Kadyrov and M. Petrou, "The trace transform as a tool to invariant feature construction," in *Fourteenth International Conference on Pattern Recognition, 1998. Proceedings.*, vol. 2, 1998, pp. 1037–1039 o.2.
- [15] M. Petrou and A. Kadyrov, "Features invariant to affine distortions from the trace transform," in *Proceedings. International Conference on Image Processing, 2001.*, vol. 3, 2001, pp. 852–855 o.3.

- [16] A. Kadyrov and M. Petrou, "Affine parameter estimation from the trace transform," in *16th International Conference on Pattern Recognition, 2002. Proceedings.*, vol. 2, 2002, pp. 798–801 o.2.
- [17] —, "Object signatures invariant to affine distortions derived from the trace transform," *Image and Vision Computing*, vol. 21, no. 13-14, pp. 1135–1143, 12/1 2003.
- [18] M. Petrou and A. Kadyrov, "Affine invariant features from the trace transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 30–44, 2004.
- [19] J. Flusser and T. Suk, "Affine moment invariants: a new tool for character recognition," *Pattern Recognition Letters*, vol. 15, no. 4, pp. 433–436, 4 1994.
- [20] S. Srisuk, R. Fooprateepsiri, M. Petrou, S. Waraklang, and K. Sunat, "A general framework for image retrieval using reinforcement learning," in *The Image and Vision Computing 2003*, Nov. 26-28 2003, pp. 36–41.
- [21] D. Siskovicova, J. Turan, and Z. Bojkovic, "Invariant image recognition using trace transform and function of autocorrelation," in *The International Conference on Computer as a Tool, 2005. EUROCON 2005.*, vol. 1, 2005, pp. 187–190.
- [22] J. Turan, Z. Bojkovic, P. Filo, and L. Ovsenik, "Invariant image recognition experiment with trace transform," in *7th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services, 2005.*, vol. 1, 2005, pp. 189–192 o. 1.
- [23] P. Brasnett and M. Bober, "Fast and robust image identification," in *19th International Conference on Pattern Recognition, ICPR 2008.*, Dec. 2008, pp. 1–5.
- [24] W. A. Albukhanajer, Y. Jin, J. A. Briffa, and G. Williams, "Evolutionary Multi-Objective Optimization of Trace Transform for Invariant Feature Extraction," in *2012 IEEE Congress on Evolutionary Computation (CEC)*, Brisbane, Australia, 10-15, June 2012, pp. 401–408.
- [25] —, "A comparative study of multi-objective evolutionary trace transform methods for robust feature extraction," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, R. Purshouse, P. Fleming, C. Fonseca, S. Greco, and J. Shaw, Eds. Springer Berlin Heidelberg, 2013, vol. 7811, pp. 573–586.
- [26] A. Frias-Velazquez, C. Ortiz, A. Pizurica, W. Philips, and G. Cerda, "Object identification by using orthonormal circus functions from the trace transform," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 2153–2156.
- [27] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering.*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.
- [28] K. Neshatian, M. Zhang, and P. Andreae, "A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 5, pp. 645–661, Oct 2012.
- [29] D. Atkins, K. Neshatian, and M. Zhang, "A domain independent genetic programming approach to automatic feature extraction for image classification," in *2011 IEEE Congress on Evolutionary Computation (CEC)*, June 2011, pp. 238–245.
- [30] H. Al-Sahaf, K. Neshatian, and M. Zhang, "Automatic feature extraction and image classification using genetic programming," in *2011 5th International Conference on Automation, Robotics and Applications (ICARA)*, Dec 2011, pp. 157–162.
- [31] H. Al-Sahaf, A. Song, K. Neshatian, and M. Zhang, "Extracting image features for classification by two-tier genetic programming," in *2012 IEEE Congress on Evolutionary Computation (CEC)*, June 2012, pp. 1–8.
- [32] F. Pettersson, N. Chakraborti, and H. Saxén, "A genetic algorithms based multi-objective neural net applied to noisy blast furnace data," *Applied Soft Computing*, vol. 7, no. 1, pp. 387 – 397, 2007.
- [33] B. K. Giri, F. Pettersson, H. Saxén, and N. Chakraborti, "Genetic programming evolved through bi-objective genetic algorithms applied to a blast furnace," *Materials and Manufacturing Processes*, vol. 28, no. 7, pp. 776–782, 2013.
- [34] B. K. Giri, J. Hakanen, K. Miettinen, and N. Chakraborti, "Genetic programming through bi-objective genetic algorithms with a study of a simulated moving bed process involving multiple objectives," *Applied Soft Computing*, vol. 13, no. 5, pp. 2613 – 2623, 2013.
- [35] W. A. Albukhanajer, H. A. Lefta, and A. A. Ali, "Effective identification of a turbogenerator in a smib power system using fuzzy neural networks," in *2014 International Joint Conference on Neural Networks (IJCNN)*, Beijing, China, July 2014, pp. 2804–2811.
- [36] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 28, no. 4, pp. 594–611, April 2006.
- [37] H. Al-Sahaf, M. Zhang, and M. Johnston, "A one-shot learning approach to image classification using genetic programming," in *AI 2013: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, S. Cranefield and A. Nayak, Eds. Springer International Publishing, 2013, vol. 8272, pp. 110–122.
- [38] P. Brasnett and M. Z. Bober, "A Robust Visual Identifier Using the Trace Transform," in *Visual Information Engineering Conference (VIE 2007)*, July 2007, pp. 25–27.
- [39] K. Murphy, A. Torralba, D. Eaton, and W. Freeman, "Object Detection and Localization Using Local and Global Features," in *Toward Category-Level Object Recognition*, ser. Lecture Notes in Computer Science, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Eds. Springer Berlin / Heidelberg, 2006, vol. 4170, pp. 382–400.
- [40] M. Petrou and F. Wang, *A Tutorial on the Practical Implementation of the Trace Transform*, ser. Handbook of Texture Analysis. Imperial College Press, 2008, pp. 313–346.
- [41] N. Srinivas and K. Deb, "Multiobjective Optimization using Nondominated Sorting in Genetic Algorithms," *Evol.Comput.*, vol. 2, no. 3, pp. 221–248, September 1994.
- [42] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, 1st ed. England: John Wiley & Sons. Ltd, 2002.
- [43] C. Smith and Y. Jin, "Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction," *Neurocomputing*, 2014.
- [44] Y. Jin and B. Sendhoff, "Pareto-based multiobjective machine learning: An overview and case studies," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, no. 3, pp. 397–415, 2008.
- [45] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [46] E. Zitzler and L. Thiele, "An evolutionary algorithm for multiobjective optimization: The strength Pareto approach," *Computer Engineering and Communication Networks Lab (TIK)*, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland, Tech. Rep. 43, 1998.
- [47] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 23, no. 2, pp. 228–233, 2001.
- [48] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library (COIL-20)," Columbia University, Tech. Rep. CUCS-005-96, February 1996.
- [49] C. Igel, V. Heidrich-Meisner, and T. Glasmachers, "Shark," *Journal of Machine Learning Research*, vol. 9, pp. 993–996, 2008.
- [50] W. Gotz and H. Druckmuller, "A fast digital radon transform an efficient means for evaluating the hough transform," *Pattern Recognition*, vol. 29, no. 4, pp. 711 – 718, 1996.
- [51] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments - A survey," *IEEE Transactions on Evolutionary Computation.*, vol. 9, no. 3, pp. 303–317, June 2005.