Information Theoretic Shape Matching

Erion Hasanbelliu, Luis Sanchez Giraldo, and José C. Príncipe, *Fellow, IEEE*

Abstract—In this paper, we describe two related algorithms that provide both rigid and non-rigid point set registration with different computational complexity and accuracy. The first algorithm utilizes a nonlinear similarity measure known as correntropy. The measure combines second and high order moments in its decision statistic showing improvements especially in the presence of impulsive noise. The algorithm assumes that the correspondence between the point sets is known, which is determined with the surprise metric. The second algorithm mitigates the need to establish a correspondence by representing the point sets as probability density functions (PDF). The registration problem is then treated as a distribution alignment. The method utilizes the Cauchy-Schwarz divergence to measure the similarity/distance between the point sets and recover the spatial transformation function needed to register them. Both algorithms utilize information theoretic descriptors: however, correntropy works at the realizations level, whereas Cauchy-Schwarz divergence works at the PDF level. This allows correntropy to be less computationally expensive, and for correct correspondence, more accurate. The two algorithms are robust against noise and outliers and perform well under varying levels of distortion. They outperform several well-known and state-of-the-art methods for point set registration.

Index Terms—Information theoretic learning, Cauchy-Schwarz divergence, correntropy, non-rigid registration, shape matching, surprise, annealing

1 Introduction

Shape matching is a central problem in computer vision, pattern recognition, medical imaging, robotics, and many other fields. Tasks such as content-based image retrieval, face recognition, object tracking, and image registration all require matching of features. Features such as points, lines, and contours are extracted from the image of interest and matched against features of another image (e.g., image registration) or of a template (e.g., object tracking).

Point matching comprises two subtasks: 1) establish the correspondence between the points in the two shapes/images, and 2) retrieve the spatial transformation/mapping that aligns the two objects. Each of these subtasks is easily solved once the solution to the other is known. The difficulty arises when attempting to solve them concurrently. Belongie et al. [1] and Jain and Zhang [2] provide shape descriptors to facilitate the correspondence problem. Shape context describes the distribution of the shape with respect to each point on the shape. Finding the correspondence between two shapes is then equivalent to finding the point in each object with a similar shape context.

The iterative closest point (ICP) algorithm [3] is the most popular method for point set registration. At each step, the algorithm utilizes the nearest-neighbor relationship to assign binary correspondence and then finds the least squares transformation relating the point sets based on the current estimate of the correspondence. The algorithm continues this iterative process until it reaches a local minimum. The method is very simple, but because its cost function is not differentiable, it exhibits local convergence and requires an initialization in the neighborhood of the optimum. With an adequate set of initial positions it can perform any rigid transformation. The method, though, suffers in the presence of outliers and is not suitable for non-rigid transformations.

Robust point matching (RPM) [4] improves upon ICP by employing a global-to-local search and soft assignment for the correspondence. The method jointly determines the correspondence and the transformation between the two point-sets via deterministic annealing and soft-assignment. The soft-assignment relaxes the one-to-one correspondence between the two point sets and the fuzzy correspondence allows for a gradual improvement without jumps in the space of binary correspondences. However, the method is not stable in the presence of outliers. Also, due to the soft-assignment approach, the complexity of the method is high.

In [5], Zheng and Doermann formulate point matching as an optimization problem to preserve local neighborhood structures. Their method is based on the fact that local relationship among neighboring points is very important for nonrigid shapes and it is generally well preserved due to physical constraints. Their approach uses graph matching, where two neighboring points are represented by an edge, and the optimal match between two graphs is the one that maximizes the number of matched edges.

These methods assume that the point sets provide a good representation of their respective objects. In most real world applications, extracting the contour of an object is quite difficult, and the point sets might be very noisy and/or have occlusions and outliers. Therefore, we can expect that measures based on second order statistics will perform poorly. Our first algorithm improves upon these methods by using a generalized correlation measure named *correntropy* [6]. Correntropy is directly related to the probability of how similar two random variables are, as estimated by kernels

• *The authors are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611.*

Manuscript received 29 Aug. 2012; revised 16 Dec. 2013; accepted 1 Apr. 2014. Date of publication 15 May 2014; date of current version 5 Nov. 2014. Recommended for acceptance by D. Forsyth.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2014.2324585

placed in the bisector neighborhood of the joint space controlled by the kernel bandwidth [7], and yields solutions that are more accurate in non-Gaussian and nonlinear signal processing. The method is robust to impulsive noise but assumes that the correspondence between the two point sets is known a priori. Therefore, the shape correspondence is determined using surprise [8], [9], which uses shape information to determine the correspondence. One of the appeals of this algorithm is that it is computationally efficient, with a complexity of $O(N)$.

To eliminate the point correspondence requirement, a global approach is taken where the point sets are represented as probability density functions (PDF). Tsin and Kanade [10] were the first to propose such a method. They modeled the two point sets as kernel density functions and evaluated their similarity and transformation updates using the kernel correlation between the two density estimates. Glaunes et al. [11] match the two point sets by representing them as weighted sums of Dirac delta functions. They use Gaussian functions to 'soften' the Dirac delta functions and use diffeomorphic transformations to minimize the distance between the two distributions.

Jian and Vemuri [12], [13] extended this approach by representing the densities as Gaussian mixture models (GMM). They derive a closed-form expression to compute the L_2 distance between the two Gaussian mixtures and to align the point sets. They also use thin-plate splines (TPS) to parameterize the smooth non-linear transformation.

Myronenko et al. also introduced a probabilistic method for point matching called coherent point drift (CPD) [14]. A GMM is fit to one of the point sets using the other data set's points as initial position for the Gaussian centroids. In addition, they do not make an assumption of the transformation model as in [12], [15] where TPS was considered. Instead, a motion coherence constraint is imposed over the velocity field of the motion of the Gaussian centroids. The smoothness on the underlying transformation is imposed based on the motion coherence theory [16] and results in a Gaussian radial basis function (GBRF).

Our second algorithm also represents the two point sets as density functions. The similarity between the two PDFs is measured using an information theoretic measure, the Cauchy-Schwarz divergence [17]. The advantage is that this divergence method compares directly PDF similarity, and it is still expressed in terms of inner products of the two PDFs which essentially estimates a normalized Rényi's quadratic cross-entropy, where the important term is what we call the *cross information potential (IP)* between the two densities [17], [18]. This term is crucial in determining the similarity between the two sets because it measures the interaction of the field created by one of the point sets on the locations specified by the other. The Cauchy-Schwarz information potential field exerts information forces on samples of the second point set forcing them to move toward a path that will provide the most similarity between the two PDFs. The Cauchy-Schwarz divergence also has a closed-form expression in terms of convolutions operators compared to other similarity measures such as Kullback-Leibler divergence, which is difficult to estimate.

In addition, our methods share the same kernel estimators and we use kernel bandwidth annealing to speed-up the convergence and improve their accuracy. Controlling the kernel bandwidth in correntropy allows us to adjust the "observation window" in which similarity is assessed resulting in an effective mechanism to eliminate the detrimental effect of outliers. Focusing on the useful data points also allows for a quicker convergence. The Cauchy-Schwarz divergence uses a kernel PDF estimation, where the kernel bandwidth, again, controls the interaction field created by the PDFs and thus the forces that will bring the shapes together. Controlling the bandwidth allows for a global-to-local alignment between the shapes and a faster convergence rate.

The rest of the paper is organized as follows. Section 2 describes the point matching process: affine transformation, non-rigid deformations using either thin-plate splines or Gaussian radial basis functions, and the corresponding regularizations. Section 3 provides a brief backgroung on information theoretic learning. Section 4 describes the correntropy similarity measure, the algorithm, and a method to compute point correspondence. Section 5 describes the Cauchy-Schwarz divergence and the derived algorithm. Section 6 analyzes the kernel bandwidth role in our algorithms, the effect of the non-rigid penalization parameter, and the choice of TPS or GRBF for the non-rigid transformation. Section 7 provides a set of experiments comparing our methods against some well-known and state-of-the-art methods. Finally, Section 8 concludes with a discussion and conclusion of this work.

2 Point Matching

Suppose we have two point sets $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ and $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$, where $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^d$. To align point-set $\mathbf{X}$ to point-set $\mathbf{Y}$, $\mathbf{X}$ has to undergo some transformation f which maps a point $\mathbf{x}_i$ onto a new location $\hat{\mathbf{x}}_i = f(\mathbf{x}_i)$, where $\mathbf{y}_i = \hat{\mathbf{x}}_i + \epsilon_i$. This then becomes the classic regression problem

$$\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon_i, \quad i = 1, \ldots, n, \tag{1}$$

where f represents the regression model and ϵ_i are the random errors. Recovering the function f that closely matches $\mathbf{x}_i$ to $\mathbf{y}_i$ is an ill-posed problem because there exists an infinite number of functions f that could satisfy our goal. To select a particular solution we need a priori knowledge on the function. We know that the transformation can be broken down into two parts: 1) an *affine* transformation, which includes the major differences between the two point sets, global and linear transformations such as rotation, scaling, shear and translation, and 2) a *nonrigid* transformation, which includes the local deformations that cannot be expressed by the affine transformation. In addition, we need to assume that the function is *smooth* which ensures that two similar inputs correspond to two similar outputs. This is crucial when dealing with non-rigid transformations. We enforce smoothness by introducing a regularization term.

2.1 Affine Transformation

A basic transformation function involving only rotation and translation applied to a point $\mathbf{x}$ can be written as follows:

$$f(\mathbf{x},\theta,\mathbf{t}) = R(\theta)\mathbf{x}+\mathbf{t}, \text{where } \mathrm{R}(\theta) = \begin{bmatrix} cos\,(\theta) & -sin\,(\theta) \\ sin\,(\theta) & cos\,(\theta) \end{bmatrix}, \tag{2}$$

where θ represent the rotation angle, $R(\theta)$ represents the rotation matrix for 2D data samples, and $\mathbf{t}$ represents the translation vector. To represent the translation vector as part of the affine transformation matrix, we consider homogeneous coordinates where the original vector points are extended to $(d+1)$ dimensions with the last element being one (e.g., $[\mathbf{x}_1,\mathbf{x}_2,1]^T$). In addition, to account for any affine transformation, not just rotation, we consider a matrix $\mathbf{A}$ of $(d+1)\times(d+1)$ dimensions and represent the affine transformation function as $f(\mathbf{x},\mathbf{A}) = \mathbf{A}\mathbf{x}$.

2.2 Non-Rigid Transformation

The most popular non-rigid transformation models utilize the radial basis functions (RBF), where the transformation is defined as a linear combination of the basis functions as

$$f(\mathbf{x},\mathbf{W},\Phi) = \sum_{i=1}^{N} \mathbf{w}_i\phi(\|\mathbf{x}-\mathbf{x}_i\|), \tag{3}$$

where $\mathbf{w}_i$ are unknown parameters, and ϕ represent the basis functions which depend on the euclidean distance between a point $\mathbf{x}$ and the control points $\mathbf{x}_i$. The most common RBFs used for non-rigid transformations are thin plate splines and Gaussian RBFs.

Thin-plate splines are defined as: $\phi_i = \|\mathbf{x}-\mathbf{x}_i\|^2 \log\|\mathbf{x}-\mathbf{x}_i\|$ in 2D, $\phi_i = \|\mathbf{x}-\mathbf{x}_i\|$ in 3D, and they do not exist in 4D or higher dimensions [19]. The function ϕ is a solution to the squared Laplacian equation $\Delta^2\phi \propto \delta_{0,0}$ [20], which means that it is proportional to the "generalized function" $\delta_{0,0}$ meaning zero everywhere except at the origin but also having an integral equal to 1.

The main advantage of TPS is that it allows for an explicit decomposition into linear and non-linear parts [4]. However, TPS provides global support where each control point has a global influence on the overall transformation, i.e., a small perturbation on one of the control points always affects the coefficients corresponding to all the other control points. This does not allow TPS to model complex localized transformations.

Gaussian radial basis function uses a Gaussian kernel for the basis function ϕ as

$$\phi_i = \exp\left(\frac{-\|\mathbf{x}-\mathbf{x}_i\|}{\beta}\right), \tag{4}$$

where β is a fixed positive parameter. GRBF, due to its kernel bandwidth β, localizes the support. This makes the GRBF advantageous over TPS as the locality of the spatial smoothness is controlled by changing the value of the kernel bandwidth. In addition, GRBF generalizes to any dimensions compared to TPS which is defined only for two and three dimensions.

2.3 Regularization

Constraining the solution with a *stabilizer* is an important part of shape matching because it enforces *smoothness* on the transformation function. Since we are dealing with a finite set of points, estimating the non-rigid transformation does not provide a unique solution. There is an infinite number of transformations that would match the corresponding points but will have very different behavior in the rest of the shape contour. Imposing the requirement of choosing the smoothest transformation function, which is controlled by a regularization term, will provide a unique solution.

The regularization theory originates from the work of Aresenin and Tikhonov [21] where the existing optimization problem is augmented with a regularization term. In our case, we have:

$$min\sum_{i=1}^{N}(\mathbf{y}_i - f(\mathbf{x}_i))^2 + \lambda R(f), \tag{5}$$

where $\sum_{i=1}^{N}(\mathbf{y}_i - f(\mathbf{x}_i))^2$ represents the *fidelity* of the approximation, $R(f)$ is the regularization term representing the constraint on the *smoothness* of function f, and λ is a free parameter that represents the tradeoff between the proximity of f to the solution and its smoothness.

2.3.1 Thin-Plate Spline

The regularization term, $R(f)$, for thin-plate splines is defined as [20], [22]:

$$R(f) = \sum_{\alpha_1,\alpha_d=1}^{2}\int_{\mathbb{R}^d}\left(\frac{\partial^2 f}{\partial x_1,\ldots,\partial x_d}\right)^2\prod_{i=1}^{d}dx_i. \tag{6}$$

TPS penalizes the squared second order derivative, and its null space includes the affine transformations.

2.3.2 Gaussian Radial Basis Function

The regularization term for Gaussian RBF is defined as:

$$\mathcal{R}(f) = \int_{\mathbb{R}^d}\sum_{m=0}^{\infty} c_m\|D^m v(\mathbf{x})\|)^2 d\mathbf{x}, \tag{7}$$

where D is a derivative operator such that $D^{2l} = \nabla^{2l}v$ (scalar operator) and $D^{2l+1} = \nabla(\nabla^{2l}v)$ (vector operator) [23]. The spectral analog is of the form:

$$\mathcal{R}(f) = \int_{\mathbb{R}^D}\frac{|\tilde{f}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})}d\mathbf{s}, \tag{8}$$

where G is a positive definite function [14], [24]. The Gaussian function was selected for G because of its properties of being symmetric and positive definite, which simplify these calculations. To select a *smooth* function, we need to simply look at the frequency domain derivation. A smooth function is one that has less energy at high frequencies. The spectral definition of the Gaussian regularization term computes the power at each frequency by taking the L_2 norm and then passes it through a high-pass filter, $\frac{1}{\tilde{G}}$, which attenuates all low frequencies. Decreasing the value of the regularization parameter $R(f)$ is then equivalent to reducing the energy at high frequencies. Note that $\tilde{G}(\mathbf{s})$ is a Gaussian defined as $\tilde{G}(\mathbf{s}) = \exp(\frac{-\|\mathbf{s}\|^2}{\beta})$ where β is a fixed positive parameter whose value determines the level of *smoothness*.

3 Brief Background on Information Theoretic Learning

Both similarity measures that we use in this paper involve information theoretic descriptors. Therefore, we provide a brief review of some core mathematical concepts from information theory developed by Alfréd Rényi. He defined a family of entropies as:

$$H_\alpha(X) = \frac{1}{1-\alpha}\log\sum_{x_i} p^\alpha(x_i), \tag{9}$$

where α is a free parameter. Rényi's entropy is a generalization of Shannon's entropy [25]. For $\alpha \to 1$ Rényi's entropy approximates Shannon's entropy [26]. The main advantage of Rényi's definition of entropy is that the logarithm appears outside the sum which separates the effect of the logarithm from the argument of entropy and allows the estimation of entropy non-parametrically from pairwise sample differences. For example, for $\alpha = 2$ we have Rényi's quadratic entropy of a continuous random variable X defined as

$$H_2(X) = -\log\int p^2(x)\,dx. \tag{10}$$

Kernel (Parzen) estimate of the PDF [27] uses a suitable kernel function $\kappa(\ )$ to compute the approximate density values $p(x)$ of an arbitrary point x as $\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N}\kappa(\frac{x-x_i}{\sigma})$, where σ is the bandwidth parameter. If we consider the Gaussian kernel, $G_\sigma(x,x_i) = \frac{1}{\sqrt{2\pi}\sigma}\exp(\frac{-\|x-x_i\|^2}{2\sigma^2})$, for the PDF estimate and substitute it in the definition of Rényi's quadratic entropy, we have a closed form solution:

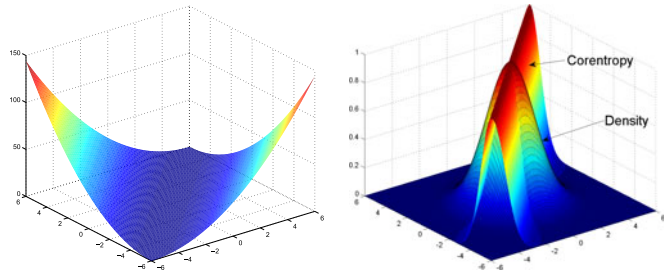
$$H_2(X) = -\log\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}G_{\sqrt{2}\sigma}(x_i,x_j). \tag{11}$$

Since the kernels in PDF estimation are positive functions that decay with the distance between samples, one can think that one kernel placed on a sample creates a potential field in the sample space. Therefore, the PDF estimated using Parzen kernels can be conceived as an information potential field over the space of the samples [28]. So, the information potential can be used to define similarity measures in this space, which do not possess the limitations of the conventional moments [29]. We can then rewrite Rényi's quadratic entropy in terms of the information potential as

$$H_2(X) = -\log IP(X). \tag{12}$$

4 Correntropy Point Matching

Mean square error (MSE) is the most widely used cost function, but it is restricted to second order statistics which only fully quantify random variables that are Gaussian distributed. Fig. 1a provides an illustration of the MSE cost function in the joint space of variables X and Y. MSE is a quadratic function with a valley along the $x = y$ line. The figure explains the behavior of MSE in the joint space where for values of x close to y MSE takes small values; however, for values away from the $x = y$ line, it takes values that increase quadratically due to the second order moment. The quadratic increase shows that MSE works well for data with rapid-decay distributions such as Gaussian. However, for heavy-tail distributions, such as those containing outliers, MSE is not optimal because results far away from the mean will have an amplified effect on the total cost.



(a) MSE (b) Correntropy

Fig. 1. Cost functions in the joint space.

4.1 Correntropy

A generalized correlation function [6], [17] between two random variables X and Y is defined as

$$v(X,Y) = E_{XY}[\kappa(X,Y)] = \iint \kappa(x,y)p_{XY}(x,y)\,dx\,dy, \tag{13}$$

where the expected value is over the joint space and $\kappa(X,Y)$ is any continuous, non-negative definite kernel. If the kernel is the first order polynomial kernel $\kappa(X,Y) = XY$, the function becomes the conventional cross-correlation. If the kernel is a translation invariant kernel like the Gaussian, then it becomes cross-correntropy. Using Taylor series expansion of the Gaussian kernel, the cross-correntropy function can be written as

$$v(X,Y) = \frac{1}{\sqrt{2\pi}\sigma}\sum_{n=0}^{\infty}\frac{(-1)^n}{2^n\sigma^{2n}n!}\iint \|x-y\|^{2n}p_{XY}(x,y)dxdy, \tag{14}$$

which involves all the even-order moments. As a result, cross-correntropy (or simply correntropy) does not suffer from the limitation of Gaussianity inherent in cost functions based on the second-order moment. Since it also quantifies higher order moments of the PDF, correntropy yields solutions that are more accurate in non-Gaussian and nonlinear signal processing.

In practice, the joint PDF is unknown and only a finite number of data points $\{(x_i,y_i)\}_{i=1}^N$ is available, leading to the sample estimator of correntropy

$$\hat{v}_{N,\sigma}(X,Y) = \frac{1}{N}\sum_{i=1}^{N}G_\sigma(x_i-y_i), \tag{15}$$

which has a computational complexity of $O(N)$. Three important properties of correntropy related to our work are: 1) correntropy is well defined provided that the translation invariant kernel $\kappa(X,Y)$ belongs to L^∞, 2) correntropy defines a metric in the joint space of the random variables [7], and 3) if the expected value in (13) is taken over the marginals, i.e., $E_XE_Y[G(X-Y)]$, we obtain the cross information potential. This last property is useful to define the centered correntropy, which is the counterpart of covariance [17]. For more details on these and other properties of correntropy refer to [17].

The cross-correntropy estimate measures the probability of how similar the two random variables are in a neighborhood along the line $x = y$. Fig. 1b provides a plot of the cross-correntropy for a Gaussian kernel. The figure shows the joint pdf and cross-correntropy of variables X and Y. It shows that correntropy, just like MSE, can be used as a similarity measure in the joint space, but different from MSE, the cost emphasizes the behavior along the $x = y$ line while attenuating contributions away from it, where the attenuation effect depends on the kernel shape and parameter. We can use cross-correntropy as a new cost function for shape matching as

$$\max_{f} \sum_{i=1}^{N} G_\sigma(\mathbf{y}_i - f(\mathbf{x}_i)) - \lambda R(f). \tag{16}$$

Note that we had to change the sign of the regularization term since now we are maximizing the cost function.

4.2 Point Matching Based on Correntropy

Suppose we have two point sets $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ and $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$, where $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^d$. To align point-set $\mathbf{X}$ to point-set $\mathbf{Y}$, $\mathbf{X}$ has to undergo some transformation f which comprises the two transformation matrices: $\mathbf{A}$ and $\mathbf{W}$ as

$$f(\mathbf{x}_i, \mathbf{A}, \mathbf{W}) = \mathbf{A} \cdot \mathbf{x}_i + \mathbf{W} \cdot \phi(\mathbf{x}_i), \tag{17}$$

where $\mathbf{A}$ is a $(d+1) \times (d+1)$ matrix representing the affine transformation, $\mathbf{W}$ is a $(d+1) \times N$ warping coefficient matrix representing the non-rigid transformation, and $\phi(\mathbf{x}_i)$ is a radial basis function of size $N \times 1$. We add a regularization term to control the warping effect, and control the smoothness by constraining the nonrigid transformation. The cost function becomes

$$\mathcal{J} = v(\mathbf{Y}, f(\mathbf{x}_i, \mathbf{A}, \mathbf{W})) - \lambda \| f(\mathbf{x}_i, \mathbf{A}, \mathbf{W}) - f(\mathbf{x}_i, \mathbf{A}) \|^2, \tag{18}$$

or more specifically

$$\mathcal{J}(\mathbf{A}, \mathbf{W}) = \sum_{i=1}^{N} e^{-\frac{\|\mathbf{y}_i - \mathbf{A}\mathbf{x}_i - \mathbf{W}\phi(\mathbf{x}_i)\|^2}{2\sigma^2}} - \lambda \|\mathbf{W}\phi(\mathbf{x}_i)\|^2. \tag{19}$$

To provide an intuitive view of the cost function, consider the model space for the transformation function f as an RKHS $\mathcal{H}$. The function f and the model space $\mathcal{H}$ can be decomposed into two functions $f = f_0 + f_1$ and subspaces $\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1$, where f_0 consists of the affine transformation functions and $\mathcal{H}_0$ is their subspace, and f_1 consists of the non-rigid transformations and $\mathcal{H}_1$ is their corresponding subspace. To control the *smoothness* of the function f, we need to penalize the non-rigid part of the function, f_1. So, the regularization term can be considered as a penalization of the projection of the function f onto the $\mathcal{H}_1$ subspace, which we express as $\|P_1 f\|^2$ where P_1 denotes the orthogonal projection operator onto $\mathcal{H}_1$. Thus, in (19), $\|P_1 f\|^2 = \|\mathbf{W}\phi(\mathbf{x}_i)\|^2$, and λ is the smoothing parameter which controls the balance between the goodness-of-fit and departure from $\mathcal{H}_0$.

Solving for A and W. Before we solve (19) for $\mathbf{A}$ and $\mathbf{W}$, let's first simplify the notation. Consider $\mathbf{X}$ and $\mathbf{Y}$ as $N \times (d+1)$ matrices where each row represents a point $\mathbf{x}_i$ and $\mathbf{y}_i$. Consider $\mathbf{K}$ as a $N \times N$ matrix of radial basis functions where each row represents $\phi(\mathbf{x}_i)$, then the warping coefficient matrix $\mathbf{W}$ needs to also be transposed to a $N \times (d+1)$ matrix. The cost function (19) then becomes

$$\mathcal{J}(\mathbf{A}, \mathbf{W}) = \mathbf{1}_\mathbf{Y}^T G(\mathbf{Y} - \mathbf{X}\mathbf{A} - \mathbf{K}\mathbf{W}) - \lambda \cdot \mathrm{tr}(\mathbf{W}^\mathrm{T}\mathbf{K}\mathbf{W}), \tag{20}$$

where $G(\mathbf{Y} - \mathbf{X}\mathbf{A} - \mathbf{K}\mathbf{W}) = \mathbf{g}$ is a column vector with entries $G_i = \exp(-\frac{\|\mathbf{y}_i - \mathbf{A}\mathbf{x}_i - \mathbf{W}\phi(\mathbf{x}_i)\|^2}{2\sigma^2})$. Taking the derivative with respect to $\mathbf{A}$ and $\mathbf{W}$ results in the following set of equations:

$$\begin{aligned} &\mathbf{X}^T D(\mathbf{g})\mathbf{X}\mathbf{A} + \mathbf{X}^T D(\mathbf{g})\mathbf{K}\mathbf{W} = \mathbf{X}^T D(\mathbf{g})\mathbf{Y}, \\ &\mathbf{K} D(\mathbf{g})\mathbf{X}\mathbf{A} + (\mathbf{K} D(\mathbf{g}) - \lambda \mathbf{I})\mathbf{K}\mathbf{W} = \mathbf{K} D(\mathbf{g})\mathbf{Y}, \end{aligned} \tag{21}$$

where $D(\mathbf{g}) = diag(\mathbf{g})$, and $\mathbf{I}$ is an identity matrix of size $N \times N$. Assuming that $\mathbf{K}$ is invertible, we simplify this set of equations to the following:

$$\begin{aligned} &D(\mathbf{g})\mathbf{X}\mathbf{A} + (D(\mathbf{g})\mathbf{K} - \lambda \mathbf{I})\mathbf{W} = D(\mathbf{g})\mathbf{Y}, \\ &\mathbf{X}^T\mathbf{W} = \mathbf{0}. \end{aligned} \tag{22}$$

Substituting $(D(\mathbf{g})\mathbf{K} - \lambda\mathbf{I})$ with notation $\mathbf{M}$ for simplification, we have the following solutions for $\mathbf{A}$ and $\mathbf{W}$:

$$\begin{aligned} \mathbf{A} &= (\mathbf{X}^T\mathbf{M}^{-1} D(\mathbf{g})\mathbf{X})^{-1}\mathbf{X}^T\mathbf{M}^{-1} D(\mathbf{g})\mathbf{Y}, \\ \mathbf{W} &= \mathbf{M}^{-1}[\mathbf{I} - D(\mathbf{g})\mathbf{X}(\mathbf{X}^T\mathbf{M}^{-1} D(\mathbf{g})\mathbf{X})^{-1}\mathbf{X}^T\mathbf{M}^{-1}] D(\mathbf{g})\mathbf{Y}. \end{aligned} \tag{23}$$

To keep the affine and non-rigid spaces separated so that the penalization above does not interfere with the affine component we decompose the matrix $\mathbf{X}$ using the QR decomposition as

$$\mathbf{X} = [\mathbf{Q}_1 \mathbf{Q}_2] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}, \tag{24}$$

where $\mathbf{Q}_1$, $\mathbf{Q}_2$, and $\mathbf{R}$ are $N \times (d+1)$, $N \times N - (d+1)$, and $(d+1) \times (d+1)$ matrices. $\mathbf{Q} = [\mathbf{Q}_1, \mathbf{Q}_2]$ is an orthonormal matrix, and $\mathbf{R}$ is an upper triangular and invertible matrix. The $\mathbf{Q}_1$ and $\mathbf{Q}_2$ matrices allow us to consider the affine and non-rigid transformations separately. For example, consider the second equation in (22). We have $\mathbf{R}^T\mathbf{Q}_1^T\mathbf{W} = \mathbf{0}$, which means that $\mathbf{Q}_1$ operates in the null space $\mathcal{H}_0$, and since $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are orthonormal matrices, $\mathbf{Q}_2$ operates in $\mathcal{H}_1$.

Since $\mathbf{X}^T\mathbf{W} = \mathbf{0}$, we have $\mathbf{Q}_1^T\mathbf{W} = \mathbf{0}$ and $\mathbf{Q}_2\mathbf{Q}_2^T\mathbf{W} = \mathbf{W}$. Multiplying the first equation in (22) by $\mathbf{Q}_2^T D(\mathbf{g})^{-1}$ and using the fact that $\mathbf{Q}_2^T\mathbf{X} = \mathbf{0}$, we have $\mathbf{Q}_2^T D(\mathbf{g})^{-1}\mathbf{M}\mathbf{Q}_2\mathbf{Q}_2^T\mathbf{W} = \mathbf{Q}_2^T\mathbf{Y}$, which results in:

$$\mathbf{W} = \mathbf{Q}_2\big(\mathbf{Q}_2^T D(\mathbf{g})^{-1}\mathbf{M}\mathbf{Q}_2\big)^{-1}\mathbf{Q}_2^T\mathbf{Y}. \tag{25}$$

Multiplying the first equation in (22) by $\mathbf{Q}_1^T D(\mathbf{g})^{-1}$, we have $\mathbf{R}\mathbf{A} = \mathbf{Q}_1^T D(\mathbf{g})^{-1}(\mathbf{Y} - \mathbf{M}\mathbf{W})$, which results in:

$$\mathbf{A} = \mathbf{R}^{-1}\mathbf{Q}_1^T(\mathbf{Y} - D(\mathbf{g})^{-1}\mathbf{M}\mathbf{W}). \tag{26}$$

4.3 Point Correspondence Using Surprise

The iterative closest point algorithm [3] utilizes the nearest-neighbor relationship to assign a binary correspondence between the two point sets. This estimate of the correspondence is then used to update the transformation function,

which in turn is used to re-estimate the correspondence between the two sets. This is a simple and quick approach, which always converges to a local minimum, and with an adequate set of initial poses, can always find a global match for rigid transformations.

To improve the correspondence problem, Belongie et al. [1] and Jain and Zhang [2] used shape descriptors to provide *context* of the rest of the shape with respect to a given point in the shape. Finding correspondences between the points of two shapes is then equivalent to matching points with similar shape context. We introduce a recently proposed information theoretic descriptor, *surprise* [8], [9], that plays a similar role in shape matching.

A quantitative definition of the relevance a sample point has on the overall set is required to subjectively measure the information available in it based on the current knowledge expressed by the rest of the points. Surprise quantifies how much information a new sample contains relative to a learning system model [30]. To explain this measure, we utilize a few basic definitions from information theory. In information theory, information measures the uncertainty or probability of occurrence of an outcome [31]. The information content of an outcome x, whose probability is $p(x)$, is defined as $I(x) = \log\frac{1}{p(x)}$. For a random variable (r.v.) X, the average information content is defined as $H(X) = \sum_x p(x)\log\frac{1}{p(x)}$ (Shannon's entropy). This classical information measure is based on the full knowledge of the PDF (probability mass function (PMF) in this definition), i.e., it only applies if the PDF is known a priori. This is hardly ever the case in real world pattern recognition applications such as point matching. In an online environment where a new sample x brings more information to the system which, based on partial information, already has an estimate q(x) of the true p(x), it is important to discuss the subjective information content [32]. The subjective information content of an outcome x to the system with subjective probability $q(x)$ is then defined as $I_s(x) = \log\frac{1}{q(x)}$. The average subjective information, $H_s(X)$, is given by the expectation value of the subjective information, $I_s(x)$, taken with respect to the true probabilities $p(x)$ as $H_s(X) = \sum_x p(x)I_s(x) = \sum_x p(x)\log\frac{1}{q(x)}$. Since $H_s(X)$ measures the uncertainty of a system that does not know the correct probabilities, it should be larger than the uncertainty of an ideal observer that knows the true probabilities. Thus, we can state that $H_s(X) \geq H(X)$, with equality being true only when the system has full knowledge of its environment, i.e., the subjective and objective probabilities coincide, $q(x) = p(x)$ [30]. The difference between H_s and H is then defined as:

$$H_m(X) = H_s(X) - H(X) = \sum_x p(x)\log\frac{p(x)}{q(x)}, \qquad (27)$$

which represents the information that the system is missing from the data and that can be learned by adjusting the subjective probabilities closer to the true probabilities. This is called the *information gain* and is the basis for surprise as it measures the system's ignorance [33]. This can be understood from the fact that (27) is nonnegative and vanishes if and only if $p(x)$ and $q(x)$ coincide (Kullback-Leibler divergence) [34].

Assuming that $\mathcal{X}$ is an incomplete point set and x_n is a new sample point, the subjective probability $q(x) = P(\mathcal{X})$ and the "objective" probability $p(x) = P(\mathcal{X}\,|\,x_n)$. Then, the surprise factor is measured as:

$$S(x_n, \mathcal{X}) = H_m(x_n) = KL(P(\mathcal{X}\,|\,x_n), P(\mathcal{X})). \qquad (28)$$

In essence, surprise measures the difference between the prior and the posterior distributions of the point set $\mathcal{X}$ based on the sample point x_n. If the posterior is the same as the prior, the data point carries no new information, thus leaving knowledge about the point set unaffected. However, if the posterior significantly differs from the prior, the sample point carries an element of surprise indicating its importance.

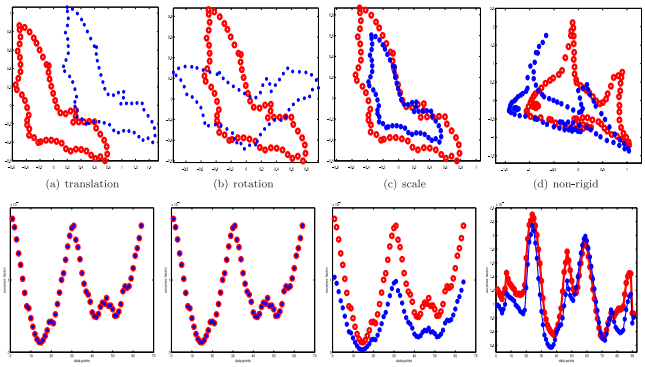


Fig. 2. Transformation effect on surprise value.

4.3.1 Transformation Effect on Surprise

Surprise behavior of corresponding points in an object does not change with respect to most transformations. Fig. 2 shows three common affine transformations: (a) translation, (b) rotation, (c) scaling, and (d) non-rigid transformation. The corresponding surprise graphs show no difference for the translation and rotation transformations, a shift and scaling for the scale transformation, and slight differences for the non-rigid transformation. The surprise factor for each data point is computed with respect to the relative position of the other points. Translation and rotation do not affect the relative position thus leaving surprise unchanged. Scaling alters the relative distance among the points causing the shift and scaling of the surprise graph. The changes noticed under scaling are also due to the PDF estimation used to compute surprise. In our case, we use Parzen window to estimate the PDF; as a result, the kernel bandwidth has an effect on the surprise factor. If we adjust the kernel bandwidth to account for the scaling factor of the object, then both surprise graphs would match. The non-rigid transformation does change the surprise values of the individual points since their relative position is changed; however, the overall shape of the surprise graph before and after the non-rigid transformation remains very similar.

4.3.2 Noise Effect on Surprise

Noise, on the other hand, does affect surprise behavior, but not drastically. Fig. 3 shows four cases where an object is corrupted by (a) white Gaussian noise (b) plus non-rigid transformation, and (c) impulsive noise (d) plus non-rigid transformation. The corresponding surprise graphs show the difference of surprise values before and after the addition of noise and transformation. In the case of the Gaussian noise (a) and (b), while noise has changed

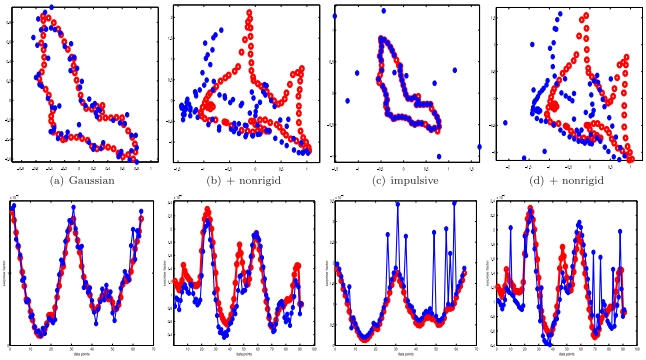


Fig. 3. Noise effect on surprise value.

the surprise values, the overall shape of the surprise graph remains the same. In the case of the impulsive noise (c) and (d), the surprise value for the noisy points is very different from the original; however, the rest of the points have surprise values very close to the originals. Overall, the shapes of the surprise graphs before and after the noise corruption are similar for both types of noise.

4.3.3 Point Correspondence

Based on these results, surprise shows to be a good indicator to quantify differences in object shapes. To demonstrate its capability to correctly determine the correspondence between two point sets, we provide four examples in Fig. 4. The first column shows the initial point correspondence for each example: 1) rotation + white Gaussian noise, 2) rotation + translation + impulsive noise, 3) non-rigid transformation + Gaussian noise, and 4) non-rigid transformation + rotation + scaling. In addition, the data points are reordered so that the first 20 data points in one contour are placed at the end. The second column shows the surprise graphs before and after the transformation and noise corruption. In the first two examples, it is easy to notice that the two graphs are similar except of a circular shift. To determine the circular shift, a circular convolution is applied to the two graphs. The maximum peak, which should be at 20, is shown on the third column for each of the examples. Shifting the data points of the second contour by this amount results in the correct correspondence shown on the fourth column.

4.4 Algorithm

The point set registration algorithm using correntropy is outlined in Algorithm 1. More details about the free parameters and computational complexity are provided in Section 6.

Algorithm 1: Point Set Registration Using Correntropy

input: the two point sets $\mathbf{X}$ and $\mathbf{Y}$
output: a transformed set $\tilde{\mathbf{X}}$ which best aligns with $\mathbf{Y}$
begin
1) Estimate correspondence using *surprise*.
2) Initialize matrices $\mathbf{A}$ and $\mathbf{W}$.
3) Initialize parameters σ and λ.
repeat
4) Compute $\mathbf{A}$ and $\mathbf{W}$ using (26) and (25).
5) Anneal the parameters σ and λ.
until the stopping criterion is satisfied.
end

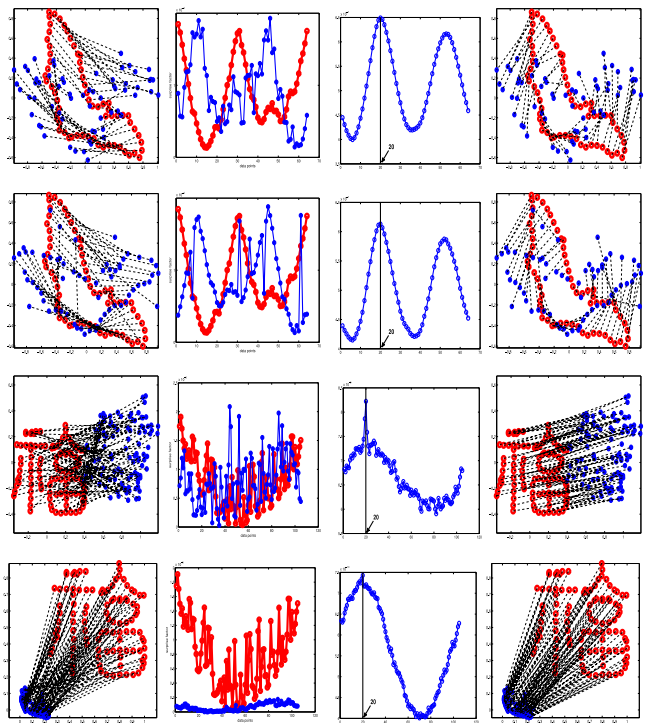


Fig. 4. Using surprise to determine point correspondence.

5 Cauchy-Schwarz Divergence Matching

The main problem using correntropy is that the correspondence of the points on the two sets has to either be predetermined or a method like surprise is required to compute it. In this section, we provide another algorithm where the point sets are represented as probability density functions and the problem of registering point sets is treated as aligning two distributions. Using PDFs instead of points mitigates the need to establish a correspondence between the two point sets. In addition, it provides a robust way of dealing with outliers and noise. The penalty to be paid is a higher computational complexity.

The algorithm operates on the distance between the PDFs of the two point sets to recover the spatial transformation function needed to register them. The distance measure used is the Cauchy-Schwarz divergence which is derived from the inequality with the same name. The algorithm is robust to noise and outliers, and it performs very well on varying degrees of affine and non-rigid transformations.

5.1 Cauchy-Schwarz Divergence

Given two density functions f and g, the Cauchy-Schwarz divergence ($\mathcal{D}_{CS}$) [17] measures the distance between them similarly to the Kullback-Leibler divergence [34]. The measure is derived from the Cauchy-Schwarz inequality [35]:

$$\int f(x)g(x)dx \leq \sqrt{\int f^2(x)dx \int g^2(x)dx}, \qquad (29)$$

where, for PDFs, the equality holds if and only if $f(x) = Cg(x)$ with $C = 1$. To simplify the calculations, we

take the square of (29) and the Cauchy-Schwarz divergence of two PDFs [17] is defined as

$$\mathcal{D}_{\mathcal{CS}}(f \parallel g) = -\log \frac{\left(\int f(x)g(x)\,dx\right)^2}{\int f^2(x)dx \int g^2(x)\,dx}. \tag{30}$$

$\mathcal{D}_{\mathcal{CS}}(f \parallel g)$ is a symmetric measure and $\mathcal{D}_{\mathcal{CS}}(f \parallel g) \geq 0$, where the equality holds if and only if $f(x) = g(x)$. However, the triangle inequality property does not hold, so it cannot be considered as a metric. $\mathcal{D}_{\mathcal{CS}}(f \parallel g)$ can be broken down to and rewritten as:

$$\begin{aligned}\mathcal{D}_{\mathcal{CS}}(f \parallel g) = &-2\log \int f(x)g(x)\,dx \\ &+ \log \int f^2(x)dx + \log \int g^2(x)\,dx.\end{aligned} \tag{31}$$

The argument of the first term, $\int f(x)g(x)dx$, estimates the interactions on locations within the support of $f(x)$ when exposed to the potential created by $g(x)$ (or viceversa). This term measures the similarity (distance) between the two PDFs. The term itself is *Rényi's quadratic cross entropy* [17]. It can also be interpreted as the information gain from observing g with respect to the "true" density f. The other two terms are the negative Rényi's quadratic entropies of the respective PDFs and are considered as normalizing terms that act as regularizers. Remember that averaging correntropy over all possible correspondences results in the cross information potential which is the argument of the logarithm in Rényi's quadratic cross entropy.

The Cauchy-Schwarz divergence is then interpreted as:

$$\mathcal{D}_{\mathcal{CS}}(f \parallel g) = 2H_2(f;g) - H_2(f) - H_2(g). \tag{32}$$

5.2 Algorithm Using $\mathcal{D}_{\mathcal{CS}}$

Suppose we have two point sets $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$ and $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_M)^T$, where $\mathbf{x}_i, \mathbf{y}_j \in \mathbb{R}^d$. We estimate their PDFs using the kernel (Parzen) estimate [27] and consider Gaussian as the kernel function. Substituting the Gaussian kernel PDF estimator in the Cauchy-Schwarz divergence (31) and performing some straightforward manipulations (the integral of the product of two Gaussians is *exactly evaluated* as the value of the Gaussian computed at the difference of the arguments and whose variance is the sum of the variances of the two original Gaussian functions [17]) results in the estimator:

$$\begin{aligned}\mathcal{D}_{\mathcal{CS}}(P(\mathbf{Y}) \parallel P(\mathbf{X})) = &-2\log \sum_{i=1}^{M}\sum_{j=1}^{N} G(\mathbf{y}_i - \mathbf{x}_j) \\ &+ \log \sum_{i=1}^{M}\sum_{j=1}^{M} G(\mathbf{y}_i - \mathbf{y}_j) + \log \sum_{i=1}^{N}\sum_{j=1}^{N} G(\mathbf{x}_i - \mathbf{x}_j).\end{aligned} \tag{33}$$

To represent the affine and non-rigid transformations, we follow the same steps as with correntropy in (17). In addition, the second term in (33), $\log \sum_{i=1}^{M}\sum_{j=1}^{M} G(\mathbf{y}_i - \mathbf{y}_j)$, depends only on the desired point set and its value never changes nor does it affect the transformation function f; thus, from this point on, we remove it from the optimization. Substituting $\mathbf{x}$ with the transformed $f(\mathbf{x})$ yields

$$\begin{aligned}\mathcal{D}_{\mathcal{CS}}(P(\mathbf{Y}) \parallel P(f(\mathbf{X}))) = &-2\log \sum_{i=1}^{M}\sum_{j=1}^{N} G(\mathbf{y}_i - \mathbf{A}\mathbf{x}_j - \mathbf{W}\mathbf{k}_j) \\ &+ \log \sum_{i=1}^{N}\sum_{j=1}^{N} G(\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j) + \mathbf{W}(\mathbf{k}_i - \mathbf{k}_j)) + C.\end{aligned} \tag{34}$$

To control the *smoothness* of the non-rigid function, we need to again include a regularization parameter. The cost function then becomes:

$$\mathcal{J} = \mathcal{D}_{\mathcal{CS}}(P(\mathbf{Y}) \| P(f(\mathbf{X}))) + \lambda \cdot \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{K}\mathbf{W}), \tag{35}$$

where $\mathbf{K}$ is the $N \times N$ Gram matrix of initial point set. *Solving for A and W.*

Again, let's first simplify the notation. Denote $G(\mathbf{y}_i - \mathbf{A}\mathbf{x}_j - \mathbf{W}\mathbf{k}_j), \forall \{\mathbf{y}_i\}_{i=1}^{M}, \{\mathbf{x}_j\}_{j=1}^{N}$ by $G(\mathbf{Y}, \mathbf{X}^T)$. Similarly, $G(\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j) + \mathbf{W}(\mathbf{k}_i - \mathbf{k}_j)), \forall \{\mathbf{x}_i\}_{i=1}^{N}, \{\mathbf{x}_j\}_{i=1}^{N}$ by $G(\mathbf{X}, \mathbf{X}^T)$. Then, (35) can be written as

$$\begin{aligned}\mathcal{J}(\mathbf{A}, \mathbf{W}) = &-2\log\left(\mathbf{1}_{\mathbf{Y}}^T G(\mathbf{Y}, \mathbf{X}^T)\mathbf{1}_{\mathbf{X}}\right) \\ &+ \log\left(\mathbf{1}_{\mathbf{X}}^T G(\mathbf{X}, \mathbf{X}^T)\mathbf{1}_{\mathbf{X}}\right) + \lambda \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{K}\mathbf{W}).\end{aligned} \tag{36}$$

Taking the derivative with respect to $\mathbf{A}$ and $\mathbf{W}$ results in the following equations:

$$\begin{aligned}\mathbf{X}^T(\mathbf{DDG})(\mathbf{X}\mathbf{A} + \mathbf{K}\mathbf{W}) &= \mathbf{X}^T\tilde{G}(\mathbf{Y}, \mathbf{X}^T)^T\mathbf{Y}, \\ \mathbf{K}(\mathbf{DDG})(\mathbf{X}\mathbf{A} + \mathbf{K}\mathbf{W}) + \lambda\mathbf{K}\mathbf{W} &= \mathbf{K}\tilde{G}(\mathbf{Y}, \mathbf{X}^T)^T\mathbf{Y},\end{aligned} \tag{37}$$

where $\mathbf{DDG} = D_{\mathbf{Y}}\{\tilde{G}(\mathbf{Y}, \mathbf{X}^T)\} - D_{\mathbf{X}}\{\tilde{G}(\mathbf{X}, \mathbf{X}^T)\} + \tilde{G}(\mathbf{X}, \mathbf{X}^T)$, $\tilde{G} = \frac{G}{1^T \cdot G \cdot 1}$ to account for the derivative of $\log(\,)$, and $D_{@}\{G\} = diag(1_{@}^T \cdot G)$. The solutions for $\mathbf{A}$ and $\mathbf{W}$ are:

$$\begin{aligned}\mathbf{A} &= [\mathbf{X}^T\mathbf{M}^{-1}(\mathbf{DDG})\mathbf{X}]^{-1}\mathbf{X}^T\mathbf{M}^{-1}\tilde{G}(\mathbf{Y}, \mathbf{X}^T)^T\mathbf{Y} \\ \mathbf{W} &= \mathbf{M}^{-1}\{\mathbf{I} - (\mathbf{DDG})\mathbf{X}[\mathbf{X}^T\mathbf{M}^{-1}(\mathbf{DDG})\mathbf{X}]^{-1}. \\ &\quad\ \mathbf{X}^T\mathbf{M}^{-1}\}\tilde{G}(\mathbf{Y}, \mathbf{X}^T)^T\mathbf{Y},\end{aligned} \tag{38}$$

where $\mathbf{M} = (\mathbf{DDG})\mathbf{K} + \lambda\mathbf{I}$. Utilizing the QR decomposition of $\mathbf{X}$ we can separate the computation of $\mathbf{A}$ and $\mathbf{W}$ as:

$$\begin{aligned}\mathbf{W} &= \mathbf{Q}_2(\mathbf{Q}_2^T(\mathbf{DDG})^{-1}\mathbf{M}\mathbf{Q}_2)^{-1}\mathbf{Q}_2^T(\mathbf{DDG})^{-1}\tilde{G}(\mathbf{Y}, \mathbf{X}^T)^T\mathbf{Y}, \\ \mathbf{A} &= \mathbf{R}^{-1}\mathbf{Q}_1^T(\mathbf{DDG})^{-1}(\tilde{G}(\mathbf{Y}, \mathbf{X}^T)^T\mathbf{Y} - \mathbf{M}\mathbf{W}),\end{aligned} \tag{39}$$

The direct solutions for $\mathbf{A}$ and $\mathbf{W}$ (38) or through the QR decomposition (39) are not very stable. Both updates depend on the parameter λ, which may cause instability. Therefore, the algorithm requires a very slow annealing process to reach the optimal solution. To avoid the instability and speed up the convergence, we provide another set of solutions, based on fixed point update rules, where the new values for $\mathbf{A}$ (40) and $\mathbf{W}$ (41) are calculated with

respect to their old values [36]. Notice that the transformation matrices $\mathbf{A}$ and $\mathbf{W}$ are present in both sides of the fixed point update equations. Also, while there exist many ways of performing the fixed point update, this particular solution set was chosen because of its smooth behavior on both $\mathbf{A}$ and $\mathbf{W}$ across iterations. The $\mathcal{D}_{\mathcal{CS}}$ algorithm follows the same steps as correntropy (Algorithm 1) except of computing the correspondence between the point sets.

$$\begin{aligned}\mathbf{A}^* = (\mathbf{X}^T D_{\mathbf{Y}}\{\tilde{G}(\mathbf{Y},\mathbf{X}^T)\}\mathbf{X})^{-1}\{&\mathbf{X}^T\tilde{G}(\mathbf{Y},\mathbf{X}^T)^T\mathbf{Y}\\ &+\mathbf{X}^T(D_{\mathbf{X}}\{\tilde{G}(\mathbf{X},\mathbf{X}^T)\}-\tilde{G}(\mathbf{X},\mathbf{X}^T))\mathbf{X}\mathbf{A}\\ &+\mathbf{X}^T(D_{\mathbf{X}}\{\tilde{G}(\mathbf{X},\mathbf{X}^T)\}-\tilde{G}(\mathbf{X},\mathbf{X}^T)\\ &-D_{\mathbf{Y}}\{\tilde{G}(\mathbf{Y},\mathbf{X}^T)\})\mathbf{K}\mathbf{W}\},\end{aligned} \tag{40}$$

$$\begin{aligned}\mathbf{W}^* = (\mathbf{K}D_{\mathbf{Y}}\{\tilde{G}(\mathbf{Y},\mathbf{X}^T)\}\mathbf{K}+\lambda\mathbf{I})^{-1}\{&\mathbf{K}\tilde{G}(\mathbf{Y},\mathbf{X}^T)^T\mathbf{Y}\\ &+\mathbf{K}(D_{\mathbf{X}}\{\tilde{G}(\mathbf{X},\mathbf{X}^T)\}-\tilde{G}(\mathbf{X},\mathbf{X}^T))\mathbf{K}\mathbf{W}\\ &+\mathbf{K}(D_{\mathbf{X}}\{\tilde{G}(\mathbf{X},\mathbf{X}^T)\}-\tilde{G}(\mathbf{X},\mathbf{X}^T)\\ &-D_{\mathbf{Y}}\{\tilde{G}(\mathbf{Y},\mathbf{X}^T)\})\mathbf{X}\mathbf{A}\}.\end{aligned} \tag{41}$$

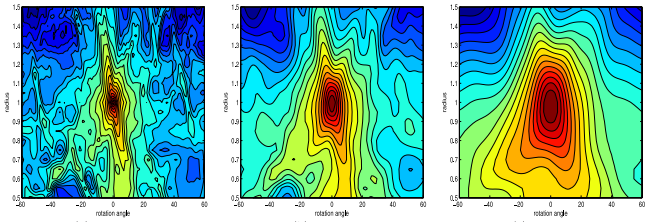
6 Analysis

6.1 Correntropy versus Cauchy-Schwarz Divergence

Both correntropy and $\mathcal{D}_{\mathcal{CS}}$ utilize descriptors from information theory, entropy and divergence, which are estimated directly from the data. Correntropy is a metric applied directly on data samples which makes it less computationally expensive, however it relies on another method to first determine the correspondence between the point sets. $\mathcal{D}_{\mathcal{CS}}$, on the other hand, is applied on densities which mitigates the need for prior correspondence but is more expensive.

Consider the number of samples in each point set as N, the number of dimensions as d, and the number of iterations to reach the stopping criterion as M. Based on Algorithm 1, the computational cost of using correntropy depends on steps (1) and (4). Step (1) estimates the correspondence using surprise with a cost of $2dN^3 + N\log(N)$. Step (4) solves for $\mathbf{A}$ and $\mathbf{W}$ with costs of $dN^2 + 3(d^2+d)N + d^3$ and $2N^3 + 3(d+1)N^2$ respectively. Considering that step (4) is repeated M times, the total computational cost using correntropy is: $2(M+d)N^3 + (4d+3)MN^2 + N\log(N) + 3(d^2+d)MN + d^3M$. The computational cost of using $\mathcal{D}_{\mathcal{CS}}$ depends only on the solutions for $\mathbf{A}$ and $\mathbf{W}$ with a cost of $4dN^2 + (4d^2+d+3)N + 2d^3$ and $6N^3 + (3d+1)N^2 + (d^2+3)N$ respectively. Considering M iterations, the total computational cost is: $6MN^3 + (7d+1)MN^2 + (5d^2+d+6)MN + 2d^3M$. The algorithm using $\mathcal{D}_{\mathcal{CS}}$ is roughly three times more expensive than the one using correntropy. Note that these computational costs were calculated using gradient descent. Other optimization algorithms could be used to lower the cost and/or speed up convergence.

Both correntropy and Cauchy-Schwarz divergence also require two free parameters: kernel bandwidth and the regularization term. These parameters are essential during the optimization process. The next two sections provide descriptions of the two parameters and their adaptations so as to speed up convergence and improve accuracy.



(a) $\sigma = 0.01$ (b) $\sigma = 0.05$ (c) $\sigma = 0.10$

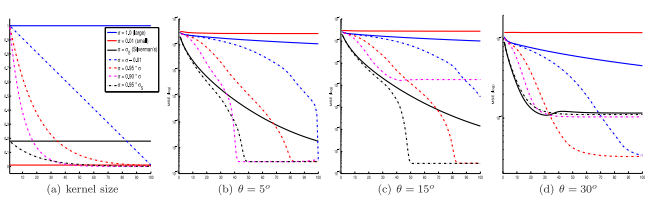
Fig. 5. Performance surface for different kernel sizes.

6.2 Kernel Bandwidth

Correntropy and $\mathcal{D}_{\mathcal{CS}}$ require the kernel bandwidth for estimation directly from samples. In correntropy, the kernel size controls the *observation window* in which similarity is assessed; and as a result, it controls the shape of the performance surface. In $\mathcal{D}_{\mathcal{CS}}$, it is used to estimate the PDF, which again determines the shape of the performance surface. Fig. 5 provides three examples of the correntropy performance surface under various degrees of rotation and scaling for three different kernel sizes. For small kernel sizes, there exist many local maxima in the performance surface, which will cause the algorithm to converge to suboptimal points and rely heavily on the initial conditions. As the kernel size is increased, the performance surface becomes smoother and most of the local maxima vanish. However, for very large kernel sizes, correntropy becomes equivalent to MSE and it will face similar drawbacks. The same point can be made for the Cauchy-Schwarz divergence.

To better understand the effect of kernel bandwidth, σ, on the convergence rate, a simple experiment of matching point sets against a wide range of kernel sizes is shown in Fig. 6. The experiment uses the shapes on the MPEG-7 shape database [37] which are rotated using three angles: 5, 15, and 30 degree to demonstrate the effect of the kernels sizes on different levels of matching difficulty. The kernel sizes used in the experiment are: $\sigma = 1.0$ to demonstrate the effect of a large kernel size, $\sigma = 0.01$ to demonstrate the effect of a small kernel size, $\sigma = \sigma_S$ using Silverman's rule [38], $\sigma = \sigma - 0.01$ to show the effect of a linear decrease, $\sigma = 0.95 \cdot \sigma$ to show the effect of an exponential decrease, $\sigma = 0.90 \cdot \sigma$ a more drastic exponential decrease, and $\sigma = 0.95 \cdot \sigma_S$ using Silverman's estimation as the initial value.

A large σ allows for a quick overall lineup at first, but slows down when it comes to the shape details and may not even converge. A small σ requires a much longer time to converge. If the kernel size is too small, everything may seem different and no convergence occurs as demonstrated here. Using Silverman's rule provides very good accuracy in cases of simple deformations, Figs. 6b and 6c, but it does not perform well for more difficult cases, Fig. 6d.



(a) kernel size (b) $\theta = 5°$ (c) $\theta = 15°$ (d) $\theta = 30°$

Fig. 6. The effect of the kernel bandwidth on the convergence rate for different rotations.

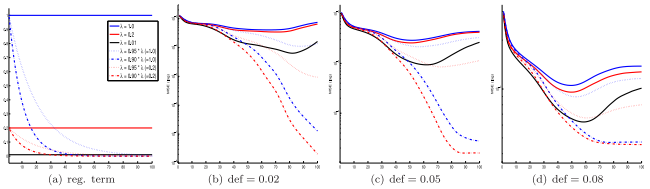


Fig. 7. The effect of the λ parameter on the convergence rate for different deformations.

Based on these observations, we determined that the best approach would be to change the kernel size during optimization. Basically, start the algorithm with a large kernel size and slowly decrease its value through iterations, which is similar to convolution smoothing, a global optimization procedure [39]. This results in a performance surface that changes through iterations but provides two key advantages: 1) the algorithm will be less prone to getting stuck in local extrema, and 2) the convergence rate will be faster (as the gradient of the surface will increase with lower kernel sizes). To this end, we tested various functions to decrease kernel size but provide only the best cases in Fig. 6. The exponential decay proved to be the best method of decreasing the kernel size. However, the initial value, final value and decrease rate still need to be determined. A fast decay rate will converge quickly but its performance deteriorates with the deformation difficulty. A slow decay rate will always ensure a good performance, but will have slower convergence. The initial kernel size needs to be large to ensure not getting stuck in local extrema, especially for difficult deformations as shown in Fig. 6d. The final value needs to be close to but not reach zero.

6.3 λ Parameter

The solutions for $\mathbf{A}$ and $\mathbf{W}$ in both algorithms depend on the parameter λ to control the smoothness of the function. If λ is large, the non-rigid transformation is limited. If λ is small, the non-rigid transformation is too flexible, which could make the algorithm unstable. In Fig. 7, we provide another simple experiment of aligning non-rigidly deformed shape using a wide range of λ values. The data set contains a Chinese character which has undergone three different levels of non-rigid deformation [40]. As expected, a large value of λ provides very little improvement. A small value provides some improvement. However, if the deformed shape had incurred other distortions such as rotation or translation, the small λ value would have caused the points to randomly align with any neighboring point from the other shape.

To control the level of rigidity/flexibility and thus the stability of the algorithm, similar to the kernel bandwidth, the λ parameter needs to be slowly decreased. The goal is to allow the algorithm to first match the two point sets using affine transformations. Then, when the algorithm has reached the optimum solution in the affine space, allow it to match the two point sets using the non-rigid transformations. By keeping the value of the λ parameter large during the first iterations of the algorithm, we restrict the non-rigid transformation and allow the algorithm to perform only global, affine transformations. Then, as the value of λ parameter is gradually decreased, we slowly introduce non-rigid transformations.

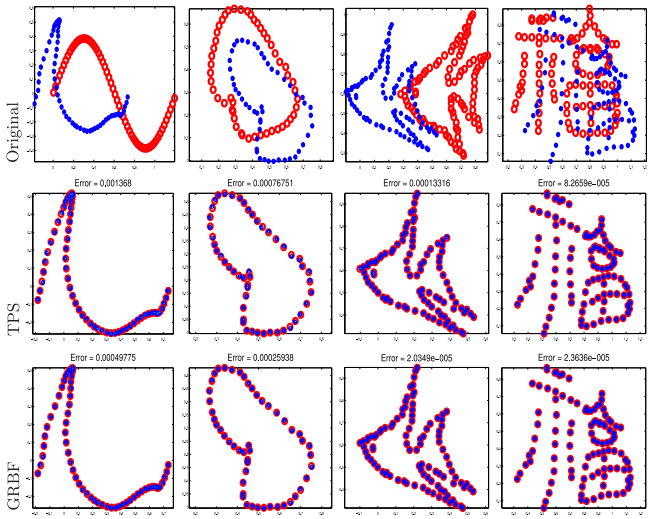


Fig. 8. Comparison of TPS and Gaussian RBF on various non-rigid transformations.

Determining the decay rate is a challenging problem. Through our experiments, we determined that the decay rate needs to be higher than the kernel bandwidth decay rate. Notice in Fig. 7 (more pronounced in (d)) that there is a dip in the performance curve for most parameters. This dip also shifts through iterations depending on the λ value (large-to-small). This occurs because the kernel bandwidth is decreasing through iterations and with that the "observation window" narrows down. The λ parameter needs to allow the function to be flexible enough so that the data points from the two sets are always within this window, otherwise its effect diminishes and the algorithm focuses only on a subset of points. Determining the exact decay rate is a challenging problem that depends on the data set and the problem goals. The optimization of this process is currently under study and will not be pursued any further in this paper.

6.4 TPS versus Gaussian RBF

To understand the effect of the radial basis functions on the non-rigid transformation, we compare TPS and Gaussian RBF on several shapes with various degrees of non-rigid transformation. Fig. 8 shows four experiments where different shapes are aligned using both methods. Both TPS and Gaussian RBF perform well on each case. Visually, there is no difference in alignments. The difference is shown in the title of each sub-figure, which is the MSE value of the final alignment. It shows that Gaussian RBF performs better than TPS. This is due to TPS having a global effect whereas GRBF's effect is localized based on its kernel bandwidth β. While this is an advantage over TPS, it comes at the price of setting an additional parameter.

Fig. 9 demonstrates the problem that might arise from the kernel size in the Gaussian RBFs. The Chinese character above is aligned using three values of β: $0.1, 0.5$, and 1.0. As the value of β increases, the alignment suffers. This is due to the increase in the effect that one control point has over the rest of the points. In the case of $\beta = 1.0$ the control becomes global and it is shown on the poor alignment between the two instances of the character. Controlling the value of β is a problem specific matter that needs to be addressed every

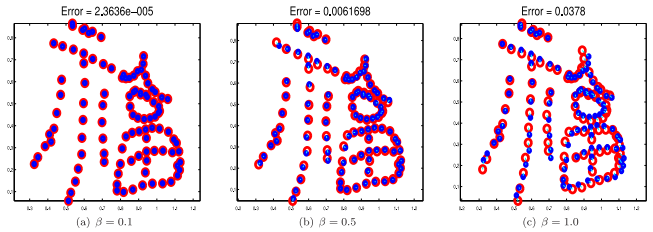


Fig. 9. Non-rigid transformation using Gaussian RBF of various sizes: (a) small, (b) medium, (c) large.

time. Heuristics could be used to assign a value to β by analyzing the distribution of the control points similar to methods used to determine the kernel bandwidth in kernel (Parzen) estimation, which in this case is ≈ 0.1.

6.5 $\mathcal{D}_{CS}$ Compared to GMM

From the algorithms reviewed in the Introduction, GMM [12], [13] most closely resembles $\mathcal{D}_{CS}$. Both algorithms use similarity measures which are analogous. GMM uses the L2 distance, whereas $\mathcal{D}_{CS}$ uses the Cauchy-Schwarz divergence. Also, both methods represent point sets as PDFs. However, GMM is a parametric method using Gaussian mixture models, whereas $\mathcal{D}_{CS}$ is nonparametric using kernel density estimates. Even though $\mathcal{D}_{CS}$ uses Gaussian as the kernel, a Gaussian is put on each sample as a way to compute the distance. $\mathcal{D}_{CS}$ is not assuming a Gaussian model of the data, the PDF can have any shape. In other words, the Gaussian kernel appears at two different scales: for GMM at the macro scale, and for $\mathcal{D}_{CS}$ at the micro scale. In addition, $\mathcal{D}_{CS}$ utilizes kernel bandwidth annealing as mentioned above. By changing the kernel size during optimization, $\mathcal{D}_{CS}$ modifies its emphasis. It starts with a global scale for a quick general alignment, and then follows with a local scale for finer registration.

7 Experimental Results

7.1 2D Quantitative Evaluation

We evaluated the robustness of Correntropy and Cauchy-Schwarz Divergence on various levels of: deformation, noise, outliers, rotation, and occlusion on synthetic data. The data sets used in these experiments were synthesized from [40], and [5]. Each data set contains two shapes, a Chinese character and a fish, which have undergone different degrees of distortion. Figs. 10 and 11 provide examples of the two shapes under different distortions along with the corresponding shape alignments using both our algorithms. In the deformation data set, different levels of deformation are applied to the template point set to create a target point set. In the rest of the data sets, first a slight deformation has been applied to the template point set, and then the deformed template undergoes the respective distortion. A total of 100 point sets are generated at each level of distortion.

Our algorithms along with coherent point drift [14], diffeomorphic matching (Diff) [11], thin-plate splines robust point matching (TPS-RPM) [40], robust point matching by preserving local neighborhood structures (PLNS) [5], and registration using Gaussian mixture models [12] were tested in each data set. Except otherwise stated, the correntropy algorithm uses *surprise* to compute the correspondence

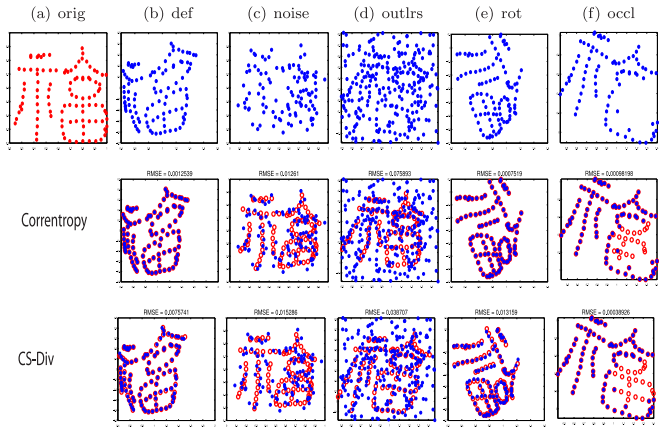


Fig. 10. Corruption examples of the Chinese character, and shape matching using Correntropy and CS-Div.

between point-sets. Note that surprise is run only once at the beginning of the algorithm. To keep most of the parameters the same in all algorithms, both correntropy and CS-Div use thin-plate splines for the non-rigid transformation. To allow each algorithm to converge, 300 iterations was used as the stopping criterion. The initial values for our free parameters were $\sigma = 1$ and $\lambda = 1$, and their exponential decay rates were 0.97 and 0.94. The matching accuracy was quantified using root mean square error (RMSE). The evaluation metric was selected following the work of [5] on the same data set. The statistical results: error mean and standard deviation for each data set are shown on the top row of Figs. 12 and 13.

The standard deviation of most algorithms at higher levels of distortion becomes very large because at certain high-distortion-level trials, an algorithm may completely fail to determine the distortion that the point-set has undergone. This results in high error values rendering standard deviation useless. Therefore, in addition to the mean and standard deviation, the median results are also shown on the bottom row of Figs. 12 and 13.

7.1.1 *Deformation*

Correntropy outperforms all the other methods. This indicates that surprise is robust to deformations and is a good method to obtain correspondence under these conditions.

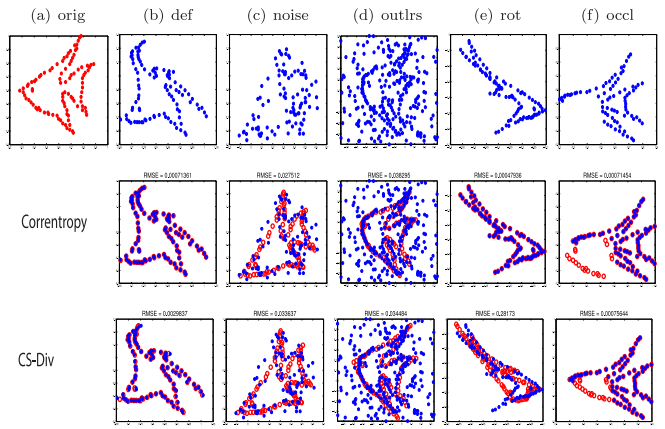


Fig. 11. Corruption examples of the fish point set, and shape matching using Correntropy and CS-Div.

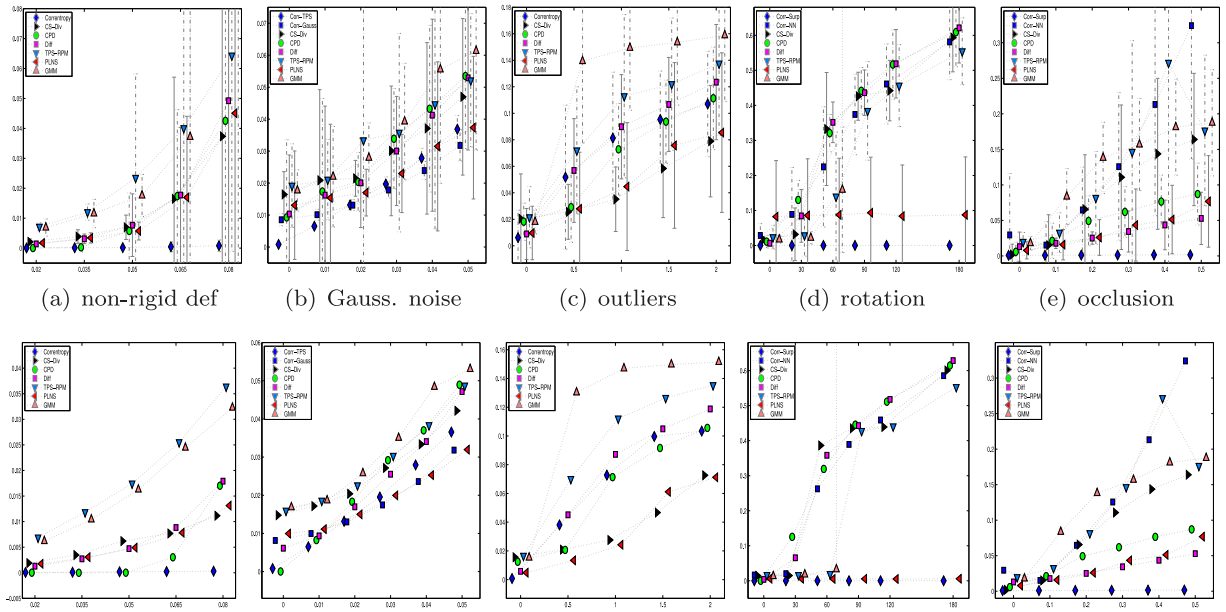


Fig. 12. Matching performance comparison under various levels of deformation for the Chinese character data set. Algorithms used: Correntropy using: TPS ◆, GRBF ■, Surprise ◆, Nearest Neighbor ■, CS-Div ▶, CPD ●, Diff ■, TPS-RPM ▼, PLNS ◀ , and GMM ▲.

CS-Div performance is comparable to the other algorithms. The CS-Div mean error is lower than the rest of the algorithms on most of the deformation levels especially on higher levels. However, some of them fall inside its standard deviation on all the levels. This means that we cannot statistically claim that CS-Div is better. The median graphs reinforce CS-Div performance accuracy especially at higher degrees of non-rigid warping, notably the fish data set.

7.1.2 Noise

For the noise data sets, we include two versions of correntropy: using TPS (Corr-TPS) and Gaussian RBF (Corr-Gauss) to show the advantage of GRBF over TPS. GRBF performs better at higher levels of noise. This is due to GRBF acting on a localized area whereas TPS has more of a global effect. This global effect is noticed on high levels of noise where changes in certain control points resonate over the whole template. While GRBF performs better at higher levels of noise, it is outperformed by TPS at lower levels. This highlights the difficulty of dealing with an additional free parameter. The results show that the β parameter was not optimal for the data set.

The CS-Div also performs well at higher levels of noise where only PLNS has a better mean or median error, however the difference is not statistically significant.

7.1.3 Outliers

Next, we introduce different levels of outliers to the point-set which are measured as outlier-to-data ratio. In these

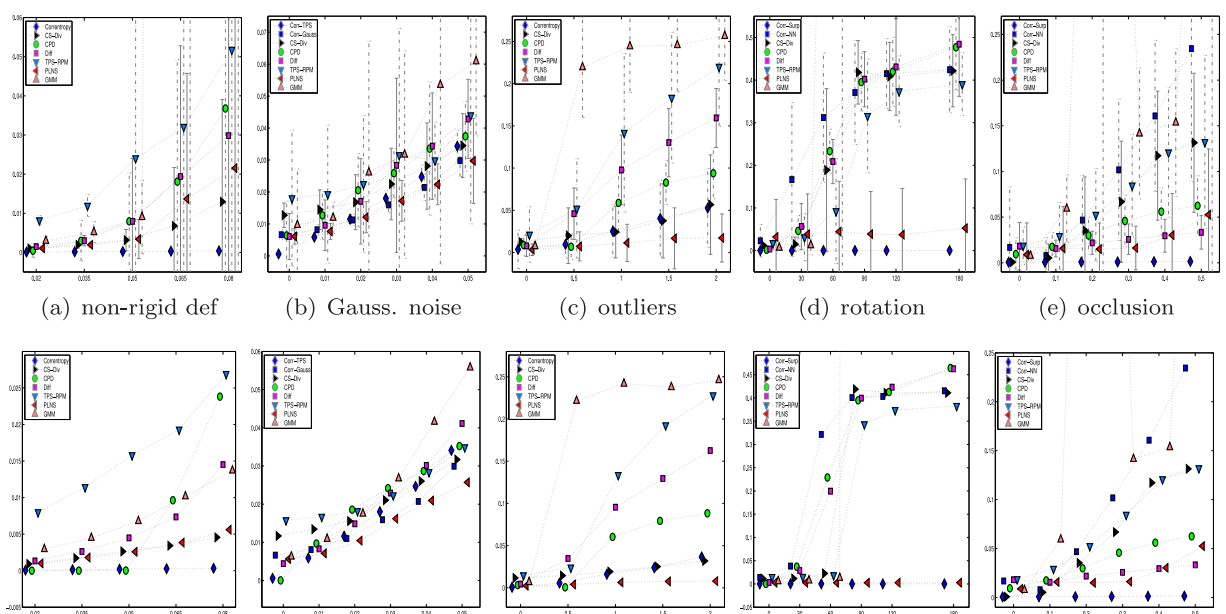


Fig. 13. Matching performance comparison under various levels of deformation for the fish data set.

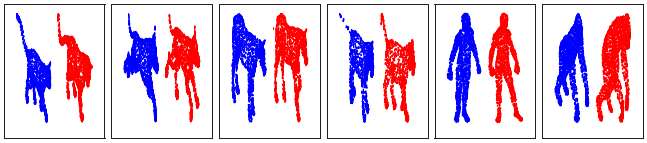


Fig. 14. 3D examples of different non-rigid transformations for six objects: cat, dog, horse, lion, man, and gorilla.

data sets, surprise has difficulty determining the correspondence between the template and the corrupted instance, because the addition of outliers rather than corruption of existing points changes the surprise 'signature' of the shape. For these data sets, nearest-neighbor is used to determine the correspondence. This inaccurate means of determining the correspondence causes the correntropy algorithm to perform poorly. Notice that in the fish data set, when the outlier corruption is not that overwhelming, correntropy still performs well. The CS-Div performs very well in both data sets, especially on high outlier-to-data ratios, which shows that the method is robust against outliers. Its performance is comparable to PLNS, which includes shape information.

7.1.4 Rotation

The next data sets compare the algorithms on various rotation angles. Figs. 10e and 11e show two different angle transformations for the two shapes. The character is rotated 60 degree, whereas the fish is rotated 180 degree. Notice that correntropy performs very well in both cases. This is due to the fact that correntropy utilizes surprise to determine the correspondence between the shapes, and surprise is robust to rotations. CS-Div performs well for any angles below 45 degree. For angles above that, while there are cases that it may find the correct rotation (e.g. 60 degree Chinese character), correct alignment is not guaranteed. For most cases on angles above 45 degree CS-Div converged to a local minimum.

Figs. 12d and 13d show the mean + st. dev. and median results of each algorithm on the different angle transformations. For these data sets, two methods were used to determine the correspondence for correntropy, surprise (Corr-Surp) and nearest-neighbor (Corr-NN). The Corr-Surp outperforms all other methods. This is due to the robustness of surprise against rotation transformations. Corr-NN performs comparably well for transformation angles up to 30 degrees, and fails for any larger angles since it gets stuck in a local maximum. CS-Div also performs well for transformation angles up to 45 degrees. For angles up to 60 degrees, depending on the shape, CS-Div may correctly determine the angle. This is shown in the fish data sets bottom row of Fig. 13d where for many of the trials, CS-Div is able to determine the correct transformation up to 60 degrees. Beyond 60 degrees, CS-Div is not be able to find the correct transformation and converges to a local minimum. This is the case for most of the algorithms that do not utilize any special features about the shapes. PLNS, on the other hand, similar to Corr-Surp, uses information about the structure of the shape to determine the correspondence between the two point sets, and as a result, does not deteriorate on the high rotation angles.

Fig. 15. Alignment of first pair of each object using the different algorithms.

7.1.5 Occlusion

Figs. 10f and 11f show the two shapes with 30 percent occlusion and the final shape alignment of both algorithms. In both shapes, correntropy aligns very well with the target except for the area where occlusion occurs. This happens because those template points do not have any corresponding points on the target, and as a result, their non-rigid transformation weights are missing. The final template alignment for those points depends only on the affine transformation. CS-Div, on the other hand, performs very well even on the occluded areas because CS-Div is based on the full structure of the shape, its PDF, rather than on individual correspondence.

For these data sets, the correntropy algorithm is again run using both surprise and nearest neighbor. Notice that the performance of correntropy using surprise changes very slightly through the different levels of occlusion. This is due to the accurate assignment of the point correspondence between the shapes. Correntropy using nearest-neighbor performs much worse. This is due to the nearest-neighbor approach assigning the wrong correspondence between the two shapes, resulting in higher failure rate as the occlusion rate increases. CS-Div performs well on levels of low occlusion, but its performance starts degrading on higher occlusion levels and cannot be compared to the state-of-the-art methods. There are two reasons for the poor performance: the occlusion area is very large compared to the rest of the shape, and the non-rigid transformation causes confusion in the shape alignment. If the occluded target shapes had only affine transformations, the performance would have been better, due to the overall shape structure remaining intact.

7.2 3D Evaluation

This set of experiments involves aligning three-dimensional nonrigid shapes. The data set, borrowed from [41], [42],

TABLE 1
Alignment Results for the 3D Data Set

	cat	dog	horse	lion	man	gorilla
Correntropy	.0166 ± .0020	.0174 ± .0025	.0212 ± .0037	.0185 ± .0032	.0129 ± .0027	.0157 ± .0064
CS-Div	.0137 ± .0006	.0137 ± .0010	.0155 ± .0012	.0166 ± .0032	.0127 ± .0025	.0141 ± .0037
GMM	.0268 ± .0046	.0218 ± .0044	.0234 ± .0021	.0277 ± .0075	.0207 ± .0063	.0221 ± .0076
CPD	.0171 ± .0016	.0175 ± .0022	.0194 ± .0040	.0199 ± .0049	.0137 ± .0023	.0156 ± .0045
TPS-RPM	.0191 ± .0018	.0203 ± .0025	.0220 ± .0068	.0274 ± .0104	.0190 ± .0059	.0208 ± .0045

contains different animals on a variety of poses that could be used for nonrigid shape similarity experiments. We selected a subset which includes six objects: cat, dog, horse, lion, man, and gorilla on different pairs of nonrigid motions as shown in Fig. 14. The pairs are selected so that the object seems to be modified through a natural, nonrigid motion. Each object contains approximately 2,000 points. However, there does not exist an exact point-correspondence among the different instances of an object.

We compared the performance of our algorithms against GMM [12], CPD [14], and TPS-RPM [40]. We were not able to modify the rest of the algorithms for 3D data. The final alignment of the first pair for each object using the different algorithms is shown in Fig. 15. Overall, correntropy performs well except for some data points that do not have the correct correspondence, and as a result, seem to be scattered out of place. CS-Div performs very well in all cases. The two shapes are almost always perfectly aligned. GMM performs worse than all the other methods. The pairs seem to barely be aligned with major mistakes in the overall non-rigid motion. CPD is comparable to CS-Div, however it fails to accurately determine the non-rigid motion on the shape extremities, such as the legs. TPS-RPM performs well, but again, cannot completely determine the non-rigid motion.

Table 1 provides the quantitative results of this experiment. The alignment accuracy is again quantified using root mean square error. The table shows the RMSE and standard deviation of each algorithm for the six objects. As expected, CS-Div performs the best, it has the lowest RMSE and standard deviation for all the objects. Correntropy's performance is comparable to CPD. Its mistakes primarily occur due to inaccurate correspondence between the point sets. GMM, as expected from the qualitative observations on Fig. 15 performed the worst.

7.3 Real Data Evaluation

Last, we compared the algorithms on the CMU house sequence[1] data set, which contains 111 images of a toy house captured from moving viewpoints. In each image, 30 landmark points were manually marked with known correspondence as shown in Fig. 16. We compared all pairs of images with a separation of 10 to 100 frames resulting in 101 to 11 pairs respectively. Fig. 17a shows the algorithms performance. Both correntropy and CS-Div outperform the other methods as the frame separation increases. CPD has a comparable performance but only when the separation between frames is low.

1. http://vasc.ri.cmu.edu/idb/html/motion/house/.

In addition, we compared the algorithms using the recall metric [43], where recall is defined as the proportion of the true positive correspondences to the ground truth. A correspondence is considered as true-positive when the pair falls within a given accuracy threshold in terms of the pairwise distance. Figs. 17b and 17c shows the recall curves for 50 and 100 separation frames. The average value is taken over image pairs with the same spacing in the sequence. Accuracy thresholds are equally spaced between 0 and 4 pixels. For reference, all images in the data set are of size 384×576 pixels. Correntropy and CS-Div have the highest recall rates. CPD provides high recall at low thresholds but never reaches the maximum. GMM has poor performance at low thresholds and slowly improves as the threshold level increases.

8 Conclusion

In this paper, we have proposed two new nonparametric algorithms to compare pairs of point sets. The algorithms utilize information theoretic similarity measures which provide more information than conventional algorithms and thus improve performance. The first algorithm is based on a non-linear similarity measure known as correntropy. Correntropy measures the similarity of two point sets in a neighborhood of the joint space controlled by the kernel bandwidth. The kernel induces non-linearity to the measure which provides information about higher order moments of the joint PDF. This in turn yields solutions that are more accurate than most of the other methods which rely on MSE as the cost function. In addition, since correntropy works directly on the samples, it has a very low computational complexity, $O(N)$.

Correntropy is a point-based similarity measure which measures the similarity along the line $x = y$ in the joint space. Consequently, it depends on predetermined correspondence between the two point sets. Since the correspondence between point sets is usually not known a priori, we introduced a new method to determine the correspondence,

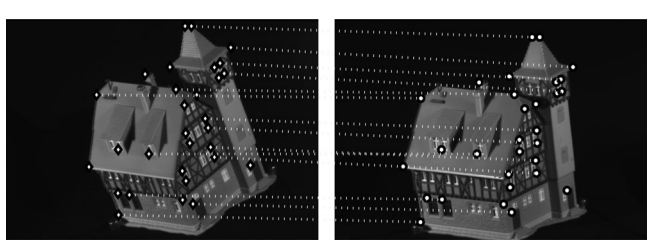


Fig. 16. Two images (frame 0 and 50) from the CMU house sequence. Thirty landmarks points were manually marked in each of image with known correspondences.

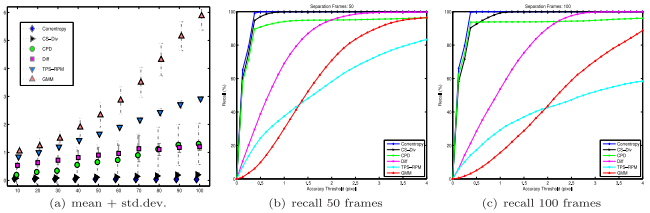


Fig. 17. Matching performance comparison for increasing frame separation: a) rmse and b, c) recall.

surprise, which provides subjective information about each point on the set. It measures the information gain of an exemplar with respect to the sample set, which is called the surprise factor. Provided that the correspondence between the point sets is accurate, correntropy outperforms all other point matching methods.

To bypass the correspondence problem, we proposed a related information theoretic algorithm where point sets are represented as PDFs. The similarity measure known as the Cauchy-Schwarz divergence is used to nonparametrically measure the distortion between the reference point set and the template through the cross-entropy between the two sets. The algorithm gradually emphasizes the importance of local detail into the cost function. Note that while representing point sets as PDFs bypassed the correspondence problem, it increased the computational complexity, $O(N^2)$.

Our two algorithms provide a tradeoff between computational cost and the correspondence problem. We suggest that one chooses the appropriate algorithm based on the data set. If the shape contours are well defined, surprise can accurately determine the correspondence and correntropy will lower computational cost, e.g., automatic target recognition on image-based industrial inspection. If that is not the case, Cauchy-Schwarz divergence could be used.

Both algorithms have two free parameters, kernel size and regularization term. We used deterministic annealing to slowly reduce the value of the kernel bandwidth. This prevents the algorithms from getting stuck in local extrema. In addition, it provides a faster convergence rate. The annealing also makes the methods robust against noise and outliers/occlusion. To constrain the non-rigid transformation so that the structural integrity of the final transformed shapes is not destroyed, a penalization term was introduced in both methods where high values would restrict the non-rigid transformation and low values would allow it to be too flexible. To control the rigidity of the transformation, similar to the kernel bandwidth, we slowly reduce its value so that at the beginning we are primarily focused on global transformations and later we slowly introduce non-rigid transformations to refine the mapping. Determining the free parameters initial values and decay rate is problem dependent and more careful work needs to be done.

Finally, we provided closed-form solutions for both algorithms, which make them easy to compute and feasible for many real-world applications. We compared our methods against other well-known and state-of-the-art methods and showed that our methods provide similar, and in certain cases, better performance.

REFERENCES

[1] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[2] V. Jain and H. Zhang, "Shape correspondence using geodesic shape context," in *Proc. Pacific Graphics*, 2005, pp. 121–124.

[3] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 42, pp. 239–256, Feb. 1992.

[4] H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2000, vol. 2, pp. 44–51.

[5] Y. Zheng and D. Doermann, "Robust point matching for nonrigid shapes by preserving local neighborhood structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 643–649, Apr. 2006.

[6] I. Santamaria, P. Pokharel, and J. Principe, "Generalized correlation function: Definition, properties, and application to blind equalization," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2187–2197, Jun. 2006.

[7] W. Liu, P. Pokharel, and J. Principe, "Correntropy: Properties and applications in non-Gaussian signal processing," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5286–5298, Nov. 2007.

[8] E. Hasanbelliu, K. Kampa, J. C. Principe, and J. T. Cobb, "Online learning using a Bayesian surprise metric," in *Proc. Int. Joint Conf. Neural Netw.*, Jun. 2012, pp. 1–8.

[9] E. Hasanbelliu, K. Kampa, J. Cobb, and J. Principe, "Bayesian surprise metric for outlier detection in on-line learning," in *Proc. SPIE* , vol. 8017, p. 12, 2011.

[10] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *Proc. 8th Eur. Conf. Comput. Vis.*, 2004, vol. 3023, pp. 558–569.

[11] J. Glaunes, A. Trouve, and L. Younes, "Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and submanifolds matching," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, Jun. 2–Jul. 2004, vol. 2, pp. II-712–II-718.

[12] B. Jian and B. Vemuri, "A robust algorithm for point set registration using mixture of Gaussians," in *Proc. IEEE 10th Int. Conf. Comput. Vis.*, 2005, vol. 2, pp. 1246–1251.

[13] B. Jian and B. C. Vemuri, "Robust point set registration using Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1633–1645, Aug. 2011.

[14] A. Myronenko, X. Song, and M. A. Carreira-Perpinan, "Non-rigid point set registration: Coherent point drift," presented at the 20th Advances in Neural Information Processing Systems 19, Vancouver, BC, Canada, 2006.

[15] H. Chui and A. Rangarajan, "A feature registration framework using mixture models," in *Proc. IEEE Workshop Math. Methods Biomed. Image Anal.*, 2000, pp.190–197.

[16] A. Yuille and N. Grzywacz, "The motion coherence theory," in *Proc. 2nd Int. Conf. Comput. Vis.*, 1988, pp. 344–353.

[17] J. C. Principe, *Information Theoretic Learning, Renyi's Entropy and Kernel Perspectives*. New York, NY, USA: Springer, 2010.

[18] R. Jenssen, J. C. Principe, D. Erdogmus, and T. Eltoft, "The Cauchy-Schwarz divergence and Parzen windowing: Connections to graph theory and Mercer kernels," *J. Franklin Inst.*, vol. 343, no. 6, pp. 614–629, 2006.

[19] R. Sibson and G. Stone, "Computation of thin-plate splines," *SIAM J. Sci. Stat. Comput.*, vol. 12, no. 6, pp. 1304–1313, 1991.

[20] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, Jun. 1989.

[21] V. I. Aresenin and A. N. Tikhonov, *Solutions of Ill-Posed Problems*, Washington, DC, USA: Winston and Sons, 1977.

[22] G. Wahba, *Spline Models for Observational Data*. Philadelphia, PA, USA: SIAM, 1990.

[23] A. L. Yuille and N. M. Grzywacz, "A mathematical analysis of the motion coherence theory," *Int. J. Comput. Vis.*, vol. 3, no. 2, pp. 155–175, 1989.

[24] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Comput.*, vol. 7, pp. 219–269, 1995.

[25] C. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL, USA: Univ. Illinois, 1949.

[26] A. Renyi, *Selected Papers of Alfred Renyi*. Budapest, Hungary: Akademia Kiado, 1976.

[27] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist*, vol. 33, pp. 1065–1076, 1962.
[28] D. Erdogmus, "Information theoretic learning: Renyi's entropy and its applications to adaptive system training," Ph.D. dissertation, ECE Dept., Univ. Florida, Gainesville, FL, USA, 2001.
[29] W. Liu, P. Pokharel, and J. Principe, "Correntropy: A localized similarity measure," in *Proc. Int. Joint Conf. Neural Netw.*, 2006, pp. 4919–4924.
[30] E. Pfaffelhuber, "Learning and information theory," *Int. J. Neuroscience*, vol. 3, no. 2, pp. 83–88, 1972.
[31] T. M. Cover and J. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 1991.
[32] M. Belis and S. Guiasu, "A qualitative-quantitative measure of information in cybernetic system," *IEEE Trans. Inf. Theory*, vol. IT-14, pp. 593–594, Jul. 1968.
[33] E. Pfaffelhuber, "Information-theoretic stability and evolution criteria in irreversible thermodynamics," *J. Statist. Phys.*, vol. 16, no. 1, pp. 69–90, 1977.
[34] S. Kullback, *Information Theory and Statistics*. New York, NY, USA: Wiley, 1959.
[35] W. Rudin, *Principles of Mathematical Analysis*. New York, NY, USA: McGraw-Hill, 1976.
[36] E. Hasanbelliu, "Information theoretic similarity measures for shape matching," Ph.D. dissertation, ECE Dept., Univ. Florida, Gainesville, FL, USA, 2012.
[37] S. Jeannin and M. Bobe, "Description of core experiments for MPEG-7 motion/shape," Tech. Rep. MPEG-7, ISO/IEC JTC1/SC29/WG11/MPEG99/N2690, 1999.
[38] B. Silverman, *Density Estimation for Statistics and Data Analysis*. New York, NY, USA: Taylor & Francis, 1986.
[39] R. Y. Rubinstein, "Smoothed functionals in stochastic optimization," *Math. Oper. Res.*, vol. 8, pp. 26–33, 1983.
[40] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Comput. Vis. Image Underst.*, vol. 89, pp. 114–141, 2003.
[41] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Efficient computation of isometry-invariant distances between surfaces," *SIAM J. Sci. Comput.*, vol. 28, no. 5, pp. 1812–1836, Sep. 2006.
[42] S. Young, B. Adelstein, and S. Ellis, "Calculus of nonrigid surfaces for geometry and texture manipulation," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 5, pp. 902–913, Sep./Oct. 2007.
[43] J. Starck and A. Hilton, "Correspondence labelling for wide-timeframe free-form surface matching," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

Erion Hasanbelliu received the BS degree in computer science and the MS degree in systems and software design from Jacksonville State University in 2002 and 2004, respectively. During this time, he was in the Knowledge Systems Laboratory in the areas of artificial intelligence, computer vision, and bioengineering. In 2006, he pursued a doctorate degree at the University of Florida, where he received the MS and PhD degrees in electrical and computer engineering in 2008 and 2012, respectively. During the doctorate, he was at the Computational Neuro-Engineering Laboratory in the areas of machine learning and signal processing.

Luis Gonzalo Sanchez Giraldo received the BS degree in electronics engineering and the MEng degree in industrial automation from Universidad Nacional de Colombia in 2005 and 2008, respectively, and the PhD degree in electrical and computer engineering from the University of Florida in 2012. Between 2004 and 2008, he was appointed as a research assistant at the Control and Digital Signal Processing Group (GCPDS) at Universidad Nacional de Colombia. During his PhD studies, he was a research assistant at the Computational Neuro-Engineering Laboratory (CNEL) at the University of Florida. His main research interests are in machine learning and signal processing.

José C. Príncipe received the master's and PhD degrees in electrical engineering from the University of Porto, Portugal, University of Florida, and Honoris Causa degrees from the Universita Mediterranea in Reggio Calabria, Italy, Universidade do Maranhao, Brazil and Aalto University, Finland. He is a distinguished professor of electrical and biomedical engineering at the University of Florida since 2002. He is a BellSouth professor and the founding director of the University of Florida Computational Neuro-Engineering Laboratory (CNEL). He joined the University of Florida in 1987, after an eight year appointment as a professor at the University of Aveiro, in Portugal. He served as the president of the International Neural Network Society in 2004, as an editor in chief of the *IEEE Transactions of Biomedical Engineering* from 2001 to 2007, and as a member of the Advisory Science Board of the FDA from 2001 to 2004. He is currently the founding editor in chief of the *IEEE Reviews in Biomedical Engineering*. He has been heavily involved in conference organization and several IEEE society administrative committees. He chaired 78 PhD and 61 Master student committees, and he is an author of more than 600 refereed publications (five books, seven edited books, 19 book chapters, 201 journal papers, and 427 conference proceedings). He holds 22 patents and has submitted seven more. He is a fellow of the IEEE (2000), AIMBE (2006), IAMBE (2012), and received the INNS Gabor Award, the IEEE Engineering in Medicine and Biology Society Career Achievement Award, and the IEEE Computational Intelligence Society Neural Network Pioneer Award.