NON-RIGID IMAGE DEFORMATION ALGORITHM BASED ON MRLS-TPS

Huabing Zhou[1], Yuyu Kuang[1], Zhenghong Yu[2], Shiqiang Ren[1], Anna Dai[1], Yanduo Zhang[1],Tao Lu[1],Jiayi Ma[3]

[1]Hubei Provincial Key Laboratory of Intelligent Robot, Computer Science and Engineering School, Wuhan Institute of Technology, Wuhan 430205, China
[2]Guangdong Polytechnic of Science and Technology, Zhuhai 519090, China
[3]Electronic Information School, Wuhan University, Wuhan 430072, China

ABSTRACT

In this paper, we propose a novel closed-form transformation estimation method based on moving regularized least squares optimization with thin-plate spline (MRLS-TPS) for non-rigid image deformation. The method takes the user-controlled point-offset-vectors as the input data, and estimates the spatial transformation about the two control point sets for each pixel. To achieve a realistic deformation, we formulates the transformation estimation as a vector-field interpolation problem by a moving regularized least squares method. Unlike MLS, the mapping function is modeled by a non-rigid function thin-plate spline with regularization technique, such that the deformation can satisfy both global linear affine motion and local non-rigid warping. We derive a closed-form solution of the transformation and achieve a fast implementation. In addition, the proposed method can give a wonderful user experience, fast and convenient manipulating. Extensive experiments on real images demonstrated the proposed method outperforms other state-of-the-art methods and the commercial software Adobe PhotoShop CS 6, especially in case of flexible object motion.

Index Terms— Deformation; moving regularized least-squares method; thin-plate spline function

1. INTRODUCTION

Image deformation, which aims to render new visual effects via geometric transformation and pixel interpolation, is developed upon computer graphics and image processing. It has a number of useful applications ranging from medical imaging, movie effects, virtual reality, criminal detection to animation, structure from motion, face synthesis and analysis [1, 2, 3, 4, 5, 6, 7, **?**, **?**, 10]. The deformation is typically controlled by the user-selected handles. As the user modifies the position of these handles, the image should deform in an intuitive fashion.

Some work has focused on specifying deformations using different types of handles. These handles take the forms of points, lines, and even polygon grids [11]. Alternatively, some methods investigate the type of transformations to perform the desirable deformation. These methods view the problem as scattered data interpolation, and the goal is to generate a dense correspondence based on a set of user-controlled handles. Bookstein *et al*. [12] used thin-plate splines to compute deformation fields corresponding to scattered data samples (manually controlled by the user). Lee *et al*. [13] computed the B-spline basis functions and weights for scattered data interpolation, and proposed a method for adaptively varying the resolution of the control point lattice to avoid a very large number of basis functions. Schaefer [1] proposed to deform images based on Moving Least Squares (MLS) [14] using linear functions such as rigid transformation. The use of MLS and rigid formations makes the deformation 'as-rigid-as-possible' [15].

The methods mentioned above model the deformation with rigid transformation. However, some deformation does not have the property of piecewise rigid, such as the waving flag and the change of expression. We need to use non-rigid model to solve this problem. The spline tool TPS [16] has a close-form solution which can be decomposed into global linear affine motion and local non-rigid warping component controlled by coefficients. It can produces a smooth functional mapping for supervised learning and has no free parameters that need manual tuning. To generate a smooth mapping fitting for the control points, we choose the TPS for parametrization.

Therefore, we formulate the transformation estimation as a vector-field interpolation problem by a moving regular least squares method with thin-plate spline. We name it a moving regularized least squares method with the thin-plate spline (MRLS-TPS). The contribution in this paper includes the following two aspects. Firstly, we introduce the TPS function and regularization technology to the deformation problem, which can help to get more realistic deformation. Secondly,

The authors gratefully acknowledge the financial supports from the National Natural Science Foundation of China under Grants 41501505, 61503288 and 61502354, the PhD Start-up Fund of Guangdong Natural Science Foundation under Grant 2016A030310306, the Natural Science Fund of Hubei province under Grant 2014CFA130 and the Hubei province scientific and technological research project under Grant Q20151504.

we provide a fast implementation, which enables our method to handle large-scale datasets.

2. METHOD

In this paper, the deformation is built based on a set of user controlled points. Let $\{\mathbf{x}_i\}_{i=1}^n$ be a set of control points, and $\{\mathbf{y}_i\}_{i=1}^n$ the corresponding deformed positions of the control points, where $\mathbf{x}_i$ and $\mathbf{y}_i$ are D dimensional column vectors (typically $D = 2$), n is the numbers of points in the two sets.

We view the deformation as a function $\mathbf{f}$ that maps the points in the input image to those in the deformed image, and formulate the function estimation as a vector-field interpolation that should satisfy the following three properties: (i) Interpolation: the points $\{\mathbf{x}_i\}_{i=1}^n$ should map directly to $\{\mathbf{y}_i\}_{i=1}^n$ under deformation; (ii) Smoothness: $\mathbf{f}$ should produce smooth deformations; (iii) Identity: $\mathbf{f}$ should be the identity function if the deformed handles $\{\mathbf{y}_i\}_{i=1}^n$ are the same as $\{\mathbf{x}_i\}_{i=1}^n$ (i.e., $\forall i, \mathbf{p}_i = \mathbf{q}_i \Rightarrow \mathbf{f}(\mathbf{x}) = \mathbf{x}$ with $\mathbf{x}$ being an arbitrary point in the image). These properties are very similar to those used in scattered data interpolation.

Next, we construct a non-rigid deformation function $\mathbf{f}$ satisfying these three properties with a closed-form solution. Note that the proposed method is not influenced by the dimension of the input data.

2.1. Problem Formulation

The mathematical formulation is based on Moving Least Squares (MLS) [1]. For any point $\mathbf{p}$ in the image, MLS solves for a rigid-body transformation $\mathbf{f}_\mathbf{p}(\mathbf{x})$ that minimizes a weighted least squares error functional

$$\sum_{i=1}^n w_i(\mathbf{p})\|\mathbf{f}_\mathbf{p}(\mathbf{x}_i) - \mathbf{y}_i\|^2, \tag{1}$$

where $w_i(\mathbf{p})$ is a non-negative weight function defined as

$$w_i(\mathbf{p}) = \|\mathbf{p} - \mathbf{x}_i\|^{-2\alpha} \tag{2}$$

with α controlling the weight of each control point and $\|\cdot\|$ being the Euclidean distance metric. The global deformation function $\mathbf{f}$ is obtained from a set of local functions, and is defined as $\mathbf{f}(\mathbf{p}) = \mathbf{f}_\mathbf{p}(\mathbf{p})$, which is continuously differentiable.

For traditional MLS methods, the deformation function is defined as a parameter mode, such as affine and similarity. However, for motions of coherent objects such as the waving flag and the change of expression, where the shape of the object deforms under certain nonrigid models, the principle of as-rigid-as-possible deformation then may not work well. Therefore, we introduce regularization terms and generalize this formulation to the non-rigid case. Given an arbitrary point $\mathbf{p}$ in the image, we solve for the optimal transformation function $\mathbf{f}$ that minimizes a weighted regularized least squares error functional

$$\sum_{i=1}^n w_i(\mathbf{p})\|\mathbf{f}_\mathbf{p}(\mathbf{x}_i) - \mathbf{y}_i\|^2 + \lambda\phi(\mathbf{f}_\mathbf{p}). \tag{3}$$

It can be seen that the form is similar with MLS except the regularization term $\phi(\mathbf{f}_\mathbf{p})$, with λ controlling the balance of two aspects. We use deformation function to minimize the error probability of the weighted regularized least squares. The constrained deformation function is different from the previous one, depending on the control points and regularization technique to smooth the deformation.

The function $\mathbf{f}_\mathbf{p}(\cdot)$ is a vector-valued function built on the set $\{\mathbf{x}_i\}_{i=1}^n$, and for any correspondence $(\mathbf{x}_i, \mathbf{y}_i)$, it has $\mathbf{y}_i = \mathbf{f}_\mathbf{p}(\mathbf{x}_i)$. We lie it in a specific functional space $\mathcal{H}$, namely a Reproducing Kernel Hilbert Space (RKHS) [17]. Thus the functional ϕ has the form

$$\phi(\mathbf{f}_\mathbf{p}) = \|\mathbf{f}_\mathbf{p}|_\mathcal{H}^2, \tag{4}$$

where $\|\cdot\|_\mathcal{H}$ denotes the norm of $\mathcal{H}$. We will discuss the detailed forms of $\mathbf{f}_\mathbf{p}$and $\|\cdot\|_\mathcal{H}$ later.

To generate a smooth map of fitting control point transformation [18, 19], we use the TPS function to construct the model. TPS is a general spline tool that produces smooth mapping function in the context of supervised learning, which has the advantages of free parameter and closed solution. The TPS model can be decomposed into a global affine transformation and a local bending function, being controlled by affine matrix and bending function respectively:

$$\mathbf{f}(\mathbf{p}) = \mathbf{A}\overline{\mathbf{p}} + \mathbf{g}_\mathbf{p}(\mathbf{p}), \tag{5}$$

$$\mathbf{g}_\mathbf{p}(\mathbf{p}) = \sum_{i=1}^n \mathbf{K}(\mathbf{p}, \mathbf{p}_i)\overline{\mathbf{c}}_i, \tag{6}$$

where $\mathbf{A}$ defined as a $(D+1)\times(D+1)$ matrix is the global affine transformation. While $\overline{\mathbf{c}}_i$ is a $(D+1)\times 1$ matrix. $\overline{\mathbf{p}}$ is homogeneous coordinates of the point where we set $\overline{\mathbf{p}} = (\mathbf{p}^T, 1)^T$.

The radial basis function $\mathbf{K}$ is a TPS kernel:

$$\mathbf{K}(\mathbf{p}, \mathbf{p}_i) = \|\mathbf{p} - \mathbf{p}_i\|^2 \log(\|\mathbf{p} - \mathbf{p}_i\|). \tag{7}$$

Note that the smooth regularization term now can be defined as $\|\mathbf{f}_\mathbf{p}\|_\mathcal{H}^2 = \sum_{i=1}^N\sum_{j=1}^N\langle\mathbf{K}(\mathbf{p}_i, \mathbf{p}_j)\overline{\mathbf{c}}_i, \overline{\mathbf{c}}_j\rangle = \mathrm{tr}(\mathbf{C}^T\mathbf{K}\mathbf{C}))$, where $\mathbf{C} = (\overline{\mathbf{c}}_1, \ldots, \overline{\mathbf{c}}_N)^T$ is a bending coefficient matrix of size $N \times (D+1)$ and $\mathbf{K} \in \mathbb{R}^{N\times N}$ is a kernel matrix with $\mathbf{K}_{ij} = \mathbf{K}(\mathbf{p}_i, \mathbf{p}_j)$.

The transformation function is smooth. Meanwhile, for the optimal transformation function of control point $\mathbf{p}$, $\mathbf{f}(\mathbf{x}_i) = \mathbf{y}_i$, so $\mathbf{f}$ has interpolation property. Moreover, if $\forall i, \mathbf{x}_i = \mathbf{y}_i$, then $\mathbf{A} \equiv \mathbf{I}$ and $\mathbf{g}_\mathbf{p}(\mathbf{x}) \equiv \mathbf{0}$, therefore, $\mathbf{f}$ is the identity transformation.

The weights w_i are dependent on the evaluation point $\mathbf{p}$, and the regularization technique with TPS kernel is used to impose smoothness. Therefore, we call this a moving regularized least squares optimization method based on the thin-plate spline (MRLS-TPS) minimization.

2.2. A closed-form solution

By substituting Eq. (5) and Eq. (6) into Eq. (1), selecting regularization parameter λ , the non-linear mapping $\mathbf{f}$ can be obtained by minimizing the following TPS energy function:

$$E(\mathbf{A},\mathbf{C}) = \|\mathbf{W}^{\frac{1}{2}}(\overline{\mathbf{Y}}-\overline{\mathbf{X}}\mathbf{A}^T-\mathbf{K}\mathbf{C})\|^2+\lambda\mathrm{tr}(\mathbf{C}^T\mathbf{K}\mathbf{C}), \quad (8)$$

where $\overline{\mathbf{Y}} = (\overline{\mathbf{y}}_1,\ldots,\overline{\mathbf{y}}_N)^T$,$\overline{\mathbf{X}} = (\overline{\mathbf{x}}_1,\ldots,\overline{\mathbf{x}}_N)^T$, TPS kernel describes the internal structural relationship between control points. Non-rigid deformation can be obtained combining with the bending coefficient $\mathbf{C}$. Here, the transformation function can be expressed by matrix $\mathbf{A}$ and $\mathbf{C}$.

We set $\widetilde{\mathbf{Y}} = \mathbf{W}^{\frac{1}{2}}\overline{\mathbf{Y}}$,$\widetilde{\mathbf{X}} = \mathbf{W}^{\frac{1}{2}}\overline{\mathbf{X}}$. Since the image points use homogeneous coordinates, the second smoothing term is the standard TPS regularization term, which is the so-called bending energy. It has special physical meaning and is independent of the linear part of the map.

First of all, we use QR decomposition on matrix $\overline{X}$ to compute the TPS parameters $\mathbf{A}$ and $\mathbf{C}$:

$$\overline{\mathbf{X}} = [\mathbf{Q}_1|\mathbf{Q}_2]\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}, \quad (9)$$

where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are defined as orthogonal matrix of $N\times(D+1)$ and $N\times(N-D-1)$ respectively, $\mathbf{R}$ is a upper triangular matrix of $(D+1)\times(D+1)$. Substituting QR decomposition into equation (8), we can obtain

$$\begin{aligned}E(\mathbf{A},\widetilde{\mathbf{C}}) = &\parallel \mathbf{Q}_2^T(\widetilde{\mathbf{Y}}-\mathbf{W}^{1/2}\mathbf{K}\mathbf{Q}_2\widetilde{\mathbf{C}})\parallel^2\\ &+\parallel \mathbf{Q}_1^T\widetilde{\mathbf{Y}}-\mathbf{R}\mathbf{A}^T-\mathbf{Q}_1^T\mathbf{W}^{1/2}\mathbf{K}\mathbf{Q}_2\widetilde{\mathbf{C}}\parallel^2\\ &+\lambda\mathrm{tr}(\widetilde{\mathbf{C}}^T\mathbf{Q}_2^T\mathbf{K}\mathbf{Q}_2\widetilde{\mathbf{C}}),\end{aligned} \quad (10)$$

where $\mathbf{C} = \mathbf{Q}_2\widetilde{\mathbf{C}}$, implies $\mathbf{X}^T = \mathbf{0}$, making it clear that mapping can be divided into affine and non-affine two parts. $\widetilde{\mathbf{C}}$ is a matrix of $(N-D-1)\times(D+1)$. By minimizing the energy equation (10) about $\widetilde{\mathbf{C}}$ and $\mathbf{A}$, we could get

$$\mathbf{C} = \mathbf{Q}_2\widetilde{\mathbf{C}} = \mathbf{Q}_2(\mathbf{S}^T\mathbf{S}+\lambda\mathbf{T}+\varepsilon\mathbf{I})^{-1}\mathbf{S}^T\mathbf{Q}_2^T\widetilde{\mathbf{Y}}, \quad (11)$$

$$\mathbf{A} = (\widetilde{\mathbf{Y}}-\mathbf{W}^{1/2}\mathbf{K}\mathbf{C})^T\mathbf{Q}_1\mathbf{R}^{-T}, \quad (12)$$

where $\mathbf{S} = \mathbf{Q}_2^T\mathbf{W}^{1/2}\mathbf{K}\mathbf{Q}_2$, $\mathbf{T} = \mathbf{Q}_2^T\mathbf{K}\mathbf{Q}_2$, with $\varepsilon\mathbf{I}$ controlling the stability of the numerical value.

Substituting Eq. (11) and Eq. (12) into Eq. (5) and Eq. (6), we get the closed-form solution of $\mathbf{f}$:

$$\mathbf{f}(\mathbf{p}) = \mathbf{A}\overline{\mathbf{p}}+(\mathbf{K}_\mathbf{p}\mathbf{C})^T, \quad (13)$$

where $\mathbf{K}_\mathbf{p}$ is a row vector of $1\times N$, the ith element is denoted as $\mathbf{K}(\mathbf{p},\mathbf{p}_i)$.

In order to obtain a deformation image, each point can be deformed by using formula (3). We summarize our approach in Alg. 1.

Algorithm 1: The Proposed Algorithm

Input: Image, parameters α and λ
Output: Deformed image
Choose a set of control correspondences $\{\mathbf{x}_i,\mathbf{y}_i\}_{i=1}^n$;
Approximate the image with a grid;
repeat
  Choose a vertex $\mathbf{p}$ on the grid;
  Compute the weight $\mathbf{W}$ by Eq. (2);
  Compute the vector $\mathbf{K}_\mathbf{p}$ by Eq (7);
  Compute the affine matrix $\mathbf{A}$ by Eq (12);
  Compute the bending coefficient $\mathbf{C}$ by Eq (11);
  Compute $\mathbf{f}$ at vertex $\mathbf{p}$ by using Eq. (13);
until *all the vertexes are computed* ;
The deformed image is generated by a bilinear interpolation of $\mathbf{f}(\mathbf{p})$.

2.3. Fast implementation and computational complexity

The formula (13) relates to matrix operations at each point, while the most time-consuming matrix operation is the computation of matrix $\mathbf{C}$ in solution (11), which involves a matrix inversion of $N\times N$. Therefore, the time complexity of the proposed method is $O(mN^3)$, where m is defined as the number of mesh vertices involved in the computation. In the image deformation problem, the number of control points is usually very small, generally not more than 10, so that we get the matrix inversion quickly. Meanwhile, $\mathbf{X}$ is generally fixed for the QR decomposition of the point set. Therefore, a lot of steps can be calculated in advance in formula (11). In fact, matrix $\mathbf{C}$ and $\mathbf{A}$ can be rewritten as

$$\mathbf{f}(\mathbf{p}) = (\mathbf{Y}^T-\mathbf{T}^T\mathbf{H}\mathbf{W}^{1/2})\mathbf{B}\overline{\mathbf{p}}+\mathbf{Y}^T\mathbf{M}. \quad (14)$$

$$\mathbf{C} = \mathbf{Q}_2\widetilde{\mathbf{C}} = \mathbf{Q}_2(\mathbf{S}^T\mathbf{S}+\lambda\mathbf{T}+\varepsilon\mathbf{I})^{-1}\mathbf{S}^T\mathbf{Q}_2^T\widetilde{\mathbf{Y}}. \quad (15)$$

The most time consuming step of our proposed algorithm is to solve the transformation $\mathbf{f}_\mathbf{p}$ using linear system (11), which requires $O(M^3)$ time complexity and may pose a serious problem for large values of M. Even when it is implementable, a suboptimal but faster method may be a better choice. In this section, we provide a fast implementation based on a similar kind of idea as the subset of regressors method [20, 21].

Setting $\mathbf{E} = \mathbf{Q}_2(\mathbf{S}^T\mathbf{S}+\lambda\mathbf{T}+\varepsilon\mathbf{I})^{-T}-\mathbf{Q}_2^T$, combining solution (11) with formula (12), we can get $\mathbf{B} = \mathbf{Q}_1\mathbf{R}^{-T}$, $\mathbf{H} = \mathbf{E}\mathbf{K}$, $\mathbf{M} = \mathbf{E}(\mathbf{K}_\mathbf{P})^T$. $\mathbf{B}$, $\mathbf{H}$ and $\mathbf{M}$ can be calculated in advance, thus greatly improving the computational efficiency of the algorithm. In the process of solving the model, there is only a simple multiplication and addition calculation. Therefore, the method has better computational efficiency in practical application.

Parameter Setting: There are mainly two parameters in our method: α and λ, Parameter α controls the weight of each control point. Parameter λ controls the complexity of the transformation. We set $\alpha = 2$ and $\lambda = 8$.

Table 1. Runtime of MLS and MRLS-TPS.

	Wolf	*Flag*	*Face*	*Leaf*
#ctrl pt	8	8	4	15
grid	30×20	30×48	30×33	30×30
MLS (s)	0.7864	0.5832	0.5745	0.5624
MRLS-TPS (s)	0.4821	0.3351	0.3449	0.3458

3. EXPERIMENTAL RESULTS

In order to verify the feasibility of the algorithm, we test the performance of our proposed algorithm on quite a lot of real images. Some state-of-the art deformation methods such as MLS [1] and the commercial software Adobe PhotoShop CS 6 are well-established for comparison. The parameters of the methods are all fixed. The experiments have been conducted on a laptop with 2.5-GHz Intel Core CPU, 8-GB memory, and MATLAB code. in general, our method can always produce satisfying results. Here, we present the deformation results on only four images because they have four representative types of deformations.

As shown in Fig. 1, the first column presents the original images with the control points marked by green points, while the rest columns are the corresponding deformation results of MLS, Adobe PhotoShop CS 6 and our MRLS-TPS respectively with the initial control points marked by blue points. For all the images, the deformed control points are marked by red circles. For the first row (*Wolf*), to make the wolf's ears smaller, our MRLS-TPS is robust and can produce a natural deformation, in contrast, the MLS gives a undesirable deformation that appear on its eyes and head; and the control points of Adobe PhotoShop CS 6 can not reach the desired position while the control points have a large displacement. We further consider the second row (*Flag*), it is shown that all of the three methods can reshape the red flag well when we choose enough control points on the flag. However, our result is more natural which can be seen from the smoothness of the line. In the third row (*Face*), we only choose two points to control the deformation, and keep other two points motionless. This generally occurs in the case of fine adjustment. Because MLS is a parameter model, it begins to degenerate. Finally, the example in the last row (*Leaf*) also shows that our method is capable of handling local strong deformation. Note that in the result of MLS, i.e., the second figure, the deformation area on right line does not look gentle, while in the result of Adobe PhotoShop CS 6, i.e., the third figure, some points on the right line become raised, which are undesirable.

The software Adobe PhotoShop CS 6 does not provide the tool for calculating the runtime, we only compare the runtime with MLS. Table 1 summarizes the runtime of the methods of MLS and MRLS-TPS. We see that all algorithms perform quite fast and can be performed in real-time. Due to the closed-form solution only involves simple matrix operations such as addition and multiplication, the use of non-rigid model in MRLS-TPS does not lead to efficiency decreases.

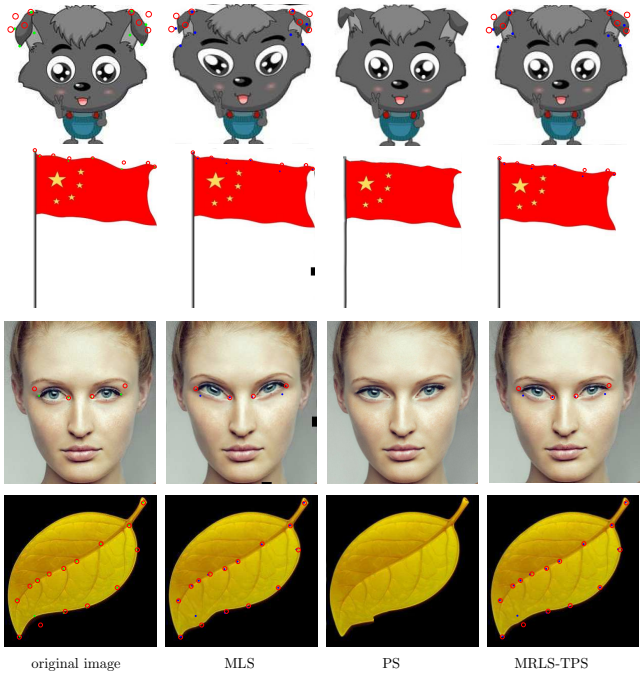

Fig. 1. 2D image deformation results. From left to right: original images, results of MLS [1], Adobe PhotoShop CS 6 (PS for short) and MRLS-TPS, respectively. From top to bottom: *Wolf*, *Flag*, *Face*, and *Leaf*. The marked green and blue points in the images are initial control points, and the marked red circles in the images are deformed control points.

4. CONCLUSION

Within this paper, we presented a new method called moving regularized least squares optimization with thin-plate splines. In our method, the spatial mapping related to control points is parameterized by TPS, and then the mapping can be clearly decomposed into linear and nonlinear components. Moreover, the bending energy minimized by TPS has a specific physical explanation. This may be beneficial in the case of image deformation with non-rigid motions. The results show that the proposed method is able to produce natural non-rigid deformations, especially for motions of coherent objects. Moreover, it is very fast and can be performed in real-time.

5. REFERENCES

[1] S. Schaefer, T. McPhail, and J. Warren, "Image deformation using moving least squares," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 533–540, 2006.

[2] J. Ma, J. Zhao, and J. Tian, "Nonrigid image deformation using moving regularized least squares," *IEEE Signal Processing Letters*, vol. 20, no. 10, pp. 988–991, 2013.

[3] T. Ju, J. Warren, G. Eichele, C. Thaller, W. Chiu, and J. Carson, "A geometric database for gene expression data," in *ESGP*, 2003, pp. 166–176.

[4] H. Zhou, J. Ma, Y. Zhang, Z. Yu, S. Ren, and D. Chen, "Feature guided non-rigid image/surface deformation via moving least squares with manifold regularization," in *ICME*, 2017.

[5] Q. Zhou, ur R. Shafiq, Y. Zhou, X. Wei, L. Wang, and B. Zheng, "Face recognition using dense sift feature alignment," *Chinese Journal of Electronics*, vol. 25, no. 6, pp. 1034–1039, 2016.

[6] Y. Gao and A. L. Yuille, "Exploiting symmetry and/or Manhattan properties for 3D object structure estimation from single and multiple images," in *CVPR*, 2017.

[7] Y. Gao and A. L. Yuille, "Symmetric non-rigid structure from motion for category-specific object structure estimation," in *ECCV*, 2016, pp. 408–424.

[8] J. Jiang, J. Ma, C. Chen, X. Jiang, and Z. Wang, "Noise robust face image super-resolution through smooth sparse representation," *IEEE Transactions on Cybernetics*, 2017.

[9] J. Jiang, C. Chen, J. Ma, Z. Wang, Z. Wang, and R. Hu, "Srlsp: A face image super-resolution algorithm using smooth regression with local structure prior," *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 27–40, 2017.

[10] Y. Gao, J. Ma, and A. L. Yuille, "Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2545–2560, 2017.

[11] R. MacCracken and K. I. Joy, "Free-form deformations with lattices of arbitrary topology," in *ACM SIGGRAPH*, 1996, pp. 181–188.

[12] F. L. Bookstein, "Thin-plate splines and the atlas problem for biomedical images," in *IPMI*, 1991, pp. 326–342.

[13] S. Lee, G. Wolberg, and S. Y. Shin, "Scattered data interpolation with multilevel b-splines," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 3, pp. 228–244, 1997.

[14] D. Levin, "The approximation power of moving least-squares," *Mathematics of Computation*, vol. 67, no. 224, pp. 1517–1531, 1998.

[15] T. Igarashi, T. Moscovich, and J. F. Hughes, "As-rigid-as-possible shape manipulation," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 1134–1141, 2005.

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

[17] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.

[18] J. Ma, J. Zhao, Y. Zhou, and J. Tian, "Mismatch removal via coherent spatial mapping," in *ICIP*, 2012, pp. 1–4.

[19] J. Chen, J. Ma, C. Yang, and J. Tian, "Mismatch removal via coherent spatial relations," *Journal of Electronic Imaging*, vol. 23, no. 4, pp. 043012, 2014.

[20] J. Ma, J. Zhao, J. Tian, X. Bai, and Z. Tu, "Regularized vector field learning with sparse approximation for mismatch removal," *Pattern Recognition*, vol. 46, no. 12, pp. 3519–3532, 2013.

[21] J. Ma, J. Zhao, J. Jiang, and H. Zhou, "Non-rigid point set registration with robust transformation estimation under manifold regularization," in *AAAI*, 2017, pp. 4218–4224.