

Chinese Handwriting Generation by Neural Network Based Style Transformation

Bi-Ren Tan^{1,2}, Fei Yin¹, Yi-Chao Wu^{1,2}, and Cheng-Lin Liu^{1,2}(✉)

¹ National Laboratory of Pattern Recognition (NLPR),
Institute of Automation of Chinese Academy of Sciences, Beijing 100190, China
{biren.tan,fyin,yichao.wu,liucl}@nlpr.ia.ac.cn

² University of Chinese Academy of Sciences, Beijing, China

Abstract. This paper proposes a novel learning-based approach to generate personal style handwritten characters. Given some training characters written by an individual, we first calculate the deformation of corresponding points between the handwritten characters and standard templates, and then learn the transformation of stroke trajectory using a neural network. The transformation can be used to generate handwritten characters of personal style from standard templates of all categories. In training, we use shape context features as predictors, and regularize the distortion of adjacent points for shape smoothness. Experimental results on online Chinese handwritten characters show that the proposed method can generate personal-style samples which appear to be naturally written.

Keywords: Handwriting generation · Style transformation
Neural network learning

1 Introduction

Handwriting synthesis is important for many applications, such as computer font personalization, data enhancement for training recognition systems, handwriting-based communication. The synthesis of personal-style handwriting is one of the most important research problems in this field. Its goal is to generate handwritten characters in the same style as the target writer. This is particularly useful for Chinese handwriting, which is known for the large number of characters (e.g. 27,533 categories for GB18030-2000) and the complex characters structures. It is hard for a person to write thousands of characters to make a personal font library, and instead, it is desirable to generate large number of stylized characters by learning from a small number of written characters. The challenge of synthesis is how to grasp the personal style of specific writer, and how to generate stylized characters with smooth shape.

The problem of handwriting modeling and synthesis has been studied for a long time, and there are many related works in the literature. The methods of

handwriting synthesis can be generally divided into three groups: perturbation-based generation, fusion-based generation, statistical model-based generation.

The perturbation-based methods generate new characters by changing the geometric characteristics of original samples, such as size, stroke thickness, tilt and so on [1, 2]. However, this approach is not suitable for synthesizing of personal-style handwritten characters, since the synthesized samples may be unnatural due to random and non-calibrated parameter settings.

Fusion-based methods combine existing samples into new synthesized ones [3–5]. They are more suitable for composing words from letters. The methods of [6–8] tried to split Chinese characters into strokes and then generate new characters by recombining the strokes. As an example Zhou et al. [9] developed a system to construct the shapes of 2,500 simplified Chinese character by recombining radicals of 522 characters written by a user, and thus built a small-scale Chinese font library in the users’ handwriting style. The challenge of this method lies in accurate segmentation of character components, which is the key to make the combined characters look more natural and smooth.

Statistical model-based methods capture the statistics of natural handwriting variations between different styles. A common modeling method [3] is to obtain the mapping relation between the sample points of corresponding character templates, and then obtain the displacement of the matched sample points. Then a new style character can be generated from the statistical model by moving the sample points of the standard template. Lian et al. [10] presented a system to automatically generate a handwriting font library in the user’s personal style with huge amounts of Chinese characters by learning variation of stroke shape and layout. However, this method relies on the precise locating and matching each stroke on the characters.

To generate personal-style handwritten characters flexibly with reduced human efforts, this paper proposes a learning-based method to online personal-style handwriting generation. We take some given characters written by an individual as training samples, and use a neural network to learn the personal style (transformation function) after corresponding the samples points between the handwritten characters and standard templates. The transformation function is then used to generate stylized samples of all categories by transforming standard templates. We validated our algorithm on online Chinese handwritten characters, and the experimental results show that the proposed method can generate qualified handwriting characters of specific personal-styles when learning transformation from only 300 handwritten samples.

The remainder of this paper is organized as follows. Section 2 describes the proposed handwriting generation method. Section 3 introduces the character style transform algorithm based on neural network. Section 4 presents experimental results and Sect. 5 concludes the paper.

2 Handwriting Generation Method

We work with online Chinese handwritten characters, as it is easier to extract stroke trajectories from online characters, and it is trivial to generate offline

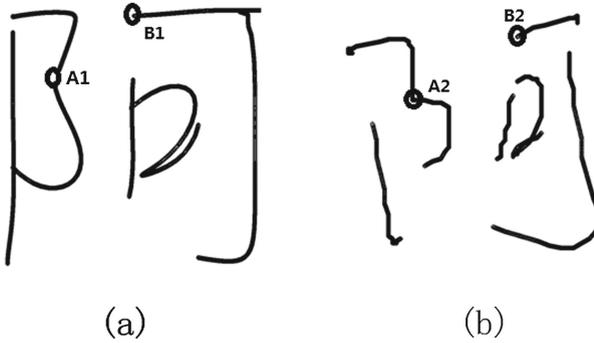


Fig. 1. Examples of online character samples. (a) is the standard template; (b) is the personal-style handwritten character. A1, A2 and B1, B2 are two pairs of corresponding points.

character images from online characters. To generate stylized handwritten characters for a large category set from a small number of handwritten samples, we have standard templates (such as carefully written samples or printed characters) for all the categories in a set (say, GB2312-80 or GB18030-2000). The handwritten samples are matched with the corresponding standard templates to get the correspondence of stroke points, and transformation function is learned by neural networks from the correspondence. The learned transformation is applied to the standard templates of all categories to generate stylized handwritten characters. In Fig. 1, we show a pair of corresponding characters where (a) is the standard template and (b) is a personal handwritten character. In the following, we describe the procedures of character sample points matching and the measure of matching distance. The style transformation method is detailed in Sect. 3.

2.1 Sample Points Matching

This task aims to obtain the corresponding relation between two sets of sample points. The problem of point set registration has been studied for a long time. There are many available algorithms for point set matching. In this paper, we choose the TPS-RPM algorithm [11] to implement registration between standard and target point set. Specifically, $C_x = (p_{x_1}, \dots, p_{x_n})$ represents an online standard character with n sample points, and $C_y = (p_{y_1}, \dots, p_{y_m})$ is the target character with m points.

The TPS function can be used to simulate the non rigid deformation by decomposing the spatial transformation into a global affine transformation and a local non rigid transformation. Generally speaking, this point matching process is divided into two steps. Firstly, we modify matching matrix $\{M_{ij}\}$, under the current transformation parameter (d, w) ; Secondly, we make the matching matrix unchanged and estimate the TPS parameters. Under the framework of the deterministic annealing technique, these two steps are iterated until convergence with the gradual decline of the control temperature T . In this process, the

author obtained the matching matrix and the TPS parameters by minimizing the following objective functions:

$$E_{TPS}(M, d, w) = \sum_{i=1}^m \sum_{j=1}^n M_{ij} \|p_{x_i} - p_{y_j} d - \phi w\|^2 + \lambda \text{trace}(w^T \phi w), \quad (1)$$

where M_{ij} represents the matching probability between sample point p_{x_i} and p_{y_j} , d and w is the affine and non rigid transformation parameter, respectively.

2.2 Matching Distance Between Characters

The online characters are represented as combined of strokes $C = (S_1, S_2 \dots S_N)$, where S_N is the N th stroke of character C , and the strokes can be seen as a set of ordered points $S = (p_1, p_2 \dots p_M)$. The similarity between two characters is defined as the average distance of matching points. Sect. 2.1, we describe the matching method between two corresponding characters C_1 and C_2 , where $C_1 = (p_1^{C_1}, p_2^{C_1} \dots p_n^{C_1})$, $C_2 = (p_1^{C_2}, p_2^{C_2} \dots p_m^{C_2})$, the corresponding matching point set of C_1 is $C_{\text{match}} = (p_1^{M12}, p_2^{M12} \dots p_n^{M12})$, the average matching distance between C_1 and C_2 define as follow:

$$d_{12} = \frac{1}{n} \sum_{i=1}^n \sqrt{(p_i^{C_1} - p_i^{M12})^2}, \quad (2)$$

where the value of d_{12} determines the similarity of C_1 and C_2 . The problem of calculating the similarity between two characters thus become that of computing the distance between two sets of matching points.

3 Style Transformation Learning

Give a number of personal-style handwritten characters as training samples, our method first match each training sample with its standard template of same category and get the corresponding pairs of sample points. We then use a neural network to learn the transformation function from the sample points of training samples to those of standard templates. Shape context features are extracted from the neighborhood of each sampled point as predictors (inputs of the neural network). And to guarantee the smoothness of generated samples, we propose multiple sampled points regression taking into account the spatial relationship between the points.

3.1 Sample Point Context Feature Extraction

In our learning model, we use shape context features as predictors, and regularize the distortion of adjacent points for shape smoothness. In this paper the shape context feature [12] is obtained by analyzing the distribution of the peripheral sample points. For example, in Fig. 2 we take a sample point in the standard

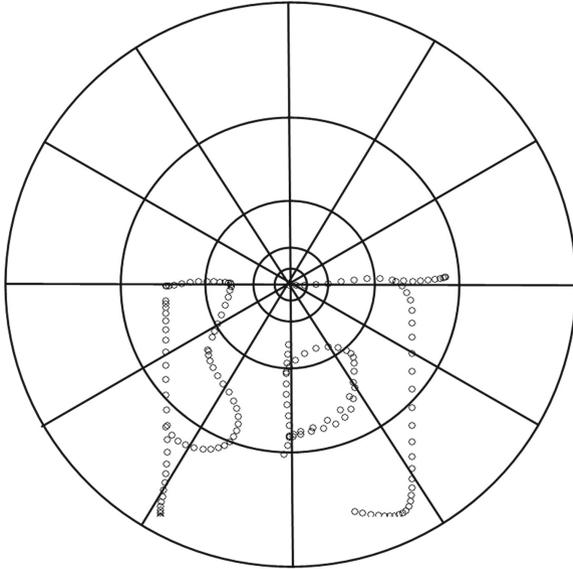


Fig. 2. Obtain the shape context feature of sample point.

character as the center of a circle whose radius equals to the width of the character, then we divide the circular region into 60 bins, so that we can obtain the distribution histogram of the sample points as 60 dimensional context features. Based on the statistical distribution of sample points, the shape context feature can describe the global information of a character. To solve specific problems, we can change the number of bins to obtain the most suitable of shape context feature in the experiment.

3.2 Single Points Regression

We first introduce single point regression model which predicts the displacement of only one point. In this model, the features of sample point consist of position information and context information. They are represented by the coordinates of the sample point and its 120 dimensional shape context feature. The two dimensional coordinates of target point are the outputs of FNN, and its structure was experimentally chosen as $I * H_1 * H_2 * H_3 * O = 122 * 100 * 100 * 100 * 2$, where I, H and O represent the input, hidden and output layers, respectively. Besides, the mean square error of the output coordinates is used as the network loss function:

$$L = \frac{1}{2m} * \sum_m (Y - Y_o)^2. \tag{3}$$

However, from the experimental results, the performance of single point regression model didn't meet our expectation because some synthesized characters were distorting. This is due to drastic change of relative position between

adjacent points. To solve this problem, we need to smooth each stroke by a post-processing. Let $C^D = (S_1^D, S_2^D \dots S_n^D)$ represent a deformed character composed of n strokes, and $S_K^D = (p_1^D, p_2^D \dots p_m^D)$ is its K th stroke which is consisted of m regression points. The following is the smoothing process of S_K^D :

1. Calculate the new coordinates of every point:

$$p_j^{\text{new}} = \begin{cases} p_1^D & j = 1 \\ (p_{j-1}^D + p_j^D + p_{j+1}^D) / 3 & 1 < j < m \\ p_m^D & j = m. \end{cases} \quad (4)$$

2. Repeat step 1 until the stroke look natural, usually we only need to repeat 3 times. It should be noted that each stroke is smoothed independently.

Experimental results show that adding the position constraint of adjacent points is an effective method to improve the synthetic quality of strokes. Further, in the following section we will try to regularize the distortion of adjacent points for shape smoothness in the training process.

3.3 Multi-point Regression

In training the neural network to fit multiple samples points simultaneously, the spatial relationship between the points is considered to smooth the deformation of stroke shape. We first consider to restrict the relative position of two adjacent points. Because of the proximity of these two input points, they have a similar shape context feature, so we just need to choose one of their shape context as the common feature. Usually we take the first point as the center point and the second point as a constraint point. The 120 dimensional shape context feature of center point and the coordinates of adjacent points consist as the inputs of Neural Network. The structure of network is changed as $I * H_1 * H_2 * H_3 * O = (120 + 2 * N) * 100 * 100 * 100 * (2 * N)$, where N represents the number of input sample points. The objective function of double point regression model becomes as follows:

$$L = \frac{1}{2m} * \sum_m (Y - Y_o)^2 + w * \frac{1}{2m} * \sum ((p_{y_1} - p_{y_2}) - (p_{y_{o_1}} - p_{y_{o_2}}))^2, \quad (5)$$

where Y are the real coordinate of matching points, Y_o is the output of the network, p_{y_1}, p_{y_2} and $p_{y_{o_1}}, p_{y_{o_2}}$ represent two pairs of adjacent points respectively. In (5), the first term is the mean square error, the second term is the penalty term for the change of relative position working as smoothness constraints.

Double point regression restricts the relative position of two adjacent points by controlling their displacement, however, it limits only one direction. In order to further strengthen the smoothness constraints, we constraint the displacement in both ahead and hinder directions. Therefore, we need to use the coordinate of front and rear points as constraint information.

We assume that $X = (p_{x_{c-n}}, p_{x_{c-n+1}}, \dots, p_{x_c}, \dots, p_{x_{c+n-1}}, p_{x_{c+n}})$ is a section of stroke within standard character, p_{x_c} is the center of this section which contains of $2n + 1$ points. We use coordinates of X and the shape context of center point as the input of network.

The matching point set of X is $Y = (p_{y_{c-n}}, p_{y_{c-n+1}}, \dots, p_{y_c}, \dots, p_{y_{c+n-1}}, p_{y_{c+n}})$, Y is the target output of network.

The optimal solution is obtained by minimizing the following objective functions:

$$L = \frac{1}{2m} * \sum_m (Y - Y_o)^2 + w * \frac{1}{2m} * \sum_m ((Y - p_{y_c}) - (Y_o - p_{y_{oc}}))^2, \quad (6)$$

where $(Y - p_{y_c}) - (Y_o - p_{y_{oc}})$ is the penalty term for the change of position relative to center point, and Y_o whose center point is $p_{y_{oc}}$ represents the actual output of network.

4 Experiments

In the experiment, we collected different styles of personal handwritten character sets, each of which has 6,763 characters. We chose one of the well written online character sets as the standard template, and selected one of the remaining sets as the target personal-style.

However, sometimes it is difficult to collect a carefully written samples set and not all handwritten characters qualify as standard characters directly. To solve this problem, we need to normalize the handwritten characters again. The printed Song typefaces are ideal for standard character set, but we can't obtain the stroke trajectories information of printed characters directly. Inspired by Thin Plate Spline deformation [11], we normalize the standard templates by single character deformation. In the Sect. 2.1, we can obtain the deformation functions parameters of two characters during point matching. To normalize one standard character, we first calculate the deformation of corresponding points between an online handwritten character and the stroke trajectory of its corresponding printed character, and then estimate the TPS transformation parameters. Finally, we use the TPS transformation function to deform the shape of character template, and then we can obtain an online character with standard Song typeface style. Figure 3 shows the normalization effect.



Fig. 3. Normalize each template character. (a) are handwritten characters; (b) are normalization results.

We took 300 standard characters and their corresponding target characters as the training samples. We normalized the template character to the same size by keeping the width to height ratio. It was better when the training set was composed of different structural characters. Then we matched the corresponding sample point between different style character templates using TPS point registration algorithm and extracted the 120 dimensional shape context information of each sample point. Finally we learned the transformation function by neural network. Then we used the transformation function to generate stylized samples of all categories by transforming standard templates. Following are the results of our experiments.

4.1 Deformation Effect of Different Learning Models

We compared the synthesis performance of different learning models according to direct observation and matching distance. In the experiment, we set the constraint coefficient $w = 2$, several characters generated by different models are shown in Fig. 4. In this Figure, we could intuitively find that the generated characters are similar to the target temple in both size and layout structure. That proves our regression model is effective and feasible. By contrast we can also find that the performance of multi-point models are better than single point model. In the next experiment, We will further compare the learning performance of each model by calculating the matching distance of sample points.

In Table 1, we show the change of average matching distance of 100 pairs of corresponding characters from different models. D_{ori} is the average matching point distance between original standard character and target character, and D_{def} is the average matching point distance between deform character and target character.



Fig. 4. Compare the generating results with different methods when $w = 2$. (a) are the standard characters, (b) are the target characters, (c), (d), (e), (f) are synthesis results of 1 point, 2 points, 3 points and 5 points regression respectively.

Table 1. The change of average matching distance.

Distance	1 point	2 points	3 points	5 points
D_{ori}	0.15145	0.15145	0.15145	0.15145
D_{def}	0.08765	0.07445	0.06522	0.05644

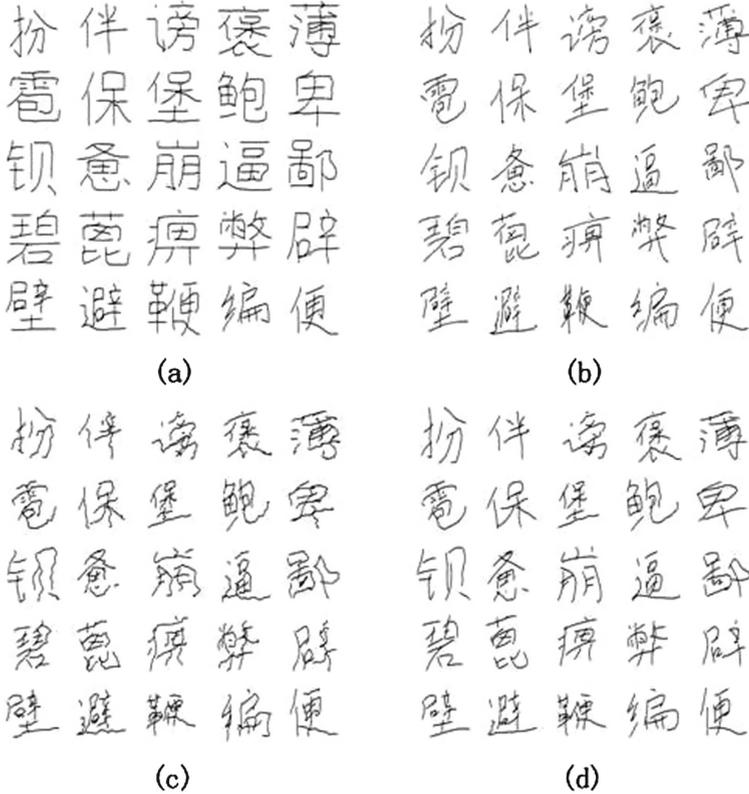


Fig. 5. The effect of smoothness regularization in five-point regression model. (a) Standard templates; (b) target personal-style characters; (c) generated characters without smoothness regularization; (d) generated characters with smoothness regularization.

Comparing the change of matching distance between standard characters and target characters, we further prove our conclusions that multi point regression model can effectively utilize the local features of sample points, and synthesize characters with higher quality.

In the multiple points regression model, shape context features are extracted from the neighborhood of each sampled point as inputs of the neural network, and the distortion of adjacent points are regularized for shape smoothness. In order to illustrate the effect of smoothness constraint, we do a comparative experiment on 5 points regression model. Figure 5 shows the results of smoothness

constraint in five point regression model. Obviously, the generated characters appear to be more smooth and natural after adding constraints. This is obviously due to the constraints of the relative position between neighbor points prevent the occurrence of outliers during the sample points regression.

4.2 Generating Characters of Different Writing Styles

Finally, we compared the generated samples of different styles base on five-point regression model to validate the effectiveness of our algorithm. In the experiment, four personal-style handwriting sets were selected, we take one of them as the standard template and the other three as the target style. We take 300 pairs of templates as training samples for each style.

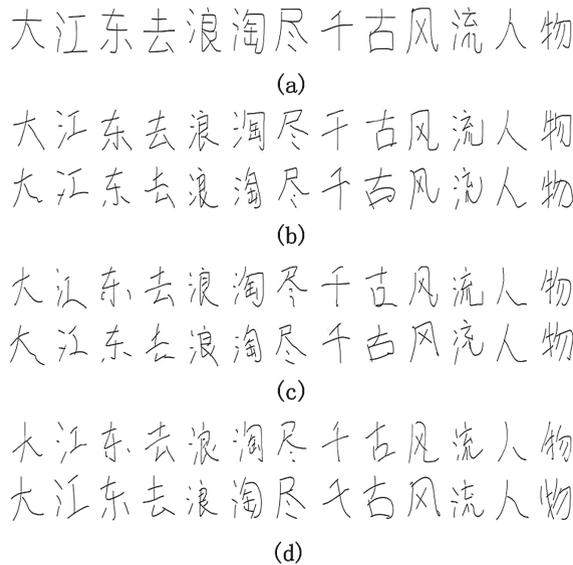


Fig. 6. Generated characters of three different personal styles. (a) Samples of standard template; (b), (c), (d) the target style characters and the synthetic characters, in every pictures, the first row represent the target character and the second row represent the synthetic characters.

Table 2. Average distance of matching points.

Style	Training set		Test set	
	D_{ori1}	D_{def1}	D_{ori2}	D_{def2}
Style1	0.12684	0.0501	0.1362	0.07354
Style2	0.1193	0.04458	0.1244	0.08225
Style3	0.1356	0.6257	0.1343	0.07903

According to the comparative experiments, we find that the generated characters are obviously different from the standard template, however, they have the same stroke features and structure characteristics as the target style. Figure 6 shows the generated results of different personal-style characters, none of these samples had appeared in the training set. Table 2 shows the matching distance of each style. The matching distance between standard characters and target characters become smaller after deforming which means that similarity between characters becomes higher. This is consistent with the observation of the human eye. The generated results proves that our model is effective in learning different styles with a small training set.

5 Conclusion

This paper proposes a novel learning-based approach to generate personal-style handwriting by style transformation. We learn the transition function between writing styles by predicting the displacement of the sample points. In order to synthesize high quality handwriting characters, we use shape context features as predictors, and regularize the distortion of adjacent points for shape smoothness. The experimental results demonstrated that our algorithm can learn the handwriting style and generate natural target style characters from a small number of training samples. However, in the course of the experiment, we also find the limitations of our method. For example, it was still a difficult task for us to simulate a writing style of rapid cursive. Besides, our algorithm is verified on the online character, further, we can try to extend this method to synthesis the writing trajectories of offline characters by adding the information of stroke width.

Acknowledgment. This work has been supported by the National Natural Science Foundation of China (NSFC) Grant No. 61573355.

References

1. Varga, T., Bunke, H.: Off-line handwritten textline recognition using a mixture of natural and synthetic training data. In: Proceedings of the 17th International Conference on Pattern Recognition, vol. 2, pp. 545–549 (2004)
2. Varga, T., Bunke, H.: Perturbation models for generating synthetic training data in handwriting recognition. In: Marinai, S., Fujisawa, H. (eds.) Machine Learning in Document Analysis and Recognition. SCI, vol. 90, pp. 333–360. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-76280-5_13
3. Wang, J., Wu, C., Xu, Y.-Q., Shum, H.-Y., Ji, L.: Learning-based cursive handwriting synthesis. In: Proceedings of Eighth International Workshop on Frontiers in Handwriting Recognition, pp. 157–162 (2002)
4. Wang, J., Wu, C., Xu, Y.-Q., Shum, H.-Y.: Combining shape and physical models for online cursive handwriting synthesis. *Int. J. Doc. Anal. Recogn. (IJDAR)* **7**(4), 219–227 (2005)

5. Guyon, I.: Handwriting synthesis from handwritten glyphs. In: Proceedings of the Fifth International Workshop on Frontiers of Handwriting Recognition, pp. 140–153 (1996)
6. Zong, A., Zhu, Y.: StrokeBank: automating personalized Chinese handwriting generation. In: AAAI, pp. 3024–3030 (2014)
7. Xu, S., Jin, T., Jiang, H., Lau, F.C.: Automatic generation of personal Chinese handwriting by capturing the characteristics of personal handwriting. In: AAAI (2009)
8. Shin, J., Suzuki, K.: Interactive system for handwritten-style font generation. In: Fourth International Conference on Computer and Information Technology, pp. 94–100 (2004)
9. Zhou, B., Wang, W., Chen, Z.: Easy generation of personal Chinese handwritten fonts. In: International Conference on Multimedia and Expo (ICME), pp. 1–6 (2011)
10. Lian, Z., Zhao, B., Xiao, J.: Automatic generation of large-scale handwriting fonts via style learning. In: SIGGRAPH ASIA 2016 Technical Briefs, p. 12 (2016)
11. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.* **89**(2), 114–141 (2003)
12. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(4), 509–522 (2002)