Proceedings of the World Congress on Engineering 2007 Vol I
WCE 2007, July 2 - 4, 2007, London, U.K.

Transformation Model Estimation for Point Matching Via Gaussian Processes

Xin Yu, Jinwen Tian, Jian Liu *

Abstract—**One of main issues in point matching is the choice of the mapping function and the computation of its optimal hyperparameters. In this paper, we propose an attractive approach to determine the mapping function based on Gaussian processes (GPs) model. The mapping function is assumed to belong to a GPs model specified by a mean and a covariance function. Meanwhile, hyperparameters optimization of mapping function is replaced by adaptation of GP model. Experiments show that the algorithm has efficient mapping capability and practical implementation in both synthetic and real cases.**

Keywords: ***point matching, geometric distortion, Gaussian processes, Bayesian inference***

1 Introduction

It is a fundamental yet still open problem in computer vision and image processing to match two point-sets to find the geometric mapping and correspondence between two sets of points in 2D or in 3D [1]. There are several possible choices for what type of geometric transformation model is used for characterizing the mapping of two given point-sets correspondences. Broadly one can dichotomize the types of geometric transformation models into the class of deterministic models and the class of statistical models. Deterministic models always need to exploit some prior knowledge about the distortion to choose a model of mapping function and determine optimal values of models parameters. Much effort has been expended in development of deterministic transformation [5]. The most frequently used deterministic mapping functions are polynomials of a lower degree. Higher order polynomials usually are not used in practical applications because they may introduce unnecessary warping. The most often used representatives of the kernel-based deterministic mapping functions are radial basis functions. Their name 'radial' reflects an important property of the function value at each point: it depends just on the distance of the point from the control points set, not on its particular position. Multiquadrics, reciprocal multiquadrics, Gaussians, Wendland's functions, and thin-plate splines (TPS) are several examples of the radial basis functions [2, 3, 4, 5]. For deterministic models, the task to be solved consists of choosing of the mapping function and estimating of its parameters. Unfortunately, because the distortion between the two point-sets is often unknown as prior, choosing a 'true' mapping function may not be a very simple step. Furthermore, estimation of the parameters is always solved to minimize the mean square error over control points, but this solution may be incorrect when the training data set has small size or contain some noise. Therefore, simple deterministic models may lack expressive power in the case of complex data sets, and their more complex counter-parts may not be easy to work with in practice.

Recently, statistical models have been studied as the geometric transformation models. In [6], the least square support vector machines (LS-SVM) had been proposed as an adaptive transformation model estimation approach. The approach is based on the structural risk minimization theory and can control the tradeoff between minimizing the error on the control points and minimizing the capacity of mapping functions. As mentioned by the authors, it is reasonable to believe that their approach should perform at least as well as the deterministic model as mentioned above. The drawback of this approach is that the kernel hyperparameters optimization procedure by cross validation is not a very efficient way in practice.

To address the concern for geometric mapping, we concern ourselves with statistical models to find an appropriate transformation model estimation method, which should have some properties as below: (1) excellent mapping capability even for complex case; (2) efficient computation of optimal values of model hyperparameters. In this paper, an attractive approach is proposed to determine the mapping function from given coordinates of two corresponding point-sets via GPs model. The GPs model provides a kernel machine framework and has the state of the art of performance for regression and classification [7, 8, 9]. Based on the idea of GPs model, we assume that the mapping function belongs to a prior GP, which does not depend on corresponding point-sets but specifies some properties of mapping function. Then, this prior GP is updated to a posterior GP in the light of known corresponding point-sets by Bayesian inference. The final actual mapping function is assuming to be one sample from the posterior GPs. Furthermore, the properties of

*Huazhong University of Science and Technology, State Key Laboratory for Multispectral Information Processing Technologies, Wuhan P. R. China 430074 Tel/Fax: 86-27-87556302 Email: uxinhenry@gmail.com

ISBN:978-988-98671-5-7

the prior GP can be learned in the light of known corresponding point sets by Bayesian inference. This learning procedure actually replaces the computation of optimal values of hyperparameters in deterministic models. The experimental results indicate that the new approach has the properties cited above.

2 Gaussian Process Model

A GP is fully specified by its mean function $m(x)$ and covariance function C_Θ . The GP prior over an unknown function can be written as:

$$p(f|X,\Theta) = \frac{1}{(2\pi)^{N/2}|C_\Theta|^{1/4}} \bullet \texttt{exp}\left\{-\frac{1}{2}(f-m)^T C_\Theta{}^{-1}(f-m)\right\} \quad (1)$$

or

$$f \sim \mathcal{GP}(m, C_\Theta)$$

where the mean is usually assumed to be zero $m = 0$ and each term c_{ij} of a covariance matrix $\mathbf{C}_N$ is a function of x_i and x_j, i.e. $c(x_i, x_j)$.

The GP prior will be used as a prior for Bayesian inference because of no training data information is incorporated into it. By updating this prior in the light of training data, a posterior GP can be inferred which can be used to make predictions for unseen test cases. Let assume we have been given a data set D of data point x_i with target value t_i: $D = \{(x_i, t_i), i = 1, 2, \ldots, N\}$. Given this data set, we wish to find the target value $\tilde{t}$ for a new data point $\tilde{x}$. The predicted value of $\tilde{t}$ and its stand deviation can be written respectively as:

$$\begin{aligned} \hat{t}_{N+1} &= \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}_N \\ \sigma^2_{\hat{t}_{N+1}} &= \kappa - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k} \end{aligned} \quad (2)$$

where $\mathbf{k}$ is a vector of covariance between every training case and $\tilde{x}$; $\mathbf{C}_N$ is the covariance matrix for training data set; κ is the self-variance for $\tilde{x}$.

Assuming that a form of covariance function has been chosen, but that it depends on undetermined hyperparameters. Then it attempts to 'learn' these hyperparameters from training data by Bayesian inference. The posterior distribution of the hyperparameters Θ is,

$$P(\Theta|t_N, X_N) \propto P(t_N|X_N, \Theta)P(\Theta) \quad (3)$$

The optimal value of hyperparameters Θ can be inferred by optimizing the marginal likelihood L based on its partial derivatives, which can be written respectively as:

$$L = -\frac{1}{2}\texttt{logdet}(\mathbf{C}_N) - \frac{1}{2}\mathbf{t}_N^T \mathbf{C}_N^{-1} \mathbf{t}_N - \frac{N}{2}\texttt{log}2\pi \quad (4)$$

$$\frac{\partial L}{\partial \theta} = -\frac{1}{2}\texttt{trace}(\mathbf{C}_N^{-1}\frac{\partial \mathbf{C}_N}{\partial \theta}) + \frac{1}{2}\mathbf{t}_N^T \mathbf{C}_N^{-1}\frac{\partial \mathbf{C}_N}{\partial \theta}\mathbf{C}_N^{-1}\mathbf{t}_N \quad (5)$$

Given marginal likelihood and its derivatives, it is straightforward to feed this information to an optimization package in order to obtain a local maximum of marginal likelihood. Other alternative to maximum likelihood estimation of the hyperparameters, such as cross-validation or generalized cross-validation method, would appear to be difficult when a large number of hyperparameters are involved.

3 Determining Mapping Function via GPs Model

In this section, we return to our concerned problem: determining the mapping function. Assume we have two sets of corresponding points (in either 2D or higher dimension) $P = (p_i, i = 1, 2, \ldots, K)$ and $Q = (q_i, i = 1, 2, \ldots, K)$ respectively. Each p_i represents a corresponding point coordinate $(p_i^1, p_i^2, \ldots, p_i^D)$ in one point-set and each represents a corresponding point coordinate $(q_i^1, q_i^2, \ldots, q_i^d)$ in another point-set. Obviously, it breaks down into d scattered data regression problems with the training data $(p_1, p_2, \ldots, p_K, q^1)$, …, $(p_1, p_2, \ldots, p_K, q^d)$ respectively. For the sake of simplicity, we only consider the first regression problem within two dimensions space, i.e. $d = 2$. The remaining problems can be solved similarly.

The idea of GPs modeling is, without parameterizing mapping function, to place a probability distribution directly on the space of functions. Based on the idea of GP model, we assume the geometric mapping function is distributed as posterior GP model, which implicitly means: "A form of mapping function is implicitly selected and the computation of optimal parameter for mapping model is replace by the procedure of adaptation of GP". Firstly, the mapping function is distributed as prior GP model, which does not depend on corresponding point-sets but specifies some properties of mapping function. This prior GP is fully specified by a mean and a covariance function. The mean is usually assumed to be zero. Based on experiments by Williams and Rasmussen [7], we choose the commonly used covariance function in our case, which can be written as:

$$\begin{aligned} c(p_i, p_j) &= \upsilon_0 \texttt{exp}\{-\frac{1}{2}\sum_{l=1}^{2}\alpha_l(p_i^l - p_j^l)^2\} \\ &\quad + a_0 + a_1\sum_{l=1}^{2} p_i^l p_j^l + \upsilon_1 \delta_{ij} \end{aligned} \quad (6)$$

The covariance function consists of three parts: the first term, a linear regression term and a noise term. The first term expresses the idea that cases with nearby inputs will have highly correlated outputs; the α_l parameters corresponding to each input characterizes the distance in that particular direction over which target value is expected to vary significantly. The value of α_l is related to the smooth degree of mapping function, the smaller the value of α_l, the more the smoothness of mapping function. The v_0 variable gives the overall scale of the local correlations. The values of a_0 and a_1 control the scale of the bias and linear contributions to the covariance. The variable v_1 is the variance of the noise.

Then, this prior GP is updated to a posterior GP in the light of known corresponding point-sets by Bayesian inference. The final actual mapping function is assuming to be one sample from the posterior GPs. In order to determine the surface, we construct the mapping function by institute the training data (p_i^1, p_i^2, q_i^1) into Eq. (2). The value at a new data point $\tilde{p}$ in mapping function is distributed as a Gaussian distribution, which can be written as:

$$\begin{aligned} f(\tilde{p}) &\sim \mathbf{N}(\mu(\tilde{p}), \sigma_{\tilde{p}}^2) \\ \mu(\tilde{p}) &= \mathbf{k}^T \mathbf{C}^{-1} \mathbf{t} \\ \sigma_{\tilde{p}}^2 &= \kappa - \mathbf{k}^T \mathbf{C}^{-1} \mathbf{k} \end{aligned} \tag{7}$$

where

$$\begin{aligned} \mathbf{k}^T &= (c(\tilde{p}, p_1), c(\tilde{p}, p_2), \ldots, c(\tilde{p}, p_K)) \\ \mathbf{C} &= c(p_i, p_j)_{K*K} \\ \mathbf{t} &= (q_1^1, q_2^1, \ldots, q_K^1)^T \\ \kappa &= c(\tilde{p}, \tilde{p}) \end{aligned}$$

By substituting the covariance function in Eq. (6) into Eq. (7), the posterior GPs model can be inferred very easily.

Furthermore, the properties of the prior GP can be learned in the light of known corresponding point sets by Bayesian inference. This learning procedure actually corresponds to the computation of optimal values of hyperparameters in a transformation model. To find best hyperparameters precisely, a gradient-based optimization package was used in order to obtain a local maximum of marginal likelihood. The most obvious implementation of the gradient computation is to evaluate the inverse of the covariance matrix exactly. This was done using a variety of method such as Cholesky decomposition, LU decomposition or Gauss-Jordan Elimination.

4 Experiments Results

To illustrate the properties of the proposed estimation technique, we present two series of experiments. In the first series of experiments, we conduct different degrees of warping on synthetic data to test the algorithm's performance on solving different deformations and compare the proposed method with the TPS. After a template point-set is chosen, we apply a randomly generated non-rigid transformation to warp it. Gaussian radial basis functions (RBF) are selected as the random transformation. By assuming the coefficients of RBF belong to a Gaussian distribution with a zero mean and increasing the standard deviation, larger deformation can be generated. The errors are computed as the mean square distance between the warped template using GP and TPS and the warped template using the ground truth Gaussian RBF. 150 random experiments were repeated for each degree deformation. The template that we choose comes from a Chinese character (blessing) [3]. In GP model, twelve runs were performed with different initial conditions. For each run the initial values of the hyperparameters were sampled from their priors and then a conjugate gradient optimization routine was used to solve the Eq. (4). Substituting these hyperparameters values back into Eq. (7) and computing and rounding mapping functions at each point coordinate, we obtain the point matching result. Some of the experiments are shown in Fig. 1. The error means and standard deviations are shown in Fig. 2. It is obvious that the GP algorithm has better performance than TPS when the degree of deformation becomes larger.

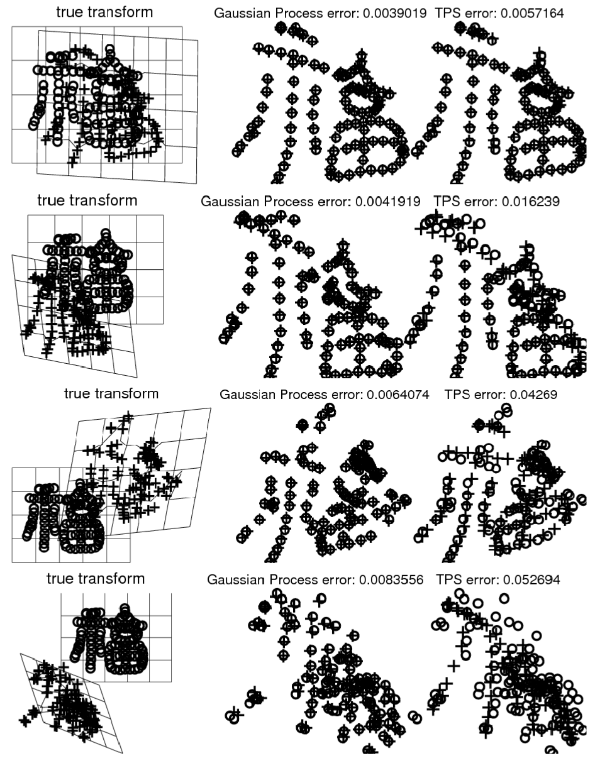

Figure 1: Synthetic Experiment Examples. Each row shows one example. Each row includes three parts arranged from left to right. First: Template and target (the warped template); Second: GP result; Third: TPS result.

In medical image registration, non-grid geometric transformation is always considered. We have applied the proposed algorithm to real sagittal images registration problem. As shown in Fig. 3, one can find that obvious deformation is contained between the two images. To determine mapping function, 35 pairs of corresponding points were selected manually or automatically [4]. Then

ISBN:978-988-98671-5-7

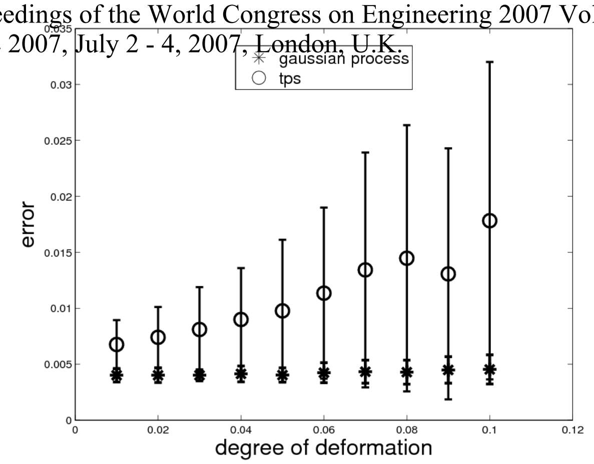

Figure 2: Statistics of the Synthetic Experiments.

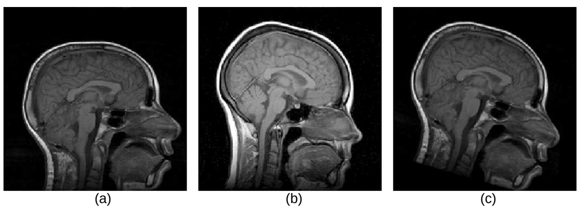

Figure 3: Registration of a pair of sagittal images from two different patients. (a) sensed image; (b) reference image; (c) registration result using the proposed method.

the proposed algorithm was used to remove geometric deformation. The result is shown in Fig. 3. It can be seen that there is a significant shift in the registration result of sensed image and the visual difference between the reference image and the registration result is reduced. The average error at the corresponding points was only 0.02 pixels.

5 Conclusions

We have considered the transformation model estimation problem in point matching. Our aim is to use Gaussian processes model to solve the problem. An interesting characteristic of our approach is that the choice of mapping function and the hyperparameters optimization are both integrated into a Bayesian framework within a GPs model. Another key characteristic is that the hyperparameters of covariance function can be optimized by maximizing the evidence from given point-sets correspondences. These characteristics do not appear in other kernel based learning methods such as TPS and LS-SVM. The experimentation results demonstrate the good capacities of this approach on both synthetic point set with varying degrees of deformation, and real images

References

[1] Grimson, W.E.L., *Object Recgonition by Computer: The Role of Geometric Constraints*, Cambridge, MA: MIT Press, 1990.

[2] Georgescu, B., Meer, P., “Point matching under large image deformations and illumination changes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, V26, pp. 674-688, 2004.

[3] Chui, H., Rangarajan, A., “A New Point Matching Algorithm for Non-Rigid Registration,” *Comput. Vis. Image Underst.*, V89, pp. 114-141, 2003.

[4] Rohr, K., Stiehl, H.S., Buzug T.M., Weese, J., Kuhn, M.H., “Landmark based elastic registration using approximating thin-plate splines,” *IEEE Trans. Med. Imag.*, V20, pp. 526-534, 2001.

[5] Zitova, B., Flusser, J., “Image registration methods: a survey,” *Image Vis. Comput.*, V21, pp. 977-1000, 2003.

[6] Peng, D.Q., Liu, J., Tian, J.W., Zheng, S., “Transformation model estimation of image registration via least square support vector machines,” *Pattern Recog. Letters*, V27, pp. 1397-1404, 2006.

[7] Rasmussen, C.E., Williams, C.K.I., *Gaussian Processes for Machine Learning*, Cambridge, MA: MIT Press, 2006.

[8] MacKay, D.J.C., “Introduction to Gaussian processes,” *Neural Netw. Mach. Learning*, V168, pp. 133-165, 1998.

[9] Williams, C.K.I., Barber, D., “Bayesian classification with Gaussian processes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, V20, N12, pp. 1342-1351, 1998.