Kernel Learning Estimation: A Model-Free Approach to Tracking Randomly Moving Object

Yuankai Li, Yuan Wang, Xiaosu Tan, and Jiaxin Lou

Abstract Kernel learning estimation (KLE) is a kernel-based method, where the original spatial data is mapped into a high-dimensional Hilbert space by a nonlinear mapping, hiding the nonlinear mapping in a linear learning framework. The kernel function of the method can be used to replace the complex inner product operation in the high-dimensional space and avoid the Curse of Dimensionality caused by high-dimensional calculation effectively. The kernel-based method has advantages on learnability, computational complexity, precise linearization and generalization performances, providing a promising way to solve the problem of nonlinear target tracking. In traditional tracking methods, nonlinear tracking models are usually built as a priori to predict the current state of target motion, emphasizing on tracking accuracy and real-time performance. However, kernel-based method provides a general way of linearization processing, which can be independent of specific models to achieve highly efficient data-driven computation. Introducing the kernel learning mechanism into target tracking problem is expected to improve the environmental adaptability. In this paper, a review on kernel learning method with application to randomly moving target tracking is presented, including kernel-based algorithms for target detection, kernel-based algorithms for generative tracking and for discriminant tracking, and multi-kernel learning methods with multiple kernel functions. Further research is prospected in optimization of kernel function, long-term robust tracking, feature extraction, target occlusion and other potential aspects on moving target tracking using kernel learning theory.

Y. Li (✉) · Y. Wang · X. Tan · J. Lou
University of Electronic Science and Technology of China, Chengdu, China
e-mail: yuankai.li@uestc.edu.cn

Y. Wang
e-mail: 463932587@qq.com

X. Tan
e-mail: 2495646072@qq.com

J. Lou
e-mail: 2801197250@qq.com

M. A. Chaurasia and C.-F. Juang (eds.), *Emerging IT/ICT and AI Technologies Affecting Society*, Lecture Notes in Networks and Systems 478,
https://doi.org/10.1007/978-981-19-2940-3_4

Keywords Kernel learning method · State estimation · Pattern recognition · Target tracking · Target detection

1 Introduction

Kernel learning estimation (KLE) method, which is built based on kernel function and statistical learning theory, is a newly developing branch of pattern recognition and applied to the techniques of support vector machines (SVMs) originally [1, 2]. The kernel-based method maps the data from original space into a high-dimensional feature space by using nonlinear mapping function, transforming nonlinear problem into linear analysis in sample space. It is an effective method to deal with nonlinear pattern recognition problems. Also, the kernel-based method can provide a way of efficient calculation. By using the kernel function, it can hide the nonlinear mapping in a linear learning machine for synchronous calculation and replace complex inner product operation in high-dimensional space that provides a new way to solve the problem of high-computational complexity in high-dimensional space, avoiding the Curse of Dimensionality.

Driven by emerging technologies, techniques of target tracking are developing rapidly, related to many disciplines such as pattern recognition, dynamic systems and artificial intelligence. It has already achieved significant applications in intelligent decision-making [3], unmanned driving [4–6], biomedicine [7], human–computer interaction, reconnaissance and other application areas, and still has great development prospects. The problem of moving target tracking essentially is a typical nonlinear pattern recognition problem. The tracking model is established through the prior knowledge of moving target, followed by predicting the motion state or motion pattern of the target at the current time or frame. However, quality of the tracking model directly impacts on the tracking performances. Deviation of the tracking model, such as mismatching of the target motion state or unreasonable setting of the searching area, may lead to remarkable tracking errors or target losses.

The moving target tracking algorithm based on kernel learning method, by introducing nonlinear mapping structure with a learning mechanism, is independent of the specific model and can guarantee tracking quality under the nonlinear change of the target motion state, giving the tracking algorithm better adaptability to the environment.

This paper presents a review on the moving target tracking algorithms based on kernel learning methods, including discussions on the target detection algorithms, generative and discriminant target tracking algorithms using kernel methods with their research progress. For design and optimization of the kernel functions, multi-kernel learning method with typical structures of multiple kernel functions is also discussed. In the end, a prospect on research and applications of the kernel learning target tracking method is provided, with development trends in target-scale change, kernel optimization, accuracy of feature extraction and other critical problems.

The rest of the paper is organized as follows. Section 2 presents the fundamental principle of nonlinear kernel mapping. Kernel learning estimation with application to moving target tracking is discussed in detail in Sect. 3, where target detection, generative and discriminant tracking algorithms are shown, respectively. In Sect. 4, multiple kernel learning estimation is presented. Section 5 gives a discussion and prospect to the future research on kernel learning method, and Sect. 6 concludes the paper.

2 Nonlinear Kernel Mapping

The key idea of the kernel-based method is to map the data from the original space to a reproducing kernel Hilbert space (RKHS) through a fixed nonlinear mapping operator, performing data processing in a high-dimensional feature space.

Define the nonlinear mapping operator as kernel function κ. For all x and x', the kernel function satisfies

$$\kappa\left(x, x'\right) = \varphi(x), \varphi\left(x'\right) \tag{1}$$

where $\varphi(\cdot)$ is a feature vector transformed from original space to feature space.

When the kernel function is continuous, positive definite and symmetric, it is called Mercer kernel [8]. If the kernel function meets the following two conditions:

(1) For any vector of $x \in X$, $\kappa(x, z)$, its function belongs to the vector space F;
(2) $\kappa(x, z)$ is reproducible.

It is called a reproducible kernel function. A certain reproducible kernel function defines a reproducible Hilbert space. For a reproducible kernel function, the kernel $\kappa(x, \cdot)$ composes the function

$$g(\cdot) = \sum_{i=1}^{l} a_i \kappa(c_i, \cdot) \tag{2}$$

and for all i,$c_i \in X$, there is

$$g, \kappa(x, \cdot) = \sum_{i}^{l} a_i \kappa(c_i, x) = h(x) \tag{3}$$

Through definition of Mercer kernel, the reproducible kernel can be expressed as

$$\kappa\left(x, x'\right) = \sum_{i=1}^{\infty} \varsigma_i \varphi_i(x) \varphi_i\left(x'\right) \tag{4}$$

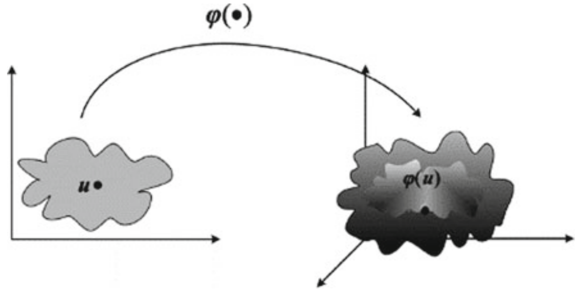

Fig. 1 Schematic diagram of the nonlinear mapping $\varphi(\cdot)$

where ς_i and φ_i are nonnegative feature value and vector, respectively. The mapping φ can be expressed as

$$\begin{cases} \varphi : X \to F \\ \varphi(x) = \left[\sqrt{\varsigma_1}\varphi(x_1), \sqrt{\varsigma_2}\varphi(x_2), \ldots\right] \end{cases} \tag{5}$$

For data u in the original input space, $\varphi(u)$ can be obtained by mapping $\varphi(\cdot)$ into the feature space. Figure 1 shows the mapping relationship of the data from the original space to the Hilbert space.

In Hilbert space, $\varphi(\cdot)$ is the mapping from the original space to the feature space, and $\varphi(x)$ is the mapped feature vector. For $x, x' \in X$, the kernel function satisfies

$$\varphi(x)^T\varphi\left(x'\right) = \kappa\left(x, x'\right) \tag{6}$$

That is, when the mapping function $\varphi(\cdot)$ is difficult to express explicitly, the calculation of the kernel function is used to replace the complex calculation to the inner product of the nonlinear mapping in the RKHS, thereby simplifying the calculation process and improving the calculation efficiency.

3 Moving Target Tracking Using Kernel Learning Estimation

Target tracking is a challenging branch in the field of intelligent systems. It aims to realize the recognition and decision-making of target motion modes. It has broad research and application prospects and has been applied in intelligent monitoring, unmanned driving, biological medicine, human–computer interaction, military and other fields. Kernel method is deeply applied in the target tracking technique due to its linearization, data driven and other advantages, and has achieved fruitful results.

Target tracking includes three main parts: appearance model, motion model and search strategy, involving image processing, data processing and machine learning. Considering that target detection is a prerequisite of target tracking, we describe the process of moving target tracking as the following general steps: detecting directive information of target motion, predicting target movement trend, and performing higher-level behavior analysis and decision-making for the randomly moving target.

3.1 Target Detection

The quality of target detection determines the performance of target tracking. Presently, target detection algorithms based on kernel methods mainly include three types of algorithms: kernel principal component analysis (KPCA), kernel Fisher linear discriminant (KFLD) and kernel independent component analysis (KICA).

Inspired by the kernel mapping theory, Scholkopf combined it with the principal component analysis (PCA) and proposed the KPCA [9] algorithm in 1998, allowing the kernel method break away from support vector machine to cooperate with other algorithms for the first time. That is of great significance for the kernel method been successfully applied to the field of target feature extraction and target detection.

Fisher linear discriminant (FLD), also known as linear discriminant analysis (LDA), was proposed by Mika [10] in 1999. The principle of it is to find a most suitable projection axis, so that the projection distance between all kinds of samples on this axis is as far as possible, and the projections of samples in each category are as compact as possible, intending to achieve the best classification effect. Also, Mika extended the FLD with kernel method and proposed the KFLD algorithm to solve the nonlinear problems. A feature space is established to judge the linear Fisher criterion for KFLD, which is helpful to obtain more accurate result for target feature extraction under nonlinear conditions.

In 2002, Bach, based on independent component analysis (ICA), proposed the KICA [11] algorithm, which further enriched kernel detection methods. KICA deals with functions in the kernel Hilbert space based on the entire nonlinear function space and uses kernel mapping technique to search the space with high efficiency. The use of function space enables it to adapt to a variety of data samples, thus making the algorithm more robust to time-variant sample distribution.

3.2 Generative Tracking

Generative tracking is a type of tracking method in which online learning is used to model the target feature and through the feature model to search and match and find the target position. A representative generative tracking method is mean shift (MS) [12].

3.2.1 Representative Algorithm: Mean Shift

During the initial stage of tracking, the MS algorithm needs to determine the search window independently to select the target, therefore, it is a semi-automatic tracking method. The algorithm calculates the histogram distribution of the initial frame search window and calculates the histogram distribution of the Nth frame window in the same way, making the search window to move along the direction of the maximum histogram density, and finally gets the position of the target. The standard steps are as follows:

(1) Target model of the initial frame

Divide the feature space into multiple eigenvalues according to pixel color values. At the initial frame, the probability of the u-th eigenvalue in the search window of the initial frame is

$$\tilde{q}_u = C\sum_{i=1}^{n} k\left(\frac{x_0 - x_i{}^2}{h}\right)\delta[b(x_i) - u] \tag{7}$$

where x_0 and x_i are the central pixel coordinates of the $N\ th$ frame search window and the coordinates of the i-th pixel, respectively; $k(\cdot)$ is the kernel function, and h is the bandwidth of the kernel function.

(2) Target model of the N-th frame

Calculate the probability of the u-th eigenvalue in search window of the N-th frame with

$$\tilde{p}_u(y) = C_h\sum_{i=1}^{n_k} k\left(\frac{y_0 - x_i{}^2}{h}\right)\delta[b(x_i) - u] \tag{8}$$

where y_0 is the coordinate of the pixel in the center of the search window.

(3) Similarity function

Similarity function represents degree of similarity between the initial frame model and the N-th frame model. Define the function as

$$\tilde{\rho}(y) \equiv \rho(\tilde{\rho}(y), \tilde{q}) = \sum_{u=1}^{m} \sqrt{\tilde{\rho}_u(y)\tilde{q}_u} \tag{9}$$

(4) Mean shift vector

By maximizing the similarity function, the mean shift vector can be obtained as

$$m_{h,G}(y) = y_1 - y_0\frac{\sum_{i=1}^{n_k} x_i w_i g\left(\frac{\tilde{y}_0 - x_i}{h}{}^2\right)}{\sum_{i=1}^{n_k} w_i g\left(\frac{\tilde{y}_0 - x_i}{h}{}^2\right)} - y_0 \tag{10}$$

in which y_0 is the central coordinate of the search window and y_1 is that of the found new search window.

3.2.2 Related Studies

The mean shift algorithm was first proposed by Fukunaga [12] in 1975. It was not widely concerned until 1995 when Cheng introduced the kernel function and weight coefficient [13] to the original MS and extended the MS theory. Comaniciu applied the MS algorithm to video image processing and realized smoothing, segmentation and tracking of visual objects [14–17].

Usually, the bandwidth of the kernel function in MS tracking algorithm is fixed, leading to large tracking errors when the target size varies. For this reason, based on scale space theory [18], Coliins proposed a scale variable MS tracking algorithm [19]. Sometimes, the color feature map in the MS algorithm cannot accurately describe the distribution of the target color in the image, leading to vacancy of the feature space. To solve this problem, the spatial color histogram is introduced into the target tracking model [20], combining the target color with the image color in the current frame to improve the accuracy of target recognition. Also, to improve the anti-occlusion ability of MS algorithm, filtering algorithms are integrated into the MS [21]. To track multi-target objects and human joints, confidence fusion propagation algorithm is introduced into the MS algorithm [22] for realizing tracking of single target, multi-target and human joints.

With good performance on processing time and robustness, MS algorithm has achieved rich results in the field of target tracking for several decades. However, in the generative tracking methods, environmental information is usually ignored so that the entire tracking process is quite easy to be disturbed. It yields a key problem of how to extract the target from background effectively that motivates the studies of discriminant tracking.

3.3 Discriminant Tracking

Discriminant tracking uses the correction theory of machine learning to model the target feature and background feature, respectively, taking the peak in the confidence map as target position and training corresponding filter to distinguish the target from background. It has two representative classes of kernel-based algorithms: circulant structure of tracking-by-detection with kernels (CSK) and kernelized correlation filters (KCF).

3.3.1 Representative Algorithm I: CSK

Correlation filter has a fast operation rate in frequency domain, and the tracking speed can reach hundreds of frames per second. It is an effective way to realize real-time moving target tracking. In 2010, Bolme [23] introduced correlation filtering into minimum output sum of square error (MOSSE) algorithm, expanding target tracking from time domain to frequency domain. Based on the MOSSE, Henriques [24] proposed the CSK algorithm. It uses correlation theory of cyclic matrix and shift the training samples and candidate samples cyclically to increase the number of two samples, meanwhile, use the training samples to train the classifier and then use the trained classifier to detect target in the candidate region which is constructed by the candidate samples, and finally, take the maximum response of the correlation calculation as the target location. Kernel mapping function is adopted for the CSK to fasten the calculation speed.

To find the response position, CSK algorithm needs to transform from frequency domain to time domain, which will lead to marginal effect similar with the leakage phenomenon of aperiodic signals in systems. Moreover, application range of CSK algorithm is narrow, and it is only suitable for gray scale feature space. It lacks the establishment of target appearance model and is easily disturbed by complex task background and environmental noises. In order to establish the target appearance model more accurately, Danelljan et al. use the color name (CN) [25] based on the CSK algorithm to enhance the description of the appearance model, mapping three-dimensional color space into a high-dimensional CN space. In the way, the algorithm can more effectively establish the target appearance model and has a better tracking effect on the problem of target partial occlusion, light change, complex background, etc. However, due to the complexity of computing in a high-dimensional space, the calculation rate of CSK decreases after introducing three-dimensional color space.

When the target scale is changed or occluded, the CSK algorithm will degenerate obviously. On the basis of the algorithm, other target tracking algorithms based on correlation filter and their improved algorithms are proposed, which have achieved better tracking effect and tracking robustness than CSK algorithm.

3.3.2 Representative Algorithm II: KCF

To further improve tracking accuracy and tracking precision, Henriques proposed the kernelized correlation filters [26] in 2014. The KCF algorithm mainly includes four typical steps that are ridge regression [27], diagonalization of cyclic matrix, kernel correlation filtering and fast target detection.

Ridge Regression

For linear regression model, the objective function is

$$f(x) = W^T X \tag{11}$$

Find the weight W_i in sample X to minimize the prediction of target Y_i, that is

$$\min_w \sum_i (f(x_i) - y_i)^2 + \lambda w^2 \tag{12}$$

where λ is the parameter that controls over-fitting. Take the derivative of (12) and set it to zero, and then, the optimal solution is obtained as

$$w = \left(X^H X + \lambda I\right)^{-1} X^H y \tag{13}$$

where X is the sample matrix, and y is the label.

Cyclic Matrix Calculation

Note $X_{n\times 1} = \{x_1, x_2, \ldots x_n\}^T$ as the positive sample and generate the training sample (negative sample) by cyclic shifting to the positive sample. Use the negative sample to train the classifier, and one-dimensional operator P in cyclic shifting is

$$P = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{14}$$

By multiplying the vector x constantly with the operator P in cyclic shifting, for one-dimensional image, a cyclic matrix $C(x)$ with combinations of n vectors is obtained. For two-dimensional image, training sample of the two-dimensional image is obtained by cyclic shifting in the region of interest, and finally, the cyclic matrix of the two-dimensional matrix is obtained.

Kernel Correlation Filtering

To improve the classifier performance, map the original spatial data into a high-dimensional space with weight w that is a linear combination of input samples as

$$w = \sum_i \alpha_i \varphi(x_i) \tag{15}$$

By introducing $\varphi^T(x)\varphi\big(x'\big) = k\big(x, x'\big)$ for kernelization, the optimal solution of (15) can be transformed into

$$\alpha = (K + \lambda I)^{-1} y \tag{16}$$

where K is also a cyclic matrix. By using the properties of the cyclic matrix, and (16) can be rewritten as

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \tag{17}$$

where k^{xx} is the first row of K and $\hat{y}$ is the discrete Fourier transform of y. In general, the classifier can be solved by calculating $\hat{\alpha}$.

Fast Target Detection

Suppose that the target state of the previous frame is x, and the input sample of the current frame is z, and the kernel matrix K is introduced to obtain the response of the test sample, that is

$$\hat{f}(z) = \hat{k}^{xz} \odot \hat{\alpha} \tag{18}$$

Performing inverse Fourier transform to (18) obtains the sample response. Take the maximum response value as the target position, and update the classifier parameters and the target model by

$$\begin{cases} x^i = (1-\theta)x^{i-1} + \theta z^i \\ \alpha^i = (1-\theta)\alpha^{i-1} + \theta \sigma^i \end{cases} \tag{19}$$

where x^i is the target position model and α^i is the classifier parameters predicted for the i frame, respectively; z^i and σ^i are the detected target position model and classifier parameters, respectively; θ represents the learning rate.

3.3.3 Related Studies

Traditional target tracking algorithms, such as mean shift algorithm, support vector machine [28], Kalman filtering [29], particle algorithm based on Bayesian filtering [30, 31], tracking learning detection (TLD) [32] based on target characteristics, and compressive tracking (CT) [33] algorithm based on compression perception, appear deficiencies in tracking performance and in computation efficiency. KCF algorithm, absorbing advantages of CSK, shows high tracking precision and calculation speed with improved classifier training and sample detection.

To solve the problems of scale change, target occlusion and target loss in the process of target tracking, a tracking method with multi-scale correlation filters is proposed in [34], where two ridge regression models having strong plasticity and strong stability are adopted. The model with good plasticity tracks the target position to construct image pyramid, putting the position as the center. The model with good stability predicts the target-scale change, realizing multi-scale detection and tracking. In [35], a depth scaling kernelized correlation filtering (DSKCF) algorithm is developed, which extends the RGB tracking algorithm in KCF to form an RGB-D tracking algorithm by integrating depth features. The algorithm gives an improved solution for the problems of target-scale change and target occlusion. To deal with the marginal effect caused by cyclic shift in KCF algorithm, Danelljan introduced the spatial setting factor to constrain the weight of filters, alleviating the marginal effect at the expense of calculation complexity of the algorithm [36].

For many complex scenes, it is usually difficult for a single kernel to satisfy the requirements of various tracking performances, and multi-kernel fusion for the KCF algorithm provides an extension scheme.

4 Multiple Kernel Learning

Kernel function design is an important part of the kernel-based target tracking algorithm, which determines the tracking accuracy. To improve the tracking adaptivity to complex environments, fusion of multiple kernel functions is studied recently. Many works [37–39] have shown that multi-kernel model has better flexibility and robustness than model with single kernel. Presently, multi-kernel learning methods can be categorized into three main types, which are synthetic kernel, multi-scale kernel and infinite kernel.

Synthetic kernel method is the fundamental method of multi-kernel learning. One can use many ways to construct the synthetic kernel, such as multi-kernel linear combinatorial synthesis method, multi-kernel extended synthesis method, non-stationary multi-kernel learning, local multi-kernel learning and non-sparse multi-kernel learning. Due to simplicity of the synthetic kernel method, it has been applied in many practical problems, such as target eigenvalue extraction and processing [40, 41], classification [42–45], image segmentation [46] and system identification [47].

Synthetic kernel method generally uses linear combination of kernel functions to generate an integrated kernel function, which is questionable for desired tracking accuracy when facing unbalanced distribution of samples. Multi-scale kernel method introduces the concept of scale space and merges multi-scale kernel functions into a more flexible new kernel function. The multi-scale kernel method needs to train bandwidth for each kernel function. The kernel function with smaller bandwidth is used to track samples with drastic changes, and the function with larger bandwidth is to track samples with gentle changes. Using SVM regression or RKHS to extend the multi-scale kernel method is beneficial to improve the tracking accuracy. The multi-scale kernels are classified in [48], where a typical multi-scale kernel method is provided.

The aforementioned two kernel methods use limited number of kernel functions for linear combination or fusion. However, for large-scale data processing, the multi-kernel processing methods with limited number of kernels become hard to achieve desired tracking performance with optimal decision. Therefore, a trend of expanding from finite to infinite kernel functions for multi-kernel learning emerges recently, but there are few reports about its application in the field of target tracking.

5 Discussion and Prospect

Presently, target tracking method based on kernel learning estimation is developing fast. Compared with other types of tracking methods, kernel-based target tracking is free of nonlinear error and promising to be model-free and data-driven, however for future use, it still needs further research to satisfy requirements under practical conditions. The potential work is mainly reflected in the following aspects.

(1) Reduce the dependency to tracking model.

Traditional Bayesian tracking algorithms strongly depend on a prior math model and model parameters. Introducing kernel method into classical tracking algorithms can expand the algorithm to nonlinear applications precisely. With kernel function replacing inner product operation in sample space and target dynamics modeled by trained measurements, high-dimensional computing complexity and dependency on tracking model can be reduced greatly.

(2) Improve the accuracy of feature extraction.

The quality of feature extraction is the key factor affecting the efficiency of visual target tracking. Convolution features have advantages over artificial features, but the design of training network still needs further research. Multi-feature fusion is helpful to describe the target much more precisely, but the increasing computation complexity needs to be controlled to balance tracking accuracy and response speed.

(3) Improve the balance of base-kernel design.

Single kernel function is difficult to obtain satisfied performances in complicated tracking tasks. Multi-kernel learning method can optimize kernel function through combination of multiple base kernels and adjust weight coefficients and parameters of the base kernels to deal with complex application scenarios. However, multi-kernel learning method increases computational complexity. Learning efficiency is also necessary to be considered in algorithm design, and the balance of multi-kernel learning framework also needs for further exploration.

(4) Improve the adaptability to environmental dynamics.

Changes and occlusion of the target will decrease the tracking accuracy or lead to target missing. It is significant for tracking algorithm to maintain high precision of tracking when target shape or scale is changing, flipping or occluded. Trajectory prediction and target blocking are effective to improve environmental adaptability.

(5) Improve the persistence of stable tracking.

For long-term tracking of random-moving target, uncertainties have much more chance to appear, leading to weak robustness and decreasing tracking accuracy. The algorithms for short-term tracking are generally lack of framework from long-term tasks to construct continuity of stable tracking, which is meaningful to be addressed in further research.

6 Conclusion

Kernel learning estimation method gives a feasible bridge from linear to nonlinear estimation problems and forms a unified framework of solutions in pattern analysis fields. By applying nonlinear kernel mapping to the tracking problem and replacing complicated inner product operation by designed kernel functions, kernel-based target tracking method can achieve good performance in capabilities of nonlinear processing, data-driven and generalization. The method has significant research prospects and potential applications in decision intelligence, pattern recognition, autonomous unmanned vehicles and other information processing systems.

References

1. Juang CF, Chiu SH, Chang SW (2007) A self-organizing TS-type fuzzy network with support vector learning and its application to classification problems. IEEE Trans Fuzzy Syst 15(5):998–1008
2. Chen GC, Juang CF (2013) Object detection using color entropies and a fuzzy classifier. IEEE Comput Intell Mag 8(1):33–45
3. Smeulders AWM, Chu DM, Cucchiara R et al (2014) Visual tracking: an experimental survey. IEEE Trans Pattern Anal Mach Intell 36(7):1442–1468
4. Wang D (2017) Design of intelligent video surveillance system based on motion detection. North University of China, Taiyuan
5. Gündüz G, Acarman AT (2018) A lightweight online multiple object vehicle tracking method. In: Proceedings of IEEE intelligent vehicles symposium, Changshu, China, pp 427–432
6. Lu Y, Dai H, Hu Y et al (2020) Research on collaborative target tracking algorithm of UAV based on machine vision. Electronic Devices 43(5):1096–1099
7. Gao T (2012) Medical image analysis based on multi-target tracking. Xidian University
8. Aronszajn D (1950) Theory of reproducing kernel. Trans Am Math Soc 68:337–404
9. Scholkopf B, Smola AJ, Muller KR (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10(5):1299–1319
10. Mika S, Ratsch G, Weston J et al (1999) Fisher discriminant analysis with Kenels. In: Proceedings of IEEE signal proceeding society workshop, pp 41–48

11. Bach FR, Jordan MI (2003) Kernel independent component analysis. In: Proceedings of IEEE international conference on acoustics, speech, and signal processing, HK, China
12. Fukunaga K, Hostetler L (1975) The estimation of gradient of a density function with applications in pattern recognition. IEEE Trans Inf Theory 21(1):32–40
13. Cheng Y (1995) Mean shift, mode seeking, and clustering. IEEE Trans Pattern Anal Mach Intelli 17(8):790–799
14. Comaniciu D, Ramesh V, Meer P (2000) Real-time tracking of non-rigid objects using mean shift. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 142–149
15. Comaniciu D, Meer P (1999) Mean shift analysis and application. In: Proceedings of the IEEE international conference on computer vision, pp 1197–1203
16. Comaniciu D, Meer P (2002) Mean shift: a robust application toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619
17. Comaniciu D, Meer P (1997) Robust analysis of feature spaces: color image segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 750–755
18. Lindeberg T (1998) Feature detection with automatic scale selection. Int J Comput Vision 30(2):194–203
19. Collins RT (2003) Mean shift blob tracking through scale space. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 234–240
20. Birchfield ST, Rangarajan S (2005) Spatiograms versus histograms for region-based tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1158–1163
21. Welch G, Bishop G (2006) An introduction to the Kalman filter,UNC-Chapel Hill, NC
22. Park M, Liu Y, Collins R (2008) Efficient mean shift belief propagation for vision tracking. In: Proceedings of the IEEE conference computer vision and pattern recognition
23. Bolme DS, Beveridge JR, Draper BA et al (2010) Visual object tracking using adaptive correlation filters. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition, pp 2544–2550
24. Henriques JF, Caseiro R, Martins P et al (2012) Exploiting the circulant structure of tracking-by-detection with kernels. In: Proceedings of European conference on computer vision, pp 702–715
25. Danelljan M, Shahbaz KF, Felsberg M et al (2014) Adaptive color attributes for real-time visual tracking. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 1090–1097
26. Henriques JF, Caseiro R, Martins P et al (2015) High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596
27. He X (2005) Multivariate linear model and ridge regression analysis. Huazhong University of Science and Technology, Wuhan
28. Liang CW, Juang CF (2015) Moving object classification using local shape and HOG features in wavelet-transformed space with hierarchical SVM classifiers. Appl Soft Comput 28:483–497
29. Liang CW, Juang CF (2015) Moving object classification using a combination of static appearance features and spatial and temporal entropies of optical flows. IEEE Trans Intell Transp Syst 16(6):3453–3464
30. Arulampalam MS, Maskell S, Gordon NA (2002) Tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans Signal Process 50(2):174–188
31. Juang CF, Chang CW, Hung TH (2021) Hand palm tracking in monocular images by fuzzy rule-based fusion of explainable fuzzy features with robot imitation application. IEEE Trans Fuzzy Syst 29(12):3594–3606
32. Kalal Z, Mikolajczyk K, Matas J (2011) Tracking-learning-detection Kernel. IEEE Trans Pattern Anal Mach Intell 34(7):409–422
33. Zhang K, Zhang L, Yang MH (2012) Real-time compressive tracking. In: Proceedings of European conference on computer vision, pp 864–877
34. Xia X, Zhang X, Li J (2017) Kernel correlation filter target tracking method combined with scale prediction. Electronic Design Eng 25(2):130–136

35. Hannuna S, Camplani M, Hall J et al (2019) DSKCF: a real-time tracker for RGB-D data. J Real-Time Image Proc 16(5):1439–1458
36. Danelljan M, Hager G, Khan FS et al (2015) Learning spatially regularized correlation filters for visual tracking. In: Proceedings of IEEE international conference on computer vision, pp 4310–4318
37. Gonen M, Alpaydin E (2011) Multiple Kernel learning algorithms. J Mach Learn Res 12:2211–2268
38. Lanckriet G, Cristianini N, Bartlett P et al (2004) Learning the kernel matrix with semidefinite programming. J Mach Learn Res 5(1):27–72
39. Lee WJ, Verzakov S, Duin RP (2007) Kernel combination versus classifier combination. In: Proceedings of the international workshop on multiple classifier systems. Czech Republic, Prague, pp 22−31
40. Mak B, Kwok JT, Ho S (2004) A study of various composite kernels for kernel eigenvoice speaker adaptation. In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing. Montreal, Canada, pp 325−328
41. Fu SY, Yang GS, Hou ZG, Liang Z, Tan M (2008) Multiple kernel learning from sets of partially matching image features. In: Proceedings of the international conference on pattern recognition
42. Zheng S, Liu J, Tian JW (2005) An efficient star acquisition method based on SVM with mixtures of kernels. Pattern Recogn Lett 26(2):147–165
43. Fung G, Dundar M, Bi J, Rao B (2004) A fast iterative algorithm for fisher discriminant using heterogeneous kernels. In: Proceedings of the international conference on machine learning. Banff, Canada, pp 40−47
44. Damoulas T, Girolami MA (2009) Combining feature spaces for classification. Pattern Recogn 42(11):2671–2683
45. Damoulas T, Girolami MA (2009) Pattern recognition with a Bayesian kernel combination machine. Pattern Recogn Lett 30(1):46–54
46. Gustavo V, Luis C, Jordi M et al (2006) Composite kernels for hyperspectral image classification. IEEE Trans Geosci Remote Sens Lett 3(1):93–97
47. Gustavo V, Manel R, Rojo-Alvarez J et al (2007) Nonlinear system identification with composite relevance vector machines. IEEE Signal Process Lett 14(4):279–282
48. Kingsbury N, Tay DBH, Palaniswami M (2005) Multi-scale kernel methods for classification. In: Proceedings of the IEEE workshop on machine learning for signal processing. Washington, DC, USA, pp 43−48