Proceedings of 2016 IEEE
International Conference on Mechatronics and Automation
August 7 - 10, Harbin, China

An Adaptive Compressive Tracking Algorithm for Amphibious Spherical Robots

Shaowu Pan[1, 2], Shuxiang Guo[1, 2, 3], Liwei Shi[1, 2, *], Ping Guo[1, 2], Yanlin He[1, 2] and Kun Tang[1, 2]

[1] *Key Laboratory of Convergence Medical Engineering System and Healthcare Technology, the Ministry of Industry and Information Technology, School of Life Science, Beijing Institute of Technology, No.5, Zhongguancun South Street, Haidian District, 100081 Beijing, China.*

[2] *Key Laboratory of Biomimetic Robots and Systems, Ministry of Education, Beijing Institute of Technology, No.5, Zhongguancun South Street, Haidian District, 100081 Beijing, China.*

[3] *Faculty of Engineering, Kagawa University, 2217-20 Hayashi-cho, Takamatsu, Japan.*

panshaowu@bit.edu.cn, guoshuxiang@bit.edu.cn, shiliwei@bit.edu.cn

[*]Corresponding author

***Abstract* – As a critical important function for autonomous mobile robots, visual tracking is a challenge work in the field of computer vision, for the reason that factors like illumination variance, partial occlusions and target appearance changes shall be carefully considered. Focus on applications of our amphibious spherical robots, an adaptive visual tracking algorithm was proposed on the basis of compressive tracking. A feature selection method was designed to choose random Haar-like feature templates in various scales by calculating Fisher's criterion functions of features. On this basis, a random feature pool, which tried to preserve discriminative features at different frames, were constructed and then maintained on-line to provide candidate appearance model of the target. Moreover, an adaptive update mechanism was adopted for selectively updating feature templates and classifier parameters of the improved compressive tracking algorithm, which alleviated the drift problem. Experimental results with various image sequences demonstrated the effectiveness and robustness of the proposed tracking algorithm, which can meet practical application requirements of the amphibious spherical robots.**

Index Terms –Compressive Tracking. Adaptive Update. Haar-like Feature. Random Feature Pool. Amphibious Spherical Robot.

I. INTRODUCTION

As a critical important function of autonomous mobile robots, visual tracking has become a hot research field of computer vision. Benefiting from the development of image detection and pattern recognition, great progresses in visual tracking have been made in recent years. And some state-of-the-art tracking algorithms, including TLD (Tracking-Learning-Detection) [1], MIL (Multiple Instance Learning) [2], STUCK (Structured output tracking with kernels) [3] and L1APG (L1 tracker using Accelerated Proximal Gradient) [4], have been proposed. Generally, most existing tracking algorithms can be loosely categorized as generative and discriminative algorithms according to the appearance modeling method to be used [5]. Generative algorithms represent the target with appearance features and then try to search the best matching region inside the image. Discriminative algorithms regard the tracking process as a binary classification problem and try to separate the target from background with pattern recognition methods like SVM (Support Vector Machine) and ANN (Artificial Neural Network). From our point of view, discriminative algorithms have a greater application potential, for the reason that a large number of samples are used for training the binary classifier to be used for tracking, which may lead to better robustness and higher precision.

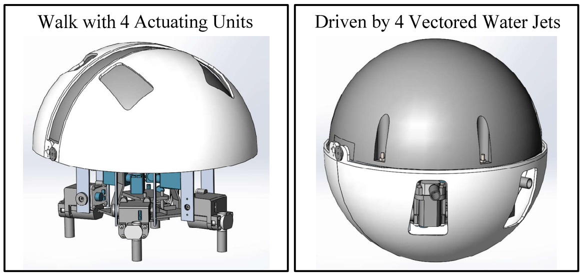
Compared with theoretical studies conducted on personal computers or workstations, designing practical visual tracking algorithms for small-scale mobile robots is a more challenge work. On the one hand, because the camera platform may move over time, factors like illumination variance, partial occlusions and target appearance changes shall be taken into account. On the other hand, the image processing ability of small-scale mobile robots is usually much weaker than workstations, thus real-time performance of the algorithm shall be specially considered.

Focus on the vision application issue of our amphibious spherical robots, an improved visual tracking algorithm was proposed on the basis of compressive tracking (CT) algorithm. A feature selection method was designed to choose random Haar-like feature templates in various scales by calculating Fisher's criterion functions of features, which enhanced robustness of the appearance model. On this basis, a random feature pool, which tried to preserve discriminative features at different frames, were constructed and then maintained on-line to provide candidate appearance model of the target. Moreover, an adaptive update mechanism was adopted for selectively updating feature templates and classifiers parameters of the improved compressive tracking algorithm, which alleviated the drift problem. Finally, evaluation experiments with various image sequences were conducted to verify the effectiveness and robustness of the proposed tracking algorithm, which demonstrated that it can meet application requirements of amphibious spherical robots.

The rest of this paper is organized as follows. An overview on our amphibious spherical robots and CT algorithm will be introduced in Section II. Details of the random Haar-like feature selection method will be elaborated in Section III. The adaptive update mechanism of the random feature templates and the improved algorithm will be described in Section IV. Experimental results will be provided in Section V. And Section VI will be conclusions and follow-up relevant research work.

II. Related Work and Application Requirement

A. Amphibious Spherical Robot



(a) Land mode (b) Underwater Mode

Fig. 1 Diagram of the improved amphibious spherical robot

As Introduced in reference [6], an amphibious spherical robot for delicate tasks in littoral regions was proposed by our team in 2012. As shown in Fig. 1, the robot consisted of a waterproof hemispheric upper hull, in which electronic devices and scientific instruments were installed, and two openable quarter-sphere lower shells [7]-[9]. In the land mode, the robot walked with four legs. And in the underwater mode, it swam with water jets. Different from most existing mobile robots or autonomous underwater vehicles, the robot was able to work in complex environments like coral reefs and pipelines.

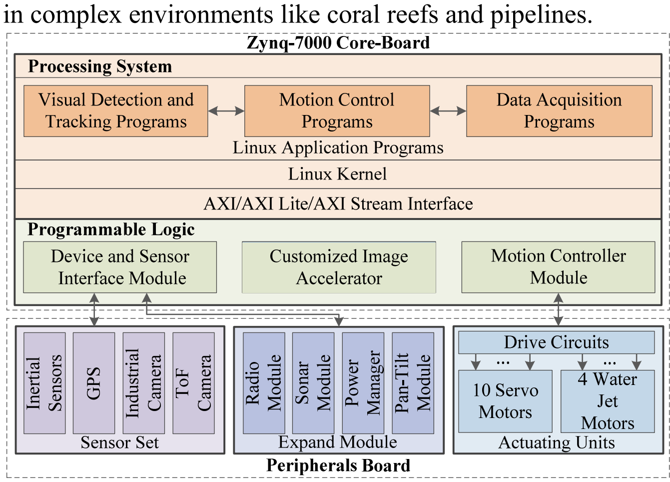
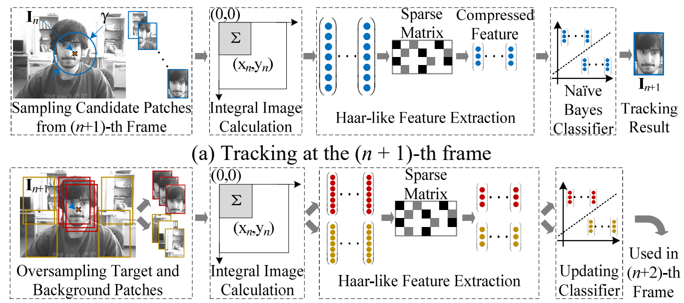


Fig. 2 Diagram of the electronic system of the amphibious spherical robot

The robot has limited load space which was less than 250 mm in diameter. So a high-speed computer or workstation, which consumers considerable power and generates a great deal of heat, is not suitable for this small-scale mobile robot. Consequently, as shown in Fig. 2, a core-board carrying a low-power Xilinx Zynq-7000 SoC (XC7Z020) was adopted to fabricate the electronic system of the improved version of the spherical robot in 2014 [10]. And functional devices and sensors like gyroscopes and cameras were integrated on a peripheral board controlled by the core-board. On this basis, robotic visual algorithms including Gaussian background model-based detection and CT were implemented on the embedded system in 2015. And a SoC-based heterogeneous computing architecture was designed to speed up image processing.

Considering the potential application scenarios and limited processing abilities, designing the tracking system of the amphibious spherical robot was a challenging task. First, because the robot may work under diverse land and underwater conditions, robustness to disturbing factors like partial occlusions was essential to the adopted tracking algorithm [11]-[13]. Second, the adopted tracking algorithm shall be carefully designed and optimized to fit the embedded processing system of the robot [14]-[15]. Third, precision and effectiveness of the adopted tracking algorithm should be acceptable to meet requirements of robotic applications [16]-[17].

B. Compressive Tracking Algorithm



(a) Tracking at the $(n + 1)$-th frame

(b) Updating classifier after Tracking at the $(n + 1)$-th frame

Fig. 3 Diagram of the CT algorithm.

In 2012, Zhang *et al.* proposed the CT algorithm, which provided a feasible framework for developing tracking algorithm of small-scale mobile robots [18]. Figure 3 shows major principles of the CT algorithm. As a concise discriminative algorithm, CT consists of two stages: tracking and updating. In the tracking stage, candidate image patches of the target of the $(n + 1)$-th frame are sampled around $\mathbf{I}_n$, which is the tracking result at the n-th frame. Then, low-dimensional features are extracted from the high-dimensional integral vectors of these samples using a static measurement matrix, which is in accord with the theory of compressive sensing. The process of compression or dimension reduction can be denoted as $\mathbf{v} = \mathbf{Mu}$, where $\mathbf{u} \in \mathbf{R}^n$ indicates the integral vectors, $\mathbf{v} \in \mathbf{R}^l$ indicates the feature vectors with dimensions $l \ll n$. $\mathbf{M}$ is a sparse random matrix, the entries of which were defined as:

$$m_{i,j} = \sqrt{s} \times \begin{cases} +1, \text{with probability } \frac{1}{2s} \\ 0, \text{with probability } 1-\frac{1}{s} \\ -1, \text{with probability } \frac{1}{2s} \end{cases} \tag{1}$$

where s=2 or 3. Then, the low-dimensional feature vectors are input into an online learning Naïve Bayes classifier. The sample with maximal classifier response is set to the target for determining $\mathbf{I}_{n+1}$. In the updating stage, training samples of the target and the background are sampled according to the tracking result at the $(n + 1)$-th frame $(\mathbf{I}_{n+1})$, and the compressed feature vectors of the training samples are used to update the parameters of the Naïve Bayes classifier, which will be used in the tracking stage of the $(n + 2)$-th frame.

The CT algorithm makes a balance between robustness and real-time performance, thus it has a bright future in low-power real-time visual applications. Xu *et al.* [19] designed an improved CT-based surveillance system for object tracking. Wang *et al.* [20] designed a CT-based person detection and tracking system for a mobile robot by fusing data from radio frequency identification (RFID). And a CT-based prototype tracking system for our amphibious spherical robot was design using Microsoft Kinect in 2015 [21]. However, the drift and anti-occlusion problems limited its practical applications in robotics. And some improved CT algorithms have been proposed trying to solve these problems using the Markov Chain Monte Carlo (MCMC) sampling mechanism [22], particle filters [23], kernel functions [24], etc.

III. Haar-like Feature Selection Method

A. *Analysis on Feature Model of Compressive Tracking*

From our point of view, the major contribution of the CT algorithm is the random Haar-like feature model inspired by the theory of compressive sensing. Haar-like features were originally used for face detection by calculating intensity differences between adjacent rectangle regions [25]. Because all candidate Haar-like feature templates in an image patch constitute an over-complete feature set for visual representation, selection methods like LDA (Linear Discriminant Analysis) [26] and Adaboost [27] were adopted to get feature templates with stronger separability.

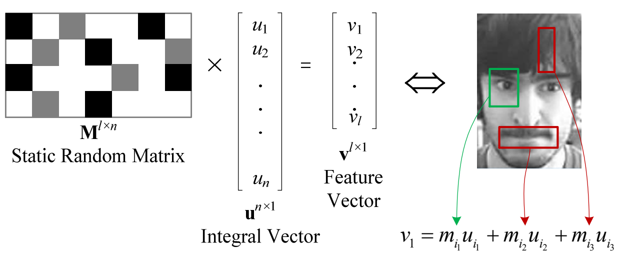
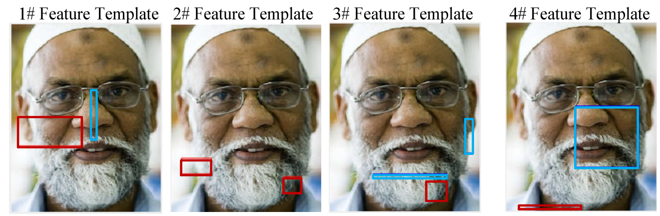


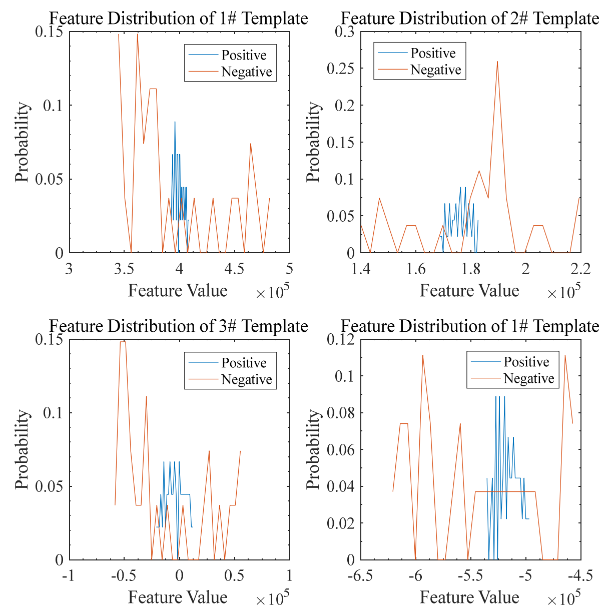
Fig. 4 Random Haar-like feature selection method of CT

As shown in Fig. 4, different from conventional solutions, a random selection method was adopted in the CT algorithm to randomly choose Haar-like feature templates using the static sparse matrix depicted in equation (1). And the random selection methods lead to a light computational consumption, which was critical important for real-time visual tracking. Ideally, feature values of negative image patches are while noises, while feature values of positive image patches obey a single Gaussian model. However, the separability of this random appearance model may be not so good and may significantly degrade with time for the reason that it has no practical physical meaning.

Figure 5 shows four samples of random feature templates selected by a CT tracker. Blocks marked in red represented random feature templates with positive coefficients, and blocks marked in blue represented random feature templates with negative coefficients. The 2# feature template covers flat regions of the target, thus it measures environmental noise rather than meaningful appearance information like edges, corners and textures. Blocks of the 3# feature template and 4# feature template are either too large or too small, so they are easily polluted by noises. Only blocks of the 1# feature template has a relatively suitable size and location, which results in a much more discriminative feature value than its counterparts.



(a) Haar-like feature templates selected by the CT algorithm



(b) Value distribution of Haar-like features selected by the CT algorithm

Fig. 5 Random Haar-like features selected by the CT algorithm

B. *Haar-like Feature Selection Method*

Although the random Haar-like feature model adopted in the CT algorithm is effective in some measure, its robustness can be enhanced by adding some constraints and update mechanisms. First of all, the separability of the appearance model and the precision of the tracker can be improved by selecting more discriminative feature templates. Moreover, because the CT algorithm adopted a set of static random feature templates, the cluster background may gradually pollute the target model. Thus a feature pool should be constructed and maintained to update the feature templates to be used online.

Considering the physical meaning of Haar-like feature, a coarse-to-fine feature selection method was designed to choose random Haar-like features. And a feature pool consisted l sub-pools, each of which contained N candidate feature templates would be set up. The feature selection method consisted of two stages.

In the coarse stage, three constraints were added into the random generation process of Haar-like feature templates for a higher probability of capturing meaningful features like edge and corner. These constraints were as follows.

(1) The size of each feature template was constrained by

$$\begin{cases} \max(\text{width}_{\min}, \alpha_{\min}\cdot \text{width}) \le w \le \alpha_{\max}\cdot \text{width} \\ \max(\text{heigth}_{\min}, \alpha_{\min}\cdot \text{heigth}) \le h \le \alpha_{\max}\cdot \text{heigth} \end{cases} \quad (2)$$

where width×height is the size of the target image patch, $w \times h$ is the size of the generated feature template, and α_{min}, α_{max}, $width_{min}$ and $height_{min}$ are coefficients set previously. As mentioned in Section II Part A, a feature template only captures information of raw pixels when its size is too small, and it has weaker spatial discriminative ability when its size is too large. Thus a medium size was adopted in this study.

(2) The shape of each feature template was constrained by

$$s_{min} \le \frac{w}{h} \le s_{max} \tag{3}$$

where s_{min} and s_{max} are coefficients set previously. As we all know, the shape of targets and textures have a relatively low possibility to be very slice and long.

(3) The size of the j-th feature template in the i-th sub-pools was constrained by

$$\begin{cases} w_{low,i} \le w_{i,j} \le w_{up,i} \\ h_{low,i} \le h_{i,j} \le h_{up,i} \end{cases} ,i \in [1,l], j \in [1,N] \tag{4}$$

where $w_{up,i}$ ($h_{up,i}$) and $w_{low,i}$ ($h_{low,i}$) are the upper and lower bound values of the width (height) of the generated feature template. And the value of $w_{up,i}$, $w_{low,i}$, $h_{up,i}$ and $h_{low,i}$ were calculated by using some common-used scale parameters in image processing [1]. This constrain was set to ensure that feature templates in the feature pool may capture appearance features in various scale. From our point of view, a diversified feature template set, which combines global and local features, will be more robust and effective.

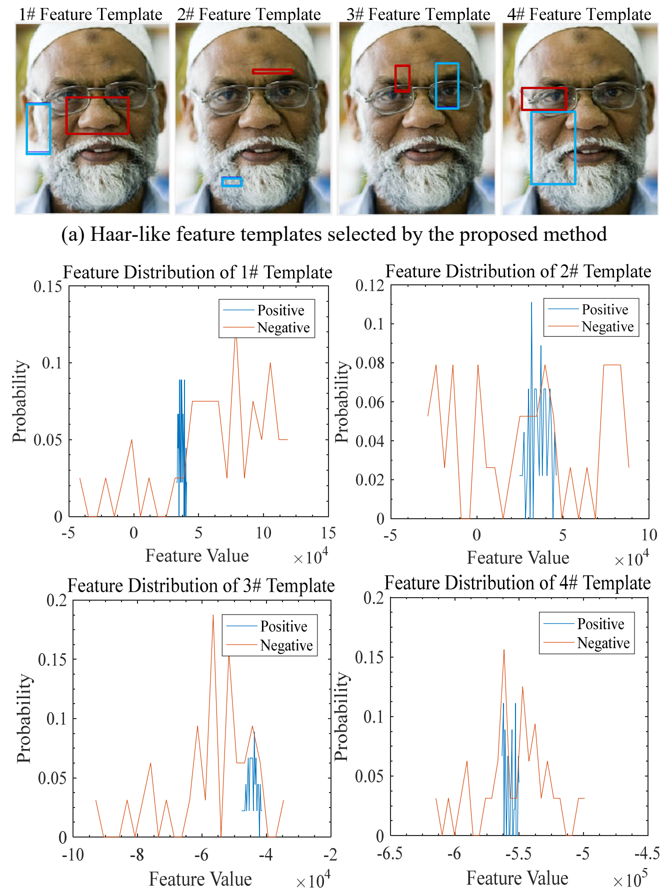
In the fine stage, $l \times N$ discriminative feature templates will be finally selected to construct the feature pool to be used. The Fisher's criterion function was adopted to measure the discriminative ability of a feature template.

$$J(f_{i,j}) = \left|\mu_+(f_{i,j}) - \mu_-(f_{i,j})\right|^2 - \lambda_1\sigma_+^2(f_{i,j}) - \lambda_2\sigma_-^2(f_{i,j}) \tag{5}$$

where $f_{i,j}$ is the j-th feature template of i-th sub-pools. $\mu_+(f_{i,j})$, $\mu_-(f_{i,j})$, $\sigma_+(f_{i,j})$, $\sigma_-(f_{i,j})$ are respectively the means and standard variances of feature values of positive and negative samples using the feature template $f_{i,j}$. And λ_1 and λ_2 are coefficients set previously. If the value of $J(f_{i,j})$ was relatively low, the feature template $f_{i,j}$ would be discarded and regenerated. The fine stage tried to maximize the distance between classes and to minimize the interclass variance. And it was hopeful to get a more clear classification result using the selected feature templates.

The computational load of the coarse stage was light because most operations could be completed by setting parameters of the random number generator. And the fine stage could be speed up by using the integral image. Thus the proposed method would not affect seriously the real-time performance.

As shown in Figure 6, by introducing the proposed coarse-to-fine feature selection method, the separability of the random feature templates were obviously improved. More or less, the selected templates have already been given some image meaning. And the shapes of the selected feature templates were more reasonable. As a result, the feature value distribution of positive samples was more concentrated than its counterpart in Fig. 5, which was beneficial for classification during tracking.



(a) Haar-like feature templates selected by the proposed method

(b) Value distribution of Haar-like features selected by the proposed method

Fig. 6 Random Haar-like features selected by the proposed method

IV. Adaptive Compressive Tracking Algorithm

A. Adaptive Compressive Tracking Algorithm

Except for the static appearance model, another factor causing the drift problem of CT is the static learning mechanism. Assuming all single feature values in $\mathbf{v}=\{v_1, v_2, \ldots, v_l\}$ are independently distributed, the tracker classified candidate image using l Naïve Bayes classifiers as

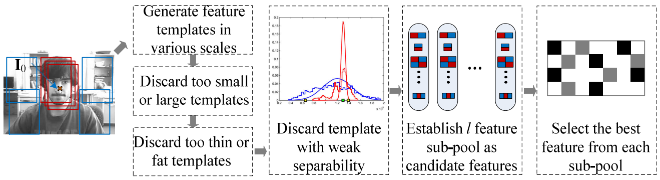
$$H(\mathbf{v}) = \log\left(\frac{\prod_{i=1}^{l} p(v_i|y=1)p(y=1)}{\prod_{i=1}^{l} p(v_i|y=0)p(y=0)}\right) = \sum_{i=1}^{l}\log\left(\frac{p(v_i|y=1)}{p(v_i|y=0)}\right) \tag{6}$$

Then parameters of these classifiers are incremental updated

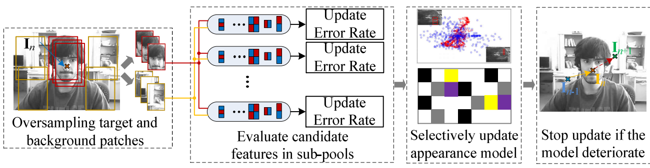
$$\begin{cases} \mu_y^{(i)} = \lambda\mu_y^{(i)} + (1-\lambda)\mu_y \\ \sigma_y^{(i)} = \sqrt{\lambda(\sigma_y^{(i)})^2 + (1-\lambda)(\sigma_y)^2 + \lambda(1-\lambda)(\sigma_y^{(i)} - \sigma_y)^2} \end{cases} \tag{7}$$

where λ is a static learning parameter. Because the CT algorithm updated the appearance model at a constant rate, drift or tracking failure would occur in some adverse scenarios. On the one hand, information of cluster background will gradually pollute the target model because the tracker accepted updating information without any identification. On the other hand, if occlusion occurred, the tracker will learn appearance features of the occluding object rather than the target, which may lead to tracking failure.

Commonly, semi-supervise learning and adaptive update mechanism are feasible tools to solve the drift problem of online learning tracking algorithms. And an adaptive update mechanism was designed for the proposed algorithm in this paper. Figure 6 shows the major principles of the proposed adaptive tracking algorithm. The proposed algorithm can be divided into three processes:



(a) Establishing feature pool and appearance model



(b) Feature pool adaptive updating

Fig. 7 Diagram of the proposed adaptive compressive tracking algorithm

(1) As shown in Fig. 7 (a), a feature pool was constructed at the first frame after randomly selecting $l\times$N discriminative features. The feature pool consisted of l feature sub-pools, each of which contained N candidate feature templates meeting constraints elaborated in Section III Part B. After that, candidate feature templates were sorted according to their separability using the training samples and the Fisher's criterion function. And the best one in each sub-pools were adopted to construct the random measurement matrix or the appearance model to be used for tracking.

(2) At the n-th frame, candidate image patches with maximum classifier response would be chosen as the target, which was as same as Fig. 3.

(3) After locating the target at the n-th frame, positive and negative training samples were sampled around $\mathbf{I}_n$ to evaluated candidate feature templates in the feature pool. If a feature template being used deteriorated, it would be replaced with the best one in the corresponding sub-pool. If most feature template being used deteriorated and the maximum classifier response is very low, occlusions or target appearance changes might have occurred. The tracker would reserve previous classifier parameters and try to predict the position of the target using a constant acceleration motion model. And the motion model was built on the basis of Kalman Filtering, which can be described as

$$\begin{cases} \mathbf{X}_{n+1} = \mathbf{\Phi}\mathbf{X}_n + \beta\mathbf{W}_n \\ \mathbf{Y}_{n+1} = \mathbf{H}\mathbf{X}_{n+1} + \alpha\mathbf{V}_n \\ \mathbf{\Phi} = \begin{pmatrix} 1 & 0 & \Delta t & 0 & \Delta t^2/2 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \Delta t^2/2 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\ \mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \\ \mathbf{X}_n = \left(x_n, y_n, v_{x,n}, v_{y,n}, a_{x,n}, a_{y,n}\right)^{\mathrm{T}} \\ \mathbf{Y}_n = \left(\tilde{x}_n, \tilde{y}_n\right)^{\mathrm{T}} \end{cases} \tag{8}$$

B. *Feature Update Mechanism*

Details of the adaptive feature update mechanism are as shown in Algorithm 1. The features to be updated included the feature templates being used and candidate feature templates in the feature pool. As to the feature templates being used ($\tilde{f}_i$), the relative error at recent frames ($\tilde{\delta}_i$), which was calculated with response values of Naïve Bayes classifier, cumulated over time. And $\tilde{\delta}_i$ was used to evaluate the effectiveness of $\tilde{f}_i$. If $\tilde{f}_i$ kept output fuzzy or error classification results, it would be replaced with the candidate feature template with best discriminative ability in the corresponding sub-pool. As to the candidate feature templates in the feature pool ($f_{i,j}$), the one with maximum accumulated errors would be regenerated with the probability $e^{-\delta_{\mathrm{worst}}/\Delta}$. That provided an additional opportunity for some features performed not so well up to now but may be discriminative in some specific circumstances. And this design tried to avoid trapping in local optimum and may be robust to appearance changes of the target.

Algorithm 1 Adaptive feature update mechanism

Input: positive and negative samples for training <**v**, y>, $y\in\{-1,+1\}$
Input/Output: feature templates to be used for tracking $\tilde{f}_i$, $i\in[1,l]$
random feature pools $f_{i,j}$, $i\in[1,l]$, $j\in[1,\mathrm{N}]$

Initialize coefficients including α_{min}, α_{max}, $width_{min}$ $height_{min}$, s_{min}, s_{max}, etc.
Update Naïve Bayes classifier parameters of all candidate features
for i=1, 2, …, l **do**//for all feature sub-pools and weak classifiers to be used
  for j=1, 2, …, N **do**//for all candidate features in a feature sub-pools
    //check the feature template using response of Naïve Bayes classifiers
    if $y_{\mathrm{low}} \le h_{i,j}(\mathbf{v}) \le y_{\mathrm{up}}$ **then**
      $\delta_{i,j}^{\mathrm{err}} = \delta_{i,j}^{\mathrm{err}} + \delta_0$
    else
      $\delta_{i,j}^{\mathrm{corr}} = \delta_{i,j}^{\mathrm{corr}} + \delta_0$
    end if
    $\delta_{i,j} = \delta_{i,j}^{\mathrm{err}}/(\delta_{i,j}^{\mathrm{corr}} + \delta_{i,j}^{\mathrm{err}})$
  end for
  if $\tilde{h}_i(\mathbf{v}) < y_{\mathrm{thresh}}$ and $\tilde{\delta}_i > \delta_{\mathrm{thresh1}}$ **then**
    $\tilde{f}_i = \arg\max_{J(f_{i,j})}(f_{i,j})$
  end if
  $f_{\mathrm{worst}} = \arg\max_{\delta_{i,j}}(f_{i,j})$
  if $\delta_{\mathrm{worst}} > \delta_{\mathrm{thresh2}}$ **then**
    regenerate f_{worst} of i-th feature sub-pool with the probability $e^{-\delta_{\mathrm{worst}}/\Delta}$
  end if
end for

The updating rate of candidate feature templates in the feature pool was constrained restrict (only 0.05 in this study) for a relatively stable appearance model. As far as we know, the principle of the random Haar-like feature model adopted in CT was similar with random forest in some measure. Thus an unstable appearance model may decrease the stability of the tracker.

In addition, the computational load of the update mechanism was increased for the reason that Naïve Bayes classifiers to be maintained on-line increased N times. However, the candidate feature templates were data independent. So the calculation process can be accelerated using parallel computing technologies. Meanwhile, l could be set to a smaller value than the original CT algorithm because the selected feature templates were more robust and discriminative. That would also lighten some computational load.

V. Experimental Results

The goal of this paper is to design an efficient and effective improved CT algorithm for our amphibious spherical robots. To verify the validation of the proposed algorithm, experiments were conducted with various image sequences on two parts:

(1) To evaluate the effectiveness of the feature selection method elaborated in Section III and the adaptive tracking algorithm elaborated in Section IV, six standard benchmark image sequences were used to conduct the off-line evaluation experiments on MATLAB R2013a. And the tracking results were compared with that of CT and MIL which is also a Haar-like feature-based tracking algorithm.

(2) To evaluate the work performance of the proposed algorithm on our amphibious spherical robots, two image sequences captured from the view of our robots were adopted to conduct the robotic evaluation experiments.

A. *Experiments with Benchmark Sequences*

ACT CT MIL

Fig. 8 Screenshots of the benchmark experimental results

TABLE I

Success Rate and Centre Location Error of the Proposed Algorithm

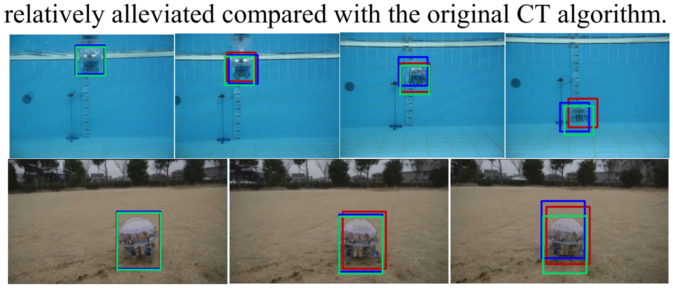
Sequences	Proposed		CT		MILTrack		Number
	SR	CLE	SR	CLE	SR	CLE	of Frames
David	98	13	94	14	71	19	462
Suv	52	54	25	54	35	54	945
Couple	90	13	69	14	67	14	140
Carscale	100	89	98	90	65	91	251
Walking	33	17	31	17	28	17	369
Coke	54	15	14	15	8	15	291

The image sequences "*David*", "*Suv*", "*Couple*", "*CarScale*", "*Walking*" and "*Coke*", taken from [28], were used to test the proposed algorithm. The success rate (SR), which is the coincidence rate between the tracked bounding box and the ground truth bounding box, and the centre location error (CLE), which is the Euclidean distance between the central points of the tracked bounding box and the ground truth bounding box, were adopted to evaluate the performance of the proposed algorithm.

Figure 8 shows some screenshots of tracking results of the proposed algorithm. And TABLE I shows the quantitative evaluation results of experiments with benchmark sequences.The results indicated that the robustness to drift, occlusion and background clutter of the proposed algorithm was reinforced by adopting feature selection method and feature update mechanism. As shown in the first image sequence, results of CT and MIL drifted when the face in the image moved erratically at #150, whereas the proposed algorithm was able to track the target precisely. As shown in third image sequence, the CT tracker and MIL tracker both missed the couple in the image when a car with similar Haar-like features appeared, but the improved CT tracker avoided this failure by predicting the motion trend of the target. As shown in the fourth and sixth image sequence, the CT tracker and MIL tracker gradually lost the target when it was partial occluded, whereas the proposed algorithm successfully detected the car when it reappeared. Although the adaptive CT algorithm had a higher success rate in the benchmark test, it made little contribution to the performance improvement when illumination variation or long-term occlusion occurred, as shown in the second and sixth image sequence. And the problem of drift still remained.

B. *Robotic Experiments*

In the robotic experiments, 2 image sequences captured from the view our robot were adopted to verify the performance of the proposed algorithm [29-32]. As shown in Fig. 9, drift appeared when the shape of the robot deformed. Limited by robustness of adopting appearance models, it seemed that the CT and MIL trackers were misled and turned to focus on legs of the robot. But the proposed algorithm can still catch up the correct position of the target, even the robot's pose and the background changed overtime. And the drift problem was relatively alleviated compared with the original CT algorithm.



ACT CT MIL

Fig. 9 Screenshots of the robotic experimental results

VI. Conclusions and Future Works

Aiming at vision applications of our amphibious spherical robots, an adaptive CT algorithm was proposed in this paper. A feature selection method was designed to select discriminative random Haar-like features and maintain adaptive feature pools. And an adaptive update mechanism was adopted for selectively updating feature templates and classifier parameters of the improved compressive tracking algorithm, which alleviated the drift problem. Experimental results demonstrated that the feature selection method and adaptive update mechanism used in the proposed algorithm alleviate the drift problem and acquired stronger robustness, which may meet requirements of our spherical robots.

The study in this paper mainly focused on the design of the algorithm. But the real-time performance of the proposed algorithm and related practical applications still remained a problem. Our future work will focus on the parallel implementation of the proposed algorithm on the robotic embedded platform.

Acknowledgment

This work was supported by National Natural Science Foundation of China (61503028, 61375094), Excellent Young Scholars Research Fund of Beijing Institute of Technology (2014YG1611), and the Basic Research Fund of the Beijing Institute of Technology (20151642002). This research project was also partly supported by National High Tech. Research and Development Program of China (No.2015AA043202). Pengyi Zhang also contributed to the idea of the algorithm proposed in this paper.

References

[1] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[2] B. Babenko, Y. Ming-Hsuan, and S. Belongie, "Visual tracking with online multiple instance learning," *Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pp. 983–990, Miami, 2009.

[3] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," *Proceedings of 2011 IEEE International Conference on Computer Vision (ICCV 2011)*, pp. 263–270, Barcelona, 2011.

[4] B. Chenglong, W. Yi, L. Haibin, and J. Hui, "Real time robust L1 tracker using accelerated proximal gradient approach," *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, pp. 1830–1837, Providence, 2012.

[5] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, pp. 1–48, 2013.

[6] S. Guo, S. Mao, L. Shi and M. Li, "Development of an amphibious mother spherical robot used as the carrier for underwater microrobots," *Proceedings of 2012 ICME International Conference on Complex Medical Engineering (CME)*, pp. 758–762, Kobe, 2012.

[7] C. Yue, S. Guo and L. Shi, "Hydrodynamic Analysis of the Spherical Underwater Robot SUR-II," *International Journal of Advanced Robotic Systems*, vol. 10, no. 3, pp. 1–12, 2013.

[8] M. Li, S. Guo, H. Hirata and H. Ishihara, "Design and Performance Evaluation of an Amphibious Spherical Robot," *Robotics and Autonomous Systems*, vol. 64, pp. 21–34, 2014.

[9] X. Li, J. Guo and S. Guo, "OFDM-based Micro-Signal Communication Method for the Spherical Amphibious Underwater Vehicle," *Proceedings of 2015 IEEE International Conference on Mechatronics and Automation*, pp. 2094–2099, Beijing, China, 2015.

[10] S. Pan, L. Shi, S. Guo, P. Guo, Y. He and R. Xiao, "A low-power SoC-based moving target detection system for amphibious spherical robots," *Proceedings of 2015 International Conference on Mechatronics and Automation (ICMA)*, pp. 1116–1121, Beijing, 2015.

[11] C. Yue, S. Guo and L. Shi, "Design and Performance Evaluation of a Biomimetic Microrobot for the Father-son Underwater Intervention Robotic System," *Microsystem Technologies*, vol. 22, no. 4, pp. 1–10, 2015.

[12] C. Yue, S. Guo, M. Li, Y. Li, H. Hirata and H. Ishihara, "Mechantronic System and Experiments of a Spherical Underwater Robot: SUR-II," *Journal of Intelligent and Robotic Systems*, vol. 80, no. 2, pp. 325–340, 2015.

[13] Y. Li, S. Guo and C. Yue, "Preliminary Concept of a Novel Spherical Underwater Robot," *International Journal of Mechatronics and Automation*, vol. 5, no. 1, pp. 11–21, 2015.

[14] C. Yue, S. Guo, M. Li and Y. Li, "Characteristics Evaluation of a Biomimetic Microrobot for a Father-son Underwater Intervention Robotic System," *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 171–176, Hamburg, Germany, 2015.

[15] L. Bi, J. Guo and S. Guo, "Virtual Prototyping Technology-based Dynamics Analysis for an Amphibious Spherical Robot," *Proceedings of the 2015 IEEE International Conference on Information and Automation*, pp. 2563–2568, Lijing, China, 2015.

[16] L. Bi, J. Guo, S. Guo and Z. Zhong, "Kinematic Analysis on Land of an Amphibious Spherical Robot System," *Proceedings of 2015 IEEE International Conference on Mechatronics and Automation*, pp. 2082–2087, Beijing, China, 2015.

[17] Z. Zhong, J. Guo, S. Guo and L. Bi, "Characteristic Analysis in Water for an Amphibious Spherical Robot," *Proceedings of 2015 IEEE International Conference on Mechatronics and Automation*, pp. 2088–2093, Beijing, China, 2015.

[18] K. Zhang, L. Zhang, and M.-H. Yang, "Real-Time Compressive Tracking," *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, pp. 866–879, Florence, 2012.

[19] X. Huazhong, and Y. Fei, "Improved Compressive Tracking in Surveillance Scenes," *Proceedings of 2013 Seventh International Conference on Image and Graphics (ICIG)*, pp. 869–873, Qingdao, 2013.

[20] W. Lijia, J. Songmin, L. Xiuzhi, and W. Shuang, "RFID and vision based person tracking of a mobile robot using improved compressive tracking," *Proceedings of 2013 IEEE International Conference on Information and Automation (ICIA)*, pp. 1171–1176, Yinchuan, 2013.

[21] S. Pan, L. Shi, S. Guo, "A Kinect-Based Real-Time Compressive Tracking Prototype System for Amphibious Spherical Robots," *Sensors*, vol. 15, no. 4, pp. 8232–8252, April 2015.

[22] W. Lan, D. Pingyang, L. Yanlong, L. Cuihua, and X. Yi, "Visual Tracking Based on Compressive Sensing MCMC Sampling," *Proceedings of 2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 4288–4293, Manchester, 2013.

[23] W. Jiayun, C. Daquan, and Y. Rui, "Real-time compressive tracking with motion estimation," *Proceedings of 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2374–2379, Shenzhen, 2013.

[24] Q. Yan, and L. Li, "Kernel sparse tracking with compressive sensing," *IET Computer Vision*, vol. 8, no. 4, pp. 305-315, 2014.

[25] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian Detection Using Wavelet Templates," *Proceedings of 1997 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1997),* pp.193-199, San Juan, *1997*.

[26] S. K. Pavani, D. Delgado, and A. F. Frangi, "Haar-like features with optimally weighted rectangles for rapid object detection," *Pattern Recognition*, vol. 43, no. 1, pp. 160–172, 2010.

[27] P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of 2001 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, pp. I-511–I-518, Kauai, 2001.

[28] Y. Wu, J. Lim, and M.-H. Yang, "Object Tracking Benchmark," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[29] Y. He, L. Shi, S. Guo, S. Pan, and Z. Wang, "Preliminary mechanical analysis of an improved amphibious spherical father robot," *Circulation*, vol. 99, no. 6, pp. 1–16, 2015.

[30] L. Shi, R. Xiao, S. Guo, P. Guo, S. Pan and Y. He. "An attitude estimation system for amphibious spherical robots," *Proceedings of 2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 2076–2081, Beijing, 2015.

[31] Y. He, L. Shi, S. Guo, P. Guo, R. Xiao. "Numerical simulation and hydrodynamic analysis of an amphibious spherical robot," *Proceedings of 2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 848–853, Beijing, 2015.

[32] L. Shi, S. Guo, S Pan, Y. He and P. Guo. "A multifunctional underwater microrobot for mother-son underwater robot system," *Proceedings of 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO),* pp. 1007–1012, Shenzhen, 2013.