# Reinforcement Learning-Based Feature Learning for Object Tracking

Fang Liu and Jianbo Su
Research Center of Intelligent Robotics & Dept. of Automation
Shanghai Jiaotong University, Shanghai, 200030, China
{fangliu, jbsu}@sjtu.edu.cn

## Abstract

*Feature learning in object tracking is important because the choice of the features significantly affects system's performance. In this paper, a novel online feature learning approach based on reinforcement learning is proposed. Reinforcement learning has been extensively used as a generative model of sequential decision-making that interacts with uncertain environment. We extend this technique to feature selection for object tracking, and further add human-computer interaction to reinforcement learning to reduce the learning complexity and speed the convergence rate. Experiments of the object tracking are provided to verify the effectiveness of the proposed approach.*

## 1. Introduction

Visual tracking is an important issue in computer vision. It has been extensively studied for more than there decades. With a stationary camera, moving object can be effectively tracked in real time using adaptive background subtractions [7]. Particle filtering algorithm, such as CONDENSATION, enables tracking in visual occlusion and clutter by reasoning over a state-space of multiple hypotheses [3]. Recently kernel-based tracking method, mean shift, has been a focus of much attention. It can reach superior tracking performance by a relatively low computational complexity. Extensions of the basic mean shift are presented by various researchers. Collins adopts Lindeberg's theory to the problem of selecting kernel scale for mean-shift blob tracking [1]. Nummiaro et al. present the integration of color distributions into particle filtering to efficiently handle non-rigid and fast moving objects under different appearance changes [6]. Normally, mean shift need complete region's contour that is usually initialized with a hand-drawn circle or ellipse in the first frame. It is sometimes inconvenient to do it. The question then is how to detect automatically the target region for posterior tracker. Xu and Fujimura apply SVM for pedestrian detection from infrared videos and combine Kalman filter and mean shift for tracking [10].

However, none of the above methods allows to select the desired features prior to the tracking procedure. Desired features selection is important because the selection result has significant influence on the performance of the feature-based tracking system. Most of the features are irrelevant or redundant for a given tracking task, which slow down the tracking speed and cause inaccuracy during acquiring and using them. The system must quickly discard the irrelevant and redundant features, so as to concentrate the resource on learning a mapping between the selected feature subset and a tracking task, which track the target efficiently. Kaneko and Hori present a new feature selection approach for reliable tracking using template matching is proposed, but the drawback is the high computational cost because of evaluation of each point [4].

Feature selection is defined as finding the minimally sized feature set that is necessary and sufficient for the task. There are two strategies for feature selection: feature filter and wrapper approach. However, these studies focus on the selection of a subset for a given task, but not visual context (i.e., the decision does not depend on the current visual environment). Machine learning allows the system to improve its ability in feature selection adaptively over time, based on the visual environment that it acts on. Then this feature learning procedure is incremental, interactive and task-oriented. Moreover, machine learning can solve the feature selection problem when there is no knowledge about the visual environment and task. As an incremental, task-oriented, and free-model learning algorithm, reinforcement learning ($Q$-learning) is very suitable for feature selection. In this paper, reinforcement learning is used as an efficient tool to learn feature selection strategy through interaction with visual environment. The system can learn to select the desired feature subset, considering the probability distribution of the visual environment. The drawback of standard reinforcement learning is that the system suffers from average update complexity depending on the size of the state-action space. To alleviate this problem, a novel learning structure integrated with reinforcement learning algorithm

IEEE COMPUTER SOCIETY

and human-computer interaction (HCI) has been proposed. Via HCI, the system can obtain the distance measurement between current feature state and desired feature state to reduce the learning complexity and speed the convergence rate.

The paper is organized as follows: Section 2 presents a feature learning theory and a HCI-based feature learning scheme. The feature learning-based object tracking is described in Section 3. Section 4 gives experimental results. And finally the main conclusion and the future directions of our research are summarized in Section 5.

## 2. Feature learning

### 2.1. Reinforcement learning-based feature learning

Primitive feature extraction and feature space construction are completed prior to feature learning. Various types of low-level or mid-level features are extracted, such as color, shape, motion, and texture feature, etc. For example, shape feature $F_s$ is obtained by calculating the Hough transform of an edge map for circles. The way to generate a feature space is by defining a feature vector as attribute compositions of several components. The feature attribute vector can be expressed as:

$$\mathbf{f} = (f_{color}, f_{shape}, f_{motion}, \cdots) \tag{1}$$

where $\mathbf{f}$ denotes the integration of all features, $f_{color}$, $f_{shape}$, and $f_{motion}$ are the feature variable of color, shape, and motion respectively.

Feature selection can be viewed as a search problem, with each state in the search space specifying a feature subset. Feature selection procedure could be modeled as a discrete control problem. In this paper, we adopt reinforcement learning algorithm to learn the feature selection based on the current state of image. We define *state* as the feature attribute vectors $\mathbf{f}$ in the feature space, and *action* $a$ as the selection of the feature subset. Via computing a similarity function, the system receives 0 for the *reward* $r$ if the action selects the desired feature subset, and -1 otherwise. The general principle of reinforcement learning can be seen in [8] and [9]. A model-free $Q$-learning algorithm has been developed to find optimal actions. Initially, all the values of $Q(\mathbf{f}, a)$ are initialized. Every time in one feature state, the system chooses and executes an action, and the $Q(\mathbf{f}, a)$ is updated as follows:

$$\begin{aligned} Q_{t+1}(\mathbf{f}_t, a_t) &= Q_t(\mathbf{f}_t, a_t) + \alpha[r_t \\ &+\gamma \max_{a_{t+1}} Q_t(\mathbf{f}_{t+1}, a_{t+1}) - Q_t(\mathbf{f}_t, a_t)] \end{aligned} \tag{2}$$

where $\alpha$ is the learning rate; $\gamma$ is the discounted rate. Watkins proved that the $Q$-values would converge to the optimal values given that each state-action pair is experienced infinitely and the environment is Markovian [9]. The optimal policy $\pi^*(\mathbf{f})$ can be obtained by the following definition:

$$\pi^*(\mathbf{f}) = arg \max_a Q(\mathbf{f}, a). \tag{3}$$

### 2.2. HCI-based feature learning

The drawback of standard reinforcement learning is that the system suffers from average update complexity depending on the size of the state-action space. An alternative solution for this is to adjust the way to obtain reward $r$ through human-computer interaction. We define the reward according to different evaluation criterions that can measure distance to goal, which has the following form:

$$r_{HCI} = k_1 r_1 + k_2 r_2 + \cdots + k_n r_n \tag{4}$$

where $r_i$ is the reward that is provided by operator based on the $i^{th}$ evaluation criterion, $k_i$ is the weight of $r_i$ , and $n$ is the number of evaluation criteria. The system can obtain the distance measurement between current feature subset and desired feature subset. It constructs an interactive learning architecture that operator can give a direct feedback to learning system according to the performance of system behavior.

There are three aspects in a typical feature selection method: 1) search direction (feature subset generation); 2) search strategies; and 3) evaluation measures [5]. Our method for feature selection differs from the conventional ones in the following properties: 1) Feature selection can be interactive with uncertain visual environment. 2) Searching can be in either forward or backward direction. 3) Evaluation criterion is independent of the system's prior knowledge. 4) Via HCI, the learning process can be simplified effectively.

## 3. Feature learning-based object tracking

The tracking system is constructed based on the above feature learning. Let $p_m(u, \mathbf{f})$ be the Epanechikov kernel-based probability of the tracking model, which is obtained from the desired feature subset, and $p_c(u, z)$ be the Epanechikov kernel-based probability of the candidate target, centered at $z$. The distance between these distributions is defined as:

$$d(p_m(\mathbf{f}), p_c(z)) = \sqrt{1 - \sum_{u=1}^{n} p_m(u, \mathbf{f}) p_c(u, z)} \tag{5}$$

Mean shift iteration is used to find the resigns in the next frame which are similar to the tracking model. Let $\Re$ be the set of the pixels that belong to the circle of the candidate target. Let $z_i$ be the location of the candidate target, defined as:

$$z_i = \frac{\sum\limits_{x_i \in \Re} x_i (\sum\limits_{u=1}^{n} \sqrt{\frac{p_m(\mathbf{f})}{\hat{p}_c(z_0)}} \delta(I(x_i) - u)) g(\|z_0 - x_i\|^2)}{\sum\limits_{x_i \in \Re} (\sum\limits_{u=1}^{n} \sqrt{\frac{p_m(\mathbf{f})}{\hat{p}_c(z_0)}} \delta(I(x_i) - u)) g(\|z_0 - x_i\|^2)} \tag{6}$$

where $w_i = \sum\limits_{u=1}^{n} \sqrt{\frac{p_m(\mathbf{f})}{\hat{p}_c(z_0)}} \delta(I(x_i) - u))$, and $g(x) = -k^{'}(x) = constant$. We obtain:

$$z_i = \frac{\sum\limits_{x_i \in \Re} (x_i (\sum\limits_{u=1}^{n} \sqrt{\frac{p_m(\mathbf{f})}{\hat{p}_c(z_0)}} \delta(I(x_i) - u))}{\sum\limits_{x_i \in \Re} \sum\limits_{u=1}^{n} \sqrt{\frac{p_m(\mathbf{f})}{\hat{p}_c(z_0)}} \delta(I(x_i) - u)} = \frac{\sum\limits_{x_i \in \Re} x_i w_i}{\sum\limits_{x_i \in \Re} w_i} \tag{7}$$

It is proved that the mean shift procedure can efficiently find the minimum of the distance $d(p_m(\mathbf{f}), p_c(z))$ [2].

## 4. Experimental results

### 4.1. The performance of feature learning

The first experiment shows that the system could learn to automatically select the desired feature subset to track the blue circle object using reinforcement learning. In Figure 1, the frames 5, 10, and 15 are shown. Figure 2 shows the feature learning, object location and tracking. As can be seen , the system successfully learns to select the shape and motion features (desired feature subset) to exactly locate the target, and track it. In Figure 2 (c), the target is also detected by selecting blue and shape features, but this feature subset is not the desired one because the blue feature hampers the performance of the object location.

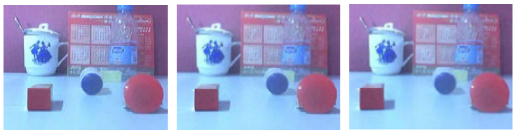
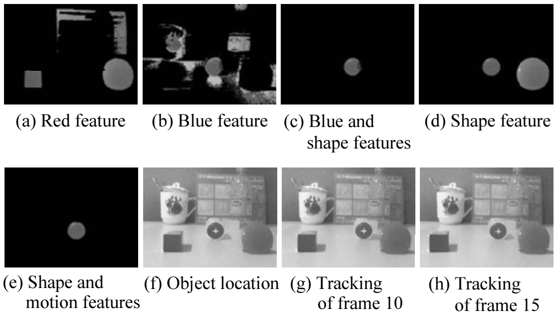


**Figure 1. The frames 5, 10, and 15 are shown.**

Figure 3 shows the learning steps per trial from the start feature subset to the desired one. The results were averaged over 200 trials. It is apparent that the number of feature selection actions decreases with the increase of the number of trials. The proposed feature learning approach is efficient to improve the performance of system.

### 4.2. Feature learning under changed visual environment

In this experiment, the ability of the system to learn the adaptive desired feature subset under a changed visual environment was tested. Figure 4 shows the frames 20, 25, and 30, where the visual environment has changed because of the disappearance of red circle object. Figure 5 shows the feature learning, object location and tracking. As can be seen, the system successfully learns to select the changed desired feature subset (shape feature) to exactly locate the target, and track it. In Figure 5 (c), the target is also detected by selecting shape and motion features, but this feature subset is not the best feature subset because of the changed visual environment. The system can learn to select changed desired feature subset by using reinforcement learning interacts with the visual environment.



(a) Red feature (b) Blue feature (c) Blue and shape features (d) Shape feature

(e) Shape and motion features (f) Object location (g) Tracking of frame 10 (h) Tracking of frame 15

**Figure 2. Feature learning, object location and tracking.**

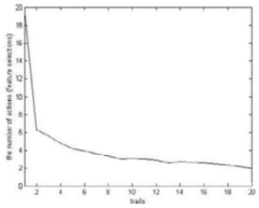


**Figure 3. The number of actions per trial.**

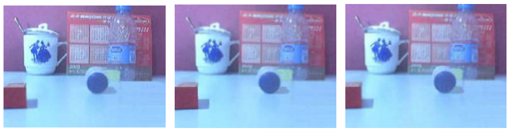
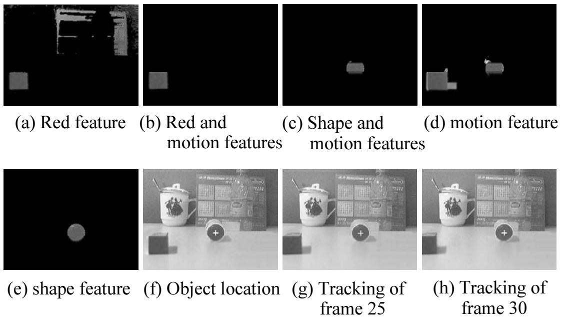


**Figure 4. The frames 20, 25, and 30 are shown, where the visual environment is changed because of the disappearance of red circle object.**

Figure 6 shows the performance of learning under changed environment. The results were averaged over 200 trials. It is apparent that the number of feature selection with changed environment is less because of the simplified visual environment.

IEEE COMPUTER SOCIETY



(a) Red feature (b) Red and motion features (c) Shape and motion features (d) motion feature

(e) shape feature (f) Object location (g) Tracking of frame 25 (h) Tracking of frame 30

**Figure 5. Feature learning, object location and tracking under changed environment.**

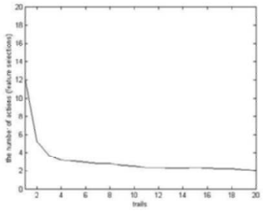


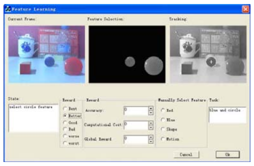
**Figure 6. The Number of actions per trial.**

## 4.3. The performance of the HCI-based feature learning

In this experiment, the performance of the HCI-based feature learning has been tested. The human-computer interface is illustrated in Figure 7. The operator should provide reward $r_{HCI}$ for reinforcement learning via HCI in order to simplify the feature learning process. Figure 8 compares the performance of reinforcement learning and HCI-based reinforcement learning on the number of actions per trial. The results were averaged over 200 trials. As can be seen, the number of feature selection with HCI-based reinforcement learning is less than standard reinforcement learning. The experiments result shows that the HCI-based reward, which can measure the distance with goal, helps the system efficiently receive the evaluation of its action to reduce the learning complexity and speed the convergence rate. When the feature space becomes larger, the further performance improvement expect to be obtained.

## 5. Conclusions

This paper proposes a reinforcement learning-based feature learning approach that continually evaluates and learns online to select the features used for special tracking task. The feature learning procedure is designed as a model of sequential decision-making that interacts with uncertain visual environment. Moreover a HCI-based reinforcement learning scheme is proposed to obtain direct feedback about feature selection behavior with the help of operator's knowledge to reduce the learning complexity and speed convergence rate. Future work lies in including more features in the learning framework so as to extend its application to more complex tasks.



State: select circle feature

Reward: ○ Best ● Better ○ Good ○ Bad ○ worse ○ worst

Reward: Accuracy: 0 Computational Cost: 0 Global Reward: 0

Manually Select Feature: ○ Red ○ Blue ○ Shape ○ Motion

Task: blue and circle

**Figure 7. Human-computer interface.**

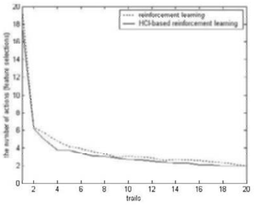


**Figure 8. The Number of actions per trial.**

## Acknowledgments

This research has been supported by NSFC, under grant 60275042.

## References

[1] R. T. Collins, "Mean-shift Blob Tracking through Scale Space," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2:234-240, 2003.

[2] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(5):564-577, 2003.

[3] M. Isard and A. Blake, "CONDENSATION-Conditional Density Propagation for Visual tracking," *Int'l J. Computer Vision*, 29(1):5-28, 1998.

[4] T. Kaneko and O. Hori, "Feature selection for reliable tracking using template matching," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1:796-802, 2003.

[5] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publishers, 1998.

[6] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, 21:99-110, 2003.

[7] C. and W.E.L.Grimson, "Adaptive Background Mixture Models for Real-time Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2:246-252, 1999.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[9] C. J. C. H. Watkins and P. Dayan, "Technical Note: Q-Learning," *Machine Learning*, 8:279-292, 1992.

[10] F. Xu and K. Fujimura, "Pedestrian detection and tracking with night vision," *Proc. IEEE Intelligent Vehicle Symposium*, 1:21-30, 2002.