

Research Article

Memory-Based Multiagent Coevolution Modeling for Robust Moving Object Tracking

Yanjiang Wang, Yujuan Qi, and Yongping Li

College of Information and Control Engineering, China University of Petroleum, No. 66, Changjiang West Road, Economic and Technological Development Zone, Qingdao 266580, China

Correspondence should be addressed to Yujuan Qi; qiyj@upc.edu.cn

Received 28 March 2013; Accepted 28 April 2013

Academic Editors: P. Agarwal, S. Balochian, V. Bhatnagar, J. Yan, and Y. Zhang

Copyright © 2013 Yanjiang Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The three-stage human brain memory model is incorporated into a multiagent coevolutionary process for finding the best match of the appearance of an object, and a memory-based multiagent coevolution algorithm for robust tracking the moving objects is presented in this paper. Each agent can remember, retrieve, or forget the appearance of the object through its own memory system by its own experience. A number of such memory-based agents are randomly distributed nearby the located object region and then mapped onto a 2D lattice-like environment for predicting the new location of the object by their coevolutionary behaviors, such as competition, recombination, and migration. Experimental results show that the proposed method can deal with large appearance changes and heavy occlusions when tracking a moving object. It can locate the correct object after the appearance changed or the occlusion recovered and outperforms the traditional particle filter-based tracking methods.

1. Introduction

The problem of object tracking is often posed as that of estimating the trajectory of objects in an image plane as objects move in a scene [1]. Although considerable efforts have been made in establishing a robust tracking framework in the research literature, the problem still remains challenging when appearance abrupt changes or occlusions occur. To address these challenges, in the literature tremendous attempts have been made in characterizing appearance models which are able to handle appearance changes. In this context, most of the extant methods tend to apply a total model updating mechanism for template updating in which the initial template model is updated gradually based on the estimated information, for example particle filters (PF). However, if an object is heavily occluded or its appearance changes abruptly, the total model updating based PF (TMU-PF) will gradually deviate from the target.

Recently, a lot of modifications have been made for improving the performance of particle filters. For example, Zhou et al. [2] presented an approach that incorporated appearance-adaptive models to stabilize the tracker. They made three extensions: (a) an observation model arising

from an adaptive appearance model, (b) an adaptive velocity motion model with adaptive noise variance, and (c) an adaptive number of particles. Li et al. [3] proposed a robust observation model to address appearance changes. Wang et al. [4, 5] developed an SMOG appearance model and an SMOG-based similarity measure to deal with appearance variations. Zhang et al. [6] embedded an adaptive appearance model into a particle filter to address the appearance changes and proposed an occlusion handling scheme to deal with occlusion situations. On the other hand, some researchers have incorporated other optimization algorithms into particle filter to enhance the performance. For example, in [7], CamShift was used into the probabilistic framework of particle filter as an optimization scheme for proposal distribution such that both the tracking robustness and computational efficiency are improved. Shan et al. [8] incorporated the mean-shift (MS) optimization algorithm into a particle filter framework to improve the sampling efficiency. Zhou et al. [9] presented a scale invariant feature transform (SIFT) based mean shift algorithm for object tracking, which improved the tracking performance of the classical mean shift and SIFT tracking algorithms in complicated real scenarios. Zhao and Li [10] applied particle swarm optimization (PSO) to find

high likelihood areas where the particles could be distributed even though the dynamic model of the object could not be obtained. Zhou et al. [11] combined multiband Generalized Cross Correlation, KF, and Weighted Probabilistic Data Association within the particle filtering framework, which improves the performance of the algorithm in noisy scenarios. Most of the above methods applied a total model updating mechanism for template updating in which the initial template model is updated gradually based on the estimated information by particle filters. However, if an object is heavily occluded or its appearance changes abruptly, the total model updating based PF (TMU-PF) will gradually deviate from the target.

To tackle the drawback of the TMU-PF, Montemayor et al. [12] introduced memory strategies into PF to store the states of particles, which can deal with some occlusion situations. Mikami et al. [13] proposed a memory-based particle filter (MPF) to handle facial pose variation by predicting the prior distribution of the target state in future time steps. However, both of the methods are neither biologically motivated nor cognitively inspired. They just apply memory to store the states of particles and could not cope with situations with sudden changes.

It is well known that humans can track and recognize an object with little difficulty in the case of appearance changes and partial occlusions. This capability of human beings benefits from the human's memory system. When humans perceive something, the related information which is stored in their memory can be recalled. As a function of information retention organs in the brain, the mechanism of memory system has been extensively studied in neural science, biopsychology, cognitive science, and cognitive informatics [14, 15].

Inspired by the way humans perceive the environment, in this paper, we present a memory-based multiagent coevolution model for tracking the moving objects. The three-stage human brain memory mechanism is incorporated into a multiagent coevolutionary process for finding a best match of the appearance of the object. Each agent can remember, retrieve, or forget the appearance of the object through its memory system by its own experience. A number of such memory-based agents are randomly distributed nearby the located object region and then mapped onto a 2D lattice-like environment for predicting the new location of the object by their coevolutionary behaviors, such as competition, recombination, and migration. Experimental results show that the proposed method can deal with large appearance changes and heavy occlusions when tracking a moving object. It can locate the correct object after the appearance changed or the occlusion recovered.

The remainder of this paper is organized as follows. In Section 2, we will first propose the memory-based multiagent coevolution model including the definitions of each behavior involved. Section 3 gives the detailed description of the memory modeling of an agent and the object appearance template updating process for each agent. Then the color object modeling and the proposed tracking algorithm are described in Section 4. Finally, the performance of our tracking algorithm is verified on different standard video

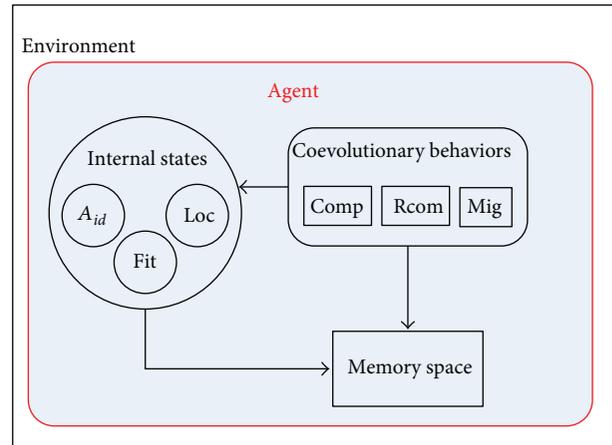


FIGURE 1: Memory-based agent model.

sequences and some conclusions are summarized in Sections 5 and 6.

2. Memory-Based Multiagent Coevolution Modeling

2.1. Memory-Based Multiagent Model. According to [16], an agent can be defined as an intelligent entity that resides in an environment and can act autonomously and collaboratively. It is driven by certain purposes and has some reactive behaviors. Based on this idea, many agent-based applications are reported during past years, such as image feature extraction [17], image segmentation [18], and optimization problems [19–24]. In our previous work [25, 26], we also proposed an evolutionary agent model for color-based face detection and location.

In this paper, we will present a memory-based multiagent model (MMAM) for moving object tracking. Each agent represents a candidate target region in a video frame; it lives in a lattice-like environment, and its main task is to compete or cooperate with its neighbor agents to continuously improve its own fitness by exhibiting its behaviors.

The schematic diagram of the proposed MMMA is shown in Figure 1.

More specifically, the memory-based multiagent model (MMAM) for object tracking can be defined as a 7-tuple: $\langle A_{id}, Loc, Fit, MS, Comp, Rcom, Mig \rangle$. Where A_{id} denotes the identity of an agent; Loc represents the position of an agent in the image, that is, the center of a candidate target; Fit symbolizes its fitness, which is defined by the similarity between the candidate target and the object template; and $MS = \{USTMS, STMS, LTMS\}$ is a set of hominine memory spaces of an agent for information storage, where $USTMS$, $STMS$, and $LTMS$ stand for the ultrashort-term memory space, short-term memory space, and long-term memory space, respectively.

The above 4 parameters describe the internal states of an agent. While $Comp$, $Rcom$, and Mig describe the external coevolutionary behaviors of an agent, where $Comp$ represents

the competition behavior, Rcom denotes the recombination behavior, while Mig refers to the migration behavior.

Suppose all the agents inhabit in a lattice-like environment, A , which is called an agent lattice, as shown in Figure 2. Each agent is fixed on a lattice point and it can only interact with its 4 neighbors. The size of A is $N \times N$ and the agent located at (i, j) is denoted by $A_{i,j}$, $i, j = 1, 2, \dots, N$. Each agent can compete or cooperate with its 4 neighbors in order to improve its fitness.

The mapping process is described as follows.

First, it randomly generates $N \times N$ agents near the located object region at beginning. The first generated agent is placed at $A_{1,1}$, the second agent is placed at $A_{1,2}, \dots$, the N th agent is placed at $A_{1,N}$, the $(N + 1)$ th agent is placed at $A_{2,1}, \dots$, and the final agent $(N \times N)$ th is placed at $A_{N,N}$. The neighbors of agent $A_{i,j}$ are defined as $Nb_{i,j} = \{A_{i-1,j}, A_{i+1,j}, A_{i,j-1}, A_{i,j+1}\}$. For the agents at the four edges of the lattice, we define

$$\begin{aligned} A_{0,j} &= A_{N,j}, & A_{i,0} &= A_{i,N}, \\ A_{N+1,j} &= A_{1,j}, & A_{i,N+1} &= A_{i,1}. \end{aligned} \quad (1)$$

According to the above definition, the neighbors of an agent on the lattice are not its real neighbors in the video image. Because each agent is generated randomly and can only evolve with its neighbors on the lattice-like environment, the mapping process can also be thought as a natural selection before their coevolution.

2.2. Multiagent Coevolutionary Behaviors. There are three coevolutionary behaviors for each agent, that is, competition, recombination, and migration. The three behaviors are defined as follows.

Definition 1 (Comp (competition behavior)). Comp means that an agent will contend with other agents for its survival.

For each agent $A_{i,j}$, if $\text{Fit}(A_{i,j}) < \text{Fit}(Nb \max_{i,j})$, where $Nb \max_{i,j}$ is the agent with maximum fitness among its 4 neighbors, then $A_{i,j}$ will be replaced by the following:

$$A_{i,j}^l = \begin{cases} \underline{l}, & (Nb \max_{i,j}^l + U(-1, 1) \times (Nb \max_{i,j}^l - A_{i,j}^l)) < \underline{l}, \\ \bar{l}, & (Nb \max_{i,j}^l + U(-1, 1) \times (Nb \max_{i,j}^l - A_{i,j}^l)) > \bar{l}, \\ Nb \max_{i,j}^l + U(-1, 1) \times (Nb \max_{i,j}^l - A_{i,j}^l), & \text{otherwise,} \end{cases} \quad (2)$$

where $U(-1, 1)$ is a uniform random number in $[-1, 1]$, l denotes the location of agent $A_{i,j}$ in the video frame, $l = (x, y)$, $L = [\underline{l}, \bar{l}]$ represents the whole searching space, that is, the video size, $\underline{l} = [\underline{x}, \underline{y}]$, $\bar{l} = [\bar{x}, \bar{y}]$.

Definition 2 (Rcom (recombination behavior)). Rcom means that an agent may exchange the x or y coordinate with other agents. It is similar to the crossover operator in genetic algorithms.

For each agent $A_{i,j}$, given a recombination probability P_r , if $U(0, 1) < P_r$, exchange the x or y coordinate of $A_{i,j}$ and $Nb \max_{i,j}$, a new agent will be created, $Ar_{i,j}$. If $\text{Fit}(A_{i,j}) > \text{Fit}(Ar_{i,j})$, $A_{i,j}$ will continue to exist in the lattice; otherwise it will be replaced by the following:

$$Ar_{i,j} = \begin{cases} (A_{i,j}^x, Nb \max_{i,j}^y), & U(0, 1) < 0.5, \\ (Nb \max_{i,j}^x, A_{i,j}^y), & \text{else.} \end{cases} \quad (3)$$

Definition 3 (Mig (migration behavior)). Mig means that an agent can move to another location by some random steps in the image other than the lattice it locates at. It is similar to the mutation operator in genetic algorithms.

For each agent $A_{i,j}$, the migration behavior will occur according to a migration probability P_m . if $U(0, 1) < P_m$, $A_{i,j}$ will be replaced by the following:

$$Am_{i,j}^l = \begin{cases} \underline{l}, & A_{i,j}^l + U^l(-10, 10) < \underline{l}, \\ \bar{l}, & A_{i,j}^l + U^l(-10, 10) > \bar{l}, \\ A_{i,j}^l + U^l(-10, 10), & \text{otherwise,} \end{cases} \quad (4)$$

where $U(-10, 10)$ is a uniform random number in $[-10, 10]$; that is, the migration steps are randomly generated within $(-10, 10)$ pixels for i and j , respectively.

3. Memory Modeling for an Agent

3.1. Three-Stage Human Brain Memory Modeling for Appearance Updating. As a faculty of information retention organs in the brain, memory has been intensively studied in psychology, neural science, and cognitive science, and several memory models have been proposed since the late 19th century. In 1890, James first divided the human memory into three components: after-image memory, the primary memory, and the secondary memory [27]. Atkinson and Shiffrin modeled the human memory as a sequence of three stages: the sensory memory, short-term memory, and long-term memory [28] (also known as the multistore model). Baddeley and Hitch proposed a multicomponent model of working memory where a central executive responsible for control processes and two slave systems providing modality-specific buffer storage [29]. Recently, Wang proposed a logical architecture of memories in the brain which includes four parts: (a) the sensory buffer memory; (b) the short-term memory; (c) the long-term memory; and (d) the action buffer memory [15, 30]. According to contemporary cognitive psychology, the popular model of a basic human brain memory includes three stages: ultrashort-term memory (USTM), short-term memory (STM), and long-term memory (LTM), as shown in Figure 3 [31].

Each stage includes three processes: (a) encoding, (b) storage, and (c) retrieval. ‘‘Encoding’’ (also referred to as registration) is the process of forwarding physical sensory input into one’s memory. It is considered as the first step in memory information processing. ‘‘Storage’’ is the process of retaining information whether in the sensory memory,

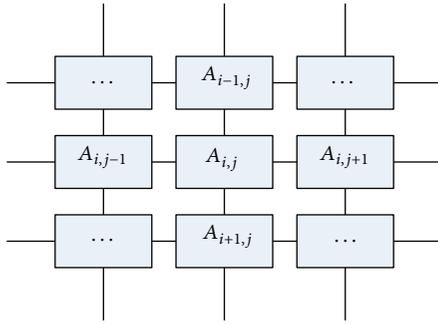


FIGURE 2: Model of the agent lattice.

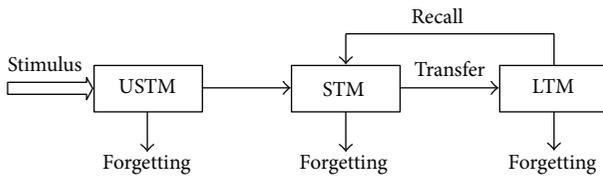


FIGURE 3: Three-stage human brain memory model.

the short-term memory, or the more permanent long-term memory. “Retrieval” (also referred to as “recall”) is to call back the stored information in response to some cues for use in a process or activity.

The memorization process can be described as follows.

- (1) USTM is used to store the basic cognitive information.
- (2) STM, which in the recent literature has been referred to as working memory, is used to make decision. The information stored in STM includes the new information from USTM, the information processed in STM, or the information recalled from LTM. Therefore, STM can be considered as a complicated system for information storing and processing.
- (3) LTM is a library used to store experienced knowledge which can inspire the individual to recall every thing that had happened, cognize all kinds of models, and solve problems (e.g., tracking problems in our work).
- (4) Forgetting is a special function of memory which helps the information either not always recalled or not commonly used to be lost from memory.

According to the above three-stage human memory model, the appearance template updating model of an agent can be described as shown in Figure 4,

where the input of the model is the candidate template estimated by the Loc of an agent in the current video frame while the output is the updated template for prediction in the next frame. USTMS, STMS, and LTMS represent the three-stage memories, respectively. They are defined as follows.

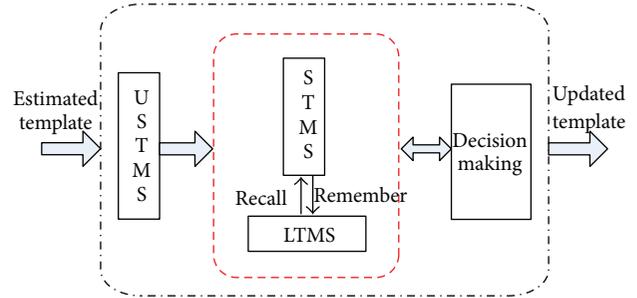


FIGURE 4: Three-stage memory model for appearance template updating.

Definition 4 (memory space (MS)). A 3-tuple which is used to store the current estimated appearance template and the past templates. Each element in MS is a memory space:

$$MS = \{USTMS, STMS, LTMS\}. \quad (5)$$

Definition 5 (USTMS). A one-element set for storing the estimated model p in the current video frame, which simulates the stage of ultrashort-term memory of human brain:

$$USTMS = \{p\}. \quad (6)$$

Definition 6 (STMS). A set of K_s temporary templates, which imitates the stage of short-term memory of human brain. Let q_i denote the i th template in STMS; then

$$STMS = \{q_i, i = 1, 2, \dots, K_s\}. \quad (7)$$

Definition 7 (LTMS). A set of K_l remembered templates, which simulates the dynamic stage of the long-term memory of human brain. Let q_{Mj} stand for the j th remembered template in LTMS:

$$LTMS = \{q_{Mj}, j = 1, 2, \dots, K_l\}. \quad (8)$$

The templates stored in STMS include the estimated template transferred from USTMS, the updated templates in STMS, or the templates recalled from LTMS.

According to the theory of cognitive psychology, only the information which is stimulated repeatedly can be stored into LTMS. Therefore, we define a parameter β for each template in STMS to determine whether the templates in STMS can be stored into LTMS or not, where β is a counter indicating the number of successful matches. The bigger β is, the more probably the template can be stored into LTMS.

More specifically, for all $q_i \in STMS, i = 1, 2, \dots, K_s$, If $q_i \cdot \beta > T_M$ (a predefined threshold), the template will be remembered and stored into LTMS.

The process of template updating can be briefly described as follows.

First, the estimated template of the current frame is stored into USTMS and checked against the current template in STMS (the first one). If they are matched, update the template; otherwise check against the remaining templates in STMS and then LTMS in turn for a match. If a match exists, it will

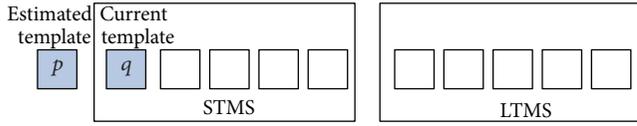


FIGURE 5: Initialization step.

be selected for the new template. Meanwhile the STMS and LTMS are updated by some behaviors, such as remembering, recall, and forgetting. These behaviors are defined as follows.

Definition 8 (remembering). An action that a template is stored into LTMS.

If there is no match in STMS and LTMS, and the STMS is full and the last template in STMS (denoted by q_{K_s}) is satisfied with $q_{K_s} \cdot \beta > T_M$, then q_{K_s} will be remembered into LTMS and replaced by q_{K_s-1} . In such a circumstance, the estimated template will be reserved for the next estimation.

Definition 9 (recall). An action that a matched template is loaded from LTMS.

If a match is found in LTMS, the matched template will be extracted and used as the current object template.

Definition 10 (forgetting). An action that a template is removed from either of STMS or LTMS.

If the LTMS is full and $q_{K_s} \cdot \beta > T_M$, the oldest template in LTMS will be forgotten in order to remember q_{K_s} .

3.2. Detailed Description of Memory-Based Appearance Updating. According to the above model, the memory-based appearance template updating algorithm can be described as follows.

Step 1 (Initialization). For each agent, store the estimated template (candidate object) p into the USTMS and the current template q into the STMS; set $q \cdot \beta = 1$ and the LTMS to be empty, where p and q are determined by the initial target region, as shown in Figure 5. It is worth mentioning that the STMS and LTMS will be filled up gradually after several time steps during tracking.

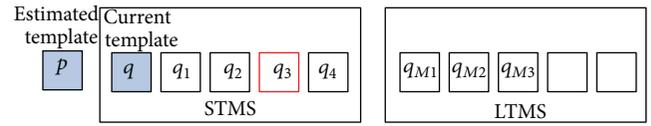
Step 2. Calculate the similarity coefficient $\rho = \rho[p, q]$, if $\rho > T_{dc}$, update the current object template by the following:

$$\begin{aligned} q &= (1 - \alpha)q + \alpha \cdot p, \\ q \cdot \beta &= q \cdot \beta + 1, \end{aligned} \quad (9)$$

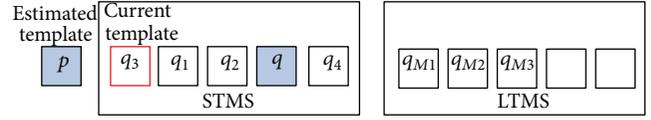
where T_{dc} is a predefined threshold for current template matching and α is the updating rate.

Step 3. If $\rho \leq T_{dc}$, check against the remaining templates in STMS for a match, if

$$\rho[p, q_i] > T_{ds}, \quad i = 1, \dots, K_s - 1, \quad (10)$$

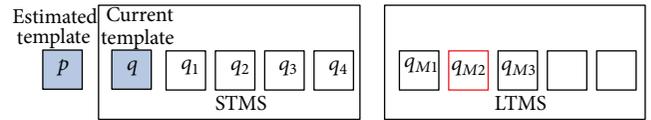


(a) A match is found in STMS

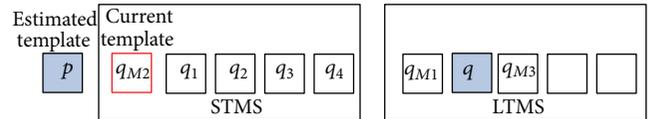


(b) Updating STMS

FIGURE 6: Illustration of updating process in STMS.



(a) A match is found in LTMS



(b) Updating STMS and LTMS

FIGURE 7: Illustration of recalling and remembering.

update the matched template by the following:

$$\begin{aligned} q_i &= (1 - \alpha) \cdot q_i + \alpha \cdot p, \\ q_i \cdot \beta &= q_i \cdot \beta + 1, \end{aligned} \quad (11)$$

where T_{ds} is the threshold for template-matching in STMS.

Then, exchange the current template and the matched one, as shown in Figure 6.

For example, if q_3 is a matched template found in STMS (as shown in Figure 6(a)), then it will be moved to the top location in STMS and used as the current template, while the previous current template q will be moved to the original location of q_3 as shown in Figure 6(b).

Step 4. If $\rho[p, q_i] \leq T_{ds}$, check in LTMS for a match, if

$$\rho[p, q_{Mj}] > T_{dl}, \quad j = 1, \dots, K_l, \quad (12)$$

where T_{dl} is the threshold for template-matching in LTMS. Then update the matched template by the following:

$$\begin{aligned} q_{Mj} &= (1 - \alpha)q_{Mj} + \alpha \cdot p, \\ q_{Mj} \cdot \beta &= q_{Mj} \cdot \beta + 1, \end{aligned} \quad (13)$$

and then recall the matched one to use as the new object template and remember the current template q , as shown in Figure 7.

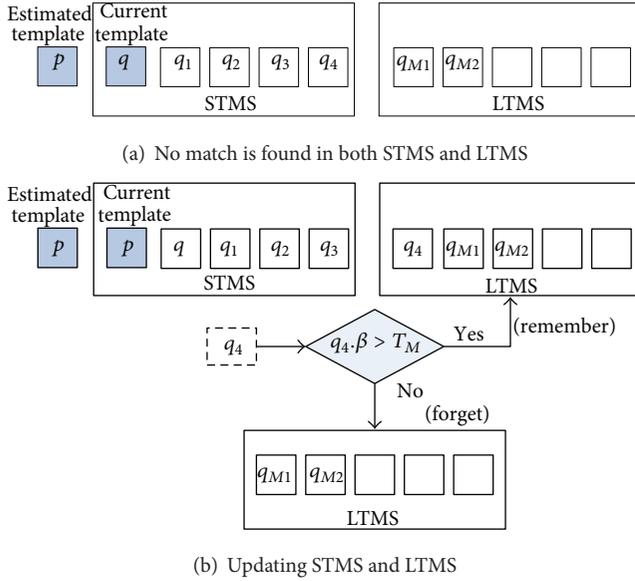


FIGURE 8: Illustration of updating STMS and LTMS when no match is found in both memory spaces.

Step 5. If $\rho[p, q_{Mj}] \leq T_{dl}$, it means that there is no any match in STMS and LTMS. The estimated template p is stored into STMS and used as the new object template (set $p.\beta = 1$), as seen in Figure 8. Meanwhile, if the STMS reaches its maximum capacity, remember or forget the oldest template in STMS (i.e., q_{K_s-1}) by the following substeps.

- (1) If $q_{K_s-1}.\beta > T_M$ and the LTMS is full, forget the oldest template in LTMS (i.e., $q_{M_{K_t}}$) and remember q_{K_s-1} .
- (2) If $q_{K_s-1}.\beta \leq T_M$, forget q_{K_s-1} .

As shown in Figure 8, when no match is found in both memory spaces, the current estimated template p is stored into STMS, while q_4 (i.e., $K_s - 1 = 4$) is either remembered ($q_4.\beta > T_M$) or forgotten ($q_4.\beta \leq T_M$).

Note that the templates in STMS and LTMS are stored in chronological order; that is, if a template is stored into STMS or LTMS earlier, it will move to the subsequent locations in order to make rooms for the newly reached templates.

4. Moving Object Tracking by MMAM

4.1. Object Detection and Modeling. To detect a color object, it is very important to obtain an effective color model to accurately represent and identify the object under various illumination conditions. In this paper, we use a histogram-based nonparametric modeling technique in YCbCr color space to model an object [32], which is much robust to lighting variations.

Giving the distribution of colors in an object region, let $px_{i,j}$, be a pixel location inside the object region with the origin at the center of the object region, the non-parametric

distribution of the object, Q , can be represented by the following [32]:

$$Q = \{q_u; u = 1, 2, \dots, m\}, \quad (14)$$

where

$$q_u = C \sum_{i=1, j=1}^{x,y} k(\|px_{i,j}\|^2) \delta[b(px_{i,j}) - u], \quad (15)$$

where k is the Epanechnikov kernel function, δ is the Kronecker delta function, and the function $b : R^2 \rightarrow \{1, \dots, m\}$ associates the pixel at location $px_{i,j}$ with its color's index $b(px_{i,j})$ in the histogram. The normalization constant C is derived by imposing the condition $\sum_{u=1}^m q_u = 1$.

Suppose P_y is the non-parametric distribution of the candidate object at position y in the image, then the similarity or Bhattacharyya coefficient can be decided by the following [32]:

$$\rho(y) = \rho[P_y, Q] = \sum_{u=1}^m \sqrt{p_u(y) q_u}. \quad (16)$$

For tracking by agents, $\rho(y)$ can be used to compute the fitness of an agent and the similarity coefficient between two appearance templates.

4.2. Implementation of the Tracking Algorithm. The memory-based multiagent model for object tracking can be described as follows.

Step 1. First locate the object in a video scene and then build the object appearance model by (14).

Step 2. Randomly generate $N \times N$ agents near the located object region by adding a 2D Gaussian distribution $G_{x,y}(0, 10)$, as shown in Figure 9(a), and then map the agents onto the 2D lattice-like environment.

Step 3. For each agent on the lattice, first retrieve the appearance template from its memory spaces, then compute the fitness of the agent, and then perform the competition, recombination, and migration behaviors when the object moves. A snapshot of multiagent coevolution is shown in Figure 9(b).

Step 4. Compute the final target by weighted averaging of all the agents on the lattice, and the tracking result after the end of coevolution is shown in Figure 9(c).

5. Experimental Results and Discussions

In this section, we aim to experimentally verify the efficacy of the proposed object tracking method. We compare the performance of the proposed method with the total model updating PF (TMU-PF) in practical tracking problems. We use some standard video sequences [33, 34] as testing dataset and the experiments are conducted on a computer with a P4 3.0 G Processor.

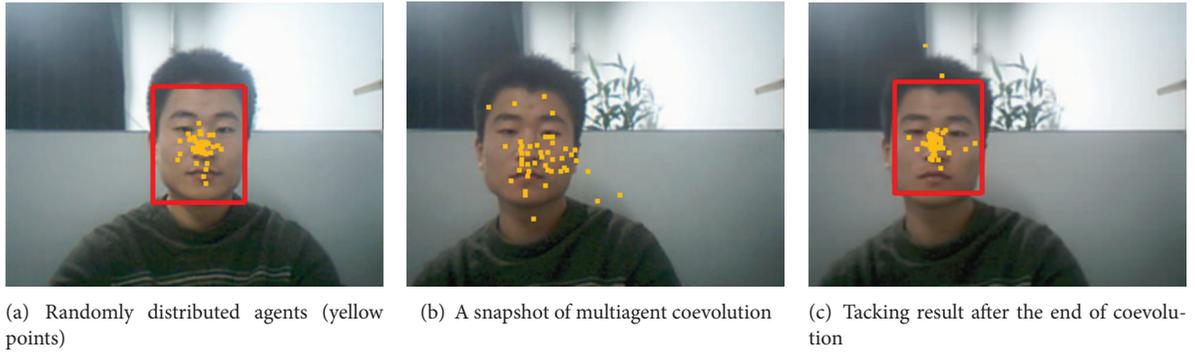


FIGURE 9: Object tracking by multiagent coevolution.

It is worth noting that the parameters for the algorithms are set initially as follows in our experiments:

- m is the number of the bins for modeling the object using histogram and is set as $m = 16 \times 16$;
- T_{dc} is used to measure the similarity between the estimated template and the current object template and is set as $T_{dc} = 0.9$;
- T_{ds} and T_{dl} are the thresholds used to find a match in STMS and LTMS, respectively, and are set as $T_{ds} = T_{dl} = 0.8$;
- K_s and K_l are the capacity of the STMS and LTMS, respectively, and are set as $K_l = K_s = 5$;
- T_M is a predefined threshold used to decide whether the template in STMS can be stored into LTMS or not and is initially set as $T_M = 1$;
- the total number of agents is 49; that is, the size of the lattice A is 7×7 , the recombination probability P_r is 0.6, and the migration probability P_m is 0.05.
- the number of the particles used in particle filter-based tracking is set as 50 (almost equal to the number of agents used).

5.1. Tracking a Person with Large Appearance Change. The first sets of experiments are to track a person with abrupt appearance changes. The video used in this experiment is clipped from the standard sequence “seq_dk” (The video sequences can be downloaded from <http://www.ces.clemson.edu/~stb/research/headtracker/seq/>) [33]. The tracking results of the man by traditional PF, TMU-PF, and the proposed method at frames 21, 58, 82, 83, 87, and 96 are shown in Figures 10(a), 10(b), and 10(c), respectively (the template is initialized manually). The human appearance changes very abruptly from frame 82 to frame 83. The results show that when the appearance is far from the initialized template, PF and TMU-PF deviate from the target gradually, while the original templates are remembered by the proposed method and when the appearance changes abruptly the relevant template can be recalled from the memory space of an agent.

Figure 11 displays experiments to track a person whose pose changes continuously in Head Pose Image Database

(The video sequences can be downloaded from <http://www-prima.inrialpes.fr/perso/Gourier/Faces/HPDatabase.html>) [34]. Experimental results show that our proposed method can track more precisely than the other two methods.

5.2. Tracking a Person with Heavy Occlusions by Others. The second set of experiments aims at tracking persons who are occasionally occluded by another object.

The sequence used in the first experiment is also a standard sequence “seq_jd” [33]. In this sequence, the man is occluded twice by another person. The tracking results by PF, TMU-PF and the proposed MMAM are shown in Figures 12(a), 12(b), and 12(c), respectively (the template is initialized manually). It is worth noting that the man is totally occluded at frame 52 and frame 253. The results show that the proposed MMAM can still track the person correctly after recovered from the occlusion at frame 55 and frame 256.

Figure 13 shows the results of tracking a face which is fully occluded by another person (The templates are initialized manually).

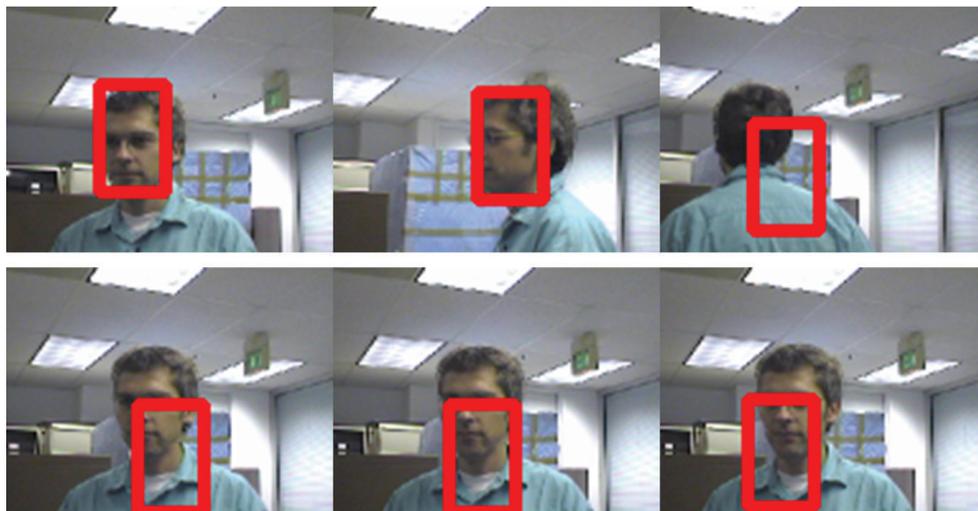
Finally, unlike the particle filter-based tracking method, the proposed approach has no restrictions to the face moving direction and speed. The face will be located and tracked at any time.

6. Conclusions

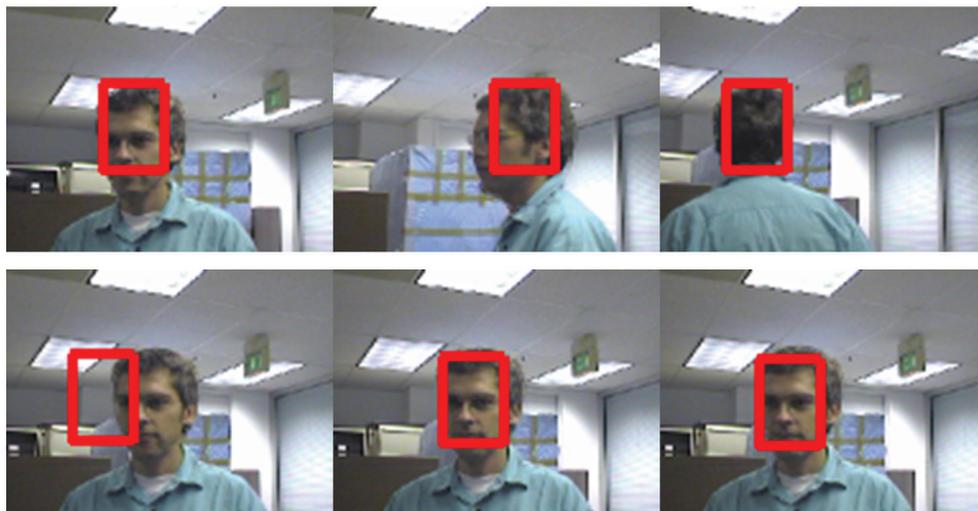
In this paper, we propose a different approach for visual tracking inspired by the way human perceive the environment. A number of memory-based agents are distributed nearby the located object region and then mapped onto a 2D lattice-like environment for predicting the new location of the object by their coevolutionary behaviors, such as competition, recombination, and migration, which imitate the process when many people search for a target in real world. The three-stage of human brain memory model is incorporated into a multiagent coevolutionary process for finding a best match of the appearance of the object. Each agent can remember, retrieve, or forget the appearance of the object through its memory system by its own experience. Experimental results show that the proposed method can deal with large appearance changes and heavy occlusions when tracking a moving object. It can locate the correct object



(a) Tracking results by PF



(b) Tracking results by TMU-PF

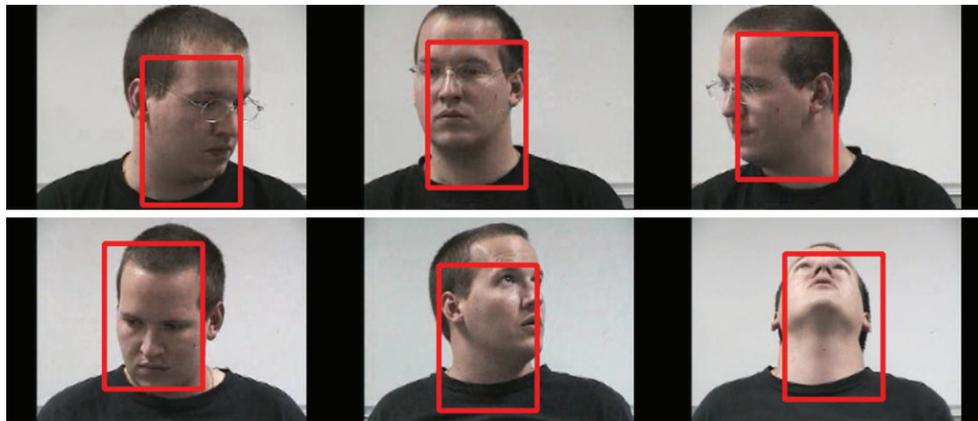


(c) Tracking results by MMAM

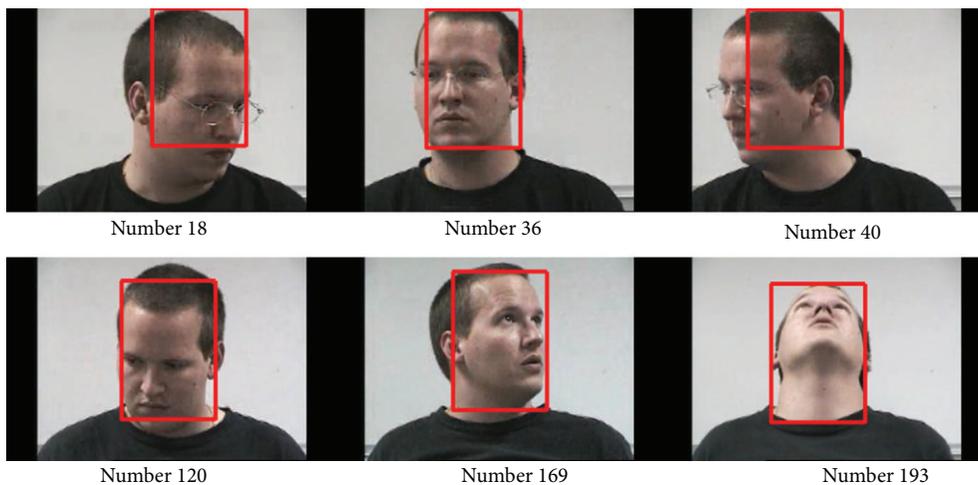
FIGURE 10: Tracking results of "seq.dk" sequence.



(a) Tracking results by PF

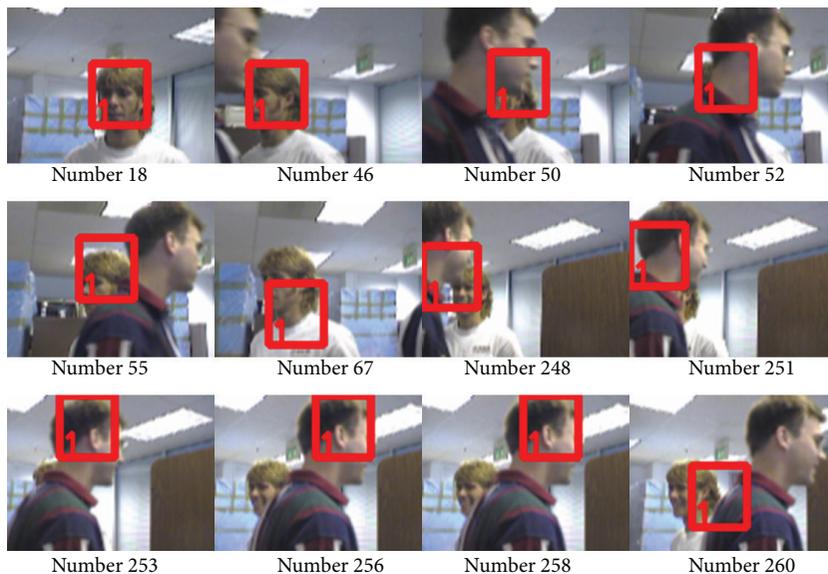


(b) Tracking results by TMU-PF

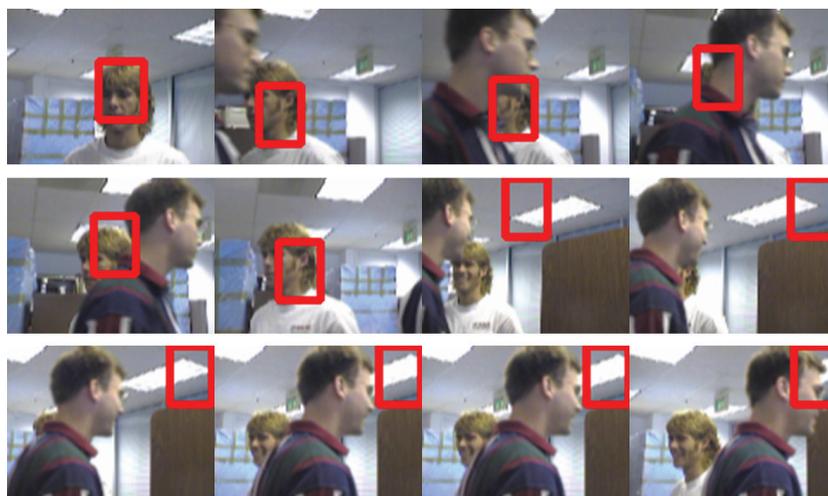


(c) Tracking results by MMAM

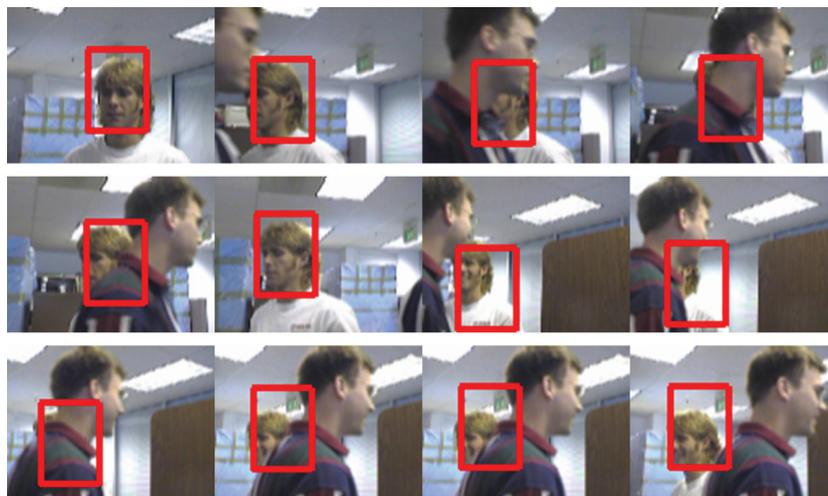
FIGURE 11: Tracking a person with pose changes.



(a) Tracking results by PF



(b) Tracking results by TMU-PF

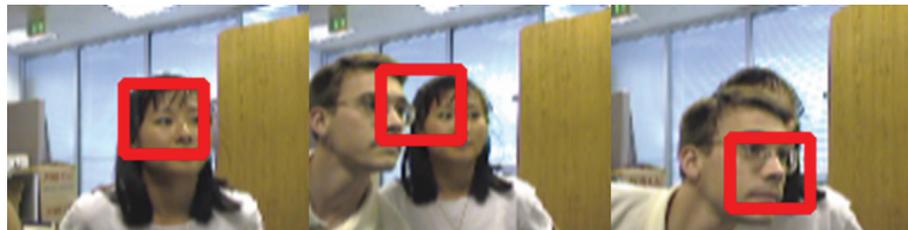


(c) Tracking results by MMAM

FIGURE 12: Tracking a fully occluded person.



(a) Tracking results by PF



(b) Tracking results by MTU-PF



(c) Tracking results by MMAM

FIGURE 13: Tracking a fully occluded face.

after the appearance changed or the occlusion recovered and outperforms the traditional particle filter based tracking.

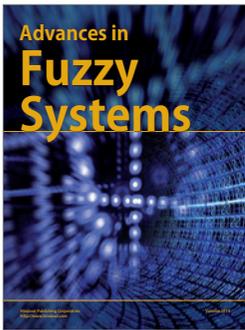
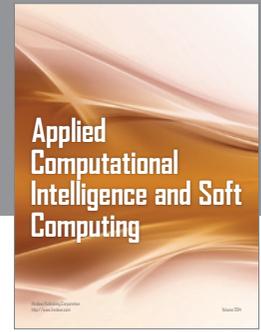
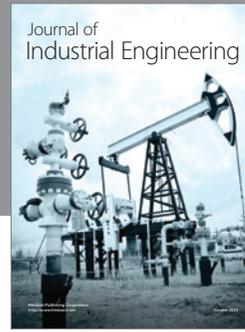
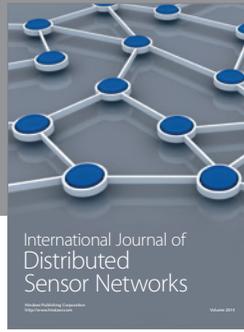
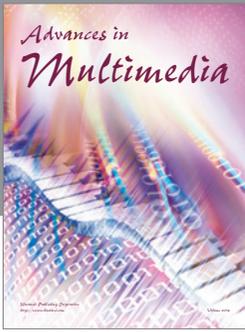
Acknowledgments

The paper is funded by the National Natural Science Foundation of China (no. 60873163, no. 61271407) and the Fundamental Research Funds for the Central Universities (no. 27R1105019A, no. R1405008A).

References

- [1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 1–45, 2006.
- [2] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1491–1506, 2004.
- [3] A. Li, Z. Jing, and S. Hu, "Robust observation model for visual tracking in particle filter," *International Journal of Electronics and Communications*, vol. 61, no. 3, pp. 186–194, 2007.
- [4] H. Wang, D. Suter, and K. Schindler, "Effective appearance model and similarity measure for particle filtering and visual tracking," in *Proceedings of the 9th European Conference on Computer Vision, Part III (ECCV '06)*, vol. 3953 of *Lecture Notes in Computer Science*, pp. 606–618, Graz, Austria, May, 2006.
- [5] H. Wang, D. Suter, K. Schindler, and C. Shen, "Adaptive object tracking based on an effective appearance filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1661–1667, 2007.
- [6] B. Zhang, W. Tian, and Z. Jin, "Robust appearance-guided particle filter for object tracking with occlusion analysis," *International Journal of Electronics and Communications*, vol. 62, no. 1, pp. 24–32, 2008.
- [7] Z. Wang, X. Yang, Y. Xu, and S. Yu, "CamShift guided particle filter for visual tracking," *Pattern Recognition Letters*, vol. 30, no. 4, pp. 407–413, 2009.
- [8] C. Shan, T. Tan, and Y. Wei, "Real-time hand tracking using a mean shift embedded particle filter," *Pattern Recognition*, vol. 40, no. 7, pp. 1958–1970, 2007.
- [9] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345–352, 2009.
- [10] J. Zhao and Z. Li, "Particle filter based on particle swarm optimization resampling for vision tracking," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8910–8914, 2010.
- [11] H. Zhou, M. Taj, and A. Cavallaro, "Target detection and tracking with heterogeneous sensors," *IEEE Journal on Selected Topics in Signal Processing*, vol. 2, no. 4, pp. 503–513, 2008.
- [12] A. S. Montemayor, J. J. Pantrigo, and J. Hernamdez, "A memory-based particle filter for visual tracking through occlusion," in *Proceedings of the International Work-Conference on the Interplay Between Natural and Artificial Computation, Part II (IWINAC '09)*, vol. 5602 of *Lecture Notes in Computer Science*, pp. 274–283, 2009.
- [13] D. Mikami, K. Otsuka, and J. Yamato, "Memory-based particle filter for face pose tracking robust under complex dynamics," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 999–1006, Miami, FL, USA, 2009.
- [14] Y. Wang and V. Chiew, "On the cognitive process of human problem solving," *Cognitive Systems Research*, vol. 11, no. 1, pp. 81–92, 2010.
- [15] Y. X. Wang, "Formal description of the cognitive process of memorization," *Transactions on Computational Intelligence*, vol. 1, no. 3, pp. 1–15, 2009.
- [16] M. Wooldridge and N. R. Jennings, "Intelligent agents: theory and practice," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.
- [17] J. Liu, Y. Y. Tang, and Y. C. Cao, "An evolutionary autonomous agents approach to image feature extraction," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 2, pp. 141–158, 1997.
- [18] E. G. P. Bovenkamp, J. Dijkstra, J. G. Bosch, and J. H. C. Reiber, "Multi-agent segmentation of IVUS images," *Pattern Recognition*, vol. 37, no. 4, pp. 647–663, 2004.
- [19] J. Liu, H. Jing, and Y. Y. Tang, "Multi-agent oriented constraint satisfaction," *Artificial Intelligence*, vol. 136, no. 1, pp. 101–144, 2002.
- [20] J. Liu, X. Jin, and K. C. Tsui, "Autonomy-oriented computing (AOC): formulating computational systems with autonomous components," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 35, no. 6, pp. 879–902, 2005.
- [21] K. C. Tsui and J. Liu, "An evolutionary multiagent diffusion approach to optimization," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 6, pp. 715–733, 2002.
- [22] W. Zhong, J. Liu, M. Xue, and L. Jiao, "A multiagent genetic algorithm for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 34, no. 2, pp. 1128–1141, 2004.
- [23] J. Liu, W. Zhong, and L. Jiao, "A multiagent evolutionary algorithm for constraint satisfaction problems," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 36, no. 1, pp. 54–73, 2006.
- [24] J. Liu, W. Zhong, and L. Jiao, "An organizational evolutionary algorithm for numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 37, no. 4, pp. 1052–1064, 2007.
- [25] Y. Wang and B. Yuan, "A novel approach for human face detection from color images under complex background," *Pattern Recognition*, vol. 34, no. 10, pp. 1983–1992, 2001.
- [26] Y. Wang and B. Yuan, "Fast method for face location and tracking by distributed behaviour-based agents," *IEE Proceedings*, vol. 149, no. 3, pp. 173–178, 2002.
- [27] W. James, *Principles of Psychology*, Holt, New York, NY, USA, 1890.
- [28] R. C. Atkinson and R. M. Shiffrin, "Human memory: a proposed system and its control processes," in *The Psychology of Learning and Motivation*, K. W. Spence, Ed., vol. 2, pp. 89–195, Academic Press, New York, NY, USA, 1968.
- [29] A. D. Baddeley and G. J. Hitch, "Working memory," in *The Psychology of Learning and Motivation*, G. H. Bower, Ed., vol. 8, pp. 47–89, 1974.
- [30] Y. X. Wang and Y. Wang, "Cognitive informatics models of the brain," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 36, no. 2, pp. 203–207, 2006.
- [31] M. W. Eysenck and M. T. Keane, *Cognitive Psychology: A Student's Handbook*, Psychology Press, New York, NY, USA, 6th edition, 2010.

- [32] C. Lersudwichai, M. Abdel-Mottaleb, and A. Ansari, "Tracking multiple people with recovery from partial and total occlusion," *Pattern Recognition*, vol. 38, no. 7, pp. 1059–1070, 2005.
- [33] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 232–237, Santa Barbara, Calif, USA, June 1998.
- [34] N. Gourier, D. Hall, and J. L. Crowley, "Estimating face orientation from robust detection of salient facial features," in *Proceedings of the Pointing International Workshop on Visual Observation of Deictic Gestures*, Cambridge, UK, 2004.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

