Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 138927, 11 pages
http://dx.doi.org/10.1155/2014/138927

Research Article

An Innovative SIFT-Based Method for Rigid Video Object Recognition

Jie Yu,[1] Fengli Zhang,[1] and Jian Xiong[2]

[1] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, No. 2006, Xiyuan Avenue, West Hi-Tech Zone, Chendu, Sichuan 611731, China*
[2] *Mathematics and Computer Science Department, Shangrao Normal University, No. 85, Zhiming Avenue, Shangrao, Jiangxi 334001, China*

Correspondence should be addressed to Jie Yu; yj19731201@126.com

Received 11 April 2014; Revised 5 June 2014; Accepted 9 June 2014; Published 6 July 2014

Academic Editor: Jer-Guang Hsieh

This paper presents an innovative SIFT-based method for rigid video object recognition (hereafter called RVO-SIFT). Just like what happens in the vision system of human being, this method makes the object recognition and feature updating process organically unify together, using both trajectory and feature matching, and thereby it can learn new features not only in the training stage but also in the recognition stage, which can improve greatly the completeness of the video object's features automatically and, in turn, increases the ratio of correct recognition drastically. The experimental results on real video sequences demonstrate its surprising robustness and efficiency.

1. Introduction

In recent years, security surveillance systems being called "sky-eye" and hand-held video cameras have increasingly grown in popularity, and the need of applications such as video-based object recognition and tracking or retrieval [1, 2] goes up rapidly. However, to identify a (identical) 3D object in videos or image sequences is still a challenging problem mainly because a 3D object's visual appearance may be different due to viewpoint or lighting changes.

For example, in Figure 1, there is a series of frame images captured from a video clip in which the vehicle is turning. If only (e) is used as the training image, then the vehicles in (h) or (t) would not be correctly recognized due to the changing of the viewpoints. Even for a human being with high sense of responsibility, just only providing him with (d) as the training image, he cannot confirm also that the cars in (h) or (t) are identical to the one in (d). However, when browsing the video clip, the source of Figure 1, everyone having normal cognitive ability can tell that the cars in (a) ~ (t) are all identical easily. Why?

With the help of selected regions in Figure 1, let us describe briefly what happens in the video browsing process of human being to serve as a modest spur to introduce the novel method proposed in this paper to come.

Before browsing the video clip, someone else, who has known the target object, should tell us which object is the target, for example, the selected region in frame image (a). Then we focus on the target object and try to dig its typical features out for saving—this part corresponds to the feature initialization of the target object. After that we keep going on to the next frame. What will we do with it? Firstly, we try to judge whether the target object is in it or not. Secondly, if the target objects are recognized in the frame, then we judge whether any new features of the target arise or not. However, how do we do that? Do we just only use incomplete target object features extracted just before? The answer is no. For our human beings, we do it by using both features and moving trajectories matching together. For example, comparing to the selected region in frame (a), we can read that although frames (b), (c), and (d) are not continuous, the cars in them not only share many features, but also keep going on the same

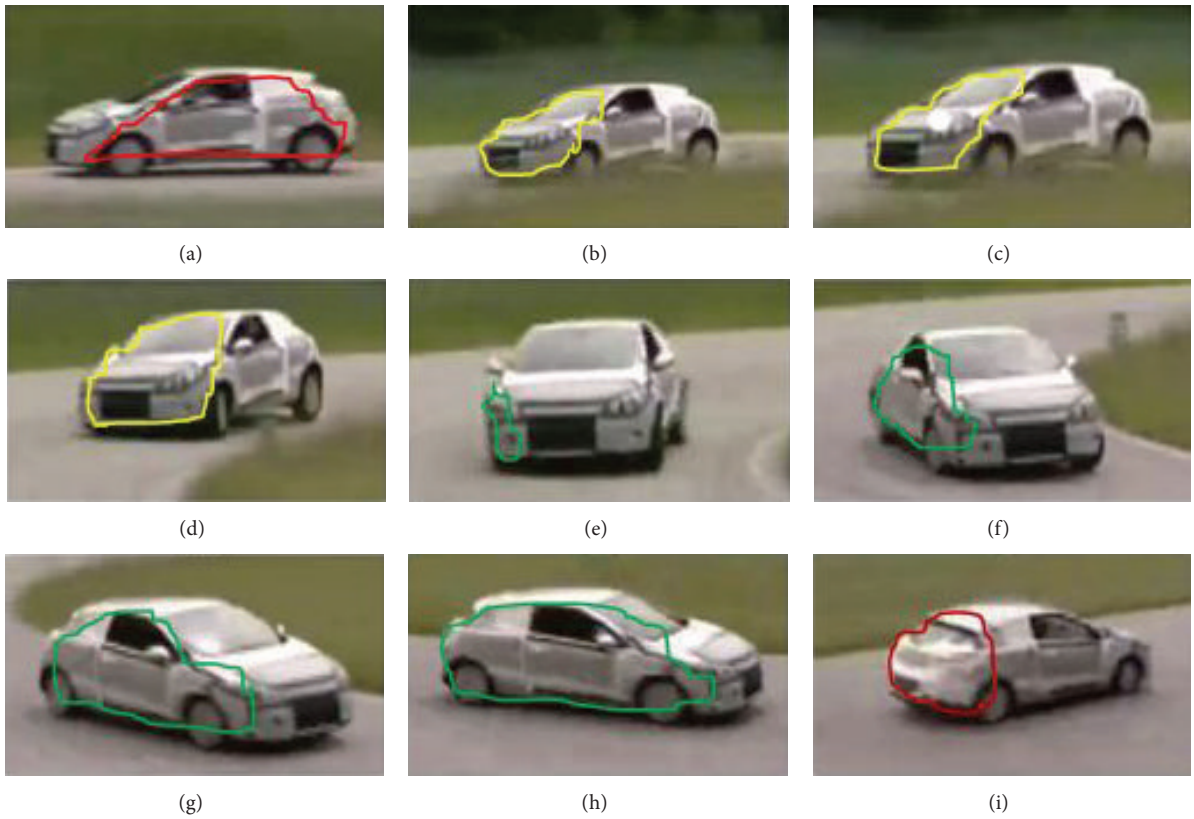

Figure 1: Frames captured from a surveillance video in which the vehicle is turning.

turning trajectory, so a conclusion can be drawn that cars in them are all identical with high probability. Meanwhile, we can find that regions in them, being surrounded by yellow line, seem to be similar and moreover keep moving on a similar trajectory, so we can conclude that they are part of the target car with high probability also. Then, new features can be extracted from the regions and be used for further recognition.

Processes of object recognizing and feature updating are being executed alternately and iteratively, and eventually all vehicles would be recognized and most of the new distinctive features would be extracted and saved.

Inspired by this physiological process of human being, a novel method for rigid video object recognition is proposed in this paper; its main contribution lies in the following.

(1) Modeled on the human recognition system, it makes the object recognition and feature updating processes organically unify together, which means that feature extraction and updating can be done not only in training stage, but also in recognition stage, which can improve greatly the completeness of features of the target video object and can in turn increase the ratio of correct recognition dramatically.

(2) Its object recognition is based on models of both feature and trajectory matching, which improve greatly the accuracy of the identification.

(3) Even if provided with only a single training image, it can create a relatively complete model of the target 3D object, using multiple 2D views automatically.

This paper is organized as follows. In Section 2, related researches are reviewed. In Section 3, the initialization of the target video object's feature database is given. In Section 4, feature point's trajectory is discussed and the iterative object is recognized, and then feature updating process is described. The experimental setup is presented and the analysis of the results is given in Section 5. Finally, in Section 6, conclusions are drawn.

2. Related Researches

There are lots of researches for 3D object recognition in which most of them are to model 3D objects using multiple 2D views. For example, in [3, 4], a method is proposed for combining multiple images of a 3D object into a single model representation; however, this approach requires that the single target object should occupy the majority of each of the training view images, which makes it meaningless in practice; the other most primary approaches are to get and describe the object's stable surface features from 2D view images, such as color feature [5], texture feature, shape feature [6], and contour feature [7].

Another direction of research is to use automatic background segmentation [2, 8], which digs the moving objects

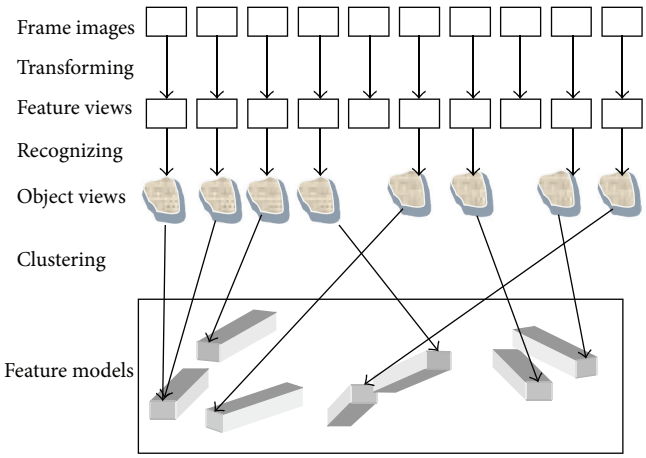

Figure 2: The Relationship between frames and feature models. Each frame is transformed into feature view; similar object views contained in different frames are clustered into one or more feature models.

out from the scenes first and then does recognition. However this can only work well with videos in which there is no dramatic background changing.

Motivated by biological vision systems, as this paper is, some simulating approaches are proposed in the long history of research. In [9], a method is proposed in which a scene is analyzed to produce multiple feature maps which are combined to form a saliency map which is used to bias attention to regions of the highest activity. And then some adjustments and improvements have been suggested to [9] by Itti et al. [10, 11].

Our work seemingly shares some themes with [12, 13] in the fact that features are learned from image sequences and/or video frames automatically. However, there are major differences between them; that is, in [12], its goal is to optimize the parameters of the known features in the tracked patches, not to learn new features of the targets; for example, for car detection, it can only extract and track features in the fixed manually labeled car's image area and optimize parameters of the known features in corresponding different frame, but it cannot learn new features of the target car from the correlations between the video frames, and in [13], its goal is to utilize offline feature tracking to observe feature transformations under large camera motions and then one can construct the database accordingly, keeping fewer images of each scene than would otherwise be needed; that is, its essence lies in using feature transformation to reduce storage costs, not feature learning either.

3. Feature Database Initialization

In this paper, the target object's feature database is organized as a set of feature models which represent different views of the target object, and each model consisted of all stable features extracted from corresponding object views which is contained in corresponding frame image. The relationship between frame image, object view, and feature model is shown in Figure 2. Furthermore, feature models are linked with each other by their sharing features. For example, in Figure 1, the vehicle views in frame images (a) ~ (d) seem to be similar and share many features, so four feature models may be built in feature database accordingly and may be linked with each other by their sharing of features. Of course, (b) and (c) seem to be identical, and maybe only one feature model needs to be built according to them. How do we estimate the degree of the similarity between object views and then decide how many feature models should be created accordingly? Refer to Section 4 please.

To initialize the feature database, it is needed to specify the target object firstly, that is, to specify the representative region of the target object firstly and then to initialize the object's feature database with it.

3.1. To Specify the Target Object. However, as shown in Figure 3, there are six selection modes; then which type is better or the best?

Before drawing a conclusion, we should figure out the meaning of the selected region to the novel recognition method thoroughly. Firstly, the region is the representative of the target object and the feature points in it are feature seeds for the next stage, so all pixels selected should belong to the target object; meanwhile, the more pixels to be selected and the more distinctive they are, which means more potential feature points, the better. Secondly, all feature points in the selected region are used to recognize the object views in the next frames, so the more the duration of the region keeping in the next frames, the better.

Based on the consideration above, the selection modes are all desirable except (b), meanwhile, of all the option, mode (e) and (f) seems to be the better, mode (c) seems to be the best. However, the mode (c) seems to be too complicated to operate, and the extraction of the point features in (c), (d), and (f) involves the operation of padding with zero pixels, so taking the convenience of the select operation and the computation complexity into account additionally, we prefer the mode (e).

3.2. To Initialize Feature Database. After specifying the representative region of the target object, to allow for efficient matching between the region and frames in video clips (namely, the training or being searched video clips), the selected region and the video frames should be represented as a set of stable features, using some kind of feature descriptor.

In this paper, RGB-SIFT [15] is adopted to compute features. For the RGB-SIFT descriptor, SIFT descriptor is computed for each of the three color channels (R/G/B), respectively. To see the details of the transformation process, refer to [15] please.

As Figure 4 showed, after the processing with the above steps of RGB-SIFT, the selected region is transformed into stable features and saved into a feature model in the target object's feature database directly. Meanwhile, each of the frames in videos is transformed into feature view and saved into the temporal feature database, respectively.

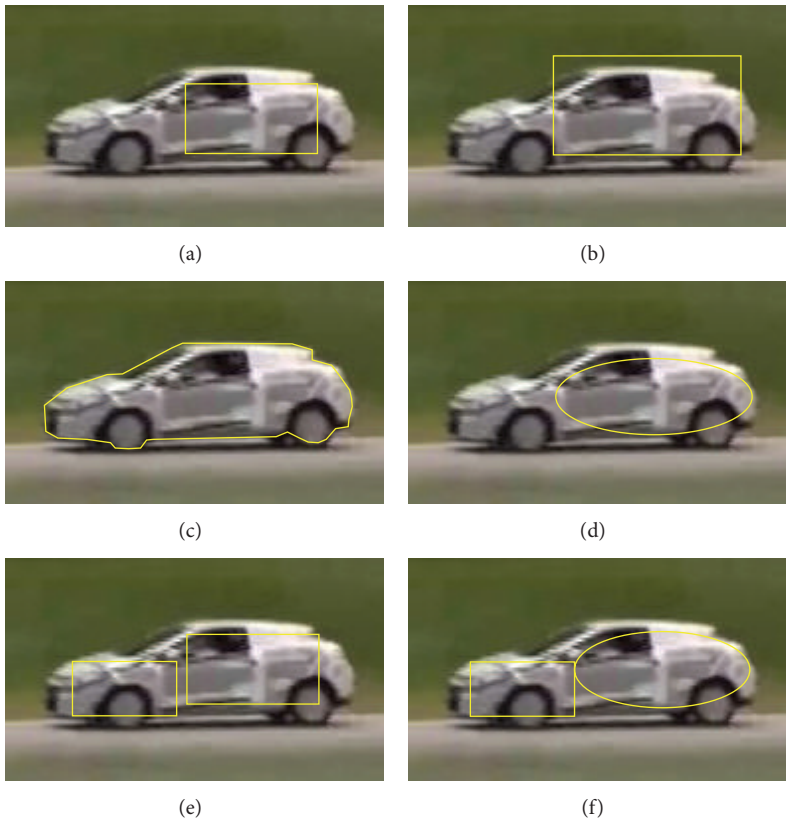

Figure 3: The region selection modes of the target object.

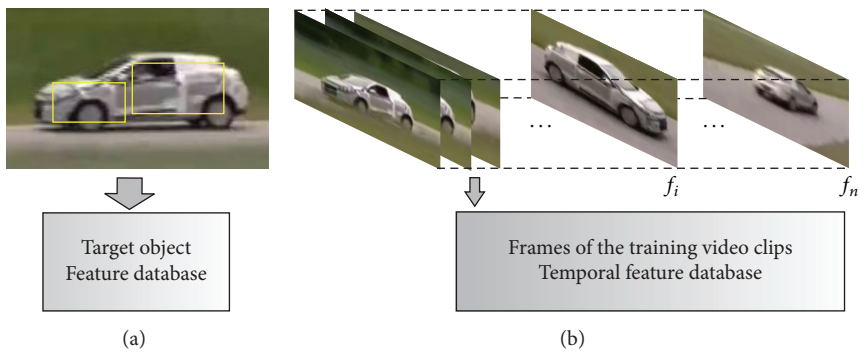

Figure 4: Features database initialization using some kind of feature descriptor.

Each SIFT feature, along with a record of the location, orientation, and scale, represents a vector of local image measurements in a manner that is invariant to scaling, translation, changes in illumination of the scene, and limited rotation of the camera's viewpoint. The size of frame image region that is sampled for each feature can be varied, but the experiments described in this paper all use a vector of 3×128 samples for each feature to sample 8 gradient orientations over a 4×4 sampling region in each color channel image. A typical frame image may produce several thousand overlapping features at a wide range of scales that form a redundant representation between the adjacent frame images.

4. Video Object Recognition Accompanied by Feature Database Updating

After the initialization stage described in the above section, in the object's feature database, there is only one feature model which consisted of all feature vectors extracted from the

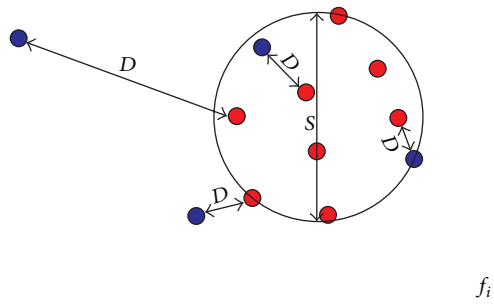

Figure 5: The diameter $S(fv_i)$ of the minimum circle covering all the matched feature keypoints in $\mathrm{Fsq}(fv_i)$ represents approximately the dimension of the target object in f_i, and D denotes the Euclidian distance from the candidate feature point to the nearest feature keypoint in the circle.

selected region. Obviously, one feature model is not enough for video object recognizing, and something should be done to enrich the feature database. Just like human being does, the process of video object recognition proposed in this paper is accompanied by object feature updating, and its main work includes the following.

(1) Video object recognition: to recognize the target object views in video frames, using both feature and trajectory matching.

(2) Feature database updating: to enrich feature models and features in it with different views of the target object contained in corresponding frames, using both feature and trajectory matching also.

The initial feature model and features in it can be used as seeds to do these works.

4.1. Feature Point Trajectory Matching. Thankfully, according to affine camera model [16, 17], each of the feature points of the target object may be a trajectory point; that is, the motion trajectory of the target object is shown as its feature point trajectories.

That is, to suppose that M feature points have been tracked between frame views f_i and f_j, then Ψ^{ij}, the corresponding affine transform matrix, can be worked out and $\overline{e}_{f_i,f_j}$, the average error of trajectories matching between them, can be estimated.

The point is that $\overline{e}$ can be used to evaluate the similarity between object views and models, which is the foundation for object recognition and feature database updating with corresponding feature views in this paper.

4.2. Feature Keypoint Matching. A set of matching feature keypoint pairs can be gained by efficient feature matching between feature models and feature views. This is the base for recognizing candidate video object views in corresponding frame images. According to [18], the best candidate match for each feature keypoint is found by identifying its nearest neighbor which is defined as the keypoint with minimum Euclidean distance for the invariant feature vector.

No algorithm is known for being any more efficient than exhaustive search in identifying the exact nearest neighbors of points in high-dimensional spaces. Our RGB-SIFT keypoint descriptor has a 3×128-dimensional feature vector; therefore, we have used an approximate algorithm, called the Best-Bin-First (BBF) algorithm [19]. This is approximate in the sense that it returns the closest neighbor with high probability. To see the details of keypoint matching, refer to [19] please.

This feature matching process between feature models and views is expressed as function feature point match (feature models, feature view, etc.) in this paper.

4.3. Notations Defined Specifications. For the sake of describing the procedure of video object recognition and feature updating conveniently, we adopt the following notations:

(1) f_i: the ith frame image in video;

(2) Q: the feature database of the target object in which all the known feature models are saved;

(3) TQ: the temporal feature database in which all the feature views are saved;

(4) fv_i: the ith feature view in TQ;

(5) Model(i): the ith feature model in Q;

(6) SSW: the scalable sliding window in which feature views being recognized with an identical feature model are saved temporally;

(7) SQ, Fsq(): feature sets in which the matching features in the matched feature model and feature view are saved, respectively;

(8) $S(fv_i)$: the dimension of the area occupied in the frame view fv_i by the known feature points in $\mathrm{Fsq}(fv_i)$. The value of it is approximated by the diameter of the minimum circle which covers all the feature points in $\mathrm{Fsq}(fv_i)$. The circle can be gained by using the Hough transform method [20] (see Figure 5);

(9) $D_{f_i}(p)$: the distance of one candidate feature point p to the known region of the target object in frame f_i. Its value is approximated by its Euclidean distance to the nearest keypoint in $\mathrm{Fsq}(fv_i)$ (see Figure 5).

4.4. Video Object Recognition in One Frame. Feature models in the object's feature database represent corresponding views of the target video object. According to [3], they can be used to recognize similar views of the object over a range of rotations in depth of at least 20 degrees in any direction.

Now, to suppose that a feature view fv_i has matched with a feature model model(j), using the feature matching procedure described in Section 4.2, and the matching features (≥ 3) between them are already saved, respectively, in $\mathrm{Fsq}(fv_i)$, SQ, then the procedure to recognize the target video object in fv_i can be described in Algorithm 1.

4.5. Feature Database Updating. Once object views are recognized in corresponding frame images, then the process of

(1) If number(Fsq(fv_i)) ≥ 3 then
(2) {
(3) Set $SQ \leftarrow SQ \cap$ Fsq(fv_i)*//to eliminate feature points without matching with any feature points in Fsq(fv_i) to ensure the similarity property between the recognized object views next*
(4) Set $n \leftarrow 0$
(5) Set recognized ← false
(6) Set $T1 \leftarrow$ a threshold value *//for example, 0.8*
(7) Set $T2 \leftarrow$ a threshold value *//for example, 0.8*
(8) Calculate $S(Fv_i)$ with feature points Fsq(fv_i) *//to estimate the dimension of the target object in f_i, using the Hough Transform method with feature points in Fsq(fv_i)*
(9) $k \leftarrow$ count the number of feature models linked by features in SQ
(10) For each of feature models linked with features in SQ do
(11) {
(12) Calculate $S(SQ)$ with feature points in SQ *//to estimate the dimension of the target object in the feature model.*
(13) $M \leftarrow$ Count the number of feature keypoints in the minimum circle determined by $S(SQ)$
(14) Calculate $\bar{e}_{SQ,f_i}$*//to estimate the residual error which shows the degree of the similarity between views that SQ and f_i implying, denoted by model(SQ) and model(f_i) respectively*
(15) If $\bar{e}_{SQ,f_i} \leq 0.25 \times \max(S(SQ), S(fv_i))$ and $(\text{num}(\text{Fsq}(fv_i)))/M \geq T1$ then
(16) {
(17) $n = n + 1$
(18) }
(19) }
(20) If $(n/k) \geq T2$ then recognized ← true
(21) Ouput f_i
(22) }
(23) Return $S(SQ)$, $S(Fv_i)$

Algorithm 1: Recognizinginframe(output: recognized or not, $S(SQ)$, $S(Fv_i)$; input: SQ, Fsq(fv_i)) *//to judge whether the target video object is in the frame f_i or not.*

feature database updating with them can be started immediately. The video object feature database consisted of feature models in which features are extracted from corresponding object views. Then the procedure of feature database updating includes two different level aspects.

(1) Feature models updating: to enrich feature models with different views of the target object contained in corresponding feature views.
(2) Model features updating: to enrich features in feature models with new features found in the corresponding similar feature views.

4.5.1. Feature Model Updating. To update feature models, it is required that at least one similar view of the target object is recognized in corresponding feature view with a specific feature model.

Now, the feature view fv_i is recognized with a feature model, and the matching feature points between them are already saved, respectively, in Fsq(f_i), SQ, and then the procedure to enrich the feature model with fv_i can be described in Algorithm 2.

4.5.2. Model Features Updating. To update features in corresponding feature models, it is required that at least two similar views of the target video object are recognized.

Also, to suppose that feature views fv_i and fv_j are recognized with a feature model in Q, and the matching features between them are already saved, respectively, in Fsq(fv_i), Fsq(fv_j), SQ, then the procedures to enrich features in corresponding models can be described in Algorithm 3.

Why is the constant multiplier in line 12 of Algorithm 1, line 2 of Algorithm 2, and lines 3 and 10 in Algorithm 3 0.25, 0.05, 0.25, and 0.85, respectively? According to [21], if we imagine placing a sphere around an object, then rotation of the sphere by 30 degrees will move no point within the sphere by more than 0.25 times the projected diameter of the sphere, and for the examples of typical 3D objects used in [16, 21], an affine solution works well with allowing residual errors up to 0.25 times the maximum projected dimension of the object. In addition, $S(fv_i)$ is less than the actual projected diameter of the target object in f_i generally, so the constant multipliers adopted in this paper would work well too. Thankfully, the experimental results support them.

4.6. General Procedure of the Video Object Recognition. So far, methods for recognizing the target video object in one frame and updating the feature database with one or two frames have been described in the above paragraphs.

So, we can write out the integrated cyclic procedure of recognizing the video object accompanied with updating the feature database briefly in Algorithm 4.

(1) Calculate $\overline{e}_{SQ,f_i}$ *//to estimate the residual error which shows the degree of the similarity between the object feature models that SQ and* f_i *implying, denoted by model*(SQ) *and model*(f_i) *respectively*
(2) set $T = 0.05 * S(fv_i)$ *//to set a lower limit to the degree of their similarity between model*(SQ) *and model*(f_i)
(3) if $\overline{e}_{SQ,f_i} > T$ then
(4) {
(5) New(model(f_i), Q) *//to create a new feature model in* Q *and save features in* $Fsq(fv_i)$ *into it*
(6) Link(model(f_i), model(SQ)) *//to link model*(f_i) *to model*(SQ) *withall matching features between them*
(7) }
(8) Else
(9) {
(10) Combine(model(f_i), model(SQ)) *//to combine model*(f_i) *with model*(SQ), *which means the new features from* f_i *should be added to the existing model model*(SQ)
(11) }
(12) Endif
(13) return Q

Algorithm 2: FModelUpdating(Output: Q; Input: Q, SQ, $S(fv_i)$, $Fsq(fv_i)$) //*to update feature models in* Q *with recognized frame image* f_i.

(1) Calculate $X^{i,j}$ with $Fsq(fv_i)$, $Fsq(fv_j)$ *//to calculate the column vector determined by the transform matrix with matching feature point pairs in* $Fsq(fv_i) \cap Fsq(fv_j)$
(2) Calculate $\overline{e}_{f_i,f_j}$ *// to estimate the residual error between model*(f_i) *and model*(f_j), *subjecting* $X^{i,j}$ *and feature point pairs in* $Fsq(fv_i) \cap Fsq(fv_j)$ *to equation* $\overline{e}$
(3) If $\overline{e}_{f_i,f_j} \geq 0.05 * \min\left(S(fv_i), S(fv_j)\right)$ then *//To determine whether there is a perceptible change between* f_i *and* f_j *or not*
(4) {
(5) FeaturepointMatch(Output: output1, output2; Input: fv_i, fv_j) *//to match features between* fv_i *and* fv_j *and the matching feature keypoints are all saved temporally in Output1and Output2 respectively except features in* $Fsq(fv_i)$ *and* $Fsq(fv_j)$
(6) For $k \leftarrow 1$ to Num(FeaturepointMatch.Output) do *//for each matching feature keypoint pair* $\left(p_k^{f_i}, p_k^{f_j}\right)$ *in the Outputs*
(7) {
(8) Calculate D_{f_i}, D_{f_j} *//to estimate the corresponding distance of this candidate feature points to the known area of the target object in* f_i *and* f_j, *respectively*
(9) Calculate $e_{f_i,f_j} = \left\| \begin{pmatrix} u^i & v^i & 0 & 0 & 1 & 0 \\ 0 & 0 & u^i & v^i & 0 & 1 \end{pmatrix} X^{i,j} - \begin{pmatrix} u^j \\ v^j \end{pmatrix} \right\|$ *//to evaluate the degree of agreement between this feature point pair.* $(u^i \quad v^i)$, $(u^j \quad v^j)$ *represent locations of the candidate matching feature point in* fv_i, fv_j *respectively*
(10) If $D_{f_i} \leq 0.25 * S(f_i)$ and $D_{f_j} \leq 0.25 * S(f_j)$ and $e_{f_i,f_j} \leq 0.85 * \overline{e}_{f_i,f_j}$ then
(11) {
(12) AddFeaturetoModel(($p_k^{f_i}$, model(f_i)); $p_k^{f_j}$, model(f_j)) *//if its distance is less than 0.25 times the projected diameter of the known area of the target object in corresponding frame view and the residual error of the projection is less than 0.85 times the average residual error, then add them into corresponding feature models, however, they will be discarded if already existed. By the way, the two models may point to a same feature model*
(13) }
(14) }
(15) }
(16) Return Q

Algorithm 3: FFeatureUpdating(Output: Q; Input: SQ, $Fsq(fv_i)$, $Fsq(fv_j)$) //*to search new features belonging to the target video object using features in* $Fsq(fv_i)$ *and* $Fsq(fv_j)$, *and then to add them into corresponding feature models in* Q *or discard.*

```
(1) T ← True //to initiate the loop variable
(2) While T Do //to begin the updating loop
(3) {
(4)   Set SQ ← Q, n ← 1
(5)   For i ← 1 to Num(TQ) do  //Num(TQ) represents the number of frame views in Tq.
(6)   {
(7)    Featurepointmatch(Output: SQ, Fsq(f_i); Input: SQ, fv_i) //to gain matching feature
       keypoint pairs between the feature view fv_i and feature models
(8)    Recognizinginfrmae(output: recognized or not, S(SQ), S(Fv_i); input: SQ, Fsq(f_i))
       //to confirm whether the frame image f_i contain the target object or not.
(9)    If recognized then
(10)   {
(11)    SSW(n) ← fv_i, n ← n + 1 //to save the corresponding feature views of the frame image
        into the scalable sliding window, as Figure 6 shown.
(12)    FModelUpdating(Output: Q; Input: Q, SQ, S(fv_i), Fsq(fv_i)) //to update feature models
        in Q with recognized feature view fv_i
(13)   }
(14)  }
(15)  If Empty(SSW) then //to judge the scalable sliding window SSW is empty or not
(16)   {
(17)    Break //to jump out the loop
(18)   }
(19)  Else
(20)   {
(21)    For i ← 1 to n − 1 do
(22)    {
(23)     For j ← i + 1 to n do
(24)     {
(25)      FFeatureUpdating(Output: Q; Input: SQ, Fsq(fv_i), Fsq(fv_j))
(26)     }
(27)     Delete(TQ, fv_i) //to delete feature view fv_i from the temporal feature database TQ
(28)    }
(29)   }
(30)  Dump(SSW) //to empty SSW
(31) }
(32) return Q
```

Algorithm 4: RecognizingthenUpdating(Output:recognized frames, Q; Input: Q, TQ) //*to find frames that contain the target object and update feature database with them iteratively*.

After this cyclic procedure of recognizing and then updating, most of the frames containing the target object can be recognized.

5. Experimental Results

The RVO-SIFT method with recognizing and then updating mechanism has shown its better abilities in experiments. The following experimental results are obtained on a computer with AMD Athlon 64 X2 2.6 GHz processor and 4G memory.

In order to fully demonstrate the ability of RVO-SIFT to acquire new features of the target video object, which is the key contribution of this paper, we use an about 2-minute-long surveillance video clip as the training video in which a Renault Megane comprehensive performance is testing and another about 20-minute video clip in which the testing vehicle is running on the highway as the target video to be recognized in, and due to that almost every perspective of the running vehicle exists in the video.

Figure 7 shows that the number of features in the feature database varies as a function of the number of training frame images containing the target video vehicle in the surveillance video with RVO-SIFT and classic RGB-SIFT, respectively. With RVO-SIFT, it can be seen that the number of feature keypoints does increase with the increasing of the number of training frame images. However, after the completeness of the feature database having increased to a considerable degree, the contribution of frame images reduces relatively. Meanwhile, with classic RGB-SIFT, it only extracts features in the specified region which is shown in Figure 1(a), and the size of the feature database keeps invariable.

In order to show how much the feature number affects the outcome of the recognition, the process to recognize the target vehicle in the target video is performed. Figure 8 shows the ratio of the correctly recognized vehicles in frames as a function of the number of feature keypoints in feature databases. We can read from the graph that the ratio of correctly recognized objects increases obviously with the

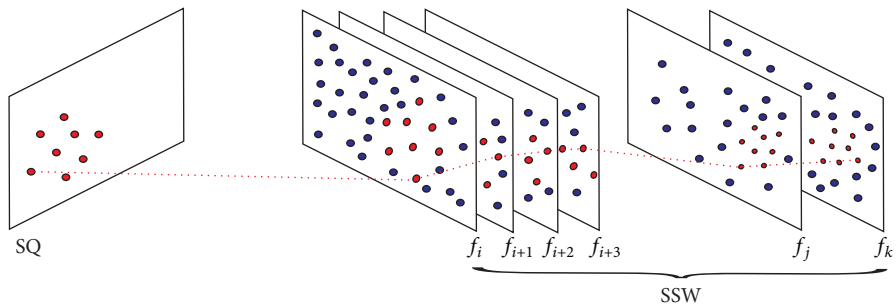

FIGURE 6: Red dots represent sharing feature keypoints that have matched between an existing model of the target object and similar frame image views in SSW, and blue dots indicate other candidate feature keypoints in each frame image view. The dotted line shows the virtual trajectory of one of the matched feature keypoints.

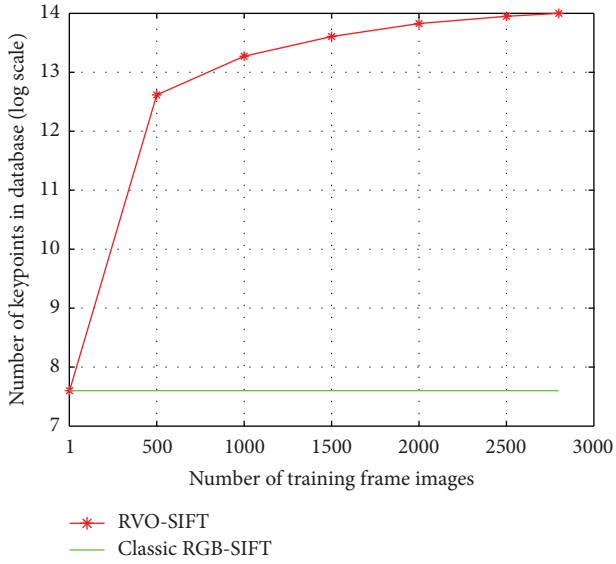

FIGURE 7: The red line shows the number of keypoints in the feature database (log scale) as a function of the number of training frame images containing the running vehicle with RVO-SIFT and classic RGB-SIFT, respectively.

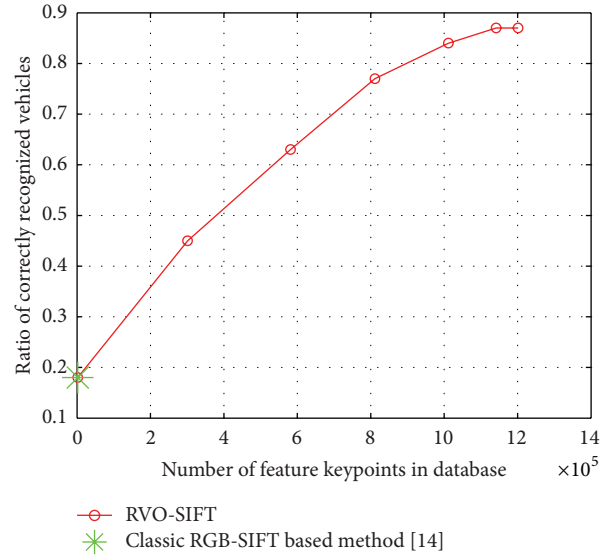

FIGURE 8: The red line and the green star show the percent of correctly recognized vehicles as a function of database size (log scale) with RVO-SIFT and classic RGB-SIFT based method [14].

increasing number of feature keypoints in database and, meanwhile, the growth rate slows down when the completeness of feature database reaches a certain degree. Some view images recognized correctly are shown in the top line of Figure 10.

Of course, accompanied by the feature database updating process, the recognition process of the RVO-SIFT consumes much more computation time. However, its average delay time is affordable. The experimental results are shown in Figure 9.

In order to show the generality of the RVO-SIFT, the recognition is performed additionally in the micromovie "The New Year, The Same Days" in which the face of the wife character is recognized and a trailer video for a blue and white porcelain, with which a beautiful chinaware is to be recognized in; the results are shown in the two lines below in Figure 10.

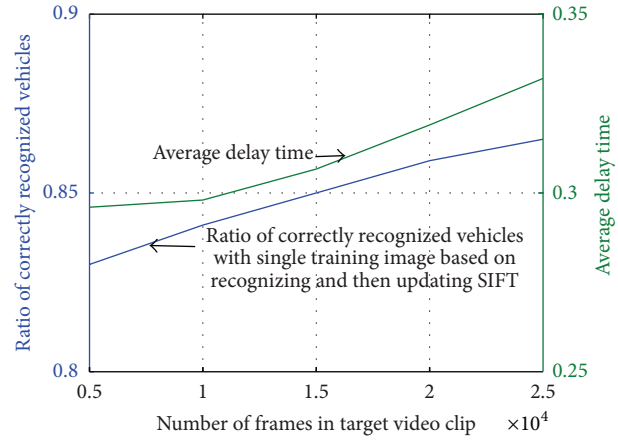

FIGURE 9: The top line in the graph shows the average delay time as a function of the number of frames in the video clip to be recognized. The bottom line shows the percent of correctly recognized vehicles in frames.

Figure 10: The experimental results show that using the selected areas in bounding boxes in the left images as the training image, the other objects on the right are recognized correctly in corresponding video by RVO-SIFT eventually.

As shown in Figure 10, the RVO-SIFT performs well with real-world videos. Even human faces without exaggerated facial expression changes can be recognized correctly with relatively high rate, just as shown in the middle line. Furthermore, experimental results also show that the RVO-SIFT even can tolerate local camouflages, which is a basic but wonderful ability of human beings, due to the fact that the features of the camouflaged region would be added into the feature database by recognizing and then updating procedure gradually and then they play their roles in recognition subsequently.

6. Conclusion and Future Work

The RVO-SIFT, in which the novel recognizing and then updating mechanism is adopted, is particularly not only a wonderful rigid video object recognizer but also a wonderful feature automatic extractor for rigid video objects. It mixes processes of the object recognizing and feature studying together, just like what human being does in recognition process. It can improve greatly the completeness of the feature database of the target video object automatically and in turn increases drastically the ratio of correctly recognized objects consequently, at the expense of the more affordable millisecond level computation time. In addition to rigid video object recognition, its other potential applications include rigid video motion tracking and segmentation and any others that require feature extraction of the rigid targets in videos or image sequences.

However, RVO-SIFT is based on rigid video object theoretically and experimentally in this paper, so one of the directions for further research is to try to apply it to semirigid video objects, such as video face recognition with exaggerated facial expression changes.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported by the National Nature Science Foundation of China under Grant no. 61163066.

References

[1] S. Gould, J. Arfvidsson, A. Kaehler et al., "Peripheral-foveal vision for real-time object recognition and tracking in video," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI '07)*, vol. 7, pp. 2115–2121, January 2007.

[2] P. Chaturvedi, A. S. Rajput, and A. Jain, "Video object tracking based on automatic background segmentation and updating using RBF neural network," *International Journal of Advanced Computer Research*, vol. 3, no. 2, p. 866, 2013.

[3] D. G. Lowe, "Local feature view clustering for 3D object recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. I-682–I-688, December 2001.

[4] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence*, vol. 31, no. 3, pp. 355–395, 1987.

[5] T. Gevers and A. W. M. Smeulders, "Color-based object recognition," *Pattern Recognition*, vol. 32, no. 3, pp. 453–464, 1999.

[6] Y. Zhong and A. K. Jain, "Object localization using color, texture and shape," *Pattern Recognition*, vol. 33, no. 4, pp. 671–684, 2000.

[7] C. Y. Chung and H. H. Chen, "Video object extraction via MRF-based contour tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 1, pp. 149–155, 2010.

[8] J. Yu and F. L. Zhang, "Distinguishing moving objects from traffic video by the dynamic background skeleton based model," in *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS '13)*, vol. 1, pp. 271–275, IEEE, 2013.

[9] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," in *Matters of Intelligence*, pp. 115–141, Springer, Amsterdam, The Netherlands, 1987.

[10] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.

[11] Y. Hu, X. Xie, W. Y. Ma et al., "Salient region detection using weighted feature maps based on the human visual attention model," in *Advances in Multimedia Information Processing—PCM 2004*, vol. 3332 of *Lecture Notes in Computer Science*, pp. 993–1000, Springer, Berlin, Germany, 2004.

[12] D. Stavens and S. Thrun, "Unsupervised learning of invariant features using video," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1649–1656, IEEE, San Francisco, Calif, USA, June 2010.

[13] A. Makadia, "Feature tracking for wide-baseline image retrieval," in *Computer Vision—ECCV 2010*, vol. 6315, pp. 310–323, Springer, Berlin, Germany, 2010.

[14] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 2, pp. 1150–1157, September 1999.

[15] K. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[16] C. Kolb, D. Mitchell, and P. Hanrahan, "Realistic camera model for computer graphics," in *Proceedings of the 22nd Annual ACM Conference on Computer Graphics and Interactive Techniques*, pp. 317–324, August 1995.

[17] L. Quan, "Self-calibration of an affine camera from multiple views," *International Journal of Computer Vision*, vol. 19, no. 1, pp. 93–105, 1996.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[19] J. S. Beis and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1000–1006, June 1997.

[20] J. Illingworth and J. Kittler, "A survey of the hough transform," *Computer Vision, Graphics and Image Processing*, vol. 44, no. 1, pp. 87–116, 1988.

[21] G. L. Foresti, "Object recognition and tracking for remote video surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 7, pp. 1045–1062, 1999.