Robust Object Tracking with a Case-base Updating Strategy

Wenhui Liao, Yan Tong, Zhiwei Zhu, and Qiang Ji
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA

Abstract

The paper describes a simple but effective framework for visual object tracking in video sequences. The main contribution of this work lies in the introduction of a case-based reasoning (CBR) method to maintain an accurate target model automatically and efficiently under significant appearance changes without drifting away. Specifically, an automatic case-base maintenance algorithm is proposed to dynamically update the case base, manage the case base to be competent and representative, and to maintain the case base in a reasonable size for real-time performance. Furthermore, the method can provide an accurate confidence measurement for each tracked object so that the tracking failures can be identified in time. Under the framework, a real-time face tracker is built to track human faces robustly under various face orientations, significant facial expressions, and illumination changes.

1 Introduction

Object tracking in video sequences is important in applications such as video compression, surveillance, human-computer interaction (HCI), content-based video indexing, and others. However, robustly and accurately tracking objects remains challenging due to the various appearance changes caused by illumination changes, partial or full occlusions, pose variations, non-rigid shape deformations, camera view changes, and so on. Therefore, a key task in object tracking research is to design a flexible and accurate model that can automatically cope with appearance changes.

The template-matching based methods have been extensively used for object tracking by searching the image region that is most similar to an object template. In general, the template-matching based methods could be classified into three groups based on how the template is constructed. The first group extracts the first frame in a video sequence as the template [Hager and Belhumeur, 1998; Li and Chellappa, 2000; Matthews *et al.*, 2004]. However, when new parts of the object come into view or the appearance of the object varies significantly during tracking, the template doesn't fit anymore thus the drifting issue becomes serious. In the second group, the preceding frame is extracted as the template [Black and Yacoob, 1997; Papanikolopoulos *et al.*, 1993; Ho *et al.*, 2004]. In this case, the tracker can inevitably use a wrong template due to partial occlusions or the accumulated errors from previous tracking steps [Nguyen *et al.*, 2001]. The third group online updates the template from a number of previous frames or several key frames [Morency *et al.*, 2003; Vacchetti *et al.*, 2004; Lim *et al.*, 2004; Lee and Kriegman, 2005; Jepson *et al.*, 2003]. It could overcome some disadvantages of the previous two groups, however, the issue is how to combine the multiple tracked objects appropriately to generate the new template. Furthermore, most of them ignore the potential errors associated with each tracked object. Therefore once a tracked object view contains an error, it may be integrated into the updated template, and errors accumulate throughout the tracking. Consequently, most of the existing tracking methods suffer from the well-known drifting issue, incapable of assessing the tracking failures or recovering from the tracking error.

In this paper, based on our previous work in [Zhu *et al.*, 2006], a robust object tracking framework based on case-based reasoning (CBR) is proposed to automatically provide an accurate 2D tracking model at each image frame. Specifically, the 2D tracking model in each frame can be online adapted dynamically by combining the object extracted from current frame and the most similar image case retrieved from a case base. Unlike our previous work [Zhu *et al.*, 2006], where the case base is fixed, the case base is constructed and automatically maintained to be competent, representative, and with a reasonable size to make sure that an image case that is most similar to the tracked object can be retrieved quickly for each frame. Under the framework, the appearance changes of the tracked object can be adapted dynamically via an adaption mechanism of CBR. As an result, an accurate 2D tracking model can be maintained online and thus the drifting issue that plagues most of the tracking techniques can be well handled. Furthermore, under the CBR paradigm, since the tracked view is always adapted from its most similar case in the case base, an accurate similarity measurement can be easily obtained to characterize the confidence level of the tracked region, so that the failure situations can be detected in time.

Based on the proposed framework, we build a real-time face tracking system to track faces in video sequences. Such

a real-time face tracker can track faces robustly under significant appearance changes in illumination, scale, facial expression, occlusion, and head movement. Experiments also demonstrate that the proposed case-base maintenance algorithm is able to automatically update the case base to include enough representative face views and delete redundant ones, so that to maintain the case base in a reasonable size with a good tracking performance. Compared to the existing techniques, our proposed technique has the following advantages: (1) handle the drifting issue well; (2) no need of a 3D model; (3) capable of tracking any object; (4) being able to assess the tracking failures with a confidence level; and (5) automatically update candidate templates based on a case base.

2 The Mathematical Framework

2.1 The 2D Object Tracking Model

Assume that an object O is moving in front of a video camera. At time t, the object is captured as an image view $I(X_t)$ at position X_t in the image frame I_t. Then the task of 2D visual tracking is to search the image view $I(X_t)$ of the object O in each image frame I_t by using an object model I_M^t. Let $I(X_t')$ be the located image view, then the tracking error ΔI_0^t can be represented as the difference between the true image view $I(X_t)$ and the located image view $I(X_t')$ of the object: $\Delta I_0^t = I(X_t) - I(X_t')$. Given the object model I_M^t, if we assume that its most similar image view can be successfully located in the image frame I_t, then apparently the tracking error ΔI_0^t is mostly caused by the inaccuracy of the utilized object model I_M^t. Therefore, a key issue for a successful visual tracking technique is to obtain an accurate 2D tracking model I_M^t of the object at each image frame I_t.

We thus propose a case-based reasoning algorithm to maintain an accurate tracking model for each image frame so that the tracking error ΔI_0^t can be minimized during tracking. Figure 1 illustrates the major steps of the proposed algorithm.

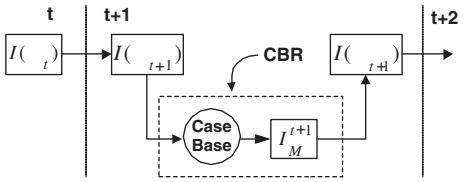


Figure 1: The diagram of the proposed algorithm

The first step is to locate the object in the image frame I_{t+1} using the tracked 2D view $I(X_t)$ as the initial object model. Let $I(X_{t+1}')$ be the located object view in the image frame I_{t+1}. Due to image variations, $I(X_t)$ may not be an accurate model for the image frame at $t+1$ so that the located image $I(X_{t+1}')$ is not accurate enough to reflect the current object. Even though, $I(X_{t+1}')$ usually contains information about the object appearance at the current time $t+1$ partially or completely. Thus, the located image view $I(X_{t+1}')$ represents an important information source that can be utilized to infer the true view of the object in the image frame I_{t+1}.

Then, in the second step, $I(X_{t+1}')$ is used to find a new object model $I_{M'}^{t+1}$ by searching a case base. The case base contains a set of representative 2D views of the object, where any unseen image view can be adapted from them with some image adaption strategy. Therefore the obtained object model $I_{M'}^{t+1}$ is able to adapt to the appearance changes in the image frame I_{t+1}, and thus also includes essential information to infer the true object view.

Finally, $I_{M'}^{t+1}$ and $I(X_{t+1}')$ are combined together to get the final image view $I(X_{t+1})$. Since both $I_{M'}^{t+1}$ and $I(X_{t+1}')$ contain important information about the object view and complementary to each other, a more accurate 2D tracking model can be obtained at each image frame. Therefore, the drifting issue accompanied with most of the visual tracking techniques can be alleviated.

2.2 The CBR Visual Tracking Algorithm

The above proposed solution can be well-interpreted and implemented in a case-based paradigm. Case-based reasoning (CBR) is a problem solving and learning approach that has grown into a field of widespread interests in both academics and industry since the 1990's [Aamodt and Plaza, 1994]. From a CBR perspective, the problem of visual object tracking in a video sequence can be solved by retrieving and adapting the previously seen views of the object to a new view of the object at each image frame. In the following, we demonstrate the CBR-based tracking algorithm with a face tracking system. More detail can be referred from our previous work at [Zhu *et al.*, 2006].

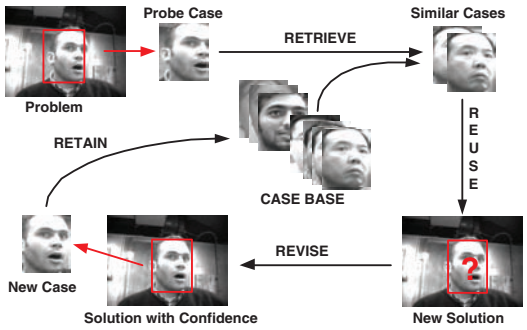


Figure 2: The CBR cycle of the face tracking system.

Figure 2 illustrates a general CBR cycle for the built face tracking system, which consists of four processes, *Retrieve*, *Reuse*, *Revise*, and *Retain*.

Case Retrieve: The Retrieval process searches the most similar face images from a case base composed of the collected representative face images with the located view $I(X_{t+1}')$. The magnitudes of a set of multi-scale and multi-orientation Gabor wavelets as in [Lee, 1996] are used as the feature representation of the image appearance. Thus the object searching is to find a case $I_{M'}^{t+1}$ that is most similar to $I(X_{t+1}')$ in terms of the Gabor response vectors.

Case Reuse: The reuse process reuses the information and knowledge in the retrieved face image to refine the previously located face image by adapting to the face appearance changes. It consists of two steps. At the first step, the selected image view $I_{M'}^{t+1}$ is utilized as a tracking model to perform a search starting at X_{t+1}' in the image frame I_{t+1} for a most similar image view, called X_{t+1}'' . Subsequently, the second step is to combine the tracked image views $I(X_{t+1}')$ and

$I(X''_{t+1})$ to obtain the final image view $I(X_{t+1})$ as follows:

$$X_{t+1} = \frac{S'_{t+1}}{S'_{t+1} + S''_{t+1}} X'_{t+1} + \frac{S''_{t+1}}{S'_{t+1} + S''_{t+1}} X''_{t+1} \quad (1)$$

where S'_{t+1} is the cosine similarity measure between the Gabor response vectors of $I(X_t)$ and $I(X'_{t+1})$, and S''_{t+1} is the cosine similarity measure between the Gabor response vectors of $I_{M'}^{t+1}$ and $I(X^{''}_{t+1})$. Intuitively, it can be seen as a minimization of the sum of two errors, the error between the target view in the current frame and the tracked view in the previous frame, as well as the error between the target view in the current frame and the selected similar case view from the case base. In other words, the tracked target view possesses one important property: it must be similar to the tracked view in the previous frame as well as the selected case view in the case base. When the tracked view satisfies this property, the drifting issue can be well handled and thus the tracking has a high chance to succeed.

Case Revision: The revise process evaluates the tracking result with a confidence measure. Once the final image view $I(X_{t+1})$ is obtained, another search will be conducted in the case base to find the most similar case. A similarity score S_{t+1} is derived after the searching is done. In practice, if the similarity score is high, tracking is usually successful; otherwise, it may fail. Therefore, the derived similarity score S_{t+1} is utilized as a confidence measure to characterize the final image view $I(X_{t+1})$. The system automatically reports the confidence measure and stores those image views with low confidence level into a temporary database. Such a temporary database will be used for case-base maintenance.

Case Retain: The retaining process enriches the case base by adding new representative image views. To retain a new case, the image views in the temporary database have to be reviewed periodically, so that only the useful image views are selected and added into the case base. Using this process, it is believed that the case base can include more and more representative 2D views of the objects. Such a process is one important task in case-base maintenance, which will be presented in the next section.

3 Case-Base Construction and Maintenance

For the proposed case-base reasoning system, it is crucial to construct a good case base and maintain the case base automatically and systematically in order to keep or improve the tracking performance in response to object changes in the video sequences. For example, as time goes by, the case-base size may become too large, so that it becomes slow to retrieval a similar case. In addition, some cases may become obsolete so that they need to be replaced by more powerful cases. Basically, case-base maintenance (CBM) can be referred as the task of adding, deleting, and updating cases, indexes and other knowledge in a case base in order to guarantee the ongoing performance of a CBR system [Zhu and Yang, 1999].

Before we give the detailed algorithms for CBM, we first introduce how to construct the initial case base. The initial case base is generated through a training process combined with human intervention before the tracking system is formally used. Specifically, around 150 face images coming from subjects of different ethnicity, gender, age, and face poses are collected. The similarity scores between any of two images have to be small otherwise one of them is regarded as redundant and is thus deleted. Overall, the cases in the initial case base are representative, accurate, and diverse. Each case is regarded as a cluster and itself is called *key case*, marked as I_i^*. During CBM, each existing cluster would be enriched and new cluster would be added.

The proposed CBM algorithm includes two processes: *case-updating* and *case-deleting*. Case-updating is triggered after each tracking task if the tracking successful rate is lower than a predefined threshold, where the successful rate is defined as the percentage of the image frames that have high confidence scores over the whole image frames. Case-deleting is triggered only when the size of the case base is too large so that the tracking speed cannot reach the real-time requirement. In the following, we present the two processes respectively.

3.1 Case-base Updating

The case-updating procedure consists of several steps. First, all the images with low confidence scores during tracking are put into a temporary database. Second, an off-line multi-view face detector [Wang and Ji, 2005] is applied to identify all the face images. The non-face images are then deleted. The third step, also the key step, is to update the case base with the remaining images in the temporary database since the faces in these images are not covered by the current case base. However, we cannot simply add all the identified face images into the case base due to the limitation of the case-base size. Also, it is not necessary to put all those images into the case base because most of the images are very similar to each other. By adding some of them into the case base is usually enough to track the remaining images successfully.

Let D_t be the temporary database with only face images, D_o be the current case base. The goal of case-updating is to find a case set D^* from the union of D_t and D_o that achieves an optimal tradeoff between the tracking performance and the overall size of the case base.

$$D^* = \arg\max_{D_i} \{ f(D_i, (D_t \cup D_o) \setminus D_i) - g(|D_i|)|D_i| \} \quad (2)$$

where D_i is any case subset in $D_t \cup D_o$, $f(D_i, D_j)$ is the tracking performance function, which is the tracking successful rate of using D_i as the case base to track the images in D_j, and $g(x)$ is the penalty function, which is defined as $\frac{c}{1+e^{-a(x-b)}}$. Basically $g(x)$ is a modified sigmoid function. When the size of D_i is very small or very large, the penalty increases little as the size of the case base increases; otherwise, the penalty increases more obviously as the size of the case base increases.

However, it is an NP-hard problem to find such an optimal case base. In practice, we use a greedy approach as shown in Table 1 to find an optimal or near-optimal solution.

The CaseUpdating function first finds the most similar image I^* among the key cases to the face images obtained from the temporary database. If the largest similarity score is still not large enough ($< T_s$), it means that the tracking failure is caused by a new face that is not covered in the current case base. Thus the CaseAdding function is called to select a set of

images from the temporary database until the tracking performance cannot be improved by adding more cases with Equation 2. These selected images are added into the case base as a new cluster and the first selected image during CaseAdding is marked as the key case. And the unselected images are stored in the testing pool associated with the new cluster.

On the other hand, if the largest similarity score is large enough ($> T_s$), it means the current case base may already have the similar faces, however, the new faces may be different because of the variation of pose, size, facial expressions, or other factors. In that case, it is not necessary to add a new cluster. Instead, the algorithm moves the images in the same cluster of I^* in the case base as well as the images in the testing pool associated with I^* into the temporary database. And then, the CaseAdding function is called to select a new cluster from the updated temporary database. Such a new cluster replaces the previous cluster and a new key case is also generated for the updated cluster. Figure 3 illustrates how a cluster is updated in the case base.

CaseUpdating(D_o, D_t, T_s)

$max \leftarrow -\inf$;
for each $I_i \in D_t$
  for each $I_j^* \in D_o$
    if Similarity$(I_i, I_j^*) > max$
      $max \leftarrow$ Similarity(I_i, I_j^*);
      $I^* \leftarrow I_j^*$
if $max < T_s$
  CaseAdding(D_o, D_t);
else
  $D_p \leftarrow$ the image set in the testing pool associated with I^*;
  $D_i \leftarrow$ the images in the same cluster of I^* in D_o;
  $D_o' \leftarrow D_o \setminus D_i$;
  $D_t' \leftarrow D_t \cup D_p \cup D_i$;
  CaseAdding(D_o', D_t');

CaseAdding(D_o, D_t)

$R_0 = f(D_o, D_t) - g(|D_o|)|D_o|$;
$max \leftarrow 0$;
while $max \geq 0$ & $D_t \neq \emptyset$
  $max \leftarrow 0$;
  for each $I_i \in D_t$
    $D_o \leftarrow D_o \cup \{I_i\}$;
    $R(I_i) = f(D_o, D_t \setminus I_i) - g(|D_o|)|D_o|$;
    if $R(I_i) - R_0 > max$
      $max \leftarrow R(I_i) - R_0$;
      $I^* \leftarrow I_i$;
    $D_o \leftarrow D_o \setminus I_i$;
  $R_0 \leftarrow R(I^*)$;
  $D_o \leftarrow D_o \cup \{I^*\}$;
  $D_t \leftarrow D_t \setminus I^*$;

Table 1: Algorithms of Case-updating

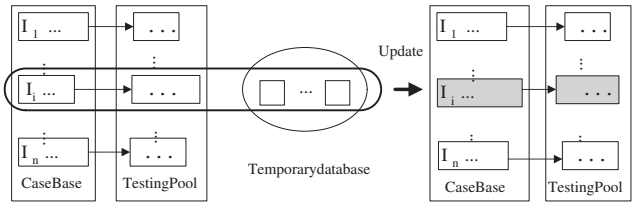


Figure 3: An illustration of how a cluster in the case base is updated. Basically, the testing pool includes the images that the system fails to track in the past and the temporary database includes the images that the system fails to track recently. By considering the images from both the past and current, the updated cluster tends to achieve "global optimal" instead of "local optimal".

Obviously, the structure of the initial case base changes through the updating. In the beginning, each cluster only includes one case and such a case is the key case. And the testing pool is empty. As the updating goes on, more clusters will be added. Also, some clusters will be enriched with more cases and the original key case may be replaced with another case or remain the same. In addition, each cluster is associated with a group of images in the testing pool, which are the images that can be handled by this cluster.

With such an updating procedure, the new representative images that are not covered by the case base can be added automatically and timely. Also, the less representative cases in the case base can be replaced with more powerful cases. Furthermore, since the updating procedure is guided by the performance of the tracking system, it guarantees that the newly added cases always improve the tracking system.

3.2 Case-base Deleting

Although the case-updating procedure performs both adding and deleting actions to the case base, it is still possible that the overall size of the case base may become too large so that the tracking speed is affected and slows down. If that happens, it is necessary to delete some cases.

The case base includes two types of cases: key case, and non-key case, as defined in the updating procedure. We associate each case with a utility value. It is defined as the frequency that the case is selected as the most similar case during tracking. For each newly added case, the utility is set equal to the current lowest utility value in the case base (or 1 if the current minimal value is 0). Obviously, the higher the utility is, the more desirable to keep the case in the case base. The deletion procedure starts from the non-key case that has the lowest utility value. For the cases with the same utility value, the algorithm deletes them evenly from each cluster, to avoid that one cluster is almost empty while another one is full of cases. In fact, since the searching algorithm and computational methods are efficient in our system, the system can achieve a real-time performance even when the size of the case base is around 750. Therefore the deletion procedure rarely happens.

Overall, with the proposed automatic CBM algorithms, the case base tends to include all the representative face views and delete redundant ones so that to maintain the case base in a reasonable size with a good tracking performance.

4 Experimental Results

Based on the proposed CBR visual tracking framework, a real-time face tracking system is built. The whole system runs at 26 frame per second (fps) in a machine with a 2.8GHZ CPU. When a person appears in the view of the camera, the person's face is automatically localized via the proposed frontal face detector [Wang and Ji, 2005] and tracked subsequently by our proposed face tracker. The initial case base includes 164 face images, which are automatically updated as more and more video sequences are tested. To test the performance of the built face tracking system, a set of face video sequences with several new subjects are collected. We first demonstrate the performance of the proposed case-base maintenance (CBM) algorithm, and then demonstrate the performance of the face tracker.

4.1 Case-base Maintenance

To test the performance of the proposed case-base maintenance algorithm, 20 testing sequences from 13 novel subjects are collected. These subjects are not included in the initial case base, and with different races and gender. Each of the testing sequence consists of 600 frames.

Figure 4 demonstrates the overall performance of the system with and without the CBM algorithm. As shown in the figure, without the CBM algorithm, the successful rates for most testing sequences are below 0.6 because the original case base cannot cover most of the new face images. With the CBM algorithm, the case base is enriched with the new faces gradually thus the system performance is continually improved. After around 10 testing sequences, the average successful rate is higher than 0.95, and the size of the case base converges and becomes stable, which implies that the current case base is competent for our current subject pool.

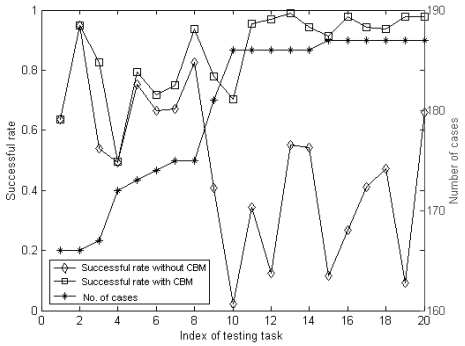


Figure 4: Comparison of the face tracking performance with CBM and without CBM: the curve marked with stars represents the number of cases; the curve marked with diamonds represents the tracking successful rate without CBM; the curve marked with squares represents the tracking successful rate with CBM.

Figure 5(I) gives an example of how the CBM works in a specific video sequence. The two curves represent the confidence measurements with and without CBM respectively. In most sequences, the faces can be tracked successfully, which can be reflected by the fact that most of the confidence scores are higher than 0.5. Figure 5(IV) lists the three most-frequently retrieved cases from the case base during the tracking. It shows that although the subjects in the three cases are different from the subject in the video sequence, they can still help to track the faces in most time, which demonstrates the robustness of the CBR-based method of our system. However, for the frames around a, b, c, d, and e, the tracking fails because the cases in the case base are not similar enough. Figure 5(II) lists the five images corresponding to a, b, c, d, and e in the video sequence, and Figure 5(III) lists the most similar cases that are retrieved from the case base. After adding the images in Figure 5(II) to the case base, all the faces are tracked well. Since the newly added cases are very different in facial appearances or face poses from the cases in the case base, they increase the representativeness of the case base and thus can improve the system performance.

4.2 Tracking Performance

We first demonstrate the self-correction (drifting-elimination) capability of the proposed tracking scheme. Two other popular tracking techniques are implemented to compare to our method. The first technique is the traditional two-frame-based tracking method [Matthews *et al.*, 2004], which utilizes the previously tracked face image to update the tracking model dynamically. The second one is the tracking technique with the incremental subspace learning [Lim *et al.*, 2004], which dynamically updates the object model by incremental learning its eigen-basis from a set of previously tracked faces.

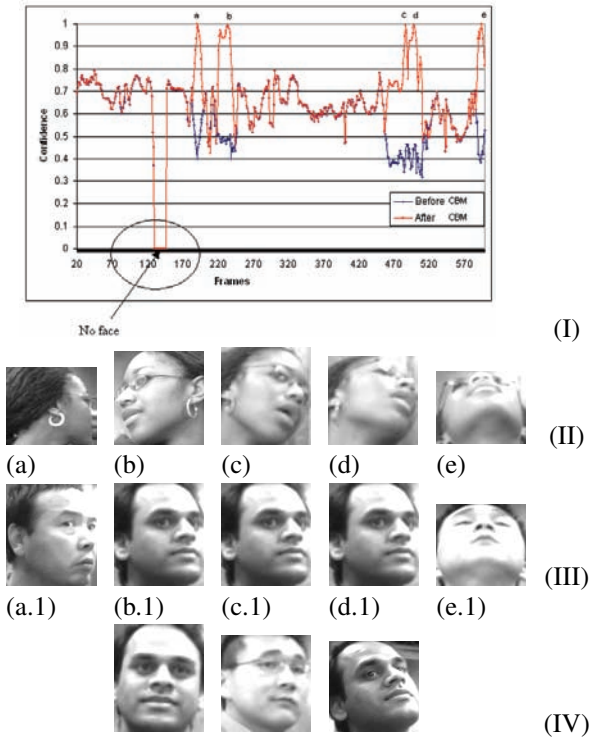
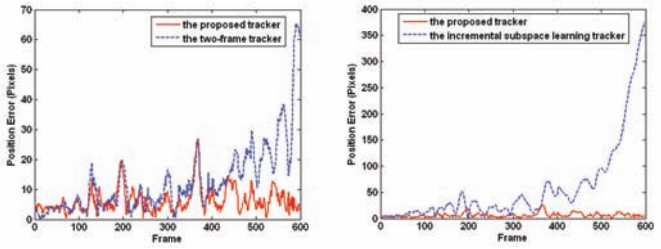


Figure 5: (I) Comparison of the estimated confidence measurements before and after case updating; (II) the corresponding images in the video sequence; (III) the most similar cases retrieved from the case base before updating; (IV) the most-frequently retrieved cases from the case base during tracking.



(a) (b)

Figure 6: Comparisons of the tracked face position error between the proposed tracker and (a) the two-frame tracker; (b) the incremental subspace learning tracker.

Figure 6 compares the performance of the three face trackers for a face sequence under significant head movements and facial expressions. From Figure 6 (a), it is obvious that the

tracking error of the two-frame tracker accumulates as the tracking continues. Eventually it drifts away from the face and tracks the wrong object. Figure 6 (b) shows that the tracking technique with the incremental subspace learning also drifts and tracks the wrong object eventually. One possible reason is that because the tracked face view contains errors or non-face image pixels, they are learned and accumulated in the model throughout the video sequence and eventually lead to drifting. However, our proposed tracking method can eliminate the tracking error in each frame gradually during tracking, while still tracking the face robustly.

In addition, when the two-frame tracker and the subspace learning tracker fail, they cannot automatically detect failures and still continue tracking. On the other hand, our proposed method can indicate the failures through confidence measurements. In practice, we found that when the confidence score is higher than 0.5, the tracking is usually successful.

Furthermore, the proposed method can perform well with different individuals, under significant illumination changes, different facial expressions, and various face orientations. As shown in Figure 7, the appearance of the persons' faces changes drastically due to the illumination changes, face orientations, facial expressions as well as partial occlusions, which makes the tracking task extremely difficult. However, the proposed method can still track the faces well because of its capability of maintaining a good object model.

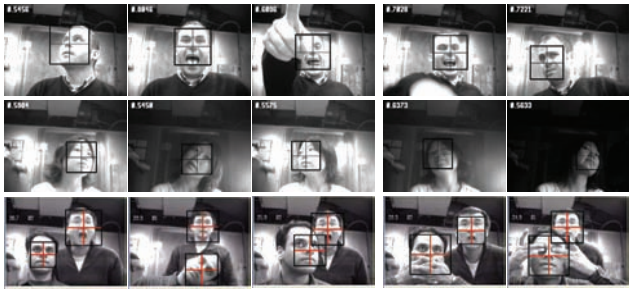


Figure 7: The face tracking results with significant facial expression changes, head movements, occlusions, illumination changes, and multi-face presenting.

5 Conclusions

In this paper, a simple but effective visual tracking framework based on CBR paradigm with confidence level is introduced to track faces in video sequences. Under the proposed framework, different faces can be tracked robustly under significant appearance changes without drifting via the assistance of the most similar case retrieved from the case base. The case base is automatically maintained to be competent, representative, and with a reasonable size for real-time retrieval of useful cases. In addition, a confidence measurement can be derived accurately for each tracked face view so that the failures can be assessed successfully. It provides a nice framework to handle the drifting issue that has plagued the face tracking community for a long time. All of these merits of the proposed framework are demonstrated through a real-time face tracking system that can track human faces under various face orientations, significant facial expressions, and illumination changes. Such a framework can be easily generalized to track other objects by only updating its case base.

Acknowledgments

This project was supported in part by the Air Force of Scientific Research (AFOSR) under the grant number F49620-03-1-0160 and in part by the Defense Advanced Research Project Agency (DARPA) under the grant number N00014-03-1-1003.

References

[Aamodt and Plaza, 1994] A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, 1994.

[Black and Yacoob, 1997] M. Black and Y. Yacoob. Recognizing facial expression in image sequences using local parameterized models of image motion. *IJCV*, 25(1):23–48, 1997.

[Hager and Belhumeur, 1998] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, October 1998.

[Ho *et al.*, 2004] J. Ho, K. Lee, M. Yang, and D. Kriegman. Visual tracking using learned linear subspaces. In *CVPR*, 2004.

[Jepson *et al.*, 2003] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi. Robust online appearance models for visual tracking. *PAMI*, 25(10):1296–1311, 2003.

[Lee and Kriegman, 2005] K. Lee and D. Kriegman. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In *CVPR*, 2005.

[Lee, 1996] T. Lee. Image representation using 2d gabor wavelets. *PAMI*, 18(10):959–971, October 1996.

[Li and Chellappa, 2000] B. Li and R. Chellappa. Simultaneous tracking and verification via sequential posterior estimation. *CVPR*, 2:110–117, 2000.

[Lim *et al.*, 2004] J. Lim, D. Ross, R. Lin, and M. Yang. Incremental learning for visual tracking. In *NIPS04*, 2004.

[Matthews *et al.*, 2004] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *PAMI*, 26(6):810–815, 2004.

[Morency *et al.*, 2003] L.P. Morency, A. Rahimi, and T. Darrell. Adaptive view-based appearance model. In *CVPR*, 2003.

[Nguyen *et al.*, 2001] H.T. Nguyen, M. Worring, and R. van den Boomgaard. Occlusion robust adaptive template tracking. *ICCV*, pages 678–683, 2001.

[Papanikolopoulos *et al.*, 1993] N. Papanikolopoulos, P. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Trans. on Robotics and Automation*, 9:14–35, 1993.

[Vacchetti *et al.*, 2004] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3d tracking using online and offline information. *PAMI*, 26(10):1391–1391, 2004.

[Wang and Ji, 2005] P. Wang and Q. Ji. Learning discriminant features for multi-view face and eye detection. In *CVPR*, 2005.

[Zhu and Yang, 1999] Jun Zhu and Qiang Yang. Remembering to add: Competence-preserving caseaddition policies for case base maintenance. *IJCAI*, pages 234–241, July 1999.

[Zhu *et al.*, 2006] Zhiwei Zhu, Wenhui Liao, and Qiang Ji. Robust visual tracking using case-based reasoning with confidence. *CVPR*, pages 806–813, 2006.