Face Tracking via Affine Invariant Features in Video

Sofia Karygianni
University of Toronto
sofiakar@cs.toronto.edu

Kiriakos N.Kutulakos
University of Toronto
kyros@cs.toronto.edu

Abstract

This thesis presents an algorithm for face tracking in video sequences. We investigate the application of affine-invariant, local features for face tracking under random poses and expressions.

In order to capture as much as possible of the facial variability, a combination of region detectors is used to extract the various facial points of interest. Pairwise matching of SIFT descriptors for those regions is used to identify possible similarity transformations between consecutive frames. If the matching process does not provide satisfying candidates, various translation parameters are used to determine the set of possible candidates.

The similariy transformations are finally ranked according to their compatibility with the color and orientation descriptors of the previous template. The candidate with the best score is chosen as the new template.

We have applied the above method in a small data set of video sequences and found it to work well under various settings and conditions.

1. Introduction

Face tracking is a hard but useful application. Systems in various areas (such as human-computer interaction, video surveillance, robotics, biometric security etc.) often require a reliable face tracking algorithm. The non-rigid nature of the human face and its high variability are the main sources of diversity and complexity. The same face may appear completely different even under the same viewpoint due to changes in expression. On the other hand, faces of different people can be similar, fairly different, or contrasting. Face tracking can be generic or person-specific but it is a hard problem in either case.

One of the main objectives of this work is to study whether affine invariant features can be used effectively for face tracking. These features have be used for describing objects and identifying connections between their instances. Currently, multiple schemes exist effected by the widespread applicability of the features. The concept is simple and convenient: the detectors discover the (affine invariant) regions of interest and the descriptors provide a suitable representation for them. Matches established between such features across different scenes may provide useful information about the relative movement or the existence of an object. However, the premise of invariance is limited in real situations; and as a result the features cannot be used for every task indiscriminantly.

In this work, we combine affine invariant features with orientation and color descriptors in order to create a more desriptive and robust scheme. Training is not required (the method uses features from successive frames without building an explicit model). Furthermore, the procedure is the same for every possible face and can thus be applied to any person without modification. In short, our system is a person-independent face tracker that does not require training.

The rest of the paper is organized as follows. A brief overview of related work is presented in Section 2. A detailed description of the features is provided in Section 3 while the face tracking algorithm is introduced in Section 4. The performance of the algorithm is discussed in Section 5; and, finally, conclusions and future work are given in Section 6.

2. Related Work

Tracking is a very well-known application and various tracking schemes have been proposed [8, 1, 11, 7]. The key property of such schemes is their ability to adapt to appearance changes of the target and to distinguish correctly the tracking region from the background.

In order to achieve this goal, the authors in [8] suggest the use of an online appearance model that is composed by three different components. The first one is the stable model which represents the slowly varying properties of the target. The second is the wandering model which adapts in sudden appearance changes, while the third component accounts for data outliers due to noise, occlusion, etc. On the other hand, in [1] tracking is considered to be a classification problem between the object and the background. The classification is performed with an ensemble of weak classifiers which is trained online to classify pixels correctly. Finally, in [11] and [7] tracking is achieved via particle and kernel-based filtering respectively.

In our case, the face model is composed by the invariant features and the color and orientation descriptors. The face instances between consecutive frames are linked with similarity transformations. The algorithm adapts to changes in face appearance by updating its model at each frame. It also succeeds in distinguishing the tracking target from the background by selecting the most representative for the tracking region transformation among a set of proposed candidates.

Besides general tracking algorithms, various models have been suggested specially for face tracking. Active Appearance Models (AAMs) [17, 10] are among the most popular. They are generative parametric models composed by shape and appearance components. AAM methods track faces by fitting their model to the new image (minimizing the distance between the model and image appearance). They can be either person-specific or generic. As shown in [17], generic AAMs have higher effective dimensionality and are thus more demanding and limited. On the other hand, generic models are of greater interest because of their broader applicability and they are more comparable with our model which is generic as well.

In [10], the authors tried to overcome dimensionality issues by constructing clustered models of similar faces. In order to track an unseen face, the AAM of the most similar cluster is chosen to perform the task. A pretrained parametric model based on a mesh over user-defined feature points is used.

On the other hand, our approach does not require training and does not suffer from dimensionality issues because it does not build explicit facial models. The fitting procedure is reduced to comparisons of potential similarity transformations as induced by the matching pairs of features.

Invariant features have been previously used in other applications for establishing correspondences between images (but, to the best of our knowledge, not for face tracking). For example, in [2], SIFT features are used to estimate interframe motion for video stabilization purposes. In [20], such features are used to connect multi-view images of a real world scene or object and build a metric model for augmenting it.

Feature point tracking for connecting faces has also been used in [19]. The goal of that work is automatic naming of characters in video sequences based on face tracks classification with pretrained SVMs . In order to deal with face tracking, the authors use two face detectors – a frontal and a profile one – and the KLT point tracker [18]. Face tracking is then performed by connecting the isolated face detections through the generated point tracks. Instead, our method uses the invariant features and the color and orientation descriptors to connect the face instances between consecutive frames. Frontal and non-frontal views are linked through the slowly varying face template consisting of the above features, without the use of previously trained detection schemes. Moreover, our algorithm connects the face instances with similarity transformations, providing information about both the position and the pose of the face.

3. Features

3.1. Affine invariant detectors & descriptors

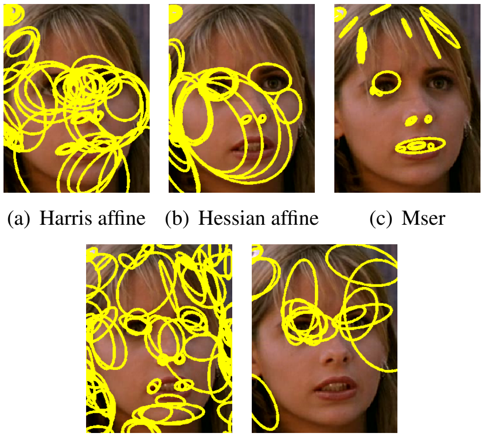
Recent work in computer vision has put a considarable emphasis on detection and description of affine invariant features in 2D images.

There are a number of different detection schemes, each focusing on different image properties [15, 13, 21, 9]. Notable examples include the Harris-Affine detector and the Hessian-Affine detector. The former discovers corner-like points while the latter gives strong responses on blobs and ridges. Their combination can be used to detect distinct facial regions (see an example in figure 1).

Using any single detector constrains the descriptive power of the underlying algorithm per the limitations of the detector. On the other hand, using multiple detectors could increase the descriptive power substantially. In our method we have chosen the following detectors :

∗ Harris-Affine
∗ Hessian-Affine
∗ Maximally Stable External Region(MSER)
∗ Intensity-Based Region(IBR)
∗ Edge-Based Region(EBR)

A detailed description of the properties of those detectors as well as their comparative performance under certain image distortions can be found in [14].



(a) Harris affine (b) Hessian affine (c) Mser

(d) Intensity based (e) Edge based

Figure 1. Results of the detectors on a face. The yellow ellipses indicate the generated regions.

Our method also requires a descriptor for the detected regions. We have chosen the SIFT descriptor [12]. Information about various descriptor schemes can be found in [16]. The authors demonstrate that the SIFT descriptor is a favorable choice because it has good discriminative power and it is robust to image distortions.

3.2. Color and Orientation Descriptors

In our approach, the face representation consists of a number of affine invariant features (which are generated by a set of detectors and the SIFT descriptor) and two other descriptors: one for local orientation and one for color.

Local orientation descriptors have previously been used for face representation with the form of the HOG descriptor [3, 19]. The HOG decriptor measures the histograms of gradient orientations in a coarse grid of spatial locations. It involves a grid of overlapping blocks with each block further divided into a sub-grid of cells. The orientation histograms of the cells are locally normalized and incorporated into the descriptor. In our implementation, we use a 9x9 grid of blocks, a 2x2 sub-grid of cells and a 6-bin orientation histogram as in [19]. The final dimension of the HOG descriptor is then 1,944.

The color descriptor is computed over the same grid of cells. Each cell is characterized by the mean color over its pixels measured in the chosen colorspace. The final color descriptor is composed by the set of all cell colors'. The available colorspaces are many [5]. RGB, HSV, CIE-XYZ and CIE-Lab are some of the most well-known and frequently used.

In our case, we used the RGB colorspace over a 10x10 grid of cells , resulting in a 300 dimensional descriptor.

Examples of the color and orientation descriptors for a face can be seen in figure 2.

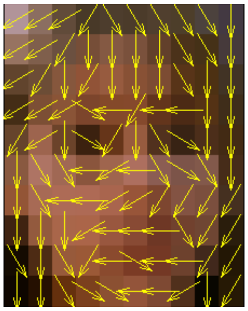


Figure 2. Color and orientation descriptors. The arrows indicate the dominant orientation in each cell.

4. Face Tracking Algorithm

The intention of the algorithm is to track a specific face while moving in a video sequence. A face may undergo several transformations throughout a video. What's more, its non-rigid nature suggests that different facial areas may experience different kinds of distortion between two instances. However, under mild assumptions, and based on the time proximity of consecutive video frames, we can assume that the existing face instances can be related by a similarity transformation which can transform the bounding box of the first instance to the bounding box of the second.

In each frame, the face is contained within its bounding box and represented by the in-box detected invariant features and the corresponding color and HOG descriptors. Such a face representation is called a *template*. Each part of the template performs a different task. Invariant features are easy to compute and match; and are used to link the face instances between consecutive frames. However, the matching of such features may yield unreliable pairs. Therefore, the existence of an evaluation process over the induced – by the matched pairs – results is necessary. The evaluation has to be based on face characteristics that are more stable than the invariant features. The color and HOG descriptors are used for this job. Despite their limited applicability in establishing correspondences between images, their slowly varying appearance within a video sequence makes them suitable for the evaluation task.

The algorithm can be divided into three main steps:

i. Matching
ii. Creation of candidate transformations
iii. Evaluation

The algorithm takes as input a video sequence and the template for the first frame and the output is the estimated templates for the rest of the sequence. A block diagram of the procedure executed for each pair of frames can be found in figure 3. The pseudocode of the method is presented at the end of the paper, in figure 9.

Matching The first step of the algorithm is feature matching. The features of the template from the previous frame are matched against a portion of those in the current frame using Nearest-Neighbor-Matching (NN). The portion of the features used is determined by the location of the previous template. The corner points of the bounding box in the previous frame are expanded by 50% of the corresponding dimension at each direction. The features located inside the resulting region are included in the matching procedure.

At the end, each invariant feature of the previous template is paired with a feature of the new frame in an attempt to determine correspondences between the two face instances. However, the absence of any kind of check or verification over the induced pairs results in a set of matches that may contain a significant portion of wrongly paired features.

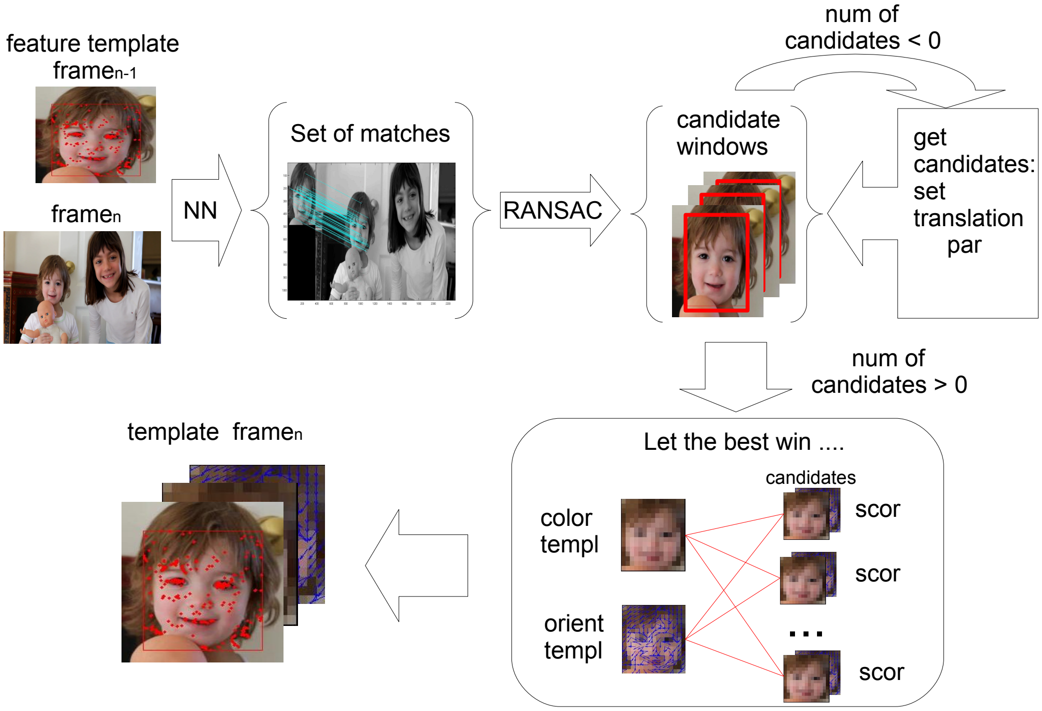


Figure 3. Block diagram of the tracking procedure.

Candidate Transformations In this step, the aim is to find candidate similarity transformations connecting the face instances between the frames. The pairs of features obtained in the previous step can be used to achieve this goal. Three matched pairs are enough to estimate the transformation. However, such a method would not be robust enough and would not guard against frequent statistical outliers in the set of matched pairs. Therefore, we need to have many potential transformations and choose the best one at the end. Unfortunately, the high number of existing pairs doesn't allow us to try all the combinations. Instead, RANSAC [4] is used as a means of getting a set of candidates out of a noisy and large set of matches.

The number of iterations that RANSAC should complete depends on the expected percentage of outliers, the desired confidence, as well as the number of points chosen at each iteration [6]. Making the rather conservative choice of setting the percentage of outliers at 70% and the confidence at 0.01, and knowing that we need three points, we get that the required number of iterations is close to 300. During each iteration, three matches are randomly chosen and used to estimate a potential similarity transformation. A first evaluation of the result is made through its *consensus set* (1). All the locations of the template features are tranformed and the resulting locations are compared to those of the matches. The pairs with Euclidean distance between the locations less than lim build the consensus set of the transformation [6].

$$conSet_T = \{(x, y)| \quad ||Tx - y||_2 < lim\} \qquad (1)$$

where

(x, y)	is a pair of matched locations
x	is the location from the first instance
y	is the location from the second instance
T	is the transformation.

An appropriate value for lim can be determined by observing the distribution of the occuring distances and test the performance of some corresponding thresholds. In our case, we found the median of the distances to usually lie in the interval $[2, 5]$. We have tested the performance for thresholds in $[1, 4]$ with step 0.5. The best results where obtained when lim was set to 2.

Candidates with consensus sets consisting merely of outliers should be rejected. One way to check the quality of a candidate is through the size of its consensus set. Only the transformations with consensus sets of sufficient size should be considered further (2).

$$|conSet_T| > d \qquad (2)$$

where

d is the size threshold
T is the transformation

The threshold d can be determined by demanding a low value for the probability that a set of size greater than d consists merely of outliers [6]. In our implementation, we have set this probability to 0.03 and the corresponding value for d was 10.

Overall, the performance of the algorithm is moderately sensitive to the values of the parameters. In particular, small changes in the number of iterations that RANSAC completes has little effect on the outcome. On the other hand, settings for d and lim affect more the performance as they determine the boundary between good and bad candidates.

There might be cases where the above mechanism fails to come up with a non-empty set of candidates. This can happen when the features are not robust or discriminative enough to be matched correctly and give a good estimation of the face transformation. That is often the case when the frame is blurred. To address the issue, we need a complementary mechanism. Taking into account the time proximity of consecutive frames, we can assume that the face location doesn't differ much. Therefore, by combining several translation parameters, resulting in small to average displacements, we can create a promising set of candidate transformations. Of course, this procedure fails to account for changes in the face orientation and scale. However, it is used rather infrequently and the introduced error is thus limited.

Evaluation In AAM-based methods the appearance of the face in the current image is compared to the appearance generated by the model to optimize the model parameters for the current image. In our case we use the color and orientation descriptors of the tracked face template to rate the candidate transformations and choose the best one for the current frame.

More specifically, a set of possible templates is generated using the candidate transformations to adjust the bounding box of the previous template. Subsequently, the color and orientantion descriptors of each candidate are measured over the image patch indicated by its box. In order to get HOG descriptors comparable to those of the previous template, we need to take into consideration the relative nature of the local face orientation with the orientation of the surrounding box. Therefore, the rotation incorporated into each transformation is used to adjust the pixel gradient orientation before building the histograms of the HOG descriptor. Finally, candidate templates are evaluated by measuring their agreement with the color and orientation descriptors of the previous template. The one with the lowest score is then selected as the current template (lower score indicates greater agrement in color and orientation).

The score measure used in the evaluation procedure is the Euclidean distance between the color and orientation descriptors. Each of them is compared individually resulting in two distinct distances, $D_{C_{T_i}}$ and $D_{O_{T_i}}$ respectively, corresponding to each template.

$$D_{C_{T_i}} = |color_{T_i} - color_{n-1}|_2 \quad (3)$$

$$D_{O_{T_i}} = |HOG_{T_i} - HOG_{n-1}|_2 \quad (4)$$

where

T_i is a candidate template for frame n
$color_{n-1}$ is the color descriptor of the previous template
HOG_{n-1} is the orientation descriptor of the previous template

The final score is then calculated as shown in (5). It is greater or equal to 2 and it takes its lower value whenever the two distance measures are minimized for the same template.

$$score = D_{C_{T_i}}/min_{T_i}(D_{C_{T_i}}) + D_{O_{T_i}}/min_{T_i}(D_{O_{T_i}}) \quad (5)$$

where T_i ranges over the set of candidate templates.

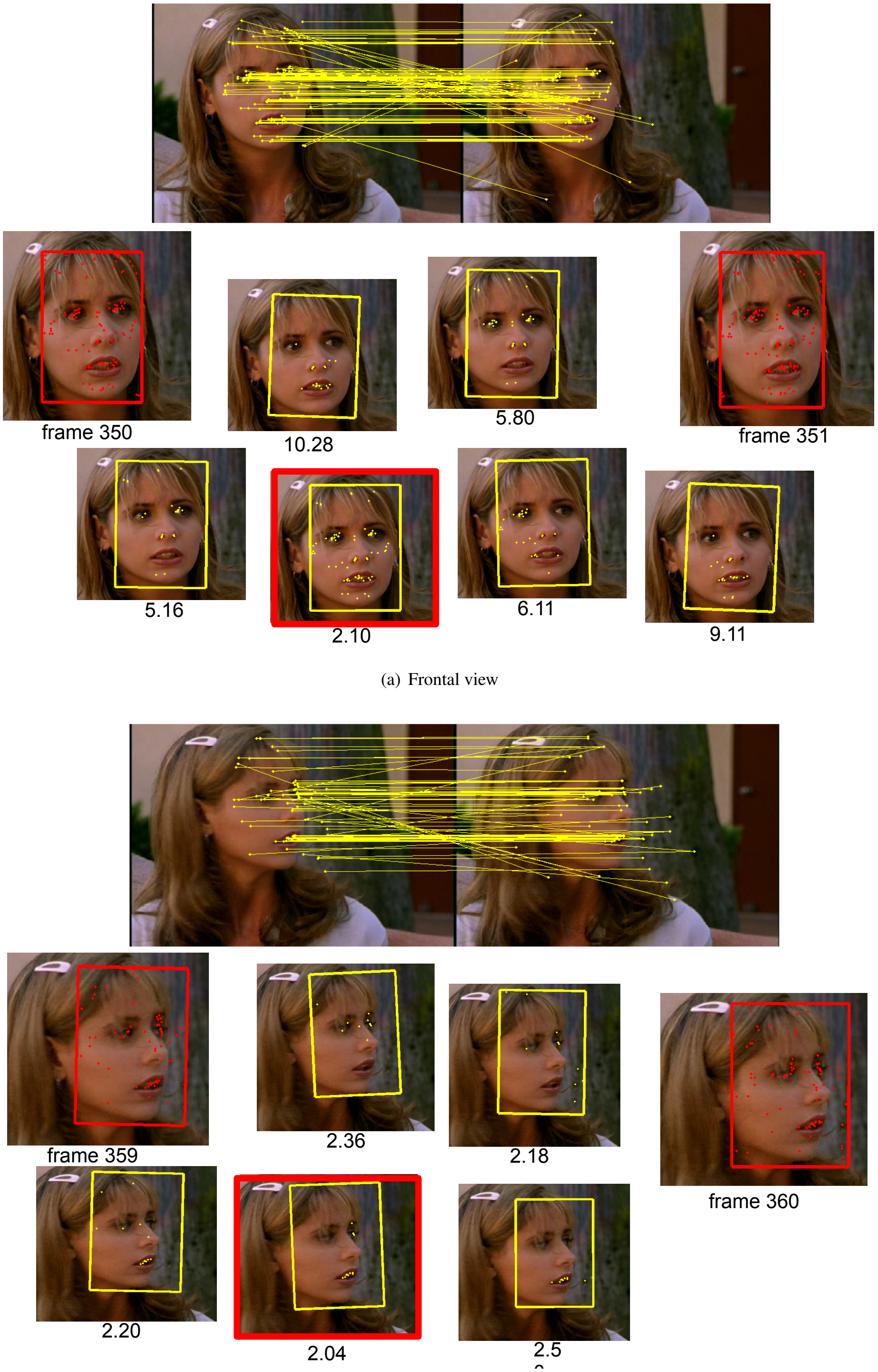
Winner is considered to be the template with the smallest score. Examples of the matching procedure, the resulting candidates and their scores can be seen in figure 4.

5. Results

We have tested the performance of the algorithm under different settings in order to identify the face distortions that can be tolerated and the conditions, if any, that may lead to failure. The video data used for evaluation are taken from our personal collections as well as the TV-series 'Buffy, The Vampire Slayer'. The data contain a number of challenging situations: expression and pose changes, excessive movement of the subject, poor illumination, and poor image quality due to video compression.

Expression In the first set of experiments, we study the consequences of extensive changes in expression and test the capability of the algorithm to track the face correctly while its shape and appearance is altered. An example of the resulting tracks in two differrent video sequences is shown in figure 5(a).

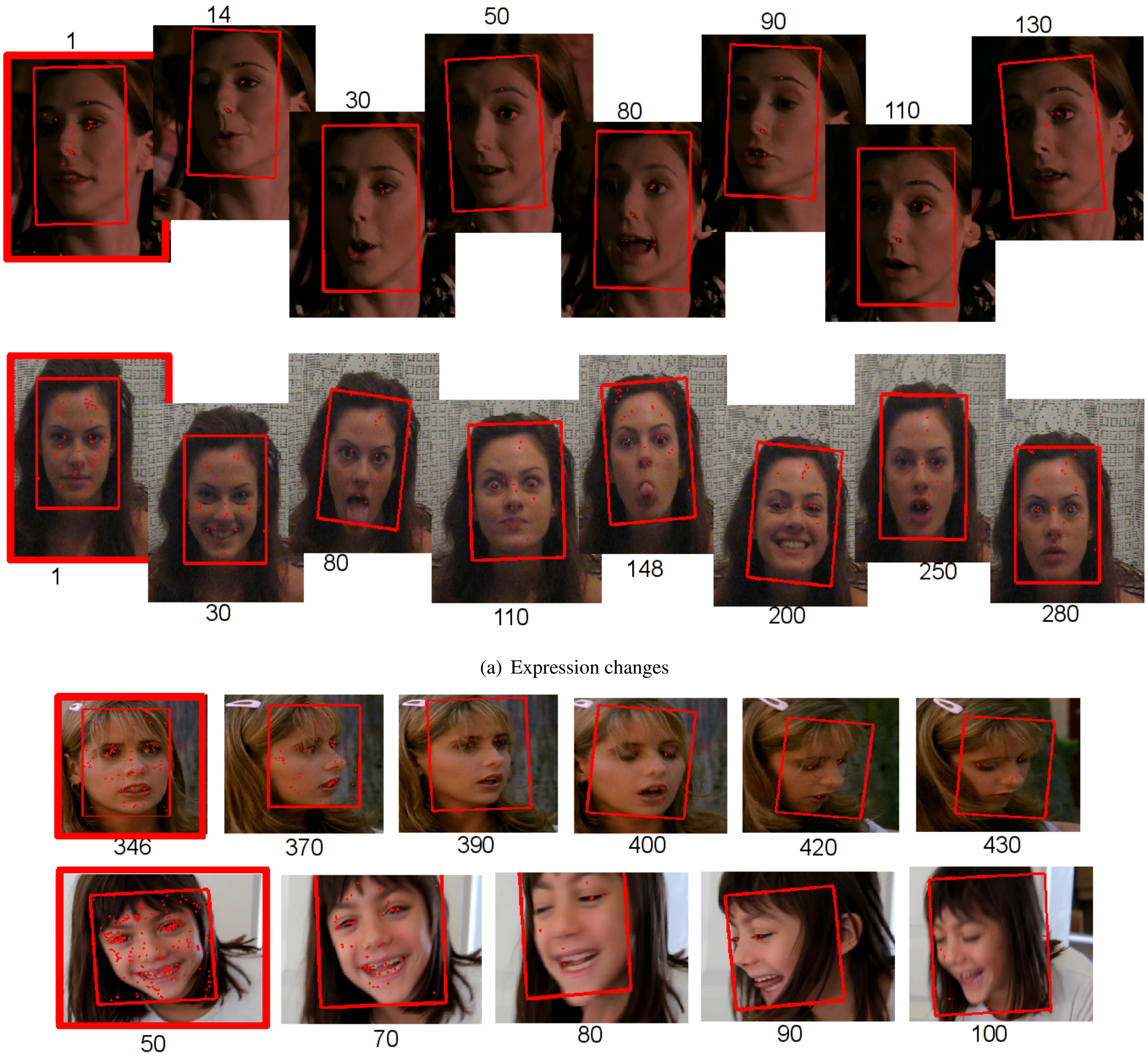
The results of the tests show that, in most cases, changes in expression do not affect the tracking procedure. The facial area is correctly identified despite the existence of significant differences between the tracked face instances. The only potential implication is the inability to include the whole area of the chin accurately in cases where there is a major extension to its length. An example is frame 80 in figure 5(a).



(a) Frontal view

(b) Non-frontal view

Figure 4. Examples of the matching and evaluation steps for frontal and non-frontal views. The tracking starts from the frame at the left where the red box and the red points indicate the current template box and its invariant features. The set of matches is shown at the top of each figure, and the images with the yellow boxes below it, indicate some of the induced candidates. For each candidate, the invariant features building its consensus set are represented by yellow points and its score is written under the image. The winner is surrounded by a red window. Finally, the new template box and its invariant features are shown at the right side.



(a) Expression changes

(b) Non-frontal views

Figure 5. Tracking examples. The number accompanying each picture is the corresponding frame No. The red window indicates the starting frame where the initial template box and the centers of the corresponding invariant features are shown in red. The red boxes in the following frames show the template boxes estimated by the algorithm. The red points indicate the centers of the features building the consensus set of the selected transformations.

Non-frontal views Tracking a face under variations in pose is a challenging task. Frontal and profile views differ in great extent. Combining and handling them simultaneously is frequently a problem for many computer vision systems. The problem is made even harder by the extensive variety of possible poses.

Our tracking algorithm has been tested under such conditions and some partial results are shown in figure 5(b). The algorithm manages to track faces correctly from frontal to profile views and vice versa as well as some more extreme poses like those of frames 420 and 430 in figure 5(b). Possible drift may occur in case of a number of consecutive, highly blurred frames with intense face movement. Such an example is shown in figure 6. Usually, under excessive blurring the affine invariant features are not detected or, perhaps, are not matched correctly. In that case, the candidate templates are generated by a purely translational model which fails to predict more complicated moves. This fact, in combinanion with the poor face appearance due to blurring, leads to drifting, if the conditions are present in many adjacent frames.

Blurring is a challenging distortion not only for tracking

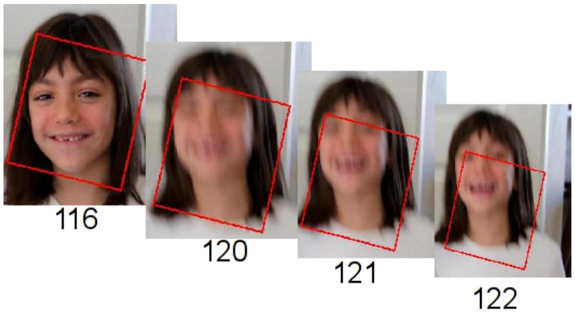


Figure 6. Drift example in case of intense blurring and abrupt motion

but also for detection algorithms. The implementation of Viola-Jones frontal face detector included in Open CV also failed to detect the face in frames 120,121 and 122 of figure 6.

Translation All the previous face distortions may happen while a person is moving. In such cases, there is a greater displacement of the face and the background as well as the illumination of the scene change rapidly . We have tested our algorithm under such conditions and figure 7 shows some of the resulting tracks. As we can see, the extra motion present in the scene doesn't affect the performance of our system which manages to keep track of the face accurately. However, if the person moves far away from the camera, the region of the face becomes small. The restricted size of the tracking target – especially in case of non-frontal views – and the fair image quality, because of the video compression, limits the number of detected features and diminishes the performance of the algorithm. An example of a small, profile, face instance with insufficient number of invariant features is shown in figure 8.

6. Conclusions & Future Work

In this paper, we have studied the application of affine invariant features in combination with color and orientation descriptors for face tracking. We tested the proposed algorithm under various circumstances and found that it succeeds in tracking faces accurately from frontal to profile views under changes in expression, illumination, background clutter and intense motion. The only identified condition that decreases the performance is the existence of a number of adjacent blurred frames. In such cases the poor image quality and the simplicity of the complementary mechanism, used as the tentative option, leads to drifting effects.

In the future, we would consider the following changes:

a. Development of a more sophisticated complementary mechanism capable of handling blurring in adjacent frames.

b. Use of more than three points for estimating candidate transformations. The resulting overconstrained system would be solved by Least-Squares optimization and RANSAC would be used to build a set of such candidates.

c. Inclusion and development of new detectors and descriptors with greater robustness under blurring.

References

[1] Shai Avidan. Ensemble tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):261–271, 2007.

[2] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato. Sift features tracking for video stabilization. In *CIAP07*, pages 825–830, 2007.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893, 2005.

[4] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[5] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*, chapter 3. Prentice Hall, 2002.

[6] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*, chapter 17. Prentice Hall, 2002.

[7] Bohyung Han, Ying Zhu, Dorin Comaniciu, and Larry Davis. Kernel-based bayesian filtering for object tracking. In *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 227–234, 2005.

[8] Allan D. Jepson, David J. Fleet, and Thomas F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, 2003.

[9] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector, 2004.

[10] A. Kobayashi, J. Satake, T. Hirayama, H. Kawashima, and T. Matsuyama. Person-independent face tracking based on dynamic aam selection. In *FG08*, pages 1–8, 2008.

[11] J.H. Kwon, K.M. Lee, and F.C. Park. Visual tracking via geometric particle filtering on the affine group with optimal importance functions. pages 991–998, 2009.

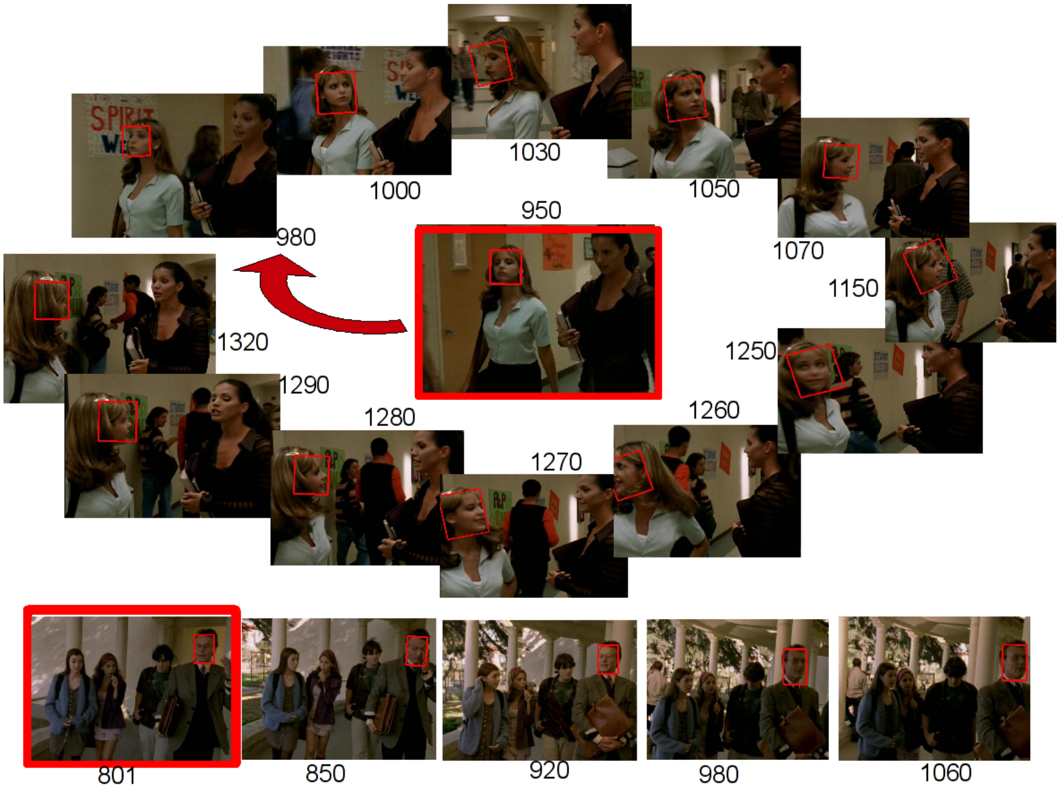


Figure 7. Tracking while the person is moving. The number accompanying each picture is the corresponding frame No. The red window indicates the starting frame. The red boxes in the following frames show the template boxes estimated by the algorithm.

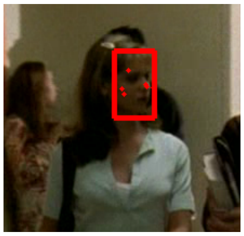


Figure 8. Example of a profile face instance with a small number of invariant features indicated by the red points.

[12] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[13] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *In British Machine Vision Conference*, volume 1, pages 384–393, 2002.

[14] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, 2005.

[15] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, 2004.

[16] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[17] Iain Matthews Ralph Gross and Simon Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(1):1080–1093, November 2005.

[18] Jianbo Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600, 1994.

[19] J. Sivic, M. Everingham, and A. Zisserman. "Who are you?" – Learning person specific classifiers from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[20] Iryna Skrypnyk and David G. Lowe. Scene modelling, recognition and tracking with invariant image features.

```
INPUT: face teplate in the first frame ( Tem1 )
OUTPUT: face templates for the rest of the video sequence
PARAMETERS: k, d, lim

current template = Tem1
ROI = region of Tem1 extended by 50% at each direction

for all frames (except for the first one)

  1. MATCHING
  A = {invariant features of the current template}
  B = {invariant features of current frame in the ROI}
  S = NN-matching(A,B)

  2. CREATION OF CANDIDATES
  for iterations = 1 : k
    Pick 3 pairs from S
    Calculate the corresponding similarity transformation T
    if |consensus set of T| > d
      add to candidates

  if no valid candidate found
    for various tx and ty
        add T = [ 1 0 tx ; 0 1 ty ; 0 0 1] to candidates

  3. EVALUATION
    Calculate scores for every candidate
    New template = candidate with lowest score
    ROI = region of new template extended by 50% at each direction
end
```

Figure 9. Pseudocode of the algorithm

In *ISMAR '04: Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 110–119, Washington, DC, USA, 2004. IEEE Computer Society.

[21] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *Int. J. Comput. Vision*, 59(1):61–85, 2004.