Online Feature Selection Using Mutual Information for Real-Time Multi-View Object Tracking

Alex Po Leung and Shaogang Gong

Department of Computer Science
Queen Mary, University of London
London, E1 4NS
{alex, sgg}@dcs.qmul.ac.uk

Abstract. It has been shown that features can be selected adaptively for object tracking in changing environments [1]. We propose to use the variance of Mutual Information [2] for online feature selection to acquire reliable features for tracking by making use of the images of the tracked object in previous frames to refine our model so that the refined model after online feature selection becomes more robust. The ability of our method to pick up reliable features in real time is demonstrated with multi-view object tracking. In addition, the projective warping of 2D features is used to track 3D objects in non-frontal views in real time. Transformed 2D features can approximate relatively flat object structures such as the two eyes in a face. In this paper, approximations to the transformed features using weak perspective projection are derived. Since features in non-frontal views are computed on-the-fly by projective transforms under weak perspective projection, our framework requires only frontal-view training samples to track objects in multiple views.

1 Introduction

Much effort has been made to solve the problem of real-time object tracking over the years. However, tracking algorithms still suffer from fundamental problems including drifts away from targets [4] (partially due to change of viewpoint), inability to adapt to changes of object appearance, dependence on the first frame for template matching [5], instability to track objects under deformations (e.g. deformed contours), the inefficiency of Monte Carlo simulations for temporal tracking [6], and reliance on gradients by active contours [7], i.e. problems with similar intensities on the background and the object, or high gradient edges on the object itself. These problems are due to the complexity of the object dynamics. We also have to deal with difficult tracking conditions which include illumination changes, occlusions, changes of viewpoint, moving cameras and non-translational object motions like zooming and rotation.

Recent techniques use more complex and descriptive representations for tracking [8], [9], [10], [11]. A more descriptive representation may reduce the dependency on temporal information for tracking. There are a number of advantages to use a more descriptive representation. It makes tracking more robust in cluttered scenes. Less constrained physical state trajectories such as those containing discontinuities may also be

tracked. If the representation can encode the appearance of the object more discriminatively, it allows the tracking of objects largely relying on framewise detections without much temporal analysis, such as Viola-Jones detector-based tracking [8]. However, it is both difficult and expensive to obtain statistics to build a 3D model for object detection or tracking while 2D appearance models such as [17], [3], [9] and [11] have been very successful. When multi-views are considered, a huge amount of data is needed for each view for the training for a particular object. Such a huge dataset is impractical to create and it is also computationally expensive to train such a multi-view model. It is hard to obtain thousands of samples in each view and train a system for weeks or even months to track a particular object.

In this paper, a technique to track non-rigid objects in changing views with only frontal-view training samples is developed. Non-frontal views are deduced from frontal-view samples by geometric transformations. Using weak perspective projection, our method can track objects with a roughly flat surface such as faces or cars. It is obvious that, even for a roughly flat surface, there could be some uneven structures such as the nose on a face. We further use Mutual Information for online feature selection to acquire reliable features which are the relatively flat in our case. Our implementation picks up flat features in real time for multi-view object tracking.

Haar-like features selected by AdaBoost [3] can model non-rigid objects under different lighting conditions. We explore the possibility to devise a tracking algorithm using Haar-like features selected by AdaBoost as the representation [3]. Kalman filters are adopted to track the state variables after projective warping in every frame. They are used to temporally confine the parameter space of the transform. Our tracker is able to track non-rigid objects and the initialization of tracking is completely automatic. A single appearance model for both detection and tracking means a smooth transition from detection to tracking. No assumption on color is made in our model.

In the rest of this paper, Section 2 presents our proposed methods to compute warped Haar-like features. A technique for online feature selection using Mutual Information is proposed in Section 3. Section 4 presents experiments to test our proposed framework. Conclusions and future work are given in Section 5.

2 Projective Warping of Rectangle Features

Viola and Jones [3] make use of an intermediate representation for images called the integral image or summed-area table [12] to obtain the sum of pixel values for rectangle features with no more than four array references. The integral image is vital to computational efficiency for computing rectangle features. However, features are no longer rectangular after projective transforms. Therefore, we cannot calculate the features directly from the integral image. We propose to use a generalization of the method to calculate the features while we can still use the integral image. The generalization was proposed originally by Glassner [13] for texture mapping. It computes the average pixel value within a quadrilateral to an arbitrary degree of accuracy using the integral image with additional computation depending on the accuracy required. Glassner approximates a non-rectangular shape by rectangles. Two methods can be used to do this:

additive and subtractive synthesis. Arbitrarily accurate features can be obtained and the integral image can still be used to retain the efficiency of the original appearance model.

An alternative way is to approximate projective transforms. This method makes the computation much more efficient. A planar projective transformation is a transformation with eight free parameters. A search in the parameter space could be computationally very expensive. An advantage to approximate projective transforms is to reduce the dimensionality of the parameter space. High dimensionality leads to expensive computation and sparsity of data which prevents the search from finding the correct set of parameters. A common approach is to approximate projective transforms by considering weak perspective projection such as planar affine transforms. For a planar affine transform, the number of free parameters is reduced from eight to six.

2.1 Approximating Projective Transforms

We may use weak perspective projection to approximate the perspective projection of rectangle features such as Haar-like features. Let us consider a rectangle feature with corners P_i' where $i = 1$ for the top left, 2 for the top right, 3 for the bottom right and 4 for the bottom left.

$$P_i = R_o P_i', \tag{1}$$

where $R_o = R_{o1}(\alpha)R_{o2}(\beta)R_{o3}(\gamma)$ is the rotation of the object and P_i are the corners after rotating the feature. We consider tracking the out-of-plane rotations of an object (i.e. pitch and yaw):

$$P_i = R_{o1}(\alpha)R_{o2}(\beta)P_i'. \tag{2}$$

The rotational matrix R_o for the object rotation with pitch and yaw is $R_{o_1}(\alpha)R_{o_2}(\beta) =$

$$\begin{bmatrix} cos\beta & 0 & sin\beta \\ sin\alpha sin\beta & cos\alpha & -sin\alpha cos\beta \\ -cos\alpha sin\beta & sin\alpha & cos\alpha cos\beta \end{bmatrix}.$$

The corner of a rectangle feature after the pitch and yaw rotations in world coordinates is, therefore,

$$X_w = cos\beta X_w', \tag{3}$$

$$Y_w = sin\alpha sin\beta X_w' + cos\alpha Y_w', \tag{4}$$

where (X_w', Y_w') is the corner before rotations in world coordinates. Note that we rotate the object symmetrically by locating it on the x-y plane and its center to be in the origin in world coordinates so $Z_w' = 0$ and, under weak perspective,

$$\bar{Z_w} \approx 0. \tag{5}$$

A rectangle feature can be on any part of the object. Thus, $\bar{Z_w}$ is not exactly zero. In homogeneous coordinates, the matrix equation of perspective projections can be written

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = M \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix},$$

where $x = \frac{x_1}{x_3}$, $y = \frac{x_2}{x_3}$ are in image coordinates and

$$M = \begin{bmatrix} -fr_{11} & -fr_{12} & -fr_{13} & fR_1^T T \\ -fr_{21} & -fr_{22} & -fr_{23} & fR_2^T T \\ r_{31} & r_{32} & r_{33} & -R_3^T T \end{bmatrix}.$$

where R_i, $i = 1, 2, 3$, is a three dimensional vector formed by the i-th row of the matrix R. Under weak perspective projection,

$$x_{wp} = \frac{x_1}{x_3} \approx \frac{fR_1^T(T - P_w)}{R_3^T(\bar{P_w} - T)},$$

$$y_{wp} = \frac{x_2}{x_3} \approx \frac{fR_2^T(T - P_w)}{R_3^T(\bar{P_w} - T)}.$$

Let $R = I$, i.e. there is no rotation between the world coordinates and the camera coordinates. Thus,

$$x_{wp} \approx -\frac{f(X_w - T_X)}{\bar{Z_w} - T_Z}, \qquad y_{wp} \approx -\frac{f(Y_w - T_Y)}{\bar{Z_w} - T_Z}.$$

Using Equation 5, a corner in image coordinates under weak perspective projection is then

$$P_{i_{wp}} = \begin{bmatrix} \frac{f(X_w - T_X)}{T_Z} & \frac{f(Y_w - T_Y)}{T_Z} & f \end{bmatrix}^T. \tag{6}$$

By combining Equations 3, 4 and 6, a corner after the rotations of the object becomes

$$P_{i_{wp}} = \begin{bmatrix} \frac{f(cos\beta X'_w - T_X)}{T_Z} \\ \\ \frac{f(sin\alpha sin\beta X'_w + cos\alpha Y'_w - T_Y)}{T_Z} \\ \\ f \end{bmatrix}$$

under weak perspective projection in image coordinates. Let us assume there is only the pitch rotation or the yaw rotation and the two rotations don't occur at the same time. That means either $\alpha = 0$ or $\beta = 0$. So, $sin\alpha sin\beta X'_w = 0$. In reality, especially for face tracking, it is natural to assume the object to rotate either with the pitch or the yaw. Therefore, when α becomes large, $\beta \approx 0$, or when β becomes large, $\alpha \approx 0$. Hence, $sin\alpha sin\beta X'_w \approx 0$ and

$$P_{i_{wp}} = \begin{bmatrix} \frac{f(cos\beta X'_w - T_X)}{T_Z} & \frac{f(cos\alpha Y'_w - T_Y)}{T_Z} & f \end{bmatrix}^T.$$

Notice that, since x_{wp} in the above is independent of Y'_w and y_{wp} independent of X'_w after rotations, a rectangle feature after rotations is still rectangular under weak perspective. The width and height of the rectangle feature after rotations in image coordinates are

$$x_{2_{wp}} - x_{1_{wp}} = \frac{f\ cos\beta(X'_{2_w} - X'_{1_w})}{T_Z}, \text{ and}$$

$$y_{1_{wp}} - y_{4_{wp}} = \frac{f\ cos\alpha(Y'_{1_w} - Y'_{4_w})}{T_Z}.$$

The aspect ratio of the rectangle feature η after the rotations α and β becomes $\frac{cos\beta}{cos\alpha}\eta_0$, where $\eta_0 = (X'_{2_w} - X'_{1_w})/(Y'_{1_w} - Y'_{4_w})$ is the aspect ratio before rotations.

This shows that, under weak perspective projection, the projective warping of a rectangle feature can be approximated by simply varying the aspect ratio of the rectangle feature. It gives us an extremely efficient means to track a rotating object. Only the aspect ratio η, the scale s and the centroid location (x_l, y_l) need to be tracked.

3 Feature Selection

It is both difficult and expensive to obtain statistics to build a 3D model for object detection or tracking. For face detection and tracking, different people have their own 3D face shapes. 2D appearance models cannot be trained easily to cope with view variations due to both the lack of the huge amount of labelled data for multi-views and the computational cost of training.

We use projective warping to transform learned Haar-like features. However, not all features are roughly flat. Therefore, the warping can introduce tracking errors due to the linearity of projective transformation if a measure of feature "goodness" is not evaluated on-the-fly. The best features which are approximately flat need be selected in real time after projective transforms have been made. We make use of the images of the object which has been tracked in previous frames to refine our model so that the refined model after online feature selection becomes more robust to track the object in different views.

3.1 The Mutual Information

We use Mutual Information to select approximately flat features which should be reliable for projective warping as the object rotates. The mutual information measures the statistical dependence between two variables. It has been shown to be a very effective measure for selecting a small set of relevant features from a large set of potential features very quickly [16].

We have a set of features selected by AdaBoost for objects in single view. Redundancy between features is not considered because redundancy is eliminated during the AdaBoost training. Hence, for computational efficiency, we simply use the mutual information instead of the conditional mutual information [16] considered to take into account redundancy between features. For continuous probability distributions, the mutual information is defined as

$$\mathcal{I}(i) = \int_{x_i}\int_{y} p(x_i, y) log \frac{p(x_i, y)}{p(x_i)p(y)} dxdy.$$

It is hard and inefficient to estimate the continuous distributions $p(x_i, y)$, $p(x_i)$ and $p(y)$ [18] for Feature i. Instead of estimating the distributions of the features directly, we use the output of the weak classifiers [3]. The statistical dependence of the weak classifer

Fig. 1. Experiment 1 - Tracking a non-frontal female face in real-time. The figure shows example images from an indoor sequence (Video Sequence 1).

output of a feature and the output of the AdaBoost cascade [3] is determined by the mutual information. Both of the outputs are Boolean values so we can use the discrete form of the mutual information $I(i) =$

$$\sum_{x_i}\sum_{y} P(X = x_i, Y = y) log \frac{P(X = x_i, Y = y)}{P(X = x_i)P(Y = y)}.$$

Given a finite training set, one using frequency counts can only obtain an estimate of the mutual information as follows:

$$\hat{I}(i) = \frac{log\, n}{n} \sum_{x_i y} n_{x_i y} log \frac{n_{x_i y}}{n_{x_i} n_y},$$

where n is the total number of occurrences and $n_{x_i y}$, n_{x_i} and n_y are respectively the numbers of occurrences of the pair (x_i, y), x_i and y. Hutter [2] obtained the distribution, expectation value and variance of the mutual information by using a Bayesian approach. The expectation value defined as follows containing a correction term, $\frac{(r-1)(s-1)}{2n}$, is a more accurate estimate for the mutual information:

$$E\{I(i)\} = \sum_{x_i y} \frac{n_{x_i y}}{n} log \frac{n_{x_i y} n}{n_{x_i} n_y} + \frac{(r-1)(s-1)}{2n} + O(n^{-2}).$$

When the tracked face is frontal, all features learned by AdaBoost are almost equally discriminative. However, the more the face rotates, the lower the mutual information of an uneven feature gets due to the linearity of projective transformation. On the contrary,

Fig. 2. Experiment 2 - Tracking a non-frontal male face in real-time. The figure shows example images from an outdoor sequence with a moving hand-held camera (Video Sequence 2).

for an ideally flat feature, the mutual information remains the same as the face rotates. Thus, as we transform the features geometrically under weak perspective, features relatively flat are more stable for tracking and, thus, associate with small variations in the mutual information (i.e. small variances) when the view is changing. Instead of finding a set of features with the largest mutual information, we should look for a set of features with the smallest corresponding variances of the mutual information so that features more stable and, therefore, flat are selected. To measure the stability of a feature, the variance of the mutual information [2] is $Var\{I(i)\} =$

$$\frac{1}{n}\sum_{x_i y}\frac{n_{x_i y}}{n}\left(log\frac{n_{x_i y}n}{n_{x_i}n_y}\right)^2 - \frac{1}{n}\left(\sum_{x_i y}\frac{n_{x_i y}}{n}log\frac{n_{x_i y}n}{n_{x_i}n_y}\right)^2 + O(n^{-2}).$$

It to the order of n^{-1} can be written

$$\frac{(log\ n)^2}{n^2}\left(\sum_{x_iy} n_{x_iy}\left(log\frac{n_{x_iy}}{n_{x_i}n_y}\right)^2 - \frac{1}{n}\left(\sum_{x_iy} n_{x_iy} log\frac{n_{x_iy}}{n_{x_i}n_y}\right)^2\right).$$

When we compare the variances of the mutual information of the features, the scaling factor $\frac{(log\ n)^2}{n^2}$ can be ingored. Thus, to select the most reliable features, we compare

$$\sum_{x_iy} n_{x_iy}\left(log\frac{n_{x_iy}}{n_{x_i}n_y}\right)^2 - \frac{1}{n}\left(\sum_{x_iy} n_{x_iy} log\frac{n_{x_iy}}{n_{x_i}n_y}\right)^2.$$

In other words, we select the most reliable or stable features by picking up the smallest corresponding variances of the mutual information of the features. Additionally, for the strong classifier of AdaBoost, we set the weight of Feature i, $\alpha_i = 0$ to reject Feature i so that the weights of the majority vote remain the same except for the rejected features.

4 Experimental Results

We use the MIT-CBCL face dataset [14] which consists of 6,977 cropped images (2,429 faces and 4,548 nonfaces). The resolution of the images is 19×19 and slightly lower than 24×24 used by Viola and Jones [3].

After the Viola-Jones detector initializes our tracker, four Kalman filters are separately used to track the aspect ratio η, the scale s and the centroid location of the object (x_l, y_l). A 5-stage cascade of AdaBoost [3] is used in our experiments. There are only 127 features in the 5 stages. The 5 stages separately compute 2, 7, 20, 50 and 50 features.

Experiment 1 (see Figure 1) shows a video (Video Sequence 1) with $|\beta| < 90°$. It shows that faces with relatively large $|\beta|$ could also be tracked. It is clear that the side views share some common features with the frontal view after projective transforms. Experiment 2 (Figure 2) shows tracking a non-frontal male face outdoors with a moving hand-held camera (Video Sequence 2). Both experiments demonstrate that our tracker can track deformable objects from different viewpoints, i.e. faces with different expressions in different views in this case. In order to evaluate the performance of our warping method and the proposed Mutual Information (MI) feature selection technique, additional four experiments (3, 4, 5 and 6) are performed using the two videos used in Experiment 1 and Experiment 2. Figure 3 shows tracking failures in our experiments and Table 1 shows the comparisons of those experiments. In Figure 5, the number of features rejected by the variance of the mutual information is shown to be stabilized after the initial two hundred frames in Experiments 1 and 2. Furthermore, the number of subwindows accepted by the cascade of AdaBoost, to a certain extent, indicates the stability of the tracker. So, we compare the numbers of subwindows accepted during

tracking with MI feature selection and without MI feature selection to better understand the effect of online feature selection. Besides, since Figure 3 shows all of the tracking failures in our experiments are due to the fact that no subwindow is classified to be a face, we are also interested in seeing when the number of subwindows classified to be a face becomes low. Figures 6 and 7 are the plots of the numbers of subwindows accepted during tracking respectively without MI feature selection and with MI feature selection for Video Sequence 1. Moreover, Figures 8 and 9 are the plots of the numbers of subwindows accepted during tracking respectively without MI feature selection and with MI feature selection for Video Sequence 2. We found that, when the number of subwindows becomes zero, the tracker does not necessarily fail because of the Kalman filters. However, when the number of subwindows becomes zero in several consecutive frames, the tracker usually fails. As we can see in Figures 6, 7, 8 and 9, the tracker with our proposed online MI feature selection method is much less likely to lose track of the face when the number of subwindows goes below 3. In Experiment 7 (Figure 4), a tracker is initiated to track a person wearing glasses in the background. The resolution of the face is approximately 19×19 in order to evaluate tracking with low-resolution images. Our tracker can track low-resolution faces provided that the out-of-plane rotation angle β is not very large to avoid high quantization noise.

In our current experiments, the tracking frame rate is 7.4 frames per second with the frame size 320×240. The code for the interface is in Matlab. Our core code is compiled by gcc on Cygwin on an AMD Athlon 1.68GHz machine.

Table 1. Comparisons of Our Experiments

Experiment Number	MI Feature Selection Used	Warping Used	Video Sequence Number	Number of Frames Tracked
1	Yes	Yes	1	500 (End of Sequence)
2	Yes	Yes	2	526 (End of Sequence)
3	No	Yes	1	431 (Background)
4	No	No	1	17 (Non-Frontal View)
5	No	Yes	2	499 (Partial Occlusion)
6	No	No	2	141 (Non-Frontal View)

5 Conclusion

We have demostrated a system using the projective warping of 2D features to track 3D objects in non-frontal views in real time. Mutual Information for online feature selection to acquire reliable features for tracking is proposed. We demonstrate the ability of

our method to pick up reliable features in real time with multi-view object tracking. Our framework requires only frontal-view training samples. Features in other views are computed by projective transforms under weak perspective projection on-the-fly. Approximations to the transformed features using weak perspective projection are derived.

Future work includes pose estimation making use of the out-of-plane rotation angles α and β, and making the tracker more efficient by using noisy optimization such as implicit filtering for searches in the parameter space for projective transforms.

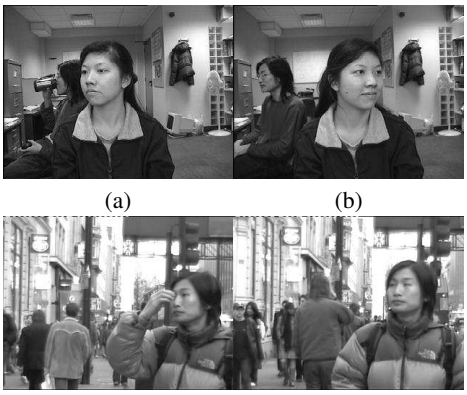

Fig. 3. Notice that all tracking failures in our experiments are due to the fact that no subwindow is classified to be a face in several consecutive frames. (a) and (c) show the failure of the tracker after tracking respectively 431 frames in Experiment 3 due to the background and 499 frames in Experiment 5 due to a partial occlusion. No feature selection is used. The tracker is significantly less robust without online MI feature selection. In (b) and (d), Experiment 4 and Experiment 6 show the failure of the tracker due to view changes after respectively tracking 17 frames and 141 frames. In Experiments 4 and 6, neither feature selection nor geometric transformation is used. The tracker is only able to track very few frames in the sequences without online MI feature selection and geometric transformation.

References

1. R. Collins, Y. Liu and M. Leordeanu, "On-Line Selection of Discriminative Tracking Features," *PAMI*, 2005.
2. M. Hutter, "Distribution of Mutual Information", *Advances in Neural Information Processing Systems*, 14 (2002) 399-406.
3. P. Viola and M. Jones, "Robust real-time object detection," *IJCV*, 2002.
4. I. Matthews, T. Ishikawa and S. Baker, "The Template Update Problem," *PAMI(26)*, No. 6, 2004, pp. 810-815.
5. F. Jurie and M. Dhome, "Hyperplane Approximation for Template Matching," *IEEE PAMI 24(7)*, 996-1000, 2002.

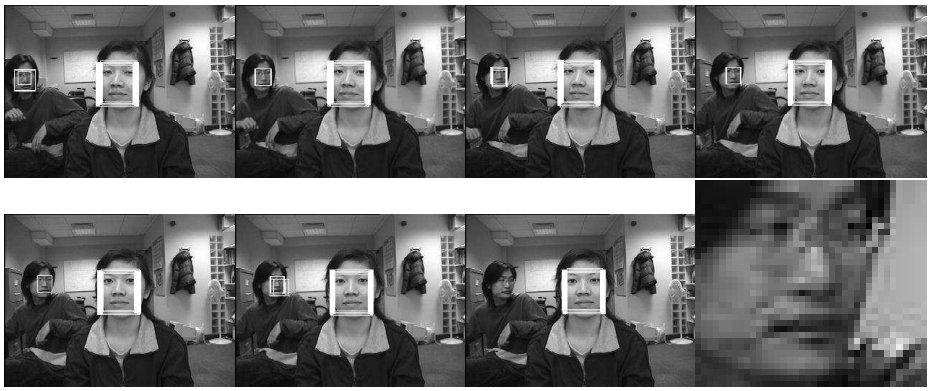

Fig. 4. Experiment 7 - Two trackers are initiated to track both the person in the foreground and the person wearing glasses in the background. The resolution of the face in the background is approximately 19×19 as shown in the magnified image. The tracker loses the face in the last frame due to the large quantization errors of projective warping when the out-of-plane rotation angle β is large with the low resolution.

6. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *Transaction of Signal Processing*, 50(2):174-188, 2002.
7. M. Isard and A. Blake, "CONDENSATION – conditional density propagation for visual tracking," *IJCV*, 29, 1, 5–28, (1998).
8. G. Shakhnarovich, P. A. Viola and B. Moghaddam, "A Unified Learning Framework for Real-Time Face Detection and Classification," *IEEE FG*, pp. 14-21, May 2002.
9. S. Avidan, "Support Vector Tracking," *PAMI(26)*, No. 8, August 2004, pp. 1064-1072.
10. O. Williams, A. Blake and R. Cipolla, "A Sparse Probabilistic Learning Algorithm for Real-Time Tracking," *ICCV*, Nice, France, 2003.
11. M. J. Black and A. Jepson, "EigenTracking: Robust matching and tracking of articulated objects using a view-based representation," *ICCV*, 26(1), pp. 63-84, 1998.
12. F. Crow, "Summed-Area Tables for Texture Mapping," *SIGGRAPH*, Pages 207-212, July, 1984.
13. A. Glassner, "Adaptive Precision in Texture Mapping," *SIGGRAPH*, 20, 4, August 1986, pp. 297-306.
14. CBCL Face Database #1, MIT Center For Biological and Computation Learning, http://www.ai.mit.edu/projects/cbcl.
15. P. Sinha, "Qualitative representations for recognition," *In Lecture Notes in Computer Science, Springer-Verlag*, LNCS 2525, pp 249-262, 2002.
16. F. Fleuret, "Fast Binary Feature Selection with Conditional Mutual Information", *Journal of Machine Learning Research*, 5 (2004) 1531-1555.
17. H. Schneiderman and T. Kanade, "Object Detection Using the Statistics of Parts", *IJCV*, v.56 n.3, p.151-177, February-March 2004.
18. K. Torkkola, "Feature Extraction by Non-Parametric Mutual Information Maximization", *Journal of Machine Learning Research*, 3(Mar):1415-1438, 2003.

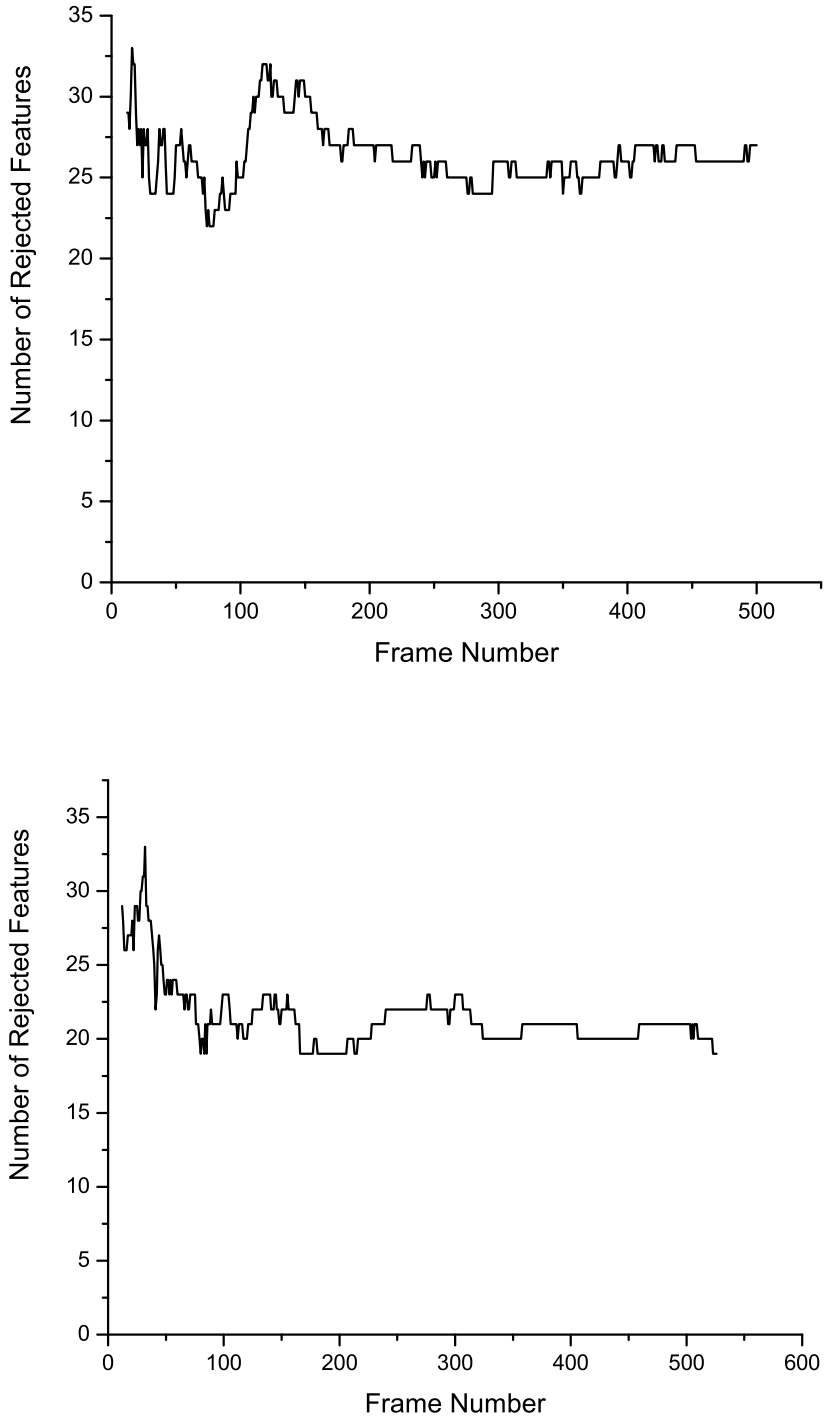

Fig. 5. In Experiment 1 (top figure) and Experiment 2 (bottom figure), the number of features rejected by the variance of the mutual information becomes stabilized after approximately the initial two hundred frames.

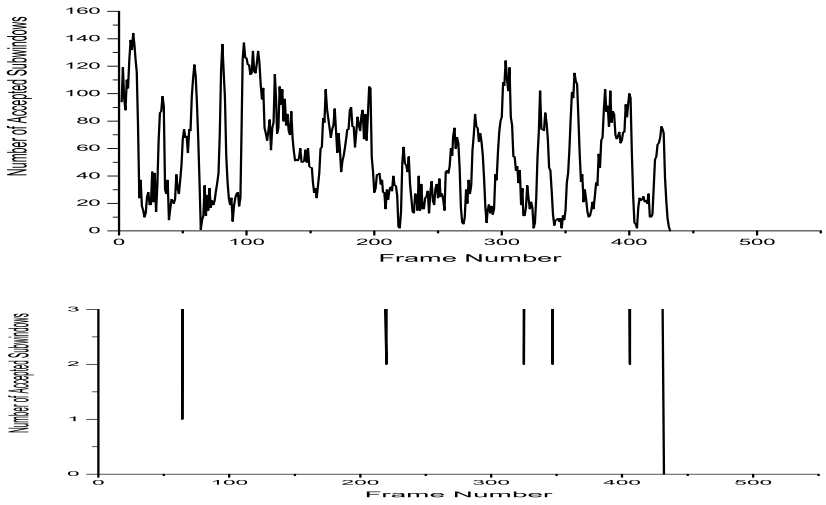

Fig. 6. The above are the same figure with different scales. In Experiment 3, the tracker without online MI feature selection loses the face at Frame 432.

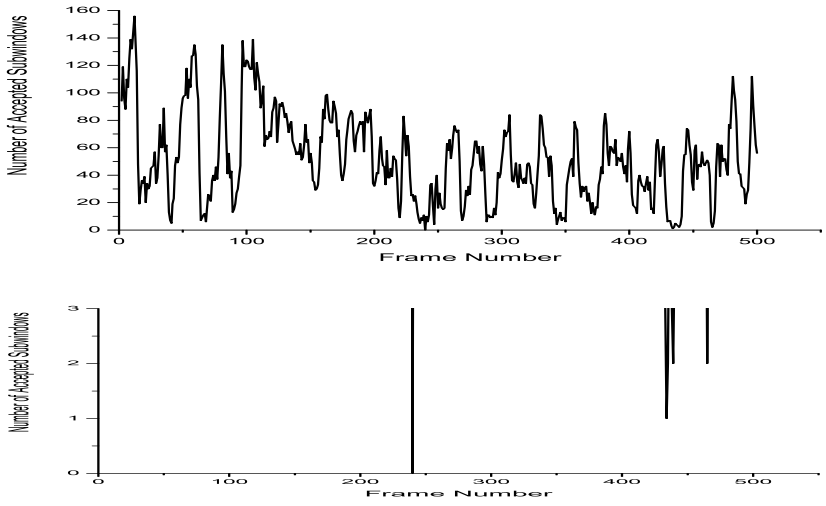

Fig. 7. The above are the same figure with different scales. In Experiment 1, the number of accepted subwindows is much less likely to go below 3 than it is without online MI feature selection as shown in Figure 6.

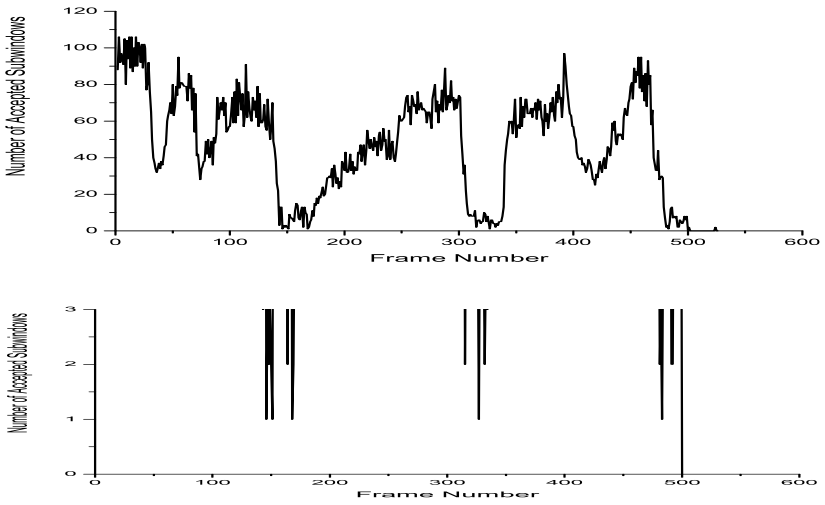

Fig. 8. The above are the same figure with different scales. In Experiment 5, the tracker without online MI feature selection loses the face at Frame 500.

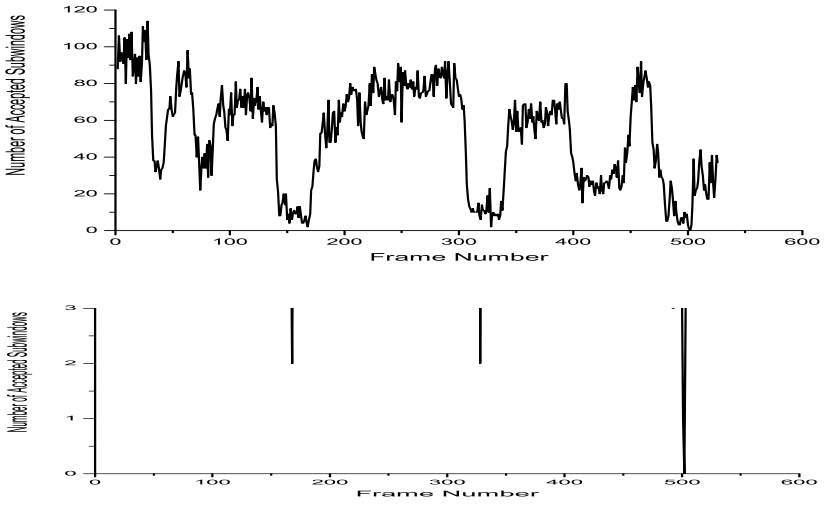

Fig. 9. The above are the same figure with different scales. In Experiment 2, the number of accepted subwindows is much less likely to go below 3 than it is without online MI feature selection as shown in Figure 8.