Tracking Using Online Feature Selection and a Local Generative Model

Thomas Woodley * Bjorn Stenger † Roberto Cipolla *

* Dept. of Engineering
University of Cambridge
{tew32|cipolla}@eng.cam.ac.uk

† Computer Vision Group
Toshiba Research Europe
bjorn@cantab.net

Abstract

This paper proposes an algorithm for online feature selection which improves robustness to occlusions by referring to a localized generative appearance model. Discriminative classifiers based on feature extraction have classically either prepared a fixed prior model by training offline, or continually adapted their classification parameters to any apparent appearance changes. By combining the attractive qualities of each approach, our framework can cope with appearance changes of a target object and will maintain proximity to a static appearance model. Our main contribution is the use of a generative model to guide the online feature selection to regions of an image which maintain a valid appearance. The generative model exhibits the properties of non-negativity, localization and orthogonality. We demonstrate the system in a tracking framework to show improved tracking performance through occlusions.

1 Introduction

A major challenge in visual tracking is handling the appearance variation of the target object. This can be caused by a number of factors including pose variation, shape deformation, lighting changes, as well as occlusions. In this paper we follow a discriminative approach to tracking where a classifier is used to distinguish the object from the background. The classifier uses a set of discriminative local features which is updated at each time step using on-line boosting [4]: using the previous object location as positive example and surrounding regions as negative examples the classifier updates its feature pool and corresponding weights. This flexibility is advantageous for tracking through large appearance changes but leads to the known *template update problem* [14]. The key question is how much adaptability to allow the tracker. In other words: How can one decide whether an appearance change is simply due to pose or lighting variation or due to occlusion of the object?

One strategy is to use a generative object model and determine whether the current target estimate is still valid given this model. One such model representation is an eigenspace model, where the image is modelled by a linear combination of orthogonal basis functions. The model can be used to statistically evaluate the presence of the object. However,

as it is a global object representation it does not provide a straightforward way to estimate local occlusions.

In this paper we introduce a method that uses a *local* generative model to constrain the selection of local features in the classifier in the case of outliers caused by local occlusion. Our generative model is computed from the first few frames in the sequence by local non-negative matrix factorization (LNMF) where the basis functions are orthogonal and sparse and spatially constrained. The key idea in this paper is to identify occluded regions by projecting the current image onto the basis functions. Such outliers are determined by comparing with the distribution of individual coefficients. This information is integrated with online feature selection as follows: If a region is labelled as occluded, the local features in this region are discarded and new features in the non-occluded regions added. Alternatively, the features in the occluded regions can be de-activated for the duration of the occlusion. This way the adaptation to outlier regions is avoided while being able to keep valid classifiers in memory that have large feature support in the target region.

The rest of the paper is laid out as follows: Section 2 discusses related prior work. In section 3 we introduce a new algorithm for combining discriminative and local generative models for improved online feature selection. We report on experiments carried out to verify the approach in section 4 while section 5 concludes with a discussion of the contributions.

2 Prior Work

Adaptive object tracking is a core technique in many applications and has therefore been widely explored. We provide a brief summary of the work most relevant in the context of this paper.

Tracking using classifiers Avidan introduced the idea of using a binary classifier to track an object [1]. A Support Vector Machine (SVM) classifier is trained off-line to discriminate between object and background. Tracking is carried out by estimating transformation parameters that maximise the SVM score. This idea was extended by Williams et al. [19] who provided a probabilistic formulation allowing to propagate observation distributions over time. A mapping from image space to transformation parameter space is learned from seed images. These methods are quite robust, however they rely on the appearance variation being fully encoded within the classifier, which is not updated once the tracker is running. Collins et al. [3] proposed to update a classifier by selecting a set of discriminative features in each frame. It is assumed that the object was correctly located and that the surrounding area belongs to the background. The idea of updating a classifier based on AdaBoost using discriminative feature selection was introduced by Grabner and Bischof [4]. In [5] they show tracking over large appearance variation. However, neither of these methods [3, 5] maintains an explicit object model to prevent drift or adaptation to outliers.

Tracking using subspace models Subspace models have been used to model object appearance. The idea is that the images of a particular object lie on a lower dimensional manifold. The eigenspace tracking approach was introduced by Black et al. in [2]. Since then the idea has been extended to handling appearance changes. Several methods have

been proposed to learn an eigenspace representation and incrementally update it over time [6, 13, 17]. Lee and Kriegman adapt a generic appearance model to a specific one given a number of images [10]. Appearance is modelled by a union of linear manifolds and the model is updated by incrementally updating their eigenbases. None of these papers explicitly addresses outlier handling, although robust error norms may be able to handle some cases of partial occlusion.

Outlier handling Jepson et al. [8] use an adaptive appearance model where each pixel intensity is modelled by a three-component mixture. One of the components ('lost') is used to handle outliers caused for example by occlusion. Williams et al. [18] compute an outlier mask using a Markov Random Field with Ising prior in order to increase the tolerance of a foreground/background classifier to occlusions. The estimates are quite accurate, however this method of outlier handling adds significant computation overhead. Outlier handling is also being investigated in contexts other than tracking. Leonardis and Bischof [11] modified the eigenspace approach for recognition to handle partial occlusions. Instead of computing the PCA coefficients by projecting the complete image, the coefficients are found using only a subset of image points. Several subsets of points are hypothesized and tested using the backprojection error. Oh et al. [15] subdivide a face image into regions and construct separate eigenmodels for each. The occluded regions are found by separately computing the distance to the nearest input training sample. Recognition is performed on the occlusion free regions.

3 Discriminative Feature Selection Using A Local Generative Model

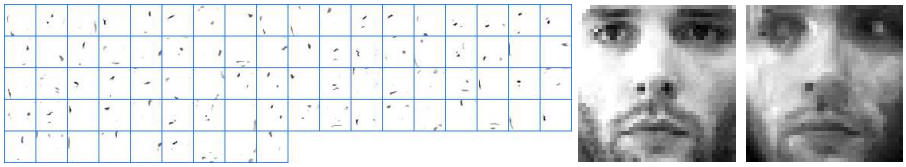
In this section we first review the principles of online boosting and local non-negative matrix factorization. We then show how to combine these two techniques to improve an adaptive tracking algorithm in the case of local occlusion.

3.1 Online tracking with AdaBoost

Online boosting for tracking has recently been introduced by Grabner et al. [4, 5]. The principle is to locate the object by maximizing the score of a boosted classifier at every time step. Following the localization step the classifier itself is updated with online boosting. The target estimate is used as a new positive example and surrounding regions as negative examples. The classifier, called a strong classifier, is built by a linear combination of a number of weak classifiers, where each weak classifier corresponds to evaluating a single feature. The features are chosen from a global feature pool by the online feature selection process based on their classification performance so far. With each update the algorithm uses the new training sample to compute features and classification weights with which to compute the updated strong classifier. For a further details, see [4, 5].

3.2 Local non-negative matrix factorization

Non-negative matrix factorization (NMF) is a method for finding a lower dimensional representation of data. Given a non-negative data matrix $\mathbf{X}$, NMF finds an approximate



(a) (b) (c)

Figure 1: **Local non-negative factorization on face images.** This figure shows **(a)** basis images found by LNMF. Note that the positive entries are sparse and localized. The basis images are used to approximate an input image **(b)**, the result is shown in **(c)**.

factorization $\mathbf{X} = \mathbf{BH}$ into non-negative matrices $\mathbf{B}$ and $\mathbf{H}$, i.e. their elements must be equal to or greater than zero [16]. Lee and Seung compared NMF to principal component analysis (PCA) and vector quantization (VQ), and showed that these can be written as factorizations with different constraints [9]. NMF is able to learn a parts-based representation while PCA and VQ both learn holistic representations which in the case of PCA can also contain negative entries. Several extensions of the original NMF algorithm have been proposed, in particular versions that impose a sparseness constraint on the matrix $\mathbf{H}$ [7, 12]. In this section we follow the exposition of Li et al. [12], who suggested an algorithm for local non-negative matrix factorization (LNMF) of images: Writing a set of N_T images into an $n \times N_T$ matrix $\mathbf{X}$, so that each column consists of the n pixel values, LNMF factorizes this matrix into an $n \times m$ matrix $\mathbf{B}$ containing a set of $m < n$ basis images, and an $m \times N_T$ coefficient matrix $\mathbf{H}$, such that $\mathbf{X} \approx \mathbf{BH}$. Additionally, LNMF imposes the following three constraints: (i) Maximum sparsity in $\mathbf{H}$, (ii) maximum expressiveness of $\mathbf{B}$, and (iii) maximum orthogonality of $\mathbf{B}$. A locally optimal solution is found by iteratively updating $\mathbf{B}$ and $\mathbf{H}$. See [12] for details. Once the subspace images are found, an image represented as an n-vector $\mathbf{x}$, is projected into the space by $\mathbf{h} = \mathbf{B}^{+}\mathbf{x}$, where $\mathbf{B}^{+}$ denotes the pseudo-inverse of $\mathbf{B}$. Figure 1 shows basis images found by LNMF (the columns of $\mathbf{B}$) as well as an example image where these basis images are used for approximating an input image.

3.3 Feature selection using a local generative model

The orthogonality of the basis images found with LNMF implies that the value of a pixel x_i in the decomposition of an image $\mathbf{x}$ is determined solely by the positive value in one of the basis images at the corresponding location i, and the corresponding weight, that is

$$x_i \approx \sum_j h_j b_{ij} \approx h_k b_{ik} \quad , k \in \{0, ..., m\}. \tag{1}$$

From the training images we obtain distributions $p^{fg}(h_j)$ for the weights of each dimension j of the subspace. We model the weight distribution in each dimension j with a Gaussian mixture which is used to determine the foreground likelihood. Thus we have a means of determining how well a new image locally matches the appearance model and we can create a foreground likelihood map for a new input image. If the likelihood is below a given threshold value, the corresponding regions are treated as outliers. In practice, LNMF factorization can result in more than one disconnected component in a basis

Algorithm 1 Online feature selection using a local generative model

Input: New image as n-vector $\mathbf{x}$, projection matrix $\mathbf{B}^+$ (computed off-line), coefficient inlier distributions $p^{fg}(h_j)$ (computed off-line), threshold values θ_{fg}, θ_2.

1. Compute LNMF coefficients $\mathbf{h} = \mathbf{B}^+\mathbf{x}$.
2. Initialise outlier map as n-vector $\mathbf{c} = 0$.
3. for each coefficient h_j
   if $p^{fg}(h_j) < \theta_{fg}$
      set $c_i = 1$ for $\{i \,|\, b_{ij} > 0\}$
4. Remove small connected components in $\mathbf{c}$.
5. for each feature f_k in classifier feature pool
   if $\sum_{i \in support(f_k)} c_i > \theta_2$
      replace f_k with new feature in non-occluded region
6. Update classifier using the online AdaBoost algorithm [5].

image, with some components being very small (see Fig. 1a). Under the assumption that the true region of outliers in an image is approximated by the union of basis image components, we threshold the binary image on component size to remove small components. This information is used to guide the online feature selection: Local features whose support region overlaps more than a threshold value with outlier regions are excluded from the classifier. Thus the algorithm only considers features in regions consistent with the generative model, and the discriminative classifier avoids locking on to occluded or background regions. Additionally, we cache classifiers in cases that are not occluded and use them to regain lock after the target has been occluded significantly. Algorithm 1 details the adapted online learning algorithm.

4 Experimental results

This section presents experimental results to validate our algorithm. In all experiments the appearance model was created by resampling the training images to 40x40 pixels, and factorizing with a subspace dimension of 81 in order to achieve sufficient localization. The LNMF model is learned from training sequences containing approximately 700 frames. Once the subspace basis is computed tracking is executed in real-time (around 50 ms per update). The strong classifier uses 50 selectors and the feature pool contains 250 weak classifiers. These are the same settings as in [5], however we only use local rectangle features ('Haar-like features') to demonstrate the improved feature selection. All experiments were carried out on a standard 3.4 GHz PC with 2GB RAM.

4.1 Detecting occlusions

As a proof of concept we show the distribution of coefficients h_j for one particular basis image over a sequence with significant occlusion. Figure 2 shows the results. One can observe a shift in the distribution during the time of occlusion. This shows that occluded regions can be detected by a change in the weight values of the basis images with positive values in those regions.

Figure 2: **Shift of coefficient distribution under occlusion.** *The histogram (left) shows the distribution of weights for one of the LNMF basis functions. There is a shift during the occluded case. The images show from left to right: basis image (white=zero, dark=positive), an unoccluded example, an example with occlusion intersecting the positive support region of the basis image.*

Figure 3: **Online feature selection under occlusion.** *The model helps to avoid erroneous feature selection in the occluded regions. Row 1 shows sample frames. Row 2 shows regions detected as occluded (red) and support regions of features used in the classifier (highlighted).*

We perform the adapted online feature selection algorithm (see Alg. 1) on a number of test sequences. Figure 3 shows sample frames in the top row and the algorithm output below. The occlusion map is shown in red and the support area of features currently used in the classifier is highlighted in white. We can see there is agreement between the occlusion map and the actual occlusion. Further, the feature selection is restricted to those areas labelled as not occluded. Figure 4 shows an example sequence with a large pose change. In such cases the target image can contain background regions. The outlier detection prevents feature selection in these regions.

4.2 Synthetic occlusion

We take a single test image, and create a test sequence by adding fixed size, randomly positioned black squares to simulate occlusion. Figure 5 shows the accuracy of the occlu-

Figure 4: **Feature selection under pose change.** *Large pose changes can lead to large regions of background being inside the target window. By labelling these as outliers, we avoid adaptation to the background. Row 1 shows input frames. Row 2 shows regions detected as occluded (red) and support regions of features (highlighted).*

sion detection with varying occlusion size from 5×5 to 155×155 pixels within a window of size 160×160 pixels. We measure the true positive rate and false positive rate in terms of pixel classification rate for each frame in a test sequence of random fixed-size occlusions and take the average to plot a point on the curve. Additional points are generated by running test sequences of differing occlusion size. The detection rate increases with the size of the occluded region. However, larger occlusions lead to an increase in false detections. This is due to the fact that there is no perfect alignment between occluded regions and support regions of basis images.

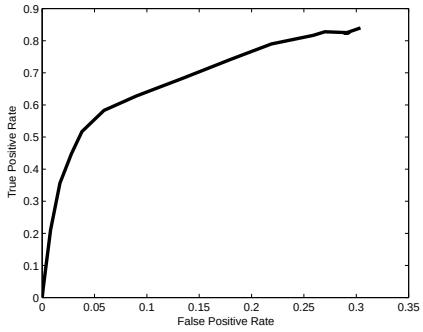


Figure 5: **Accuracy for different sizes of occlusion.** *Occlusion detection accuracy against occlusion size (increasing from 5×5 bottom left to 155×155 top right). Occlusion detection increases with size of occluded region but with the consequence of an increased false detection rate.*

Figure 6: **Real-time tracking with occlusion.** *Row 1 shows tracking results with feature selection using online boosting. The flexibility of the classifier leads to adaptation to the occluding object. Row 2 shows corresponding results with the new algorithm.*

Figure 7: **Real-time tracking during occlusion.** *Tracking continues during partial occlusion. During occlusion new features are allocated in the visible regions allowing tracking to continue. When the occluding object is removed the whole region is again used for feature selection.*

4.3 Tracking results

We run the tracker on test sequences containing heavy occlusion, both without and with the adapted online feature selection. The original tracker adapts to the appearance of the occluding object, and starts to track this. With our adapted feature selection the tracker is able to note the occlusion and stop feature selection in the occluded areas, regaining lock when the target re-appears. Figure 6 shows sample frames from the tracking sequence: row 1 shows the original tracker, row 2 shows the tracker using the results from our adapted online feature selection. Figure 7 shows continued tracking during partial occlusion. During occlusion new features are allocated in the visible regions. When the occluding object is removed the whole region is again used for feature selection.

Figure 8 shows tracking results on a publicly available sequence [8]. Row 1 shows the tracker using online boosting. The sequence was tracked successfully in [5] using a larger variety of features. Note that when using rectangle features only we observe a small shift in the target region estimate. Row 2 shows the result of our proposed algorithm. The occlusion of the face by the hand is detected correctly and tracking continues successfully.

Figure 8: **Real-time tracking with occlusion.** *Tracking results for the public 'Dudek' sequence [8]. Row 1 shows sample tracking results without using our adapted online feature selection. Row 2 shows corresponding results with the new algorithm. Regions detected as occluded are coloured white. Note that in contrast to [5] we use rectangle features only.*

5 Summary and conclusions

In this paper we have shown how a localized generative appearance model can be used by an online learning algorithm to focus feature selection on regions in an image which maintain a good proximity to the target object's appearance. We have demonstrated how this approach can improve the robustness of an adaptive tracker to occlusions while maintaining real-time performance. To our knowledge this is the first time that local non-negative matrix factorization has been employed within an object tracking context.

It can be noted that the outlier detection is dependent on the regions of positive value in the basis images, which we have no explicit control over in LNMF. There is also a trade-off between adaptiveness and outlier detection: A training sequence which shows little variation will result in good outlier region detection due to the smaller variance of the weight values. However, adaptiveness will be limited due to the tighter bounds on the appearance model. Conversely, a training sequence with large appearance variation will allow a lot more adaptiveness, but be less capable of detecting outlier regions. Future work will focus on a more thorough evaluation of the tracker, and increasing the flexibility of the appearance model.

References

[1] S. Avidan. Support vector tracking. *Trans. Pattern Analysis and Machine Intelligence*, 26(8):1064–1072, 2004.

[2] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. Journal of Computer Vision*, 26(1):63–84, 1998.

[3] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *Trans. Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.

[4] H. Grabner and H. Bischof. Online boosting and vision. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 260–267, 2006.

[5] H. Grabner, M. Grabner, and H. Bischof. Realtime tracking via online boosting. In *Proc. British Machine Vision Conference*, volume 1, pages 47–56, 2006.

[6] J. Ho, K. Lee, M. Yang, and D. Kriegman. Visual tracking using learned linear subspaces. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 782–789, 2004.

[7] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Machine Learning Research*, pages 1457–1469, 2004.

[8] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *Trans. Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, 2003.

[9] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[10] K.-C. Lee and D. Kriegman. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 852–859, 2005.

[11] A. Leonardis and H. Bischof. Dealing with occlusions in the eigenspace approach. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 453–458, 1996.

[12] S. Li, X. Hou, and H. Zhang. Learning spatially localized, parts-based representation, 2001.

[13] J. Lim, D. Ross, R. Lin, and M. Yang. Incremental learning for visual tracking. In *Advances in Neural Information Processing Systems*, pages 793–800, 2005.

[14] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *Trans. Pattern Analysis and Machine Intelligence*, 26:810–815, 2004.

[15] H. J. Oh, K. M. Lee, S. U. Lee, and C.-H. Yim. Occlusion invariant face recognition using selective LNMF basis images. In *Proc. Asian Conf. on Computer Vision*, pages 120–129, 2006.

[16] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.

[17] D. Ross, J. Lim, and M. Yang. Adaptive probabilistic visual tracking with incremental subspace update. In *Proc. Europ. Conf. on Computer Vision*, volume 2, pages 470–482, 2004.

[18] O. Williams, A. Blake, and R. Cipolla. The variational ising classifier (VIC) algorithm for coherently contaminated data. *Advances in Neural Information Processing Systems*, pages 1497–1504, 2004.

[19] O. Williams, A. Blake, and R. Cipolla. Sparse Bayesian learning for efficient visual tracking. *Trans. Pattern Analysis and Machine Intelligence*, 27:1292–1304, 2005.