Dictionary Learning for a Sparse Appearance Model in Visual Tracking

Sylvain Rousseau, Pierre Chainais, Christelle Garnier

DICTIONARY LEARNING FOR A SPARSE APPEARANCE MODEL IN VISUAL TRACKING

Sylvain Rousseau[1], *Pierre Chainais*[1,2]

[1] École Centrale de Lille,
CRIStAL, UMR CNRS 9189,
[2] INRIA Lille-Nord Europe, SequeL
Villeneuve d'Ascq, France
sylvain.rousseau@ec-lille.fr
pierre.chainais@ec-lille.fr

Christelle Garnier

Institut Mines-Telecom / Telecom Lille
CRIStAL, UMR CNRS 9189,
Villeneuve d'Ascq, France
christelle.garnier@telecom-lille.fr

ABSTRACT

This paper presents a novel approach to visual object tracking based on particle filtering. The tracked object is modelled by a sparse representation provided by dictionary learning. Such an approach permits to describe the target by a model of reduced dimension. The likelihood of a candidate region is built on a similarity measure between the sparse representations of a set of patches (at known positions) in the dictionary learnt from the reference template. Experimental validation is performed on various video sequences and shows the robustness of the proposed approach.

Index Terms— dictionary learning, sparse coding, particle filtering, object tracking

1. INTRODUCTION

Tracking objects in a video sequence is an essential task in computer vision. Among the many tracking methods (see [1]), particle filters are currently widely used. The performance strongly depends on the appearance model used to characterise the target object and on the likelihood function which measures the similarity between the candidate region and the reference object. Recently new models based on sparse representation, initially introduced in [2], have been successfully applied in visual tracking [3]. A global sparse appearance model is proposed in [4]. The dictionary is composed of two parts: target templates used to represent the entire object and trivial templates introduced to account for noisy pixels. A candidate is then sparsely represented with a few atoms of the complete dictionary. The likelihood is related to the reconstruction error in the target dictionary. The discriminative ability of the method can be improved by adding background templates in the dictionary [5]. To better handle partial occlusions, a local sparse appearance model based on a dictionary of patches sampled from target templates [6] or learnt from these patches [7] is more appropriate. The principle is to extract patches from a candidate and to compute their sparse encoding in the dictionary. The representation coefficients or sparse codes are then exploited as appearance features. The eventual step, referred to as the pooling step, is to extract meaningful coefficients from these appearance features. The likelihood is based on the similarity between the candidate and the object features vectors.

Several pooling operators have been considered. The average pooling [8] averages the sparse codes to build up the final feature vector. The max pooling method [7] keeps the maximum absolute value of the sparse code coefficients. Both methods are imprecise because they lose the location of patches and thus the spatial structure of the object. To overcome this loss, an alignment pooling method is developed in [9]. It preserves the spatial information, but it imposes constraints on the dictionary construction. It is composed of patches extracted from target templates according to a fixed spatial structure that can not change over time.

In this paper we propose to combine the principle of alignment pooling with dictionary learning for sparse representation of the target object patches. Then tracking is performed in a space of reduced dimension by comparing sparse codes of patches in the candidate region with sparse codes of patches extracted from the target template at known locations. The paper is organised as follows. Section 2 recalls related previous works. Section 3 presents the proposed sparse coding model. In Section 4, the tracking algorithm based on particle filtering is described. Section 5 presents performance results obtained on public datasets and Section 6 concludes.

2. RELATED WORK

We first recall the alignment pooling approach [9]. The target is described by several templates. For simplicity, we consider only one target template T, typically a rectangle comprising M overlapping square patches of size c^2. A set of $N \leq M$ patches is extracted from the template at predefined

Thanks to the BNPSI ANR project no ANR-13-BS-03-0006-01, the ARCIR REPAR project and to the Fondation École Centrale Lille for funding.

locations according to a spatial layout. The matrix $\boldsymbol{L_p} = [\boldsymbol{p}_1, \dots, \boldsymbol{p}_N] \in \mathbb{R}^{c^2 \times N}$, formed by the concatenation of the vectorised patches $(\boldsymbol{p}_i)_{i=1}^N$ is directly used as the dictionary. Then a candidate region C is modelled as follows. The candidate is first resized to match the target template's size. A set of N patches $(\boldsymbol{q}_i)_{i=1}^N$ is sampled at the *same locations* as in the target. The sparse codes $\boldsymbol{u}_i$ of each of these patches in the dictionary $\boldsymbol{L_p}$ of size N are obtained by solving the following ℓ_1-constrained minimisation problem

$$\arg\min_{\boldsymbol{u}_i} \left\| \boldsymbol{q}_i - \boldsymbol{L_p} \boldsymbol{u}_i \right\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{u}_i\|_1 \leq \rho, \tag{1}$$

where ρ controls the sparsity of the code $\boldsymbol{u}_i \in \mathbb{R}^N$. Each patch $\boldsymbol{q}_i$ is then approximated by a sparse combination of the patches $(\boldsymbol{p}_i)_{i=1}^N$ weighted by $\boldsymbol{u}_i$. The alignment pooling of $\boldsymbol{u}_i$ consists in selecting the contribution of patch $\boldsymbol{p}_i$ at the same location as patch $\boldsymbol{q}_i$ which is the i-th coefficient u_{ii} of $\boldsymbol{u}_i$. Consequently, if the $\boldsymbol{u}_i$'s are the columns of the square matrix $\boldsymbol{U} \in \mathbb{R}^{N \times N}$, the pooled coefficients are located on the diagonal. The log-likelihood of candidate C with respect to template T is defined as the trace of matrix U

$$\mathcal{L}(C, T) = \operatorname{Tr} \boldsymbol{U} = \sum_{i=1}^{N} u_{ii}. \tag{2}$$

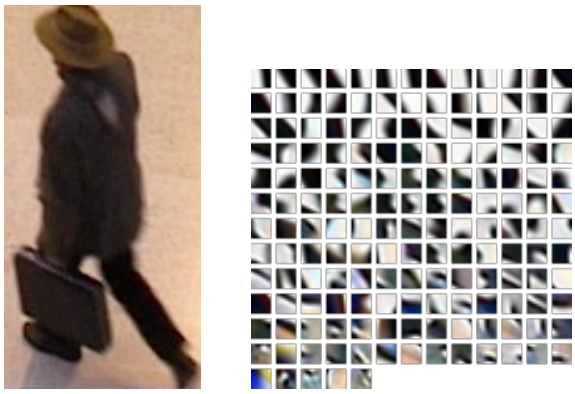
For comparison, the max and average pooling operators respectively compute the maximum and average of each $\boldsymbol{u}_i$, before summing the results. The alignment pooling is more accurate because it takes into account the locations of extracted patches. However the major drawback is the lack of flexibility. To increase the number of patches or change their locations for better performance, the dictionary has to be completely redefined. The size of the dictionary is necessarily the number N of extracted patches so that if one wants to take all available patches into account, the size of the dictionary becomes large since $N = M$. Moreover, as shown on Fig. 2, that method may suffer from a lack of robustness.

3. PROPOSED APPROACH

To overcome these limitations, we propose to learn a dictionary $\boldsymbol{D_p} \in \mathbb{R}^{c^2 \times K}$ of fixed size $K \ll M$ from a set of patches $\boldsymbol{L_p} = [\boldsymbol{p}_1, \dots, \boldsymbol{p}_N] \in \mathbb{R}^{c^2 \times N}$ extracted from template T to obtain an adapted representation of the data. This is carried out by solving the minimisation problem

$$\arg\min_{\boldsymbol{D_p}, \boldsymbol{Z}} \left\| \boldsymbol{L_p} - \boldsymbol{D_p} \boldsymbol{Z} \right\|_F^2 \quad \text{s.t.} \quad \|\boldsymbol{z}_i\|_1 \leq \rho, \ \forall i, \tag{3}$$

where $\boldsymbol{z}_i$ is the i-th column of the matrix $\boldsymbol{Z} \in \mathbb{R}^{K \times N}$ and $\|\cdot\|_F$ denotes the Frobenius norm. Dictionary learning is performed on centered patches while average values are processed apart. Consequently, $\boldsymbol{z}_i$ is the sparse code of the patch $\boldsymbol{p}_i$ with respect to the dictionary $\boldsymbol{D_p}$. The dimension reduction performed by dictionary learning imposes no limit



(a) Target template (b) Dictionary $\boldsymbol{D_p}$

Fig. 1: Atoms of a learnt dictionary by order of contributions in the image.

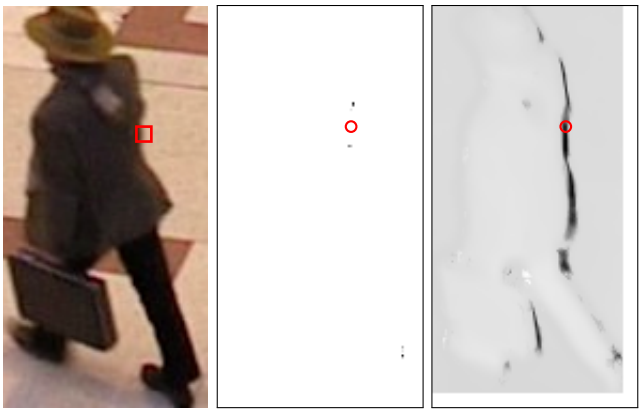
on the number N. We can afford to use all the overlapping patches ($N = M$) of size c^2 within the template to learn the dictionary of size $K \ll M$. Figure 1 illustrates the atoms of the dictionary $\boldsymbol{D_p}$ of size $K = 161$ learnt over all the $N = 16109$ patches of size 8×8 extracted from the target template image in Fig. 1a. To compare, the alignment pooling has to solve the minimisation problem (1) with a dictionary of size 16109 whereas the proposed method uses a dictionary $\boldsymbol{D_p}$ that is 100 times smaller. To describe a candidate region, n patches $(\boldsymbol{q}_i)_{i \in J}$ are extracted at locations defined by some set $J \subset \{1, \dots N\}$ and $\#J = n$. We compute their sparse codes $\boldsymbol{V} = [\boldsymbol{v}_i]_{i \in J} \in \mathbb{R}^{K \times n}$ in $\boldsymbol{D_p}$ by solving

$$\arg\min_{\boldsymbol{v}_i} \left\| \boldsymbol{q}_i - \boldsymbol{D_p} \boldsymbol{v}_i \right\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{v}_i\|_1 \leq \rho. \tag{4}$$

Compared to [9] where $n = N$, the choice of the number of patches n and their locations is completely free. To build the log-likelihood of a candidate region C, we compare the sparse representations using $\boldsymbol{D_p}$ of these n patches from C and T at the same locations defined by J. This comparison is quantified by matched filtering which provides optimal SNR for pattern detection. Thus, the log-likelihood of a candidate set of n sparse codes $\boldsymbol{V}$ with respect to the template sparse codes $\boldsymbol{Z_J} = [\boldsymbol{z}_i]_{i \in J}$ is defined as a cross-correlation:

$$\frac{1}{\|\boldsymbol{Z_J}\|_F} \operatorname{Tr} \boldsymbol{Z_J^T} \boldsymbol{V}. \tag{5}$$

This log-likelihood optimally detects the presence of the template patches represented by $\boldsymbol{Z_J}$ at known positions. This approach can be interpreted as an alignment pooling strategy since the *trace* operation pools the individual alignment scores $\boldsymbol{z}_i^T \boldsymbol{v}_i$ of the set of n patches (which all 'vote' for the same target position). Note that the similarity proposed in [9] can be seen as a special case of (5) where $\boldsymbol{D_p} = \boldsymbol{L_p}$ (no dictionary learning) and $\boldsymbol{Z_J} = \boldsymbol{I}$ (trivial coding). Last remains to take into account the average values of patches denoted by



(a) (b) (c)

Fig. 2: (a) A patch $\boldsymbol{q}_{i_0}$ taken from a resized candidate. (b) Reshaped sparse code $\boldsymbol{u}_{i_0}$. (c) Cross-correlation $\boldsymbol{Z}_{\boldsymbol{J}}^T \boldsymbol{v}_{i_0}$

$\bar{p}_i$ and $\bar{q}_i$: large differences of average gray levels $|\bar{p}_i - \bar{q}_i|$ should be penalised. We propose to add a term $-\lambda|\bar{p}_i - \bar{q}_i|^2$ to (5) to finally get the log-likelihood

$$\mathcal{L}_J(C,T) = \frac{1}{\|\boldsymbol{Z}_{\boldsymbol{J}}\|_F} \operatorname{Tr} \boldsymbol{Z}_{\boldsymbol{J}}^T \boldsymbol{V} - \lambda \sum_{i \in J} |\bar{p}_i - \bar{q}_i|^2 , \qquad (6)$$

where $\lambda > 0$ is some weighting parameter (default is 1).

Figure 2 compares the proposed approach to the alignment pooling method. A patch $\boldsymbol{q}_{i_0}$ is selected in a resized candidate region of Fig. 2a. Figures 2b and 2c show the similarity maps between $\boldsymbol{q}_{i_0}$ and all the patches $(\boldsymbol{p}_i)_{i \in J}$ with $J = \{1, \ldots, M\}$ extracted from T. Without dictionary learning, see Fig. 2b, this similarity map is directly given by the sparse code $\boldsymbol{u}_{i_0}$ of the patch $\boldsymbol{q}_{i_0}$ in $\boldsymbol{L}_{\boldsymbol{p}}$. With the proposed approach at Fig. 2c, it is given by $\boldsymbol{Z}_{\boldsymbol{J}}^T \boldsymbol{v}_{i_0}$ where $\boldsymbol{v}_{i_0}$ is the sparse code of $\boldsymbol{q}_{i_0}$ in $\boldsymbol{D}_{\boldsymbol{p}}$. The alignment pooling consists in selecting the coefficient of the sparse code corresponding to the contribution of the patch $\boldsymbol{p}_{i_0}$ at the same location as $\boldsymbol{q}_{i_0}$ (red circles in Fig. 2b and 2c). With a standard alignment pooling, little matching is found between the patches $\boldsymbol{p}_{i_0}$ and $\boldsymbol{q}_{i_0}$. In the set of the template patches used as the dictionary, a lot of atoms (patches) are similar and the atoms selected by the sparse decomposition do not necessarily include the aligned patch $\boldsymbol{p}_{i_0}$. This effect can be reduced by considering templates from different frames. Our approach gives better matching results because the learnt dictionary is of reduced dimension and the template patches $(\boldsymbol{p}_i)_{i \in J}$ are described by several atoms of $\boldsymbol{D}_{\boldsymbol{p}}$. A match is found every time $\boldsymbol{p}_i$, $i \in J$, shares atoms with $\boldsymbol{q}_{i_0}$. Figure 2c shows several matches in particular with the aligned patch.

4. TRACKING ALGORITHM

Object tracking is carried out by particle filtering [10]. Given a sequence of observations $\boldsymbol{y}_{1:k} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ up to time k,

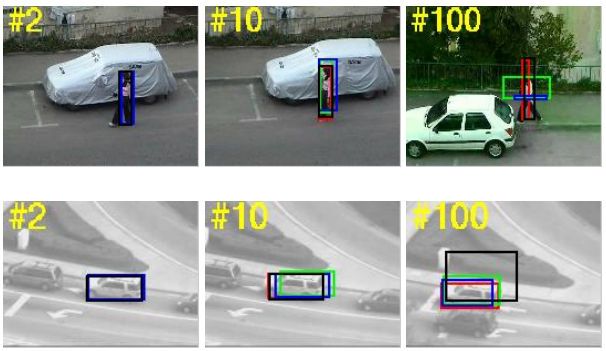


Fig. 3: Qualitative tracking results on the sequences `Woman` and `Suv` with different pooling strategies: proposed in red, maximum pooling in green, average pooling in blue and alignment pooling in black.

the aim is to estimate the object state $\boldsymbol{x}_k$ from the posterior density $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ which is recursively approximated with a set of N_p weighted particles $\{\boldsymbol{x}_k^{(m)}, w_k^{(m)}\}_{i=1}^{N_p}$: $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) \approx \sum_{i=1}^{N_p} w_k^{(m)} \cdot \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(m)})$. The target object is represented by a bounding window. The state vector is defined as $\boldsymbol{x}_k = [\boldsymbol{c}_k, \boldsymbol{d}_k]^T$ where $\boldsymbol{c}_k = \left[c_k^x, c_k^y\right]$ is the position of the top left corner and $\boldsymbol{d}_k = \left[d_k^x, d_k^y\right]$ is the size of the window. To obtain the set of particles at time k from the previous particles $\{\boldsymbol{x}_{k-1}^{(m)}, w_{k-1}^{(m)}\}_{i=1}^{N_p}$, the particle filter algorithm includes two major steps: prediction and update. During prediction, the samples $\boldsymbol{x}_k^{(m)}$ are propagated according to the dynamic model using the prior density $p(\boldsymbol{x}_k^{(m)}|\boldsymbol{x}_{k-1}^{(m)})$. As often, we assume that $\boldsymbol{x}_k$ evolves as a Gaussian random walk: $\boldsymbol{x}_k|\boldsymbol{x}_{k-1} \sim \mathcal{N}(\boldsymbol{x}_{k-1}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma}$ is a diagonal covariance matrix which defines the uncertainty region around the previous state. During the second stage, the weights $w_k^{(m)}$ are updated according to the recursive expression: $w_k^{(m)} = w_{k-1}^{(m)} \cdot p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(m)})$ where the observation likelihood $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ measures the matching between the observation and the state. We define the likelihood directly from expression (6) as

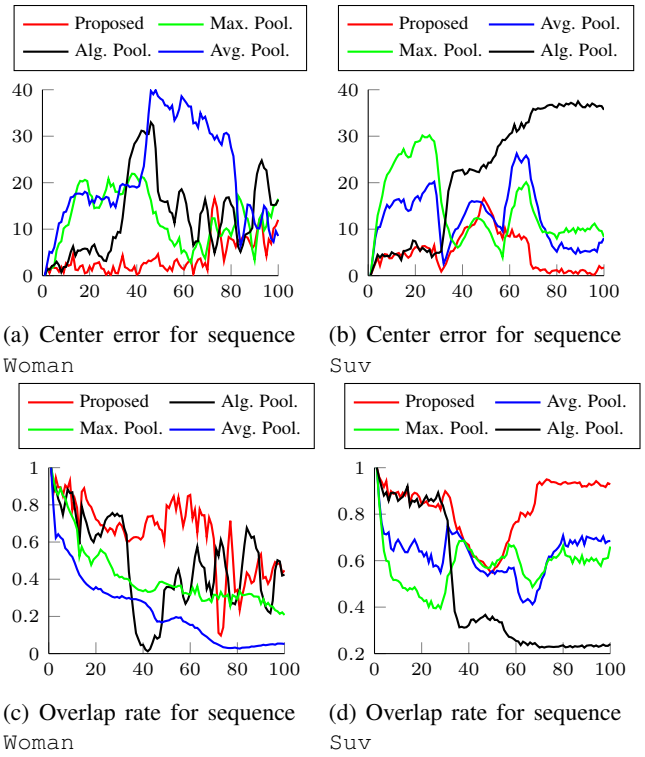
$$p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(m)}) \propto \exp\left\{\mu \cdot \mathcal{L}_J(C,T)\right\}, \qquad (7)$$

where C is the candidate region identified by $\boldsymbol{x}_k^{(m)}$ and μ is a tuning parameter. Finally, after weight normalisation and if necessary, particle resampling, the object state is obtained by MMSE (Minimum Mean Square Error) estimation

$$\widehat{\boldsymbol{x}_k} = \mathbb{E}\left[\boldsymbol{x}_k|\boldsymbol{y}_{1:k}\right] = \sum_{m=1}^{N_p} w_k^{(m)} \boldsymbol{x}_k^{(m)}. \qquad (8)$$

5. EXPERIMENTAL VALIDATION

In this section, we validate our approach by comparing it to other pooling strategies. The optimisation problems (1), (3)



(a) Center error for sequence `Woman`

(b) Center error for sequence `Suv`

(c) Overlap rate for sequence `Woman`

(d) Overlap rate for sequence `Suv`

Fig. 4: Quantitative tracking results on the sequences `Woman` and `Suv`

and (4) are solved using the Spams Matlab toolbox[1] [11] with $\rho = 0.25$. J is set to a fixed grid of step 8. For particle filtering, $N_p = 600$ particles, $\mu = 4.6$ and $\mathbf{\Sigma} = \text{diag}\,(20, 20, 4, 4)$. Concerning the proposed method, the size of the dictionary is set to $K = 64$.

Four pooling strategies are considered: alignment pooling (without dictionary learning) and maximum, average or alignment pooling (proposed method) with dictionary learning. They are compared in the first 100 frames of the sequences `Suv` and `Woman`. For a fair comparison, all the trackers adopt the same dynamical model, only the observation likelihood differs. Figure 3 shows the qualitative tracking results obtained on a few images of the sequences. These sequences present challenging scenarios with partial occlusion and illumination changes. Figure 4 shows the quantitative tracking results. The first row depicts the center location error (in pixels) between the ground truth bounding box and the estimated bounding box. The second row depicts the overlap rate [12] which is the ratio between the area of the intersection of both bounding boxes and the area of the union.

The maximum and average pooling give the worst performance in the first frames of both videos, because they use a similarity between patches regardless of their locations. However this lack of precision is compensated by a good robustness, and the performances are maintained or even can improve along the sequences. On the opposite, the alignment pooling is more accurate at the beginning of the sequences, because its takes into account the spatial information and a similarity is found between aligned patches. But the tracking results degrade along the videos. As shown in Figure 2, it can miss the matching between patches that are similar. For example on the `Suv` sequence, the target partially exits the image around frame 30. When the target comes back, the maximum and average pooling can find it whereas the alignment pooling fails. Along both sequences, the best tracking performance is achieved with our approach. The proposed cross-correlation based similarity is both accurate and robust. As the maximum and average pooling, our tracker can retrieve the target after a partial exit and as the alignment pooling, it preserves the spatial information.

[1] `http://spams-devel.gforge.inria.fr/`

	$R(\alpha = 0.5)$			AUC		
	Max	Alg	Prop	Max	Alg	Prop
Bolt	0.05	**0.09**	0.03	0.05	**0.14**	0.04
CarScale	0.14	**1.00**	0.98	0.23	0.74	**0.74**
David3	0.04	0.24	**0.29**	0.05	0.21	**0.22**
FaceOcc1	0.22	0.18	**1.00**	0.29	0.27	**0.82**
FleetFace	0.29	0.55	**0.62**	0.40	**0.49**	0.48
Freeman4	0.03	0.01	**0.03**	0.02	0.08	**0.12**
Singer1	0.34	0.59	**0.61**	0.29	0.57	**0.65**

Table 1: Success rate $R(\alpha = 0.5)$ and Area Under the Curve (AUC) $\int R(\alpha) d\alpha$ for the proposed method, max pooling and alignment pooling.

Table 1 gathers results obtained on a set of publicly available video sequences [13]. It compares success rates $R(\alpha)$ that are the proportions of frames for which the overlap rate is $\geq \alpha$. To this aim, we compare the values of $R(\alpha = 0.5)$ and, in the spirit of the usual AUC, the Area Under the Curve $\{\alpha \in (0, 1), R(\alpha)\}$ which is a more general indicator. The proposed method either equals or outperforms other methods.

6. CONCLUSIONS

In this paper we propose a sparse representation of patches in a learnt dictionary to model the appearance of an object. Then tracking is performed by a matched filter on the sparse codes of the patches extracted from the candidate. This method generalises the alignment pooling method and improves on other pooling strategies with learnt dictionaries such that the max or average pooling. Numerical results show that it compares favourably with previous works.

7. REFERENCES

[1] Hanxuan Yang, Ling Shao, Feng Zheng, Liang Wang, and Zhan Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, 2011.

[2] Bruno A Olshausen and David J Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *NATURE*, vol. 381, pp. 13, 1996.

[3] Shengping Zhang, Hongxun Yao, Xin Sun, and Xiusheng Lu, "Sparse coding based visual tracking: Review and experimental comparison," *Pattern Recognition*, vol. 46, no. 7, pp. 1772–1788, 2013.

[4] Xue Mei and Haibin Ling, "Robust visual tracking using ℓ_1 minimization," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2009, pp. 1436–1443.

[5] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang, "Robust object tracking via sparsity-based collaborative model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 1838–1845.

[6] Baiyang Liu, Junzhou Huang, Lin Yang, and Casimir Kulikowsk, "Robust tracking using local sparse appearance model and k-selection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 1313–1320.

[7] Qing Wang, Feng Chen, Jimei Yang, Wenli Xu, and Ming-Hsuan Yang, "Transferring visual prior for online object tracking," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3296–3305, 2012.

[8] Shengping Zhang, Hongxun Yao, and Shaohui Liu, "Robust visual tracking using feature-based visual attention," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 1150–1153.

[9] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang, "Visual tracking via adaptive structural local sparse appearance model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 1822–1829.

[10] Michael Isard and Andrew Blake, "Condensation–conditional density propagation for visual tracking," *International journal of computer vision*, vol. 29, no. 1, pp. 5–28, 1998.

[11] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[13] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 2411–2418.