OBJECT TRACKING WITH ADAPTIVE ELASTIC NET REGRESSION

Shunli Zhang, Weiwei Xing

School of Software Engineering
Beijing Jiaotong University

ABSTRACT

Recently, various regression based tracking methods have achieved great success. However, in most of these methods, all of the extracted features are made use of to represent the object without feature selection. In this paper, we propose a novel tracking method based on elastic net regression with adaptive weights. On one hand, tracking is formulated as an elastic net regression problem which can not only make full use of the spatial information, but also automatically select features to alleviate the influence of the unstable or inaccurate points. On the other hand, the weights of the ℓ_1-norm and ℓ_2-norm regularization in the regression model are adaptively adjusted to better improve the performance. Experimental results in the benchmark dataset demonstrate that the proposed adaptive elastic net regression based tracking method can achieve desirable tracking performance.

Index Terms— Object tracking, elastic net regression, adaptive weight

1. INTRODUCTION

As one of the hot research topics in computer vision, object tracking has wide application in many fields, e.g. video surveillance, motion analysis. However, tracking faces several different factors, such as heavy occlusion, deformation, complex background, etc., which make it difficult to realize robust tracking [1].

Appearance model plays an important role in tracking. Commonly, the traditional appearance model can be divided into two types: generative model [2, 3, 4, 5, 6] and discriminative model [7, 8, 9, 10, 11, 12]. Recently, some regression based discriminative models have achieved great development and obtained state-of-the-art tracking performance. For example, Hare et al. [13] propose the Struck method based on structured output regression, where the structured output prediction is used to avoid the intermediate classification step. Henriques et al. [14] formulate tracking as a correlation filtering problem, where correlation filter is constructed based on a ridge regression model. Zhang et al. [15] present the hybrid support vector machine based tracking method, where a support vector regression model is built based on the neighboring samples of the target. Different from the traditional binary classification based discriminative model, the regression model has some obvious advantages. e.g. the objectives of regression and the tracking task are always consistent and the background information can be fully exploited. However, most of the regression based tracking methods use all of the extracted features. Each feature is assigned to equal weight and the importance of the features is not considered. In other words, the features are not selected.

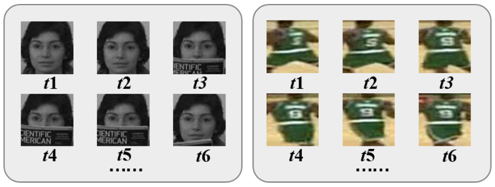


Fig. 1. Appearance changes caused by occlusion (left) and deformation (right).

During the tracking process, we can observe that the shape of the target may deform or the target has pose changes. Besides, the target may be occluded by another object (Fig. 1). These factors indicate that some parts of the target are not stable which may affect the performance of the appearance model. The occlusion will also contaminate the training samples, using which will degrade the accuracy of the model. By selecting the stable and accurate parts of the samples to build the appearance model, the effect of the deformation and occlusion may be alleviated and the robustness will be improved.

In this paper, we formulate tracking as an adaptive elastic net regression problem. Elastic net is a regression technique which contains both the ℓ_1-norm and ℓ_2-norm regularization. It can be regarded as the combination of lasso and ridge regression, thus it retains the advantages of the above two regression models. Elastic net has been used in many domains and achieved successful applications [16, 17]. Formulating tracking as an elastic net regression problem can bring two benefits. On one hand, the elastic net maintains the advantage of the regression model, which can makes use of more information of the background. On the other hand, the elastic net can adaptively select stable features for training, which

This work is supported by the National Natural Science Foundation of China (Grant No. 61601021, No. 61370128), the Project funded by China Postdoctoral Science Foundation, the Program for New Century Excellent Talents in University (NCET-13-0659) and the Fundamental Research Funds for the Central Universities (2016RC015).

978-1-5090-2175-8/17/$31.00 ©2017 IEEE

can lead to more accurate appearance model. Furthermore, we present an adaptive strategy to determine the weights of the ℓ_1-norm and ℓ_2-norm terms automatically. Experimental results demonstrate that the proposed method can achieve comparable tracking results to many state-of-the-art methods.

2. TRACKING WITH ELASTIC NET REGRESSION

2.1. Appearance model with adaptive elastic net regression

2.1.1. Formulation

Although different regression strategies, e.g., ridge regression and support vector regression, have been used to construct the regression appearance model, the regularization term about the weight vector $\mathbf{w}$ often adopts the ℓ_2-norm, which does not pay enough attention to the robustness of the features. Hereby, we formulate tracking as an elastic net regression problem. Elastic net can adaptively select the most stable features to learn the function, which can make a trade-off between the ℓ_1-norm and ℓ_2-norm. Therefore, we employ elastic net to select the most robust and stable features for appearance representation, reducing the influence of appearance changes caused by deformation, occlusion and other factors.

Concretely, assume the training sample set is X, and its element corresponding to a sample is $\mathbf{x}_i$ with the regression value $\mathbf{y}_i$. The linear regression function can be defined as $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, where both the training sample $\mathbf{x}_i$ and the regression value $\mathbf{y}_i$ are centered by subtracting the corresponding mean value. Then the optimization problem of the elastic net regularized regression can be represented as follows

$$\min_{\mathbf{w}} \sum_i \frac{1}{2}\|\mathbf{w}^T\mathbf{x}_i - \mathbf{y}_i\|_2^2 + \lambda(\alpha_1\|\mathbf{w}\|_1 + \alpha_2\|\mathbf{w}\|_2^2), \quad (1)$$

where λ denotes the trade-off parameter between the errors and the regularization, α_1, α_2 control the ratio of the ℓ_1-norm and ℓ_2-norm regularization and $\alpha_1 + \alpha_2 = 1$.

Further, we regard $\alpha = [\alpha_1, \alpha_2]^T$ in Eqn. 1 as the parameter to be optimized as well. Since the stable features of the object in different frames of a video sequence may be different, and the objects in different video sequences are different as well, it is beneficial to adjust the trade-off parameter adaptively. After that, Eqn. 1 becomes

$$\min_{\mathbf{w},\alpha} \sum_i \frac{1}{2}\|\mathbf{w}^T\mathbf{x}_i - \mathbf{y}_i\|_2^2 + \lambda(\alpha_1\|\mathbf{w}\|_1 + \alpha_2\|\mathbf{w}\|_2^2) + \frac{\rho}{2}\|\alpha\|_2^2, \quad (2)$$

where ρ is a trade-off parameter as well. A new regularization term $\|\alpha\|_2^2$ is added to control the balance of the ℓ_1-norm and ℓ_2-norm. By solving the optimization problem in Eqn. 2, the appearance model can be represented by the optimal $\mathbf{w}$ and α.

2.1.2. Optimization

Since there are two different parameters $\mathbf{w}$ and α, the optimization problem of Eqn. 2 can be optimized by a two-stage iteration algorithm. First, by fixing α, the optimization problem in Eqn. 2 degrades to Eqn. 1, which is a standard elastic net regression problem. There have been many off-the-shelf algorithms to solve the optimization problem in Eqn. 1, e.g. the LARS-EN algorithm. Hereby, we adopt the SPAMS Toolbox [18] to solve this problem.

In the second step, after obtaining $\mathbf{w}$, we fix it and solve the optimization problem w.r.t α

$$\min_{\alpha} \sum_i \lambda(\alpha_1\|\mathbf{w}\|_1 + \alpha_2\|\mathbf{w}\|_2^2) + \frac{\rho}{2}\|\alpha\|_2^2. \quad (3)$$

This is a quadratic optimization problem, which can be directly solved by Matlab. Then these two steps can continue iteratively and we can get the optimal $\mathbf{w}$ and α.

2.1.3. Preparing the samples

We densely sample the training samples in grid based on the sliding window strategy. Assume that the location of the target center is (x_0, y_0), the width and the height of the target region are w and h respectively, and the normalization size of the sample is $N_s \times N_s$. Besides, we denote the coordinates of the normalized target sample as $(0, 0)$. If we densely sample the training samples according to the normalized target in pixel with step size d, the training radius in normalization domain is N_s, i.e., we hope the training samples can cover the target region with different degree. Therefore, the location of the normalized sample $\mathbf{x}_i$ in normalization domain is (p_i, q_i), where $p_i, q_i \in [-N_s, N_s]$. Correspondingly, the location of $\mathbf{x}_i$ before normalization is $(p_i w/N_s, q_i h/N_s)$, which can be obtained by reverse mapping. By varying p_i and q_i in $[-N_s, N_s]$, we can obtain all the training samples with different overlaps to the target.

The samples selected in grid are stored in the neighbor buffer $\mathbf{B}_n$, which provides the compact spatial constraints. In our method, each sample in $\mathbf{B}_n$ corresponds to a specific position. Besides, we also build another training sample buffer, the target buffer $\mathbf{B}_t$ with depth D, which is filled by the previous tracking results and affords temporal constraint.

2.1.4. Determining the regression values

To make the objectives of the regression and tracking consistent, hereby, we adopt Gaussian function as the regression function, in which the largest regression function value corresponds to the location of the target. Since we have normalized the training samples into fixed size, the Gaussian function can be implemented in terms of the normalization size to avoid the effect of the size of different targets,

$$\mathbf{y}_i = \exp(-(\bar{x}(i)^2 + \bar{y}(i)^2)/\sigma^2), \quad (4)$$

where $\bar{x}(i)$ and $\bar{y}(i)$ denote the normalized relative horizontal and vertical locations of the sample $\mathbf{x}_i$, respectively, and σ^2 denotes the variance of the Gaussian function. With the prepared training samples and the determined regression values, we can train the elastic net regression model by optimizing the problem in Eqn. 1 and acquire the model parameter $\mathbf{w}$.

2.2. Searching strategy

With the built appearance model, we can complete tracking frame by frame. Hereby, we utilize a simple but effective sliding-window sampling strategy to search for the optimal tracking result. Denote the obtained location of the target in frame $t-1$ as l_{t-1} and a candidate sample $\mathbf{x}_j$ in frame t as $l(\mathbf{x}_j)$. We slide the sampling window around l_{t-1} in frame t, and obtain a series of candidate samples. If l_{t-1} and $l(\mathbf{x}_j)$ satisfy $\|l(\mathbf{x}_j) - l_{t-1}\|_2 < R_s$, $\mathbf{x}_j$ will be selected as a candidate sample. By normalizing the candidate samples into fixed size and extracting features, the regression value $f(\mathbf{x}_j)$ for a candidate sample $\mathbf{x}_j$ can by calculated and the confidence score of $\mathbf{x}_j$ can be represented as

$$conf(\mathbf{x}_j) = \exp(-(f(\mathbf{x}_j) - \mathbf{y}_{max})^2). \tag{5}$$

Then, the optimal candidate sample can be determined by

$$\mathbf{x}_{opt} = \arg\max_{\mathbf{x}_j} conf(\mathbf{x}_j). \tag{6}$$

2.3. Update scheme

After obtaining the tracking result $\mathbf{x}_{opt}$, we update the elastic net model to fit the changes of the appearance. Hereby, we adopt a retraining strategy to update the elastic net model.

Since we have divided the samples into the target part $\mathbf{B}_t$ and neighbor part $\mathbf{B}_n$, we update the samples in $\mathbf{B}_t$ and $\mathbf{B}_n$, respectively. For $\mathbf{B}_t$, we update the samples by the First-In-First-Out (FIFO) rule, i.e., the sample which came into the target buffer earliest will be replaced by $\mathbf{x}_{opt}$. For $\mathbf{B}_n$, since there is only one sample corresponding to each position, we select N_u samples which least coincide the existing model for update. Based on the location of $\mathbf{x}_{opt}$, we first select the samples around $\mathbf{x}_{opt}$ according to the introduction in Section. 2.1.3. Then we measure how the new samples coincide the existing model by the regression model. Assume the new sample set is X' with element $\mathbf{x}_i$. Then the measure corresponding to $\mathbf{x}_i$ is $m_i = |\hat{\mathbf{y}}_i - \mathbf{y}_i|$, where $\hat{\mathbf{y}}_i$ is the prediction value for $\mathbf{x}_i$ by the elastic net model, and $\mathbf{y}_i$ is the predefined ideal regression value. A larger value of m_i indicates that the sample $\mathbf{x}_i$ less fit the current model. Therefore, we select the N_u samples with larger measure than the others to replace the samples in the same neighboring position in $\mathbf{B}_n$. With the updated samples, we can retrain the elastic net model by solving Eqn. 1. Based on the proposed update scheme, the regression model can be updated to accommodate the appearance changes as far as possible.

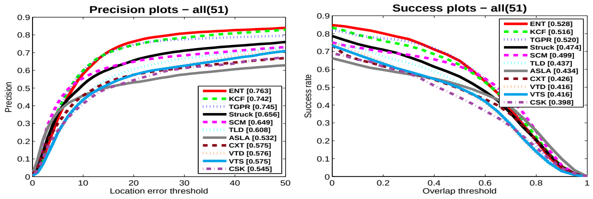


Fig. 2. Precision plots and success plots of ENT and the competing trackers on all 51 sequences. The values in the square brackets represent the precision at $Th_p = 20$ pixels on precision plots and the AUC on success plots, respectively.

Table 1. The comparison results of CLE (in pixel), VOR, Precison and SR of ENT and the competing trackers.

Trackers	Average CLE	Average VOR	Precision ($Th_p = 20$)	SR ($Th_s = 0.5$)
ENT	44.0	0.534	0.763	0.528
KCF	35.3	0.522	0.742	0.516
TGPR	45.3	0.526	0.745	0.520
Struck	50.5	0.477	0.656	0.559
SCM	54.1	0.505	0.649	0.616
TLD	48.1	0.440	0.608	0.521
ASLA	73.0	0.438	0.532	0.511
CXT	68.4	0.429	0.575	0.492
VTD	47.4	0.418	0.576	0.493
VTS	50.7	0.419	0.575	0.496
CSK	88.8	0.401	0.545	0.443
LSK	58.9	0.397	0.505	0.456
DFT	69.2	0.392	0.496	0.444

3. EXPERIMENTS

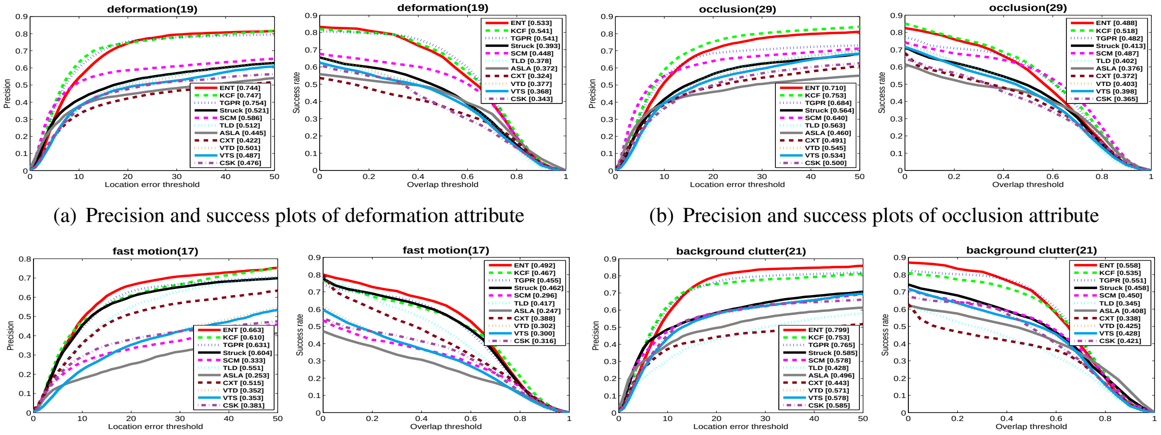
3.1. Initialization

The proposed adaptive elastic net based tracking method is represented as ENT, which is initialized as follows: HOG features with 5-pixel window size and 9 orientations are extracted for representation. The normalization size of the samples is 30×30. The training radius and searching radius are set to 30 pixels and 26 pixels. For the Gaussian function, $\sigma^2 = 0.01$. For the regularization parameters, α_1 is set as 0.5 for initialization, λ and ρ are set as 0.5 and 10 respectively. The ratio N_u/M for update is set as 0.05. All of the parameter are fixed for all sequences.

3.2. Comparison with state-of-the-art trackers

We compare the performance of the proposed ENT tracker with several state-of-the-art tracking methods in the benchmark dataset [1]. The competing trackers include TGPR [19], KCF [20], Struck [13], SCM [21], TLD [22], ASLA [23], CXT [24], VTD [25], VTS [26], CSK [14], LSK [27] and DFT [28]. We utilize several different criteria [1], including the average center location error (CLE), the average Pascal VOC overlap rate (VOR), the precision and the success rate (SR), the precision plots and success plots, and the area under the curve (AUC) score of the success plot for evaluation.

We first evaluate the overall performance of the ENT tracker and the competing trackers and show the comparison



(a) Precision and success plots of deformation attribute

(b) Precision and success plots of occlusion attribute

(c) Precision and success plots of fast motion attribute

(d) Precision and success plots of background clutter attribute

Fig. 3. Precision plots and success plots of ENT and the competing trackers on the sequences with different attributes.

results in Table. 1 and Fig. 2. From Table. 1 we can observe that, the CLE of our ENT tracker is 44.0 pixels, which is some larger than KCF, but smaller than the rest trackers. The VOR obtained by ENT is 0.534, which outperforms all the competing trackers. The precision at $Th_p = 20$ pixels of ENT is 0.763 and the SR at $Th_s = 0.5$ is 0.528, both of which are better than the competing trackers. Fig. 2 displays the precision plots and the success plots of ENT and the competing trackers, which indicates that ENT performs the best on both of these two plots.

We further compare the performance of ENT and the competing trackers in some representative conditions, including occlusion, deformation, fast motion, background clutter, etc. Fig. 3(a) displays the comparison results in the condition of deformation. We can observe that the precision at $Th_p = 20$ pixels of ENT is similar to KCF, and the AUC score is the second best result. Since the elastic net regression can automatically deal with the unstable features caused by deformation, ENT can get results as good as KCF and TGPR. Fig. 3(b) shows the comparison results on the sequences with occlusions. It can be seen that ENT ranks the second on both the precision at $Th_p = 20$ pixels and the SR at $Th_s = 0.5$. In our method, we improve the robustness of the tracker to occlusion by the adaptive elastic net regression, which can adaptively alleviate the effect of unstable features. The comparison results in the condition of fast motion are displayed in Fig. 3(c). We can observe that, the proposed ENT tracker outperforms the other trackers on both precision and AUC score. Because the rectangle used to represent the target often includes some background, using elastic net to train the model can effectively exclude the influence of the background changes caused by fast motion. Fig. 3(d) shows the comparison results in background clutter. We can observe that ENT achieves better results on both plots than the other trackers. The elastic net can reduce the effect of the background and the regression framework can increase the discriminability of the appearance model, which makes ENT robust to background clutter.

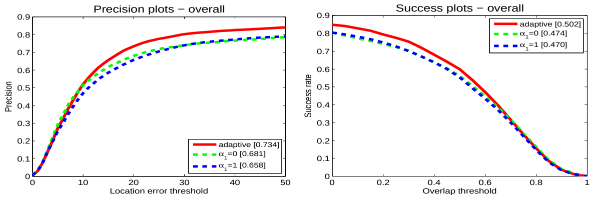


Fig. 4. Precision plots and success plots of complete ENT and the competing trackers with only a single norm.

3.3. Parameter analysis

In our method, α is taken as the optimization variable according to different frames and sequences. Hereby, we investigate the role of the elastic net regression formulation by constructing another two trackers. The tracker with $\alpha_1 = 1$ only uses the lasso formulation, and the tracker with $\alpha_1 = 0$ adopts the ridge regression. It can be seen that each of these two trackers only employs a single norm. The overall performance comparison of ENT and the competing trackers is demonstrated in Fig. 4. It can be observed that the standard ENT with both two norms significantly outperforms the other two trackers with only a single norm. Moreover, the tracker with $\alpha_1 = 0$ performs some better than the tracker with $\alpha_1 = 1$.

4. CONCLUSION

In this paper, we employ the elastic net regression to construct a novel tracking method. By formulating tracking as an elastic net regression problem, we can not only take the advantage of the regression model, but also utilize elastic net to select the most stable features for representation. We evaluate the tracking method in the benchmark dataset and the experimental results show that the proposed method can outperform many other regression based tracking methods.

5. REFERENCES

[1] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *CVPR*. IEEE, 2013, pp. 2411–2418.

[2] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *CVPR*, 2006, pp. 798–805.

[3] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *TPAMI*, vol. 25, no. 5, pp. 564–577, 2003.

[4] Xue Mei and Haibin Ling, "Robust visual tracking using l1 minimization," in *ICCV*, 29 2009-oct. 2 2009, pp. 1436 –1443.

[5] D.A. Ross, J. Lim, R.S. Lin, and M.H. Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1, pp. 125–141, 2008.

[6] Hanxi Li, Chunhua Shen, and Qinfeng Shi, "Real-time visual tracking using compressive sensingdai," in *CVPR*, 2011, pp. 1305–1312.

[7] S. Avidan, "Support vector tracking," *TPAMI*, vol. 26, no. 8, pp. 1064–1072, 2004.

[8] S. Avidan, "Ensemble tracking," *TPAMI*, vol. 29, no. 2, pp. 261–271, 2007.

[9] B. Babenko, Ming-Hsuan Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *TPAMI*, vol. 33, no. 8, pp. 1619–1632, 2011.

[10] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *BMVC*, 2006, pp. 47–56.

[11] S. Zhang, X. Yu, Y. Sui, S. Zhao, and L. Zhang, "Object tracking with multi-view support vector machines," *TMM*, vol. 17, no. 3, pp. 265–278, March 2015.

[12] S. Zhang, Y. Sui, S. Zhao, and L. Zhang, "Graph regularized structured support vector machine for object tracking," *TCSVT*, 2015.

[13] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *ICCV*, 2011, pp. 263–270.

[14] J. F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *ECCV*, 2012, pp. 702–715.

[15] S. Zhang, Y. Sui, X. Yu, S. Zhao, and L. Zhang, "Hybrid support vector machines for robust object tracking," *Pattern Recognition*, vol. 48, no. 8, pp. 2474–2488, 2015.

[16] Hui Zou and Trevor Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[17] Christine De Mol, Ernesto De Vito, and Lorenzo Rosasco, "Elastic-net regularization in learning theory," *Journal of Complexity*, vol. 25, no. 2, pp. 201–230, 2009.

[18] J Mairal, F Bach, J Ponce, G Sapiro, and R Jenatton, "Spams: Sparse modeling software," *WILLOW, INRIA*, vol. 2, 2011.

[19] Jin Gao, Haibin Ling, Weiming Hu, and Junliang Xing, "Transfer learning based visual tracking with gaussian processes regression," in *European Conference on Computer Vision*. Springer, 2014, pp. 188–203.

[20] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *TPAMI*, 2015.

[21] W. Zhong, H. Lu, and M.H. Yang, "Robust object tracking via sparsity-based collaborative model," in *CVPR*, 2012, pp. 1838–1845.

[22] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *TPAMI*, vol. 34, no. 7, pp. 1409–1422, 2012.

[23] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang, "Visual tracking via adaptive structural local sparse appearance model," in *CVPR*, 2012, pp. 1822–1829.

[24] Thang Ba Dinh, Nam Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *CVPR*, June 2011, pp. 1177–1184.

[25] Junseok Kwon and Kyoung Mu Lee, "Visual tracking decomposition," in *CVPR*, 2010, pp. 1269–1276.

[26] Junseok Kwon and Kyoung Mu Lee, "Tracking by sampling trackers," in *ICCV*, 2011, pp. 1195–1202.

[27] Baiyang Liu, Junzhou Huang, Casimir Kulikowski, and Lin Yang, "Robust visual tracking using local sparse appearance model and k-selection," *TPAMI*, vol. 35, no. 12, pp. 2968–2981, 2013.

[28] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *CVPR*, June 2012, pp. 1910–1917.