VISUAL TRACKING USING LEARNED COLOR FEATURES

Ting Liu, Rahul Rama Varior, Gang Wang

School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

ABSTRACT

Robust object tracking is a challenging task in computer vision. Color features have been popularly used in visual tracking. However, most conventional color-based trackers either rely on luminance information or use simple color representations for image description. During the tracking sequences, the perceived color of the target may change because of the varying lighting conditions. In this paper, we learn the color patterns offline from pixels sampled from images across different camera views. In the new color feature space, the proposed tracking method performs robustly in various environment. The new color feature space is learned by learning a linear transformation and a dictionary to encode pixel values. To speedup the feature extraction, we use the marginal regression to calculate the sparse feature codes. Experimental results demonstrate that significant improvement can be achieved by using our learned color features, especially on the video sequences with complicated lighting conditions.

Index Terms— Visual tracking, color features, marginal regression, feature learning

1. INTRODUCTION

Visual object tracking is one of the most challenging problems in computer vision. It plays a crucial role in many applications, such as surveillance, human computer interaction and auto-control systems [1, 2]. Several factors, such as illumination variations, appearance deformation, and occlusions complicate the problem. In this paper we aim to enhance the performance of object tracking by learning color features which have the capability of handling complicated color appearances.

Color-based features have been proven to be an important cue for visual tracking. However, color measurements can vary significantly over an image sequence due to variations in illuminant, shadows, shading, specularities, camera and object geometry. For example, the perceived color of same parts for a target may appear to be different. Considering this observation and as validated from our experiments, using color features such as RGB, HSV or even more complicated color attributes representation may not be adequate to achieve an robust color representation[3–8]. A recent work [9] proposes a framework to address the illumination variation in person re-identification. Hence we propose a data driven framework that learns rich and robust color features from raw pixel values for visual tracking problems.

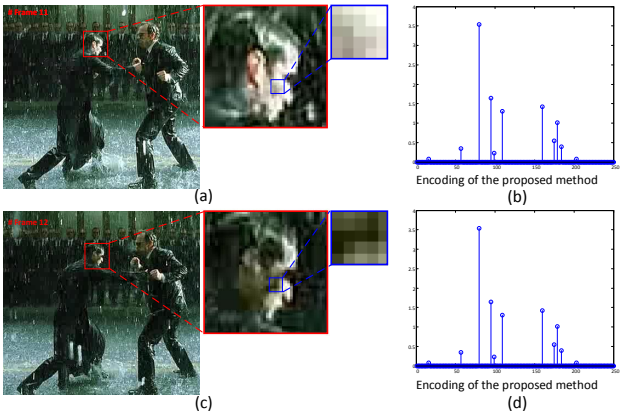

Fig. 1: (a)Sampled pixels (blue box) from one tracking frame, in which the target's face (red box) is under bright lighting condition. (b) Encoding of a randomly sampled pixel from this bright patch. (c) Sampled pixels (blue box) from one frame, in which the target's face (red box) is under dark lighting condition. (d) Encoding of a randomly sampled pixel from this dark patch. It can be seen from (a) and (c), the color for the corresponding sampled patches are not close to each other. (b) and (d) show the results of the encoding using our approach. Using our proposed algorithm, the obtained encoding values are very similar to each other for corresponding pixels in the tracking target.

Visual tracking is generally classified in two categories: generative and discriminative. The generative tracking methods adopt an appearance model to express the target observations. Some of the most popular generative tracking methods are incremental tracker[10], structural sparse tracker[11] and sampling Tracker [12]. Discriminative tracking methods address the tracking as a classification problem [13, 14]. The strategy of tracking is to search the target location, which optimally extracts the target from the background [15, 16]. Kalal et al. [14] train a binary classifier from labeled and unlabeled examples. Babenko et al. [17] propose a multiple instance

T. Liu and R. Rama Varior have equal contributions to this work.

learning algorithm for object tracking. Recently, Henriques et al. [18] explores a dense sampling strategy while showing that the process of taking subwindows in a frame induces circulant structure. Danelljan et al. [5] extend the KCF tracker with color attributes. Given the competitive performance and fast speed, we base our method on the KCF tracker.

In this paper, we aim to learn new color features for videos under various lighting environment for visual tracking. In the learned color space, the codes of the pixels are robust to various angles of view, shadows, shading, etc. An auto-encoder based framework is used to transform the RGB pixel to a higher dimensional space. Then the transformed pixel values are encoded using marginal regression technique to attain more discriminative sparse codes. Fig.1 shows the difference of original RGB pixels and the corresponding encoding with proposed method in a tracking sequence.

2. THE KCF TRACKER

We aim to learn color features to enhance the state-of-the-art tracking methods. Recently, the tracking system with the Kernelized Correlation Filter (KCF) [18] achieves very good performance. Hence, we integrate our feature learning method into this system. In this section, we briefly introduce the tracking system. Readers may refer to [18] for more details.

The classifier of KCF is trained using a single grayscale image patch X that is centred around the target. Taking advantage of the cyclic property and appropriate padding, KCF considers all cyclic shifts x_i as the training examples for the classifier. These examples are labelled with a Gaussian function correspondingly.

The goal of training is to find a function $f(z) = w^T z$ that minimizes the squared error over samples x_i and their regression targets y_i,

$$\min_w \sum_i |\langle \phi(x_i), w \rangle - y_i|^2 + \lambda \|w\|^2 \tag{1}$$

where ϕ is the mapping to the Hilbert space induced by the kernel κ, defining the inner product as $\langle \phi(x), \phi(x') \rangle = \kappa(x, x')$. The λ is a regularization parameter that controls overfitting.

After mapping the inputs of a linear problem to a nonlinear feature-space $\phi(x)$, the solution w can be expressed as $w = \sum_i \alpha_i \phi(x_i)$.

$$\hat{\alpha}^* = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \tag{2}$$

where the hat $\hat{}$ denotes the Discrete Fourier Transform of a vector. Because of the cyclic structure of the samples, $\hat{k}^{xx}$ is the kernel correlation of x with itself, in the Fourier domain.

In the detection step, a grayscale patch z is cropped out in the new frame. The detection scores are calculated as $\hat{f}(z) = (\hat{k}^{xz})^* \odot \hat{\alpha}$, where k^{xz} is the kernel correlation of x and z. Here x denotes the grayscale patch of the target appearance, which is learned over multiple frames. The target position in the new frame is then estimated by finding the translation that maximizes the score $f(z)$. Readers may refer to [18] for more details.

3. LEARNING COLOR FEATURES

As shown in Fig. 1, in the practical tracking tasks, the appearance of color changes during the sequences due to the lightning. It can be seen that the patches sampled appear to be of different colors. Previous color feature based tracking methods usually use RGB, HSV, color attributes or other color handcrafted features to represent the target objects. However, such features cannot capture the essential color information which should be robust to various lighting conditions, shadows, shading, specularities and object geometry often occurred in object tracking. Hence, we aim to learn a novel color space to solve the challenging problems in visual tracking.

3.1. Training Dataset

Since the color of an object could change significantly due to different views and illumination, a good tracker desires features which are robust to various lighting conditions. Therefore, we employ a learning model to learn color features from selected images to handle illumination changes of objects in visual tracking. Note that this is performed offline.

To train a robust color space, we need to collect training data under different lighting conditions. We use the CAVIAR4REID dataset [19] to train the system. It is a person re-id evaluation dataset which was extracted from the caviar dataset for evaluation of people tracking and detection algorithms. It is a dataset which contains a total of 72 pedestrians: 50 of them with two camera views and 22 with one. The patches are carefully selected so that sampled patches are distributed among different colors under various lighting environment.

3.2. Objective Formulation

To learn a robust color space, the feature codes of colors should be able to capture the essential color information which is robust through the tracking videos.

We use a linear auto-encoder [20] to discover the stable structures and patterns in the data and projects it into a robust color space. Through the auto-encoder, we enforce the encoded pixels of same color to have same codes.

Recently, sparse coding [21][22] has been found to improve the performance in classification, detection, tracking, etc. Hence, to further improve the performance, a sparse encoding is applied to the descriptors. We propose a color features learning framework to learn the transformation and dictionary simultaneously.

$$\min_{W,D,\alpha} \left\|[W^T(WX+b_1)+b_2]-X\right\|_2^2+\lambda\left\|W\right\|_2^2 \\ +\eta\left\|(WX+b_1)-D\alpha\right\|_2^2+\upsilon\|\alpha\|_1 \quad (3)$$

The first row of the formulation is the auto-encoder loss function; the second row is the loss function of sparse coding. In the formulation, α is the encoding of the same color under different lighting conditions; X denotes the input RGB values of randomly sampled pixels from patches extracted from images under different lighting environment; W is the linear transformation matrix that transforms each of the pixels into a new space; D is the sparse coding Dictionary to encode these pixels in a more discriminative space; the parameter η controls the trade-off between the two terms. b_1 and b_2 are the bias term. Because the training work is based on 3-dimensional RGB pixel values, the input X is constituted of sampled pixels. And the output α is the encoding of corresponding pixel.

3.3. Marginal regression for Sparse Coding

Due to the complexity of the tracking problem, the desired feature extraction procedure should be computationally efficient. The training time should be as short as possible to improve the efficiency of the whole system. In both of the training and feature extraction procedure, sparse coding is the main time consuming operation. In the past, several methods have been proposed to improve the speed of sparse coding, however it is still too slow for practical applications. Recently the Lasso [23, 24] is a popular tool for ℓ_1 optimization. Similar to other ℓ_1 optimization solutions, Lasso has complicated operations. The gradient descent algorithms and LARS algorithm [25] for Lasso require $O(p^3+np^2)$ operations. To overcome this limitation, in this section we show how marginal regression can be used to obtain sparse codes faster.

Consider a regression model, $Y = D\beta + z$, where $Y = (Y_1, Y_2, \cdots, Y_n)^T$, D is a $n \times p$ design matrix, coefficients $\beta = (\beta_1, \beta_2, \cdots, \beta_p)^T$ and $z = (z_1, z_2, \cdots, z_n)^T$ is the noise variable. For sparse coding, the main problem is variable selection: determining which components of β are nonzero.

Marginal regression [26] (also called correlation learning, simple thresholding [27], and sure screening [28]) is an efficient method for variable selection in which the outcome variable is regressed on each covariate separately and the resulting coefficient estimates are screened. To compute the marginal regression estimates for sparse coding, we begin by computing the marginal regression coefficients, assuming D has been standardized, we calculate:

$$\hat{\beta} = D^T Y \quad (4)$$

Then, we threshold $\hat{\beta}$ using a parameter $\tau > 0$:

$$\alpha_i = \begin{cases} \hat{\beta}_i & \left|\hat{\beta}_i\right| > \tau \\ 0 & otherwise \end{cases} \quad (5)$$

We sort these coefficient in terms of their absolute values, and select the top largest coefficients whose ℓ_1 norm is bounded by τ. The thresholding parameter τ is selected by cross validation.

This procedure requires just $O(np)$ operations. When p is much larger than n, marginal regression provides two orders of magnitude speedup over Lasso. This is a significant advantage of marginal regression because it is now tractable for much larger problems. [28] and [29] introduce more details about the comparison of marginal regression and Lasso.

3.4. Formulation Optimization

We alternatively optimize the objective function between α, D, W, b_1 and b_2. The L-BFGS gradient based optimization procedure is used to update these values. Initially, α is updated based on their gradients while keeping D, W, b_1 and b_2 fixed. Because it is difficult to solve the L1 norm in gradient based optimization. Instead of calculating $\|\alpha\|_1$, we use the marginal regression to constrain updated results of α. Given the pages limitation, we only show the partial derivative of the loss function w.r.t α:

$$\frac{\partial L}{\partial \alpha} = -2\eta \times D^T \times (WX + b_1 - D\alpha) \quad (6)$$

where L is Eq.3. When we calculate the α from the above function, Eq.5 is considered as the principle to select coefficients. Then D is updated based on the similar gradients of itself and keeping α, W, b_1 and b_2 fixed. Finally, keeping α and D fixed, W, b_1 and b_2 are updated together.

During tracking, for each pixel of the candidate window, we first use W and b_1 to transform it into the higher dimensional space and use the learned dictionary D to compute the sparse codes by marginal regression. When the size of a tracking candidate is $M \times N$, the dimensionality of the features for the candidate will be $M \times N \times d$ where d is the number of learned sparse coding dictionaries. Fig.1(b) and 1(d) show the representations obtained by using the color space learning framework. It can be seen that final encoding is very close to each other for corresponding pixels in the tracking target.

4. EXPERIMENTS

We evaluate our learned color features to demonstrate its robustness to complicated lighting conditions on five challenging tracking sequences. These videos are recorded in indoor and outdoor environments and have variations of illumination change, occlusion, etc. Besides the baseline tracker KCF [18], we also compared the proposed algorithm with another

nine state-of-the-art visual trackers: Frag[30], SMS [31], MIL [17], IVT [10], ASL [11], TLD [14], VTS [12],ADA [5]. All our experiments are performed using MATLAB R2012b on a 3.2 GHZ Intel Core i5 PC with 16 GB RAM. For fair comparison, we use the source codes provided by the authors. They were initialized using their default parameters.

To assess the performance of the proposed tracker, two criteria, the center location error as well as the overlap rate, are employed in our paper. A smaller average error or a bigger overlap rate means a more accurate result. Given the tracking result of each frame R_T and the corresponding ground truth R_G, we can get the overlap rate by the PASCAL VOC [32] criterion, $score = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)}$. Table 1 and 2 report the quantitative comparison results respectively.

From Table 1 and 2, we can see clearly that in most situations, our method performs better than other state-of-the-art methods. It proves that the learned color features make the tracker robust to different lighting conditions. The results of Matrix and Shaking sequences of VTS are directly quoted from reference [12].

We also plot the results of Frag, SMS, MIL, IVT, ASL, TLD, VTS, ADA and KCF trackers in visualization comparison. The results of visualization are shown in Fig.2. It can be seen that our proposed method performs very well on all these challenging sequences. When the appearance of target in Fig.2(a) changes significantly at frame 399, most of other trackers lose the face. In Fig.2(b), because of the heavy illumination changes, only the proposed method tracks the actor's face correctly through the whole sequence. Except for our method, VTS, ADA and KCF also propose well in sequence Shaking (Fig.2(c)). For the Coke sequence in Fig.2(d), after the heavy occlusion at frame 269, our method still tracks the can robustly. Although the skiing man moves quite fast in Fig.2(e), our method succeed to catch the essential color information of the target.

Table 1: Average center location error (in pixels). The bold font indicate the best performance.

Sequence	Frag	SMS	MIL	IVT	ASL	TLD	VTS	ADA	KCF	Ours
Matrix	78.5	59.9	55.0	64.4	65.1	57.2	12.0	79.2	63.6	**10.5**
Shaking	39.1	81.7	24.0	85.7	63.5	37.1	**5.0**	14.7	17.1	5.7
Trellis	59.5	36.8	21.5	29.7	7.5	31.0	24.3	20.6	18.8	**7.2**
Coke	54.8	81.9	46.7	82.9	60.1	25.0	62.4	30.7	18.6	**10.9**
Skiing	101.0	95.5	66.9	152.3	66.6	142.8	72.0	74.4	47.5	**6.0**

Table 2: Average overlap rate(%). The bold font indicate the best performance.

Sequence	Frag	SMS	MIL	IVT	ASL	TLD	VTS	ADA	KCF	Ours
Matrix	16	17	20	32	42	16	60	12	13	**65**
Shaking	18	21	45	23	21	39	70	59	58	**72**
Trellis	29	21	30	35	80	48	49	59	58	**86**
Coke	34	28	25	12	17	40	18	42	55	**62**
Skiing	13	16	16	28	19	27	20	18	16	**52**

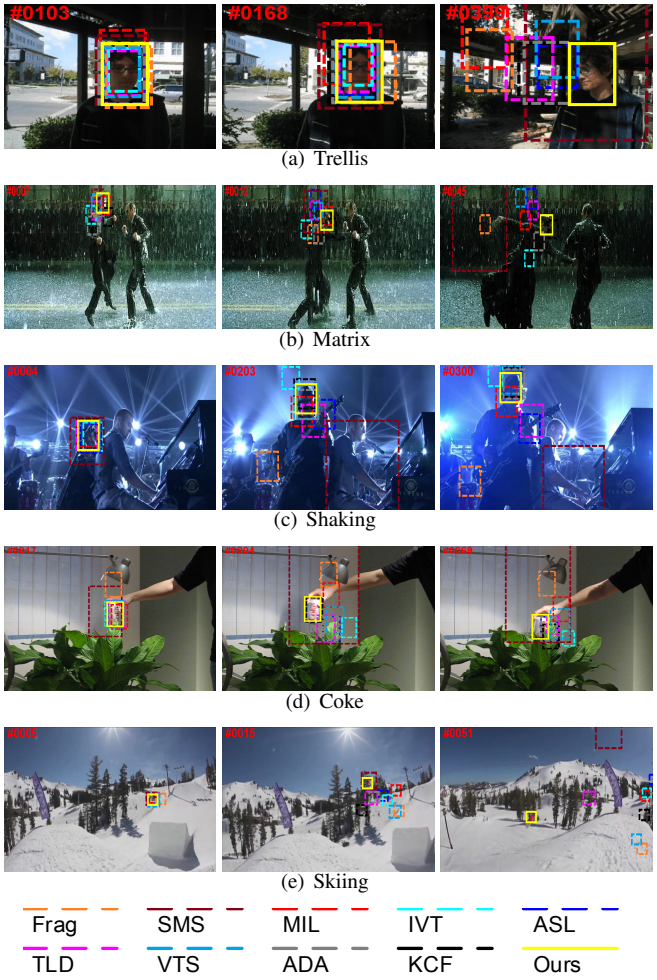

Fig. 2: Comparison of our approach with state-of-the-art trackers in challenging situations such as illumination variations. The example frames are from the Trellis, Matrix, Shaking, Coke and Skiing sequences respectively. The results of proposed tracker are indicated by yellow boxes.

5. CONCLUSION

In this paper, we propose a framework for learning robust and rich color features for KCF tracker. We learn the generic features from a person re-identification dataset and apply the learned features to challenging visual object tracking. Moreover, we propose a marginal regression based sparse coding method to speedup the training and feature extraction procedure. Experimental results demonstrate that our learned color features are robust to complicated lighting conditions and are able to improve the performance of the baseline tracker.

Acknowledgements: The research is supported by Ministry of Education (MOE) Tier 1 RG84/12, Ministry of Education (MOE) Tier 2 ARC28/14 and A*STAR Science and Engineering Research Council PSF1321202099.

References

[1] Alper Yilmaz, Omar Javed, and Mubarak Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, pp. 13, 2006.

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

[3] Katja Nummiaro, Esther Koller-Meier, and Luc Van Gool, "An adaptive color-based particle filter," *Image and vision computing*, vol. 21, no. 1, pp. 99–110, 2003.

[4] Patrick Pérez, Carine Hue, Jaco Vermaak, and Michel Gangnet, "Color-based probabilistic tracking," in *Computer visionłECCV 2002*, pp. 661–675. Springer, 2002.

[5] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014*. IEEE, 2014.

[6] Gang Wang, Derek Hoiem, and David Forsyth, "Learning image similarity from flickr groups using fast kernel machines," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2177–2188, 2012.

[7] Jiwen Lu, Gang Wang, and Pierre Moulin, "Human identity and gender recognition from gait sequences with arbitrary walking directions," *Information Forensics and Security, IEEE Transactions on*, vol. 9, no. 1, pp. 51–61, 2014.

[8] Gang Wang, David Forsyth, and Derek Hoiem, "Improved object categorization and detection using comparative object similarity," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 10, pp. 2442–2453, 2013.

[9] Rahul Rama Varior, Gang Wang, and Jiwen Lu, "Learning invariant color features for person re-identification," *CoRR*, vol. abs/1410.1035, 2014.

[10] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

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

[12] Junseok Kwon and Kyoung Mu Lee, "Tracking by sampling trackers," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1195–1202.

[13] Robert T Collins, Yanxi Liu, and Marius Leordeanu, "Online selection of discriminative tracking features," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1631–1643, 2005.

[14] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 49–56.

[15] Huiyu Zhou, Yuan Yuan, Yi Zhang, and Chunmei Shi, "Non-rigid object tracking in complex scenes," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 98–102, 2009.

[16] Jing Wen, Xinbo Gao, Yuan Yuan, Dacheng Tao, and Jie Li, "Incremental tensor biased discriminant analysis: A new color-based visual tracking method," *Neurocomputing*, vol. 73, no. 4, pp. 827–839, 2010.

[17] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie, "Visual tracking with online multiple instance learning," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 983–990.

[18] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *arXiv preprint arXiv:1404.7584*, 2014.

[19] Michele Stoppa Loris Bazzani Dong Seon Cheng, Marco Cristani and Vittorio Murino, "Custom pictorial structures for re-identification," in *British Machine Vision Conference (BMVC)*, 2011, p. 68.1C68.11.

[20] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[21] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1794–1801.

[22] Kai Yu, Tong Zhang, and Yihong Gong, "Nonlinear learning using local coordinate coding," in *Advances in neural information processing systems*, 2009, pp. 2223–2231.

[23] Robert Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[24] Scott Shaobing Chen, David L Donoho, and Michael A Saunders, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[25] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al., "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[26] Krishnakumar Balasubramanian, Kai Yu, and Guy Lebanon, "Smooth sparse coding via marginal regression for learning sparse representations," in *ICML (3)*, 2013, pp. 289–297.

[27] David L Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[28] Jianqing Fan and Jinchi Lv, "Sure independence screening for ultrahigh dimensional feature space," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 5, pp. 849–911, 2008.

[29] Christopher R Genovese, Jiashun Jin, Larry Wasserman, and Zhigang Yao, "A comparison of the lasso and marginal regression," *The Journal of Machine Learning Research*, vol. 98888, no. 1, pp. 2107–2143, 2012.

[30] Amit Adam, Ehud Rivlin, and Ilan Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. IEEE, 2006, vol. 1, pp. 798–805.

[31] Robert T Collins, "Mean-shift blob tracking through scale space," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. IEEE, 2003, vol. 2, pp. II–234.

[32] 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.