An Improved Struck Tracking Method Based on Fast Search Method

Ni Yunfeng
School of Electrical Engineering, Xi'an University of Science & Technology
Shaanxi Electric Power Research Institute, Xi'an
Xi'an, China

Xu Li
School of Electrical Engineering, Xi'an University of Science & Technology, Xi'an, 710054
,Xi'an, China
E-mail: niyunfeng@xust.edu.cn

Hou Ying
School of Electrical Engineering, Xi'an University of Science & Technology
Xi'an, China
E-mail: houying@xust.edu.cn

Abstract-Traditional methods such as adaptive tracking-by-detection approaches generate a set of samples and, depending on the type of learner, producing training labels. ,However, it is not clear how to best perform their sample step. Furthermore, the objective for the classifier (label prediction) is not explicitly coupled to the objective for the tracker (accurate estimation of object position.). Then a new method named ,Struck, it avoids these steps and operates directly on the tracking .output. Struck uses a kernelized structured output support vector machine (SVM), which is learned ,online to provide adaptive tracking. What's ,more, to allow for real-time application, it applies a budgeting mechanism which prevents the unbounded growth in the number of support vectors which would otherwise occur during tracking. However, this method does not run fast and may affect its real-time performance. To further improve its operating speed and simplify algorithm without reducing much ,accuracy, researchers introduce fast searching method to replace its original initial sampling and change some of its default .parameters. Just from the amount of ,calculation, the method can partly develop algorithm speed with good performance.

Keywords-Struck; SVM; Fast Searching Method; NTSS; Object Tracking

I. INTRODUCTION

Computer vision for tracking arbitrary objects is widely studied .recently. Visual object tracking is one of the core problems of computer vision, with wide-ranging applications including human-computer interaction, surveillance and augmented reality, to name just a few. For other areas of computer vision which aim to perform higher-level tasks such as scene understanding and action recognition, object tracking provides an essential component[1].

Tracking-by-detection [2], which treats the tracking problem as a detection task applied over time, has become particularly popular recently. There are mainly two factors for its popularity: one is due to the great deal of progress

© 2015. The authors - Published by Atlantis Press

made recently in object detection, with many of the ideas being directly transferable to tracking [2]. Another factor is that the classifiers used by these approaches are allowed to be trained online which provides a natural mechanism for adaptive tracking, for example [3,4,5].

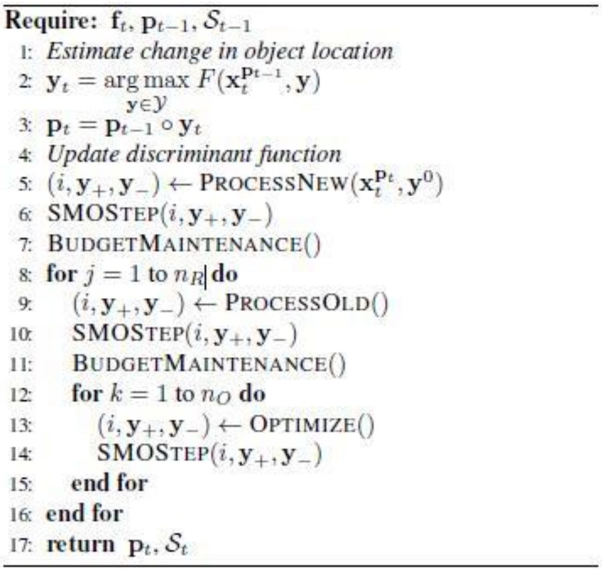
These algorithms separate the adaptation phase of the tracker into two distinct parts: one is the generation and labeling of samples; the other is the updating of the classifier[1]. However, this separation raises many issues, which may influence the robustness of the classifier due to poorly labeled samples. ,Firstly, it is necessary to design a strategy for generating and la- belling samples, and it is not clear how this progress should be done in a principled manner. Furthermore, the objective for the classifier (label prediction) is not explicitly coupled to the objective for the tracker (accurate estimation of object position.). Some of these kind adaptive methods mainly focus on improving tracking performance by increasing the robustness of the ,classifier, such as using robust loss functions [6], semi-supervised learning [7], and multiple-instance learning [3].

Different from these tracking-by-detection methods, struck[1] takes a different way and frame the overall tracking problem as one of structured output prediction, in which the task is to directly predict the change in object location between frames[1]. Not only does this method presents a novel and principled adaptive tracking via detection framework, which integrates the learning and tracking, but also avoids the need for updating strategies. Struck method makes use of online structured output SVM learning method proposed in [10, 11] and adapt it to the tracking problem, due to their good generalization ability, robustness to label noise, and flexibility in object representation through the use of kernels [8, 9]. To meet real-time ,operation, a budget maintenance step is involved since the number of support vectors of online learning with kernels increase largely with the amount of training data. Experiments illustrates that struck results in large performance gains over state-of-the-art tracking by detection approaches[1].

However, even adopting a budget, computational efficiency is not so .satisfactory. Our experiments regrettably turn out that under VS2010 software with openness vision library and eigen matrix library, the running time of each frame of the default video to be tracked will be more than 20 seconds. How to further improve its operating speed and simplify algorithm without reducing much accuracy reminds a big problem to be solved. Therefore, researchers hope to introduce fast searching method to replace some of its intermediate steps and change some of its default parameter to improve the efficiency of the algorithm.

II. Online structured output tracking

In the following section, researchers provide an overview of struck tracking method (as is shown in Fig.1) and main steps of algorithm, please tend to reference [1] for more .details.



```
Require: $\mathbf{f}_t, \mathbf{p}_{t-1}, \mathcal{S}_{t-1}$
Estimate change in object location
$\mathbf{y}_t = \arg\max_{\mathbf{y}\in\mathcal{Y}} F(\mathbf{x}_t^{\mathbf{p}_{t-1}}, \mathbf{y})$
$\mathbf{p}_t = \mathbf{p}_{t-1} \circ \mathbf{y}_t$
Update discriminant function
$(i, \mathbf{y}_+, \mathbf{y}_-) \leftarrow$ ProcessNew($\mathbf{x}_t^{\mathbf{p}_t}, \mathbf{y}^0$)
SMOStep($i, \mathbf{y}_+, \mathbf{y}_-$)
BudgetMaintenance()
for $j = 1$ to $n_R$ do
    $(i, \mathbf{y}_+, \mathbf{y}_-) \leftarrow$ ProcessOld()
    SMOStep($i, \mathbf{y}_+, \mathbf{y}_-$)
    BudgetMaintenance()
    for $k = 1$ to $n_O$ do
        $(i, \mathbf{y}_+, \mathbf{y}_-) \leftarrow$ Optimize()
        SMOStep($i, \mathbf{y}_+, \mathbf{y}_-$)
    end for
end for
return $\mathbf{p}_t, \mathcal{S}_t$
```

Figure 1. Struck: Structured Output Tracking

A. *Structured output SVM*

Struck proposes learning a prediction function $f : x \rightarrow y$, where x is the bounding box and y is the desired transformation of the target, to directly estimate the object transformation between frames. Here struck is applied in single target tracking. When struck initialize the first frame of the default video to locate target using haar features, it uses radials sample in the 2D translation case it was sufficient to sample from y on a polar grid, rather than considering every pixel offset. When the target is located, struck box it with a 30 x 30 rectangular search window x. Then set its midpoint as center, 30 pixel as radius, Divide the circle into $2\pi / nt$ parts and take $30 / nr$ points on each radius, as the sampling points are shown in fig.2(here researchers set $nt = 8$, $nr = 5$ In real program these parameters are bigger). Every new point will be the mid points of the new box x.

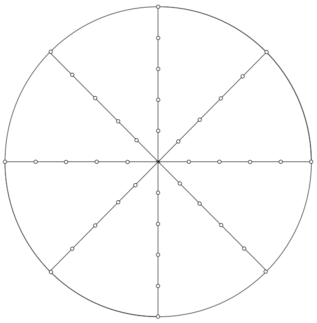


Figure 2. Initial sampling

Then output space is the space of all transformations y. In this approach, a labeled example is a pair (x,y), researchers learn f in a structured output SVM framework [12, 13], which introduces a discriminant function $F : x \times y \to R$ that can be used for prediction according to the step 2 in figure 1, Where t=1,...,T is the time, P_{t-1} is the position of the object at time t-1. A maximization step is performed to predict the best object transformation y_t and researchers find the new position of the target by $P_t = P_{t-1} \times y_t$ (step 3). The function F includes the label y explicitly that can be used into the learning .algorithm. A labeled example relative to the new tracker location $(x_t^{P_t}, y^0)$ is supplied to update the prediction function online. F measures the compatibility between (x,y) pairs and gives high scores to those which are well matched [1]. A form $F(x,y) = \langle w, \Phi(x,y) \rangle$, where $\Phi(x,y)$ is a joint kernel map (to be shown in section C), can be learned in a large-margin framework from a set of example pairs $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ by minimizing the convex objective function

$$\min_{w} \ \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi_i \quad s.t. \ \ \forall i : \xi_i \geq 0$$

$$\forall i, \forall y \neq y_i : \langle w, \delta\Phi_i(y) \rangle \geq \Delta(y_i, y) - \xi_i \qquad (1)$$

Where, $\delta\Phi_i(y) = \Phi(x_i, y_i) - \Phi(x_i, y)$. Struck uses a loss function to address the issue raised previously of all samples being treated equally, and it is based on bounding box overlap, and use

$$\Delta(y, \bar{y}) = 1 - s_{Pt}^{o}(y, \bar{y}) \qquad (2)$$

Where $s_{Pt}^{o}(y, \bar{y})$ is the overlap measurement.

B. *Online optimization*

To optimize (1) in an online setting, struck chooses the approach in [10,11]. Using standard Lagrangian duality techniques, (2) can be converted into its equivalent dual form

$$\max_{\alpha} \sum_{i, y \neq y_i} \Delta(y, y_i) \alpha_i^y - \frac{1}{2} \sum_{\substack{i, y \neq y_i \\ j, y \neq y_i}} \alpha_i^y \alpha_j^{\bar{y}} \langle \delta\Phi_i(y), \delta\Phi_j(\bar{y}) \rangle$$

$$s.t. \ \ \forall i, \forall y \neq y_i : \alpha_i^y \geq 0$$

$$\forall i : \sum_{y \neq y_i} \alpha_i^y \leq C \qquad (3)$$

Then set discriminant function as

$F(x,y) = \sum_{i, \bar{y} \neq y_i} \alpha_i^{\bar{y}} \langle \delta\Phi_i(\bar{y}), \Phi(x,y) \rangle$. As in the case of classification SVMs, it can be defined implicitly in terms of an appropriate joint kernel function (to be discussed in Section D).

$k(x, y, \bar{x}, \bar{y}) = \langle \Phi(x,y), \Phi(\bar{x}, \bar{y}) \rangle$. As in [2], by reparametrising (3) according to

$$\beta_i^y = \begin{cases} -\alpha_i^y & if \quad y \neq y_i \\ \sum_{\bar{y}\neq y_i} \alpha_i^{\bar{y}} & otherwise \end{cases}$$

Then the dual can be considerably simplified to

$$\max_{\beta} \; -\sum_{i,y}\Delta(y,y_i)\beta_i^y - \frac{1}{2}\sum_{i,y,j,\bar{y}}\beta_i^y\beta_j^{\bar{y}}\left\langle \Phi(x_i,y),\Phi(x_j,\bar{y})\right\rangle$$

$$s.t. \quad \forall i, \forall y : \beta_i^y \leq \delta(y,y_i)C$$

$$\forall i : \sum_y \beta_i^y = 0 \qquad (4)$$

Where $\delta(y,\bar{y}) = 1$ if $y = \bar{y}$ and 0 otherwise. Then researchers can note that discriminant function is simplified to $F(x,y) = \sum_{i,\bar{y}} \beta_i^{\bar{y}} \left\langle \Phi(x_i,\bar{y}),\Phi(x,y)\right\rangle$.

Then struck chooses pairs (x_i, y) for which $\beta_i^y \neq 0$

as support vectors and those x_i included in at least one support vectors as patterns. For a given support pattern x_i, only the support vector (x_i, y_i) will have $\beta_i^{y_i} > 0$

while any other support vectors (x_i, y) , $y \neq y_i$,will

have $\beta_i^y < 0$ .Struck refers to these as positive and negative support vectors respectively. The core step in the optimization algorithm of [10, 11] is an SMO-style step [14] (step 6 and 10) which improves (4) with respect to a pair of β_i^{y+} and β_i^{y-} and with the constraint $\sum_y \beta_i^y = 0$ coefficients must be modified by opposite amounts (for more details please tend to [1]). For a given $(i, y+, y-)$
are chosen to define the feasible search direction with respect to a single coefficient β_i^y is given by

$$= -\Delta(y, y_i) - F(x_i, y) \qquad (5\,)$$

Here struck possesses three different update steps: PROCESSNEW,PROCESSOLD and OPTIMIZE. PROCESSNEW and PROCESSOLD both have the ability to add new support vectors, which gives the learner the ability to perform sample selection during tracking and discover important background elements. The OPTIMIZE case only considers existing support vectors, so is a much less expensive operation. Following the suggestion in [11], struck schedule these update steps as is shown in Fig. 1.

During tracking, researchers maintain a set of support vectors S . For each $(x_i, y) \in S$, researchers store the coefficients β_i^y and gradients $g_i(y)$, which are both incrementally updated during an SMO step. If the SMO step results in a

β_i^y becoming 0, the corresponding support vector is removed from S .

C. *Incorporating a budget*

Since evaluating $F(x, y)$ requires evaluating kernel functions between (x, y) and each support vector, which results in both the computational and storage costs growing linearly with the number of support vectors. Therefore, struck proposes an approach for incorporating a budget into the algorithm. Similar to [15] to remove the support vector which results in the smallest change to the weight vector names w, as measured by $\|\Delta w\|^2$ but also considering the SMO step used during optimization and ensuring that the constrain $\sum_y \beta_i^y = 0$ remains satisfied. Each time the budget is exceeded struck removes the support vector resulting in the minimum.

D. *Kernel functions and image features*

SVM classification is more accurate and sings a structured output SVM framework provides great flexibility in how images are actually represented. As in [12], struck proposes using a restriction kernel, which uses the relative bounding box location y to crop a patch from a frame x_t^{poy}, and then applies a standard image kernel between pairs of such patches,

$$k_{xy}(x, y, \bar{x}, \bar{y}) = k(x^{poy}, \bar{x}^{\bar{p}o\bar{y}})$$

The use of kernels makes it straightforward to incorporate different image features into the approach, and in the experiments, researchers consider a number of examples. Researchers also investigate by using multiple kernels in order to combine different image features together. Here researchers set harr feature and Gauss kernel function as default configuration.

III. IMPROVEMENT BY FAST SEARCH METHOD

Our improvement plan is to replace Struck's original target initialization sampling method to block motion estimation algorithm thus simplify partial calculation. The common block motion estimation algorithms include full search method FS, three-step search TSS, new three-step search NTSS, four-step search FSS and diamond search DS. In this paper researchers choose NTSS as the improvement plan.

For the initial sampling part, adding original box, there are 41 sampling boxes to be analyzed for every 5 frames. As to default video, there are 501 frames which means researchers should totally calculate 4100 boxes let alone the following amount computation. Here researchers use NTSS to replace the original steps.

As is shown in Fig. 3, the first step researchers set the search window to 9 x 9 rectangular and sample the center 17 points for the following matching operation. If the first step searches the minimum SAD in the center of the adjacent points, here are two possibilities for the second step. According to the first step, if the minimum SAD point is near the center of 9 x 9 rectangular then the second step, the boxes whose center points are this17 points are what researchers need for the sampling, if the minimum SAD point is in the edge points of 9 x 9 rectangular, researchers set this point as the new 9 x 9 rectangular and go on the above two steps.

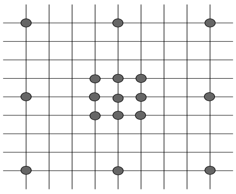


Figure 3. first search point of INTSS algorithm

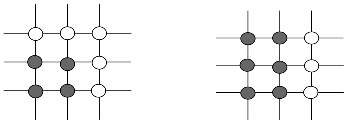


Figure 4. first search point of INTSS algorithm

Researchers use the search range of 15 * 15 to illustrate the performance of this algorithm. In the best case, NTSS algorithm just does 17 points matching, and the worst case needs to do 33 points matching, because the center-biased property of the motion vector is ubiquity in the real video sequences. Generally, the probability of the NTSS algorithm to do 33 point matching is relatively small.

Therefore, in the low rate video applications, such as video phone or video conference, the advantages of NTSS algorithm can be better used. Struck is mainly applied in face-tracking algorithm, and the object changes little between each frame, so the NTSS can be adapted in sampling step to simplify operation process.

IV. CONCLUSION

In view of the study above, researchers can come to the conclusion below:

1) In terms of tracking accuracy, struck using SVM classification is satisfactory in the area of single target tracking, but this method does not run fast which may affect its real-time performance.

2) With the introduction of the NTSS, the amount of computation of sampling step is less in slow motion video tracking.

However, there are still some issues to be solved in the further study. The NTSS takes into account the center-biased property of the motion vector and makes a matching operation on the center of the initial search. When the object moves in a small range, this improvement is very effective and can greatly reduce the amount of computation. However, when the scope of object motion is large, this improvement may bring additional computing capacity. And the actual situation is more complex, and researchers should consider more situations, such as sudden change in motion area and occlusion.

REFERENCES

[1] Sam Hare, Amir Saffari, Philip H.S.Torr. Struck: Structured Output Tracking with Kernels. In Proc ICCV, 2011.

[2] S. Avidan. Support Vector Tracking. IEEE Trans. on PAMI, 26:1064–1072, 2004.

[3] B. Babenko, M. H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In Proc. CVPR,2009.

[4] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In Proc. BMVC, 2006.

[5] A. Saffari, M. Godec, T. Pock, C. Leistner, and H. Bischof. Online Multi-Class LPBoost. In Proc. CVPR, 2010.

[6] C. Leistner, A. Saffari, P. M. Roth, and H. Bischof. On Robustness of On-line Boosting - A Competitive Study. In Proc. ICCV-OLCV, 2009.

[7] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In Proc. ECCV, 2008.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. IEEE Trans. on PAMI, 32(9):1627–1645, Sept. 2010.

[9] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In Proc. ICCV, 2009.

[10] A. Bordes, L. Bottou, P. Gallinari, and J. Weston. Solving multiclass support vector machines with LaRank. In Proc. ICML, 2007.

[11] A. Bordes, N. Usunier, and L. Bottou. Sequence Labelling SVMs Trained in One Pass. In Proc. ECML-PKDD, 2008.

[12] M. B. Blaschko and C. H. Lampert. Learning to Localize Objects with Structured Output Regression. In Proc. ECCV, 2008.

[13] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large Margin Methods for Structured and Interdependent Output Variables. JMLR, 6:1453–1484, Dec. 2005.

[14] J. C. Platt. Fast training of support vector machines using sequential minimal optimization, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[15] Z. Wang, K. Crammer, and S. Vucetic. Multi-Class Pegasos on a Budget. In Proc. ICML, 2010.

[16] Zhang Qishan, Dong Haiyan. A fast motion estimation algorithm for real-time applications [J]. Computing 32, 2006 (2): 223-225.