REAL-TIME ROBUST TRACKING VIA SPARSE REPRESENTATION: A MODE-SEEKING APPROACH

R. Venkatesh Babu

Video Analytics Laboratory
SERC, Indian Institute of Science
Bangalore, India

ABSTRACT

In this paper, we propose a robust real-time tracking as a mode seeking process over likelihood map via sparse representation. In order to estimate the likelihood map, the target and candidate models were represented as overlapping patches. The likelihood map of the target candidate is obtained by sparsely representing the candidate patches in the space spanned by target patches and trivial bases. The object is localized by iteratively seeking the mode of this likelihood map. Since the mode-seeking process localizes the object in few iterations, it achieves real-time speed. Since the local patches are less sensitive to global appearance and illumination changes, the proposed approach shows robustness to the aforementioned challenges. We quantify the performance of the proposed tracker on many video sequences with various challenges involving occlusion, illumination change and pose variations. The proposed approach shows excellent performance in terms of robustness and speed compared to other trackers.

Index Terms— Visual Tracking, Sparse Representation, Mode seeking, Likelihood Map, $l_1$ minimization.

1. INTRODUCTION

Object tracking is one of the critical tasks in computer vision applications such as video surveillance and monitoring. The major challenges of visual tracking arise from variability of object itself (intrinsic) and those in the surroundings (extrinsic). Intrinsic factors include pose, scale changes in the object, complexity of motion and partial/self occlusion. Extrinsic factors include illumination, background clutter, image acquisition noise and presence of similar objects. Hence, a computationally efficient robust tracking algorithm is essential to reduce the influence of these distracting factors and track the object faithfully in real-time.

Recently, sparse representation of signals via $l_1$ minimization has been used for various applications including visual tracking [1, 2]. The work by Wright et al. [3] pioneered the successful application of compressed sensing to various signal processing applications. In their work, they have exploited the discriminative nature of sparse representation to perform face recognition with the dictionary constructed from the training face samples. The face is classified by sparsely representing the test sample in the space spanned by the target templates. The candidate with the least projection error is considered as the tracked target in the successive frames. This tracker is computationally very expensive since tracking is performed in particle filter framework. Further the quality of tracking depends on the number of particles (target candidates) used. Typically few hundreds of candidate locations are examined in order to localize the target. Here, the tracking is achieved at higher computational cost and hence it is not amenable for real-time tracking.

In this paper, we propose a method to achieve robust tracking in real-time with reduced computational cost. In the proposed approach, we pose the tracking problem in an iterative mode seeking framework, similar to mean-shift procedure over likelihood map, obtained via sparse representation. Due to the gradient ascent nature of this framework, it localizes the object in few iterations, leading to real-time performance. In the proposed work, the target from the initial frame, and candidate, from the next frame are represented by local patches extracted from the object and candidate regions respectively. Each candidate patch is sparsely represented as a linear combination of the target patches; and each target patch as a sparse linear combination of candidate patches. The coefficients of these sparse representation were used for measuring the mutual affinity between the target and the candidate patches. The maximum mutual affinity of each candidate patch indicates the likelihood of the patch belonging to object region. The mode of the likelihood map obtained from all the candidate patches indicate the object location in the current frame. An iterative mode seeking algorithm is deployed to rapidly find the mode of this likelihood map. The mode seeking algorithms localizes the object in few iterations. Compared to the particle filter based tracker, the proposed tracker achieves more than 50 times speedup and feasible for real-time applications. The proposed approach shows greater robustness to appearance change since the local patches of the object are less affected by the target’s appearance change compared to conventional object template based models. The proposed approach exhibits excellent robustness to illumination change since the intra patch illumination variation is much less compared to the object template model, especially when the object undergoes partial illumination change. We quantify the performance of the proposed tracker on many challenging video sequences involving occlusion, illumination change and pose variations. The experiments show that the proposed tracker, even without any model update, shows excellent robustness compared to other trackers.

The rest of the paper is organized as follows. The next section briefly summarizes the related works. Section 3 describes the various modules of the proposed robust real-time tracker in sparse representation framework. Section 4 reports the experimental results and finally, section 5 provides concluding remarks.
Visual tracking has been one of the major research areas in computer vision for the past few decades. Varieties of trackers have been reported in many different frameworks including appearance based, feature based, statistical methods based, knowledge based and machine learning based. More details regarding the existing trackers can be obtained from these survey papers [4, 5, 6, 7]. In this section we will briefly capture the trackers developed in sparse representation framework.

The initial tracker in $l_1$ minimization framework was proposed by Mei et al. [1]. They have utilized particle filter to select the candidate particles and then represent them sparsely in the space spanned by the object templates using $l_1$ minimization. This requires a large number of particles for reliable tracking and thus results in high computational cost and brings down the speed of the tracker. Bao et al. [8] improved the execution time of the above tracker by adding a $l_2$ norm regularization on the coefficients associated with the trivial templates. However, this approach fails to track at real-time. Wei et al. [9] proposes a sparsity-based discriminative classifier that exploits holistic templates and a generative model that extracts spatial information of each patch based on histogram. This also does not improve the speed of the tracker. Xu et al. [10] uses an alignment-pooling strategy to locate the target more accurately. It includes a template update strategy to combine incremental subspace learning and sparse representation. This approach improves the robustness but slows down the tracker. Kaihau et al. [11] proposed a high template update strategy to combine incremental subspace learning and non-linear pool strategy to locate the target more accurately. It includes a non-linear pool strategy in compressed domain. This approach deteriorates the accuracy of the tracker to larger extent.

### 2.1. Tracking in sparse representation framework

The problem of representing a signal as a sparse linear combination of overcomplete dictionary has seen a recent explosion in various signal processing application domains. Wright et al. [3] successfully used the discriminative nature of sparse representation for recognizing faces. They used the $l_1$ minimization framework for robust recognition of human faces under adverse conditions such as noise and occlusion. This discriminative power of sparse representation led to a range of applications in computer vision including background subtraction [12], human detection [13] texture segmentation [14], action recognition [15], optic disk detection [16] and tracking [2, 9, 10, 17, 18].

In object tracking, the target candidate in future frames can be sparsely represented by the space spanned by the target templates, created in the initial frame. Such representation is meaningful since the target candidate closely resembles the original target in the successive video frames within a reasonable time window. The problem now is to find the location at which the target candidate is represented most compactly using the target templates (dictionary elements).

### 3. PROPOSED TRACKER

Sparse representation technique endeavors at expressing the input signal using least number of basis vectors (dictionary elements) that most compactly represent the input, leading to its discriminative property. This discriminative property of sparse representation has revolutionized the way of creating the dictionary elements from meaningless basis to more meaningful overcomplete dictionary elements. This paradigm shift that enabled the creation of meaningful dictionary elements constitutes a very important turning point in computer vision and pattern recognition research [19].

In this work, we propose a tracker that inherits the robustness of ‘sparse representation’ and the speed of ‘mode seeking procedure’. In the proposed approach, the target and candidate are represented as a collection of ordered patches as shown in Fig. 1. The object template is downsampled to a smaller dimension before extracting the patches, centered around each pixel location. These patches form target and candidate atoms of the corresponding dictionary. Each candidate patch is sparsely represented in the space spanned by the target dictionary elements by solving the $l_1$ minimization problem.

The likelihood of the candidate patch belonging to the target is indicated by the highest coefficient value. In order to obtain a robust likelihood measure, mutual affinity between candidate and target is considered. On the other hand, the response of the trivial coefficients indicate the locations of non object components present in the patch. Hence, the likelihood of each patch is measured as a ratio between the maximum mutual affinity to cumulative response of the trivial coefficients. A meanshift like iterative mode seeking algorithm applied over this likelihood map localizes the object rapidly within few iterations. Since the proposed tracker localizes the target by gradient ascent procedure, $l_1$ minimization is applied only to few candidates along the mode seeking path. This provides huge computational advantage with real-time performance over the particle filter based tracker that typically examines hundreds of candidates for localizing the object. Further, the proposed approach provides excellent robustness to illumination and appearance changes. This is due to the fact that object patches occupy smaller region of target, and hence are least affected by global changes. For example, when a part of the object is shadowed/illuminated, the change in global object model is severe compared to the local patches of the object. The speed and robustness of the proposed tracker is illustrated with many challenging videos.

### 3.1. Sparse Representation

Consider a signal $y \in R^n$ that allows sparse representation, then it can be represented by linear combination of few elements of dictionary $T \in R^{n \times k}$. Let $T$ be the set of target patches, where each element is a vectorized image patch $t_i \in R^n$. A candidate patch in the next frame $(y_i)$ can be approximately represented by a sparse linear combination of target template patches $t_i$ as shown in Fig 1.

$$y_i \approx Ta_i = a_{i1}t_1 + a_{i2}t_2 + \ldots + a_{ik}t_k \quad (1)$$

where, $t_1 \ldots t_k$ are target bases and $a_{i1} \ldots a_{ik}$ are the corresponding target-coefficient vectors. The coefficient $a_{ij}$ indicates the affinity between $i$-th candidate and the $j$-th target patch.

The coefficient vector $a_i = \{a_{i1}, a_{i2}, \ldots a_{ik}\}$ is obtained as a solution of

$$\min_{a_i \in R^k} \|a_i\|_1 \quad s.t. \quad \|y_i - Ta_i\|_2^2 \leq \lambda \quad (2)$$

Where, $\lambda$ is the reconstruction error tolerance in mean square sense. Considering the effect of occlusion and noise, (1) can be written as,

$$y_i = Ta_i + \epsilon \quad (3)$$

The nonzero entries of the error vector $\epsilon$ indicate noisy or occluded pixels in the candidate patch $y_i$. Following the strategy in [3], we use trivial templates $I = \{[1] \ldots [1] \in R^{n \times 1}\}$ to capture the noise and occlusion.

$$y_i = Ta_i + Ic_i \quad (4)$$

where

$$c_i = a_{i1}t_1 + \ldots + a_{ik}t_k + c_{i1}1_1 + \ldots + c_{in}1_n$$
where, each of the trivial templates \( \{ t_1, \ldots, t_k \} \) is a vector having nonzero entry only at one location and \( \{ c_1, \ldots, c_m \} \) is the corresponding coefficient vector. A non zero trivial coefficient indicates the reconstruction error at the corresponding pixel location, possibly due to noise or occlusion.

**Fig. 1.** Patch representation: (a) target patches \( t_i \) and (b) candidate patches \( y_i \). Here, each candidate \( (y_i) \) patch is represented as a sparse linear combination of target patches \( (t_1, t_2, \ldots, t_k) \) and vice versa.

### 3.2. Likelihood Map Estimation

The appearance of an object does not change drastically across successive frames. This proximal-similarity to the target object, in neighboring frames is utilized for tracking. In our approach, we partition the target object region into many overlapping smaller image patches. The dictionary is created by cascading these vectorized image patches \( (t_i) \). Similarly, the candidate patches \( (y_i) \) are generated from the next frames centered at the pervious object location.

The patches extracted from the object window of a new frame can be represented sparsely using target object dictionary \( T \). Since the object appearance in the new frame is similar to the target object, most of the patches extracted from the co-located object window of new frame \( (y_i) \) can be represented sparsely using target object dictionary elements \( y_i \). Coefficient \( a_{ij} \) indicates the match between \( i \)-th candidate and the \( j \)-th target patch.

For robust estimation of the likelihood map, the target dictionary elements \( (t_i) \) can be represented as a sparse linear combination of candidate patches \( (y_i) \) and trivial bases.

\[
t_j = Y_j b_j + \epsilon = b_{ji} y_i + \ldots + b_{jk} y_k + c_{j1} 1_1 \ldots + c_{jn} 1_n
\]  

(5)

Let \( b_{ji} \) indicates the affinity between \( j \)-th target and \( i \)-th candidate patch. The coefficient vector \( b_j \) is obtained as a solution of (2) by reversing the roles of target \( (T) \) and candidate \( (Y) \). Now the robust matching score between \( j \)-th candidate and the \( j \)-th target patch is given by their mutual affinity.

\[
f_{ij} = a_{ij} b_{ji}
\]  

(6)

The maximum value of the mutual affinity, \( \max_{j} \{ f_{ij} \} \) indicates the likelihood of \( i \)-th candidate patch \( y_i \) belonging to object. Similarly, the cumulative response of the trivial basis coefficients \( \sum_j c_{ij} \) indicates the occlusion or noise component of the corresponding patch. The final likelihood of the patch is obtained as the ratio of maximum mutual affinity to the cumulative response of the trivial coefficients as given in (7).

\[
L(g(i)) = L(u, v) = \frac{\max_j \{ f_{ij} \}}{1 + \alpha \sum_j c_{ij}} K(u, v)
\]  

(7)

where, \( L \) is the likelihood map, \( L(g(i)) \) is the likelihood of \( i \)-th candidate patch and \( \alpha \) is the parameter that weighs trivial basis component corresponding to noise or occlusion. Since the boundary likelihood values are less reliable due to possible inclusion of background pixels or occlusion, a kernel \( (K) \) is typically used to provide more weight to interior object patches and less weight to boundary patches. Typical kernel used are Gaussian or Epanechnikov.

### 3.3. Tracking by Mode Seeking

The object is localized by iteratively seeking the mode of the estimated likelihood similar to mean-shift gradient ascent. Let \( C_0 \) be the initial object center. The object center in the \( (k + 1) \)-th iteration is given by:

\[
C_{k+1} = \frac{\sum_{(u,v)} L_k(u,v) \cdot \sum_{(u,v)} v \cdot L_k(u,v) - \sum_{(u,v)} u \cdot L_k(u,v) - \sum_{(u,v)} L_k(u,v)}{\sum_{(u,v)} L_k(u,v)} + C_k
\]  

(8)

where, \( L_k \) is the likelihood map at \( k \)-th iteration. The iteration can be terminated if the mode location does not change much between two successive iterations.

**Fig. 2.** Mode seeking over likelihood space: Here, first column illustrates the initial target and the corresponding likelihood. Columns 2 to 4 show the target and likelihood map at each iteration of tracking for walk1 sequence.
4. RESULTS AND DISCUSSION

The proposed approach has been tried on various publicly available complex video sequences\(^1\) including *trellis*, *panda*, *walk* and *birch*. The results are compared with mean-shift tracker (MS) [20], particle filter based \(l_1\) minimization tracker (L1-PF) [1], Compressive tracking (CT) [11], fragment-based tracking (FragTrack) [21], SGM [9], incremental visual tracking (IVT) [22] and visual tracker (VT) [10]. For mean-shift tracker the object is represented by a joint histogram with 10 bins along each dimension. The histogram is obtained by spatially weighing the object region with a Gaussian kernel. For all other tracks, we have used the code provided by the corresponding authors. For the proposed approach, the target and candidate patches are created after resizing the object window to \(12 \times 12\). Overlapping patches of size of \(7 \times 7\) are extracted from these \(12 \times 12\) object window in order to build the target and candidate atoms of the dictionary. The value of \(\alpha\) in (7) is set at 10. For \(l_1\) minimization, the sparse decomposition toolbox of open source Sparse Modeling software (SPAMS) is used [23]. The performance of the trackers are measured as the RMSE deviation with respect to the ground truth trajectory.

\[\text{Position Error (pixels)} \]

\[\begin{array}{cccccc}
\text{Seq} & \text{Prop} & \text{MS} & \text{L1-PF} & \text{SGM} & \text{IVT} & \text{CT} \\
\text{Trellis} & 98.98 & 98.84 & 67.87 & 247.2 & 358.3 & 54.6 \\
\text{Walk} & 5.47 & 15.2 & 11.06 & 6.3 & 211.8 & 77.3 \\
\text{Panda} & 3.24 & 32.67 & 16.26 & 60.4 & 100.9 & 29.7 \\
\text{Birch} & 3.52 & 9.14 & 3.74 & 169.4 & 56.3 & 21.1 \\
\end{array}\]

\[\text{Frames} \]

\[\begin{array}{cccccc}
\text{Seq} & \text{Prop} & \text{MS} & \text{L1-PF} & \text{SGM} & \text{IVT} & \text{CT} \\
\text{Trellis} & 0.02 & 0.72 & 2.2 & 0.08 & 0.1 & 0.22 \\
\text{Walk} & 0.014 & 1.0 & 2.72 & 0.08 & 0.1 & 0.22 \\
\text{Panda} & 0.015 & 0.91 & 2.4 & 0.08 & 0.08 & 0.2 \\
\text{Birch} & 0.014 & 0.76 & 2.0 & 0.07 & 0.08 & 0.17 \\
\end{array}\]

The tracking windows for all trackers at various time instances of the sequences are shown in Fig. 3. The position error plots with respect to the ground truth for all videos are given in Fig. 4. The performance of the trackers are evaluated by measuring the RMSE with respect to the ground truth trajectory. Table 1 provides the RMSE values for various video sequences. The average execution time of the proposed algorithm is compared in table 2, which clearly indicates the real-time capability of the proposed tracker. The proposed tracker is implemented in Matlab and the execution time is provided for the PC with Intel Core 2 Duo CPU @ 2.93 GHz and 2GB RAM. From the table 2, it can be observed that the proposed algorithm tracks at 50-80 frames/sec, whereas the \(l_1\) minimization tracker (L1-PF) runs at 1 frames/sec with 300 particles.

\[\text{Table 1. Trajectory RMSE with respect to ground-truth}\]

\[\text{Table 2. Average execution time for various sequences (sec)}\]

In this paper, we have proposed a robust and computationally efficient object tracker in \(l_1\) minimization framework. The proposed tracker estimates the candidate likelihood map and localizes the object by iterative mode seeking procedure. Here, the target and candidate are represented as a collection of overlapping patches. The likelihood map of the object in the consecutive frames are obtained by measuring the affinity of the candidate patches with target dictionary via \(l_1\) minimization. Unlike the particle filter based approach, the proposed approach does not examine numerous candidates for localizing the object location. Hence, the proposed approach attains real-time performance when executed on a PC. Though the proposed approach is quite robust to scale change, currently it does not estimate the scale of the object, which is an issue with most of the non-particle filter based trackers. The proposed approach shows better robustness and accuracy due to the patch based object modeling, compared to various state of the art trackers. The accuracy and speed of the proposed approach is quantified with various complex video sequences.

\[\text{Fig. 3. Result for *trellis*, *panda*, *walk* and *birch* video sequences (from top to bottom).}\]

\[\text{Fig. 4. Trajectory position error with respect to ground truth for: (a) *trellis* (b) *panda* (c) *walk* (d) *mbseq* sequences}\]

---

\(^1\)The original videos are obtained from the following links: *trellis*: http://www.cs.toronto.edu/~dross/ivt/index.html *panda*: http://info.ee.surrey.ac.uk/Personal/Z.Kalal/TLD/TLD dataset.ZIP *mb_seq*: http://vision.stanford.edu/~birch/headtracker/seq/
6. REFERENCES


