ABSTRACT

In this paper, we propose an interest point based object tracker in sparse representation (SR) framework. In the past couple of years, there have been many proposals for object tracking in sparse framework exhibiting robust performance in various challenging scenarios. One of the major issues with these SR trackers is its slow execution speed mainly attributed to the particle filter framework. In this paper, we propose a robust interest point based tracker in \(l_1\) minimization framework that runs at real-time with better performance compared to the state of the art trackers. In the proposed tracker, the target dictionary is obtained from the patches around target interest points. Next, the interest points from the candidate window of the current frame are obtained. The correspondence between target and candidate points are obtained via solving the proposed \(l_1\) minimization problem. A robust matching criterion is proposed to prune the noisy matches. The object is localized by measuring the displacement of these interest points. The reliable candidate patches are used for updating the target dictionary. The performance of the proposed tracker is bench marked with several complex video sequences and found to be fast and robust compared to reported state of the art trackers.

Index Terms— Visual Tracking, \(l_1\) minimization, Interest points, Harris corner, Sparse Representation.

1. INTRODUCTION

Visual tracking has been one of the key research areas in computer vision community for the past few decades. Tracking is a crucial module for video analysis, surveillance and monitoring, human behavior analysis, human computer interaction and video indexing/retrieval. The major challenges arise in real life scenario are due to intrinsic and extrinsic factors. The intrinsic factors include pose, appearance and scale changes. The extrinsic factors are: illumination, occlusion and clutter.

There has been several proposals for object tracking in the past. Based on object modeling, the majority of the trackers can be brought under the following two main themes: i) Global object model and ii) Local - object model. In global approach, the object is typically modeled using all the pixels corresponding to the object region or some global property of the object. The simple template based SSD (sum of the squared distance) tracker, color histogram based mean-shift trackers [1, 2] and the machine learning based trackers [3, 4] are examples of global trackers. Traditional Lucas-Kanade tracker [5] and many bag-of-words model trackers are some examples for interest point tracker. A detailed survey on various object tracking methods can be found in [6, 7, 8].

The proposed interest point based tracker falls under the local object model based tracking. Shi et al. showed [9] that the corner-like points are more suitable for reliable tracking due to its stability and robustness to various distortions like rotation, scaling, illumination etc. Though the interest-point based trackers show more robustness to various factors like rotation, scale and partial occlusion, the major issues surface from description and matching of interest points between the successive frames. For example, Kloihofer et al. [10] use SURF [11] descriptors of interest points as feature descriptors. The object is tracked by matching the object points of the previous frame with candidate points in the current frame. The displacement vectors of these points are utilized for localizing the object in the current frame [12]. Matching the features of interest points between two frames is a crucial step in estimating the correct motion of the object and typically Euclidian distance is used for matching. The recently proposed vision algorithms in sparse representation framework clearly illustrates its superior discriminative ability at very low dimension even among similar feature patterns [13, 14]. This motivated us to examine the matching ability of sparse representation for interest points.

In this paper, we have proposed a robust interest point based tracker in sparse representation framework. The interest points of the object are obtained from the initial frame by Harris corner detector [15] and a dictionary is constructed from the small image patch surrounding these corner points. The candidate corner points obtained from the search window of the current frame is matched with object points (dictionary) by sparsely representing the candidate corner patches in terms of dictionary patches. The correspondence between the target and candidate interest points are established via the maximum value of the sparse coefficients. A ‘robust matching’ criterion has been proposed for pruning the noisy matches by checking the mutual match between candidate and target patches. The object is localized based on the displacement of these matched interest points. The matched candidate patches are used for updating the target dictionary. Since the dictionary elements are obtained from a very small patch surrounding these corner points, the proposed approach is robust and computationally very efficient compared to the particle filter based \(l_1\) trackers [14, 17, 18, 19, 20].

The rest of the paper is organized as follows: Section 2 briefly reviews related works. Section 3 explains the proposed tracker in sparse representation framework. Section 4 discusses the results and concluding remarks are given in section 5.

2. SPARSE REPRESENTATION BASED TRACKING

The concept of sparse representation recently attracted the computer vision community due to its discriminative ability [13]. Sparse representation has been applied to various computer vision tasks including face recognition [13], image video restoration [21], image denoising [22], action recognition [23], super resolution [24], and tracking
In this paper, we propose a tracker that utilizes the flexibility of the interest points and robustness of sparse representation for matching these interest points across frames. The proposed tracker represents the object as a set of interest points. The interest points are represented by a small image patch surrounding it. The vectorized image patches are $l_2$ normalized to form the atoms of the target dictionary. In order to localize the object in the current frame, the interest points are obtained from the predefined search window in the current frame. Each target interest point (object dictionary element) is represented as a sparse linear combination of the candidate dictionary atoms. The discriminative property of sparse representation enables matching between target and candidate patches robustly. For each target patch, the candidate patch whose corresponding coefficient is maximum, is chosen as the match. The noisy matches are pruned via the proposed ‘robust matching’ criteria by checking the mutual match between candidate and target patches. The mutual correspondence is confirmed by sparsely representing the candidate interest points in terms of target interest points. Only the points that mutually agree with each other were considered for object localization. The object location in the current frame is estimated as the median displacement of candidate points with respect to it’s matching target points. The overview of proposed tracker is shown in Fig 1. Since the proposed tracker is a point tracker, it shows robustness to global appearance, scale and illumination changes. Further, the proposed tracker uses only a small patch around the interest points as dictionary elements, making it compact and computationally very efficient.

### 3.1. Sparse Representation

Let $T = \{t_i\}$ be the set of target patches corresponding to the target interest points, represented as $l_2$ normalized vectors. Now, the candidate interest points, represented as the vectorized image patches $y_i$, obtained within a search window of the current frame could be represented by a sparse linear combination of the target dictionary elements $t_i$.

$$y_i \approx T a_i = a_{i1} t_1 + a_{i2} t_2 + \ldots + a_{ik} t_k$$

(1)

where, $\{t_1, t_2, \ldots, t_k\}$ are target bases and $\{a_{i1}, a_{i2}, \ldots, a_{ik}\}$ are the corresponding target-coefficient vectors. The coefficient $a_{ij}$ indicates the affinity between $i$-th candidate and the $j$-th target interest point based on the discriminative property of sparse representation.

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

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

(2)

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

$$y_i = T a_i + \epsilon$$

(3)

The nonzero entries of the error vector $\epsilon$ indicate noisy or occluded pixels in the candidate patch $y_i$. Following the strategy in [13], we use trivial templates $I = \{I_1, \ldots, I_n\} \in \mathbb{R}^{n \times n}$ to capture the noise and occlusion.

$$y_i = T a_i + I c_i$$

(4)

$$= a_{i1} t_1 + \ldots + a_{ik} t_k + c_{i1} I_1 + \ldots + c_{in} I_n$$
where, each of the trivial templates \( \{t_1, \ldots, t_n\} \), 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.

### 3.2. Object Localization

![Fig. 2. Robust matching of interest points. (a) Target window with interest points (b) Candidate window with interest points. The red patches are the mutually matched points, green patches matched only one way (either target to candidate or candidate to target) and the blue patches are unmatched ones.](image)

The object is tracked by finding the correspondence between the target and candidate interest points in successive frames. Only the reliable candidate interest points that mutually match with each other are considered for estimating the object location. Let \( l \) be the number of interest points in the current frame within the search window (all points shown in Fig 2(b)). Only \( k \) \((k < l)\) interest points out of \( l \) points that match with the dictionary atoms will be considered for object localization (all green-red points shown in Fig. 2(b)).

In order to find the candidate points that match with the target points, each target interest point \( t_j \) is represented as a sparse linear combination of candidate points (\( y_i \)) and trivial bases.

\[
\mathbf{t}_j = \mathbf{Y} \cdot \mathbf{b}_j + \epsilon = b_{j1}y_1 + \ldots + b_{jt}y_t + c_{j1}k_1 + \ldots + c_{jn}k_n \tag{5}
\]

Utilizing the discriminative property of sparse representation, the matching candidate point for the given target point \( t_j \) is determined by the maximum value of the candidate dictionary coefficient \( (b_j) \). Let the coefficient \( b_{j_i} \) corresponding to the candidate dictionary element \( y_i \) be the maximum value. Then the matching candidate point for the target point \( t_j \) is declared as \( y_i \). Mathematically, \( b_{j_i} = \max_r \{ b_{jr} \}, \ r \in \{1, 2, \ldots, l\} \), indicates the match between \( j \)-th target point and \( i \)-th candidate point. All the matched candidate points corresponding to each target dictionary element is shown by green and red patches in Fig. 2(b). If two or more target points match with same candidate point, only the candidate patch with higher coefficient value will be considered. Hence, the number of candidate matching points is less than or equal to the number of target dictionary elements.

### 3.3. Robust Matching

Let \( k_1 (k_1 \leq k) \) be the number of matched candidate interest points (all red and green points shown in Fig. 2(b)). In order to prune the noisy matches, now each of these candidate interest point \( (y_i) \) is represented as a sparse linear combination of target points \( (t_k) \) and trivial bases as in (5). Let \( a_{ir} = \max_r \{ a_{jr} \}, \ r \in \{1 \ldots k\} \) indicates the match between \( i \)-th candidate point and \( j \)-th target point. The \( i \)-th candidate point is selected if:

\[
\max_r \{ b_{jr} \} = b_{ji}, \ r \in \{1 \ldots l\} \tag{6}
\]

\[
\max_r \{ a_{ir} \} = a_{ij}, \ r \in \{1 \ldots k\} \tag{7}
\]

All the candidate points that agree with the above robust matching criteria were considered for object localization (all red points shown in Fig. 2(b)). Let there be \( k_2 (k_2 \leq k_1) \) such candidate points located at \([x'_i, y'_i], \ i \in \{1 \ldots k_2\}\) with respect to object window center \((x, y)\). Let \([x'_i, y'_i], \ i \in \{1 \ldots k_2\}\) be the corresponding locations of target points with respect to the same object window center \((x_o, y_o)\). Now the displacement of object is measured as median displacement of candidate points with respect to it’s matching target points. Let \( dx_i \) and \( dy_i \) be the displacement of \( i \)-th candidate point measured as:

\[
dx_i = x_i - x'_i, \ i \in \{1, 2, \ldots, k_2\} \tag{8}
\]

\[
dy_i = y_i - y'_i, \ i \in \{1, 2, \ldots, k_2\} \tag{8}
\]

The object location in the current frame is obtained using the median displacement of each coordinate.

\[
x_o = x_o + \text{median}(dx_i), \ i \in \{1, 2, \ldots, k_2\} \tag{9}
\]

\[
y_o = y_o + \text{median}(dy_i), \ i \in \{1, 2, \ldots, k_2\} \tag{9}
\]

### 3.4. Dictionary Update

Since the object appearance and illumination changes over time, it is necessary to update the target dictionary for reliable tracking. We employ a simple update strategy based on the confidence of matching. The top 10% of matched candidate points are added to the dictionary if the matched sparse coefficient is above certain threshold. Same number of unmatched points from the target dictionary are removed to accommodate these new candidate points.

**Algorithm 1 Proposed Tracking**

1. Input: Initialize object window centered at \([x_o, y_o]\).
2. Obtain the interest points located within object window using Harris corner detector.
3. Create Target Dictionary \( t_i \) from each object interest point located at \([x'_i, y'_i]\) with respect to \([x_o, y_o]\).
4. Repeat
5. Obtain the candidate interest points from the current frame, within a search window. Select image patch \( y_i \) centered around each interest point located at \([x'_i, y'_i]\) with respect to \([x_o, y_o]\).
6. Obtain the matching candidate points corresponding to target points according to (5)
7. Robust Matching: Select only those candidate points that mutually match with the target points as in (6).
8. Estimate the displacement of the object as the median displacement of each coordinate of mutually matched candidate points as in (7).
9. Localize the object by adding this displacement to current object location as in (8)
10. Update the target dictionary
11. Until Final frame of the video.
4. RESULTS AND DISCUSSION

The proposed tracker is implemented in MATLAB and experimented on publicly available complex video sequences with various challenging scenarios such as illumination change, partial occlusion and appearance change. These videos were shot with different sensors such as color, gray-scale, infra-red and were recorded in indoor and outdoor environments. To solve $l_1$ minimization problem we have used the Sparse Modeling Software (SPAMS) package [28] with the regularization constant $\lambda$ set at 0.1 for all the experiments. For interest point detection, we have used Harris corner detector obtained from [29] with Nobel’s corner measure [30]. Since the number of interest points are typically proportional to size of the object, we have set the parameters of Harris corner detector depending on the size of the target. For all video sequences we have used $\sigma = 5$ and threshold = 1. For target size less than $50 \times 50$ we have used radius = 0.5 and radius = 2 for size greater than $50 \times 50$. For satisfactory tracking, we need at least $K$ target interest points for an object. In our experiments we have observed that only 25 - 30 % of the interest points satisfy our robust matching criteria. Based on this observation we have set the value of $K$ at 100. For some reason, if the number of target interest points are too low or too high compared to $K$, the threshold and radius parameter can be adjusted appropriately to get the desired number of interest points. For each interest point, a $5 \times 5$ image patch centered around that point is used for creating dictionary atoms. For evaluating the performance of the proposed tracker, we have compared the results with $l_1$ tracker proposed by Mei et al. [14], using 300 particles, meanshift tracker [1], fragment-based tracking (FragTrack) [31], SGM [32], APG [17] and compressive tracking (CT) [27]. We have shown results for the following four different video sequences: pkest02 (515 frames), panda (900 frames), Dudek (1145 frames) and car4 (431 frames).

![Fig. 3](http://www.cs.toronto.edu/vis/projects/dudekfaceSequence.html)

**Fig. 3.** Result for pkest02, panda, dudek and car video sequences (from top to bottom).

The tracking windows for all trackers at various time instances of these sequences are shown in Fig. 3. Tables 1 and 2 summarizes the performance of the trackers with respect to accuracy (pixel deviation from ground truth) and execution time (in sec.). It can be observed that the proposed tracker achieves real time performance with better tracking accuracy compared to many recent state of the art trackers such as CT, SGM and APG. For all trackers, author’s original implementation has been used.

![Table 1](http://www.cs.toronto.edu/vis/projects/dudekfaceSequence.html)

**Table 1. Average RMSE**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>pkest02(515)</td>
<td>46.64</td>
<td>114.3</td>
<td>20.5</td>
<td>89.24</td>
<td>20.5</td>
<td>102.2</td>
<td>103.9</td>
</tr>
<tr>
<td>panda(900)</td>
<td>9.62</td>
<td>11.97</td>
<td>28.88</td>
<td>440.7</td>
<td>58.84</td>
<td>172.4</td>
<td>8.81</td>
</tr>
<tr>
<td>Dudek(1145)</td>
<td>52.37</td>
<td>151.42</td>
<td>208.0</td>
<td>95.13</td>
<td>22.90</td>
<td>95.44</td>
<td>20.11</td>
</tr>
<tr>
<td>car4(431)</td>
<td>10.13</td>
<td>178.1</td>
<td>8.64</td>
<td>13.1</td>
<td>9.84</td>
<td>134.8</td>
<td>6.66</td>
</tr>
</tbody>
</table>

![Table 2](http://www.cs.toronto.edu/vis/projects/dudekfaceSequence.html)

**Table 2. Execution time (secs)**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>pkest02(515)</td>
<td>0.019</td>
<td>0.23</td>
<td>0.35</td>
<td>3.32</td>
<td>0.09</td>
<td>0.013</td>
<td>0.021</td>
</tr>
<tr>
<td>panda(900)</td>
<td>0.041</td>
<td>0.32</td>
<td>0.35</td>
<td>3.6</td>
<td>0.1</td>
<td>0.023</td>
<td>0.023</td>
</tr>
<tr>
<td>Dudek(1145)</td>
<td>0.042</td>
<td>0.32</td>
<td>0.35</td>
<td>3.6</td>
<td>0.1</td>
<td>0.024</td>
<td>0.025</td>
</tr>
<tr>
<td>car4(431)</td>
<td>0.038</td>
<td>0.23</td>
<td>0.35</td>
<td>3.3</td>
<td>0.09</td>
<td>0.017</td>
<td>0.022</td>
</tr>
</tbody>
</table>

5. CONCLUSION

The interest point based object modeling along with the proposed robust matching criterion provides robustness to various challenges including partial occlusion, illumination/appearance changes. The performance of the proposed real-time tracker has been bench-marked with various publicly available complex video sequences. The quantitative evaluation of the proposed tracker is compared with the many recent state of the art trackers including CT, SGM, APG and $l_1$ trackers. The overall performance of the proposed tracker is found to be better in terms of both accuracy and speed.
6. REFERENCES


