Abstract

In this paper, the accuracy of visual tracking is enhanced by leveraging a novel measure for observation quality. We measure observation quality with mutual information, then look at the interval covered by that mutual information. As observation uncertainty the interval length is proposed. The best observation is considered the one that both maximizes the observation quality and minimizes the observation uncertainty. We show that searching for the best observation in these terms amounts to preprocessing the image by subtracting the background, detecting salient regions, and rendering the image illumination-invariant. These preprocessing steps are very fast and can precede any existing tracker. In experiments it is shown that the performance of several trackers can be substantially boosted when they run on our preprocessed images, rather than the raw input for which they were intended. In all cases the version with preprocessing significantly outperforms the original tracker's performance.
that can handle a specific change of a target. Because VTS uses a limited number of instances however, it can only cover the types of changes previously observed in the target.

Hence, it may be better to consider all possible instances of the models rather than choosing a subset [Benavoli et al. 2011]. Hence we represent a model instance as a probability distribution. Then, by obtaining the lower and upper bounds of the distribution, we can employ all possible instances for visual tracking. Recently, some trackers made inroads into this direction and used all possible instances. DBB Kwon et al. (2014) estimated the lower and upper bounds of a state to solve the uncertainty in representing highly non-rigid objects. IT Kwon and Lee (2014a) inferred the posterior interval by integrating both likelihood and state intervals. However, there have been few efforts to consider the uncertainty of observation extraction in tracking.

The present study tackles the modeling of uncertainty in the observations. The quality of an observation can be measured by its mutual information \( I(Y_t, L) \), where \( Y_t \) is the observation (image) at time \( t \) and \( L \) determines the classes of observations. To fix ideas, \( L \) can here be understood as an image with value \( f \) or \( b \) at every pixel. \( Y_t \) and \( L \) can be represented by a matrix. Let \( l \) be each pixel of \( L \), where \( l \in \{f, b\} \). Then \( l = f \) if an observation / pixel belongs to the foreground and \( l = b \) otherwise. Due to the observation uncertainty, mutual information is estimated as an interval. Conventionally mutual information is measured as a scalar value. We argue and empirically demonstrate that this mutual information is measured differently at each time due to the measuring error in real situation. Then gathering all values measured forms an interval.

We can minimize the uncertainty by searching for an observation that gives the shortest interval length of mutual information. On the other hand, we want the quality of the observation to be high, i.e. we want it to maximize its mutual information. In this paper, we show that following such strategy supports applying certain types of preprocessing operations to the images (i.e. before tracking), as illustrated in Fig. 1. Minimizing the observation uncertainty leads to applying saliency detection and increasing the invariance under illumination changes. Maximizing the observation quality leads to background subtraction. With these preprocessing algorithms, we can optimize the image where the tracker gets its observations from, in the context of our information-theoretical account. These conclusions will also be validated in our experiments.

Fig. 2 shows the whole pipeline of our method.

The remainder of this paper is organized as follows: Related work is introduced in Section 2. Section 3 presents how to estimate observation quality with mutual information. In Section 4 observation quality estimation is transformed into observation preprocessing. Then a robust visual tracking method is proposed, which is augmented by preprocessing, in 5. Section 7 presents the experimental results. Section 9 concludes this study with the discussions.

2. Related Work and Our Contribution

Our main contribution is to enhance trackers by leveraging observation uncertainty (in terms of mutual information). Mutual information has been widely used in visual tracking. For example, Dowson and Bowden employed mutual information in the Lucas-Kanade framework Dowson and Bowden (2008), where a reference image is registered with a current observation image. Dame and Marchand used Mutual Information
as a metric for real-time visual tracking through alignment [Dame and Marchand (2010). Most conventional tracking methods, including the aforementioned two, do not exploit the uncertainty in the mutual information, however. Our method leverages it to improve tracking performance.

As another contribution, we theoretically underpin that the observation uncertainty problem can be resolved by three image processing techniques: background subtraction, saliency detection, and extracting illumination-invariant (intrinsic) intensities. There already are many papers that process images to improve visual tracking performance. For example, background subtraction helped avoid background clutter for robust visual tracking in Stauffer and Grimson (1998); Kim (2008). Saliency detection helped to find good features inside an object region Yang et al. (2007); Mahadevan and Vasconcelos (2009); Hong et al. (2015). Thus, just using image processing techniques for visual tracking is not a novel idea per se. Yet, so far it was not clear which image processing technique is useful for robust visual tracking, nor why. Our paper comes with a rationale of why to use the particular combination of preprocessing operations that we do.

Using background subtraction, saliency detection and illumination-invariant detection separately may be weak to several noises in the video. However we empirically demonstrated that integration of three tasks above can become robust to those noises compared with the individual use of tasks. Three tasks are not chosen manually and the integration is not ad-hoc because this combination comes from the mathematic in measuring observation quality.

Our approach can be plugged into any kind of visual trackers. IT in Kwon and Lee (2014a) also proposed an enhancement that can be combined with other trackers. To be combined, however, IT had to modify the Markov Chains inside those tracking algorithms, which is not an easy task. Our method does not change any procedure inside the other trackers but only requires to preprocess their input images.

3. Estimating Observation Quality with Mutual Information

In the remainder of the paper, \( Y_t \) stands more specifically for the (preprocessed) values within a Region of Interest (ROI), that probably consists of the target. \( Y^c_t \) denotes the image outside the ROI\(^1\). The ROI will be implicitly determined by the preprocessing presented in Sec.4

\(^1\)\( Y_t \cap Y^c_t = \emptyset \), \( Y_t \cup Y^c_t = U \), where \( U \) is the universal set.
With these definitions for the observation, we measure the observation quality using mutual information $I(Y_t,L)$. If we do not consider the observation uncertainty, the best observation can be obtained by using

$$Y^{(\text{best})}_t = \operatorname{argmax}_{Y_t} I(Y_t,L).$$  \hfill (1)

However, the observation uncertainty inevitably exists in real world and the uncertainty induces mutual information interval $I(Y_t,L) \leq \bar{I}(Y_t,L) \leq \tilde{I}(Y_t,L)$, where $I(Y_t,L)$ and $\bar{I}(Y_t,L)$ are the lower and upper bounds of mutual information, respectively. In this case, the best observation is defined as the one that maximizes the average mutual information and simultaneously minimizes the mutual information interval:

$$Y^{(\text{best})}_t = \operatorname{argmax}_{Y_t} \frac{I(Y_t,L) + \bar{I}(Y_t,L)}{2}, \text{ simultaneously,}$$ \hfill (2)

$$Y^{(\text{best})}_t = \operatorname{argmin}_{Y_t} \bar{I}(Y_t,L) - I(Y_t,L).$$ \hfill (3)

**Lemma 1.** The average mutual information in (2) is approximated by

$$\frac{I(Y_t,L) + \bar{I}(Y_t,L)}{2} \approx I(Y^{(\text{best})}_t,L).$$ \hfill (4)

The mutual information interval in (3) is approximated by

$$\bar{I}(Y_t,L) - I(Y_t,L) \approx H(Y_t|Y^{(\text{best})}_t) + H(Y_t) + H(Y^{(\text{best})}_t),$$ \hfill (5)

where $H$ denotes the entropy.

**Proof.** Please refer to Section [6](#).

By using (4), the objective of (2) is changed to find $Y^{(\text{best})}_t$ that maximizes $I(Y^{(\text{best})}_t,L)$:

$$Y^{(\text{best})}_t = \operatorname{argmax}_{Y_t} I(Y^{(\text{best})}_t,l = \{f,b\}),$$ \hfill (6)

where $I(Y^{(\text{best})}_t,l = \{f,b\})$ is designed in (7). By decomposing (5) into two terms, $H(Y_t|Y^{(\text{best})}_t)$ and $H(Y_t) + H(Y^{(\text{best})}_t)$, the objective of (3) is changed to separately minimize each:

$$Y^{(\text{best})}_t = \operatorname{argmin}_{Y_t} H(Y_t|Y^{(\text{best})}_t),$$ \hfill (7)

and

$$Y^{(\text{best})}_t, Y^{(\text{best})}_t = \operatorname{argmin}_{Y_t,Y^{(\text{best})}_t} [H(Y_t) + H(Y^{(\text{best})}_t)].$$ \hfill (8)

To get the optimal solution, (7) and (8) should be jointly optimized. Since the joint optimization is NP-hard, however, we iteratively optimize (7) and (8), which gives a suboptimal solution.

4. **Transforming Observation Uncertainty into Observation Preprocessing**

The aforementioned goals (6), (7), and (8) are achieved by preprocessing the raw observation image as follows.

- **GOAL 1** in (6) can be achieved by maximizing the mutual information for the background, which is achieved by background subtraction [Mahadevan and Vasconcelos, 2008] (background pixel values will all become almost 0, so their values are highly predictable).

- **GOAL 2** in (7) is to minimize the conditional entropy, which is achieved by background detection [Li et al., 2009] (yielding typically high values in the foreground and low values in the background).

- **GOAL 3** in (8) is to minimize the entropy. The objective function of this goal is same as producing an intrinsic image in [Finlayson et al., 2004]. The intrinsic image minimizing the entropy is achieved by extracting illumination-invariant intensity information in both foreground and background [Finlayson et al., 2004] (eliminating much of the intensity variations not due to surface texture).

Please note that these preprocessing methods can indeed be performed before the actual tracking is activated. These goals will be analysed in more detail next.

4.1. **max $I(Y^{(\text{best})}_t,L)$ : Background Subtraction**

The background subtraction is performed by first detecting background regions and then subtracting those from an image. The background regions are detected by maximizing their mutual information with the background label. Maximizing mutual information with
this label can be interpreted as a classification problem [Vasconcelos (2002)]:

\[
I(Y_c^t, L) = H(Y_c^t) - H(Y_c^t|L) \\
= H(Y_c^t) - p(l = f)H(Y_c^t|l = f) - p(l = b)H(Y_c^t|l = b),
\]

where the mutual information is maximized if \( p(l = f) = p(Y_c^t \in \text{foreground}) = 0 \) (i.e. the region \( Y_c^t \) does not belong to the foreground) and \( H(Y_c^t|l = b) = 0 \) (i.e. \( Y_c^t \) is completely determined by the background label \( l = b \)). Hence, a large \( I(Y_c^t, L) \) implies that we detect more parts of background.

To get the \( Y_c^t \) that maximizes \( I(Y_c^t, L) \) in (6), we use the method in Mahadevan and Vasconcelos (2008). It adopts the dynamic texture (DT) feature and constructs foreground and background models at an initial frame. The models are updated over time via an autoregressive generative process. With the assumption that the observations are Gaussian, the computation of (6) requires the evaluation of the KL divergence between DT features, which is efficiently solved by Chan and Vasconcelos (2004).

The output of Mahadevan and Vasconcelos (2008) is the background probability map. The higher the values are, the more likely the pixels are background. To get the foreground probability map, we subtract the background probability map from one:

\[
M_f(Y_f^{(x,y)}) = 1 - \frac{1}{N_f}I(Y_f^{(x,y)}, L),
\]

where \( N_f = \max I(Y_f^{(x,y)}, L) \) is a normalization factor.

An example foreground probability map is shown in Fig. 2(c) and Fig. 3(b).

4.2. \( \min H(Y_i|Y_c^t) \): Saliency Detection

Saliency detection plays an important role in human vision system [Vince and Gallant (2000)] and in improving tracking accuracy [Hong et al. (2015)].

Given the surrounding area \( Y_c^t \), the saliency is defined as the minimum uncertainty of the center region \( Y_i \) in Li et al. (2009). Such uncertainty is typically measured by the conditional entropy, \( H(Y_i|Y_c^t) \). Then, by minimizing the conditional entropy, we can detect a saliency region in an image. For example, the center \( Y_i \) yields the highest saliency if \( Y_i \) can be completely determined by its surroundings \( Y_c^t \) (minimum uncertainty). In this case, \( H(Y_i|Y_c^t) \) is zero, which is the minimum value of conditional entropy. Conversely, the center \( Y_i \) yields the lowest saliency if \( Y_i \) and \( Y_c^t \) are totally independent each other (maximum uncertainty). In this case, \( H(Y_i|Y_c^t) \) is equal to \( H(Y_i) \), which is the maximum value of conditional entropy.

To get \( Y_i \) that minimizes \( H(Y_i|Y_c^t) \) in (7), we follow the approach proposed in Li et al. (2009). However, getting a solution of (7) is computationally intractable. To solve this problem, the approach assumes that \( Y_i \) and \( Y_c^t \) consist of multivariate Gaussian data, and approximates \( H(Y_i|Y_c^t) \) by the lossy coding length. When \( H(Y_i|Y_c^t) \) is represented by the lossy coding length, we have a closed solution of (7). Please refer to Li et al. (2009) for details.

The output of the method is the saliency probability map:

\[
M_s(Y_s^{(x,y)}) = 1 - \frac{1}{N_s}H(Y_s^{(x,y)}|Y_c^t),
\]

where \( Y_s^{(x,y)} \) denotes the observation at the \((x, y)\) position and \( N_s = \max H(Y_s^{(x,y)}|Y_c^t) \) is the normalization factor.
An example of saliency probability map is shown in Fig. 2(c) and Fig. 4(b).

4.3. min \[H(Y_t) + H(Y_{i}^c)\] : Illumination-Invariant Observations (Foreground+Background)

With the entropy minimization, our method changes an original image into an illumination-invariant one, as shown in Fig. 2(b). The illumination-invariant image can be made as follows.

In the 2D log-chromaticity space, the image pixels under illumination changes approximately form a straight line. If all these pixels are projected onto a line perpendicular to the straight line, the projected pixels become illumination-invariant [Finlayson et al. (2004)]. The key idea of Finlayson et al. (2004) is the observation that if the pixels are projected in the perpendicular direction, the histogram has a broader distribution, which yields lower entropy. Otherwise, the histogram has a set of sharp peaks, which yields high entropy. Hence we search for the best rotation degree \( \hat{\theta} \), in which the projected pixels minimize the entropy:

\[
\hat{\theta} = \arg \min_{\theta} H(\text{Proj}_\theta(Y_t)),
\]

where \( \text{Proj}_\theta() \) denotes the projection function with the rotation degree \( \theta \).

The output of the method is the intrinsic probability map:

\[
M_i(Y_t^{(x,y)}) = 1 - \frac{1}{N_i} H(\text{Proj}_\theta(Y_t^{(x,y)})),
\]

where \( N_i = \max H(\text{Proj}_\theta(Y_t)) \) is the normalization factor. An example of intrinsic probability map is shown in Fig. 2(c).

Fig. 5(b) shows the effect of our Illumination-Invariant preprocessing.

Algorithm 1 Leveraging Observation Uncertainty (LOU) for Tracking

Input: Original observation, \( Y_t \)
Output: Preprocessed observation, \( Y_t^{(\text{best})} \)

1: Estimating Probability Maps
2:  
3:  
4:  
5:  
6: Preprocessing the Observation
7:  

5. Robust Visual Tracking Augmented by Preprocessing

For efficiency reasons, our method independently finds three probability maps that achieve the goals of (6), (7), and (8). Note that the probability maps are obtained by using the methods explained in Sections 4.2, 4.1, and 4.3. Then a single probability map is estimated by averaging over these probability maps:

\[
M_a(Y_t^{(x,y)}) = w_f M_f(Y_t^{(x,y)}) + w_s M_s(Y_t^{(x,y)}) + w_i M_i(Y_t^{(x,y)}),
\]

where \( w_f \), \( w_s \), and \( w_i \) are weight parameters satisfying \( w_f + w_s + w_i = 1 \). We assume that each probability map equally contributes to the observation thus \( w_f = w_s = w_i = \frac{1}{3} \).
Even with this assumption, we empirically obtained good observation images that greatly improve the tracking performance. In Sec 7.1.2, we show how each part contributes to the final result with experimental results.

The final observation image is generated by masking the original observations / raw image with the averaged probability map. To that end, the original observation is first projected into the HSV color space [Perez et al. (2002)] and masked with the map only using the value channel:

$$Y_{I}^{(h, v, z)}(V) = Y_{I}^{(h, v, z)}(V) * M_{I}(Y_{I}^{(h, v, z)}),$$

where $Y_{I}^{(h, v, z)}(V)$ denotes the observation at the $(x, y)$ position of the value channel. An example of a best observation image and its probability are shown in Fig. 2(d).

We can employ many trackers and improve their tracking performance with our preprocessed observation images. However, for the demonstration, we select 4 top visual trackers as baselines according to the recent survey paper [Wu et al. (2013)], namely MEEM [Zhang et al. (2014a), SCM Zhong et al. (2012), STRUCK Hare et al. (2011), and ASLA Jia et al. (2012)]. We thus run these trackers with our preprocessed observation images as input, coining these enhanced versions MEEM+, SCM+, STRUCK+, and ASLA+, respectively. We will also refer to such tracker+ versions that use our preprocessed observation as Leveraging Observation Uncertainty (LOU) trackers. Algorithm 1 illustrates the whole process of our method.

6. Proof of Lemma 1

A good observation is available if $Y_{I}$ maximizes mutual information with the foreground label, $I(Y_{I}, l = f)$, whereas $Y_{I}$ minimizes $I(Y_{I}, l = f)$. So the observation quality is proportional to $I(Y_{I}, L) - I(Y_{I}, L)$. If $l = b$, the quality is proportional to $I(Y_{I}, L) - I(Y_{I}, L)$. So generally we can say that the quality is proportional to $I(Y_{I}, L) - I(Y_{I}, L)$, which has the upper bound like

$$|l(Y_{I}, L) - l(Y_{I}, L)|$$

$$= |H(Y_{I}) - H(Y_{I}) - H(Y_{I})|$$

$$
\leq |H(Y_{I}) - H(Y_{I})| + |H(Y_{I}) - H(Y_{I})|
$$

where $H(Y_{I})$ denotes the observation entropy. Because $|H(A) - H(B)| \leq H(A, B) - H(B)$ as derived in [Zhang 2007], (16) can be reformulated as

$$|l(Y_{I}, L) - l(Y_{I}, L)|$$

$$\leq |H(Y_{I}) - H(Y_{I})| + |H(Y_{I}) - l(Y_{I}, L)|$$

By using the relation $H(A, B) = H(A|B) + H(B)$, (17) is changed to

$$|l(Y_{I}, L) - l(Y_{I}, L)|$$

$$\leq |H(Y_{I}|Y_{I}) + |H(Y_{I}|L) - l(Y_{I}, L)|$$

$$\leq |H(Y_{I}|Y_{I}) + l(Y_{I}|L)| + H(Y_{I}|L).$$

Because $H(A|B) \leq H(A)$, the following equation can be derived:

$$|l(Y_{I}, L) - l(Y_{I}, L)|$$

$$\leq |H(Y_{I}|Y_{I}) + l(Y_{I}|L) + H(Y_{I}).$$

Now we can get the lower and upper bounds of the mutual information as follows:

$$I(Y_{I}, L) - \Delta l \leq I(Y_{I}, L) \leq I(Y_{I}, L) + \Delta l,$$

where

$$\Delta l = H(Y_{I}|Y_{I}) + \Delta l + H(Y_{I}).$$

The observation quality is determined by averaging the two mutual information bounds.

$$\frac{I(Y_{I}, L) - \Delta l + I(Y_{I}, L) + \Delta l}{2} = I(Y_{I}, L).$$

The uncertainty in observation quality is determined by computing the difference between the two bounds.

$$I(Y_{I}, L) + \Delta l - I(Y_{I}, L) + \Delta l = 2\Delta l.$$

7. Experiments

Our four trackers (MEEM+, SCM+, STRUCK+, and ASLA+) were compared with 29 recent tracking methods.
presented in Wu et al. (2013). Due to space limitations, we only depict the results of 11 top visual trackers though, which are MEEM Zhang et al. (2014a), SCM Zhong et al. (2012), STRUCK Hare et al. (2011), ASLA Jia et al. (2012), TLD Kalal et al. (2010), CXT Dinh et al. (2011), VTD Kwon and Lee (2010), VTS Kwon and Lee (2011), CSK Henriques et al. (2012), LSK Liu et al. (2011), and MTT Zhang et al. (2012b). All tracking methods were evaluated using a total of 50 challenging sequences presented in Wu et al. (2013). Our tracker used fixed parameter settings in all the experiments. The result video is provided at https://youtu.be/SQEaB1u0T5s.

The evaluation metrics are the precision plot and the success plot. We measure the precision with the center location error, which is the Euclidean distance between the ground truth and the estimated centers. The precision plot is the percentage of frames where the center location error is within the given threshold Babenko et al. (2011). The success rate is the amount of successfully tracked frames (score > threshold), where the score is defined by the overlap ratio between the predicted bounding box $BB_p$ and the ground truth bounding box $BB_g$: $\frac{\text{area}(BB_p \cap BB_g)}{\text{area}(BB_p \cup BB_g)}$. The success plot depicts the success rate when the threshold varies from 0 to 1.
7.1. Analysis of Our Tracking Method

7.1.1. Performance of Plug-in

We compared our plugged-in methods, MEEM+, SCM+, STRUCK+, and ASLA+, with the original tracking methods, MEEM, SCM, STRUCK, and ASLA. As shown in Fig. 6, our plugged-in methods outperform the original tracking methods by more than 10% (SCM: 0.649 → 0.760, STRUCK : 0.656 → 0.725, ASLA : 0.532 → 0.682). These experimental results demonstrate that our preprocessed images help the original trackers to significantly enhance their tracking accuracy. Among MEEM+, SCM+, STRUCK+, and ASLA+, MEEM+ achieves the best performance both in terms of precision and success rate.

Because our preprocessing algorithms are very fast, the speed of MEEM+, SCM+, STRUCK+, and ASLA+ is similar to that of MEEM, SCM, STRUCK, and ASLA, respectively. MEEM+ perform at the speed of 8.5 FPS (MEEM: 12 FPS), STRUCK+ at the speed of 15.0 FPS (STRUCK: 20.2 FPS), and ASLA+ at the speed of 5.0 FPS (ASLA: 8.5 FPS). SCM+ tracks the target at the speed of 0.3 FPS (SCM: 0.51 FPS). The above tracking algorithms run on a PC with Intel i7 3770 CPU (3.4GHz).

7.1.2. Component-wise Performance of Our method

We analyzed the contribution of each preprocessing algorithm to the success of the whole tracking algorithm. For this analysis, we compared STRUCK+ with STRUCK_B, STRUCK_S, and STRUCK_I, which are STRUCK combined with Background subtracted observations, Saliency detection observations, and Illumination invariant observations, respectively. As shown in Fig. 7, STRUCK_B is the best, which indicates that background subtraction contributes the most to the tracking accuracy improvement. Maximizing the observation quality by background subtraction is more important than minimizing the observation uncertainty. Nonetheless, the improvements brought by saliency detection and illumination-invariance are not negligible, as STRUCK_S and STRUCK_I are also superior to the original STRUCK, which means that minimizing the observation uncertainty adds to the performance. The combination of all these preprocessing algorithms gives the best tracking performance, as demonstrated by STRUCK+.

7.2. Comparison with Other Tracking Methods

7.2.1. Quantitative Comparison

We compared our best tracker MEEM+ with conventional tracking methods. In Fig. 8, MEEM+ is the best in terms of the precision and success rate. The performance gap between MEEM+ and the second best methods, MEEM, SCM and STRUCK, exceeds 10%.

Tables 1 and 2 show the attribute-based performance of several tracking algorithms. For this evaluation, the test sequences were assigned attributes, from a list of 11. For example, 25 sequences have an illumination variation attribute in the sense that the illumination of the target varies substantially during the sequence. Note that a sequence
Table 1: Attribute-based Performance in terms of average precision. Bold indicates the best results. Numbers in () indicate the improvement rate, which is computed by \( \frac{\text{Improved precision} - \text{Original precision}}{\text{Original precision}} \times 100\%. \) Please note that we evaluated 30 recent tracking methods on 50 test sequences but only report results of the top 11 tracking methods.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>MTT</th>
<th>LSK</th>
<th>CSK</th>
<th>VTS</th>
<th>VTD</th>
<th>CXT</th>
<th>TLD</th>
<th>ASLA+</th>
<th>STRUCK+</th>
<th>SCM+</th>
<th>MEEM+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illumination variation (25)</td>
<td>0.351</td>
<td>0.449</td>
<td>0.481</td>
<td>0.573</td>
<td>0.557</td>
<td>0.501</td>
<td>0.537</td>
<td>0.517</td>
<td>0.558</td>
<td>0.594</td>
<td>0.778</td>
</tr>
<tr>
<td>Out-of-plane rotation (39)</td>
<td>0.473</td>
<td>0.525</td>
<td>0.540</td>
<td>0.604</td>
<td>0.620</td>
<td>0.574</td>
<td>0.596</td>
<td>0.518</td>
<td>0.597</td>
<td>0.618</td>
<td>0.853</td>
</tr>
<tr>
<td>Scale variation (28)</td>
<td>0.461</td>
<td>0.480</td>
<td>0.503</td>
<td>0.582</td>
<td>0.597</td>
<td>0.550</td>
<td>0.606</td>
<td>0.552</td>
<td>0.639</td>
<td>0.672</td>
<td>0.808</td>
</tr>
<tr>
<td>Occlusion (29)</td>
<td>0.426</td>
<td>0.534</td>
<td>0.500</td>
<td>0.534</td>
<td>0.545</td>
<td>0.491</td>
<td>0.563</td>
<td>0.460</td>
<td>0.564</td>
<td>0.640</td>
<td>0.814</td>
</tr>
<tr>
<td>Deformation (19)</td>
<td>0.332</td>
<td>0.481</td>
<td>0.476</td>
<td>0.487</td>
<td>0.501</td>
<td>0.422</td>
<td>0.512</td>
<td>0.445</td>
<td>0.521</td>
<td>0.586</td>
<td>0.859</td>
</tr>
<tr>
<td>Motion blur (12)</td>
<td>0.308</td>
<td>0.324</td>
<td>0.342</td>
<td>0.375</td>
<td>0.375</td>
<td>0.509</td>
<td>0.518</td>
<td>0.278</td>
<td>0.551</td>
<td>0.339</td>
<td>0.740</td>
</tr>
<tr>
<td>Fast motion (17)</td>
<td>0.401</td>
<td>0.375</td>
<td>0.381</td>
<td>0.353</td>
<td>0.352</td>
<td>0.515</td>
<td>0.551</td>
<td>0.253</td>
<td>0.604</td>
<td>0.333</td>
<td>0.757</td>
</tr>
<tr>
<td>In-plane rotation (31)</td>
<td>0.522</td>
<td>0.534</td>
<td>0.547</td>
<td>0.579</td>
<td>0.599</td>
<td>0.610</td>
<td>0.584</td>
<td>0.511</td>
<td>0.617</td>
<td>0.597</td>
<td>0.809</td>
</tr>
<tr>
<td>Out of view (6)</td>
<td>0.374</td>
<td>0.515</td>
<td>0.379</td>
<td>0.455</td>
<td>0.462</td>
<td>0.510</td>
<td>0.576</td>
<td>0.333</td>
<td>0.539</td>
<td>0.429</td>
<td>0.730</td>
</tr>
<tr>
<td>Background clutter (21)</td>
<td>0.424</td>
<td>0.504</td>
<td>0.585</td>
<td>0.578</td>
<td>0.571</td>
<td>0.443</td>
<td>0.428</td>
<td>0.496</td>
<td>0.585</td>
<td>0.578</td>
<td>0.808</td>
</tr>
<tr>
<td>Low resolution (4)</td>
<td>0.510</td>
<td>0.304</td>
<td>0.411</td>
<td>0.187</td>
<td>0.168</td>
<td>0.371</td>
<td>0.549</td>
<td>0.156</td>
<td>0.545</td>
<td>0.305</td>
<td>0.494</td>
</tr>
<tr>
<td>Whole</td>
<td>0.475</td>
<td>0.500</td>
<td>0.545</td>
<td>0.575</td>
<td>0.575</td>
<td>0.608</td>
<td>0.532</td>
<td>0.656</td>
<td>0.649</td>
<td>0.840</td>
<td>0.885</td>
</tr>
</tbody>
</table>

Table 2: Attribute-based Performance in terms of average success rate. Bold indicates the best results. Numbers in () indicate the improvement rate, which is computed by \( \frac{\text{Improved success rate} - \text{Original success rate}}{\text{Original success rate}} \times 100\% \).

<table>
<thead>
<tr>
<th>Attribute</th>
<th>MTT</th>
<th>LSK</th>
<th>CSK</th>
<th>VTS</th>
<th>VTD</th>
<th>CXT</th>
<th>TLD</th>
<th>ASLA+</th>
<th>STRUCK+</th>
<th>SCM+</th>
<th>MEEM+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illumination variation (25)</td>
<td>0.305</td>
<td>0.311</td>
<td>0.369</td>
<td>0.429</td>
<td>0.42</td>
<td>0.368</td>
<td>0.399</td>
<td>0.429</td>
<td>0.428</td>
<td>0.475</td>
<td>0.548</td>
</tr>
<tr>
<td>Out-of-plane rotation (39)</td>
<td>0.362</td>
<td>0.400</td>
<td>0.386</td>
<td>0.425</td>
<td>0.434</td>
<td>0.418</td>
<td>0.420</td>
<td>0.422</td>
<td>0.432</td>
<td>0.470</td>
<td>0.569</td>
</tr>
<tr>
<td>Scale variation (28)</td>
<td>0.348</td>
<td>0.373</td>
<td>0.350</td>
<td>0.400</td>
<td>0.405</td>
<td>0.389</td>
<td>0.421</td>
<td>0.452</td>
<td>0.425</td>
<td>0.518</td>
<td>0.510</td>
</tr>
<tr>
<td>Occlusion (29)</td>
<td>0.342</td>
<td>0.409</td>
<td>0.365</td>
<td>0.398</td>
<td>0.403</td>
<td>0.372</td>
<td>0.402</td>
<td>0.376</td>
<td>0.413</td>
<td>0.437</td>
<td>0.563</td>
</tr>
<tr>
<td>Deformation (19)</td>
<td>0.280</td>
<td>0.377</td>
<td>0.343</td>
<td>0.368</td>
<td>0.377</td>
<td>0.324</td>
<td>0.378</td>
<td>0.372</td>
<td>0.393</td>
<td>0.448</td>
<td>0.582</td>
</tr>
<tr>
<td>Motion blur (12)</td>
<td>0.274</td>
<td>0.302</td>
<td>0.305</td>
<td>0.304</td>
<td>0.309</td>
<td>0.369</td>
<td>0.404</td>
<td>0.258</td>
<td>0.433</td>
<td>0.298</td>
<td>0.565</td>
</tr>
<tr>
<td>Fast motion (17)</td>
<td>0.333</td>
<td>0.328</td>
<td>0.316</td>
<td>0.300</td>
<td>0.302</td>
<td>0.388</td>
<td>0.417</td>
<td>0.247</td>
<td>0.462</td>
<td>0.296</td>
<td>0.568</td>
</tr>
<tr>
<td>In-plane rotation (31)</td>
<td>0.395</td>
<td>0.413</td>
<td>0.399</td>
<td>0.416</td>
<td>0.430</td>
<td>0.452</td>
<td>0.416</td>
<td>0.425</td>
<td>0.444</td>
<td>0.458</td>
<td>0.535</td>
</tr>
<tr>
<td>Out of view (6)</td>
<td>0.342</td>
<td>0.430</td>
<td>0.349</td>
<td>0.443</td>
<td>0.446</td>
<td>0.427</td>
<td>0.457</td>
<td>0.312</td>
<td>0.459</td>
<td>0.361</td>
<td>0.597</td>
</tr>
<tr>
<td>Background clutter (21)</td>
<td>0.337</td>
<td>0.388</td>
<td>0.421</td>
<td>0.428</td>
<td>0.425</td>
<td>0.338</td>
<td>0.345</td>
<td>0.408</td>
<td>0.458</td>
<td>0.450</td>
<td>0.578</td>
</tr>
<tr>
<td>Low resolution (4)</td>
<td>0.389</td>
<td>0.235</td>
<td>0.350</td>
<td>0.166</td>
<td>0.177</td>
<td>0.312</td>
<td>0.309</td>
<td>0.157</td>
<td>0.372</td>
<td>0.279</td>
<td>0.367</td>
</tr>
<tr>
<td>Whole</td>
<td>0.376</td>
<td>0.395</td>
<td>0.399</td>
<td>0.416</td>
<td>0.416</td>
<td>0.426</td>
<td>0.437</td>
<td>0.434</td>
<td>0.474</td>
<td>0.499</td>
<td>0.572</td>
</tr>
</tbody>
</table>

can have several attributes, i.e. they pose multiple challenges to the trackers. As demonstrated in Tables 1 and 2, our trackers, MEEM+, SCM+, STRUCK+, and ASLA+ significantly outperform conventional tracking methods for several of these challenges, in terms of the average precision and average success rate. Our trackers are robust to many different attributes because the preprocessing algorithms reduce their effect on the tracking.

In most attributes, we can observe that the proposed tracker, MEEM+, outperforms conventional trackers with large gaps. MEEM+ is weak only to the low resolution attribute included in test sequences. In the low resolution images, the effect of preprocessing is relatively low.

7.2. Qualitative Comparison

Fig. 9(a), Fig. 11(a), and Fig. 12(a) show tracking results of SCM, ASLA, STRUCK, and our method. Although there are severe obstacles such as occlusions, illumination changes, and pose variations, our method successfully tracked the target because these issues were mitigated with the help of the preprocessed observations, as shown in Fig. 9(b), Fig. 11(b), and Fig. 12(b). On the other hand, the SCM, ASLA, and STRUCK trackers easily drift towards the background due to the aforementioned issues and failed to recover from there.

8. Discussion

- During the preprocessing step, there may be risks that the truly useful contents are eliminated as well,
skiing seq.  motorRolling seq.  soccer seq.  bolt seq.

(a) Tracking results

(b) Corresponding preprocessed observation

Figure 9: Qualitative Comparison 1. The red, green, blue, and white boxes represent the tracking results of SCM, ASLA, STRUCK, and our method, respectively.

Figure 10: Plugged by CNN based tracking method especially when the quality of the video sequences are extremely low. To avoid this problem, in this paper, we get the regression results from the preprocessing step rather than the binary ones. In the binary case, the useful contents could be removed entirely and completely. In the regression case, however, the contents can survive even though they are wrongly considered as background. But, in this case, the contents have lower weights than others, which are considered as foreground.

- We further conduct a empirical study on the effects of the proposed method on other different kinds of trackers. For this study, our pre-processed image sequences are fed into a CNN based tracker [Wang et al. 2015] called FCN+. Then we compared FCN+ with the original method, namely FCN. Fig. 10 shows the precision and the success rate of FCN+ and FCN.

As demonstrated by Fig. 10 our approach could enhance a wide variety of trackers.

- We selected specific methods to fulfill the goals of different optimization problems presented in Section 4. Notably objective functions of the specific methods are equal to those of ours. For example, (6) can be achieved by a specific method [Mahadevan and Vasconcelos 2008], which object function is equal to (6). (7) is achieved by [Li et al. 2009], which object function is the same as that of (7). Similarly the objective function of (8) is same as that of [Finlayson et al. 2004].

- Please note that our main contribution is not to just use three algorithms to enhance the images for tracking purposes but to find out theoretical reasons why these three algorithms should be used for robust visual tracking. There are many other preprocessing algorithms and different combinations of them. We argue that, among a lot of preprocessing algorithms, only three specific preprocessing methods presented in this paper and their combination can maximize the image quality and minimize the image uncertainty.

9. Conclusion and Discussion

We proposed to improve images used for tracking by maximizing the observation quality and minimizing the
observation uncertainty. We theoretically showed that maximizing the observation quality is well served by background subtraction and that minimizing the observation uncertainty is helped by saliency detection and the extraction of illumination-invariant intensity information. We experimentally demonstrated that the accuracy of tracking methods can be greatly improved by our preprocessing step.

Our preprocessing algorithms are typically robust to the illumination change, background clutters, and noises. The algorithms, however, do not explicitly resolve the occlusion problems. Hence, failure cases occur mainly when there exists severe occlusions. Although our method fails to track the target in the aforementioned case, the tracking accuracy of the method is better than or at least similar to that of a baseline tracker plugged into our method. Empirically, our approach does not degrade the baseline trackers at least. Note that applications are not limited to visual tracking. The idea of leveraging the observation uncertainty can be used for other tasks such as object detection and object classification.

Acknowledgments

This Research was supported by the Chung-Ang University Research Scholarship Grants in 2016.
References

Babenko, B., Yang, M., Belongie, S., 2009. Visual tracking with online multiple instance learning. CVPR.


Dame, A., Marchand, E., 2010. Accurate real-time tracking using mutual information. ISMAR.


Godec, M., Roth, P.M., Bischof, H., 2011. Hough-based tracking of non-rigid objects. ICCV.

H. Grabner, C.L., Bischof, H., 2008. Semi-supervised online boosting for robust tracking. ECCV.

Han, B., Davis, L., 2005. On-line density-based appearance modeling for object tracking. ICCV.

Hare, S., Saffari, A., Torr, P.H.S., 2011. Struck: Structured output tracking with kernels. ICCV.


Hong, S., Han, B., 2014. Visual tracking by sampling tree-structured graphical models. ECCV.

Hong, S., You, T., Kwak, S., Han, B., 2015. Online tracking by learning discriminative saliency map with convolutional neural network. ICML.


Kalal, Z., Matas, J., Mikolajczyk, K., 2010. P-n learning: Bootstrapping binary classifiers by structural constraints. CVPR.


Kim, Z., 2008. Real time object tracking based on dynamic feature grouping with background subtraction. CVPR.


