# Robust Object Tracking via Locality Sensitive Histograms

Shengfeng He, *Member, IEEE,* Rynson W.H. Lau, *Senior Member, IEEE,* Qingxiong Yang, *Member, IEEE,* Jiang Wang, Ming-Hsuan Yang, *Senior Member, IEEE*

***Abstract*—This paper presents a novel locality sensitive histogram (LSH) algorithm for visual tracking. Unlike the conventional image histogram that counts the frequency of occurrence of each intensity value by adding ones to the corresponding bin, a locality sensitive histogram is computed at each pixel location and a floating-point value is added to the corresponding bin for each occurrence of an intensity value. The floating-point value reduces exponentially with respect to the distance to the pixel location where the histogram is computed. An efficient algorithm is proposed that enables the locality sensitive histograms to be computed in time linear in the image size and the number of bins. In addition, this efficient algorithm can be extended to exploit color images. A robust tracking framework based on the locality sensitive histograms is proposed, which consists of two main components: a new feature for tracking that is robust to illumination change and a novel multi-region tracking algorithm that runs in real-time even with hundreds of regions. Extensive experiments demonstrate that the proposed tracking framework outperforms the state-of-the-art methods in challenging scenarios, especially when the illumination changes dramatically. Evaluation using the latest benchmark shows that our algorithm is the top performer.**

***Index Terms*—locality sensitive histograms, visual tracking, illumination invariant, multi-region tracking.**

## I. Introduction

Histograms are one of the most important statistical tools for image analysis and widely used in various applications. One application is to model object appearance for visual tracking. The main issue of robust visual tracking is to handle appearance changes of the target object. While numerous algorithms have been proposed with demonstrated success, it remains a challenging task to develop a tracking algorithm that is both accurate and efficient. In order to address the challenging factors of appearance changes in visual tracking, various features and models have been proposed to represent target objects.

In this paper, we focus on handling illumination variations and occlusions problems with a multi-region representation. We propose a novel locality sensitive histogram (LSH) algorithm that takes into account contributions from every pixel in an image, instead of from pixels inside a local neighborhood only like the local histogram algorithm. It operates in a way similar to conventional image histograms. However, instead of counting the frequency of occurrences of each intensity value by adding ones to the corresponding bin, a floating-point value is added to the bin for each occurrence of the intensity value. The floating-point value reduces exponentially with respect to the distance to the pixel location where the locality sensitive histogram is computed. Thus, the proposed histogram is more suitable for applications such as visual tracking, which assigns lower weights to pixels further away from the target center (as these pixels are more likely to contain background information or occluding objects, and hence their contributions to the histogram should be reduced).

The proposed histogram has an $O(NB)$ complexity, where $N$ is the number of pixels and $B$ is the number of bins. This facilitates a framework for real-time object tracking. In addition, we show that the proposed histogram can be efficiently applied to color images. The proposed tracking framework effectively deals with drastic illumination change by extracting dense illumination invariant features using the proposed locality sensitive histograms. It also handles significant pose and scale variation, occlusion and visual drifts, out of plane rotation and abrupt motion, and background clutters with the use of a novel multi-region tracking algorithm. Unlike existing multi-region trackers that need to represent a target object with a limited number of non-overlapping regions due to the use of rectangular local histograms, the proposed tracking algorithm efficiently describes a target object with a large number of overlapping regions using the LSHs, which take into account contributions from every pixel adaptively. This unique property facilitates robust multi-region tracking, and the efficiency of the LSHs enables the proposed algorithm to track a target object in real time. Evaluations are conducted on two datasets. The first dataset contains 20 image sequences to examine the proposed method using color and illumination invariant features. We then use a recent benchmark dataset [35] with 50 image sequences to evaluate the proposed tracker using color and illumination invariant features, showing that it outperforms the state of the art.

The main contributions of this paper are:

- We propose a novel histogram that takes contributions from every image pixel into account and an algorithm

Shengfeng He, Rynson W.H. Lau, Qingxiong Yang, and Jiang Wang are with the Department of Computer Science, City University of Hong Kong, Hong Kong. E-mail: shengfeng_he@yahoo.com, {rynson.lau, qiyang, jiangwang6}@cityu.edu.hk

Ming-Hsuan Yang is with the School of Engineering, University of California, Merced, CA 95344. E-mail: mhyang@ucmerced.edu

to compute intensity and color histograms efficiently.
- We propose an efficient illumination invariant feature for visual tracking.
- We propose a multi-region tracking algorithm that outperforms state-of-the-art methods both in terms of accuracy and speed.

The rest of this paper is organized as follows. We first discuss the related works on visual tracking, and put this work in proper context in Section II. Sections III and IV present the proposed locality sensitive histograms and novel illumination invariant features for visual tracking. A multi-region tracking algorithm based on the proposed locality sensitive histograms is presented in Section V. Section VI summarizes our evaluation experiments. We conclude this paper with remarks on our future work in Section VII.

A preliminary version of this work was presented in [13]. This paper revises on the preliminary version in the following main ways. First, we comprehensively review the most related tracking algorithms in Section II. Second, we extend the proposed LSH to color images in an efficient manner in Section III-B. Third, we evaluate the proposed trackers with more experimental validations and analyses in Section VI-A. Finally, we compare the proposed trackers on the latest benchmark [35] using the evaluation metrics proposed in the benchmark [35] and the latest survey paper [31] in Section VI-B.

## II. Related Work

In this section, we briefly review the most related algorithms on tracking. Comprehensive literature review on this topic can be found in [36], [31].

Tracking methods often fail in the presence of large illumination change, heavy occlusion, pose variation and motion blur. Different features or models have been used to address these problems.

**Illumination variation** may drastically change object appearances. It has been shown that an object taken at the same pose but under different illumination conditions cannot be uniquely identified as being the same object or different ones [17]. To deal with this problem, numerous methods have been proposed based on illumination invariant features [7]. Early visual tracking methods represent objects with contours [16] with success when the brightness constancy assumption holds. The eigentracking algorithm [6] operates on the subspace constancy model learned from a set of training images to account for appearance change. Recently, Harr-like features and online subspace models have been used in object tracking to deal with large lighting variation [20], [3], [4], [32]. However, these methods typically entail time-consuming operations due to the use of mixture models or optimization formulations.

**Occlusion** is mainly handled by multi-region representations. The fragment-based tracking method [1] divides the target object into several regions and represents them with multiple local histograms. The possible positions of each patch are voted and combined to determine the target object movement. However, computing multiple local histograms and the vote map can be time consuming even with the integral histogram [25]. As a trade-off between accuracy and speed, the fragment-based method [1] uses up to 40 regions to represent the target object, causing jitter effects. To account for large appearance changes, recent multi-region trackers combine local and global representations by adapting region locations to geometric variations, instead of using a fixed grid layout [1]. In [24], each target object is modeled by a small number of rectangular blocks, with the spatial configuration being adaptively determined. In [20], [32], a fixed number of object parts are dynamically updated to account for appearance/shape changes. All these methods achieve better accuracy at the expense of speed.

Other than dealing with specific problems, recent tracking algorithms focus on representation schemes using either discriminative or generative models, to account for target object appearance variations.

**Discriminative algorithms** consider visual tracking as a binary classification problem to differentiate the foreground region from the background. A classifier is used to estimate the target object location by searching for the maximum classification score within a local neighborhood of the previously known position, and the classifier is usually updated to adapt to appearance change. Avidan [2] introduces a support vector machine (SVM) classifier into an optical flow framework for visual tracking. However, the SVM classifier needs to be trained offline using a large number of examples for a particular class of objects. In [11], Grabner et al. propose a semi-supervised approach where the training samples from the first frame are considered as correctly labeled and all the others as unlabeled within a boosting framework. Babenko et al. [3] present an online multiple instance learning algorithm for visual tracking, which alleviates the drift problem. In [12], Hare et al. introduce a structured output regression method for visual tracking by predicting the change in object location across frames. Kalal et al. [19] utilize the constraints underlying both labeled and unlabeled data to train a classifier for distinguishing the target object from the background. Gao et al. [10] apply transfer learning to assist the final decision by learning prior knowledge on auxiliary examples. In general, if a large number of labeled training examples are available, discriminative tracking approaches tend to perform well. However, online discriminative tracking methods usually do not have sufficient amount of labeled data at the outset.

**Generative algorithms** represent a target object in a particular feature space, and search for the best matching score within an image region. In recent years, generative methods with appearance update have been shown to be effective for visual tracking. Existing histogram-based methods [8], [25], [1] use image statistics to account for large appearance change at the expense of losing some spatial information. To address this problem, Birchfield and Rangarajan [5] propose a spatiogram that combines the spatial mean and covariance of each bin. However, this method is less effective for dealing with heavy occlusion. In addition, it entails a high computational cost when using a high order histogram. Other features have been

used in generative appearance methods to represent target objects with adaptive update schemes. The IVT algorithm [27] represents target objects using a subspace model with incremental learning. Kwon et al. [21] utilize multiple motion and observation models to account for large appearance change caused by pose and illumination variation. The L1 tracker [4] represents the target object using a sparse linear combination of target templates and trivial templates. Multi-task sparse learning [40], [41] and low-rank sparse learning [39] are proposed with the integration of particle filters to exploit the interdependency between particles. This idea is then extended by multi-task multi-view tracking [15] to utilize the shared information between particles and views. Due to the efficient implementation, kernelized correlation filters [14] are exploited to model natural image translations, resulted in real-time tracking performance.

Despite the demonstrated success, these generative algorithms are less effective when heavy occlusion occurs. Although the multi-region based method [1] performs well in handling heavy occlusion, the adopted representation based on a number of local histograms is not updated and hence difficult to account for large appearance change caused simultaneously by motion, occlusion and illumination change. In contrast, the proposed algorithm facilitates the use of hundreds of overlapping regions efficiently to better account for large appearance change in visual tracking. In addition, the input of the traditional multi-region tracker [1] is limited to grayscale images, as the computational cost of the histogram increases exponentially with the number of dimensions. We propose an approach to represent target objects using color histograms in an efficient manner, while maintaining the most distinctive information of color images. Four evaluation metrics (proposed in the latest benchmark [35] and survey paper [31]) demonstrate that the proposed method is robust to different challenging factors.

## III. Locality Sensitive Histogram (LSH)

We first briefly introduces the proposed LSH algorithm for grayscale images. (Detailed explanation can be found in [13]). We then extend the proposed efficient algorithm to color images.

### A. LSHs for Grayscale Images

A local histogram records statistics within a region of a pixel, and is widely used in computer vision, graphics and multimedia applications. However, for applications such as visual tracking, pixels further away from the target center should be weighted less as they are more likely from the background or other occluding objects. Hence, their contributions to the histogram should be reduced. We propose a novel *locality sensitive histogram* (LSH) algorithm to address this problem. The LSH, $\mathbf{H}_p^E$, at pixel $p$ is computed by:

$$\mathbf{H}_p^E(b) = \sum_{q=1}^{W} \alpha^{|p-q|} \cdot Q(\mathbf{I}_q, b), \quad b = 1, \ldots, B, \qquad (1)$$

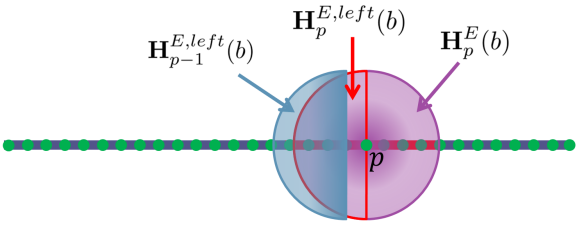



Fig. 1: Illustration of the proposed LSH. The purple region centered at pixel $p$ is the LSH to be computed. We separate the purple region into two parts. The left part (red line) can be computed by the previous left part (blue region) times $\alpha$, and adding 1 to the corresponding bin of pixel $p$. The right part can be computed in a similar way.

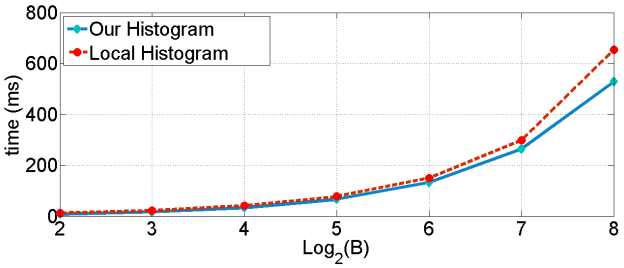



Fig. 2: Speed comparison on a 1-megapixel grayscale image w.r.t. the logarithm of the number of bins $B$. The time for computing the local histograms from the integral histograms and for computing the locality sensitive histograms is very close. Both can be computed in real time when the number of bins is small. For instance, when $B = 16$, the LSHs can be computed at 30 FPS.

where $\alpha \in (0, 1)$ is a parameter controlling the decreasing weight as a pixel moves away from the target center. Same as the local histogram, the computational complexity of the brute-force implementation of Eq. 1 is $O(WB)$ per pixel. However, similar to the integral histogram, the proposed LSH can be computed efficiently when the input image is 1D as:

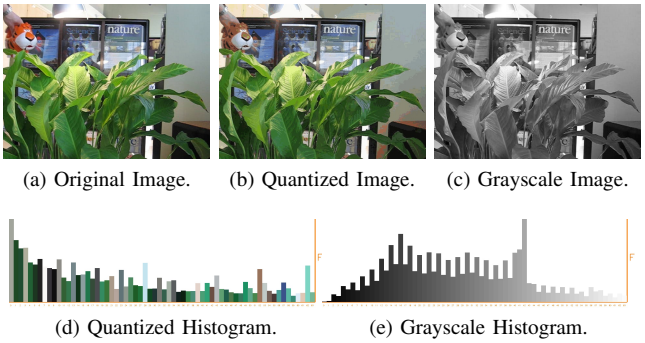
$$\mathbf{H}_p^E(b) = \mathbf{H}_p^{E,left}(b) + \mathbf{H}_p^{E,right}(b) - Q(\mathbf{I}_p, b), \qquad (2)$$

where

$$\mathbf{H}_p^{E,left}(b) = Q(\mathbf{I}_p, b) + \alpha \cdot \mathbf{H}_{p-1}^{E,left}(b), \qquad (3)$$

$$\mathbf{H}_p^{E,right}(b) = Q(\mathbf{I}_p, b) + \alpha \cdot \mathbf{H}_{p+1}^{E,right}(b). \qquad (4)$$

Based on Eqs. 3 and 4, pixels on the right of pixel $p$ do not contribute to the LSH on the left hand side $\mathbf{H}_p^{E,left}$, while pixels on the left of pixel $p$ do not contribute to the LSH on the right hand side $\mathbf{H}_p^{E,right}$. The summation of $\mathbf{H}_p^{E,left}$ and $\mathbf{H}_p^{E,right}$, however, combines the contribution from all pixels and the weight of the contribution drops exponentially with respect to the distance to pixel $p$. Clearly, only $B$ multiplications and $B$ additions are required at each pixel location in order to compute $\mathbf{H}_p^{E,left}$ (or $\mathbf{H}_p^{E,right}$). Thus, the computational complexity of the locality sensitive histogram is reduced to $O(B)$ per pixel.



(a) Original Image. (b) Quantized Image. (c) Grayscale Image.

(d) Quantized Histogram. (e) Grayscale Histogram.

Fig. 3: A quantized image (middle) retains sufficient distinctive colors, even through only 64 bins are used.

Fig. 4: Illumination invariant features. (a)-(c): input images with different illuminations. (d)-(f): the corresponding outputs. These three results are obtained with the same $\alpha$.

A special constraint is that $Q(\mathbf{I}_p, \cdot)$ in Eqs. 3 and 4 is a $B$-dimensional vector containing zero values except for the bin corresponding to the intensity value $\mathbf{I}_p$. Hence, the addition operation in Eqs. 3 and 4 can be removed except for the bin corresponding to $\mathbf{I}_p$. The proposed algorithm is illustrated in Figure 1.

The algorithm presented in Eqs. 2-4 is derived for 1D images. However, its extension to multi-dimensional images is straightforward. We simply perform the proposed 1D algorithm separately and sequentially in each dimension. Figure 2 shows the speed of computing the LSHs for a 1-megapixel 2D image with respect to the logarithm of the number of bins $B$. Note that the proposed algorithm is as efficient as the integral histogram method [26].

### B. LSHs for Color Images

While the proposed histogram captures locality intensity information in an efficient manner, intensity may not be distinctive enough for some vision tasks (e.g., object tracking in Figure 3c). One possible solution is to utilize color information. The color attribute has been shown to be effective in [9]. However, the histogram is known to be inappropriate for use with multi-dimensional data. The number of bins will increase exponentially when representing a color image in the full color space. Zivkovic and Krose [43] use only the H and S channels from the HSV color space for tracking, with a $8 \times 8$ histogram. Zhai and Shah [37] use only the luminance channel to reduce the computational cost. Although the number of bins is reduced, some distinctive information may be missing.

As the most representative colors in a natural image are the most frequently occurring ones, and typically cover only a fraction of the full color space, we selectively quantize the full RGB color space for object tracking. For each channel, the colors are first quantized to 12 values, reducing the number of colors to $12^3 = 1728$. Due to the sparsity nature of natural images, only a small portion of the full color space is used to represent an image. As a result, these 1728 colors are further reduced by considering only the most frequently occurring ones. Note that these steps are performed based on the histogram, and a simple histogram-based quantization is used for efficiency. This step is able to hugely reduce the number of colors. The number of colors is set to 64 in all the experiments, and these colors typically cover $80\%$ of the pixels in an image. The remaining pixels are replaced by the closest colors. The corresponding bins are determined by the first frame, and they will not change in the rest of the sequences.

Although this strategy is not suitable for cases with colorful and frequently changing backgrounds, it is robust enough for most of the real world cases, where the object is typically tracked within the same scene throughout the entire tracking process. Figure 3 shows one example result of the proposed quantization. Even through only 64 bins are used to represent a color image, the visual quality is maintained well. When compared with a grayscale image histogram (Figure 3e), Figure 3d shows that the histogram of the quantized image distributes well to all the bins and the color differences between bins are distinct. More importantly for tracking, the target object (e.g., tiger in Figure 3) can be represented more distinctively than using intensity. Color information, when available, is used as one of the features for object tracking in our experiments (Section VI).

## IV. Illumination Invariant Features

Images taken under different illumination conditions have drastic effects on object appearance. Under the assumption of affine illumination change, we can synthesize images of the scene presented in Figure 4(a) with the ones captured under different illumination conditions as shown in Figure 4(b) and 4(c). The intensity (or color) values are obviously not effective for image matching.

We propose a method to extract dense illumination invariant features based on an image transform. Let $\mathbf{I}_p$ and $\mathbf{I}'_p$ be the intensity values of pixel $p$ before and after an affine illumination change. We have:

$$\mathbf{I}'_p = \mathcal{A}_p(\mathbf{I}_p) = a_{1,p}\mathbf{I}_p + a_{2,p}, \tag{5}$$

where $a_{1,p}$ and $a_{2,p}$ are two parameters of the affine transform $\mathcal{A}_p$ at pixel $p$.

Let $\mathbf{H}_p^S$ be the histogram computed from a window $S_p$ centered at pixel $p$, and $b_p$ be the bin corresponding to intensity value $\mathbf{I}_p$. According to the definition of the histogram, the number of pixels in $S_p$ which intensity values fall within $[b_p - r_p, b_p + r_p]$ is:

$$\mathcal{I}_p = \sum_{b=b_p-r_p}^{b_p+r_p} \mathbf{H}_p^S(b), \tag{6}$$

where parameter $r_p$ controls the interval of integration at pixel $p$. If $r_p$ scales linearly with the illumination such that

$$r'_p = a_{1,p} r_p, \tag{7}$$

the integrated value $\mathcal{I}'_p$ obtained under a different illumination condition corresponds to the number of pixels with intensity values falling within $[a_{1,p}b_p + a_{2,p} - a_{1,p}r_p,\ a_{1,p}b_p + a_{2,p} + a_{1,p}r_p]$ = $[a_{1,p}(b_p - r_p) + a_{2,p},\ a_{1,p}(b_p + r_p) + a_{2,p}]$ = $[\mathcal{A}_p(b_p - r_p),\ \mathcal{A}_p(b_p + r_p)]$. If we ignore the quantization error, $\mathcal{I}'_p$ is equal to $\mathcal{I}_p$. Thus, $\mathcal{I}_p$ is independent of affine illumination change and can be used as an invariant feature under different illumination conditions as long as Eq. 7 holds. We set:

$$r_p = \kappa |\mathbf{I}_p - \bar{\mathbf{I}}_p|, \tag{8}$$

where $\kappa = 0.1$ is a constant. $\bar{\mathbf{I}}_p = \frac{1}{|S_p|}\sum_{q \in S_p} \mathbf{I}_p$ is the mean intensity value of window $S_p$. $|S_p|$ is the number of pixels in $S_p$. With an additional assumption that the affine illumination change is locally smooth so that the transform is the same for all pixels inside window $S_p$, we have:

$$\begin{aligned} r'_p &= \kappa |\mathbf{I}'_p - \bar{\mathbf{I}}'_p| \\ &= \kappa |a_{1,p}\mathbf{I}_p + a_{2,p} - \frac{1}{|S_p|}\sum_{q \in S_p}(a_{1,p}\mathbf{I}_q + a_{2,p})| \\ &= a_{1,p}\kappa |\mathbf{I}_p - \bar{\mathbf{I}}_p| \\ &= a_{1,p} r_p. \end{aligned} \tag{9}$$

As a result, Eq. 7 holds when interval $r_p$ is obtained adaptively from Eq. 8.

The proposed illumination invariance holds under the assumption that the affine illumination change is the same for all the pixels from which the traditional histogram is computed. In practice, however, it is inaccurate to define an exact local window inside which the affine illumination transform remains unchanged. We thus need to adopt an approximation by giving higher weights to pixels that are close to the center pixel and vice versa. Hence, we replace histogram $\mathbf{H}_p^S$ in Eq. 6 with the proposed locality sensitive histogram $\mathbf{H}_p^E$, which adaptively takes into account the contributions from all image pixels. In addition, we use a "soft" interval to reduce the quantization error, and thus Eq. 6 becomes:

$$\mathcal{I}_p = \sum_{b=1}^{B} \exp\left(-\frac{(b - b_p)^2}{2\max(\kappa, r_p)^2}\right) \cdot \mathbf{H}_p^E(b), \tag{10}$$

where $\bar{\mathbf{I}}_p = \sum_{b=1}^{B} \mathbf{H}_p^E(b) \cdot b$. Since $a_{2,p}$ is relatively small, $r_p$ can be replaced by $\kappa \mathbf{I}_p$. The invariant features computed from Figure 4(a)-(c) are presented in Figure 4(d)-(f). Unlike intensity values, they remain the same even under dramatic illumination change. This converted image form one of the inputs for the tracking algorithm proposed in Section V. (The other input is the quantized color image, when available, as discussed in Section III-B.) Note that if illumination invariant features are used, LSHs need to be computed twice, one for extracting illumination invariant features and one for tracking based on the converted image.

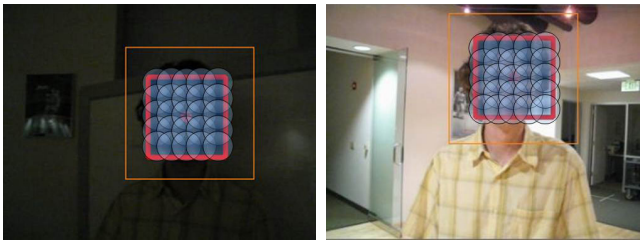
## V. Multi-Region Tracking

In [1], a multi-region tracking method is proposed based on integral histograms. While using multiple regions to represent a target object may capture the spatial information of the object (as opposed to tracking methods based on one region) to account for appearance change, it is less effective to use a large number of regions as the incurred computational cost of local analysis and region-by-region matching is prohibitively high. In this work, we exploit the proposed locality sensitive histogram for multi-region tracking, which has two major advantages. First, the proposed algorithm is able to represent and track objects using a large number of regions as matching multiple regions based on the LSHs can be computed rather efficiently. Second, as integral histograms do not support overlapping regions [1], the region size becomes smaller when more regions are used to represent objects. This, in turn, affects the tracking accuracy as insufficient visual information is contained in each region for matching. Since the proposed LSHs support overlapping regions, it is feasible to use a large number of regions with sufficient visual information. As such, the proposed tracking algorithm performs efficiently and accurately. In addition, the use of color (Section III-B) and illumination invariant features (Section IV) facilitates the proposed algorithm achieving robust tracking results.

### A. Tracking via Locality Sensitive Histograms

The proposed tracking algorithm represents a target object with multiple overlapping regions where each one describes some local configuration. The spatial relationship of these regions remains fixed and is used for region-to-region matching between the template and the potential target object in the current frame. The spacing between regions depends on the size of the target object and the pre-defined number of regions.

In the fragment-based tracking method [1], the kernel function is approximately by computing histograms from three rectangular regions of different sizes with different weights. This approximation scheme significantly increases the computational cost. In contrast, a weighting kernel is inherent in the proposed algorithm. The pixels within each region are weighted according to their locations from the region center. This facilitates the proposed algorithm achieving more robust matching results when the regions are partially occluded. Since object movements are typically non-ballistic, object tracking entails only searches within the nearby area of the current target location. Figure 5



(a) Template. (b) Current frame.

Fig. 5: Multi-region tracking. The target template is defined manually in the first frame. In each subsequent frame, the red rectangle indicates the tracking result, the blue circles show multiple regions of the target object, and the orange rectangle represents the search region.

shows one example of the proposed multi-region tracking algorithm, where the blue circles indicate the regions used to describe the target object. For ease of presentation, only a few regions are shown here. The orange rectangle indicates the search region of the current frame.

From the tracking result in the previous frame, we aim to locate the new target location within a search region. Similar to recent tracking-by-detection methods, we perform an exhaustive search within the search region, where every pixel is considered as a candidate target center. Each region at the candidate location is matched correspondingly against the template to produce a score, based on the Earth Mover's Distance (EMD) [28].

Although the conventional EMD algorithm is computationally expensive, the EMD between two normalized 1D histograms can be computed efficiently with time complexity linear in the number of bins [30]. It is equal to the $\ell_1$-distance between their cumulative histograms [34]. The proposed LSH is normalized as presented in Section III. The distance between histograms can be computed by

$$d(S_1, S_2) = \sum_{b=1}^{B} |C_1(b) - C_2(b)|, \tag{11}$$

where $S_1$ and $S_2$ are the corresponding regions of the template (centered at $p1$) and the candidate (centered at $p2$) in the current frame. In addition, $C_1$ and $C_2$ are the cumulative histograms of $\mathbf{H}^E_{p1}$ and $\mathbf{H}^E_{p2}$ defined by $C(b) = \sum_1^b \mathbf{H}^E_p(b)$.

Once the matching scores of all regions at a candidate location are computed, a vote map is obtained. Similar to the fragment-based tracking method, we use the least-median-squares estimator to accumulate all the votes. This scheme has been shown robust to occlusion, since it assumes that at least one quarter of the target object is visible, and the occluded regions are considered as outlier measurements. The new target location is the candidate location with the lowest joint score (as the vote map measures the dissimilarity between regions).

### B. Online Template Update

Visual tracking over a long period of time using a fixed template is not effective, as the object appearance is likely to change significantly. It is also likely to cause jitter and drift, as observed by [1]. To address these issues, we update the region histograms of the template incrementally. By using multiple regions for object representation, updating a fraction of them in each frame allows the template to adapt to appearance change and alleviates the tracking drift problem. Once the new target location is determined, the local histograms are updated as follows:

$$H^E_{p1}(\cdot) = H^E_{p2}(\cdot) \qquad \text{if } F_1 \cdot M < d(S_1, S_2) < F_2 \cdot M, \tag{12}$$

where $M$ is the median distance of all the regions at the new position. $F_1$ and $F_2$ are the forgetting factors, which define the appearance model update rate. In general, the regions with high dissimilarity are from the occluders or the background. Hence, we update only the regions with medium dissimilarity. In our implementation, we set the forgetting factors $F_1$ and $F_2$ as 0.99 and 1.01, respectively. For a target object with 1,000 regions, about 15 of them (1.5%) are updated in each frame.

## VI. Experiments

We have implemented the proposed tracking method in C language on a desktop computer with an Intel i5 3.3GHz CPU (using only a single core) and 8 GB RAM. We first evaluate the proposed tracker using color and illumination invariant features in a dataset containing 20 sequences. We then use a recent benchmark dataset [35] to evaluate the proposed tracker with two different features.

In both evaluations, we use the same parameters for all the sequences. Note that some of the sequences are included in both evaluations. These sequences share the same initial target setting. Theoretically, $\alpha$ should be set according to the space between regions. As 400 regions (to be discussed in Section VI-A1) are optimized to represent the target object, $\alpha$ is accordingly set to 0.915 for all the image sequences in our experiments to reduce overlapping between the LSHs. The number of bins is 16 for grayscale (or IIF) LSH, and 64 for color LSH. The searching radius is set to 25 pixels to handle fast motion, and the LSH is computed only within the search region.

### A. Evaluation 1: 20 Sequences

The experimental evaluations are carried out using 20 image sequences, which contain challenging factors, including drastic illumination change, pose and scale variation, heavy occlusion, background clutter, and fast motion. The frame sizes ranging from $300 \times 400$ to $800 \times 1000$. We compare the proposed tracking algorithm using illumination invariant features (**LSHT**) and color (**LSHT_C**) with 17 state-of-the-art methods (using implementations provided by the authors for fair comparisons). Three state-of-the-art multi-region based methods are evaluated including the fragment-based tracking method (**Frag**) [1], the articulating block

TABLE I: The average center location errors (in pixels) of the 20 sequences. The best and the second best performing methods are shown in red color and blue color, respectively. The total number of frames is 10,918. The entries in **TLD** marked with '– –' indicate that their values are not available as the algorithm loses track of the target object. The sequences with illumination changes are marked with a *.

| Sequence | LSHT | LSHT_C | L1T [4] | SPT [33] | CT [38] | Frag [1] | MIL [3] | Struck [12] | VTD [21] | TLD [19] | BHT [24] | LGT [32] | DFT [29] | MTT [40] | SCM [42] | ASLA [18] | VTS [22] | TGPR [10] | KCF [14] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Basketball | 9.5 | 8.5 | 14.9 | 20.0 | 62.9 | 25.2 | 102 | 165 | 8.6 | – – | 153 | 15.2 | 239 | 287 | 51.1 | 174 | 62.2 | 31.6 | 106 |
| Biker | 10.2 | 8.3 | 40.4 | 52.9 | 26.0 | 79.8 | 31.0 | 12.7 | 69.4 | – – | 30.1 | 85.5 | 46 | 76.1 | 13.5 | 35.8 | 39.4 | 13.3 | 13.0 |
| Bird | 8.3 | 8.6 | 57.8 | 14.1 | 17.1 | 26.6 | 17.2 | 23.9 | 61.9 | – – | 14.7 | 15.6 | 10.9 | 73.2 | 63.2 | 7.2 | 16.6 | 35.8 | 21.2 |
| Board | 10.5 | 11.5 | 174 | 52.6 | 34.7 | 25.2 | 35.0 | 34.6 | 17.8 | – – | 42.4 | 36.6 | 145 | 50.3 | 20.4 | 60.4 | 8.4 | 22.8 | 27.2 |
| Bolt | 6.4 | 6.0 | 186 | 72.0 | 9.0 | 73.2 | 8.1 | 8.4 | 17.7 | – – | 149 | 48.6 | 50.9 | 9.0 | 7.2 | 18.6 | 76.1 | 8.0 | 6.4 |
| Box | 12.0 | 10.9 | 77.5 | 169 | 107 | 65.8 | 109 | 10.6 | 114.1 | – – | 111 | 68.9 | 120 | 54.8 | 8.5 | 49.1 | 62.6 | 8.4 | 12.3 |
| Car* | 3.9 | 7.7 | 36.8 | 4.5 | 38.3 | 40.2 | 37.9 | 6.4 | 35.3 | 9.4 | 156 | 60.9 | 39.0 | 34.1 | 3.4 | 21.8 | 68.0 | 4.8 | 5.9 |
| Coupon | 5.4 | 5.4 | 69.7 | 6.1 | 17.1 | 35.7 | 18.6 | 5.7 | 65.3 | 7.9 | 45.7 | 21.9 | 5.5 | 4.5 | 7.8 | 5.3 | 28.9 | 7.0 | 6.3 |
| Crowds* | 5.3 | 8.7 | 15.3 | 433 | 351 | 435 | 465 | 5.0 | 380 | – – | 344 | 232 | 4.5 | 226 | 75.1 | 61.1 | 262 | 33.7 | 8.5 |
| David indoor* | 10.6 | 11.2 | 28.5 | 29.8 | 25.8 | 103 | 44.1 | 30.5 | 64.6 | 10.4 | 122 | 13.4 | 28.6 | 10.7 | 17.8 | 20.3 | 28.7 | 20.5 | 24.6 |
| Dragon baby | 20.0 | 18.2 | 72.8 | 49.7 | 30.8 | 51.1 | 48.6 | 59.1 | 42 | – – | 83.1 | 87 | 148 | 78.9 | 29.6 | 76.1 | 50.8 | 37.6 | 47.2 |
| Man* | 2.1 | 3.1 | 2.8 | 23.1 | 9.4 | 48.6 | 37.7 | 2.4 | 20.7 | 4.4 | 73.0 | 24.1 | 40.0 | 2.6 | 3.8 | 12.0 | 41.6 | 3.1 | 2.6 |
| Motor rolling | 13.1 | 10.8 | 187 | 76.2 | 198 | 110 | 178 | 84 | 148 | – – | 137 | 178 | 170 | 180 | 63.8 | 111 | 85.9 | 56.0 | 61.2 |
| Occluded face 2 | 4.0 | 4.0 | 20.2 | 37.8 | 13.2 | 15.5 | 15.3 | 16.4 | 15.9 | 11.5 | 44.5 | 30.9 | 23.8 | 16.8 | 8.3 | 15.3 | 10.3 | 9.8 | 12.4 |
| Shaking* | 19.5 | 20.8 | 36.0 | 103 | 41.0 | 186 | 15.8 | 42.3 | 18.3 | – – | 185 | 41.4 | 10.6 | 41.8 | 21.2 | 31.7 | 148 | 30.0 | 36.7 |
| Surfer | 7.4 | 6.8 | 122 | 78.2 | 78.6 | 150 | 128 | 14.6 | 93.3 | 5.4 | 93.3 | 188.1 | 143 | 93.1 | 23.0 | 164.4 | 83.4 | 16.5 | 13.4 |
| Sylvester | 18.3 | 14.2 | 25.4 | 42.2 | 5.4 | 10.4 | 10.9 | 5.1 | 8.1 | 13.1 | 11.3 | 28.8 | 40.1 | 6.3 | 8.1 | 13.2 | 10.3 | 7.0 | 7.5 |
| Tiger2 | 8.5 | 6.8 | 46.3 | 85.7 | 11.9 | 45.7 | 7.7 | 9.8 | 32.9 | – – | 66.8 | 27.9 | 33.6 | 25.5 | 9.1 | 18.9 | 39.8 | 10.0 | 10.3 |
| Trellis* | 8.3 | 11.8 | 15.5 | 18.8 | 67.3 | 100 | 65.8 | 22.9 | 32.4 | – – | 75.4 | 16.1 | 51.3 | 56.9 | 11.5 | 35.7 | 62.4 | 9.8 | 18.5 |
| Woman | 6.4 | 5.7 | 136 | 8.9 | 130 | 7.3 | 144 | 5.5 | 136 | – – | 71.9 | 19.3 | 8.6 | 154 | 30.8 | 43.5 | 8.3 | 16.5 | 7.5 |
| **Average** | 9.49 | 9.45 | 68.2 | 68.7 | 62.7 | 81.7 | 76.6 | 28.3 | 69.1 | – – | 101 | 62.0 | 68.0 | 74.0 | 23.9 | 48.8 | 59.7 | 19.1 | 22.4 |

TABLE II: The success rates (%) and the average frames per second (FPS) of the 20 sequences. The best and the second best performing methods are shown in red color and blue color, respectively. The total number of frames is 10,918. The sequences with illumination changes are marked with a *.

| Sequence | LSHT | LSHT_C | L1T [4] | SPT [33] | CT [38] | Frag [1] | MIL [3] | Struck [12] | VTD [21] | TLD [19] | BHT [24] | LGT [32] | DFT [29] | MTT [40] | SCM [42] | ASLA [18] | VTS [22] | TGPR [10] | KCF [14] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Basketball | 83 | 88 | 75 | 84 | 32 | 78 | 27 | 2 | 96 | 1 | 20 | 44 | 3 | 3 | 58 | 24 | 62 | 67 | 41 |
| Biker | 64 | 75 | 23 | 44 | 34 | 14 | 40 | 49 | 45 | 38 | 46 | 7 | 46 | 44 | 57 | 54 | 45 | 63 | 61 |
| Bird | 98 | 95 | 44 | 74 | 53 | 48 | 58 | 48 | 13 | 12 | 71 | 5 | 91 | 13 | 28 | 100 | 71 | 65 | 74 |
| Board | 95 | 88 | 3 | 47 | 73 | 82 | 76 | 71 | 81 | 16 | 38 | 5 | 23 | 63 | 84 | 61 | 97 | 93 | 90 |
| Bolt | 81 | 87 | 18 | 8 | 66 | 15 | 73 | 76 | 26 | 3 | 6 | 2 | 8 | 68 | 78 | 60 | 36 | 86 | 87 |
| Box | 86 | 89 | 4 | 8 | 33 | 42 | 18 | 90 | 34 | 60 | 8 | 9 | 37 | 25 | 92 | 60 | 46 | 92 | 91 |
| Car* | 92 | 65 | 43 | 73 | 43 | 40 | 38 | 59 | 44 | 58 | 10 | 11 | 43 | 49 | 97 | 54 | 38 | 93 | 82 |
| Coupon | 100 | 100 | 24 | 98 | 58 | 67 | 77 | 100 | 38 | 98 | 58 | 12 | 100 | 100 | 92 | 100 | 75 | 100 | 100 |
| Crowds* | 81 | 63 | 59 | 7 | 9 | 2 | 4 | 82 | 8 | 16 | 4 | 3 | 85 | 9 | 63 | 60 | 8 | 79 | 81 |
| David indoor* | 93 | 86 | 41 | 64 | 46 | 35 | 24 | 67 | 32 | 90 | 7 | 24 | 45 | 92 | 77 | 73 | 63 | 88 | 86 |
| Dragon baby | 67 | 83 | 16 | 28 | 30 | 35 | 38 | 43 | 33 | 15 | 28 | 4 | 23 | 24 | 58 | 43 | 49 | 63 | 59 |
| Man* | 100 | 96 | 98 | 41 | 60 | 21 | 21 | 100 | 31 | 98 | 18 | 8 | 22 | 100 | 91 | 80 | 45 | 100 | 98 |
| Motor rolling | 83 | 88 | 5 | 28 | 11 | 24 | 9 | 11 | 6 | 14 | 30 | 1 | 10 | 5 | 55 | 29 | 47 | 59 | 46 |
| Occluded face 2 | 100 | 100 | 60 | 22 | 100 | 80 | 94 | 79 | 77 | 76 | 43 | 8 | 49 | 82 | 96 | 78 | 88 | 100 | 97 |
| Shaking* | 65 | 62 | 13 | 3 | 72 | 7 | 37 | 7 | 82 | 1 | 2 | 18 | 93 | 3 | 58 | 39 | 5.1 | 55 | 37 |
| Surfer | 75 | 83 | 1 | 3 | 3 | 2 | 2 | 67 | 2 | 86 | 2 | 1 | 3 | 3 | 61 | 9 | 29 | 75 | 78 |
| Sylvester | 66 | 73 | 49 | 26 | 70 | 76 | 78 | 87 | 82 | 78 | 81 | 8 | 42 | 89 | 86 | 84 | 65 | 87 | 87 |
| Tiger2 | 66 | 85 | 10 | 3 | 65 | 5 | 77 | 65 | 17 | 26 | 5 | 2 | 21 | 27 | 69 | 50 | 12 | 74 | 73 |
| Trellis* | 91 | 81 | 67 | 72 | 35 | 18 | 34 | 70 | 54 | 31 | 18 | 2 | 45 | 34 | 84 | 58 | 39 | 91 | 88 |
| Woman | 83 | 85 | 8 | 80 | 6 | 71 | 6 | 87 | 5 | 30 | 34 | 4 | 80 | 8 | 69 | 65 | 83 | 77 | 83 |
| **Average** | 83.4 | 83.6 | 33.1 | 40.7 | 44.9 | 38.2 | 41.6 | 63.0 | 40.3 | 42.4 | 26.5 | 8.9 | 43.5 | 42.1 | 72.7 | 59.1 | 50.2 | 80.3 | 76.9 |
| **Average FPS** | 10.8 | 5.4 | 10.2 | 0.2 | 34.8 | 3.5 | 11.3 | 13.5 | 0.5 | 9.5 | 3.3 | 2.8 | 5.8 | 1.0 | 0.6 | 9.3 | 6.5 | 2.8 | 172 |

and histogram tracker (**BHT**) [24] and the local-global tracker (**LGT**) [32]. The other state-of-the-art algorithms include the real time L1 tracker (**L1T**) [4], the super-pixel tracker (**SPT**) [33], the real-time compressive tracker (**CT**) [38], the multiple instance learning tracker (**MIL**) [3], the structured output tracker (**Struck**) [12], the visual tracking decomposition method (**VTD**) [21], the **TLD** tracker [19], the distribution field tracker (**DFT**) [29], the multi-task sparse learning tracker (**MTT**) [40], the sparsity-based collaborative method (**SCM**) [42], the adaptive structural local sparse appearance method (**ASLA**) [18], the visual tracker sampler (**VTS**) [22], the transfer learning based gaussian processes regression method (**TGPR**) [10], and the kernelized correlation filters method (**KCF**) [14]. For the trackers that involve random feature selections, the average scores of five runs are used for evaluations. The source codes, tracking results, and data sets are available at the project page [23].

*1) Quantitative Evaluation:* Two evaluation criteria are used in the conducted experiments: the *center location error* and the *tracking success rate*, both computed against the manually labeled ground truth. The overlapping ratio is computed by $\frac{area(B_T \cap B_G)}{area(B_T \cup B_G)}$ where $B_T$ and $B_G$ are the bounding boxes of the tracker and of the ground-truth, respectively. When the overlapping ratio is larger than 0.5, the tracking result of the current frame is considered as a success. The success rate indicates the percentage of frames that are tracked with overlapping ratio larger than 0.5.

Figure 6 shows the tracking performance of our method with respect to different numbers of regions. The curves

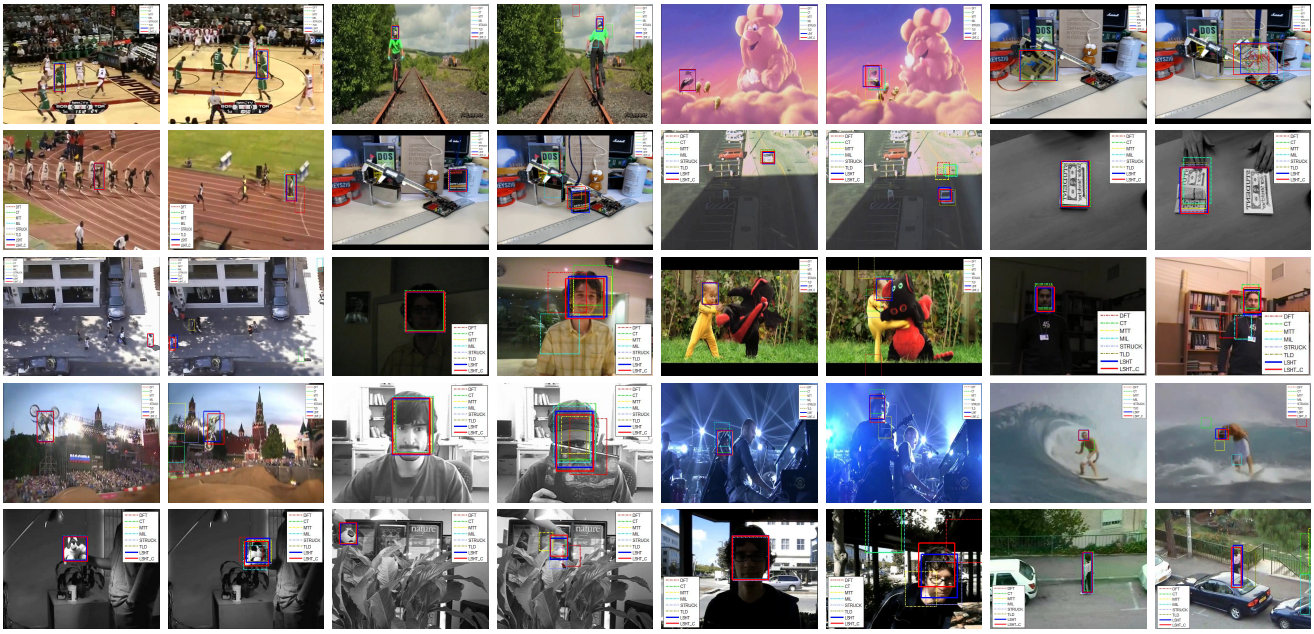


Fig. 7: Screenshots of the visual tracking results. The figures are placed in the same order as Table II. Blue and red solid rectangles correspond to the results of the proposed **LSHT** and **LSHT_C**. Dark red, green, yellow, azure, purple, and olive dashed rectangles correspond to the results from **DFT**, **CT**, **MTT**, **MIL**, **Struck**, and **TLD**.

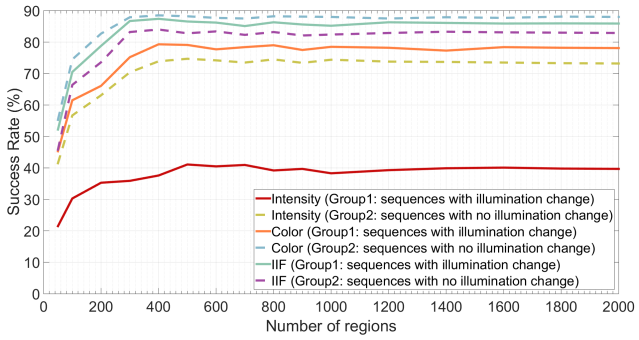



Fig. 6: Success rate (%) of the proposed tracker w.r.t. different numbers of regions.

show the average success rates of the 20 video sequences. We observe that the tracking performance of our method reaches its peak when the number of regions reaches 400, and thus we use 400 regions in the following experiments.

To demonstrate the effectiveness of the proposed illumination invariant features against intensity values, we divide the 20 sequences into two groups. The first group contains 6 sequences with large illumination changes (marked with a * in Tables I and II) and the second group includes the other sequences with different challenging factors. We then evaluate the proposed tracking algorithm on the two groups of image sequences using the proposed illumination invariant features (IIF), color and intensity. Figure 6 shows that the tracking method with the proposed IIF outperforms the one with intensity values not only on sequences with large illumination change, but also on those without. These results indicate that the proposed features are more effective than intensity values. In addition, the effectiveness of color information is also shown in Figure 6. On the sequences with illumination change, using color performs much better than intensity, as the target object can be better separated from the background with color information. Furthermore, color features outperform all the other features on the sequences without illumination change.

Tables I and II show the tracking performance and the speed (in frame rate) of our **LSHT** and **LSHT_C** tracker with the other 17 methods. We note that the **TLD** tracker does not report results (or bounding boxes) in some frames and the target objects are re-detected. Thus we only report the center location errors for the sequences that the **TLD** method does not lose track of target objects. The proposed tracker performs favorably against the state-of-the-art algorithms as it achieves the best or the second best performance in most sequences using both evaluation criteria. Furthermore, Tables I and II also show the importance of color information. Although **LSHT_C** performs not as good as **LSHT** in the sequences with illumination changes (e.g., *Car*, *Crowds*, *David Indoor* and *Trellis*), the overall performance is better than **LSHT**. This is because a single channel is difficult to distinguish the target object from background distraction. The target object can be better located with color information in some challenging scenarios, like *Dragon baby*, *Moter rolling*, *Tiger2* sequences. In addition, the proposed tracker is rather efficient with an average of 10.8 frames per second for **LSHT** and 5.4 for **LSHT_C**. Figures 7 shows the screenshots of the visual tracking results. For presentation clarity, we only show the results of **CT** [38], **MIL** [3], **TLD** [19], **Struck** [12], **DFT** [29] and **MTT** [40].

*2) Qualitative Evaluation:* We qualitatively evaluate 20 image sequences based on their main challenging factors

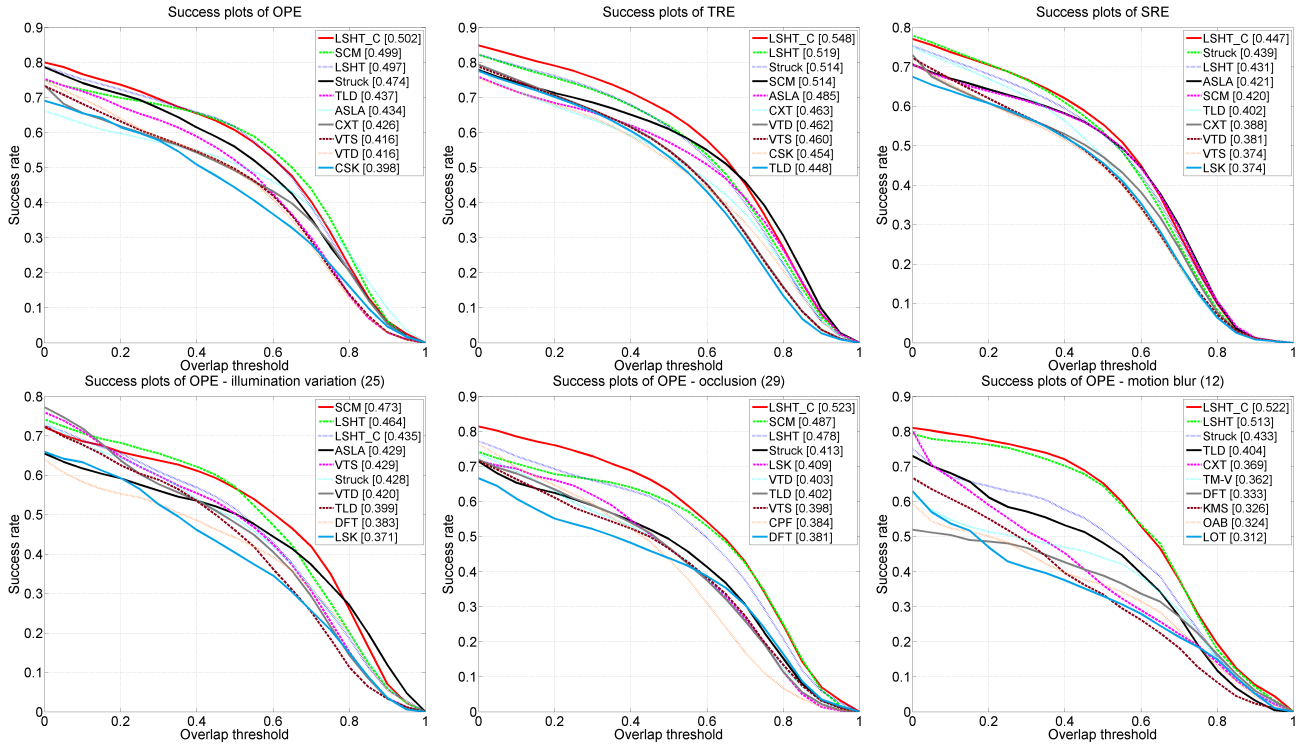



Fig. 8: The first row shows the success plots of **OPE**, **TRE**, and **SRE**. The second row shows part of the success plots of different challenging factors. The average AUC scores are shown in the legends.

as follows.

**Illumination, pose and scale variation.** In the *Man*, *Car* and *Crowds* sequences, the object appearances change drastically due to cast shadows and ambient lights. Only the **L1T**, **Struck** and the proposed **LSHT** are able to handle the illumination variation well. The proposed **LSHT_C** performs not well in *Car* and *Crowds* sequences, due to the sudden change of illumination. The *David Indoor* and *Trellis* sequences contain gradual illumination change and pose variation. We note that in most reported results using both sequences, only a subset of frames are used (i.e., not from the very beginning of the *David Indoor* sequence when the target object is in complete darkness, or until the very end of the *Trellis* sequence where the target object undergoes both sudden pose and lighting variations). In contrast, we use the full sequences for better assessment of all trackers. In both sequences, only the proposed **LSHT** is able to track the targets successfully in most frames (see the recorded videos at the project webpage [23]). This can be attributed to the use of the proposed features, which are insensitive to illumination variation. Note that **LSHT_C** performs not bad in these sequences. This is because the proposed online template update mechanism enables the tracker to handle gradual appearance change. Likewise, most of the other trackers do not perform well in the *Shaking* sequence since the object appearance changes drastically due to stage lights and sudden pose change. In addition, the proposed tracker performs well in the *Basketball* and *Bolt* sequences where the target objects undergo large pose variation.

**Occlusion and drift.** The target objects are partially occluded in the *Bird*, *Occluded face 2* and *Woman* sequences. For the *Woman* sequence, the target object encloses the whole body instead of just the upper body used in the fragment-based tracking method [1]. Most trackers do not perform well when the objects are heavily occluded. By exploiting a large number of regions, the relative spatial information among regions is maintained and thus the proposed trackers are able to handle occlusion well. Tracking drift usually occurs when a target object is heavily occluded. The *Box* sequence is challenging as the target object is heavily occluded in several frames. Only the proposed and the **Struck** trackers are able to relocate the target after heavy occlusion. As the proposed algorithm updates only some regions at any time instance, the tracking drift problem is alleviated where heavy occlusion occurs.

**Out of plane rotation and abrupt motion.** The target objects in the *Biker* and *Surfer* sequences undergo large out of plane rotation with abrupt movements. Since we do not consider large scale change in this work, only parts of these sequences are used for evaluations. We note that the scale approach in [1] can also be used in our method to deal with large scale variation. Most algorithms, except the proposed, **SCM**, and the **Struck** trackers, do not perform well in these sequences. The *Dragon baby* sequence is downloaded from Youtube where the baby moves abruptly in action scenes. The proposed algorithms track the baby well despite all the abrupt movements and out of plane rotation. The motorbiker performs acrobatic movements with 360 degree rotation in the *Motor rolling* sequence. While the proposed

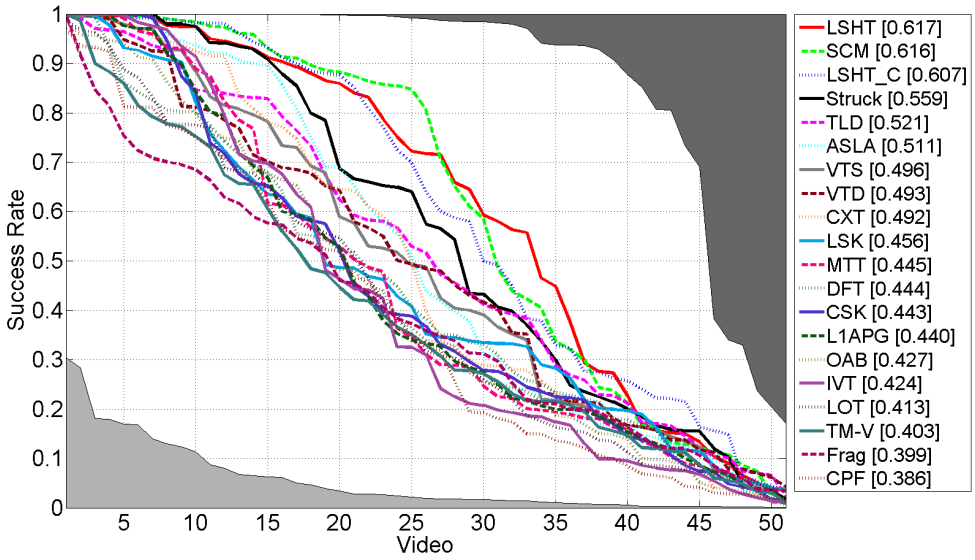



Fig. 9: Survival curves of the trackers w.r.t. success rates on the benchmark dataset. The average success rates are shown in the legend. The shaded areas represent the difficulty of this dataset. The upper right area represents the success rates that none of the 31 trackers was able to obtain. The bottom left area shows the opposite.

trackers are able to track the target object throughout the sequence, the other methods do not perform well.

**Background clutters.** In the *Tiger2* and *Board* sequences, the target objects undergo fast movements in cluttered backgrounds. The **MIL**, **SCM**, and the proposed trackers perform well, but the others fail to locate the target objects.

### *B. Evaluation 2: Benchmark Dataset*

We further evaluate the proposed tracker on a benchmark dataset [35] with several metrics. Two different features, color and illumination invariant features, are examined in this evaluation. This benchmark dataset includes 50 sequences, each of which contains different challenging factors (11 challenging factors in the dataset) and 29 tracking algorithms are included for comparison. The frame sizes vary, ranging from $300 \times 400$ to $800 \times 1000$.

*1) Evaluation Metrics:* We apply four evaluation metrics introduced by two recent survey papers [35], [31]. Three of the metrics [35] are used to evaluate the robustness to the spatial and temporal perturbations, and the last one [31] is to show the cumulative performance of the trackers on the entire dataset.

In the temporal robustness evaluation (**TRE**), each sequence is partitioned into 20 segments. Each tracker is initialized using one of these segments with ground-truth object location (i.e., bounding boxes), and then executed until the last frame of each sequence. The spatial robustness evaluation (**SRE**) metrics is to examine if the trackers are sensitive to different initial object positions. Each tracker is initialized at a position slightly away from the ground truth localization (including shifting and scaling errors) for each sequence. Together with the conventional one-pass evaluation (**OPE**) metric, these three metrics are used to evaluate the robustness of the trackers, and to compute the area under curve (**AUC**) scores of the success rates in order to summarize and rank the tracking algorithms. The **AUC** score measures the overall performance of each tracker, and it is more accurate than the score with a fixed threshold.

We further plot the survival curves [31] to evaluate the overall performance of the trackers. The survival curves were originally used to measure the effectiveness of a medical treatment based on how long the patients survived after the treatment. Here, we aim at exploring how many video sequences that the trackers can successfully track (i.e., overlapping ratio larger than 0.5). To plot the survival curves, we first compute the success rate of each tracker for every sequence. We then sort the videos according to the success rates. Note that the video orders are different for different trackers. These survival curves are able to avoid the influence of preference on particular videos, showing the overall performance of the trackers.

*2) The Performances of the Trackers:* Figure 8 shows some success plots from the benchmark evaluations where only the best ten trackers are presented for clarity. The first row of Figure 8 shows the overall performance. The evaluated trackers are effective in different aspects. **SCM** ranks second in OPE, but ranks fifth in SRE, which suggests that it is less robust to the initial position. **TLD** performs better in both OPE and SRE than TRE. This is because its re-detection mechanism works well in long sequences rather short video segments. Sparse representation-based methods (**SCM**, **ASLA**, **LSK**, **MTT**, and **L1APG**) perform well in SRE and TRE, which implies that sparse representations are able to capture appearance changes. On the contrary, the proposed tracker with color LSH (**LSHT_C**) performs well in all three criteria. In addition, the proposed tracker using only the illumination invariant features (**LSHT**) also ranks among top 3 in all criteria. We note that the proposed tracker has a higher success rate

when the overlap threshold is small, but has lower success rates than some trackers (e.g., **SCM**) when the threshold is large. This can be explained by the fact that the proposed tracker does not directly consider object scale change in the state parameters (which is also the case for several state-of-the-art methods such as **MIL**, **L1T** and **Struck**). The second row of Figure 8 shows some success plots on sequences with different challenging factors. The proposed **LSHT** using illumination invariant features perform well for sequences with large illumination variation. In addition, they also perform well for sequences with heavy occlusion and motion blur. The complete evaluation of 10 different challenging factors can be found in the supplementary material.

Figure 9 shows the survival curves of the best 20 trackers, for clarity. The proposed **LSHT** and **LSHT_C** achieve the best and the third average success rates among the 31 trackers tested. Together with **SCM**, these three methods significantly outperform all the other methods (at least by 8%). We can see that **LSHT** is robust to different types of sequences. There are 34 sequences with success rates over 50%. On the other hand, **LSHT_C** has nearly perfect performance for 12 sequences (with success rates over 98%). **SCM** performs very well in half of the 50 sequences, but it is not as robust as **LSHT** and its performance drops rapidly in the another half. The difficulty of this benchmark dataset is also indicated in the shaded areas of Figure 9. The upper right area shows the success rates that none of the 31 trackers were able to reach, which is about 12% of this benchmark dataset. The bottom left area shows the opposite that all the trackers were able to track objects, which is about 6% of this benchmark dataset. These numbers show that this benchmark dataset is challenging and none of these 31 trackers are able to handle perfectly. This figure can also be viewed together with the **OPE** plot of Figure 8, as the survival curves correspond to the performance at the threshold of 0.5. We can see that although **LSHT** performs the best at this threshold, **LSHT_C** is better at most of the other thresholds and thus achieves a better **AUC** score.

## VII. Conclusions

In this paper, we propose a novel locality sensitive histogram algorithm by taking spatial information into consideration. The proposed histogram leads to a simple yet effective tracking method. Experimental results show that the proposed multi-region tracker performs favorably against numerous state-of-the-art algorithms. As many vision problems can be modeled with histograms, our future work will focus on the extension of the proposed algorithms to other vision applications. As the proposed illumination invariant feature is a one dimensional feature, which is not as distinctive as color information in scenes with a cluttered background. Another future direction is to address both illumination and distinct color information simultaneously.

## Acknowledgments

We thank the anonymous reviewers for their insightful comments and constructive suggestions. The work described in this paper was partially supported by a GRF grant and an ECS grant from the RGC of Hong Kong (RGC Ref.: CityU 115112 and CityU 21201914). M.-H Yang is supported in part by the NSF CAREER Grant #1149783 and NSF IIS Grant #1152576.

## References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *Proc. IEEE CVPR*, pages 798 – 805, 2006.
[2] S. Avidan. Support vector tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(8):1064–1072, 2004.
[3] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(8):1619 –1632, aug. 2011.
[4] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *Proc. IEEE CVPR*, pages 1830 –1837, 2012.
[5] S. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In *Proc. IEEE CVPR*, pages 1158–1163, 2005.
[6] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int'l Journal of Computer Vision*, 26(1):63–84, 1998.
[7] H. Chen, P. Belhumeur, and D. Jacobs. In search of illumination invariants. In *Proc. IEEE CVPR*, pages 1254–1261, 2000.
[8] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):564 – 577, 2003.
[9] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. van de Weijer. Adaptive color attributes for real-time visual tracking. pages 1090 –1097, June 2014.
[10] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking with gaussian processes regression. In *Proc. ECCV*, volume 8691, pages 188–203. 2014.
[11] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *Proc. ECCV*, pages 234–247, 2008.
[12] S. Hare, A. Saffari, and P. Torr. Struck: Structured output tracking with kernels. In *Proc. ICCV*, pages 263 –270, 2011.
[13] S. He, Q. Yang, R. Lau, J. Wang, and M.-H. Yang. Visual tracking via locality sensitive histograms. In *Proc. IEEE CVPR*, 2013.
[14] J. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 37(3):583–596, March 2015.
[15] Z. Hong, X. Mei, D. Prokhorov, and D. Tao. Tracking via robust multi-task multi-view joint sparse representation. In *Proc. ICCV*, pages 649–656, Dec 2013.
[16] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *Int'l Journal of Computer Vision*, 29:5–28, 1998.
[17] D. Jacobs, P. Belhumeur, and R. Basri. Comparing images under variable illumination. In *Proc. IEEE CVPR*, pages 610–617, 1998.
[18] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *Proc. IEEE CVPR*, pages 1822–1829, June 2012.
[19] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34:1409–1422, 2012.
[20] J. Kwon and K. Lee. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In *Proc. IEEE CVPR*, pages 1208–1215, 2009.
[21] J. Kwon and K. Lee. Visual tracking decomposition. In *Proc. IEEE CVPR*, pages 1269–1276, 2010.
[22] J. Kwon and K. M. Lee. Tracking by sampling trackers. In *Proc. ICCV*, pages 1195–1202, Nov 2011.
[23] LSHT-Webpage. http://www.shengfenghe.com/visual-tracking-via-locality-sensitive-histograms.html, 2015.
[24] S. Nejhum, J. Ho, and M.-H. Yang. Visual tracking with histograms and articulating blocks. In *Proc. IEEE CVPR*, 2008.
[25] F. Porikli. Integral histogram: a fast way to extract histograms in cartesian spaces. In *Proc. IEEE CVPR*, pages 829–836, 2005.
[26] F. Porikli. Constant time O(1) bilateral filtering. In *Proc. IEEE CVPR*, 2008.
[27] D. Ross, J. Lim, R. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *Int'l Journal of Computer Vision*, 77(1-3):125–141, 2008.

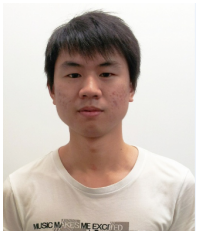
[28] Y. Rubner, C. Tomasi, and L. Guibas. The earth mover's distance as a metric for image retrieval. *Int'l Journal of Computer Vision*, 40(2):99–121, 2000.
[29] L. Sevilla-Lara and E. Learned-Miller. Distribution fields for tracking. In *Proc. IEEE CVPR*, pages 1910–1917, 2012.
[30] S. Shirdhonkar and D. Jacobs. Approximate earth mover's distance in linear time. In *Proc. IEEE CVPR*, 2008.
[31] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, July 2014.
[32] L. Čehovin, M. Kristan, and A. Leonardis. An adaptive coupled-layer visual model for robust visual tracking. In *Proc. ICCV*, 2011.
[33] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *Proc. ICCV*, pages 1323–1330, 2011.
[34] M. Werman, S. Peleg, and A. Rosenfeld. A distance metric for multidimensional histograms. *CVGIP*, 32(3):328 – 336, 1985.
[35] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proc. IEEE CVPR*, pages 2411–2418, 2013.
[36] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), Dec. 2006.
[37] Y. Zhai and M. Shah. Visual attention detection in video sequences using spatiotemporal cues. In *Proceedings of the 14th Annual ACM International Conference on Multimedia*, pages 815–824, 2006.
[38] K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In *Proc. ECCV*, 2012.
[39] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Low-rank sparse learning for robust visual tracking. In *Proc. ECCV*, volume 7577, pages 470–484. 2012.
[40] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via multi-task sparse learning. In *Proc. IEEE CVPR*, pages 2042 –2049, 2012.
[41] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via structured multi-task sparse learning. *Int'l Journal of Computer Vision*, 101(2):367–383, 2013.
[42] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *Proc. IEEE CVPR*, pages 1838–1845, June 2012.
[43] Z. Zivkovic and B. Krose. An em-like algorithm for color-histogram-based object tracking. In *Proc. IEEE CVPR*, volume 1, pages I–798– I–803 Vol.1, June 2004.

**Shengfeng He** is a Research Fellow at City University of Hong Kong. He obtained his B.Sc. degree and M.Sc. degree from Macau University of Science and Technology, and the Ph.D degree from City University of Hong Kong. His research interests include computer vision, image processing, computer graphics, and deep learning.

**Rynson W.H. Lau** received his PhD degree from University of Cambridge. He was on the faculty of Durham University and The Hong Kong Polytechnic University. He is now with City University of Hong Kong.

Rynson serves on the Editorial Board of Computer Animation and Virtual Worlds and IEEE Trans. on Learning Technologies. He has served as the Guest Editor of a number of journal special issues, including *ACM Trans. on Internet Technology*, *IEEE Trans. on Multimedia*, *IEEE Trans. on Visualization and Computer Graphics*, and *IEEE Computer Graphics & Applications*. In addition, he has also served in the committee of a number of conferences, including Program Co-chair of *ACM VRST 2004*, *ACM MTDL 2009*, *IEEE U-Media 2010*, and Conference Co-chair of *CASA 2005*, *ACM VRST 2005*, *ACM MDI 2009*, *ACM VRST 2014*.

**Qingxiong Yang** is an Assistant Professor in the Department of Computer Science at City University of Hong Kong. He obtained his BEng degree in Electronic Engineering & Information Science from University of Science & Technology of China (USTC) in 2004 and PhD degree in Electrical & Computer Engineering from University of Illinois at Urbana-Champaign in 2010. His research interests reside in Computer Vision and Computer Graphics. He won the best student paper award at MMSP 2010 and best demo at CVPR 2007.



**Jiang Wang** was a research assistant at City University of Hong Kong. He obtained his B.Sc. degree from Dalian Nationalities University. His research interests include computer graphics and computer vision.

**Ming-Hsuan Yang** is an associate professor in Electrical Engineering and Computer Science at University of California, Merced. He received the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2000. Prior to joining UC Merced in 2008, he was a senior research scientist at the Honda Research Institute working on vision problems related to humanoid robots. He coauthored the book Face Detection and Gesture Recognition for HumanComputer Interaction (Kluwer Academic 2001) and edited special issue on face recognition for Computer Vision and Image Understanding in 2003, and a special issue on real world face recognition for IEEE Transactions on Pattern Analysis and Machine Intelligence. Yang served as an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence from 2007 to 2011, and is an associate editor of the Image and Vision Computing. He received the NSF CAREER award in 2012, the Senate Award for Distinguished Early Career Research at UC Merced in 2011, and the Google Faculty Award in 2009. He is a senior member of the IEEE and the ACM.