Visual Tracking via Locality Sensitive Histograms

Yufeng Jiang

Abstract

This paper presents a novel locality sensitive histogram algorithm for visual tracking. Unlike the conventional image histogram that counts the frequency of occurrences of each intensity value by adding ones to the correspongding 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 declines exponentially with respect to the distance to the pixel location where the histogram is computed; thus every pixel is considered but those that are far away can be neglected due to the very small weights assigned.

1. Introduction

Visual tracking is one of the most active research areas in computer vision, with numerous applications including augmented reality, surveilance, and object identification. The chief issue for robust visual tracking is to handle the appearance change of the target object. Based on the appearance models used, tracking algorithms can be divided into generative tracking [4, 10, 11, 3] and discriminative tracking [5, 2, 8, 6].

Generative tracking represents the target object in a particular feature space, and then searches for the best matching score within the image region. In general, it does not requires a large dataset for training. Discriminative tracking treats visual tracking as a binary classification problem to define the boundary between a target image patch and the background. It generally requires a large dataset in order to achieve good performances. While numerous algorithms of two categories have been proposed with success, it remains a challenging task to develop a tracking algorithm that is both accurate and efficient. In order to address challenging factors in visual tracking, numerous features and models have been used to represent target objects.

2. Locality Sensitive Histograms

The conventional image histogram is a 1D array. Each of its values is usually an integer indicating the frequency of occurrence of a particular intensity value. Let $\mathbf{I}$ denote an image. The corresponding image histogram $\mathbf{H}$ is a B-dimensional vector defined as:

$$\mathbf{H}(b) = \sum_{q=1}^{W} Q(\mathbf{I}_q, b), \quad b = 1, \ldots, B, \tag{1}$$

Where W is the number of pixels, B is the total number of bins, and $Q(\mathbf{I}_q, b)$ is zero except when intensity value $\mathbf{I}_q$ (at pixel location q) belongs to bin b. If computed using Eq. 1 shown at [7], the computational complexity of the histogram is linear in the number of bins at each pixel location: $O(B)$. However, the complexity can be reduced to $O(1)$ in practive because the addition operation in Eq. 1 can be ignored when $Q(\mathbf{I}_q, b) = 0$.

Local histograms, on the other hand, record statistics within a region of a pixel in an image. They are computed at each pixel location, and have been proved to be very useful for many tasks like the bilateral filtering [13], median filtering [13, 9] and tracking [12, 1]. The computational complexity of the brute-force impementation of the local histograms is linear in the neighborhood size. Nevertheless, this dependency can be removed using integral histogram [12], which reduces the computational complexity to $O(B)$ at each pixel location. Let $\mathbf{H}_p^{\mathbf{I}}$ denote the integral histogram computed at pixel p, just as illustrated at Eq. 2 which cited from [7] It can be computed based on the previous integral histogram computed at pixel $p-1$ in a way similar to the integral image:

$$\mathbf{H}_p^{\mathbf{I}}(b) = Q(\mathbf{I}_p, b) + \mathbf{H}_{p-1}^{\mathbf{I}}(b), \quad b = 1, \ldots, B, \tag{2}$$

For simplicity, let $\mathbf{I}$ denote a 1D image. $\mathbf{H}_p^{\mathbf{I}}$ contains the cintributions of all the pixels to the left of pixel p, and the local histogram between pixel p and another pixel q on the left of p is computed as $\mathbf{H}_p^{\mathbf{I}} - \mathbf{H}_q^{\mathbf{I}}(b)$ for $b = 1, \ldots, B$.

For the local histogram, pixels inside a local neighborhood has equal contribution. However, for applications such as object tracking, pixels furtuer away from the target center should be weighted less as they more likely contain background information or occluding objects. Hence, their contribution to the histogram should be diminished. They propose a novel locality sensitive histogram algorithm to address this problem. Let $\mathbf{H}_p^E$ denote the locality sensitive

Sequence	Ours	LIT	SPT	CT	Frag	MIL	Struck	VTD	TLD	BHT	LGT	DFT	MTT
Basketball	**9.3**	14.9	20.0	62.9	25.2	102	165	**8.6**	–	153	15.2	239	287
Biker	**8.9**	40.4	52.9	26.0	79.8	31.0	**12.7**	69.4	–	30.1	85.5	46	76.1
Bird	**8.3**	57.8	14.1	17.1	26.6	17.2	23.9	61.9	–	14.7	15.6	**10.9**	73.2
Board	**10.9**	174	52.6	34.7	25.2	35.0	34.6	**17.8**	–	42.4	36.6	145	50.3
Bolt	**6.4**	186	72.0	9.0	73.2	**8.1**	8.4	17.7	–	149	48.6	50.9	9.0
Box	**13.1**	77.5	169	107	65.8	109	**10.6**	114.1	–	111	68.9	120	54.8
Car	**3.9**	36.8	**4.5**	38.3	40.2	37.9	6.4	35.3	9.4	156	60.9	39.0	34.1
Coupon	**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**
Crowds	5.9	15.3	433	351	435	465	**5.0**	380	–	344	232	**4.5**	226
David indoor	**10.6**	28.5	29.8	25.8	103	44.1	30.5	64.6	**10.4**	122	13.4	28.6	10.7
Dragon baby	**20.0**	72.8	49.7	**30.8**	51.1	48.6	59.1	42	–	83.1	87	148	78.9
Man	**2.1**	2.8	23.1	9.4	48.6	37.7	**2.4**	20.7	4.4	73.0	24.1	40.0	2.6
Motor rolling	**13.1**	187	**76.2**	198	110	178	84	148	–	137	178	170	180
Occluded face 2	**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
Shaking	13.8	42.0	144	33.4	195	**12.9**	43.2	24.0	–	237.9	35.7	**8.7**	52.2
Surfer	**7.4**	122	78.2	78.6	150	128	14.6	93.3	**5.4**	93.3	188.1	143	93.1
Sylvester	**6.1**	23.4	47.8	9.1	14.6	12.6	6.7	9.7	13.7	11.1	23.9	39	**5.1**
Tiger2	**8.5**	46.3	85.7	11.9	45.7	**7.7**	9.8	32.9	–	66.8	27.9	33.6	25.5
Trellis	**8.3**	**15.5**	18.8	67.3	100	65.8	22.9	32.4	–	75.4	16.1	51.3	56.9
Woman	**6.4**	136	8.9	130	7.3	144	**5.5**	136	–	71.9	19.3	8.6	154
Average	**8.6**	68.4	71.2	62.5	82.3	76.5	**28.4**	69.5	–	103	61.5	67.8	74.5

Table 1. The average center location errors (in pixels) of the 20 sequences. The best and the the second best performing methods are shown in red color and blue color, respectively. The total number of frames is 10,918. The entry '–' for **TLD** indicates that the value is not available as the algorithm loses track of the target object.

histogram omputed at pixel p. It can be written as:

$$\mathbf{H}_p^E(b) = \sum_{q=1}^{W} \alpha^{|p-q|} \cdot Q(\mathbf{I}_q, b), \quad b = 1, \ldots, B, \qquad (3)$$

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 ocmplexity of its brute-force implementation presented in Eq. 3 shown at [7] is $O(WB)$ per pixel. However, similar to the integral histogram, the proposed locality sensitive histogram also has an efficient algorithm when the input image is 1D. The Eq. 4 explains the algorithm cited from [7].

$$\mathbf{H}_p^E(b) = \mathbf{H}_p^{E,left}(b) + \mathbf{H}_p^{E,right}(b) - Q(\mathbf{I}_p, b) \qquad (4)$$

where

$$\mathbf{H}_p^{E,left}(b) = Q(\mathbf{I} - p, b) + \alpha \cdot \mathbf{H}_{p-1}^{E,left}(b), \qquad (5)$$

$$\mathbf{H}_p^{E,right}(b) = Q(\mathbf{I}_p, b) + \alpha \cdot \mathbf{H}_{p+1}^{E,right}(b). \qquad (6)$$

Based on Eqs. 5 and 6 shown at [7], pixels on thr right of pixel p do not contribute to $\mathbf{H}_P^{E,right}$ while pixels on the left of pixel p do not contribute to $\mathbf{H}_p^{E,right}$.

Tab. 1 cited from [7] shows the tracking performance and the speed of their method with the 12 other methods. They note that the **TLD** tracker does not report tracking result when the drift problem occurs and the target object is redetected.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, 2006. 1

[2] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE TPAMI*, 33(8):1619–1632, 2011. 1

[3] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust L1 tracker using accelerated proximal gradient approach. In *CVPR*, 2012. 1

[4] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *IJCV*, 26(1):63–84, 1998. 1

[5] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008. 1

[6] S. Hare, A. Saffari, and P. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011. 1

[7] S. He, Q. Yang, R. W. H. Lau, and J. Wang. Visual tracking via locality sensitive histograms. In *CVPR*, 2013. 1, 2

[8] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE TPAMI*, 34(7):1409–1422, 2012. 1

[9] M. Kass and J. Solomon. Smoothed local histogram filters. *ACM TOG*, 29(4):100:1–100:10, 2010. 1

[10] J. Kwon and K. M. Lee. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive Basin Hopping Monte Carlo sampling. In *CVPR*, 2009. 1

[11] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, 2010. 1

[12] F. Porikli. Integral histogram: A fast way to extract histograms in Cartesian spaces. In *CVPR*, 2005. 1

[13] P. Viola and M. Jones. Robust real-time object detection. *IJCV*, 57(2):137–154, 2004. 1