2015 International Conference on Computational Science and Computational Intelligence

Real-time tracking based on rotation-invariant descriptors

Daniel Miramontes-Jaramillo*
* *Department of Computer Science, CICESE, Ensenada, B.C. 22860, Mexico*

Vitaly Kober*†
† *Department of Mathematics, Chelyabinsk State University, Russian Federation*

Email: dmiramon@cicese.edu.mx, vkober@cicese.mx

***Abstract*—Common tracking algorithms based on descriptors usually use a bounding box containing a target for extracting of its features. Disjoint background noise inside of the box strongly affects target descriptors. We propose to compute the histograms of oriented gradients in several circular windows within the actual region of support of a target. Such descriptors are background noise-free and rotation-invariant. The suggested tracking algorithm additionally utilizes depth information from a Kinect camera for better tracking when partial occlusions of the target are faced. The performance of the proposed algorithm is tested in terms of recognition rate using the Princeton Tracking Benchmark scenarios and compared with that of the state-of-the-art tracking algorithms. Finally, in order to achieve high rate of processing, the algorithm was implemented with GPU parallel processing technologies.**

***Keywords*-tracking; oriented gradient histograms; GPU implementation;**

I. Introduction

Tracking algorithms have gained increasing popularity over the last decade. Nowadays, with development of depth cameras, numerous innovative tracking algorithms robust to environmental and technical interference were proposed. One of the most popular depth sensors is the Microsoft Kinect camera.

The state-of-the-art algorithms based on descriptors are as follows: utilizing depth information such as RGBD Occlusion+Optical Flow (RGBDOcc+OF) [1] and Occlusion Aware Particle Filter (OAPF) [2]; without using depth information such as RGBD + Optical Flow (RGBD+OF) [1], Tracking-Learning-Detection (TLD) [3] and Multiple Instance Learning (MIL) [4]. Classical tracking systems can be classified as follows [5]: template trackers use histograms and other data structures to describe objects; silhouette trackers use shapes and edges of objects; feature trackers extract interest points of targets.

In this paper we propose a tracking algorithm that takes advantage of depth information for illumination invariance, segmentation and occlusion handling. First, we extract a data structure containing circular windows within the actual region of support of a target. Then, we carry out in each frame the following steps: preprocessing to remove additive electronic noise and enhancing the contrast; localizing the object area using prediction model and depth information and reducing the search area to a fragment; extracting iteratively the histograms of oriented gradients with one pixel distance in the frame fragment; finally, matching the histograms with those in the frame fragment to locate the target in each frame of video sequence.

The paper is organized as follows. In section II, preprocessing steps are described. In section III, important components of the proposed tracking algorithm are discussed. In section IV, we illustrate the performance of the tracking with the help of the proposed and state-of-the-art algorithms. Section V summarizes our conclusions.

II. Noise Removal and Illumination Correction

Captured video frames always contain additive sensors noise and nonuniform illumination of the scene across the image.

The first stage of image processing algorithms is to remove the noise that can be caused by a sensor or environment conditions. Additive wide-band noise can be removed by a Gaussian filter. However, it is necessary to estimate the noise standard deviation to process the image correctly. The autocorrelation function of white noise is the Kronecker delta function. The variance of the noise can be calculated by linearly extrapolating the values in the vicinity of the origin of the autocorrelation in the noisy image to estimate the sample variance of the ideal image, and the rest is considered as the variance of white noise.

The estimated noise standard deviation (σ_n) is used to quantize the histogram of oriented gradients (HOG) in order to compensate errors introduced in the computed angles by white noise. The number of quantized directions Q for the histogram can be calculated as follows:

$$Q = \lceil \frac{360}{\sigma_n} \rceil . \quad (1)$$

In addition, illumination changes across the image affect the local contrast of details and fine structures of the image. Therefore, without contrast correction of the image, descriptors of the same object will be different across the image depending on the target position in the scene. In

978-1-4673-9795-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCI.2015.21

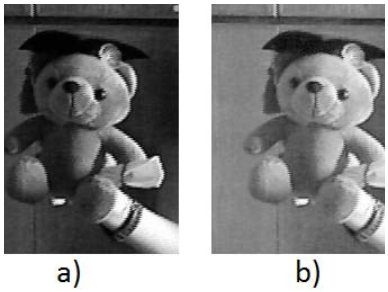

Figure 1. a) Original image, b) gamma corrected image.

order to reduce the influence of illumination we use two procedures. First, the gamma correction [6] is defined as

$$V_O = V_I^{\gamma}\,, \tag{2}$$

where V_O and V_I are the output and input images, respectively, and the gamma factor ranges between $[1/2.2, 1/2.6]$ [7]. This method reduces the illumination variation and local shadowing effects (see example in Fig. 1). Second, supposing that illumination is approximately uniform in small areas, we perform the tracking in a frame fragment instead of the entire frame. Therefore, searching small areas instead of entire image is beneficial for illumination compensation.

III. Tracking Algorithm

In this section we describe important components of the proposed tracking algorithm.

A. Geometric structure

Let us define elements needed for object descriptors computation; that is, a geometric structure of disks moving across the image and the Histograms of Oriented Gradients (HOG) [6].

Let W_i be a set of closed M disks, with distances between disks D_{ij} and angles between every three adjacent centers of the closed disks θ_i [8] (see Fig. 2). The histograms of oriented gradients are calculated in circular areas and further used for matching. It is interesting to note that at any position of the structure each disk contains image area that is unchangeable during rotation; therefore, the histogram of oriented gradients computed in a circular window is also invariant to rotation.

B. Histogram of oriented gradients

The proposed descriptor is based on histograms of oriented gradients computed within closed disks. First, we compute the gradients at each pixel with simple operators of the form:

$$gx = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}\,,\quad gy = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}\,. \tag{3}$$

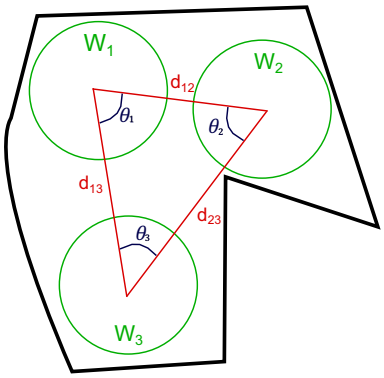

Figure 2. Circular disk structure defined inside the region of support of the target.

The magnitude and orientation at each pixel can be computed as

$$mag\,(x,y) = \sqrt{gx^2 + gy^2}\,, \tag{4}$$

$$ori\,(x,y) = \arctan\,(gy/gx)\,. \tag{5}$$

The orientation is in the range of $[0°, 360°]$. We select a number of bins Q according to the noise standard deviation σ_n. The orientation at each pixel is quantized as [9]

$$\varphi\,(x,y) = \left\lfloor \frac{Q}{360} ori\,(x,y) + \frac{1}{2} \right\rfloor\,, \tag{6}$$

where the factor $\frac{1}{2}$ rotates the origin of the histogram in counterclockwise, so the values at the beginning and end of the histogram fit into the first bin. The orientation cyclic condition is also fulfilled; that is, the gradient orientation error can set the values closer to $0°$ near to $360°$ and vice versa.

The HOG is computed by magnitude voting. Each magnitude is divided between the two closest bins of the histogram proportionally to the corresponding orientation distance for each bin.

Finally, we compute a centered and normalized histogram, which possesses rotation invariance,

$$\overline{HOG}\,(\varphi) = \frac{HOG\,(\varphi) - Mean}{\sqrt{Var}}\,, \tag{7}$$

where $Mean$ and Var are the sample mean and variance of the histogram, respectively.

The histogram of oriented gradients is computed in the defined circular window W_i of the target running across the frame fragment.

The first histogram within the frame fragment is calculated from a closed disk with the same radius as the disks in the object structure; once this histogram is computed, the closed disk advances trough the fragment pixel by pixel updating the information within the histogram in a vertical or horizontal direction as shown in Fig. 3. The iterative

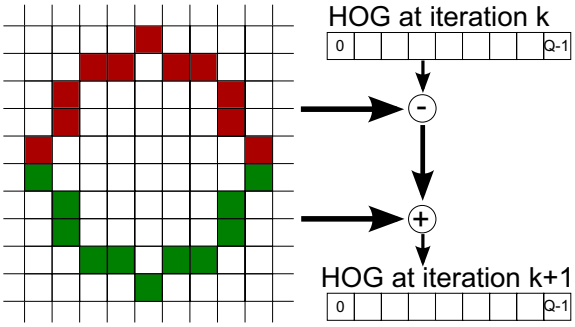

Figure 3. Recursive update of the histogram along columns.

computation of histograms allows fast processing in the frame fragment because only $2\pi r$ pixels in the circular window are processed instead of πr^2 (total window pixels) at each iteration.

C. Matching of descriptors

At any step k of the iterative process we compute the scene histogram. The matching can be performed by correlating the histograms of the ith circular window at position k and the scene histogram using the Fourier transform [10],

$$C_i^k(\alpha) = IFT\left[HS_{ik}(\omega)\, HO_i^*(\omega)\right], \tag{8}$$

where $HS_{ik}(\omega)$ is the centered and normalized Fourier Transforms of the histogram of oriented gradients inside of the kth circular window over the frame fragment correlated to $HO_i(\omega)$, that is the Fourier Transform of the HOG in the ith circular window in the target object; the $(*)$ denotes complex conjugate. The correlation peak is a measure of similarity of two histograms computed as follows:

$$P_i^k = \max_{\alpha}\left\{C_i^k(\alpha)\right\}. \tag{9}$$

The correlation peaks are in the range of $[-1, 1]$. We suggest a M-pass procedure. First, to perform the matching of the first circular window in the structure with the objective to reject as much as possible points in the frame fragment by applying a threshold Th to the correlation peaks to conserve the higher valued points and keep a low probability of miss errors. Second, only accepted points are considered to carry out the matching with the second circular window of the structure, taking into account the threshold value and the center to center distance D_{ij} to the first window. By rejecting another set of points, at the third pass, it is possible to use the angles between each three adjacent centers θ_i to quickly locate the position of the next window; and so on, evaluating the M windows in the structure. The final decision about the presence of the target object is taken considering the joint distribution of the correlation peaks for all windows. In this way, a trade-off between the probabilities of miss and false alarm errors is achieved.

IV. Prediction of target location

In order to improve the processing rate, we crop a small fragment containing a target from the entire frame. For the first frame, if the starting position of the target is unknown then we detect the object across the entire frame. The geometric structure is always defined within the actual region of support of a target. The size of frame fragments is chosen larger 1.5 times than the size of the bounding target box to provide the invariance to slight changes of the distances, the angles, and target scaling. The prediction of the target location for other frames is based on time series by fitting the target movement in x and y directions to a polynomial curve.

If occlusion occurs, the obstruct depth information will appear first in the histogram, because the obstruct object will be closer to the sensor. If the frame fragment obtained in the prediction stage has no depth histogram of the target, we correct the frame position to better locate the target in subsequent frames. To maintain reliability, the depth histogram is updated frame by frame depending on the position of the target with respect to sensor. If the object exists for while from the scene and enters, the search is carried out in entire frames until the target is detected.

V. Experimental Results

In this section we present and discuss the obtained experimental results. The experiments are carried out using validation video sequences from the Princeton Tracking Benchmark [1], which are composed of five video sequences taken with the Microsoft Kinect with the number of frames varied from 51 to 370. Each sequence contains RGBD images of the size of 640×480 pixels and depth images as well as ground truth information for validation. The benchmark also provides comparative results of popular tracking algorithms. We choose a subset of the algorithms with the best results and consider them as the state-of-the-art.

The parameters of the proposed algorithm are as follows: $M = 2$, $Q = \begin{cases} \lceil 360/\sigma_n \rceil & , 1.5 < \sigma_n < 40 \\ 64 & , otherwise \end{cases}$, $Th = 0.8$ and $r = 32$. The algorithm was implemented in a standard PC with Intel Core i7 processor with 3.2 GHz and 8 GB of RAM, ATI RADEON HD 6450 using OpenCV to read the image and compute other basic operations, and OpenCL for parallelization. The implemented algorithm achieves real-time processing with the rate of about 30 FPS.

First, the algorithms in terms of the success rate against the overlap area threshold are tested. We measure the overlap between the bounding box obtained with the algorithm and the ground truth bounding box. The success rate measures how many target bounding boxes overlap at any given rate in the sequence, and the performance of the algorithm is given by the area below the curve from a given threshold,

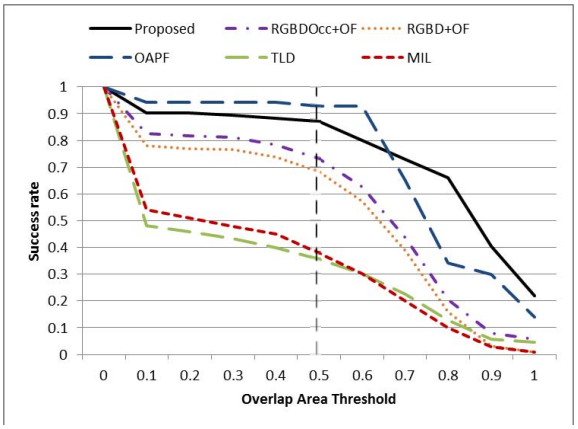

Figure 4. Average success rate versus overlap area.

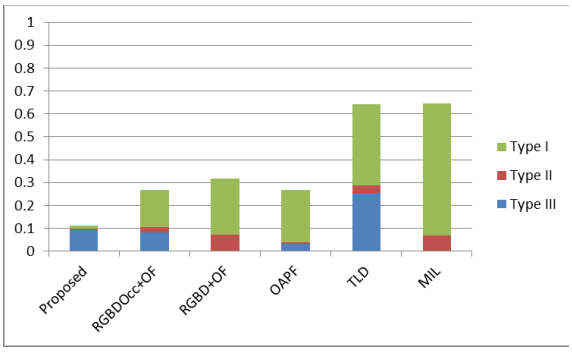

Figure 5. Performance in terms of errors of the target presence and overlap of bounding boxes.

in this case the threshold is 50% of overlap. Fig. 4 shows the success rate against the overlap area. The area under the curve shows how well the algorithm performs. One can be seen that the algorithms exploiting depth information perform better than those that do not use depth information. The proposed algorithm competes with the OAFP [2], surpassing it from 80% to 100% of overlapping and having similar values below this threshold.

The second test is performed in terms of errors defined [1]: as follows: Type I, when the target is visible but the result bounding box does not overlap with the ground truth bounding box; Type II, when the target is completely occluded but the algorithm outputs the bounding box; Type III, when the target is visible but the algorithm does not output the bounding box. Fig. 5 shows that the performance of the algorithms utilizing depth information is much better than those that do not use depth information. The proposed algorithm has the best performance in terms of the errors even when the target is partially occluded.

VI. Conclusion

In this paper we presented a real-time rotation-invariant tracking algorithm based on HOGs descriptor and depth information. The proposed algorithm is robust to noise and illumination variations, target occlusion as well as to slight scale and camera point of view changes. In addition to intensity data we use depth information to segment and track a target with the help of the position prediction model. According to our computer simulations with the Princeton Tracking Benchmark, the proposed algorithm is competitive with the state-of-the-art tracking algorithms. In future we plan to further improve the performance of the tracking algorithm with long-term fragments of video sequences containing occluded objects.

Acknowledgment

This work was supported by the Russian Science Foundation, grant no. 15-19-10010.

References

[1] S. Song and J. Xiao, "Tracking revisited using rgbd camera: Unified benchmark and baselines," in *Proc. Of 14th IEEE Int. Conf. on Comp. Vis.*, 2013, pp. 233–240. [Online]. Available: http://vision.princeton.edu/projects/2013/tracking/index.html

[2] K. Meshgi, S. Maeda, S. Oba, and S. Ishii, "Fusion of multiple cues from color and depth domains using occlusion aware bayesian tracker," in *IEICE Tech. Rep. Neurocomp.*, vol. 113, no. 500, 2014, pp. 127–132.

[3] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, 2012, pp. 1409–1422.

[4] B. Babenko, Y. Ming-Hsuan, and S. Belongie, "Visual tracking with online multiple instance learning," in *IEEE Conf. on Comp. Vis. and Patt. Rec.*, 2009, pp. 983–990.

[5] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," in *ACM Computer Surveys*, vol. 38, no. 4, 2006, p. 45.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Comp. Vis. and Patt. Rec.*, vol. 1, pp. 886–893, 2005.

[7] L. Po-Ming and C. Hung-Yi, "Adjustable gamma correction circuit for tft lcd," in *IEEE Symp. On Circ. and Syst.*, 2005, pp. 780–783.

[8] D. Miramontes-Jaramillo, V. Kober, and V. Daz-Ramrez, "Cwma: Circular window matching algorithm," in *Proc. 18th Iberoam. Cong. in Patt. Rec.*, vol. LNCS 8258, 2013, pp. 439–446.

[9] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, "Fast computation of rotation-invariant image features by approximate radial gradient transform," in *IEEE Trans. Imag. Proc.*, vol. 22, no. 8, 2013, pp. 2970–2982.

[10] W. K. Pratt, *Digital Image Processing*. John Wiley & Sons, 2007.