International Journal of Computer Applications (0975 - 8887)
Volume 153 - No.6, November 2016

Visual Tracking using Corner based Centrist Descriptor with a Robust Localization Algorithm

Mahdi Tanbakuchi
Department of Electrical Engineering
South Tehran Branch
Islamic Azad University
Tehran, Iran

Mojtaba Lotfizad
Department of Electrical and Computer Engineering
Tarbiat Modares University
Tehran, Iran

ABSTRACT

In this paper an algorithm for object tracking in the visual domain based on a novel localization method is proposed. First a part of the search area, preferably the interest points is chosen. The proposed approach drastically speeds up the process of tracking, meanwhile the intensity histogram and Centrist descriptor which is known for good coding capability of small patches of an image will be used for target's description. In order to increase the accuracy of the descriptor, this descriptor is applied to small blocks of image to encode most of the image around the target's interest points. By providing the description of object's interest points, a 1-NN classifier is used to distinguish the corresponding target's interest points in each frame. Given the matched corresponding interest points, a convolution problem is formulated to detect the center of the target. Experiments on a challenging dataset against several state-of-the-art methods demonstrate the efficiency of the proposed algorithm.

General Terms

Computer Vision; Image Processing

Keywords

Feature Extraction; Target Description; Visual Tracking; 1-NN Classification, Localization

1. INTRODUCTION

Visual target tracking is the process of finding the location of the object of interest over time in a video sequence. As this task is crucial to many applications (e.g., video indexing, automatic surveillance, etc.), it has been a hot research area and the subject of interest of many researches in the recent years. A tracking algorithm faces many difficulties to overcome: e.g., clutter, occlusion, illumination changes, pose deformation, etc. While trying to overcome such difficulties, these algorithms must run in real time and this implies a trade-off between accuracy and speed. Many researches have focused on improving the accuracy of tracking algorithms whether with direct or indirect approaches. For instance, a direct solution is given in [1] where an adaptive distance metric is introduced for robust visual tracking. On the other hand, there are indirect solutions which contribute to improvement of building blocks of visual tracking algorithms, namely feature extraction, object representation, and object localization [2]. In these solutions, most researches aimed at designing a robust appearance model of the target which can be either holistic or local based [3]. Holistic models treat the target as a whole and use global features such as intensity, color, texture, and shape features in order to model the target. Since these models do not consider spatial relation of target's components, they cannot model the geometrical structure of the target properly[3].On the other hand, local based methods use key point detector to find structures like corners or blobs in the target region and then model the target's appearance based on patches around the key points [4].

Based on the acquired target appearance model, tracking can be done with either a generative or a discriminative approach [5]. Generative tracking algorithms exploit feature extraction to model appearance of the target. Hence, quality of feature extraction algorithms plays an important role in constructing good appearance models. Mixture models, kernel based methods, subspace learning and linear representation have been proposed for object appearance modeling [6]. On the contrary, discriminative approaches form a classification problem and train a classifier to distinguish the target from its surrounding background. Training the classifier can be done either online or offline It is demonstrated that online methods are more robust against appearance variations [7]. There have been many visual trackers proposed in the literature for the mentioned categories given above. In [8] an extension to holistic target modeling (subspace learning) [9] is used for object representation in a generative tracking framework and a novel subspace learning has been proposed based on the feature observations of temporally obtained targets. Another generative tracker which uses sparse representation is proposed in [10] where only one L1 optimization is adopted for computing the weights for all particles. However, this formulation cannot handle occlusion and illumination changes effectively while using the Euclidean distance metric. Thus, a locally weighted distance metric is proposed. Yet another generative tracker which maps target's templates into binary space is proposed in [11], where inter class and intra class information are integrated to train multiple hash functions with higher discriminability which are used to project target candidates into the hamming space, making the distance calculation efficient. After projecting into the hamming space, tracking is treated as an approximate nearest neighbor searching process in a binary space.

There are many proposed discriminative tracking algorithms as well. In [12] a novel method called max-confidence boosting is proposed which explores a new way of updating ambiguous visual phenomenon. In this algorithm, uncertainty in prior knowledge is modeled using in-deterministic labels which in turn are used to update models in the previous frame and the current frame. In [13] the performance of the binary classifier is improved based on the processing of the structured unlabeled data. For training a binary classifier from labeled and unlabeled examples, a new approach called P-N learning is proposed.

Most of the generative tracking algorithms use holistic based target representation; nevertheless, these methods cannot perform well in handling partial occlusion and spatial distracters. On the contrary, their counterpart, i.e. part-based appearance modeling can handle such situations much better [3].

In this paper a part based appearance model for target representation based on corner key points and the Centrist descriptor is proposed. A new method for target localization inspired by the work of Zhang et al. [5] is also presented. The main contributions of this paper are:

(1) A new target appearance model based on the Centrist descriptor around the target's detected key points,

(2) Using a novel approach for target localization based on its detected key points.

The remainder of this paper is organized as follows: Section 2 will briefly discuss the related work on key point detection and description as well as the tracking methods. In section 3 the proposed tracker termed CCT (Corner Centrist Tracker) is presented by introducing the visual descriptor in the first step followed by the proposed approach to model the target's appearance and the novel localization method used for target localization. Afterwards, experimental results are given in section 4. Finally, section 5 concludes the paper.

2. RELATED WORK

2.1 Key Point Detection

Many key point detectors have been proposed in the literature so far. SIFT and SURF are two well-known key point detectors and descriptors [14, 15]. The key points detected by theses algorithms are invariant to scale and rotation variations. However, computational complexity of theses algorithms are demanding and thus their real time performance is poor. Considering the robustness of these descriptors, some researches aimed at improving these algorithms in term of speed [16, 17]. It is known that SURF is faster than SIFT because it approximates Gaussian derivatives with box filters and uses integral image to compute Gaussian blur. However, it is still not fast enough for tracking applications.

Corners are another type of interest points which are used in computer vision applications. Many algorithms have been proposed to detect such points [18, 19, 20, 21, 22]. In [18] the intensity change due to small shifts is approximated by Taylor series and followed by calculation of the weighted SSD. The applied weight is a Gaussian window for noise suppression. The weighted SSD is denoted by matrix form whose eigenvalues are used to determine the corner points. Furthermore, instead of calculating eigenvalues, a corner response based on the determinant and trace of the matrix is proposed for speed boost. In [19] the use of the minimum of the two eigenvalues is proposed if it exceeds a certain threshold limit. SUSAN corner detector [22] places a circular mask around the pixel to be checked. Neighboring pixels are compared with the center pixel also known as nucleus with an exponential function. The comparisons are summed to form the area of the SUSAN operator. Finally, given a geometric threshold, SUSAN operator's response is calculated as the difference between the threshold and its area if threshold exceeds the area. Finally, the corners can be found by non-maximum suppression. As feature detection must be fast due to its vital role in various applications, machine learning approaches have also been proposed to speed up the process [20, 21].

In contrast to the SIFT and SURF feature detectors, corner detectors do not have any associated descriptor. For describing corners, a patch around a corner is selected and is described by any descriptor. What must be considered is that these descriptors must be simultaneously discriminative and computationally efficient. There are many descriptors proposed in the computer vision literature, a review of which is given in the following section.

2.2 Image Descriptors

Target can be represented by its color, texture or shape features or a combination of them. A color descriptor [23] which uses color histogram for description is computationally efficient but this information alone is not sufficient. Therefore it is proposed to combine LBP texture feature [24] with color description. LBP which stands for local binary pattern was first introduced by [25] where local features are extracted by the comparison of neighboring pixels with the center pixel which makes the descriptor invariant to monolithic grey-scale changes. LBP features are then modified to be more discriminative for human detection in [26]. Histogram of oriented gradients [27] is another descriptor which uses gradient direction to determine the histogram bin index and the gradient magnitude as the voting measure. Centrist [28] is another feature descriptor mainly proposed for scene recognition. The descriptor must be calculated in block cells in order to code larger patches of the image. This descriptor is very efficient in term of computational complexity.

2.3 Tracking Methods

Based on the target's description, tracking can be done using Kalman or particle filtering or the mean shift method. Kalman and particle filtering both treat the tracking as a state estimation problem in a Bayesian inferential framework. Kalman filtering leads to promising results under additive Gaussian noise and linearity assumptions while particle filter can handle more complex (non-linear motion) situations with arbitrary noise [29]. In fact, particle filtering is a simulation based implementation of the conceptual Bayesian solution where a set of samples (particles) are drawn using importance sampling technique and is exploited because the true posterior is not available in practice [30]. In [31] Mean shift is used for target localization as well. Based on the appearance model, similarity measures are computed for some sample points of the next frame. The probability density function of the object is calculated using kernel density estimation. Finally, the target's position is found by locating the acquired pdf's maximum. An example which combines mean shift method and SIFT is proposed in [32].

The surrounding background of a target known as local context can provide substantial information for target localization. As the local context of the target does not change abruptly between two consecutive frames, its information can be used for localizing the target

$$\begin{array}{cccccc} 32 & 64 & 96 & & 1 & 1 & 0 \\ 32 & \mathbf{64} & 96 & \Rightarrow & 1 & & 0 \\ 32 & 32 & 96 & & 1 & 1 & 0 \end{array} \Rightarrow (11010110)_2 \Rightarrow \mathrm{CT} = 214$$

Fig. 1: Census Transform of a pixel and its corresponding decimal value [28]

in the subsequent frame [5]. In this paper inspired by the proposed method in [5] the target is localized given its detected key points. The following section presents the proposed tracker.

3. PROPOSED TRACKER

In this section, the various building blocks of the proposed visual tracker termed CCT are introduced. First the visual descriptor is introduced. Then the method for modeling the target's appearance is given and finally a new method for target localization based on its interest points by solving a convolution problem is proposed.

3.1 Visual Descriptor

The tracker uses Centrist as the main visual descriptor. This descriptor was first introduced for scene recognition [28] and successfully applied to the context of object detection [33]. Centrist descriptor is obtained by calculating the census transform followed by a local or global histogram calculation. Census transform compares the intensity of a pixel with its 8 neighboring pixels according to which a bit is assigned to each neighbor pixel [28] as shown in figure 1.

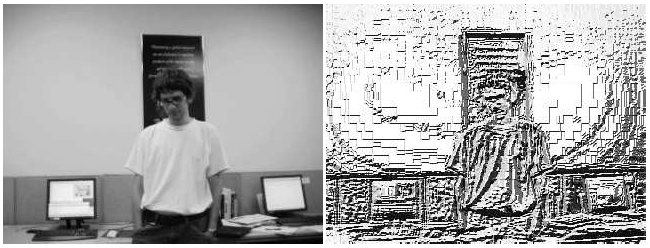
Given an input image I, the census transformed image (CT) is obtained by calculating CT value for all the pixels of image; therefore, the transformed image has the same size as the input image. An example of a transformed image along with the original image is given in figure 2. Considering the calculation of this transform, it is apparent that this descriptor is invariant to illumination variations and is easy to compute. The Centrist descriptor of the whole image is simply the histogram of the transformed image.

An experiment was designed in [28] in order to demonstrate this descriptor efficiency in encoding the image structures. In this experiment an image patch is first shuffled randomly in order to remove any existing image structure. Then exploiting the simulated annealing algorithm, pixels are substituted with each other until the Centrist descriptor of the original patch is reached. The reconstructed patch, has the same structural feature as the original patch. Another experiment which uses intensity histogram as an extra descriptor was designed in [33]. Based on these experiments, the Centrist descriptor in small image patches as well as the intensity histogram are used to describe the target's patch.

3.2 Appearance Modeling

As discussed in section 2.2, in order to capture the target's structural information, target must be divided into small patches to which the descriptors are applied. Given the target's position in the initial frame, two general possible solutions exist: 1. To divide the target into smaller patches and describe each patch using Centrist and intensity histogram as descriptors. 2. To select some repeatable points in the target's region and describe these points using local patches around each.

In the localization phase, solution number 1 is more time consuming than the second solution. For instance, assume a 20 by 20 pixel



(a) Original image (b) Transformed image

Fig. 2: Example of census transform

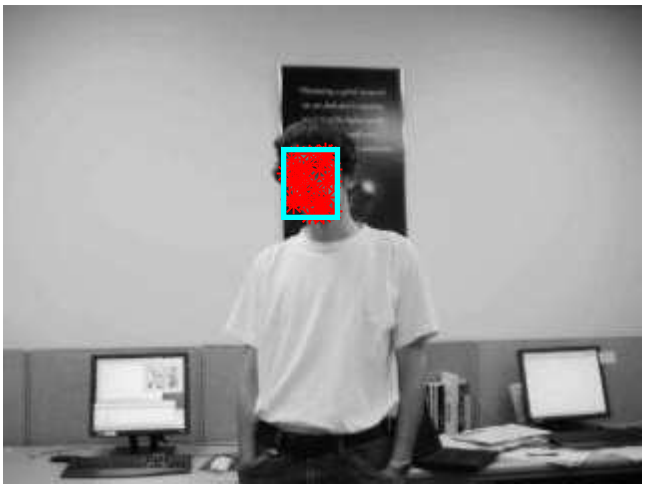


Fig. 3: Corners of a target. Target resides in the bounding box.

target in a 200 by 200 pixel image. In the first solution, a tracker must scan the whole image with a 20 by 20 pixel square and construct the descriptors for each candidate against which the target's model is compared. In this case 180^2 image patches must be considered and their descriptor must be calculated which affects the real time performance of the tracker. On the contrary, if target description is only applied to a certain target's patches, for instance patches around its key points, the localization phase only needs to find these set of key points and match the corresponding candidate descriptors with target's key points. Here, however the method used for detecting the key points plays an important role in real time performance of the final tracker. This process can be seen as narrowing the search space to only the key points of the upcoming frame. In this paper, corners are used as key points because their repeatability is satisfactory and their computational complexity is fairly low.

The target appearance model is obtained as follows: 1. The target's key points (corners) are detected (figure 3). 2. The Centrist descriptor along with the intensity histogram for each corner is calculated. In descriptor calculation, a patch around each key point is subdivided into cells of size n by n pixels (figure 4). Histograms obtained in each cell are finally concatenated to give the final feature vector.

Once the target's appearance is modeled, it can be found in the subsequent frame. In the next frame, first the key points are detected and then described. A sample of detected key points is shown in figure 5. These key point candidates are then compared against the target's key points and are matched according to the nearest neighbor rule. As a result, most similar corner points to the target's key

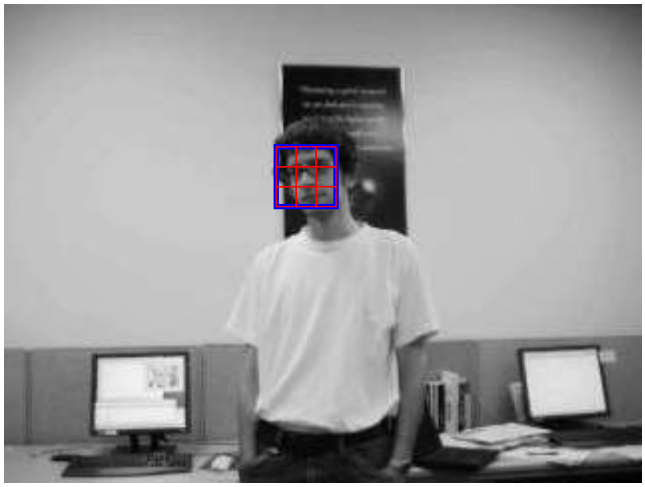


Fig. 4: Division of a large patch around a sample key point (blue square) into smaller patches (red squares) for efficient target description.

points are matched together. Figure 6 shows the matched corners for two consecutive frames.

3.3 Target Localization

In the tracking problem the position of the target must be estimated in each frame based on which the appearance model is updated. Therefore, it is very crucial to pinpoint the exact target's location given its detected corners. Nevertheless, the target's key points do not have any conceptual relation to the target's center which is needed to specify the target's location. Based on the given target's key points in the first frame and motivated by [5], a deconvolution problem is formulated through which the hidden relation can be found and used for target localization. In other words, given the target's corners along with its center position in the first frame, the objective is to find a function $weight(\cdot)$ which can relate target's corners and its center location as follows:

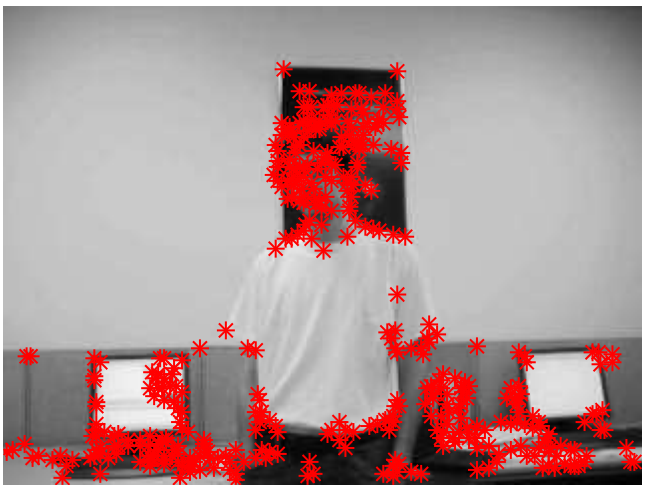
$$centermap(\mathbf{x}) = \sum_{\mathbf{z}} weight(\mathbf{x} - \mathbf{z}) \times cornermap(\mathbf{z}) \quad (1)$$

where, $centermap(\mathbf{x})$ is a map indicating the center of the target. In fact instead of specifying the center of the target with a tuple (x, y), a map which peaks at the target's center position is used. Then the weighted average of corners is calculated to give the target's center map. $cornermap(\mathbf{z})$ is the map indicating the corners' positions and $weight(\cdot)$ is the weighting function. Figure 7 illustrates corner points used as the corner map without any further processing, however it is also possible to smooth the corners with a Gaussian kernel before applying to equation 1. An example of the center map given in equation 1 for the sample target given is shown in figure 8.

Given the corner map and the center map, the weight function in equation 1 can be found using deconvolution, since equation 1 can be rewritten as:

$$centermap(\mathbf{x}) = weight(\mathbf{x}) \otimes cornermap(\mathbf{x}) \quad (2)$$

(a) Original image



(b) Corner key points

Fig. 5: A sample picture of a dataset with its detected key points.

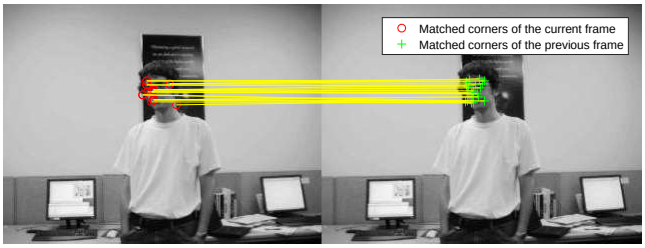


Fig. 6: Detection of the target's key points in the subsequent frame based on the Corner Centrist descriptor.

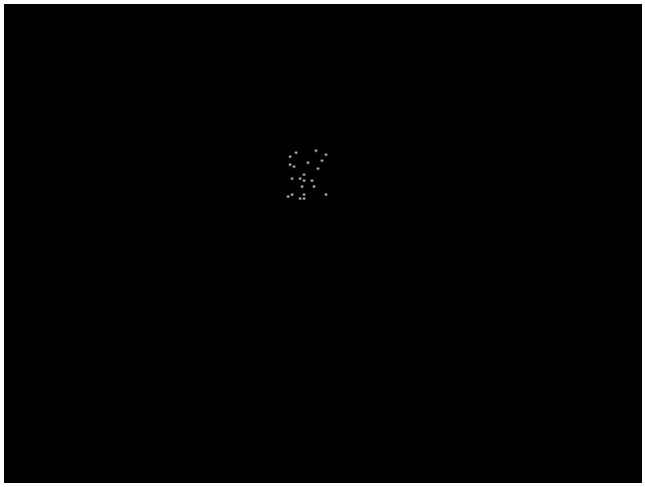


Fig. 7: Detected corners of the target as a map.

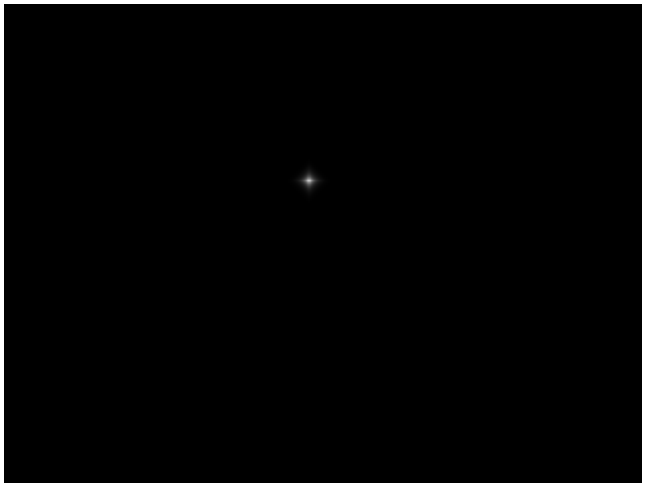


Fig. 8: A typical center map for target localization.

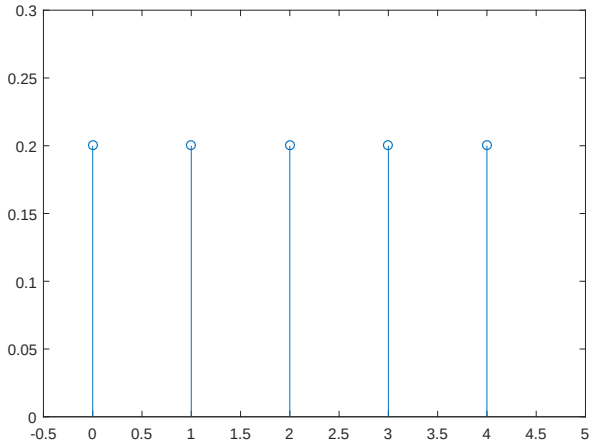
Therefore, the weight function can be easily calculated using DFT and inverse DFT as follows:

$$weight(\mathbf{x}) = idft(\frac{dft(centermap(\mathbf{x}))}{dft(cornermap(\mathbf{x}))}) \quad (3)$$

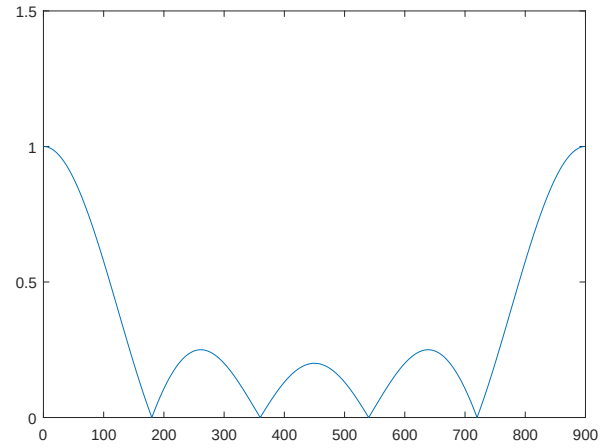
In the initialization step, the corner map is the same as the corners in the target's region. Using equation 3 the weight function can be found. In the subsequent frames, the objective is to find the center location of the target given its detected key points. In this phase using equation 2 center map is calculated, whose maximum gives the accurate target's location. Once the localization is done, the procedure for finding the weight is repeated and the weight function is updated as follows:

$$W(F) = (1-\alpha)W(F) + \alpha \times \frac{dft(centermap(\mathbf{x}))}{dft(cornermap(\mathbf{x}))} \quad (4)$$

where $W(F)$ is the discrete Fourier transform of the weight function and $cornermap(\cdot)$ is the constructed map for the corners matched with the target's corners. Since convolution converts to multiplication in the DFT domain, DFT of the center map is first obtained and then its inverse transform is calculated for estimating the target's location. The whole procedure is summarized in algorithm 1.



(a) A finite sequence



(b) The magnitude of the Fourier transform of the sequence

Fig. 9: A sample averaging filter with its frequency response magnitude.

A drawback of equation 3 is the risk of division by zero which may lead to error flow and performance degradation of the tracker. This issue can be solved by either adding a small value to the denominator (e.g. ϵ), ensuring division by zero will never happen or by taking the DFT in a way which ensures no zero element is sampled. To demonstrate the idea, let us consider the non-negative real-valued sequence $1/5 \times [1, 1, 1, 1, 1]$ and its corresponding DFT transform as shown in figure 9.

Since the N point discrete Fourier transform of any L-point sequence ($N \geqslant L$) is equivalent to sampling its frequency response with N, by choosing N to be prime with respect to the size of the sequence (L), no frequency with zero magnitude will be sampled given that the sequence is non-negative and real valued. An example of such sampling is given in figure 10. As can be seen zero entries of the frequency response are not sampled here and in the aforementioned case, this approach ensures that division by zero will never happen. To integrate this approach into the proposed lo-

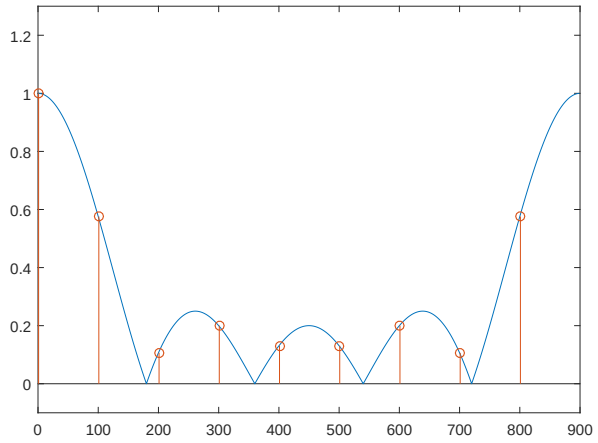


Fig. 10: Specific sampling to avoid zero terms.

calization method, an area around the detected corners of size M by M pixels is selected. M must be large enough to include all the key points and must be selected in a way that is prime to N which is used for DFT calculation.

Algorithm 1 CCT visual tracker

1: Read the initial frame
2: Specify the target's position
3: Find the target's corners
4: Describe a patch around each corner with the given descriptor
5: Find the relation between target's corners and its center position through equation 3
6: **while** There is more images in the sequence **do**
7: Read the next image of the sequence
8: Find the corner key points of the image and calculate their descriptors
9: Match these key points' descriptors with the key points' descriptors of the target using the nearest neighbor rule
10: Calculate the target's center position using equation 2
11: Update the $weight(\cdot)$ function using equation 4
12: **end while**

4. EXPERIMENTAL RESULTS

In this section the parameters used in the designed tracker are presented. Quantitative and qualitative results of the proposed tracker on a challenging dataset[1] are given afterwards.

4.1 Parameter Adjustment

The results of the tracker shown in the subsequent sections are based on the following parameters:

—Patch of size 30 by 30 pixels around each corner is considered.
—The patch is divided into 3 by 3 blocks resulting in blocks of size 10 by 10 pixels each.
—128 bin cells are used for histogram calculation.
—Parameter α used for updating the weight function is set to 0.095.
—The center map function used is an exponential function as follows:

$$\exp^{-(|\frac{x-x_0}{2.25}|+|\frac{y-y_0}{2.25}|)} \tag{5}$$

—For corner detection, the OpenCV implementation [34] of the algorithm proposed in [19] is used.
—In the localization phase, M is chosen to be the smallest power of 3 which can contain all the key points and N to be equal to $2^{(\log_3 M+1)}$. This choice for M and N will guarantee that division by zero will never occur.
—In corner point matching, the descriptors of the corners detected in the two last sequences are used.

4.2 Quantitative Results

In this section the performance of the proposed algorithm is analyzed using the conventional methods used for evaluating visual trackers [35]. Success rate plots and precision plots are used to demonstrate the overall efficiency of the proposed algorithm. Furthermore, the plots of position error versus frame number is also presented.

In success rate plots the ratio of intersection of the detected target (r_t) with its ground truth (r_{gt}) to their union is calculated (formula 6).

$$S = \frac{|r_t \cap r_{gt}|}{|r_t \cup r_{gt}|} \tag{6}$$

where $|\cdot|$ denotes the number of pixels in the region. Success in tracking is defined when the measure S exceeds a specified threshold. In success rate plots, success rate percentage is plotted versus the different thresholds varying from 0 to 1 [35].

Precision plot uses the Euclidean distance between the center of the detected target and the labeled center of the target. In these plots, if the calculated distance falls below a certain threshold the detection is considered successful. Again the percentage of the true detection against various thresholds are plotted as precision plots [35].

Based on the definitions, success rate measure takes the overlap ratio of the tracked results with the desired results into consideration, therefore the corresponding plots are more accurate for benchmarking. Thus, in order to rank different trackers, the area under the curve (AUC) of their success rate plot is used [35].

The success rate and precision plots (figure 11-12) are obtained as follows:

(1) Some challenging datasets are tracked and the tracked bounding boxes are obtained.
(2) The percentages required for the plots are calculated using all the tracked frames.
(3) Our results are compared with some state of the art algorithms. These algorithms are MIL [36], CT [37], TLD [13], FRAG [38], DFT [39]. The tracker benchmarks reported in [35] is used for comparison.

The sequences used for performance evaluations are: 1. *Man*, 2. *David2*, 3. *MountainBike*, 4. *Walking*, 5. *Walking2*, 6. *Mhyang*, 7. *CarDark*, 8. *Coupon*. Challenges such as illumination variation, background clutter, scale variation, pose deformation, occlusion, and in and out of plane rotation are included in these sequences.

[1]The dataset is available online at `http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html`

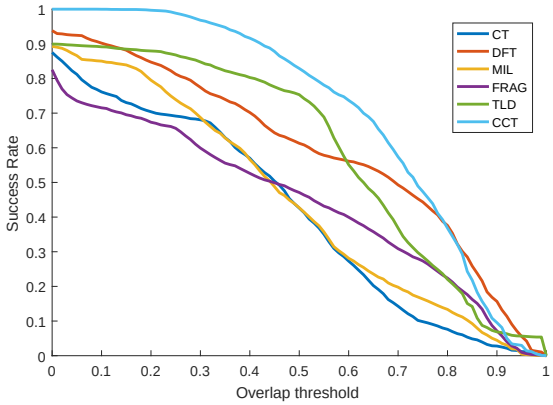


Fig. 11: Success rate plot of the CCT algorithm along with some state of the art algorithms.

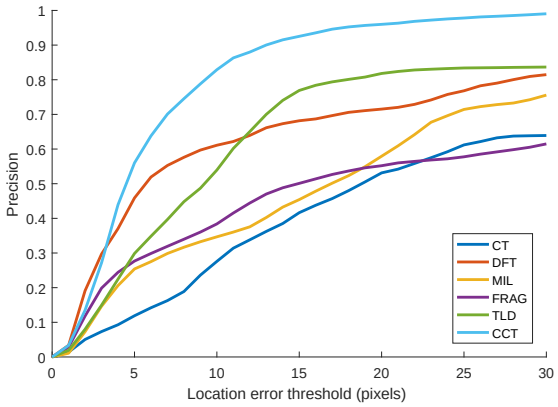


Fig. 12: Precision plot of the CCT algorithm along with some state of the art algorithms.

Based on the plots in figure 11 and 12, the proposed CCT tracker performs much better in comparison with the given state of the art algorithms. In figure 11, the CCT tracker has the highest success rate for nearly all the thresholds. It is only around threshold 0.8 and 0.9 and above that the success rate of DFT and TLD algorithms, respectively exceeds the success rate of the proposed CCT tracker. The precision plots (figure 12) also demonstrate the accuracy of the CCT tracker. The proposed tracker is the most accurate one except for errors less than about 4 pixels where the DFT tracker performs slightly better.

Given the success rate plots, the AUC of each visual tracker is calculated and used for ranking. Table 1 shows the visual trackers based on their AUC score in descending order. Accordingly, the proposed tracker has the highest AUC and DFT and TLD algorithms are the second and third algorithms, respectively.

The position error plots of these algorithms are also depicted in figure 13 (a) - (e).

In figure 13 (a) the position error plots of the visual trackers tested on the *Walking* test sequence are plotted. In this sequence, the main challenges are the scale variation and deformation of the target. The performance of the MIL tracker is superior to other trackers.

Table 1. Area under the curve of success rate plots of the competent algorithms.

Algorithm	AUC
CCT	0.7032
DFT	0.5928
TLD	0.591
MIL	0.4465
FRAG	0.4426
CT	0.4129

Figure 13 (b) shows the same quantity for the *Walking2* test sequence where the main challenge is occlusion. In this case, the MIL tracker has the least position error for up to around 200th frame while for the rest of the test sequence the proposed CCT tracker's error is the least. In fact after occlusion the MIL tracker fails to track the target correctly.

The main challenges of the *David2* test sequence whose position error plots are depicted in figure 13 (c) are in and out of plane rotation of the target. The position error of the DFT algorithm is less than other algorithms for up to about frame 290. After this frame number the two most accurate algorithms are CCT and TLD but the overall performance of the proposed CCT tracker is better.

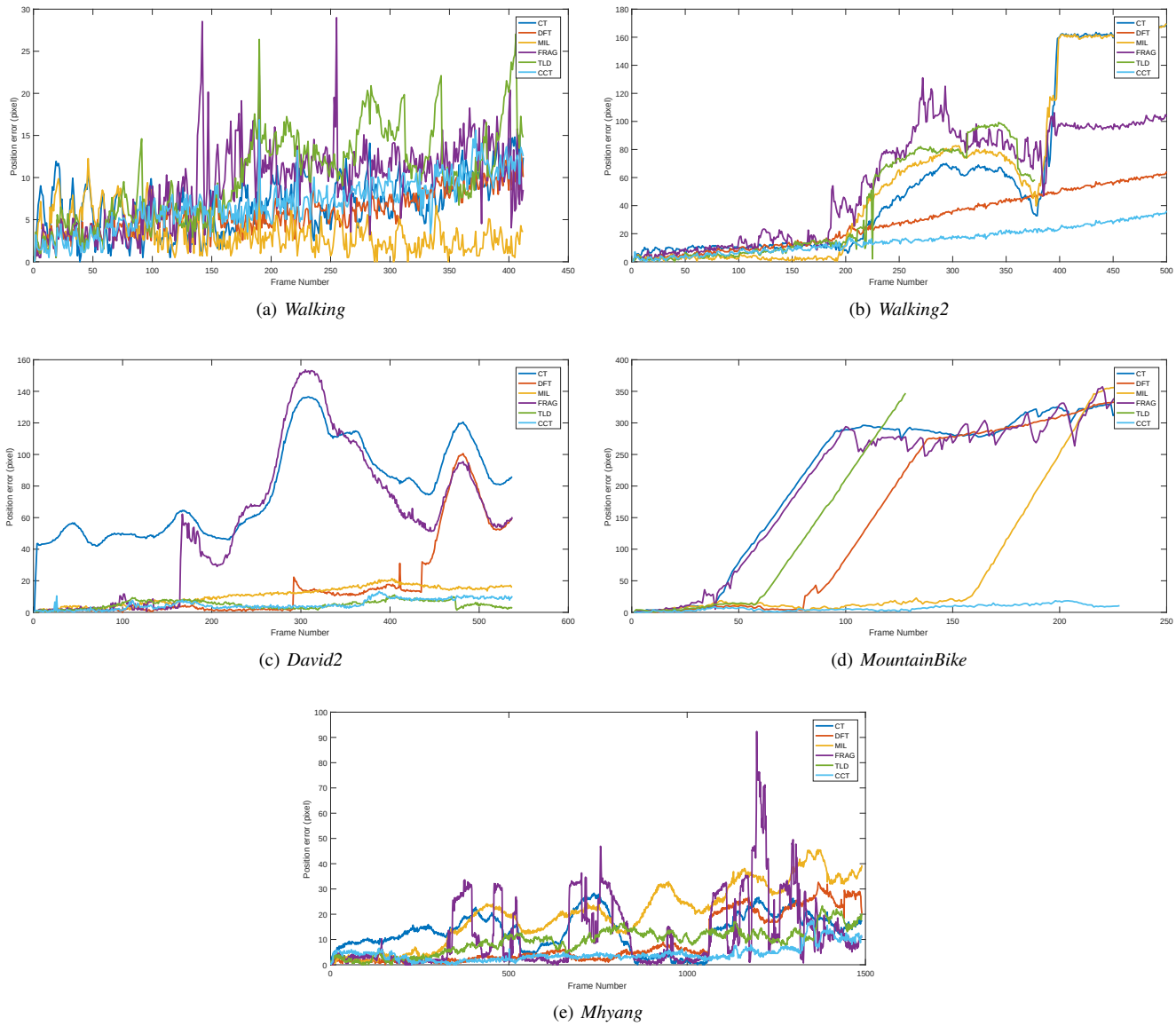
Figure 13 (d) shows the position errors of the algorithms on the *MountainBike* test sequence where the main challenges are background clutters and in and out of plane rotation. In this case, first FRAG and CT algorithms lose the target and then TLD, DFT, and MIL lose the target about frame numbers 60, 80, and 160, respectively. Nevertheless, the proposed algorithm can track the target accurately to the end of this test sequence.

In figure 13 (e) the position error plots of the algorithms on the *Mhyang* test sequence are given. In this test sequence the main challenges are illumination variations and deformation. In this case, as illustrated in figure 13 (e), the proposed CCT tracker performs much better than the competing algorithms. FRAG has the worst performance in this case.

4.3 Qualitative Results

The resulting bounding box of the proposed tracker along with bounding box positions of other trackers as well as the ground truth positions are depicted in figure 14.

The first row of figure 14 shows results of the visual trackers on the *Mhyang* sequence. As can be seen, FRAG and MIL trackers fail to locate the target in frame number 1215 while the CCT tracker performs smoothly for all the sequence's frames. The second row of the figure depicts the results on the *MountainBike* sequence. As can be seen, the CCT tracker never misses the target. The visual results on the *David2* sequence is shown in the third row. CT and FRAG trackers are not even close to the target in the second image plotted for this sequence (frame number 341). From the other three images of this sequence shown in figure 14 it is apparent that the DFT tracker loses the target as well and the three successful trackers in this case are CCT, TLD, and MIL. The fourth row of the figure shows images from the *Walking2* sequence. Based on the figure, the performance of the proposed CCT tracker is the best. The last row of the figure demonstrates results of the trackers on the *Walking* sequence. In this case all the trackers perform favorably well.



(a) *Walking* (b) *Walking2*

(c) *David2* (d) *MountainBike*

(e) *Mhyang*

Fig. 13: Position error plots of different algorithms on different test sequences.

5. CONCLUSION

In this paper, based on the Centrist capability to encode target's structure, the tracker termed CCT is proposed which uses Centrist and intensity histogram around the target's key points to model its appearance. Matching of key points are made using the simple nearest neighbor method with the Euclidean distance metric. In addition, a convolution problem is formulated to relate the target's center position to its key points. Furthermore, the idea of Fourier sampling is exploited in order to avoid the error flow. The quantitative and qualitative results of the proposed tracker signify the CCT's robustness and efficiency.

While there is a good reason to use key points for tracking and limit the search space to these key points, there exists a drawback which happens when the target contains no key points or too few key points. One solution is to increase the sensitivity of the key point detector when the number of the detected key points falls below a certain threshold.

An approach for relating the target's corners to its center position is also proposed which is general and can be exploited whenever it is desired to relate some key points to special points in an image. This approach worked perfectly for the localization of the target but it is dependent on the target's key points detected in each frame which in turn is dependent to the descriptor in use; hence, the next direction to improve the proposed tracker is to either use a more robust descriptor or improve the robustness of the proposed localization algorithm.

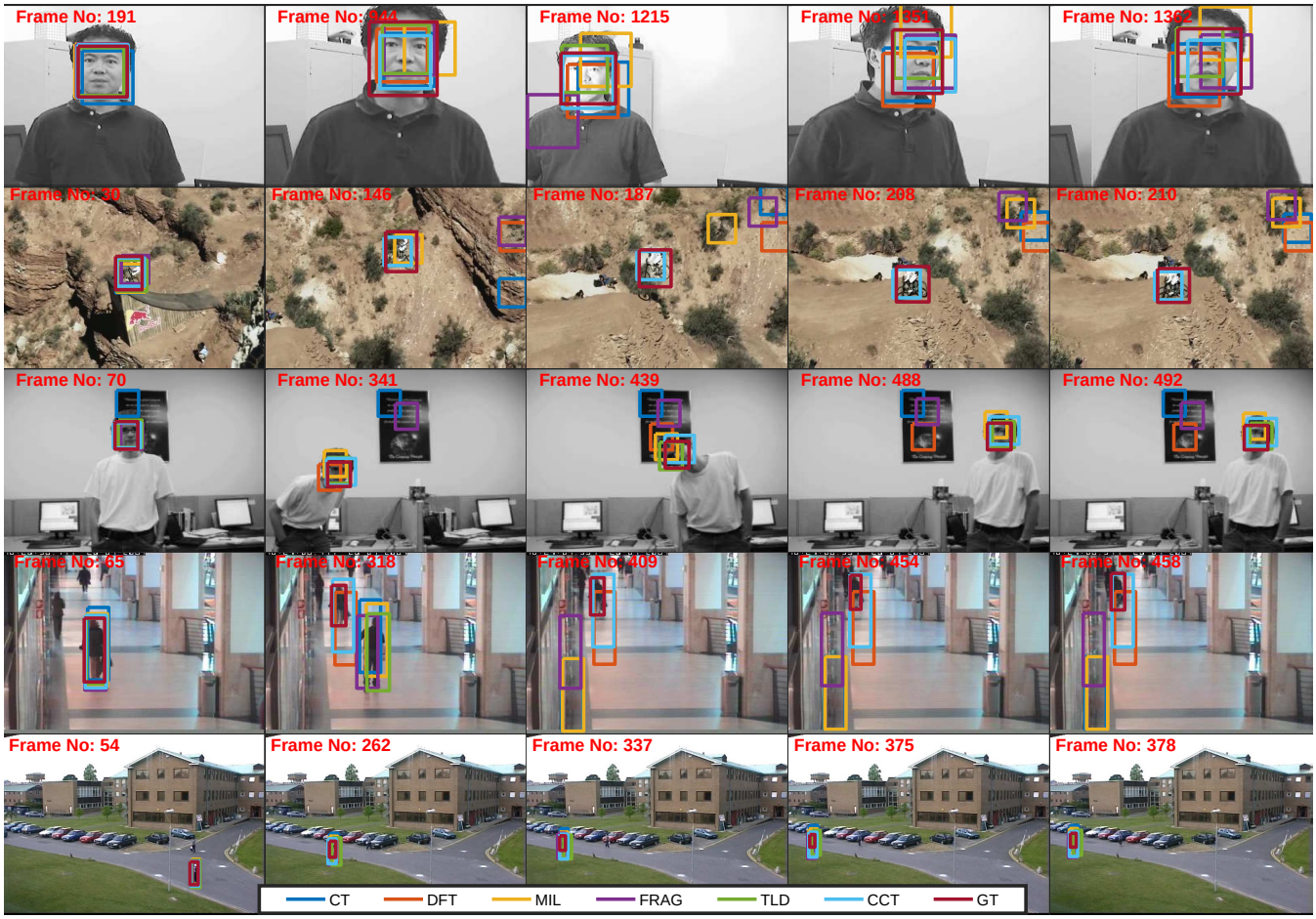


Fig. 14: Results of different algorithms on some challenging test sequences (From top to bottom: *Mhyang*, *MountainBike*, *David2*, *Walking2*, *Walking*)

6. REFERENCES

[1] N. Jiang, W. Liu, and Y. Wu. Learning adaptive metric for robust visual tracking. *IEEE Trans Image Process*, 20(8):2288–300, 2011. Jiang, Nan Liu, Wenyu Wu, Ying eng Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, Non-P.H.S. 2011/02/22 06:00 IEEE Trans Image Process. 2011 Aug;20(8):2288-300. doi: 10.1109/TIP.2011.2114895. Epub 2011 Feb 17.

[2] Javier Cruz-Mota, Michel Bierlaire, and Jean-Philippe Thiran. Sample and pixel weighting strategies for robust incremental visual tracking. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(5):898 – 911, 2013.

[3] Can-Long Zhang, Zhong-Liang Jing, Han Pan, Bo Jin, and Zhi-Xin Li. Robust visual tracking using discriminative stable regions and k-means clustering. *Neurocomputing*, 111(0):131–143, 2013.

[4] A. Satpathy, X. Jiang, and H. L. Eng. Lbp-based edge-texture features for object recognition. *IEEE Trans Image Process*, 23(5):1953–64, 2014. Satpathy, Amit Jiang, Xudong Eng, How-Lung eng 2014/04/03 06:00 IEEE Trans Image Process. 2014 May;23(5):1953-64. doi: 10.1109/TIP.2014.2310123.

[5] Kaihua Zhang, Lei Zhang, Qingshan Liu, David Zhang, and Ming-Hsuan Yang. *Fast Visual Tracking via Dense Spatio-temporal Context Learning*, pages 127–141. Springer International Publishing, Cham, 2014.

[6] X. Lan, A. J. Ma, P. C. Yuen, and R. Chellappa. Joint sparse representation and robust feature-level fusion for multi-cue visual tracking. *IEEE Trans Image Process*, 24(12):5826–41, 2015. Lan, Xiangyuan Ma, Andy J Yuen, Pong C Chellappa, Rama eng Research Support, Non-U.S. Gov't 2015/09/29 06:00 IEEE Trans Image Process. 2015 Dec;24(12):5826-41. doi: 10.1109/TIP.2015.2481325. Epub 2015 Sep 23.

[7] Si Chen, Shaozi Li, Songzhi Su, Donglin Cao, and Rongrong Ji. Online semi-supervised compressive coding for robust visual tracking. *Journal of Visual Communication and Image Representation*, 25(5):793–804, 2014.

[8] Y. Sui, S. Zhang, and L. Zhang. Robust visual tracking via sparsity-induced subspace learning. *IEEE Trans Image Process*, 24(12):4686–700, 2015. Sui, Yao Zhang, Shunli Zhang, Li eng Research Support, Non-U.S. Gov't 2015/08/11 06:00 IEEE Trans Image Process. 2015 Dec;24(12):4686-700. doi: 10.1109/TIP.2015.2462076. Epub 2015 Jul 30.

[9] Y Wu, B Shen, and H Ling. Visual tracking via online non-negative matrix factorization. *Circuits and Systems for Video Technology, IEEE Transactions on*, 24(3):374 – 383, 2013.

[10] D. Wang, H. Lu, Z. Xiao, and M. H. Yang. Inverse sparse tracker with a locally weighted distance metric. *IEEE Trans Image Process*, 24(9):2646–57, 2015. Wang, Dong Lu, Huchuan Xiao, Ziyang Yang, Ming-Hsuan eng Research Support, Non-U.S. Gov't 2015/05/03 06:00 IEEE Trans Image Process. 2015 Sep;24(9):2646-57. doi: 10.1109/TIP.2015.2427518. Epub 2015 Apr 28.

[11] L. Zhang, H. Lu, D. Du, and L. Liu. Sparse hashing tracking. *IEEE Trans Image Process*, 25(2):840–9, 2016. Zhang, Lihe Lu, Huchuan Du, Dandan Liu, Luning eng Research Support, Non-U.S. Gov't 2015/12/20 06:00 IEEE Trans Image Process. 2016 Feb;25(2):840-9. doi: 10.1109/TIP.2015.2509244. Epub 2015 Dec 17.

[12] W. Guo, L. Cao, T. X. Han, S. Yan, and C. Xu. Max-confidence boosting with uncertainty for visual tracking. *IEEE Trans Image Process*, 24(5):1650–9, 2015. Guo, Wen Cao, Liangliang Han, Tony X Yan, Shuicheng Xu, Changsheng eng Research Support, Non-U.S. Gov't 2015/03/15 06:00 IEEE Trans Image Process. 2015 May;24(5):1650-9.

[13] Z. Kalal, J. Matas, and K. Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 49–56.

[14] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[15] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[16] L. C. Chiu, T. S. Chang, J. Y. Chen, and N. Y. Chang. Fast sift design for real-time visual feature extraction. *IEEE Trans Image Process*, 22(8):3158–67, 2013. Chiu, Liang-Chi Chang, Tian-Sheuan Chen, Jiun-Yen Chang, Nelson Yen-Chung eng 2013/06/08 06:00 IEEE Trans Image Process. 2013 Aug;22(8):3158-67. doi: 10.1109/TIP.2013.2259841.

[17] S. Ehsan, N. Kanwal, A. F. Clark, and K. D. McDonald-Maier. An algorithm for the contextual adaption of surf octave selection with good matching performance: best octaves. *IEEE Trans Image Process*, 21(1):297–304, 2012. Ehsan, Shoaib Kanwal, Nadia Clark, Adrian F McDonald-Maier, Klaus D eng Research Support, Non-U.S. Gov't 2011/06/30 06:00 IEEE Trans Image Process. 2012 Jan;21(1):297-304. doi: 10.1109/TIP.2011.2160869. Epub 2011 Jun 27.

[18] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK.

[19] Shi Jianbo and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600. Institute of Electrical and Electronics Engineers (IEEE), 1994.

[20] E. Rosten, R. Porter, and T. Drummond. Faster and better: a machine learning approach to corner detection. *IEEE Trans Pattern Anal Mach Intell*, 32(1):105–19, 2010. Rosten, Edward Porter, Reid Drummond, Tom eng 2009/11/21 06:00 IEEE Trans Pattern Anal Mach Intell. 2010 Jan;32(1):105-19. doi: 10.1109/TPAMI.2008.275.

[21] Edward Rosten and Tom Drummond. *Machine Learning for High-Speed Corner Detection*, pages 430–443. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[22] Stephen M. Smith and J. Michael Brady. Susana new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

[23] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[24] Jifeng Ning, L. E. I. Zhang, David Zhang, and Chengke Wu. Robust object tracking using joint color-texture histogram. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(07):1245–1263, 2009.

[25] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[26] Mu Yadong, Yan Shuicheng, Liu Yi, T. Huang, and Zhou Bingfeng. Discriminative local binary patterns for human detection in personal album. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[27] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1. Institute of Electrical and Electronics Engineers (IEEE).

[28] J. Wu and J. M. Rehg. Centrist: A visual descriptor for scene categorization. *IEEE Trans Pattern Anal Mach Intell*, 33(8):1489–501, 2011. Wu, Jianxin Rehg, James M eng Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, Non-P.H.S. 2010/12/22 06:00 IEEE Trans Pattern Anal Mach Intell. 2011 Aug;33(8):1489-501. doi: 10.1109/TPAMI.2010.224. Epub 2010 Dec 23.

[29] Victor H. Diaz-Ramirez, Kenia Picos, and Vitaly Kober. Target tracking in nonuniform illumination conditions using locally adaptive correlation filters. *Optics Communications*, 323(0):32–43, 2014.

[30] Du Yong Kim and Moongu Jeon. Spatio-temporal auxiliary particle filtering with-norm-based appearance model learning for robust visual tracking. *Image Processing, IEEE Transactions on*, 22(2):511–522, 2013.

[31] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, may 2002.

[32] Huiyu Zhou, Yuan Yuan, and Chunmei Shi. Object tracking using sift features and mean shift. *Computer Vision and Image Understanding*, 113(3):345–352, 2009.

[33] J. Wu, N. Liu, C. Geyer, and J. M. Rehg. C4: a real-time object detection framework. *IEEE Trans Image Process*, 22(10):4096–107, 2013. Wu, Jianxin Liu, Nini Geyer, Christopher Rehg, James M eng Research Support, Non-U.S. Gov't 2013/06/26 06:00 IEEE Trans Image Process. 2013 Oct;22(10):4096-107. doi: 10.1109/TIP.2013.2270111. Epub 2013 Jun 19.

[34] G. Bradski. *Dr. Dobb's Journal of Software Tools*, 2000.

[35] Y. Wu, J. Lim, and M. H. Yang. Online object tracking: A benchmark. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2411–2418.

[36] B. Babenko, M. H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 983–990.

[37] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang. Real-time compressive tracking, 2012.

[38] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 798–805. Institute of Electrical and Electronics Engineers (IEEE).

[39] Laura Sevilla-Lara. Distribution fields for tracking, 2012.