International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.6 (2016), pp.9-24
http://dx.doi.org/10.14257/ijsip.2016.9.6.02

Visual Multiple Object Tracking with Improved Feature Detection and Kinect Color Depth Map

Kajal Sharma*

Flat No. 301, Building 206, Piorville Apartment, 24 Namyang-Dong, Seongsangu,
Changwon, Korea
Independent Researcher
kajal175@gmail.com

Abstract

Tracking multiple objects in real-time videos represents a challenging area in the era of computer vision. This paper proposes a new method to track the multiple objects under different environment conditions such as rotation, illumination, blurred, occlusion, and many others. In addition, the kinect color depth image processing is used to estimate the distance of the objects. The tracking of multiple objects is formulated as classification task which competitively use the object features in the different video frames of the video sequences. To obtain the optimal configuration of feature classification, a neural network based framework is presented to make a global influence based on winner pixel estimation between the video frames. The objects are tracked efficiently in less time as compared with SIFT techniques and distance of objects is calculated with kinect based depth image processing. Experimental results are given for real-time scenes, and many experiments are conducted to examine the performance of the proposed approach. The proposed method resulted into efficient tracking of multiple objects in various conditions including rotation, scaling, occlusion, etc. The distance of multiple tracked objects is estimated using the kinect depth processing.

Keywords: *Object Tracking; Image processing; Neural networks; SIFT; SOM; Kinect Sensor*

1. Introduction

Visual multiple object tracking is becoming an important topic of research within the vision science, robotics, and image processing community. Many researchers have provided object tracking algorithms to suit a variety of applications which include automated surveillance, video indexing, human-computer interaction, robot guidance, and traffic monitoring [1-2]. The goal of visual object tracking is to determine the position of the object in video sequences and reliably against dynamic scenes with vision sensor. A number of elegant algorithms have been proposed in the literature for visual object tracking. Varieties of approaches are based on target representation and localization, and assume a probabilistic model for the object's appearance in order to estimate its location. More specifically, color, shape, salient feature descriptors of the object masked by an isotropic kernel is used to create a histogram [3]. A characteristic method in this category is the mean shift algorithm [4] and its extensions [5-6], where the transformation of the object state is obtained by finding the maximum of a similarity function based on color histograms. These methods track only one object at a time. In other techniques, high dimensionality feature vectors are used to represent the visual objects. Avidan integrates a support vector machine classifier into an optic-flow-based tracker [6-7]. Unfortunately learning of classifiers under such a feature space is computationally expensive. Other methods have been proposed in order to simultaneously track multiple objects [8-9].

Multiple objects are tracked using graph cuts over some observations (*i.e.* possible locations of the object) with min-cut/max flow optimizations [10-11].

Grabner *et al.* [12] proposed an online semi-supervised boosting method to handle the drift problem where labeled examples come from the first frame only, and subsequent training examples are left unlabeled. This approach suffers from the tracking drift problem with the change in appearance. Yu *et al.* [13] proposed a gradient-based feature selection approach with online boosting with the objective of both feature selection and weak classifier updating. In [14-15] feature-based approaches have been used for feature matching and 3D tracking of objects for the efficient detection of objects. In multiple instance learning [14], overlapping examples of the target are put into a positive and negative bags. These bags are passed on to the learner, which is therefore allowed more flexibility in finding a decision boundary. These online versions of trackers are limited in terms of efficiency since their updating costs for high dimensional features are computationally expensive. Other algorithms employ level-sets to represent each object to be tracked by optimally grouping regions whose pixels have similar feature signatures [16-17]. An augmented reality to perform object recognition is employed in [18] to perform object recognition. In [18], the highly-distinctive scale invariant feature transform (SIFT) features are used to provide robust tracking under scene changes. However, this approach is computationally expensive in the offline-training phase. To overcome the high computational time limitation of Lowe's SIFT, this paper introduces an improved method for the efficient tracking of multiple objects along with feature matching using the neural network method.

In this work, a novel approach is introduced for the vision based multiple object tracking. The tracking is performed by determining object keypoints using an improved feature matching technique. The improved SIFT keypoints are used to extract the features in the multiple target objects. A novel approach is presented to match the multiple target object features with self-organizing map (SOM) which is based on winning pixels estimation. The tracked object distance is computed with kinect color depth processing. The various colors are assigned to the tracked objects which are used to find the object distance from the camera. The rest of the paper is organized as follows: Section 2 gives a brief overview of the SOM and kinect depth processing. The procedure of feature extraction and tracking in order to track the multiple objects is presented in section 3. The experimental results are shown in section 4 and conclusions are drawn in section 5.

2. Self-Organizing Map and Kinect Color Depth Processing

2.1. Self-Organizing Map

In this paper, the fast computation of feature vector for different objects is done using SOM network. SOM is a competitive network, which clusters or visualizes high-dimensional input feature vectors into low-dimensional output feature vectors. It is an unsupervised neural network algorithm with the capability of topology-preserving characteristics. The neurons in the network are connected to the adjacent ones according to a neighborhood relationship as presented in Figure 1. The most common topologies are the rectangular and hexagonal ones. SOM consists of a regular two-dimensional output grid map units connected via weights with *n* input feature vectors. In the proposed work, the SIFT scale space features vectors are used as an input to the SOM network.

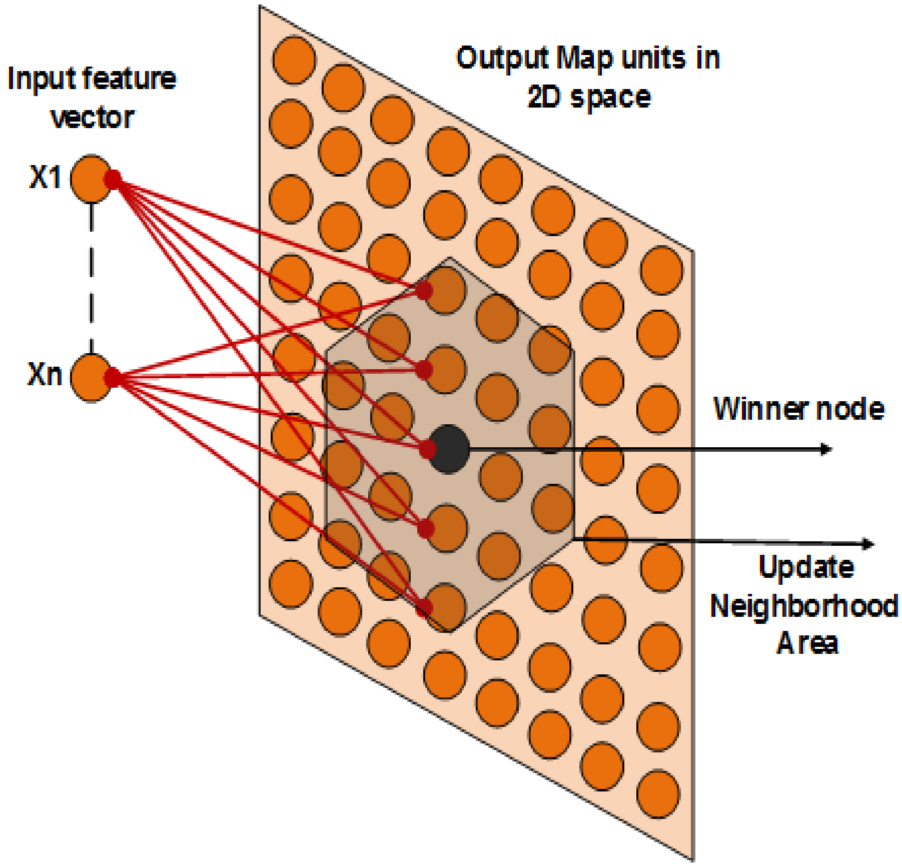


Figure 1. Network Diagram for Competitive Learning with Input Feature Vector and Output Neuron Grid of Size 8X8 Map

In the SOM network, the similar feature points in the input space are mapped onto close points in the output space according to the defined neighborhood relationship. Competitive learning procedure is used for the learning process in the SOM network. During the learning phase, the input feature vectors were supplied to the network and the Euclidean distances are computed between the input feature vector (x_s) and the nodes in the network. The node (m) with the shortest Euclidean distance is selected as the winner node and is called as the best matching unit (BMU). The winner node is given by (1):

$$m = \min_{j} \left\{ \left[\sum_{i} \left(x_{si} - w_{ji} \right)^2 \right]^{1/2} \right\} \tag{1}$$

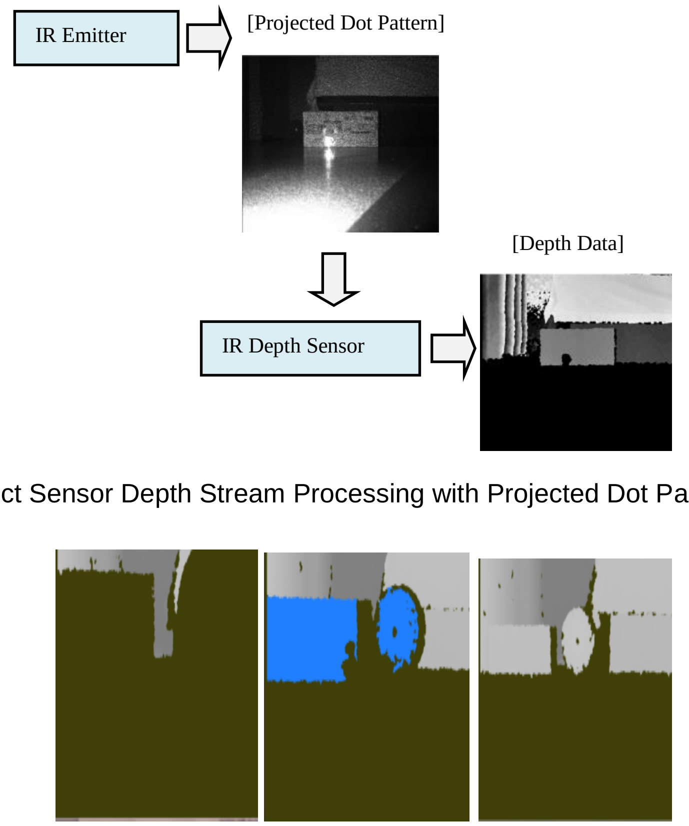
where m denotes the winner node, x_{si} is the i^{th} element of the input feature vector x_s. w_{ji} denotes the i^{th} weight of the neuron j. The winner node is the center of the update neighborhood area which is an area where node and their associated weights are updated according to the learning rule. For the input feature descriptor, this process is repeated iteratively and the winning nodes converge to the input feature vectors at the learning rate α^s accordingly. The weight for the p^{th} node in the s^{th} step be W_{ps}, the input vector is X_i and the learning rate for the s^{th} step be α^s. The winner node is updated by the following equation:

$$W_m^{s+1} = W_m^s (1 - \alpha^s) + X_i \alpha^s = W_m^s + \alpha^s (X_i - W_m^s) \tag{2}$$

The learning rate and the size of the update neighborhood decrease throughout the learning process. As a result of the training stage, the SOM grid folds onto the input data feature sets, like a flexible net, placing a larger number of neurons in areas with a higher density of data. After learning, the map units are distributed across the input feature space. While the other nodes within its neighborhood, as defined by its region-of-influence, are subsequently moved closer to the same feature point.

2.2. Kinect Color Depth Processing

The depth data is necessary for the design of distance based applications. The successive depth image frames of Kinect sensor combines to form the depth stream. The depth stream consists of a 16-bit grey scale stream with a field of view having 43 degrees vertical and 57 degrees horizontal range. The pixel of each depth stream contains the distance in millimeters between the Kinect device and the objects in front of the device. The Kinect sensor consists of an IR emitter and an IR depth sensor (Figure 2). The depth data in each depth stream is represented by the *X* and *Y* coordinates [19]. The IR emitter and an IR depth sensor work together to produce the desired depth map of the scene. The IR depth sensor reads the dots in the scene, processes the depth data and sends the processed reflected depth information.



(a) Kinect Sensor Depth Stream Processing with Projected Dot Pattern

(b) Depth Stream of An Object for Near Object and Far Object with Change in Color from Black to White

Figure 2. Kinect Depth Technology with Infrared Dot Pattern Generation and Depth Stream Formation. It Represents Near or Far Objects Position with Change in Color from Dark to Light

To obtain the 3D position of points in an image frame, stereo triangulation is done to obtain the two different views of a scene. The relative depth information is calculated by comparing the two images. The Kinect sensor depth vision ranges from around 800 mm to approximately 4000 mm (2.6 feet to 13.1 feet). The sensor returns 16-bit raw depth frame in which the first three bits of depth data are used to represent the identified players and the remaining 13 bits are used to measure distance in millimeters. The 16-bit raw depth data is converted into a 32-bit RGB frame to identify the range of distances with different colors. The pixel values having distance information are represented by RGB color value.

3. Proposed Multiple Object Tracking Algorithm

This section consists of detailed explanation of the proposed method for tracking of multiple objects in the video sequences. The proposed methodology focuses on tracking of multiple objects using an improved feature matching method. The proposed technique consists of two steps as shown in Algorithm 1 and Algorithm 2: one is the "detection module", that detects the feature descriptor with keyframes of environment and the known objects, and another is the "tracking module", that relies on frame-to-frame tracking. The tracking module runs in the foreground. To do so, the SIFT keypoints of multiple target objects are extracted first from a given reference frame. The Algorithm 1 is used for feature retrieval and generation of the descriptor vector. The proposed algorithm is based on a self-organizing map to identify the keypoints and winner pixels. The result is a list of reduced feature points of the targets, sorted by similarity with the input reference frame. These features are then used in the tracking module to track the multiple target objects in the video sequences under different scenarios. Figure 3 shows block-diagram of the proposed multiple object tracking system; the main steps are detailed in the following sub-sections.

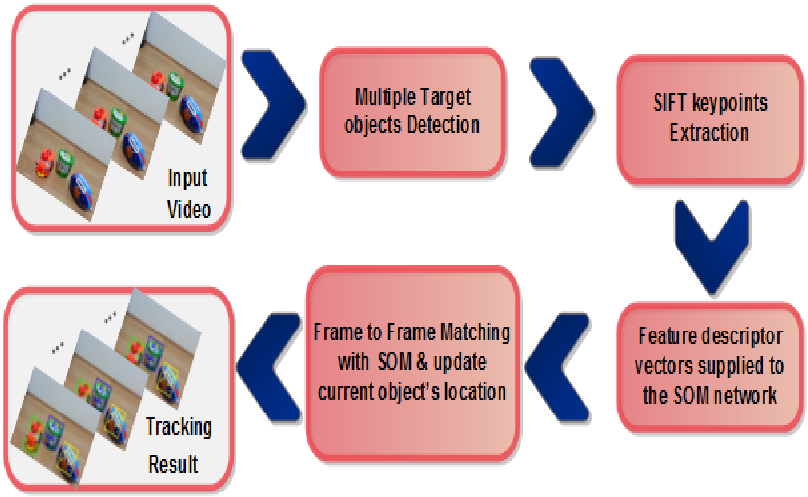

Input
Video
Multiple Target objects Detection
SIFT keypoints Extraction
Feature descriptor vectors supplied to the SOM network
Frame to Frame Matching with SOM & update current object's location
Tracking
Result

Figure 3. Block-Diagram of the Proposed Multiple Objects Tracking System

3.1. Target Objects Feature Vector Generation

In this section, the competitive learning method is explained to reduce the dimension of the feature vectors. In order to track the keyframe for the target objects in the video sequence, the features were obtained from the improved SIFT method (Algorithm 1). It is necessary to extract keypoints in the reference keyframe for the target objects in the video sequence. The pseudo-code for detecting the reduced features of multiple target objects is given in Algorithm 1. The keypoint features are then passed to the tracking module as an input to the SOM network

(Figure 1). The role of the detection module is to provide the optimized feature points at the i^{th} position for object M_i to the tracking module continuously.

$$M_i \leftarrow \bigcup_{j=1}^{j=h_i} F(X, V_{ij}) \qquad (3)$$

where F returns the set of the feature keypoints between SIFT features of the reference frame X and video frame V_{ij}. V_{ij} is j^{th} videoframe for the i^{th} object. h_i is the maximum number of video frames in the input video sequence. The exhaustive step by step description of the keypoint feature vector reduction is given in Figure 3. The keypoint features of the multiple target objects can be extracted by shape, texture, and color or correlation regardless of these targets are moving or stationary. For example, the three objects as shown in Figure 4 is successfully recognized with the SIFT features. SIFT feature detector extracts from an image a number of attributed regions in a way which is consistent with (some) variations of the illumination, rotation and other viewing conditions. The descriptor associates to the regions a signature which is a 3-D spatial histogram of the image gradients in characterizing the appearance of a keypoint. The gradient at each region pixel is regarded as a sample of a three-dimensional elementary feature vector, formed by the pixel location and the gradient orientation. Samples are weighed by the gradient normalization and accumulated in an 8 bin histogram over the direction of the gradient, which forms the SIFT descriptor of the feature space.

Algorithm 1. Pseudo-code for Multiple Object Detection

```
Algorithm 1. Pseudo-code for multiple object detection
Input: Video sequence consists of multiple target objects
Output: Features of multiple objects
for the first keyframe in the video sequence do the following:
Select the multiple target objects in the acquired video sequence
Create candidate objects to be track in the subsequent video frames of the video sequences
Extract the SIFT keypoints for the multiple objects to be track in the video sequence
 for frame=1: $N$
Find the feature vector with the SIFT method for multiple candidate objects. Assume
weights of each feature point corresponding to pixel at $(i, j)$ with intensity $w_3^{ij}$ be $(w_1^{ij}, w_2^{ij}, w_3^{ij})$.
For each feature point in the multiple objects, initialize a node with corresponding
coordinates and intensity as the initial weights. These feature vectors are supplied as inputs
to the SOM network.
  end
end
```

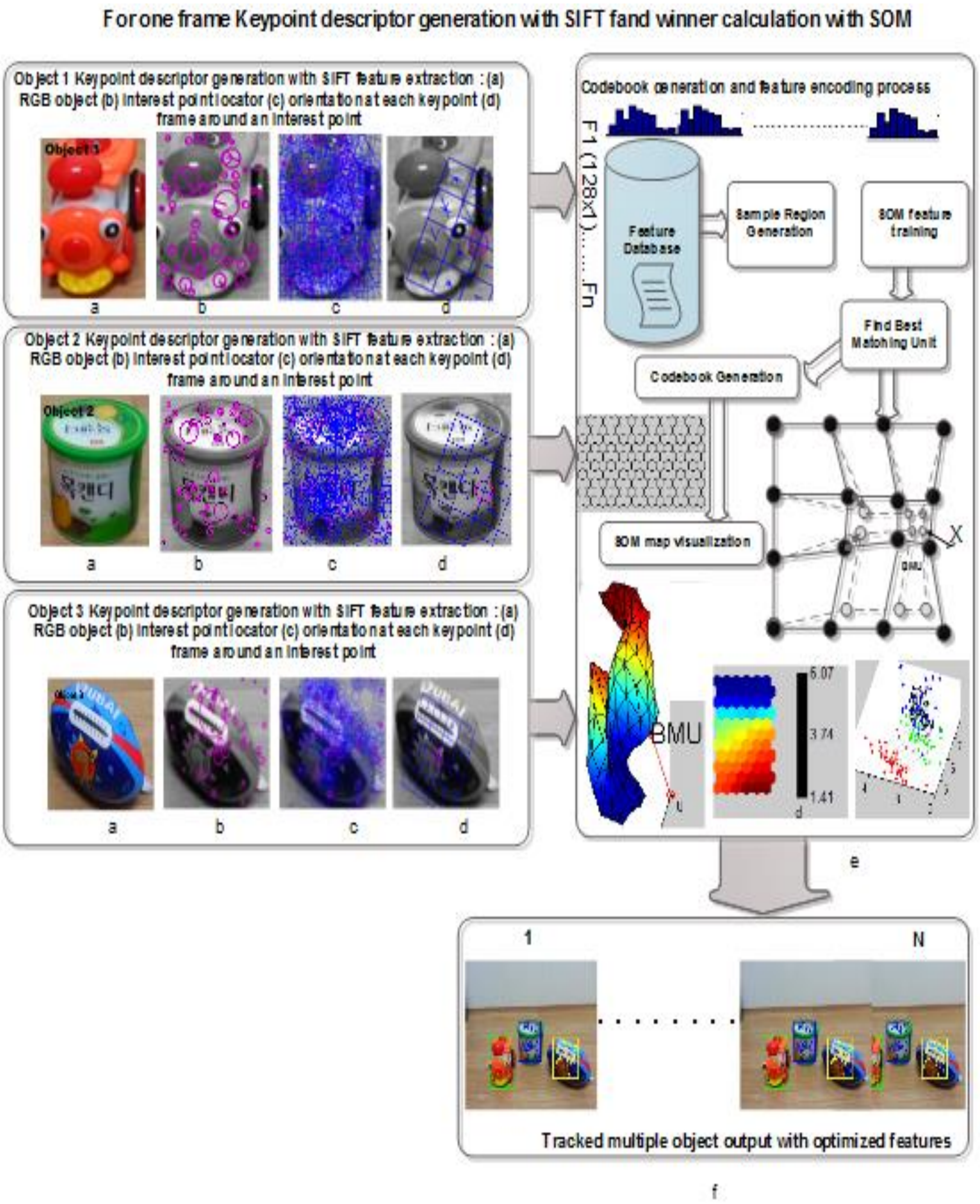

For one frame Keypoint descriptor generation with SIFT fand winner calculation with SOM
Object 1 Keypoint descriptor generation with SIFT feature extraction : (a) RGB object (b) interest point locator (c) orientation at each keypoint (d) frame around an interest point
Object 1
a
b
c
d
Object 2 Keypoint descriptor generation with SIFT feature extraction : (a) RGB object (b) interest point locator (c) orientation at each keypoint (d) frame around an interest point
Object 2
a
b
c
d
Object 3 Keypoint descriptor generation with SIFT feature extraction : (a) RGB object (b) interest point locator (c) orientation at each keypoint (d) frame around an interest point
a
b
c
d
Codebook generation and feature encoding process
F1 (128x1)......Fn
Feature Database
Sample Region Generation
SOM feature training
Find Best Matching Unit
Codebook Generation
SOM map visualization
X
BMU
BMU
6.07
3.74
1.41
d
e
1
N
Tracked multiple object output with optimized features
f

Figure 4. The Overview of the Proposed Method for Tracking Multiple Objects with Feature Generation: (a) Frame 1 of the "Toy-ball" Sequence (b) Keypoint Descriptor Patches from the Interacting Region (c) Orientation Map Showing that Features are Classified Into Patches. Blue Value Represents Higher Orientation at Each Feature Descriptor (d) a Frame around an Interest Point which is Oriented According to the Dominant Gradient Direction (e) Each Neuron is Represented by a 128 Dimensional Weight Vector. The SOM is Trained Iteratively and Best Matching unit (BMU) is Computed. Update the BMU and its Neighbors Towards the Input Feature Descriptor *X*. The Solid and Dashed Lines Represent the Situation before and after Updating, Respectively. The End Result is a Low-Dimension Feature Map of the Descriptor Vector (f) Tracking Results

The histograms are concatenated to form one 128 dimension feature vector which is passed onto the SOM network to generate codebook. Each neuron in SOM is represented by a 128 dimensional weight vector. The SOM is trained iteratively and best matching unit is computed. The end result is a low-dimension feature map of the descriptor vector. The next subsection discusses the tracking module to track multiple objects with the use of optimized SIFT features detected in the reference keyframe.

3.2. Multiple Object Tracking with Improved Feature Matching

The tracking module used the information after the detection module has finished the feature generation and optimization. The SOM is modified to deal with the matching process between the reference keyframe and the subsequent frames in the video sequence. The input frame is added as a reference keyframe and contributes to the feature matching if it passes several conditions. Algorithm 2 gives the detail pseudo-code of the tracking procedure. Given the set of optimized reduced keypoints extracted from the input keyframe *S*, the tracking module returns a list of keyframes *K* sorted by similarity with the input reference keyframe. S_I and S_J represent the set of pixels in the reference keyframe and the set of pixels in the next video frame, respectively. The set of position vectors of the pixels is given by $S_x=\{x_x, y_x\}$. The matching is accomplished between the pixels of the reference keyframe and the next frame. The similar winner pixels are tracked with the modified SOM algorithm, resulting into matching of the features in the video sequence (see Algorithm 2). This provides an improvement over the Lowe's SIFT algorithm in terms of correct matches and computation time. The differences of the corresponding matched keypoints are computed for the multiple target objects between the reference keyframe and the subsequent frames. If the computed difference is less than or equal to the threshold value, then the matched pair is selected as the stable keypoint. The matched feature keypoints are then obtained from the following equation:

$$MF\ (P,Q) = \sqrt{(x_P - x_Q)^2 + (y_P - y_Q)^2} \tag{4}$$

If the $MF(P,Q)$ is less than the Euclidean distance, it is accepted and considered to be the matched point in the keyframes. Otherwise it is detected as false match and is rejected.

(x_P, y_P) is the location of the reference target object region, and (x_Q, y_Q) is the location of the next frames in the video sequence. Finally, the proposed matching method result only the stable keypoints with efficient and correct matched keypoints. Based on the minimum distance, the winner neuron is selected with the nearest neighborhood procedure in the SOM network and the matched winning feature set consists of location-matched keypoints. In accordance, for each object feature in the reference keyframe, the corresponding matched features are determined for all the objects.

Algorithm 2. Pseudo-Code for Multiple Objects Tracking with Improved Feature Matching

```
Algorithm 2: Pseudo-code for multiple objects tracking with improved feature matching
Input: current frame, next frames, set of feature vectors for each object from Algorithm 1
Output: invariant matched features for each object, tracked objects
for all the video frames in the video sequence do the following:
frame to frame matching: select a random pixel from next video frame, and the corresponding
reduced feature vector obtained from the SIFT is supplied to the SOM network. Let
$(\alpha_1^{ij}, \alpha_2^{ij}, \alpha_3^{ij})$ be the input feature vector corresponding to pixel at $(m, n)$.
for the $(m, n)$ th input in the network, the minimum distance winning neuron node is denoted by
the index $(x, y)$ and is given by the following equation:
$(x, y) = \arg\min_{i,j} \left[ \sum_{k=1}^{3} (w_k^{ij} - \alpha_k^{mn})^2 \right]^{1/2}$
For all the neuron weight vectors the first two components are updated by the following
equations:
$(w_k^{ij} \leftarrow w_k^{ij} + h_k(i', j') g_k(\Delta I)(\alpha_k^{(m+i')(n+j')} - w_k^{ij})$
          where $i' = i - x$
$j' = j - yh_k(i', j') = \eta_k \exp\left( - \frac{i'^2 + j'^2}{2\sigma_{hk}^2} \right) g_k(\Delta I) = \exp\left( - \frac{(\Delta I)^2}{2\sigma_{gk}^2} \right) \Delta I = (w_3^{xy} - w_3^{ij})$
for $k$ =1, 2, and $\forall$ $i$, $m$ $\in$ {1, 2,…,$M$}, $\forall$ $j$, $n$ $\in$ {1, 2,…,$N$}, $\eta_k$ denotes the standard learning
rate and $\sigma_{hk}, \sigma_{gk}$ denotes the neighborhood parameters. Repeat all the above steps for a
predetermined number of $N_x$ cycles where $N_x$ =100 x $MN$.
Draw a box around all the detected features in the current matching frame and update the
current object's location.
Multiple target objects are tracked and combination of tracked frames resulting into tracked
video sequence.
End
```

4. Results and Discussion

The proposed method has been tested and evaluated in a series of image sequences demonstrating challenging tracking scenarios. In all experiments, a PC with an Intel 2.5 GHz Quad-core CPU and MATLAB is used for the implementation of the algorithms. 640×480 image resolution was used in all experiments. The four sequences (‘‘Toy-ball”, ‘‘Multiple different objects”, ‘‘Toy-ball sequence (Illumination)” and ‘‘Multiple different objects (Illumination)”) consist of 1460, 2529, 764, and 470 frames, respectively (Table 1). Individual objects are identified through the use of different features for their region and through features located on object regions. Thus, an object is successfully tracked if it maintains the same features in all occurrences. Overall, the proposed method managed to successfully track all objects in all of these videos in less time. To evaluate the efficiency of the proposed method, the results have been compared with the mean-shift tracking algorithm [4], the SIFT algorithm [18], and the multiple instance learning algorithm [14]. Characteristic snapshots of two video sequence demonstrating tracking results in different conditions are shown in Figure 5.

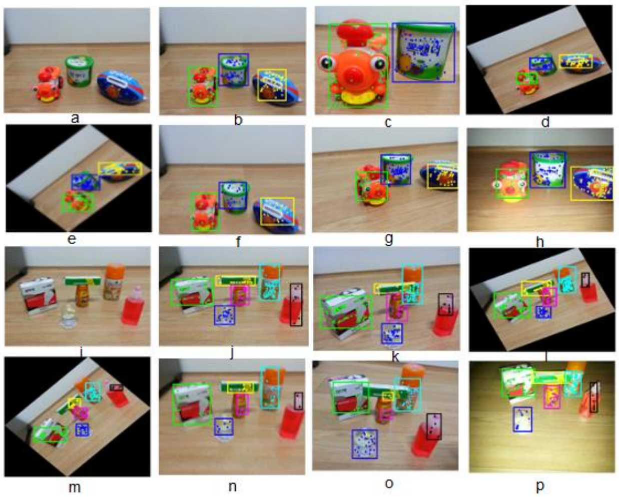
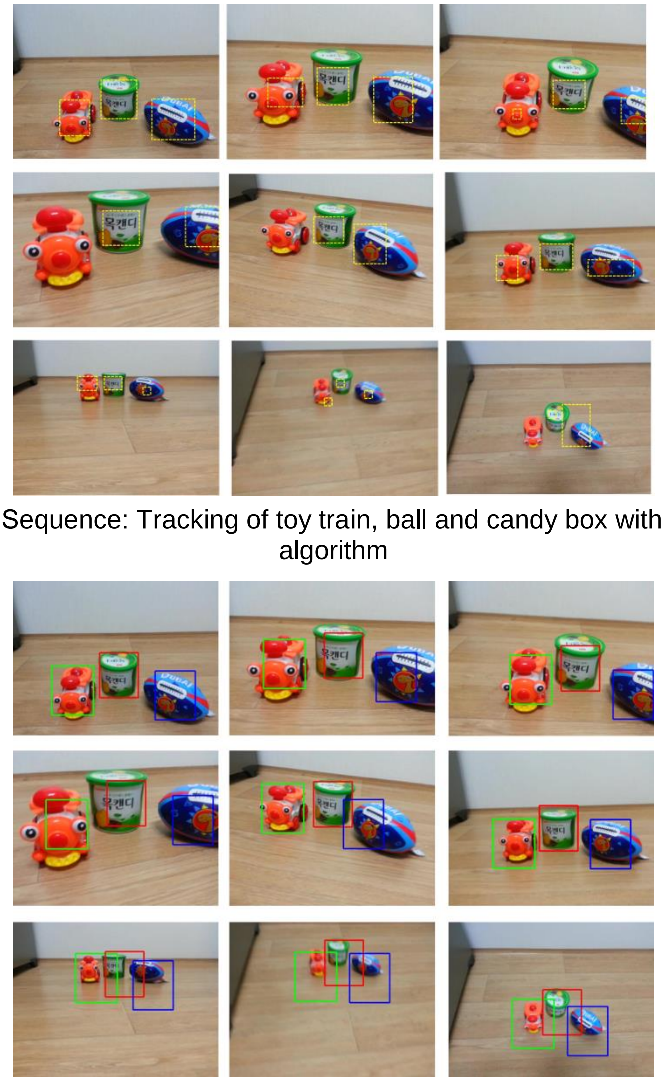


Figure 5. Output Results of the Proposed Approach: The Proposed System Continuously Tracks the Multiple Objects in the Video Sequence. Row I & II ‘‘Toy-ball” Video Sequence (a) Multiple Target Objects: Tracking of Toy Train, Ball and Candy Box, the Feature Descriptor Keypoints Extraction Inside the Multiple Target Objects. (b)-(e) Multiple Target Objects Tracked in the Simple, Scaling and Rotation Condition with Objects features Represented as Dots Inside Rectangles, (f)-(h) Multiple Target Objects Tracked in the Blurred, Occluded and Illumination Condition. Row III & IV ‘‘Multiple Different Objects” Video Sequence (i) Multiple Target Objects: Tracking of Combination of Different Objects (j)-(m) Multiple Different Objects Tracked in the Simple, Scaling and Rotation Condition with Objects Features Represented as Dots Inside Rectangles, (n)-(p) Multiple Different Objects Tracked in the Blurred, Occluded and Illumination Condition

The proposed tracking algorithm was used to track the multiple objects in the video sequences. The tracking results based on the proposed algorithm for the video sequences in different conditions are shown in Figure 5. The first two rows in Figure 5 show the tracking results obtained from the ‘‘Toy-ball” video sequence in scaled, rotation, blurred, illumination and occluded condition. The third and fourth rows in Figure 5 show the tracking results from the ‘‘Multiple different objects” video sequence in scaled, rotation, blurred and illumination condition. Figure 6 shows the results obtained from the tracking of three objects in the “Toy-ball” video sequence using the SIFT tracking algorithm and the proposed tracking algorithm. Figure 7 shows the tracking results for the six different objects in ‘‘Multiple different objects” sequence using the SIFT tracking algorithm and the proposed tracking method. The tracking of single objects with mean-shift tracking algorithm, SIFT tracking algorithm, multiple instance learning, and neural network based tracking is presented in the recent research for different video sequences [20]. The target object is lost in the mean-shift tracking algorithm based on comparing the histogram. It has the drawback of tracking object if it is lost and reappears later in the video sequence. The proposed method results are also compared with SIFT tracking algorithm [18] which can generate invariant features in image scale, noise, illumination and local geometric distortion. Due to the high computational complexity, the SIFT algorithm requires high computation of feature descriptor which is reduced by the proposed methodology.



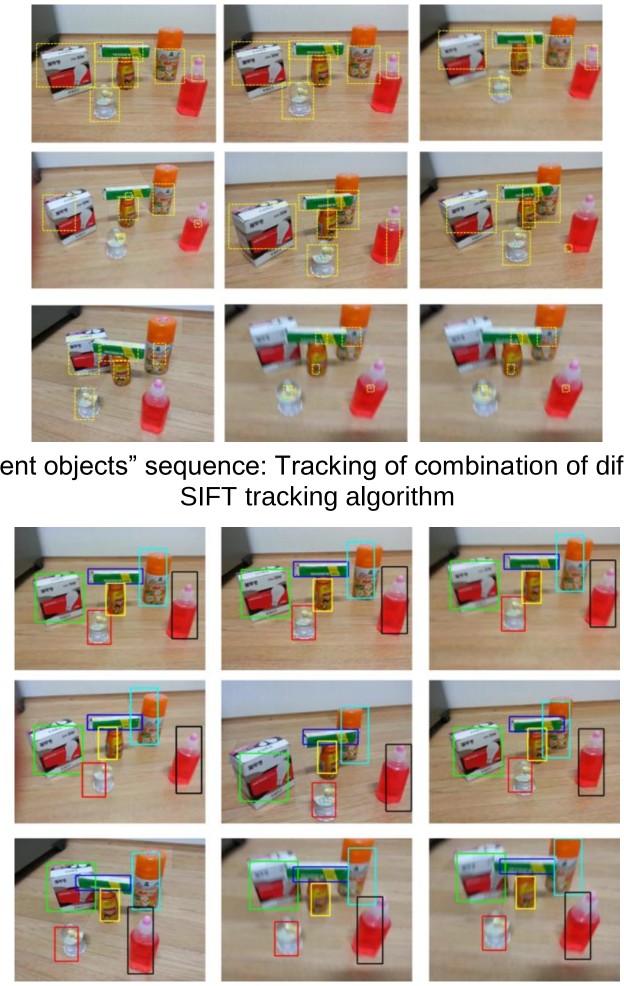
(a) "Toy-ball" Sequence: Tracking of toy train, ball and candy box with SIFT tracking algorithm

(b) "Toy-ball" sequence: Tracking of toy train, ball and candy box with proposed tracking algorithm

Figure 6. The Results Obtained for the "Toy-ball" Video Sequence with SIFT and Proposed Algorithm (a) SIFT-Based Object Tracking for the "Toy-Ball" Sequence: Tracking the 3 Objects for Frames No. 65, 175, 202, 386, 538, 837, 975, 1345, and 1435. (b) Proposed Method-Based Object Tracking for the "Toy-ball" Sequence: Tracking the 3 Objects for Frames No. 65, 175, 202, 386, 538, 837, 975, 1345, and 1435

The results are also compared with the multiple instance learning algorithm [14] where samples requires high computational complexity to update online classifier.

Thus, the proposed tracking method overcomes the difficulties of the mean-shift, SIFT tracking algorithm and multiple instance learning algorithm. The proposed method can tracks the multiple target objects using less computational time. The proposed tracking method can efficiently track the object even though the object is lost once and subsequently reappears. The proposed approach is based on SOM-based feature matching between the object region of the multiple selected objects and the objects in the subsequent video frames. The descriptor vector for multiple objects was computed using the SIFT method and the dimensions of the features were down-sampled by factor of 2.



(a)“Multiple different objects” sequence: Tracking of combination of different objects with SIFT tracking algorithm

(b) “Multiple Different Objects” Sequence: Tracking of Combination of Different Objects with Proposed Tracking Algorithm

Figure. 7. The Results Obtained for the “Multiple Different Objects” Video Sequence with SIFT and Proposed Algorithm (a) SIFT-Based Object Tracking for the “Multiple Different Objects” Sequence: Tracking the 6 objects for Frames No. 65, 175, 202, 292, 538, 837, 912, 1345, and 1435. (b) Proposed Method-Based Object Tracking for the “Multiple Different Objects” Sequence: Tracking the 6 Objects for Frames No. 65, 175, 202, 292, 538, 837, 912, 1345, and 1435.

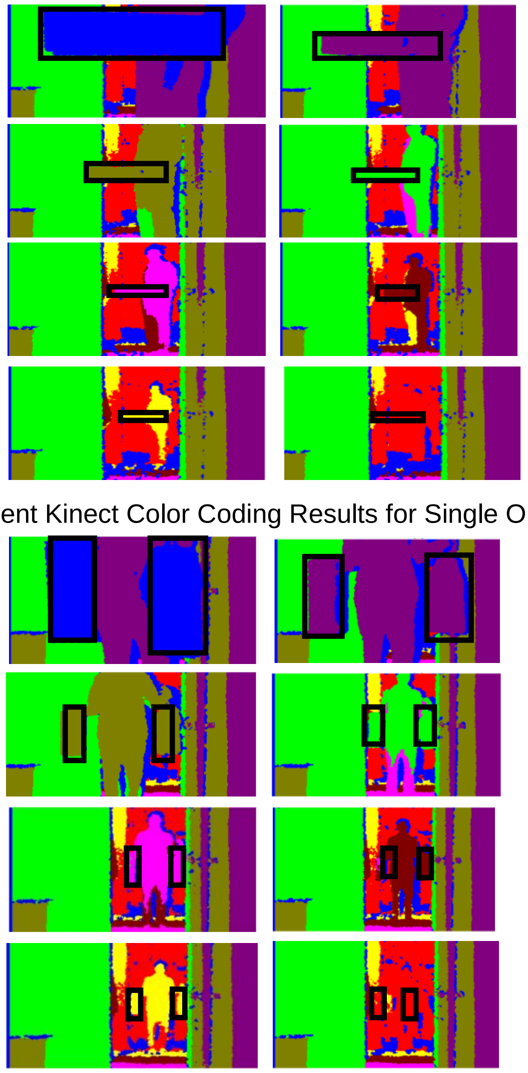
The optimum features were selected using the SOM dimension reduction method. The number of extracted features in the lower octave with respect to the higher octave is decreased by 4 times due to the down-sampling by a factor of 2 in both image directions. The size of the descriptor vectors was reduced with the dimension reduction method of SOM. The proposed algorithm was able to locate the multiple objects accurately and consistently in less time in all the video frames. The average comparison of the computation time for the four video sequences is given in Table 1. The average time taken by the mean shift tracking algorithm, SIFT tracking algorithm, multiple instance learning was 0.092908 seconds, 0.125786 seconds and 0.222852 seconds. While the average tracking time was significantly reduced to 0.025287 seconds with the proposed multiple object tracking method.

Table 1. Details of Four Video Sequences (“Toy-ball”, “Multiple Different Objects”,“Toy-Ball (Illumination)” and “Multiple Different Objects (Illumination)”) used in the Evaluation and Comparison of Results with Mean-Shift Tracking, SIFT-Tracking Technique, Multiple Instance Learning and Proposed SOM Based Multiple Tracking

Video Sequence name		Toy-ball sequence	Multiple different objects	Toy-ball sequence (Illumination)	Multiple different objects (Illumination)
Frames per sec		30 fps	30fps	30fps	30fps
Frame size		640 × 480	640 × 480	640 × 480	640 × 480
Video length		00:50 sec	01:25 sec	00:26 sec	00:16 sec
Frame number		1460	2529	764	470
Object number		3	6	3	6
Average computation time	Mean-shift tracking	0.092908	0.097427	0.092036	0.100710
	SIFT-tracking technique	0.125786	0.096363	0.112212	0.103203
	Multiple instance learning	0.222852	0.224551	0.219267	0.219049
	Proposed SOM based multiple tracking	0.025287	0.019918	0.024345	0.027393

The feature matching cost was reduced by 1.5 times as compared with the mean-shift algorithm, SIFT and multiple instance learning algorithms. The proposed tracking method consistently tracked multiple objects in different conditions with less time. After obtaining the tracking results, kinect based color processing is used to determine the near or far objects. The sensor range varies from minimum 800 mm to maximum 4000 mm with distance ranges from 2.6 ft. to 13.1 ft. With the use of C sharp programming the objects distance can be computed by coloring different range area with different colors. It can be observed from Figure 8, blue color is used to identify the objects if the object range is less than 800 mm and distance is less than 2.6. If the range is between 800 mm and 1500 mm with distance between 2.6 ft. and

5 ft., purple color is assigned. If the range is between 1500 mm and 2000 mm with distance between 5 ft. and 6 ft., olive color is assigned. If the range is between 2000 mm and 2500 mm with distance between 6 ft. and 7 ft., lime color is assigned. If the range is between 2500 mm and 3000 mm with distance between 7ft. and 8 ft. fuchsia color is assigned. Similarly maroon, yellow and red colors are used for the subsequent region space to obtain the object distance.



(a) Different Kinect Color Coding Results for Single Object

(b) Different Kinect Color Coding Results for Multiple Objects

Figure 8. Kinect Color Coding Results for Single Object and Multiple Objects to Calculate the Distance of Object from Kinect Sensor. The Color Ranges from Blue to Red and Distance Ranges from 2.6 ft. to 13.1 ft

5. Conclusion

A new method is proposed for tracking multiple objects under rotation, illumination, occlusion and blurred conditions. Feature matching between the different frames was done using a winner calculation method which highly reduces computational cost. With an improved SOM based feature matching algorithm, the tracker discriminates an object from its interacting objects by a feature classification process. Various experiments on several different sequences show that the proposed method is effective in tracking multiple objects. The kinect distance methodology is used to estimate the objects distance with color processing. Experiments verified that the proposed method could produce better solutions in fast and efficient object tracking. It can be used to develop various challenging real time applications with clutter, partial occlusions, image blur and non-uniformly colored objects.

References

[1] J. Gaspar, N. Winters and J. Santos-Victor, "Vision-based navigation and environmental representations with an omnidirectional camera," IEEE Trans. on Robotics and Automation., vol. 16, no. 6, **(2000)**, pp. 890–898.

[2] T. Boult, X. Gao, R. Micheals and M. Eckmann, "Omni-directional visual surveillance," Image and Vision Comput., vol. 22, no. 7, **(2004)**, pp. 515–534.

[3] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking," IEEE Trans. on Pattern Analysis and Machine Intelli., vol. 25, no. 5, **(2003)**, pp. 564–577.

[4] D. Comaniciu, V. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean shift," IEEE Conf. Computer Vision and Pattern Recog., Hilton Head Island, South Carolina, vol. 2, **(2002)**, pp. 142-149.

[5] H. Zhou, Y. Yuan and C. Shi, "Object tracking using SIFT features and mean shift," Comput. Vision and Image Understanding, vol. 113, no. 3, **(2009)**, pp. 345–352.

[6] S. Avidan, "Ensemble tracking," IEEE Trans. on Pattern Analy. and Mach. Intelli.., vol. 29, no. 2, **(2007)**, pp. 261–271.

[7] S. Avidan, "Support vector tracking," IEEE Trans. on Pattern Analy. and Mach. Intelli., vol. 26, no. 8, **(2004)**, pp. 1064–1072.

[8] A. A. Argyros and M. I. A. Lourakis, "Real-time tracking of multiple skin-colored object with a possibly moving camera," European Conf. Comput. Vision, **(2004)**, pp. 368–379.

[9] D. Beymer and K. Konolige, "Real-time tracking of multiple people using continuous detection," Int. Conf. Comput. Vision, **(1999)**.

[10] Y. Boykov, O. Veksler and R. Zabih, "Fast approximate energy minimization via graph cuts," IEEE Trans. on Pattern Analy. and Mach. Intelli., vol. 23, no. 11, **(2001)**, pp. 1222–1239.

[11] A. Bugeau and P. Perez, "Track and cut: Simultaneous tracking and segmentation of multiple objects with graph cuts," EURASIP J. on Image and Video Proc., **(2008)**, pp. 317278.

[12] H. Grabner, C. Leistner and H. Bischof, "Semi-supervised on-line boosting for robust tracking," European Conf. Compu. Vision, Marseille, France, Springer, Berlin, Heidelberg, vol. 1, **(2008)**, pp. 234–247.

[13] Q. Yu, T. B. Dinh and G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," European Conf. Comput. Vision, Marseille, France, Springer, Berlin, Heidelberg, vol. 2, **(2008)**, pp. 678–691.

[14] B. Babenko, M. H. Yang and S. Belongie, "Visual tracking with online multiple instance learning," IEEE Conf. Compu. Vision and Pattern Recog., Miami, FL, **(2009)**, pp. 983–990.

[15] Z. Kalal, J. Matas and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," IEEE Conf. Compu. Vision and Pattern Recog., San Francisco, CA, **(2010)**, pp. 49–56.

[16] A. Mansouri, "Region tracking via level set pdes without motion computation," IEEE Trans. on Pattern Analy. and Mach. Intelli., vol. 24, no. 7, **(2002)**, pp. 947–961.

[17] D. P. Mukherjee, N. Ray and S. T. Acton, "Level set analysis for leukocyte detection and tracking," IEEE Trans. on Image Proc., vol. 13, no. 4, **(2004)**, pp. 562–572.

[18] I. Gordon and D. G. Lowe, "Scene modelling, recognition and tracking with invariant image features," Third IEEE and ACM International Symposium on Mixed and Augmented Reality, IEEE computer Society, Washington, DC, **(2004)**, pp. 110-119.

[19] A. Jana, "Kinect for Windows SDK Programming Guide", Packt Publishing, **(2012)**.

[20] K. Sharma and I. Moon, "Improved scale-invariant feature transform feature-matching technique-based object tracking in video sequences via a neural network and kinect sensor," J. Electron. Imaging, vol. 22, no. 3, **(2013)**, pp. 033017-033017.

Author

Kajal Sharma received a B.E. degree in computer engineering from University of Rajasthan, India in 2005, and M.Tech. and Ph.D. degrees in computer science from Banasthali University, Rajasthan, India in 2007 and 2010. From October 2010 to September 2011, she worked as a postdoctoral researcher at Kongju National University, Korea. Since October 2011 to April 2013, she worked as a postdoctoral researcher at the School of Computer Engineering, Chosun University, Gwangju, Korea. Presently she is working as an independent researcher in Korea. Her research interests include image and video processing, neural networks, computer vision, and robotics. She has published many research papers in various national and international journals and conferences.