Journal of
Electronic Imaging

SPIEDigitalLibrary.org/jei

Improved scale-invariant feature transform feature-matching technique-based object tracking in video sequences via a neural network and Kinect sensor

Kajal Sharma
Inkyu Moon

SPIE

IS&T
imaging.org

Improved scale-invariant feature transform feature-matching technique-based object tracking in video sequences via a neural network and Kinect sensor

Kajal Sharma
Inkyu Moon
Chosun University
School of Computer Engineering
375 Seosuk-dong, Dong-gu
Gwangju 501-759, Republic of Korea
E-mail: inkyu.moon@chosun.ac.kr

Abstract. *Object tracking is considered to be a key technique in many computer vision applications, such as video surveillance, object recognition, and robotics. We propose a method that improves the performance of scale-invariant feature transform (SIFT)-based object tracking algorithm to track the object in the subsequent video frames. Recently, many feature-based tracking methods have been proposed. An efficient and improved SIFT feature matching-based tracking method via neural network is provided and compares the outcome of this method with other tracking method outcomes. The tracked object is assigned a distance with the Kinect sensor to determine the depth of the detected object. The experimental results show that the proposed method can track the target object under different situations such as rotation, scaling, and many others with less computation time. Self-organizing map-based improved object tracking method can also estimate the distance between the tracked object and the image sensor. The proposed tracking technique will be useful for the development of many computer vision and robot navigation applications.* © *2013 SPIE and IS&T [DOI: 10.1117/1.JEI.22.3.033017]*

1 Introduction

Vision-based object tracking is an important task in the field of computer vision, robotics, and multimedia technologies, particularly in applications such as teleconferencing, surveillance, and human–computer interfaces. The objective of object tracking is to identify the position of the object in real-time image sequences and videos.[1] The detection and tracking of moving objects result in the continuous extraction of information through a sequence of images in many computer vision, image processing, and object detection applications.[2–5] For robot navigation, there is a need to identify different obstacles in a path to achieve collision-free navigation that can be used in various applications such as automated surveillance, traffic monitoring, vehicle navigation, and human–computer interaction. A variety of algorithms has been developed in the field of computer vision for object tracking such as global template-based tracking[6–8] and local feature-based tracking methods.[9,10] Many of the tracking algorithms employ manually defined models or models that are trained in the initial stage of tracking.[11,12] These tracking methods have the problem of complex target transformations or a change in appearance. Most of the recent approaches have shown that separating the object from the background overcomes the difficulties in change in appearance model.[7,13,14] It has been shown that an object model trained via a discriminative classifier provides a significant improvement in the detection of the object.

Grabner and Bischof[15] proposed an algorithm to select features for tracking with online boosting. These algorithms use positive and negative samples when updating the classifier. With the change in appearance, this algorithm suffers from the tracking drift problem. An online semisupervised boosting method is proposed by Grabner et al.[16] to handle the drift problem, where labeled examples come from the first frame only, and subsequent training examples are left unlabeled. Yu et al.[17] proposed a gradient-based feature selection approach with online boosting with the objective of both feature selection and weak classifier updating. Babenko et al.[18] presented multiple instance learning into online tracking where samples are considered within positive and negative bags or sets. Kalal et al.[19] proposed a novel paradigm with semisupervised learning where positive and negative samples are selected via an online classifier with structural constraints.

A wide range of feature-based algorithms [e.g., Harris-Affine,[20] scale-invariant feature transform (SIFT),[21] SURF,[22] BRIEF,[23] self-organizing map (SOM)-based matching[24,25]] has been proposed for keypoint detection to obtain the features of the objects or scene. Harris and Stephens[26] proposed a corner detector to identify interest points that is robust to changes in rotation and intensity but is very sensitive to changes in scale. Schmid and Mohr[27] used Harris corners to show invariant local feature matching although they used a rotationally invariant descriptor of the local image regions and allowed features to be matched under arbitrary orientation change between the two images. Mikolajczyk and Schmid[20] proposed the Harris–Laplace detector for the detection of features by the Laplacian-of-Gaussian and provided rotation and scale-invariant features. Lowe[21]

Paper 13012 received Jan. 9, 2013; revised manuscript received Jun. 18, 2013; accepted for publication Jul. 3, 2013; published online Aug. 19, 2013.

0091-3286/2013/$25.00 © 2013 SPIE and IS&T

proposed the SIFT method by a difference-of-Gaussian that provides features invariant to image scaling and rotation, and partially invariant to changes in illumination and view. Ke and Sukthankar[28] examined the local image descriptor used by SIFT and developed principal component analysis (PCA)-based local descriptors that are more distinctive and more robust to image deformations compared with Lowe's SIFT.[21] However, such descriptors can be costly, as matching is accomplished with nearest-neighbor search, which tends to be computationally expensive.

Recently, feature-based approaches have been used for feature matching and three-dimensional (3-D) tracking of objects for the efficient detection of objects.[29,30] Gordon and Lowe[31] presented a system for augmented reality to perform object recognition with the use of highly distinctive SIFT features to provide robust tracking under scene changes. Lepetit et al.[9] proposed an efficient feature-based tracking approach and presented a lot of appearance changes for the target object. However, this approach is computationally expensive in the offline-training phase. In recent research,[24] we presented an improved feature matching method to match features between images with low computation time and compared with the Lowe's SIFT.[21] This article is focused on the tracking of stationary object in the videos using improved feature matching method. In the previous work, we focused on the feature matching between different images under variant conditions, whereas in this work, we worked on the videos for efficient tracking of object with less computation time. The proposed method is focused on the object tracking in videos with the use of improved feature matching with fast computation and to track the object under varying conditions as compared with the recent tracking algorithms. The object features are obtained for the region of interest to be tracked in the video sequence and the proposed algorithm is based on matching the features between the objects in different video frames, and it has the mechanism to reinitialize the object if it is lost and appears again. The feature matching is done with the winner calculation method on the reduced feature set, and corresponding features in different video frames are matched resulting in the detection of object in the video frames.

Traditionally, external sensors such as laser rangefinders and distance or proximity sensors were used to gather information about the surrounding environment required by autonomous navigation applications. Of late, vision sensors such as cameras, stereo-vision cameras, and infrared cameras[32] are being widely applied because of their ever-growing ability to gather information in comparison with previously used sensors. Vision sensors have various advantages over other range-based sensors such as the capability of detecting obstacles in the navigation path and the capability of enabling navigation in terrain environments. Therefore, vision sensors are becoming popular in the development of vision-based applications. Recently, Kinect sensors are also gaining popularity and importance due to their advantages over existing sensors in the field of vision science.[33] Kinect sensors have the ability to provide color and depth for an image and can recognize human skeletal joints along with gesture commands.[34–36] However, in spite of the above-mentioned advantages, they do not provide feature details associated with the objects present in the path of robot navigation. Thus, we present an efficient neural network–based algorithm to determine the object feature details to track objects in a video sequence, and we can handle the appearance variations with the proposed method.

1.1 *Contribution*

In this article, we propose a novel object tracking method based on the SOM feature matching in video sequences. We capture different sets of video sequences with the Kinect sensor device and the tracking of object is accomplished by SOM feature matching. The major research contribution in this article is proposing an efficient object tracking method in which the object tracking is done using winner calculation between the selected region of interest and the subsequent frame. To overcome the computational time problems of the object tracking algorithm, we introduced a novel method to speed up the tracking of the selected object in the videos. The scale-invariant feature descriptors generated from the selected object are supplied as an input to the SOM network and dimension of features is reduced by calculating the winner pixels to save the computational time of the object tracking.

Our novelty is to develop an efficient object tracking algorithm to track the selected object in the video sequences with less computational calculation. The tracking of object is done by matching the features between the reference region of interest and the subsequent video frames. The SOM network is used for the feature dimension reduction and to extract the meaningful information resulting into winning pixels. The reduced features are then used to track the target object in the different frames of the video sequence. The proposed object tracking method is invariant to image scale and rotation, robust to change in illumination or in 3-D viewpoint. The proposed method can generate stable keypoints and can efficiently track the object in long video sequences with less computational time because it stores the stable features during the object tracking. In this article, we present the results of different experiments on four video sequences captured with the Kinect sensor and compared the results with different tracking algorithms to examine the performance of the proposed method. Distance information is then assigned to the tracked object with the Kinect sensor with the use of depth information, thus the proposed method can efficiently track and estimate the depth of the object.

1.2 *Paper Organization*

The organization of this paper is as follows. Section 2 introduces a brief overview of the keypoint-based target detection. Section 3 explains the proposed method for object tracking with a neural network-based feature matching technique. Section 4 depicts experimental results and compares the proposed neural network-based tracking algorithm with the mean-shift tracking algorithm, the SIFT-based tracking algorithm, and the multiple instance learning algorithm. Section 4 also details the effectiveness of the proposed method. Finally, we conclude our research in Sec. 5.

2 Keypoint-Based Target Detection

SIFT is a feature descriptor proposed by Lowe[21] for extracting distinctive invariant features from images that can be invariant to image scale and rotation.[21] The SIFT method consists of four major stages: scale-space extrema detection, keypoint localization, orientation assignment, and keypoint descriptor.[21] SIFT keypoints are defined as points of local gray-level maxima and minima that are obtained from the difference-of-Gaussian images in scale space. The SIFT detector extracts a collection of keypoints from an image. The first stage is implemented efficiently using a difference-of-Gaussian in scale-space to identify potential interest points that are invariant to scale and orientation. Notation to determine the keypoints is given below:

$$\begin{aligned} D(x,y,\sigma) &= [G(x,y,k\sigma) - G(x,y,\sigma)] * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma), \end{aligned} \tag{1}$$

where $L(x,y,\sigma)$ denotes the scale-space of an image resulting from the convolution of a variable-scale Gaussian, $G(x,y,\sigma)$, with an input image, $I(x,y)$. $D(x,y,\sigma)$ is the difference of Gaussians and k denotes the constant multiplicative factor. The value of $G(x,y,\sigma)$ in Eq. (1) is given by the following equation:

$$G(x,y,\sigma) = \frac{1}{2\prod\sigma^2} e^{-(x^2+y^2)/2\sigma^2}. \tag{2}$$

In the second stage of keypoint localization, the stable keypoints are selected and the low-contrast keypoints are rejected. An orientation histogram was formed based on local image gradient directions within a region around the keypoint to assign an orientation to each keypoint location. A 4×4 array of histograms with eight orientation bins in each is obtained, thus the size of the descriptor of SIFT is $4\times4\times8 = 128$ dimensions. It is found that an image size of 640×480 contains from 200 to 1500 SIFT descriptors and keypoints in the video frames of the video sequence. The 128-dimensional SIFT descriptor value is stored with the double data type, and thus each image requires 500 kB ($128\times8\times500/1024$) of memory. In order to reduce the descriptor size, we introduced a dimensional reduction method with the use of a neural network to save memory space and to enhance computation. Figure 1 shows an overview of our proposed method in our recent research.[24] The feature similarity between the two keypoints can be measured by the winner calculation method with the neural network, which is used to track objects and will be discussed in the next section. The keypoint matching proposed by Lowe[21] in a high-dimensional feature space is time consuming, thus the dimension of the keypoints is reduced with the winner selection method in the different video frames.

Now, we give an overview of our proposed method to track the target object in video sequences with keypoint matching between different frames of the video sequences. In this article, we have used the matched keypoints in different frames for tracking the target objects. A patch of the proposed tracking method results is presented in Fig. 2, showing the tracking of objects in different video frames under varying conditions such as rotated, scaled, blurred, and noisy conditions. Section 3 presents a detailed description of our proposed tracking method with improved feature matching between the video frames of the video sequences.

3 Proposed Method for Target Tracking with Neural Network-Based Keypoint Matching

In this section, we explain the proposed method for tracking objects in video sequences. The tracking of object in the video sequences is accomplished by the proposed tracking technique by estimation of the object features. To better understand the proposed tracking approach, we will first explain the SOM-based competitive learning method to reduce the dimension of the target object feature vectors and then discuss the tracking method to track the object with the improved feature matching process.

The SOM is an artificial neural network used for the visualization and abstraction of complex data that provides a projection of multidimensional data onto a two-dimensional (2-D) map while preserving the topology of the input data space. A SOM consists of units called neurons that are organized on a regular grid, usually a 2-D rectangular, circular, or hexagonal topological grid. The SOM is applicable to a variety of image processing fields such as data mining, classification, and feature reduction in terms of nonlinear projection of multivariate data into lower dimensions.[37] The SOM is an unsupervised neural network that typically has two layers of nodes, the input layer and the output layer (see Fig. 3). The input layer consists of a set of nodes or neurons, and the output layer consists of output grid map units connected via weights with n input feature vectors obtained with the SIFT scale-space method.[21] The neurons in the SOM network are represented by a weight or prototype vector which consists of the number of components similar to the number of input variables, i.e., the dimensions of the input space. In order to preserve the topology of the input data, the input data are mapped on the grid so that close points in the input space are mapped on close points in the output space according to the defined neighborhood relationship.

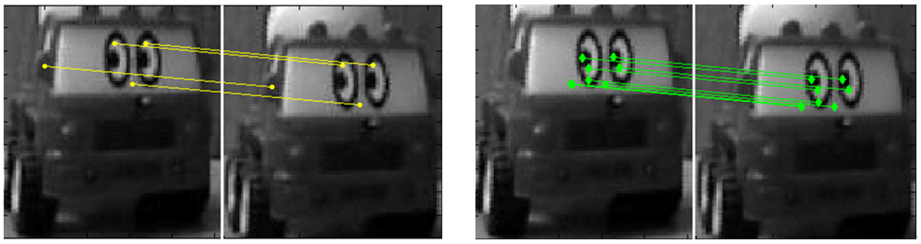
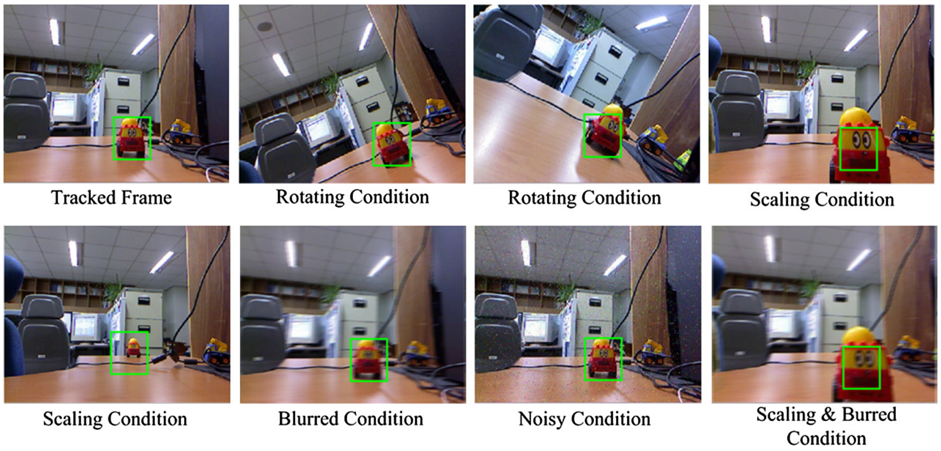

(a)
(b)

Fig. 1 Keypoint matching: the two sets of images in (a) and (b) show the keypoint matching (a) results obtained with scale-invariant feature transform (SIFT)-based keypoint matching, and in (b) the results obtained with the self-organizing map (SOM)-based improved feature matching technique (Ref. 24). The line in the two images indicates the matched points in the different frames of the video sequence.



Tracked Frame

Rotating Condition

Rotating Condition

Scaling Condition

Scaling Condition

Blurred Condition

Noisy Condition

Scaling & Burred Condition

Fig. 2 Detection of a car object with the proposed tracking method under varying conditions in a video sequence: as shown by the green line, the toy car is detected under different poses such as rotated, scaled, blurred, and noisy conditions.

The feature vector is supplied as an input to the SOM network to reduce the dimension of the features. The training algorithm was used in order to train the SOM network.[37] Competitive learning procedure is used for the learning process in the SOM network and the weights are modified that have random values during the initialization. During training, the Euclidean distance between each vector and all of the weight vectors for a predetermined number of cycles is calculated and compared. After the Euclidean distance is determined, the neuron whose weight vector has the best match and minimum distance is chosen and is called the best matching unit (BMU). For each training step, the new weight vectors are obtained using the weighted averages of the input feature vectors. The values are then updated, respectively. The similar winning pixels in the different video frames are found in order to track the presence of the object in the video sequence.

The input vector v^i is defined as $v^i = \{v_1^i, v_2^i, \ldots, v_n^i\}^T$ and the weight vector w^i at the unit i is given as $w^i = \{w_1^i, w_2^i, \ldots, w_n^i\}^T$. In order to minimize the norm, the best match is selected by taking the Euclidean distance $\|v^i - w^i\|$ as a norm. The BMU $c(v^i)$ can be defined as $c(v^i) = \arg\ \min\{\|v^i - w^i\|\}$. When $c(v^i)$ is determined, the weight vector is updated using the following equation:

$$w^i(t+1) = w^i(t) + h_{ci}(t)\{v^i(t) - w^i(t)\}, \tag{3}$$

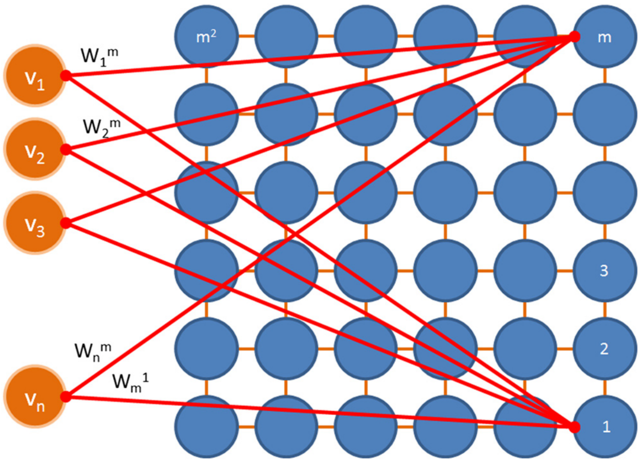


Fig. 3 The basic structure of the SOM feature map. (In this schematic image n inputs and $m \times m$ output units arranged in a grid topology. It consists of input vector $v^i = \{v_1^i, v_2^i, \ldots, v_n^i\}^T$ with weight vector $w^i = \{w_1^i, w_2^i, \ldots, w_n^i\}^T$ with output of $m \times m$ grid size).

where t is the discretized time; $t = 0, 1, 2, \ldots$ and the neighborhood function $h_{ci}(t)$ is defined as

$$h_{ci}(t) = \alpha_s \exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right), \tag{4}$$

where r_i and r_c denote the position vectors of the unit i and the BMU, respectively. α_s is the coefficient which is defined as a monotonically decreasing constant within the range $0 < \alpha_s < 1$. Function $\sigma(t)$ is defined as $\sigma(t) = \sigma(t-1) - \frac{R}{\mathrm{TS}}$; here, $\sigma(0) = R$ and $\sigma(\mathrm{TS}) = 0$. TS and R

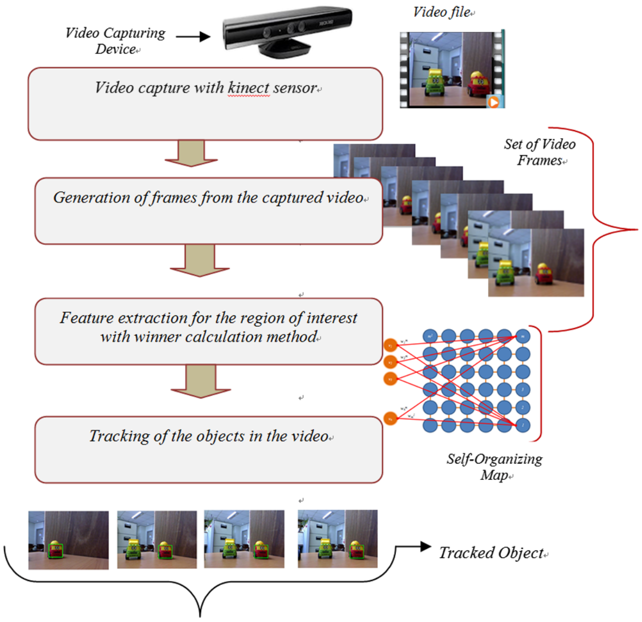

Video Capturing Device
Video file
Video capture with kinect sensor
Set of Video Frames
Generation of frames from the captured video
Feature extraction for the region of interest with winner calculation method
Tracking of the objects in the video
Self-Organizing Map
Tracked Object

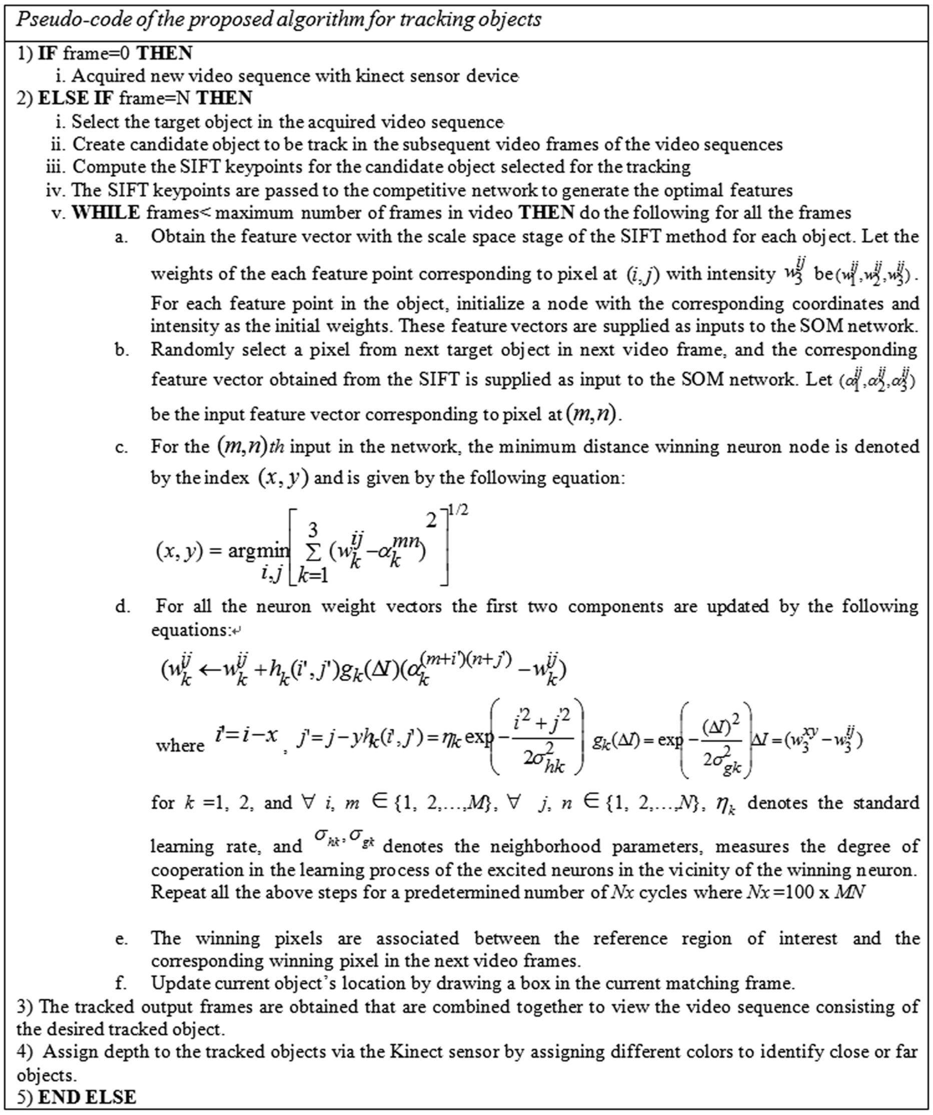
Fig. 4 Stepwise procedure of the proposed method for object tracking. Video captured with Kinect sensor and the object features is passed to feature-reduction and feature-matching stages. The output stage resulted in an effectively tracked object with the proposed method.

are the parameters that denote the number of training steps and the initial radius, respectively.

We now define the tracking procedure in order to determine the tracked object with the use of matched features. The stepwise procedure of the proposed method for object tracking is depicted in Fig. 4. The complete pseudocode of the proposed algorithm for object tracking is given in Fig. 5. The captured video sequence is fragmented into video frames. To track the region of interest for the target object in the video sequence, the features were obtained from the SIFT method. It is necessary to extract keypoints in the region of interest with the SIFT method. The above mentioned keypoint features are used for the tracking process and as an input to the SOM network. It has been found that the SOM is modified to deal with the matching process between the reference region of interest and the subsequent frames in the video sequence. SI and SJ represent the set of pixels in the reference region of interest and the set of pixels in the next video frame, respectively. $S_x = \{x_x, y_x\}$ is the set of position vectors of the pixels. The proposed algorithm performs tracking of object between the reference target object and the object in the subsequent frame (Fig. 5). The proposed algorithm also tracks similar winning pixels, which are determined in subsequent video frames.

The modified SOM algorithm can provide optimized tracked object using the neural network-based method. Lowe[21] defined a threshold value between the Euclidean distances to the nearest and the second-nearest neighbors in order to determine the correct and false matches. The defined constrained condition makes sure that the object does not contain repeating patterns and one suitable match is expected. Also because of the defined constraint, the Euclidean distance to the nearest neighbor is significantly smaller than the Euclidean distance to the second-nearest neighbor. The correct or false matches can be obtained from the positive and negative matches based on the distance to the nearest and the second nearest neighbor. Lowe[21] claims that a match is selected as positive if the distance to the nearest neighbor is 0.8 times larger than that to the second nearest one.

Our proposed algorithm provides an improvement over the SIFT algorithm in terms of correct matches and



```
Pseudo-code of the proposed algorithm for tracking objects
1) IF frame=0 THEN
      i. Acquired new video sequence with kinect sensor device
2) ELSE IF frame=N THEN
      i. Select the target object in the acquired video sequence
     ii. Create candidate object to be track in the subsequent video frames of the video sequences
    iii. Compute the SIFT keypoints for the candidate object selected for the tracking
     iv. The SIFT keypoints are passed to the competitive network to generate the optimal features
      v. WHILE frames< maximum number of frames in video THEN do the following for all the frames
            a. Obtain the feature vector with the scale space stage of the SIFT method for each object. Let the
               weights of the 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 object, initialize a node with the corresponding coordinates and
               intensity as the initial weights. These feature vectors are supplied as inputs to the SOM network.
            b. Randomly select a pixel from next target object in next video frame, and the corresponding
               feature vector obtained from the SIFT is supplied as input to the SOM network. Let $(\alpha_1^{ij}, \alpha_2^{ij}, \alpha_3^{ij})$
               be the input feature vector corresponding to pixel at $(m,n)$.
            c. 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:
               $(x, y) = \operatorname*{argmin}_{i,j} \left[ \sum_{k=1}^{3} (w_k^{ij} - \alpha_k^{mn})^2 \right]^{1/2}$
            d. 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 $i' = i - x$, $j' = j - y h_k(i', j') = \eta_k \exp\left( -\frac{i'^2 + j'^2}{2\sigma_{hk}^2} \right)$ $g_k(\Delta I) = \exp\left( -\frac{(\Delta I)^2}{2\sigma_{gk}^2} \right)$ $\Delta I = (w_3^{xy} - w_3^{ij})$
               for $k$ =1, 2, and $\forall$ $i$, $m$ $\in \{1, 2, \ldots, M\}$, $\forall$ $j$, $n$ $\in \{1, 2, \ldots, N\}$, $\eta_k$ denotes the standard
               learning rate, and $\sigma_{hk}, \sigma_{gk}$ denotes the neighborhood parameters, measures the degree of
               cooperation in the learning process of the excited neurons in the vicinity of the winning neuron.
               Repeat all the above steps for a predetermined number of $Nx$ cycles where $Nx$ =100 x $MN$
            e. The winning pixels are associated between the reference region of interest and the
               corresponding winning pixel in the next video frames.
            f. Update current object's location by drawing a box in the current matching frame.
3) The tracked output frames are obtained that are combined together to view the video sequence consisting of
   the desired tracked object.
4) Assign depth to the tracked objects via the Kinect sensor by assigning different colors to identify close or far
   objects.
5) END ELSE
```

Fig. 5 Pseudocode of the proposed algorithm for object tracking. The objects in the video sequence are tracked with feature matching between different video frames.

computation time. In order to determine the correct matches, the differences between the corresponding matched keypoints are computed for the reference target object and the subsequent frames in the video sequence. The matched pair is selected as the stable keypoint if the computed difference is less than or equal to the threshold value. In accordance with the nearest neighborhood procedure in the SOM network, for each feature in the reference target object, the corresponding feature is determined using the matching method. Based on the minimum distance, the winner neuron is selected and the matched winning feature set is selected as a set of location-matched keypoints. To search for an appropriate similarity region, launch the distance estimation method between the detected locations and determine the location-matched keypoints by estimating the minimum distance, respectively. The location-matched keypoints are then obtained from the following equation:

$$M(U,V) = \sqrt{(x_U - x_V)^2 + (y_U - y_V)^2}. \tag{5}$$

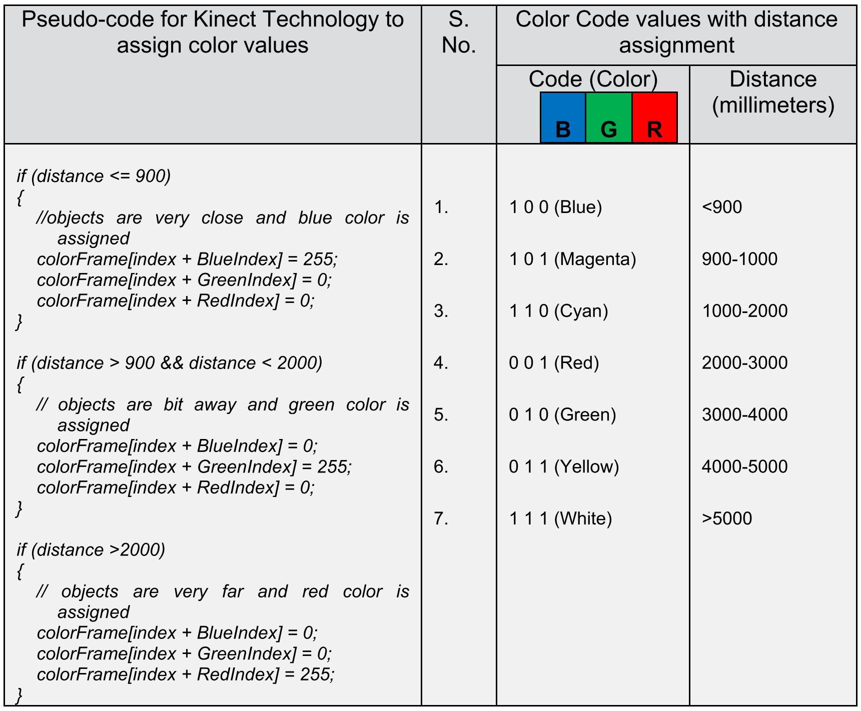
If the $M(U,V)$ is less than the Euclidean distance, it is considered to be the tracked point and is accepted. Otherwise the matched keypoints is rejected. (x_U, y_U) is the location of the reference region of interest, and (x_V, y_V) is the location of the subsequent video frames in the video sequence. Thus, only the stable keypoints are obtained with the proposed object tracking method (Fig. 5).

3.1 *Distance Assignment to the Tracked Object*

We assign a distance to the tracked objects with the Kinect technology after obtaining the tracked object with the proposed object tracking method. The Kinect depth sensing system is composed of an infrared (IR) emitter projecting structured light, which is captured by the CMOS image sensor and decoded to produce the depth image of the scene. Its range is specified between 0.7 and 6 m although the best results are obtained from 1.2 to 3.5 m. The color image resolution is 640 × 480 pixels, and depth image resolution is 320 × 240 pixels with a rate of 30 Hz. The field of view is 57 deg horizontal, 43 deg vertical and has a tilt range of ±27 deg.[34–36]

Each pixel in the depth image is labeled according to the distance and the tracked object is assigned a distance with the depth image technology. The depth information is obtained by initializing the depth stream. The depth stream data are converted to the byte array format, which is an array in row order, and the depth stream data are packed in the 32-bit RGBA pixel format. Each pixel in the depth image represents the distance in millimeters and stores the data values using the RGB data format. The depth data are obtained and converted into a distance to display the color map of the depth image. If an object is closer than 850 mm or farther than 4000 mm, the result will be 0. We used the Kinect SDK to extract the depth map and performed coding to identify the near and far objects based on the color map. We use the BGR format to assign the color to different distances (Table 1). For example, the areas in red indicate objects that are placed at a distance between 2 and 3 m. Similarly, various distances are assigned based on the colors. This technology worked well for the objects at distance in the range of 1.2 to 3.5 m. The Kinect for Windows SDK defines a constant on the DepthImageFrame class, which specifies the number of bits to shift right to obtain the depth values.

Table 1 Table for pseudocode for Kinect technology to assign color values to the tracked objects and assign distance based on the color code.



Pseudo-code for Kinect Technology to assign color values	S. No.	Color Code values with distance assignment	
		Code (Color) B G R	Distance (millimeters)

if (distance <= 900)
{
//objects are very close and blue color is assigned
colorFrame[index + BlueIndex] = 255;
colorFrame[index + GreenIndex] = 0;
colorFrame[index + RedIndex] = 0;
}
if (distance > 900 && distance < 2000)
{
// objects are bit away and green color is assigned
colorFrame[index + BlueIndex] = 0;
colorFrame[index + GreenIndex] = 255;
colorFrame[index + RedIndex] = 0;
}
if (distance >2000)
{
// objects are very far and red color is assigned
colorFrame[index + BlueIndex] = 0;
colorFrame[index + GreenIndex] = 0;
colorFrame[index + RedIndex] = 255;
}
	1.	1 0 0 (Blue)	<900

	2.	1 0 1 (Magenta)	900-1000
	3.	1 1 0 (Cyan)	1000-2000
	4.	0 0 1 (Red)	2000-3000
	5.	0 1 0 (Green)	3000-4000
	6.	0 1 1 (Yellow)	4000-5000
	7.	1 1 1 (White)	>5000

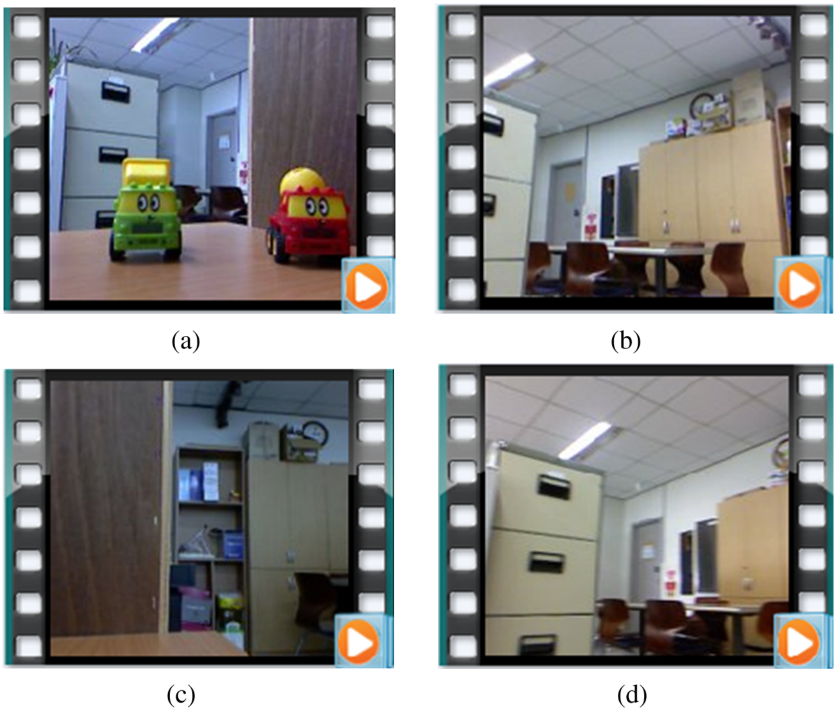


Fig. 6 Test sequences used in experiments. (a) Toy car sequence. (b) Indoor environment of the lab. (c) Box sequence. (d) Rack sequence.

The byte array holds the color pixels of the object features. The image uses four bytes: blue, green, red, and alpha. The alpha bits are used to determine the transparency of each pixel.

4 Experimental Results and Discussion

In this section, we demonstrate and discuss the experiments conducted for tracking objects, verifying the effectiveness of the proposed method by comparing it with the mean-shift tracking algorithm, the SIFT-based tracking algorithm and the multiple instance learning algorithm, and discuss the improvement of the object tracking methodology in terms of computation time. The different tracking algorithms have been implemented in MATLAB and tested on a 2.5-GHz Intel® Core™ 2 Quad CPU Q8300. We have used the Kinect sensor device designed by Microsoft for the acquisition of the video sequence. To demonstrate the tracking ability, four video sequences have been acquired and used for the experiments. The first video sequence is a sequence of three toy cars, the second video sequence is a sequence of an indoor lab environment, the third video sequence is a sequence of boxes, and the fourth video sequence is a sequence of steel rack (see Fig. 6). The resolutions of the images for the four video sequences were 640 × 480 pixels, and the video sequence was captured at a frame rate of 30 fps.

Generally, motion detection, which is initially used to determine moving objects, is necessary for automatic tracking of unidentified objects in video sequences. Motion detection algorithms, such as frame differencing, background modeling/subtracting and optical flow,[38] are robust to identify unknown moving objects which will be regarded as tracked target in the coming image sequences. On the other hand, as tracked targets with prior knowledge or predefined model, these tracked objects at first frame can be extracted easily by information such as shape, texture, and color or correlation result regardless of whether these targets are moving or stationary. For example, the motionless car shown in Fig. 7 at first frame is successfully recognized with its prior knowledge of color (yellow and red). In this

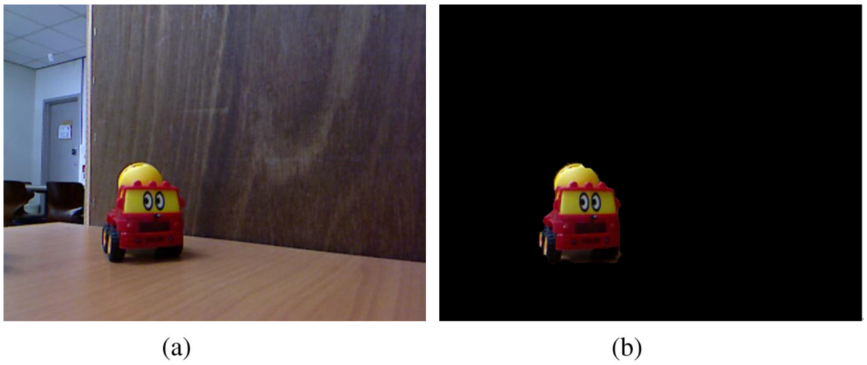


Fig. 7 Object detection with prior knowledge (color here). (a) Tracked car at first frame. (b) Identified car with color information.

article, all of the tracking algorithms are illustrated with stationary objects which means that there are no real moving objects and the image sequences are acquired by moving the Kinect camera only (motion detection is unnecessary in this case). We focus on the tracking method in terms of computational time. Consequently, any stationary objects in image sequence can be used as tracked target and we can extract specific object for tracking at first frame when the prior knowledge or predefined model about the target is given. In this experiment, some of the tracked targets at first frame are automatically determined with prior information or predefined object model while others are manually selected.

The proposed tracking algorithm has been used to track the object in the above-mentioned video sequences, and the results obtained have been compared with the mean-shift tracking algorithm,[11] the SIFT tracking algorithm, and the multiple instance learning algorithm.[18] Experimental results show the effectiveness of the proposed method in terms of computation time. The tracking results based on the mean-shift tracking algorithm, the SIFT tracking algorithm, the multiple instance learning algorithm, and the proposed algorithm for the four video sequences are shown in this section. Figure 8 shows the results obtained from tracking the red object in the toy car video sequence using the mean-shift tracking algorithm, SIFT tracking algorithm, multiple instance learning algorithm, and the proposed tracking algorithm. The tracked object (red car) at first frame is chosen from the determination result shown in Fig. 7 which is identified with prior color information. Figure 9 shows the results for tracking the chair object in an indoor lab environment sequence using the mean-shift tracking algorithm, SIFT tracking algorithm, multiple instance learning algorithm, and the proposed tracking algorithm. The chair object at first frame is initialized with correlation result by using the predefined color model and parts of them are selected for tracking. It can be observed in Fig. 9(a) that the selected target object in the mean-shift tracking algorithm has been lost by the tracking process. As the mean-shift tracking algorithm is based on comparing the histogram, this algorithm cannot track the object if it is lost and reappears after some time. The SIFT tracking algorithm[21] has the capability to generate local features robust to changes in image scale, noise, illumination, and local geometric distortion. However, the SIFT algorithm has high computational complexity. The multiple instance learning algorithms can track the target object but require high computational time. Thus, the proposed tracking method overcomes the difficulties of the above-mentioned algorithms.

Here, tracking objects at first frame in Figs. 10 and 11 are manually selected and are automatically tracked in the coming image sequences. Definitely, if the models of tracked objects are given, they can be automatically identified at first frame. Figure 10 shows the results for tracking the box object in the box sequence using the mean-shift tracking algorithm, SIFT tracking algorithm, multiple instance learning algorithm, and the proposed tracking algorithm. Figure 11 shows the results for tracking the rack object in the rack sequence using the mean-shift tracking algorithm, SIFT tracking algorithm, multiple instance learning algorithm, and the proposed tracking algorithm.

The proposed tracking approach is invariant to object loss, scaling, rotation, noise, and blurring (Fig. 2). The proposed algorithm is robust to these changes and can efficiently perform tracking under various situations. In this article, we have focused on single-object tracking and also on reducing

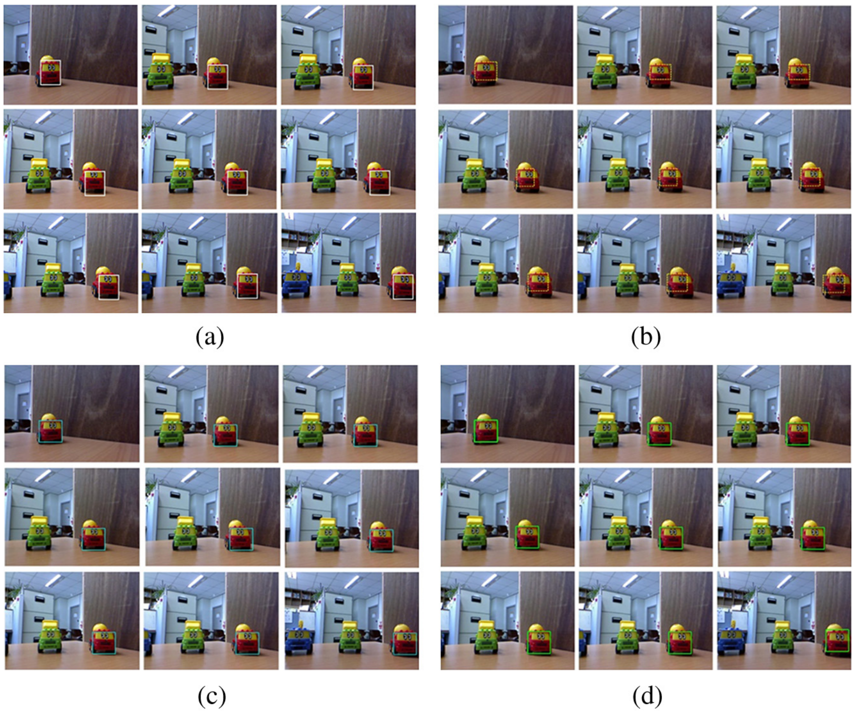


Fig. 8 (a) Mean-shift algorithm-based object tracking with toy car sequence: tracking the red object for frame nos. 1, 14, 22, 30, 40, 44, 51, 66, and 102. (b) SIFT-based object tracking with toy car sequence: tracking the red object for frame nos. 1, 14, 22, 30, 40, 44, 51, 66, and 102. (c) Multiple instance learning-based object tracking with toy car sequence: tracking the red object for frame nos. 1, 14, 22, 30, 40, 44, 51, 66, and 102. (d) Proposed method-based object tracking with toy car sequence: tracking the red object for frame nos. 1, 14, 22, 30, 40, 44, 51, 66, and 102.

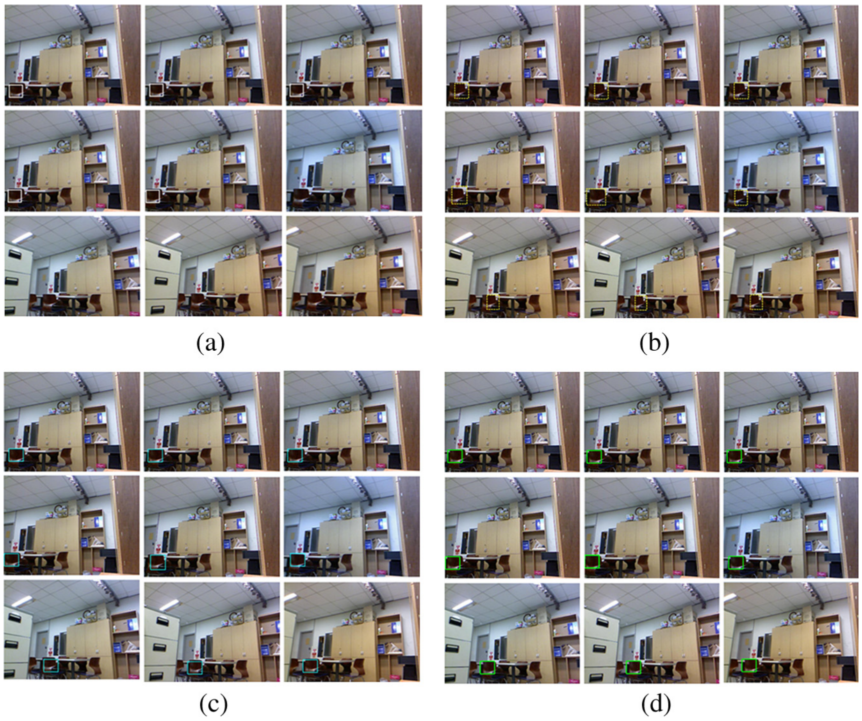


Fig. 9 (a) Mean-shift algorithm-based object tracking with indoor environment of the lab sequence: tracking the chair object for frame nos. 1, 18, 30, 44, 56, 102, 125, 134, and 147. (b) SIFT-based object tracking with indoor environment of the lab sequence: tracking the chair object for frame nos. 1, 18, 30, 44, 56, 102, 125, 134, and 147. (c) Multiple instance learning-based object tracking with indoor environment of the lab sequence: tracking the chair object for frame nos. 1, 18, 30, 44, 56, 102, 125, 134, and 147. (d) Proposed method-based object tracking with indoor environment of the lab sequence: tracking the chair object for frame nos. 1, 18, 30, 44, 56, 102, 125, 134, and 147.

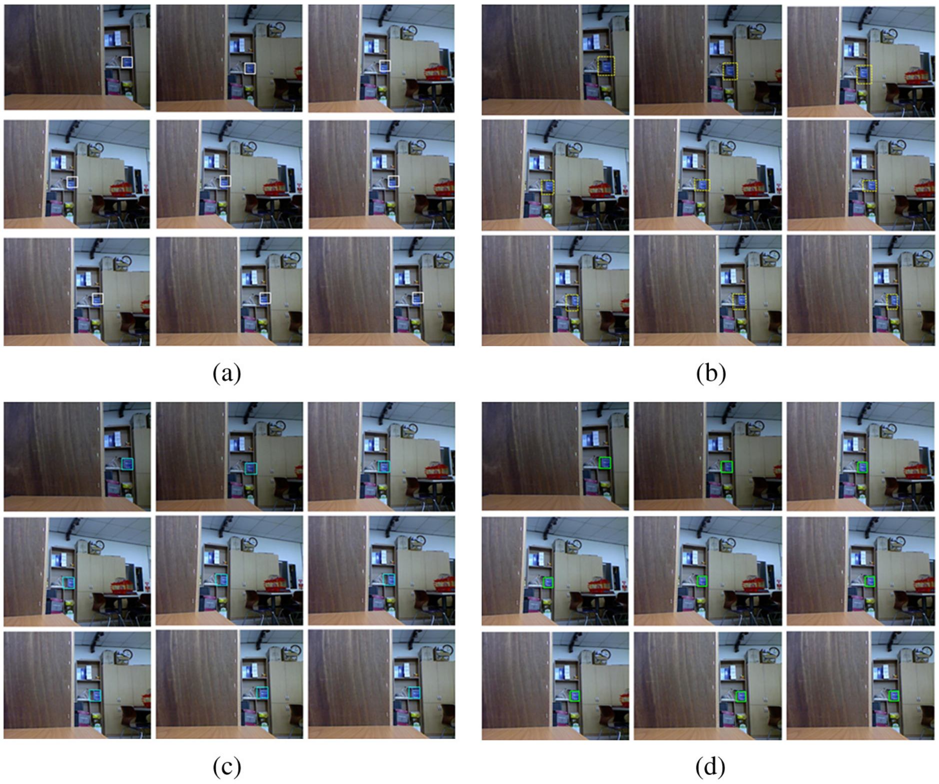


Fig. 10 (a) Mean-shift algorithm-based object tracking with the box sequence: tracking the box object for frame nos. 1, 123, 189, 229, 260, 296, 314, 378, and 420. (b) SIFT-based object tracking with the box sequence: tracking the box object for frame nos. 1, 123, 189, 229, 260, 296, 314, 378, and 420. (c) Multiple instance learning-based object tracking with the box sequence: tracking the box object for frame nos. 1, 123, 189, 229, 260, 296, 314, 378, and 420. (d) Proposed method-based object tracking with the box sequence: tracking the box object for frame nos. 1, 123, 189, 229, 260, 296, 314, 378, and 420.

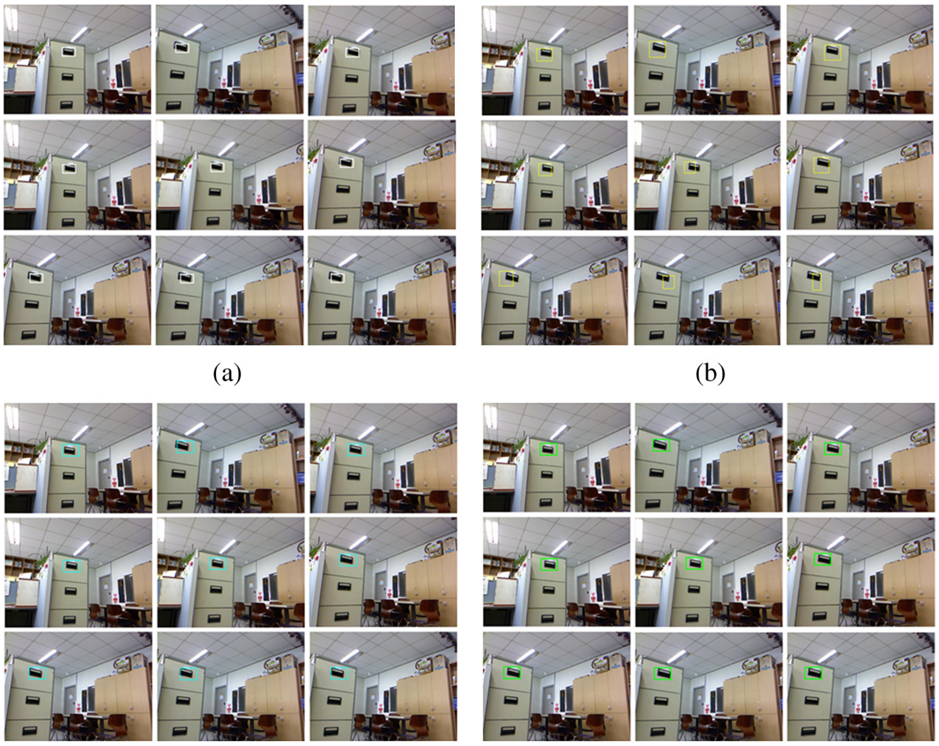


Fig. 11 (a) Mean-shift algorithm-based object tracking with the rack sequence: tracking the rack object for frame nos. 1, 104, 172, 207, 235, 269, 291, 312, and 325. (b) SIFT-based object tracking with the rack sequence: tracking the rack object for frame nos. 1, 104, 172, 207, 235, 269, 291, 312, and 325. (c) Multiple instance learning-based object tracking with the rack sequence: tracking the rack object for frame nos. 1, 104, 172, 207, 235, 269, 291, 312, and 325. (d) Proposed method-based object tracking with the rack sequence: tracking the rack object for frame nos. 1, 104, 172, 207, 235, 269, 291, 312, and 325.

the computation time with the use of the proposed tracking technique. With the use of neural network-based tracking, the method presented in this article reduces the searching time of tracking objects in the video sequence, thus greatly reducing the time complexity compared with the mean-shift tracking algorithm, the SIFT tracking algorithm, and the multiple instance learning algorithm.

We have evaluated the effectiveness of the proposed system on four video sequences, and the results clearly show the advantages of the proposed method in terms of computation time. Table 2 compares the computation time for the four sequences with the mean-shift tracking algorithm, the SIFT tracking algorithm, the multiple instance learning algorithm, and the proposed tracking algorithm. As can be clearly noticed, the results show that the proposed tracking algorithm tracks the object efficiently using less computational time. In Fig. 9(a), it should be noted that the target object is not tracked with the mean-shift algorithm in frame no. 102 because the target object is lost after frame no. 60 in the indoor lab environment video sequence. The mean-shift algorithm cannot track the object when it is lost, regardless of whether or not it reappears in the subsequent frames of the video sequence. The object can be tracked with the SIFT tracking algorithm, Fig. 9(b), but requires more time to track the object. However, the multiple instance learning algorithm can track the object but requires

Table 2 Performance comparison of computation time for the toy car sequence, indoor lab sequence, box sequence, and rack sequence with mean-shift tracking algorithm, SIFT tracking algorithm, multiple instance learning tracker, and proposed SOM-based tracking algorithm.

	Average CPU computational time (s) for the whole video sequence			
Video sequence	**Mean-shift tracking algorithm**	**SIFT-based tracking algorithm**	**Multiple instance learning tracker (Ref. 18)**	**Proposed SOM based tracking algorithm**
Toy car sequence	0.092615	0.121075	0.221472	0.024968
Indoor environment of the lab	0.098679	0.097452	0.225215	0.020568
Box sequence	0.100903	0.110437	0.229496	0.027079
Rack sequence	0.085536	0.081522	0.226072	0.022334

high computational load [Fig. 9(c)]. On the other hand, the proposed tracking method can efficiently track the object even though the object is lost once and subsequently reappears, as shown in Fig. 9(d) for frame no. 102 with less time. The proposed approach is based on neural network–based tracking of objects in the subsequent video frames and can efficiently track objects with less computation time under varying situations (Fig. 2).

In order to reduce the computational time, this article has proposed an improvement over the mean-shift tracking algorithm, the SIFT-based tracking algorithm, and the multiple instance learning algorithm with the use of Kohonen's SOM neural network methodology. The descriptor vector was computed using the SIFT method for the selected region of interest to be tracked, and the dimensions of the extracted features were down-sampled by a factor of 2. The optimum features were selected using the SOM method based on the winner calculation method. In order to determine the amount of reduced processing time, it was assumed that the number of extracted features in the lower octave with respect to the higher octave is decreased by four times due to the down-sampling by a factor of 2 in both image directions. The size of the descriptor vectors was reduced and passed to the SOM network in order to output the winner neurons in the region of interest and the subsequent video frames. Hence, the matching cost was reduced by 1.5 times as compared with the mean-shift algorithm, the SIFT-based tracking algorithm, and the multiple instance learning algorithm. The proposed algorithm was able to locate the object accurately and consistently in all the areas, whereas the mean shift algorithm produced unsatisfactory results, and the SIFT tracking algorithm and the multiple instance learning algorithm require more computational calculations. The tracking accuracy can be seen by Fig. 9 in the lab sequence which shows that the proposed tracking method consistently produced stable and satisfactory tracking results. Similarly, the proposed method tracking accuracy can be evaluated by the results of Figs. 8(d), 9(d), 10(d), and 11(d). The object is accurately tracked with the proposed SOM-based tracking method resulting into stable and satisfactory tracking results.

We have presented the comparison graph for the four video sequences, which indicates that the proposed method reduces the object tracking time compared with the mean-shift tracking algorithm, the SIFT tracking algorithm, and the multiple instance learning algorithm. The mean-shift tracking algorithm is based on a color-histogram and a window approach, without the mechanism to redefine the initial window. With the SIFT tracking algorithm and the multiple instance learning algorithm, the object tracking calculation was reduced to the selected region of interest but requires more computation to track objects under different situations. To overcome these difficulties, the proposed tracking approach efficiently tracked objects by reducing the time required for searching and the computational complexity. Figure 12 shows the performance comparison of computation time of the first 150 video frames for the toy car sequence consisting of three objects for the mean-shift tracking algorithm, the SIFT tracking algorithm, multiple instance learning algorithm, and the proposed tracking algorithm. Figure 13 shows the performance comparison of computation time for the indoor environment of the lab sequence with the mean-shift tracking algorithm, the SIFT tracking algorithm, the multiple instance learning algorithm, and the proposed tracking algorithm for the first 150 video frames. Figure 14 shows the performance comparison of computation time for the box sequence for the 420 video frames with the mean-shift tracking algorithm, the SIFT tracking algorithm, the multiple instance learning algorithm, and the proposed tracking algorithm. Figure 15 shows the performance comparison of computation time for the rack sequence for the 325 video frames with the mean-shift tracking algorithm, the SIFT tracking algorithm, the multiple instance learning algorithm, and the proposed tracking algorithm.

The average time taken by the mean shift tracking algorithm, the SIFT tracking algorithm, and the multiple instance learning algorithm was 0.0926, 0.1210, and 0.2214 s, while the average tracking time was significantly reduced to 0.0249 s when the proposed method was used. The average time of the proposed tracking algorithm was reduced because of the reduction in features with the neural network–based method. The search was conducted on the reduced set of features and provides tracking output in less computation time compared with the mean-shift tracking, SIFT tracking algorithm, and multiple instance learning algorithm. The depth is assigned to the tracked objects and the pseudocode to assign

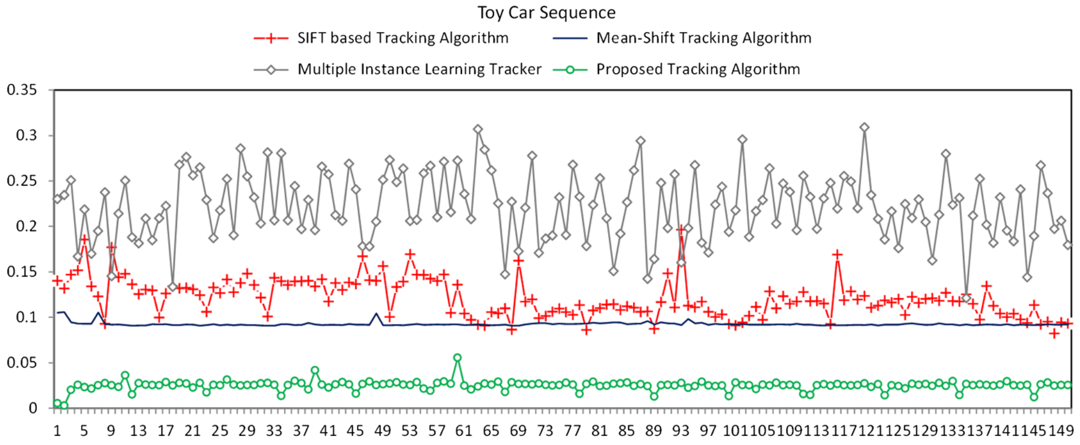


Fig. 12 Performance comparison of computation time for the toy car sequence with mean-shift tracking algorithm, SIFT tracking algorithm, multiple instance learning algorithm, and the proposed SOM-based tracking algorithm.

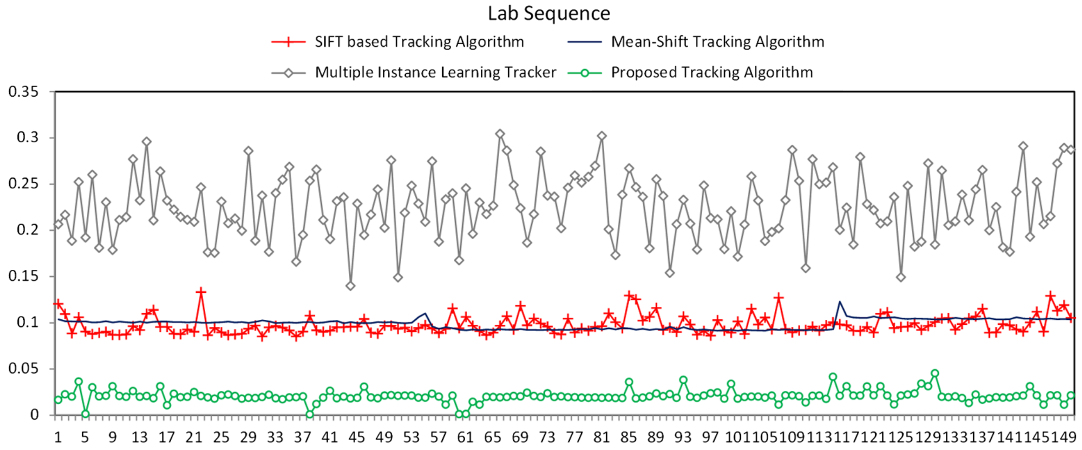


Fig. 13 Performance comparison of computation time for the indoor environment of the lab sequence with the mean-shift tracking algorithm, SIFT tracking algorithm, multiple instance learning algorithm, and the proposed SOM-based tracking algorithm.

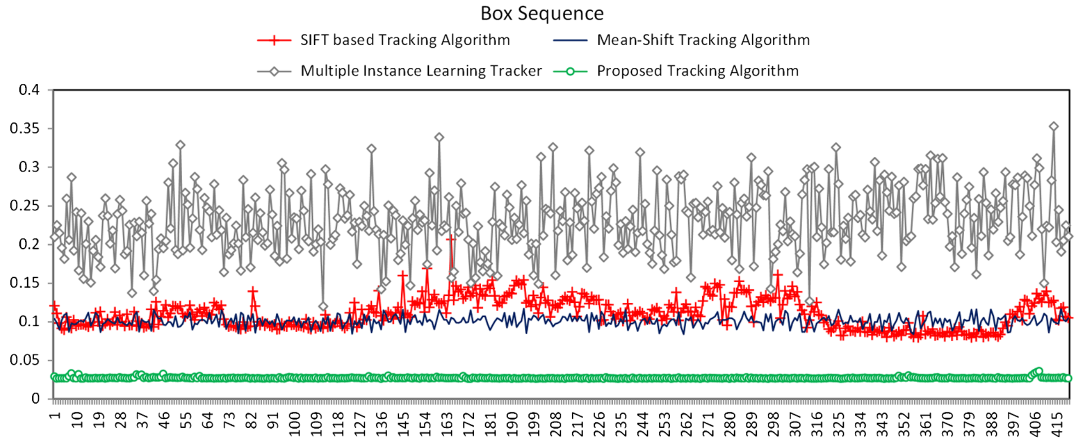


Fig. 14 Performance comparison of computation time for the box sequence with the mean-shift tracking algorithm, SIFT tracking algorithm, multiple instance learning algorithm, and the proposed SOM-based tracking algorithm.

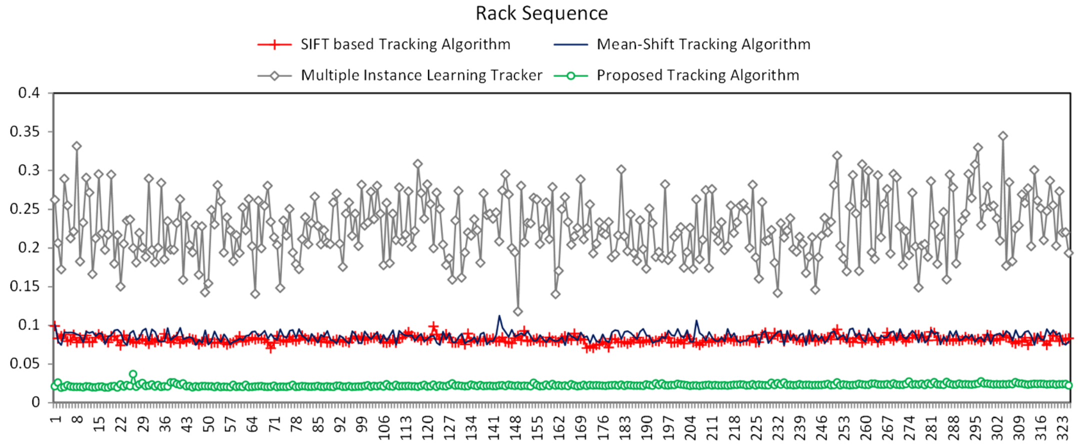


Fig. 15 Performance comparison of computation time for the rack sequence with the mean-shift tracking algorithm, SIFT tracking algorithm, multiple instance learning algorithm, and the proposed SOM-based tracking algorithm.

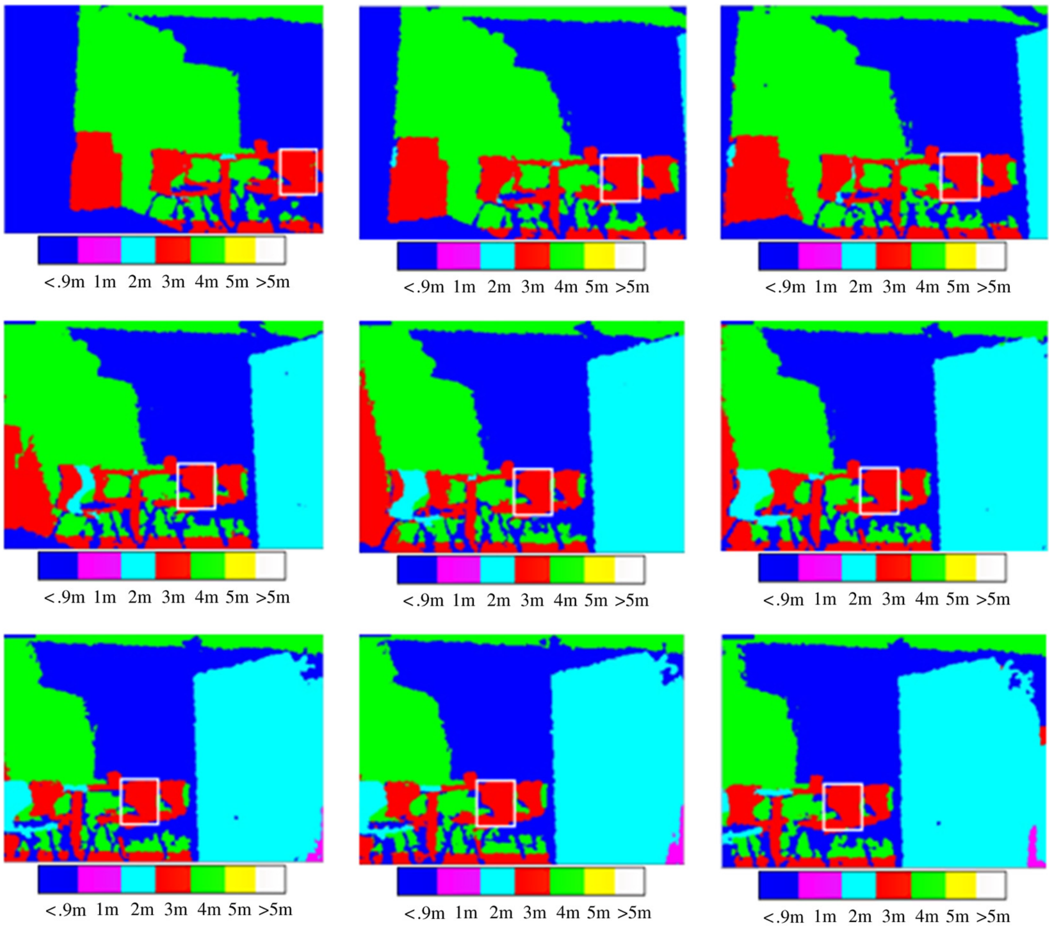


Fig. 16 Distance assignment to the tracked objects based on color information.

distance values is given in Table 1. Figure 16 shows the results for the depth assignment to the objects corresponding to the nine video frames of the video sequences. We conducted four object tracking experiments with the four different methods, and we conducted distance estimation experiments for the second video sequence. The depth estimation technique worked well for distances of >1 m. The object in the second video sequence is >1 m, and the above-mentioned method worked well to estimate the distance of the object.

5 Conclusion

In this article, we have addressed a neural network–based object tracking algorithm to detect the selected object across a video sequence in the context of developing real-time vision-based applications. In order to improve the performance of object detection and tracking based on SIFT feature matching algorithm, SOM network was applied and feature matching between the different frames was done using a winner calculation method which highly reduces computational cost. The Kinect depth technology is then employed in order to calculate the distance between the extracted target objects and the image sensor. Extensive experiments demonstrate that the proposed approach can provide efficient, reliable object tracking with less computation time. Also, the presented results prove the feasibility and usefulness of the proposed method. As future work, we will focus on detecting multiple objects using a real-time tracking methodology and will focus on the tracking of moving objects in the video sequence. The proposed method will be useful for the development of many real-time autonomous navigation and computer vision applications.

Acknowledgments

This study was supported by research fund from Chosun University, 2012.

References

1. W. Hu et al., "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst. Man Cybern.* **34**(3), 334–352 (2004).
2. A. Cilia, "Object tracking using cluster of elastically linked feature trackers," *J. Electron. Imaging* **17**(2) 023019 (2008).
3. J. Fan, X. Shen, and Y. Wu, "Scribble tracker: a matting-based approach for robust tracking," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(8), 1633–1644 (2012).
4. U. Braga-Neto, M. Choudhary, and J. Goutsias, "Automatic target detection and tracking in forward-looking infrared image sequences using morphological connected operators," *J. Electron. Imaging* **13**(4) 802–813 (2004).
5. S. F. Barrett et al., "Efficiently tracking a moving object in two-dimensional image space," *J. Electron. Imaging* **10**(3), 785–793 (2001).
6. S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(8), 1064–1072 (2004).
7. R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(10), 1631–1643 (2005).
8. O. Williams, A. Blake, and R. Cipolla, "Sparse Bayesian learning for efficient visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(8), 1292–1304 (2005).

9. V. Lepetit, P. Lagger, and P. Fua, "Randomized trees for real-time keypoint recognition," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Vol. 2, pp. 775–781, IEEE Computer Society, Washington, DC (2005).
10. J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 593–600, Cornell University, Ithaca, NY (1994).
11. D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of nonrigid objects using mean shift," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Vol. 2, pp. 142–149, IEEE Computer Society, Hilton Head Island, South Carolina (2000).
12. A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Vol. 1, pp. 798–805, IEEE Computer Society, Washington, DC (2006).
13. J. Wang, X. Chen, and W. Gao, "Online selecting discriminative tracking features using particle filter," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Vol. 2, pp. 1037–1042, IEEE Computer Society, Washington, DC (2005).
14. H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via online boosting," in *Proc. British Machine Vision Conf.*, Vol. 1, pp. 47–56, British Machine Vision Association press (2006).
15. H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Vol. 1, pp. 260–267, IEEE Computer Society, Washington, DC (2006).
16. H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. European Conf. Computer Vision*, Marseille, France, Vol. 1, pp. 234–247, Springer, Berlin, Heidelberg (2008).
17. Q. Yu, T. B. Dinh, and G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," *Proc. European Conf. Computer Vision*, Marseille, France, Vol. 2, pp. 678–691, Springer-Verlag, Berlin, Heidelberg (2008).
18. B. Babenko, M. H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 983–990, Miami, FL (2009).
19. Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: bootstrapping binary classifiers by structural constraints," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 49–56, San Francisco, CA (2010).
20. K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vision* **60**(1), 63–86 (2004).
21. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision* **60**(2), 91–110 (2004).
22. H. Bay et al., "SURF: speeded up robust features," *Comput. Vision Image Underst.* **110**(3), 346–359 (2008).
23. M. Calonder et al., "Brief: binary robust independent elementary features," in *Proc. European Conf. Computer Vision*, Heraklion, Crete, Greece, Vol. 4, pp. 778–792, Springer-Verlag, Berlin, Heidelberg, (2010).
24. K. Sharma, I. Moon, and S. Kim, "Depth estimation of features in video frames with improved feature matching technique using Kinect sensor," *Opt. Eng.* **51**(10), 107002 (2012).
25. K. Sharma, I. Moon, and S. Kim, "Extraction of visual landmarks using improved feature matching technique for stereo vision applications," *IETE Tech. Rev.* **29**(6), 473–481 (2012).
26. C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. of Alvey Vision Conf.*, pp. 147–151 (1988).
27. C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(5), 530–534 (1997).
28. Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Vol. 2, pp. 506–513, IEEE Computer Society, Washington, DC (2004).
29. S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Int. J. Rob. Res.* **21**(8), 735–758 (2002).
30. S. Hare et al., "Efficient online structured output learning for keypoint-based object tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1894–1901, Providence, RI (2012).
31. I. Gordon and D. G. Lowe, "Scene modelling, recognition and tracking with invariant image features," in *Proc. of 3rd IEEE/ACM Int. Symp. Mixed and Augmented Reality*, Arlington, Virginia, pp. 110–119, IEEE Computer Society, Washington, DC (2004).
32. R. Manduchi et al., "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous Robots* **18**(1), 81–102 (2005).
33. E. Parvizi and Q. Wu, "Real-time 3D head tracking based on time-of-flight depth sensor," in *Proc. IEEE Int. Conf. Tools with Artificial Intelligence*, Patras, pp. 517–521, IEEE Computer Society, Washington, DC (2007).
34. J. Webb and J. Ashley, *Beginning Kinect Programming with the Microsoft Kinect SDK*, pp. 49–83, Apress (2012).
35. Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE MultiMedia* **19**(2), 4–10 (2012)
36. A. Oliver et al., "Using the Kinect as a navigation sensor for mobile robotics," *Proc. 27th Conf. Image and Vision Computing*, New Zealand, pp. 509–514, ACM, New York (2012).
37. J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Networks*, **11**, 586–600 (2000).
38. G. Bradski and A. Kaehler, *Learing OpenCV*, O'Reilly Media, California (2008).

Kajal Sharma received a BE degree in computer engineering from University of Rajasthan, India, in 2005, and MTech and PhD 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, she has been working as a postdoctoral researcher at the School of Computer Engineering, Chosun University, Gwangju, 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.

Inkyu Moon received the BS and MS degrees in electronics engineering from SungKyunKwan University, Korea, in 1996 and 1998, respectively, and the MS and PhD degrees in electrical and computer engineering from University of Connecticut, Storrs, in 2007. From January 2008 to January 2009, he was a researcher in a postdoctoral position at the University of Connecticut. Since 2009, he has been an assistant professor at the School of Computer Engineering, Chosun University, Gwangju, South Korea. His research interests include digital holography, biomedical imaging, optical information processing, computational integral imaging, optical and digital encryptions, 3-D digital image processing, 3-D statistical pattern recognition, and 4-D tracking algorithms. He has over 40 publications, including over 20 peer-reviewed journal articles, over 20 conference proceedings, including over 10 keynote addresses and invited conference papers. His papers have been cited 400 times, according to the citation index of Google Scholar. He is a member of IEEE, OSA, and SPIE. He is on the editorial board of the Korea Multimedia Society.