International Journal on
Advanced Science
Engineering
Information Technology

Vol.11 (2021) No. 6
ISSN: 2088-5334

# Object Searching on Real-Time Video Using Oriented FAST and Rotated BRIEF Algorithm

Faisal Dharma Adhinata [a,b], Agus Harjoko [b], Wahyono [b,*]

[a] *Faculty of Informatics, Institut Teknologi Telkom Purwokerto, DI Panjaitan 128, Purwokerto, Indonesia*
[b] *Department of Computer Science and Electronics, Universitas Gadjah Mada, Sekip Utara Bulaksumur, Yogyakarta, Indonesia*
*Corresponding author: *wahyo@ugm.ac.id*

*Abstract*— **The pre-processing and feature extraction stages are the primary stages in object searching on video data. Processing video in all frames is inefficient. Frames that have the same information should only be once processed to the next stage. Then, the feature extraction algorithm that is often used to process video frames is SIFT and SURF. The SIFT algorithm is very accurate but slow. On the other hand, the SURF algorithm is fast but less accurate. Therefore, the requirement for keyframe selection and feature extraction methods is fast and accurate in object searching on real-time video. Video is pre-processed by extracting video into frames. Then, the mutual information entropy method is used for keyframe selection. Keyframes are extracted using the ORB algorithm. The multiple object detection in the video is done by clustering on features. The feature extraction results on each cluster are matched with the results of the feature from the query image. Matching results from keyframe on video with the query image is used to retrieve the video's frame information. The experiment shows that keyframe selection is beneficial in real-time video data processing because the keyframe selection speed is faster than feature extraction on each frame. Then, feature extraction using the ORB algorithm results 2 times faster speed results than SIFT and SURF algorithms with values not so different from SIFT algorithm. This study's results can be developed as a security warning system in public places, especially by security in providing evidence of criminal cases from videos.**

*Keywords*— **Object searching; real-time video; keyframe selection; mutual information entropy; ORB.**

*Manuscript received 23 May 2020; revised 30 Jan. 2021; accepted 25 Feb. 2021. Date of publication 31 Dec. 2021.*

## I. INTRODUCTION

The function of surveillance through Closed-Circuit Television (CCTV) camera in security is generally carried out in public places where there are valuables, such as ATMs, offices, schools, and shopping centers. Traditional surveillance through CCTV is very ineffective because the process of object searching on video requires the operator to observe the whole of the video [1]. When criminal acts such as theft, CCTV operators still look for evidence of items stolen on video manually by viewing the entire video from beginning to end. Recently, intelligent surveillance has begun for processing video data using algorithms for object searching [2], [3], [4].

The primary stages in video processing are pre-processing by extracting video into frames and extracting features from the frame. Research conducted by Jabnoun *et al.* [5] processed all video frames to detect household appliance objects. Processing all frames make delay in real-time video processing. Processing on real-time video requires a speed that can compensate the produced frame from the camera every second. Another disadvantage of processing all video frames is feature extraction on frames that no change on content to be inefficient [6], [7]. Therefore, the keyframe selection method is needed to reduce delay in real-time processing and reduce redundancy of processing frames containing the same information.

Researchers in video processing have widely used keyframe selection. Research conducted by Ouyang *et al.* [8] studied several keyframe selection methods. The best result from several keyframe selection methods is obtained by the mutual information entropy method that produces the right keyframe in traffic videos. The recall value of the mutual information entropy method is the highest compared to other keyframe selection methods, 89.7%. Li *et al.* [9] also use the mutual information entropy method for selecting a keyframe that produces a keyframe according to the video's main content.

The next stage is feature extraction on query image and keyframe. Some feature extraction algorithm research, including objects searching on video conducted by Jabnoun *et al.* [5] in the case of object searching for household appliances. The results showed that the SIFT algorithm results in an accuracy of 82% with a processing time of 30 seconds, whereas the SURF algorithm results in 18% accuracy with a processing time of 9 seconds for matching feature. The SIFT algorithm gives better accuracy results than SURF algorithm, but the processing time is the opposite of accuracy. The disadvantage of this feature extraction algorithm is used in processing real-time video that requires fast and accurate processing.

Some feature extraction algorithms are Harris Corner [10], [11], Scale Invariant Feature Transform (SIFT) [12], [13], Speed-Up Robust Feature (SURF) [14], [15], and Oriented FAST and Rotated BRIEF (ORB) [16], [17], [18]. Toapanta *et al.* [19] conducted a study of SIFT, SURF, and ORB methods to recognize human identity through iris that the highest result is ORB with accuracy of 99.6%. Yu and Kong [20] also compared Harris Corner, SURF, and ORB for stitching frames in videos that result in the fastest algorithm is ORB, 0.105 seconds per processing. Rublee *et al.* [16] also study image processing by taking 1000 features. The ORB algorithm is the fastest on the processing time of SIFT and SURF algorithms. The advantages of the ORB algorithm are also robust in image noise and rotate invariant. In this research, we propose a method to process real-time object searching on video that combining pre-processing stage using mutual information entropy for selecting keyframe and ORB for extracting features from keyframes. We also compare the speed and accuracy of the ORB algorithm with SIFT and SURF algorithms.

## II. Material and Method

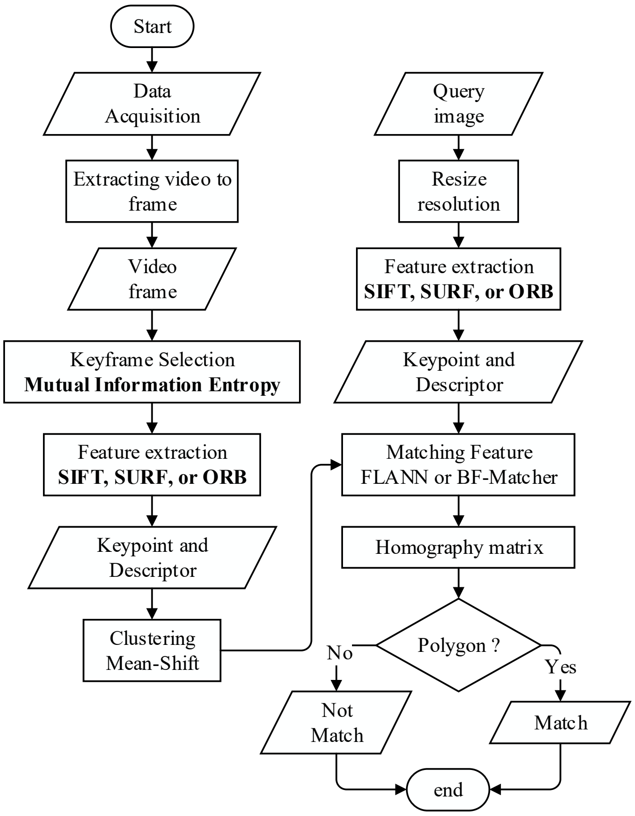

Fig. 1 Proposed method

The system starts with capturing real-time video data. Then, the video is extracted into frames. The system does not process all frames because it creates a delay when the system is running in real-time processing. We use mutual information entropy method to select frames into keyframes. Then, selected keyframes is done by feature extraction using SIFT, SURF, or ORB algorithm. Then, the keypoint and descriptor from feature extraction on the keyframe are clustered using Mean-Shift to detect multiple objects. All cluster results will be matched with keypoint and descriptor from query image. Furthermore, the input as a query image is also done by feature extraction using SIFT, SURF, or ORB algorithm. The extraction feature results on query image and keyframe are matched using FLANN for SIFT and SURF or Brute-Force Matcher for ORB. Then, the result of matching feature is done by forming a homography matrix to see the polygon formation. If the polygon is formed, the query image and keyframe is matching, and vice versa. The architecture of object searching system on video is shown in Figure 1.

### A. Data Acquisition

There are two data used in this study. The first is the query image data. Then, the second data is real-time video data. The query image is image data that contains objects to be searched on video. The resolution of the query image in this research is resized to 100×100. Figure 2 shows an example of a query image in this study. The video data is then taken from a CCTV camera with a recording height of 1.5 meters from the floor. Video data processing is done by extracting video into frames. Next, video frames are selected into keyframes.

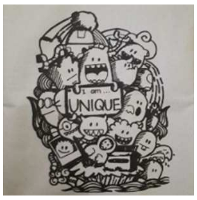

Fig. 2 The example of query image

### B. Keyframe Selection

The keyframe selection process is shown in Figure 3. The first frame automatically turns into a keyframe. Then, the next frame is calculated by the number of mutual information entropy with the previous keyframe.

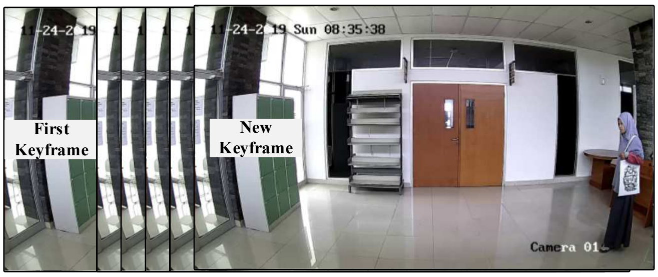

Fig. 3 Illustration keyframe selection

The mutual information entropy calculation can be seen in equation (1) [21].

$$I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \quad (1)$$

Where $I(X;Y)$ : the number of mutual information entropy on frame $x$ and $y$. $p(x,y)$ : Probability gray level on frame $x$ and $y$. Mutual information implies that two frames have the same information. If the number of mutual information from the two images is getting bigger, the two frames have almost the same information [22].

### C. Scale Invariant Feature Transform (SIFT)

According to Lowe [12], the SIFT method's algorithm consists of four stages, searching extreme values on scale-space, detecting keypoint, determination of orientation, and creating keypoint descriptors. Feature extraction using SIFT algorithm begins with constructing a scale-space (octave) using Gaussian blur with equation (2),

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \qquad (2)$$

where: $L(x,y,\sigma)$ is an image in scale space, $G(x,y,\sigma)$ is variable scales of Gaussian, and $I(x,y)$ is image intensity.

The SIFT requires 4 octaves and 5 blur scales for each detection. The first octave is two times larger than the second octave. Furthermore, each octave difference is searching using the Difference of Gaussian (DoG) with equation (3), resulting in 4 octaves of DoG.

$$D(x,y,\sigma) = \big(G(x,y,k\sigma) - G(x,y,\sigma)\big) * I(x,y) \qquad (3)$$

where $D(x,y,\sigma)$ is convolution on image with Difference of Gaussian filter.

Keypoint determination begins with taking a sample point which is compared with its neighbors (26 pixels neighboring). If the point has the smallest (local minima) or largest (local maxima) value, it will become candidate keypoint. Then, candidate keypoints are filtered against low contrast keypoints and keypoints are located near the edge. Then, keypoints are calculated on magnitude and angle. This stage makes SIFT invariant towards orientation.

In the step of creating descriptor, SIFT creates an area of 16×16-pixel size around the keypoint and 4×4 sub-areas with 8 orientation directions in each sub-area. Furthermore, each sub area is made into a bin histogram to store the orientation of the keypoint that has similarities in a certain angular range. The SIFT descriptor is normalized so the descriptor value is not affected by lighting changes. The final result is 128 descriptors from 8 directions on each sub area.

### D. Speed-Up Robust Features (SURF)

Feature extraction using SURF algorithm is searching blob which is a set of pixels that have same intensity. Bay *et al.* [14] divides SURF stage into three stages, representation of scale space, keypoint detection, and keypoint descriptors. SURF algorithm begins with the establishment of integral images using equation (4).

$$I(x,y) = \sum_{x'}^{x} \sum_{y'}^{y} N(x',y') \qquad (4)$$

Where:

$I(x,y)$ : Integral image

$N(x',y')$ : Center representation on image with consist of number gray level

SURF algorithm establishes pyramid images where is not going through blurring image. Keypoint searching is performed on scale-space that forms a pyramid image. Scale-space is resulting in keypoint that has scale invariant. Furthermore, keypoint on scale space is selecting candidate keypoint using non-maxima suppression. Candidate keypoints are sought by using local maxima and determinant of Hessian Matrix as in equation (5) at the testing points $x\,(x,y,\sigma)$ of integral images.

$$Det(H_{approx}) = D_{xx}D_{xy} - (0.9D_{xy})^2 \qquad (5)$$

SURF algorithm checks 26 neighbor points between scales. If the Hessian extremity value at the test point is greater than all the neighbors, the test point is a keypoint.

The last stage is descriptor on keypoint. Each keypoint must have a unique descriptor so it is not affected by image rotation. This process is carried out with the Haar Wavelet response to the $x$ and $y$ directions referring to the values of $dx$ and $dy$. The SURF descriptor is an area of size 20s. Each area is divided into 4×4 subarea. Each subarea is explained by the Haar Wavelet response based on a 5×5 sample with a vector containing 4 components. The result of SURF descriptor contains 64 dimensions.

### E. Oriented FAST and Rotated BRIEF (ORB)

ORB consists of FAST to detect keypoint and BRIEF to create descriptor on each keypoint [23]. ORB is free from the licensing restrictions of SIFT and SURF [16]. The ORB algorithm starts with transformation scale pyramid on image. Then, it uses FAST detector to detect corner of the image. FAST detects a keypoint such as pixel $p$ compared to 16 pixels around it that form a circle. The circle pixels are sorted into 3 classes, i.e., brighter than $p$, darker than $p$, and equal to $p$. If there are more than 8 pixels that are darker or lighter than $p$, pixel $p$ becomes a keypoint. The results of FAST detection are calculated using Harris Corner to find the best keypoint. Afterward, an orientation searching on object is performed using centroid intensity as within equation (6) [24].

$$\begin{gathered} C(\bar{x},\bar{y}) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \\ \theta = atan2(m_{01}, m_{10}) \end{gathered} \qquad (6)$$

where is $C(\bar{x},\bar{y})$ is centroid of object on image, $m_{00}$ is moment level 0 (area of object), and $m_{10}, m_{01}$ is moment level 1.

The results from keypoint detection are extracted using BRIEF algorithm, as it does not have rotational invariant. The next step is comparing all sampling pairs (the first pixel with the second pixel on image). If the first pixel is brighter than the second pixel, it has a value of 1, or else it will be 0. This step is done with following equation:

$$\tau(p;x,y) := f(x) = \begin{cases} 1, p(x) < p(y) \\ 0, p(x) \geq p(y) \end{cases} \qquad (7)$$

where $p(x)$ is the value of pixel $x$ intensity and $p(y)$ is the value of pixel $y$ intensity. This step will be repeated up to 256 pairs. Then, 256 bits is converted to byte that resulting in binary descriptor with 32 dimensions.

### F. Clustering Features on Keyframe

The feature extraction of the SIFT, SURF, and ORB algorithms is shown in Figure 4. The keypoint points are marked in green. It can be seen that the results of the ORB algorithm only produce a few keypoints. This is because the ORB algorithm has a Harris cornet detection filter. Feature extraction using SIFT, SURF, or ORB algorithms on keyframes generates keypoints and descriptors. Furthermore, keypoints and descriptors on video keyframes are clustered based on the closest neighbor which point to detect multiple objects on keyframes.

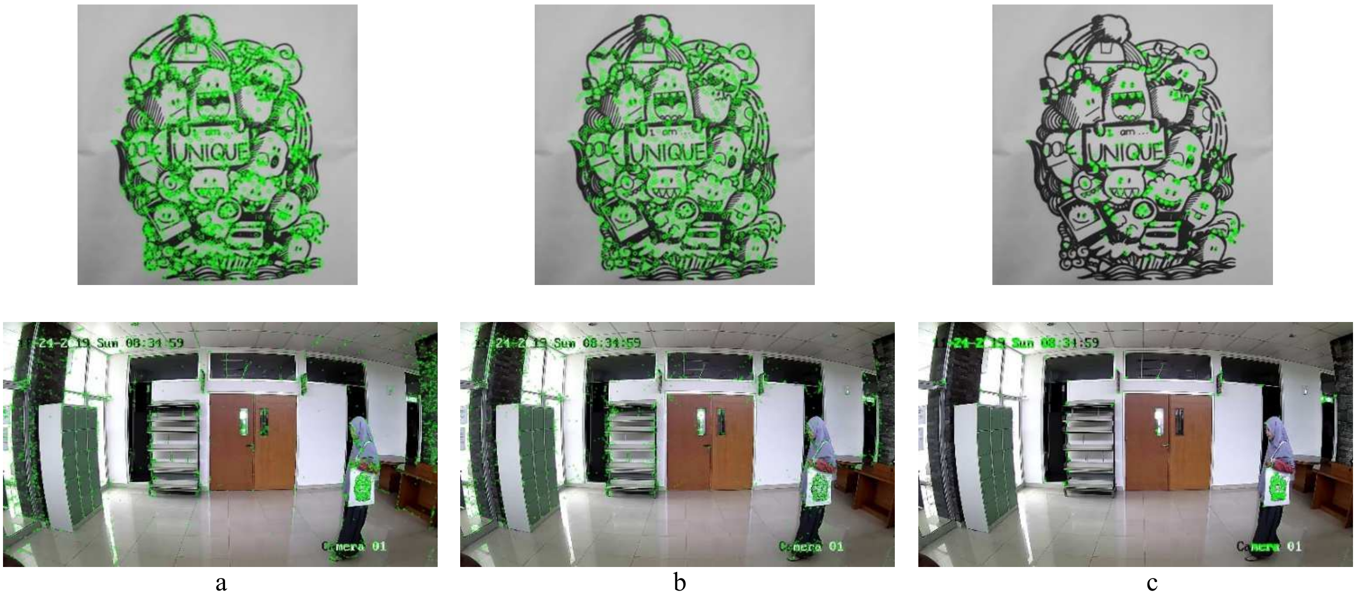

Fig. 4 The result of feature extraction a) The SIFT algorithm, b) The SURF algorithm, c) The ORB algorithm

The clustering algorithm that used in this study is Mean-Shift. The Mean-Shift algorithm is based on centroid on keypoints which continuously updates the centroid candidate by calculating the mean at all points according to the window area [25]. Furthermore, the candidate's centroid is filtered to eliminate the duplication of the adjacent centroid. Candidate centroid $x_i$ in iteration $t$ is updated continuously with following equation.

$$x_i^{t+1} = m(x_i^t) \tag{8}$$

where $x_i$ is candidate centroid and $m$ is a mean-shift vector.

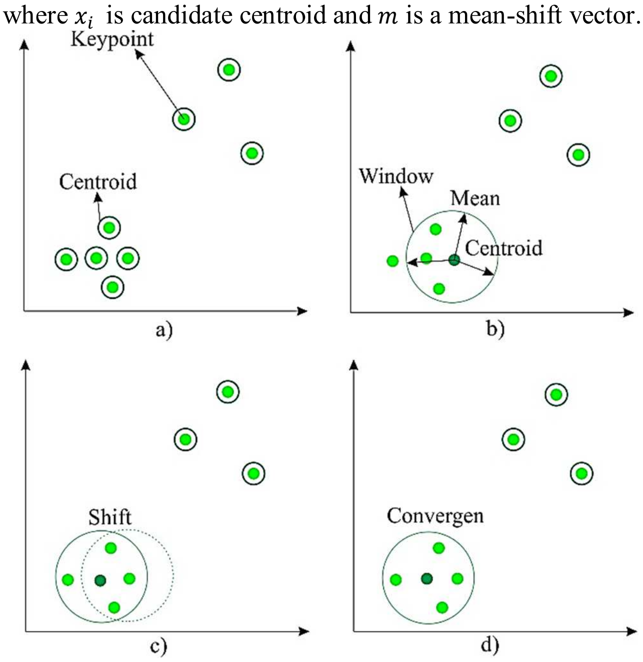

Fig. 5 The Mean-Shift algorithm illustration

Updating centroid $m(x_i)$, so it becomes mean of the sample points within the neighborhood that is done using equation (9)

$$m(x_i) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)} \tag{9}$$

where $N(x_i)$ is neighborhood of samples.

The Mean-Shift algorithm automatically determines the number of clusters. This is based on the bandwidth parameters that determine the size of the region to search through.

The clustering process using the Mean-Shift algorithm begins by making the key point from feature extraction as the center of the cluster, as shown in Figure 5a. Furthermore, the window size (kernel bandwidth) is determined automatically through the estimate bandwidth function. As the algorithm's name implies, this algorithm calculates the mean cluster center of all points in the window based on the nearest neighbors, as shown in Figure 5b. This algorithm then performs a shift in a denser area by renewing the center of the cluster's mean value with its neighboring points using equation 9, as shown in Figure 5c. The algorithm will stop when the cluster center position has not shifted, with the final result shown in Figure 5d. Figure 6 shows an example of the Mean-Shift algorithm operation on a video keyframe that produces 4 clusters.

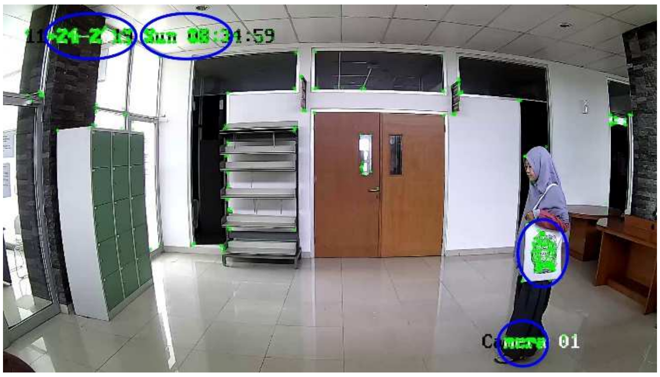

Fig. 6 The example result of Mean-Shift operation on keyframe

### G. *Matching Feature Query Image with Keyframe*

The matching feature stage SIFT and SURF algorithms are using FLANN method while ORB algorithm is using BF-Matcher. Matching features occur if there are at least 4 keypoint good matches. If the results of the best keypoint matching (good match) are more than or equal to four, a Homography matrix [26,27] search of the two images is performed. The image will have geometrical transformations such as translation, rotation, scaling, shear. The next step is checking whether the Homography matrix is formed. If it is not formed, the process will be stop which indicates it does not match. Homography matrix is used to find the angle of an image. If the corners are connected to be polygon, the query image and keyframe are declared to match. If the connected corner does not form a polygon, the image query and keyframe are declared not match.

*1) FLANN Method*: The Fast Library Approximated Nearest Neighbor (FLANN) method is used for finding the value of nearest neighbor [28,29]. Descriptors are produced by SIFT algorithm that is 128 dimensions for each keypoint while the SURF algorithm has 64 dimensions descriptor. Therefore, using the FLANN method for matching multi-dimensional data is needed. FLANN uses the K-Dimensional Tree (KD Tree) index type. KD-Tree is a multidimensional binary tree data representation that aims to separate certain data areas based on their position value. An illustration of the FLANN method using the KD-Tree algorithm is shown in Figure 7.

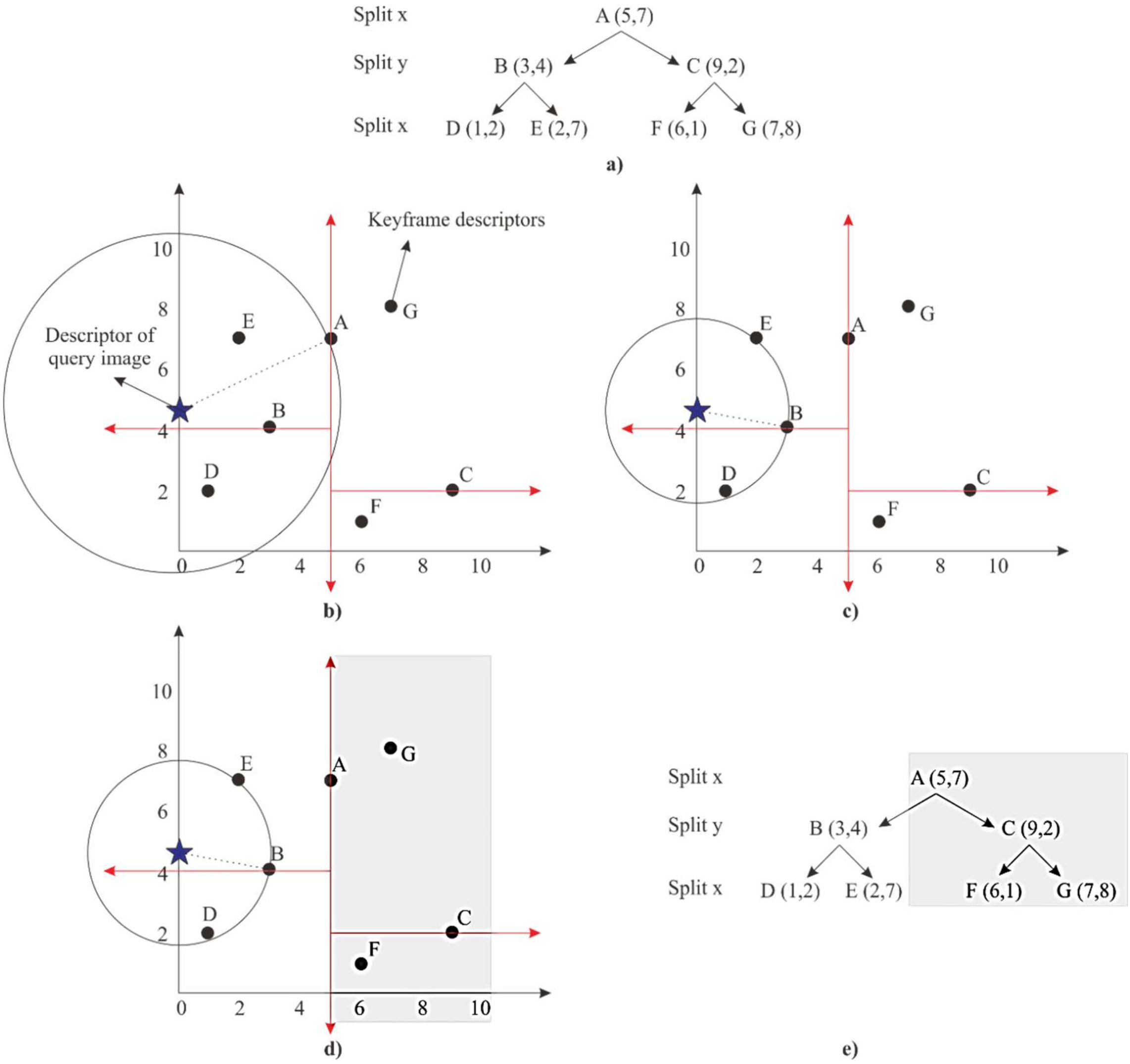

Fig. 7 The illustration of FLANN method with KD-Tree algorithm

Illustration using the KD-Tree algorithm, for example, using a 2-dimensional descriptor. For example, there are 7 keypoints, with descriptor {(5,7), (3,4), (9,2), (1,2), (2,7), (6,1), (7,8)}. Descriptor (5,7) becomes the root, the next descriptor is placed in the left or right tree depending on the split part, as shown in Figure 7a. The resulting tree is then modeled with coordinates to facilitate clustering. The boundaries of the red lines, as shown in Figure 7b, are the clusters formed. Then, suppose there is an asterisk as the query image's descriptor, as shown in Figure 7b. It turns out that point A is not the nearest neighbor of the query image. The distance used in the nearest neighbor using Euclidean Distance. Next, look for the nearest neighbor point, found at point B, as shown in Figure 5c. After finding the nearest neighbor between the query image and the keyframe, point A, G, F, C are marked as no nearest neighbors, as shown in Figure 7d and Figure 7e.

*2) BF-Matcher Method*: The ORB algorithm produces keypoint and binary descriptors in query image and keyframe. The BF-Matcher work is comparing each descriptor in the query image with all descriptors on the keyframe to find the smallest result [30]. ORB generates 32 descriptors for each keypoint, as shown in Table 1, an example of a descriptor in the query image. Then, Table 2 is an example of a descriptor on a keyframe.

TABLE I
THE EXAMPLE OF QUERY IMAGE DESCRIPTOR

| Coordinate Keypoint | The descriptor of query image |
|---|---|
| (1,6) | [2 141 2 158 148 131 174 230 22 0 **88** 2 **32** 234 48 64 89 106 19 195 **82** 184 8 255 40 38 142 **136** 20 **90** 176 136] |
| (10,6) | [2 141 2 158 148 131 174 230 22 0 90 2 96 234 48 64 89 106 19 193 82 **184** 10 255 40 38 142 192 20 90 176 136] |
| (1,13) | [2 141 2 158 148 131 174 230 22 1 88 2 96 232 48 64 89 106 2 193 114 184 8 255 40 38 142 **192** 20 90 176 136] |
| (10,13) | [2 141 2 158 148 131 174 231 22 0 90 2 96 234 32 64 89 106 19 192 114 136 10 255 40 38 142 200 20 90 176 136] |

TABLE II
THE EXAMPLE OF KEYFRAME DESCRIPTOR

| Coordinate Keypoint | The descriptor of keyframe |
|---|---|
| (9,8) | [2 141 2 158 148 131 174 230 22 0 **90** 2 **96** 234 48 64 89 106 19 195 **66** 184 8 255 40 38 142 **128** 20 **74** 176 136] |
| (9,13) | [2 141 2 158 148 131 174 230 22 0 90 2 96 234 48 64 89 106 19 193 82 **152** 10 255 40 38 142 192 20 90 176 136] |
| (3,8) | [2 141 2 158 148 131 174 230 22 1 88 2 96 232 48 64 89 106 2 193 114 184 8 255 40 38 142 **200** 20 90 176 136] |
| (3,13) | [2 141 2 158 148 131 174 231 22 0 90 2 96 234 32 64 89 106 19 192 114 136 10 255 40 38 142 200 20 90 176 136] |

The Hamming distance calculates the difference in the descriptor. The smaller the distance value, the more match the descriptor between the query image and the keyframe. For example, using the data descriptor in Table 1 and Table 2, coordinates (1,6) with (9.8) distance value 5, coordinates (10,6) with (9,13) distance value 1, coordinates (1,13) with (3,8) a distance value of 1, and coordinates (10,13) with (3,13) a distance value of 0. Then we look for the smallest and second smallest values to do a good match search.

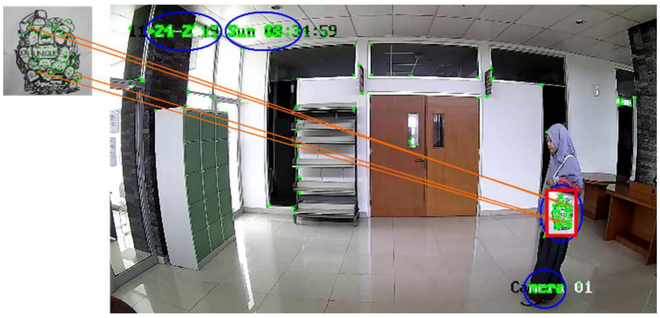

Fig. 8 The example of matching feature between query image and keyframe

An example of the matching feature between the query image and the keyframe is shown in Figure 8. Each clustering result on the keyframe is matched with the query image. The query image's bag object matches one of the clusters on the keyframe, namely the bag object. Using the Homography matrix function, the matching results between keypoints are represented according to the query image's angle. Therefore the polygon shape on the video keyframe appears in the object's corners following the query image. A red mark on the keyframe indicates a polygon is formed so that the query image matches on that keyframe.

## III. RESULTS AND DISCUSSION

### A. Data Acquisition

The video is collected from a recording using CCTV HiLook Wi-Fi PT camera with IPC-P120-D / W model. The resolution of the CCTV camera is 2.0 MP with 10 frames per second (fps). The duration of the video is 60 seconds. This study is using 4 types of bags as an object with different patterns such as textured bag (Batik), lettering patterned bag, black and white patterned bag, and color pattern bag.

### B. The Experiment of Keyframe Selection

The experiment of keyframe selection and the suitability of keyframes generated are using a combination of variations in the bin's parameter and the threshold value of mutual information entropy method. In this experiment, the number of frame data is 607 frames. Table 3 shows the results of the keyframe selection experiment. The suitability of the resulting keyframe can be seen from appearance of number of keyframes in recording with no objects which only produces one keyframe.

TABLE III
KEYFRAME SELECTION EXPERIMENT

| Bins parameter | Threshold | Keyframe | Suitability |
|---|---|---|---|
| 5 | 1.1 | 7 | Bad |
| 5 | 1.3 | 62 | Good |
| 5 | 1.5 | 607 | Bad |
| 10 | 1.1 | 1 | Bad |
| 10 | 1.3 | 1 | Bad |
| 10 | 1.5 | 8 | Bad |

From the experiment, keyframe selection with bins 7 and threshold 1.1 give better result than other combination. Then, the speed of keyframe selection is 6 ms in processing per frame. This method is much faster than speed of feature extraction method. Based on Table IV, the matching feature on all algorithm, the fastest algorithm gets 0.719 second in processing frame. Therefore, keyframe selection can reduce delays in real-time video processing.

### C. The Experiment of Recording Resolution

The resolution of the video needs to be tested because it is the determination of the best resolution for $F_1$ value and speed in real-time processing. The resolution experiment is performed on Full HD (1920×1080), HD (1280×720) and VGA (640×480). The black and white patterned bag is used to see the optimal resolution in speed and also $F_1$ value.

TABLE IV
THE EXPERIMENT RESULT OF VIDEO RESOLUTION

| Resolution | Speed of processing frame (second) | | |
|---|---|---|---|
| | SIFT | SURF | ORB |
| Full HD | 3.488 | 3.126 | 1.286 |
| HD | 1.931 | 1.893 | 0.902 |
| VGA | 0.864 | 1.103 | 0.719 |

In Figure 9, the best $F_1$ value was at Full HD resolution but based on Table 4, in Full HD resolution to process one keyframe takes an average of 3.488 seconds on the SIFT algorithm, 3.126 seconds on the SURF algorithm, and 1.286 seconds on the ORB algorithm. Speed above 1 second for real-time processing is very slow. Table 4 shows that the result of $F_1$ value experiment on the HD resolution is quite good and the speed experiment is faster than Full HD resolution. In all resolutions, the $F_1$ The ORB algorithm value is better than the SURF algorithm, and the speed experiment of the ORB algorithm is the fastest of the SIFT and SURF algorithms.

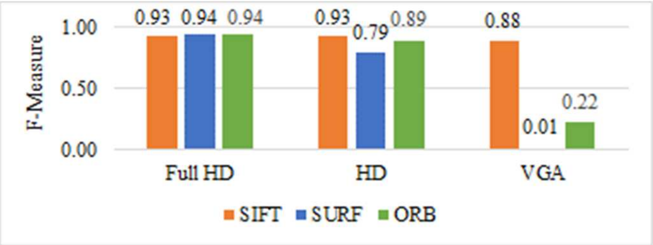

Fig. 9 The experiment result of effect on video resolution toward $F_1$ value

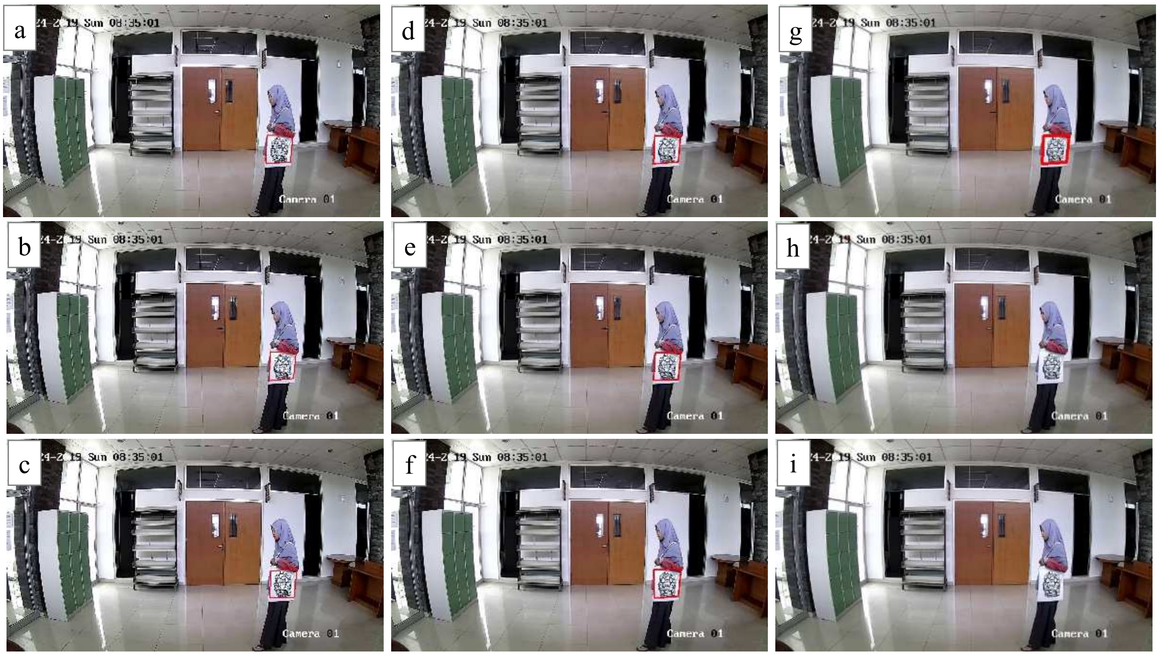

Fig. 10 The result of matching feature on Full HD resolution: a) SIFT b) SURF c) ORB, HD resolution: d) SIFT e) SURF f) ORB, VGA resolution: g) SIFT h) SURF i) ORB

Figure 10 shows that in VGA resolution, the $F_1$ value is not good. As shown in Figure 10h and 10i, in VGA resolution for SURF and ORB algorithms is not detected. This variation in resolution affects to number of features obtained. The smaller size of resolution makes fewer features that can be obtained. But if using too large resolution like Full HD, the processing time is also longer.

### D. *The Experiment of Object Distance*

In this experiment, the distance of object to CCTV camera is varied, which is 1 meter, 2 meters, and 3 meters. The object used in this experiment is textured bag (Batik), lettering patterned bag, black and white patterned bag, and color pattern bag. Figure 11 shows the results of $F_1$ value from this experiment.

On $F_1$ average results as shown in Figure 11, the $F_1$ value of SIFT, SURF, and ORB algorithms decreases at 3 meters distance experiment. If it is too close to camera at 1 meter distance, the $F_1$ value is also not optimal for the SURF and ORB algorithms. This is because some objects are not recorded, so the feature that obtained from the object is reduced. Through this experiment, it shows that all algorithm is scale-invariant. Its means that bag object with vary in the distance from camera can still detect. The best algorithm in scale invariant is SIFT that 3 meters distance can still be detected in all types of bags.

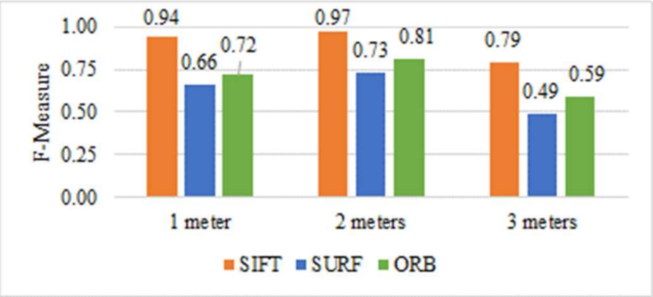

Fig. 11 The experiment result of effect on object distance toward $F_1$ value

Based on Figure 12, at distance of 3 meters is resulting in many false negative because the object in video is getting smaller which causes the feature to appear blurry, thereby reducing the number of features captured. As shown in Figure 12b, 12c, 12h, 12i which are an example of false negative. False negative is caused by features that match less than 4 so a homography matrix cannot be formed.

### E. Discussion

SIFT algorithm is good because it has 128 dimensions of descriptor and matching features is using distance matching, i.e. Euclidean distance. This is having an impact on processing time because matching uses 128 dimensions in each feature. Then, SURF algorithm uses BLOB feature so the resulting feature is not good which affects to $F_1$ value. SURF algorithm is slightly faster than SIFT because the descriptor of SURF algorithm is 64 dimensions. Whereas the ORB algorithm produces 32 dimensions of descriptor and the matching feature uses binary matching, namely Hamming distance, it is much faster than SIFT and SURF algorithms. The feature that used in ORB algorithm is using a corner, so the selection of unique features affects $F_1$ value that is not too different than SIFT algorithm.

The ORB algorithm's speed is the fastest of the SIFT and SURF algorithms based on the experimental results. In a previous study [5], the accuracy and speed in processing a video frame in the SIFT and SURF algorithms are inversely related. In this study, besides the ORB algorithm is the fastest in processing a frame, the $F_1$ value of the ORB algorithm is 0.81, which is not much different from the SIFT algorithm of 0.97 and better than the SURF algorithm, which is only 0.73. These results prove that the ORB algorithm for object searching in the real-time video is the fastest on processing time, and the $F_1$ value is not much different from the SIFT algorithm.

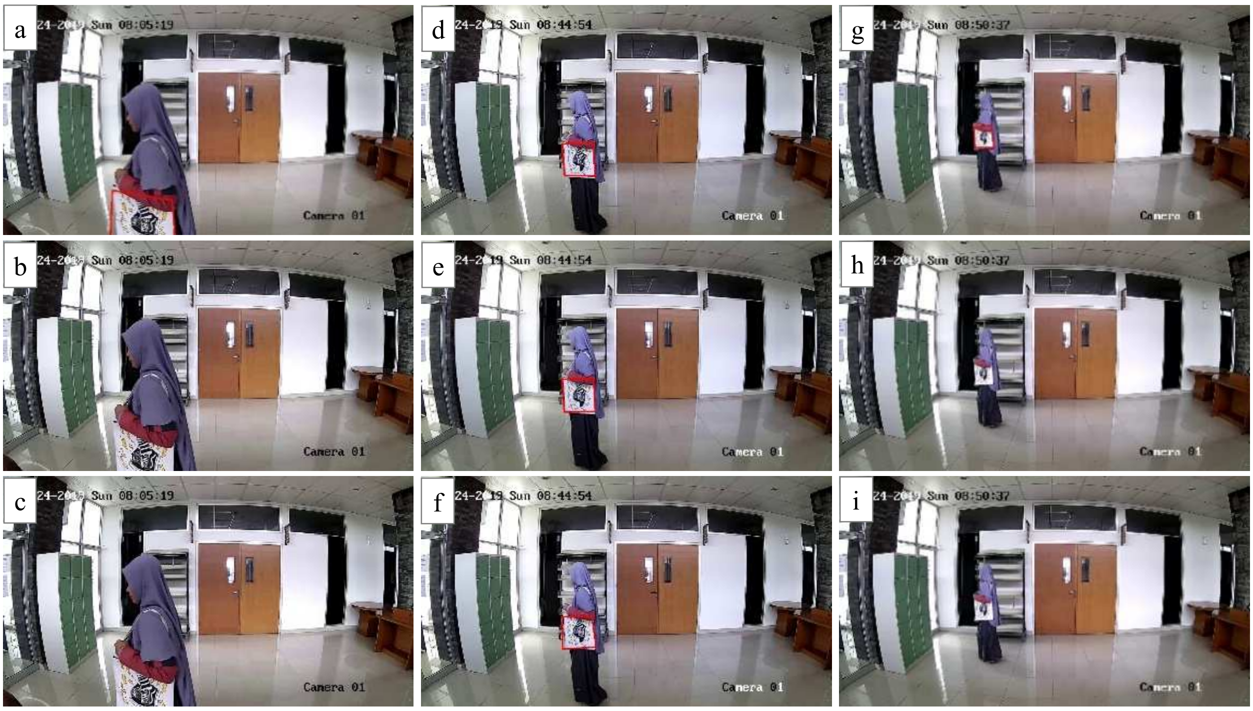

Fig. 12 The result of matching feature on 1 meter distance: a) SIFT b) SURF c) ORB, 2 meters distance: d) SIFT e) SURF f) ORB, 3 meters distance: g) SIFT h) SURF i) ORB

## IV. Conclusion

This study provides an overview of real-time video data processing algorithms that are fast and accurate in object searching. This system helps CCTV operators or security forces analyze in finding evidence of criminal acts of theft of goods on video. Through video analysis, related parties can provide data on the goods in question recorded on how many frames or seconds. In this study, three local feature algorithms were tested: the SIFT, SURF, and ORB, to process real-time video data. We also added a keyframe selection because several video frames have the same information.

In this study, Mutual Information Entropy method can be used for selecting keyframes, thereby reducing video delay in real time processing because not all frames are processed. Then, the ORB algorithm can be applied as feature extraction for object searching on video that the result of processing time is the fastest compared to SIFT and SURF algorithms with $F_1$ value that is not too different. However, the proposed method is still slow because detecting multiple objects uses the Mean-Shift method that is computationally relatively expensive. Our future work will be combined with faster clustering method.

## References

[1] G. F. Shidik, E. Noersasongko, A. Nugraha, P. N. Andono, J. Jumanto, and E. J. Kusuma, "A systematic review of intelligence video surveillance: Trends, techniques, frameworks, and datasets," *IEEE Access*, vol. 7, pp. 170457–170473, 2019, doi: 10.1109/ACCESS.2019.2955387.

[2] X. Huang, D. Mu, and Z. Li, "Intelligent traffic analysis: A heuristic high-dimensional image search algorithm based on spatiotemporal probability for constrained environments," *Alexandria Eng. J.*, vol. 59, no. 3, pp. 1413–1423, 2020, doi: 10.1016/j.aej.2020.03.045.

[3] M. A. Uddin, A. Alam, N. A. Tu, M. S. Islam, and Y. K. Lee, "SIAT: A distributed video analytics framework for intelligent video surveillance," *Symmetry (Basel).*, vol. 11, no. 7, 2019, doi: 10.3390/sym11070911.

[4] E. Cermeño, A. Pérez, and J. A. Sigüenza, "Intelligent video surveillance beyond robust background modeling," *Expert Syst. Appl.*, vol. 91, pp. 138–149, 2018, doi: https://doi.org/10.1016/j.eswa.2017.08.052.

[5] H. Jabnoun, F. Benzarti, and H. Amiri, "Object recognition for blind people based on features extraction," *Int. Image Process. Appl. Syst. Conf. IPAS 2014*, pp. 1–6, 2014, doi: 10.1109/IPAS.2014.7043293.

[6] S. Kannappan, Y. Liu, and B. Tiddeman, "DFP-ALC: Automatic video summarization using Distinct Frame Patch index and Appearance based Linear Clustering," *Pattern Recognit. Lett.*, vol. 120, pp. 8–16, 2019, doi: 10.1016/j.patrec.2018.12.017.

[7] X. Yan, S. Z. Gilani, M. Feng, L. Zhang, H. Qin, and A. Mian, "Self-supervised learning to detect key frames in videos," *Sensors (Switzerland)*, vol. 20, no. 23, pp. 1–18, 2020, doi: 10.3390/s20236941.

[8] S. Z. Ouyang, L. Zhong, and R. Q. Luo, "The comparison and analysis of extracting video key frame," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 359, no. 1, 2018, doi: 10.1088/1757-899X/359/1/012010.

[9] W. Li, D. Qi, C. Zhang, J. Guo, and J. Yao, "Video summarization based on mutual information and entropy sliding window method," *Entropy*, vol. 22, no. 11, pp. 1–16, 2020, doi: 10.3390/e22111285.

[10] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," pp. 23.1-23.6, 2013, doi: 10.5244/c.2.23.

[11] C. A. V. Hernández and F. A. P. Ortiz, "A corner detector algorithm for feature extraction in simultaneous localization and mapping," *J. Eng. Sci. Technol. Rev.*, vol. 12, no. 3, pp. 104–113, 2019, doi: 10.25103/jestr.123.15.

[12] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[13] J. Mohan and M. S. Nair, "Domain independent redundancy elimination based on flow vectors for static video summarization," *Heliyon*, vol. 5, no. 10, p. e02699, 2019, doi: 10.1016/j.heliyon.2019.e02699.

[14] H. Bay, A. Ess, T. Tuytelaars, and L. Vangool, "Speeded-Up Robust Features (SURF) (Cited by: 2272)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2006, doi: 10.1016/j.cviu.2007.09.014.

[15] C. Barajas-García, S. Solorza-Calderón, and E. Gutiérrez-López, "Scale, translation and rotation invariant Wavelet Local Feature Descriptor," *Appl. Math. Comput.*, vol. 363, 2019, doi: 10.1016/j.amc.2019.124594.

[16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2564–2571, 2011, doi: 10.1109/ICCV.2011.6126544.

[17] C. A. Díaz, D. S. Pérez, H. Miatello, and F. Bromberg, "Grapevine buds detection and localization in 3D space based on Structure from Motion and 2D image classification," *Comput. Ind.*, vol. 99, no. September 2017, pp. 303–312, 2018, doi: 10.1016/j.compind.2018.03.033.

[18] F. Previtali, P. Bertolazzi, G. Felici, and E. Weitschek, "A novel method and software for automatically classifying Alzheimer's disease patients by magnetic resonance imaging analysis," *Comput. Methods Programs Biomed.*, vol. 143, pp. 89–95, 2017, doi: https://doi.org/10.1016/j.cmpb.2017.03.006.

[19] S. M. T. Toapanta, A. A. C. Cruz, L. E. M. Gallegos, and J. A. O. Trejo, "Algorithms for efficient biometric systems to mitigate the integrity of a distributed database," *CITS 2018 - 2018 Int. Conf. Comput. Inf. Telecommun. Syst.*, pp. 1–5, 2018, doi: 10.1109/CITS.2018.8440190.

[20] H. Yu and L. Kong, "An Optimization of Video Sequence Stitching Method," *2018 IEEE Int. Conf. Mechatronics Autom.*, pp. 387–391, 2018.

[21] M. Chen, X. Han, H. Zhang, G. Lin, and M. M. Kamruzzaman, "Quality-guided key frames selection from video stream based on object detection," *J. Vis. Commun. Image Represent.*, vol. 65, p. 102678, 2019, doi: https://doi.org/10.1016/j.jvcir.2019.102678.

[22] Y. Chen and G. Liu, "Content adaptive Lagrange multiplier selection for rate-distortion optimization in 3-D wavelet-based scalable video coding," *Entropy*, vol. 20, no. 3, 2018, doi: 10.3390/e20030181.

[23] F. Hidalgo and T. Bräunl, "Evaluation of several feature detectors/extractors on underwater images towards vslam," *Sensors (Switzerland)*, vol. 20, no. 15, pp. 1–16, 2020, doi: 10.3390/s20154343.

[24] T. Rao and T. Ikenaga, "Quadrant segmentation and ring-like searching based FPGA implementation of ORB matching system for Full-HD video," *Proc. 15th IAPR Int. Conf. Mach. Vis. Appl. MVA 2017*, pp. 89–92, 2017, doi: 10.23919/MVA.2017.7986797.

[25] Y. Tian and Y. Yokota, "Estimating the major cluster by mean-shift with updating kernel," *Mathematics*, vol. 7, no. 9, pp. 1–25, 2019, doi: 10.3390/math7090771.

[26] W. Chojnacki, Z. L. Szpak, and M. Wadenbäck, "The equivalence of two definitions of compatible homography matrices," *Pattern Recognit. Lett.*, vol. 135, pp. 38–43, 2020, doi: https://doi.org/10.1016/j.patrec.2020.03.033.

[27] D. Ane Delphin, M. R. Bhatt, and D. Thiripurasundari, "Holoentropy measures for image stitching of scenes acquired under CAMERA unknown or arbitrary positions," *J. King Saud Univ. - Comput. Inf. Sci.*, 2018, doi: 10.1016/j.jksuci.2018.08.006.

[28] M. Mateu-Mateus, F. Guede-Fernández, N. Rodriguez-Ibáñez, M. A. García-González, J. Ramos-Castro, and M. Fernández-Chimeno, "A non-contact camera-based method for respiratory rhythm extraction," *Biomed. Signal Process. Control*, vol. 66, p. 102443, 2021, doi: https://doi.org/10.1016/j.bspc.2021.102443.

[29] Q. shu Qian, Y. hua Hu, N. xiang Zhao, M. le Li, and F. cai Shao, "Summed volume region selection based three-dimensional automatic target recognition for airborne LIDAR," *Def. Technol.*, vol. 16, no. 3, pp. 535–542, 2020, doi: 10.1016/j.dt.2019.10.011.

[30] M. R. Purohit and A. R. Yadav, "Comparative Analysis of Features Extraction Methods for Detection Traffic Rule Violation on Raspberry Pi Hardware," vol. 29, no. 3, pp. 10953–10963, 2020.