ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

Multiple Objects Tracking with Location Matching and Adaptive Windowing Based on SIFT Algorithm

Seok-Wun Ha[1]

[1]Department of Informatics, Gyeongsang National University, Rep. of Korea
swha@gnu.ac.kr

Abstract: Multiple objects tracking have been an interesting research topic in computer vision and its related fields. It is a very important work to detect exactly the consecutive multiple objects and to track them effectively. In this paper, we propose a robust tracking system that utilizes several techniques such as multiple objects detection from multi-lateral histogram, location matching of the feature descriptor from Scale Invariant Feature Transform (SIFT) algorithm, and adaptive windowing for effective tracking. In order to analyze the performance of the proposed tracking system three videos were tested that multiple objects show various types of appearances. Experimental results reveal that the proposed system has an advanced tracking ability in complicated circumstances.

Keywords: Multiple Objects SIFT Location Matching Adaptive Windowing

I. INTRODUCTION

In multiple objects tracking the work beforehand is to detect objects that appear sequentially and randomly with no absence. For a single object its edge is formed by calculating the brightness difference between two consecutive video frames and its area is detected using the lateral histogram [1-3]. But in case of appearances of multiple objects a problem is generated that not be able to discriminate them because objects areas are overlapped in either the horizontal or vertical axis according to the reference. In order to resolve this problem we worked out a simple extended way- multi-lateral histogram- that detects multiple object areas by subdividing the horizontal histogram line into several objects areas and then forming the vertical histogram in each horizontal area. However, even case this is used, the object area overlapping is unavoidable that generates by approaching too close to each other and it can only be solved by using the invariant features of the objects. There are several algorithms that extract the invariant features from the specific target or image such as SIFT [4], GLOH [5], and SURF [6]. SIFT gives a robust characteristic for size, noise, brightness, and local distortion. GLOH shows the better performance for a structured image comparing with SIFT, but it has a disadvantage that the grid size for feature extraction extends to be doubled. SURF shows a rapid matching speed comparing with SIFT. For realizing more precise matching between objects that presents on two consecutive frames it is very important to be only concentrated to inside area of the object and to analyze location information of invariant features extracted from the object. SIFT provides an invariant feature description vector that includes location, size, and orientation information, therefore utilizing this characteristic of the SIFT would enable to perform location analysis and matching. In object tracking, because the absolute values of location and size for the comparable objects between two consecutive frames are altered, the size of the windows and their corresponding locations should be recalculated adaptively. In this paper, we propose a multiple object tracking system that uses the aforementioned techniques, analyze their characteristics. And using several videos with a variety of movements of multiple objects the performance of the proposed system is experimented.

II. MULTIPLE OBJECTS TRACKING SYSTEM

The proposed multiple objects tracking system is totally composed of four steps of obtaining the object-included window and the object area based on the multi-lateral histogram, extraction the invariant features from the object-included window and the object area using SIFT algorithm, computing the matching rate and controlling the size of the object-included window adaptively, and tracking the multiple objects utilizing the matching results.

The detailed processing steps of the proposed multiple objects tracking system are as follows:

Step 1: make a difference between the gray-level images of the reference frame and current frame and binarize the difference image.

Step 2: obtain the multi-lateral histogram of the binarized image and process a smoothing for the multi-lateral histogram.

Step 3: Determine the areas that multiple objects exist and detect the objects in the areas.

Step 4: Set up the search window to be slightly larger than the object window based on the object window that was controlled adaptively in the previous frame.

Step 5: Extract the invariant features of the objects using SIFT algorithm and then store features to the current feature buffer.

Step 6: Compute the matching rate of the feature values with location information between the previous and current feature vectors.

Step 7: Check whether the objects participate or not to the location matching. If yes, they are considered as the existing objects and go to next step 8, or not, they are guessed as new objects or disappearing objects and jump to Step 9.

Step 8: discriminate whether they are location matched or not. If matched, the feature vector is stored to the new previous feature buffer that needs to match with the current feature buffer of the consecutive next frame, or not, they are rejected. And the object window is adjusted adaptively according to the size of the moving object.

Step 9: In step 7, if the objects don't participate to the location matching, they are either appeared newly or no longer appeared, that is exit. If newly appeared, their features are stored the new previous feature buffer, or not, they are rejected.

Step 10: Repeat from step 1 to step 9 about consecutive next current frame image. These are repeated sequentially and multiple object tracking would be continued.

Figure 1 show the overall flow chart of the proposed multiple objects tracking system.

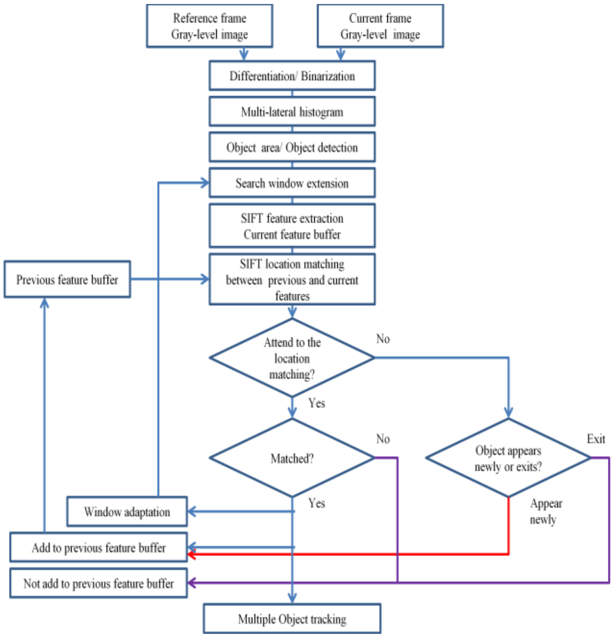

Reference frame Gray-level image
Current frame Gray-level image
Differentiation/ Binarization
Multi-lateral histogram
Object area/ Object detection
Search window extension
SIFT feature extraction Current feature buffer
Previous feature buffer
SIFT location matching between previous and current features
Attend to the location matching?
No
Yes
Matched?
No
Object appears newly or exits?
Exit
Yes
Appear newly
Window adaptation
Add to previous feature buffer
Not add to previous feature buffer
Multiple Object tracking

Figure 1: The detailed flow chart of the proposed multiple objects tracking system

A. *Multiple Objects Detection*

Traditional lateral histogram is a statistic that integrated the brightness values of the pixels for the same row and column in the horizontal and the vertical axis directions of a gray-level image and it has been used to detect the existing area and the shape of a specific object [1][7]. In case that several objects appear in a frame image, a problem is caused that multiple objects are recognized as a single object because the histograms of these objects are overlapped in the direction they are standing.

But, if each object regions are divided first from the horizontal axis histogram and then the vertical axis histogram is calculated about the individual horizontal regions, multiple objects areas could be obtained easily. This method is a simple extended one from the traditional lateral histogram and it is called multi-lateral histogram. However even using this multi-lateral histogram it is actually not possible to discriminate the objects on conditions that multiple objects are close to each other or overlapped. Because these could be solved by matching their invariant features that be able to obtain by using SIFT algorithm, this simple multi-lateral histogram technique is very useful and concrete to detect multiple objects. Figure 2 presents the examples of the multi-lateral histogram and that the objects areas are detected.

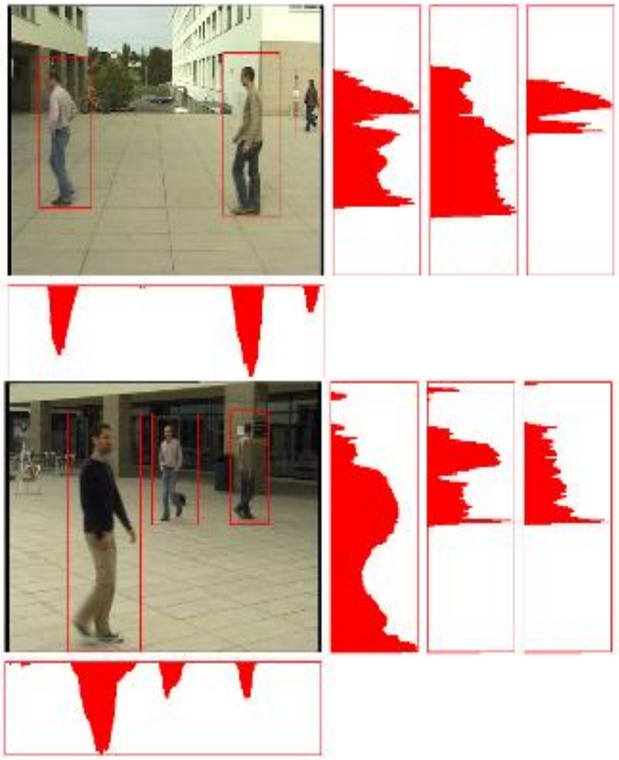


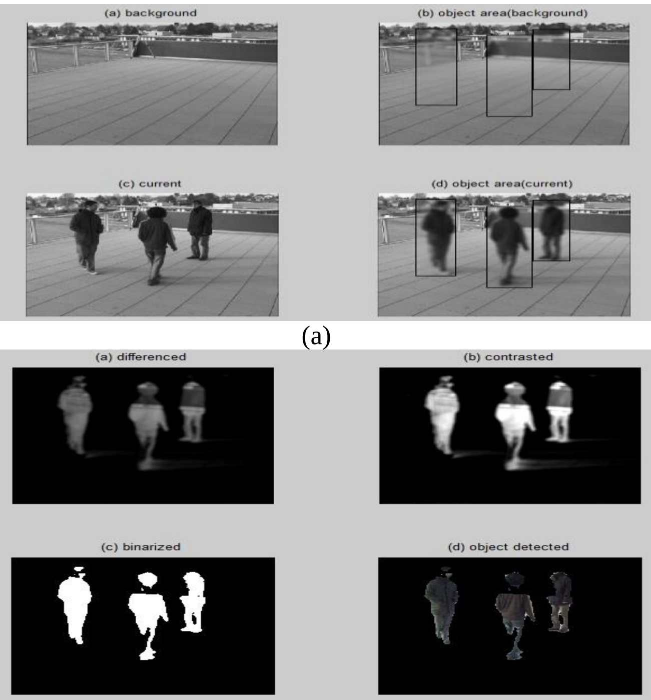
Figure 2: The examples of the detected areas and the multi-lateral histograms for multiple objects

The process to detect the multiple objects areas through the multi-lateral histogram is as in the following:

Step 1: From the horizontal histogram, process smooth filtering over the histogram.

Step 2: Obtain the threshold value using the average of the total histogram values.
Step 3: determine the region that the higher value than the threshold value exist continuously as the object region.
Step 4: Obtain the vertical histogram corresponding to the individual horizontal object regions and process smooth filtering
Step 5: Determine the areas that two corresponding horizontal multiple objects areas as the multiple objects areas.

From these objects areas that are determined using above processing steps for the multi-lateral histogram, Objects should be detected from the objects areas determined using the multi-lateral histogram to extract the keypoints that include the invariant features of the objects using SIFT. Objects are detected through next two steps.
Step 1: Obtain the difference image between the reference image and current image parts for only the object areas using the low-pass filtering.
Step 2: control the contrast to enhance the brightness difference between the objects and the neighbor reference background inside the objects areas.
Figure 3 shows the object detection results through the two steps. Figure 3(a) presents the results of Step 1 and 3(b) presents the results Step 2.



(a)

(b)

Figure 3: The results of the multiple objects detection

The entire document should be in Times New Roman or Times font. Other font types may be used if needed for special purposes. Type 3 fonts should not be used.
Recommended font sizes are shown in Table 1.

B. Shift Invariant Feature Matching

Lowe [4]'s SIFT algorithm has been used widely in fields of object detection and recognition and has largely four steps as follows:
Step 1: Scale space extrema detection: obtain the object's keypoints that invariant to size and orientations applying the DoG(Difference of Gaussians) function.
Step 2: Keypoint localization: Select the keypoints that are stable about brightness and locations.
Step 3: Orientation Assignment: assign the orientation information according to the brightness variation.
Step 4: Keypoint description: describes the invariant feature information that was calculated from the brightness variation size and directions of the neighbor keypoints.

Equation (1) presents the keypoint descriptor, that is the feature matrix, that includes the features extracted from SIFT.

$$[im, des, loc] = \text{SIFT}(\text{image}) \tag{1}$$

Where, *im* has the pixel values of the test image and *des* presents the matrix of the descriptor matrix, and *loc* has location, size, and orientation values of the all extracted keypoints.
Figure 4 shows the invariant feature keypoints of the multiple objects extracted from SIFT. Upper line shows the original objects areas and the keypoint extraction results for the object area includes the background and lower line shows the results that are concentrated to the inside area of the objects. Here, when the computing is concentrated to the inside of the object SIFT generates the more and the reliable points comparing with the case of the object area.

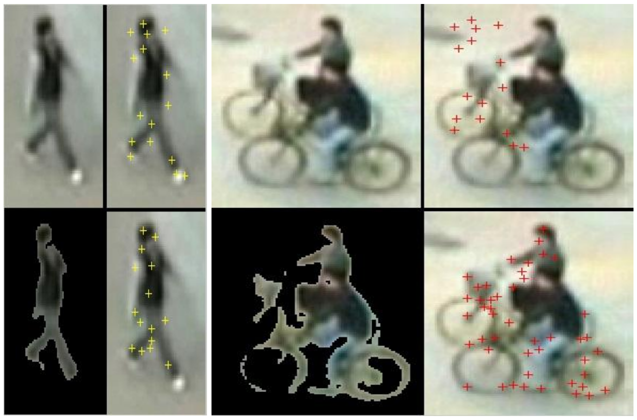


Figure 4: Results of the keypoints extraction for two cases of the object area and the concentrated to object's inside.

After the keypoints extraction of the multiple objects using SIFT algorithm, it is needed to compare the corresponding keypoints between the previous keypoints and the current ones for every objects using the *loc* information in equation (1) and it is called the keypoint location matching.
The distance between two corresponding keypoints is calculated by their inner product as equation (2).

$$\mathrm{d_{ij}} = \cos^{-1}(des_{\mathrm{Ri}} \circ des_{\mathrm{Cj}}) \qquad (2)$$

Where, des_{Ri} and $\mathrm{des_{Cj}}$ are *i*th descriptor in the previous frame and *j*th descriptor in the current frame.
The distance ratio that to be a reference of matching or mismatching is computed by equation (3).

$$dist\ ratio = \frac{the\ closest\ \ distance}{the\ second - closest\ \ distance} \qquad (3)$$

If this dist ratio is smaller than the adaptive value 0.8 presented by Lowe's experiment it is considered as matching, if not, it is considered as mismatching. Figure 5 shows the result of the keypoint matching for a walking man and in view of the matching line, we can find that there were a lot of mismatched keypoints nevertheless these are actually discriminated as the matched.

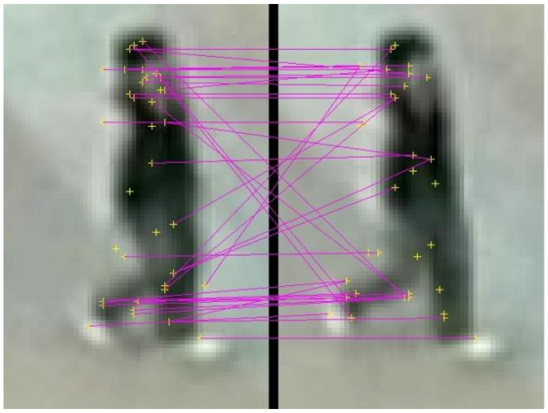


Figure 5: The results of the keypoint matching

These mismatched keypoints are classified using the location information. We analyzed the reason of the mismatching and established a matching reference that if the distance between the two corresponding keypoints is smaller than the allowable distance error it is determined as the matching case, if not, it is determined as the mismatching case. Because this matching reference uses the location distance information it is called as the location matching. By utilizing this reference the more exact matching is possible and ultimately it would be contributed to advance the more robust multiple objects tracking.

C. Object Window Adaption

In multiple objects tracking, the objects are changed into various sizes and directions in process of time and therefore the window sizes for the keypoint extraction and matching are required to be controlled according to the object size adaptively.
We propose a new method and it is a method that finds the extended rate of the changed size from the representative three matched keypoints of the two frame objects to be compared for matching. Figure 6 shows the geometrical figure of the proposed window adaption method.

Where, *W* and *H* are the width and the width of the current object area and *W/4* and *H/4* are the extended value for the moved object search and *x* and *y* are the locations of the starting point of the extended search, and *dx* and *dy* are offsets from the starting point to the starting point of the object area in the previous frame and the new object area in the current frame.

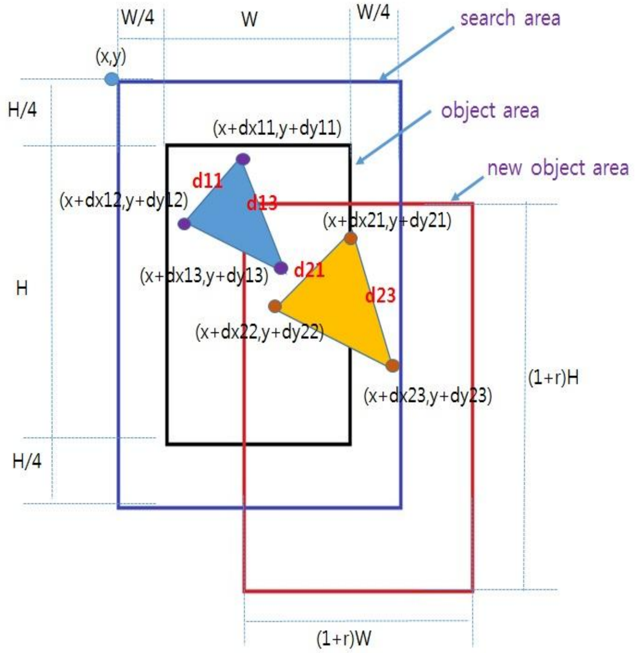


Figure 6: The geometrical figure of the proposed window adaptation

The processing steps are as follows:
Step 1: Select three location matched keypoints in the object area of the previous frame.

$$(x + d_{x11}, y + d_{y11}), (x + d_{x12}, y + d_{y12}), (x + d_{x13}, y + d_{y13}) \qquad (4)$$

Step 2: For the two pairs of keypoints, calculate the distance (Euclidean distance)

$$d_{12} = \sqrt{(d_{x11} - d_{x12})^2 + (d_{y11} - d_{y12})^2} \qquad (5)$$
$$d_{13} = \sqrt{(d_{x11} - d_{x13})^2 + (d_{y11} - d_{y13})^2}$$

Step 3: In the previous frame, select the search area by extending the object area by W/4 and H/4 in directions of the horizontal and vertical axis.
Step 4: In the current frame, select the search area that is identical location and size of the search area in the previous frame
Step 5: In the current search area, select three keypoints that is matched with the three keypoints selected in step 4.
Step 6: For the two pairs of keypoints in step 5, calculate the distance (Euclidean distance)

$$d_{21} = \sqrt{(d_{x21} - d_{x22})^2 + (d_{y21} - d_{y22})^2} \qquad (6)$$

$$d_{23} = \sqrt{(d_{x21} - d_{x23})^2 + (d_{y21} - d_{y23})^2}$$

Step 7: Calculate the extended object scaling rates between the previous and the current frames, r_{12} and r_{13} and then calculate the average scaling rate *r*.

$$r_{12} = \frac{d_{21}}{d_{11}}, r_{13} = \frac{d_{23}}{d_{13}}, r = \frac{r_{12}+r_{13}}{2} \qquad (7)$$

Step 8: Based on this average scaling rate, a new object window that includes the object in the current frame. The position of the new starting point is computed like next equations.

x position: $(x + d_{x21}) - rd_{x11}$ (8)
y position: $(y + d_{y21}) - rd_{y11}$

Step 9: Set the new object area that was controlled adaptively in the current frame. It has the size of $(1 + r)W$ and $(1 + r)H$ in the horizontal and vertical directions from the starting point.

Step 10: The windows of all objects are controlled adaptively using steps from 1 to 9.

These adaptive windows become the basis of tracking the multiple objects that appear in the next frame.owe[4]'s SIFT algorithm has been used widely in fields of object detection and recognition and has largely four steps as follows

D. Multiple Objects Tracking

Using the object feature matching that considers the location information the tracking procedure of the multiple objects is as follows.

Step 1: Store the keypoint feature information extracted in the previous frame.

Step 2: Test the matching rate between the feature data stored in the previous feature buffer and the current feature data extracted with identical size and location to the adaptive window selected in the previous frame.

$$\text{matched} = \begin{cases} \text{yes} & \text{if matched points } \geq 3 \\ \text{no} & \text{otherwise} \end{cases} \qquad (9)$$

Here, we adopted 3 points as the threshold because the adaptive window is determined using only three keypoints.

Next two cases are the references that objects are presented newly in the current frame or that they are disappeared.

Case 1: Newly presented case, any object area with features that doesn't belong to the previous feature. This object feature is added to the new previous buffer for next matching and tracking.

Case 2: Disappeared case, this is a case that the comparable features don't exist in the current feature buffer.

III. EXPERIMENTS AND RESULTS

First, we compared how do the number of keypoints generate in two cases of the object-included window area and the only object inside and table 1 show the results of two examples for one man and two bicycle men in Figure 4.

	One man	Two bicycle men
Object window area	17	18
Object inside	24	73

Table 1: Compared results of the extracted kepoints number for two cases of one man and two bicycle men

In table 1, the number of the extracted keypoints in case of the object inside is much more than ones in case of the object window area. This means that concentrating to the inside of the object that doesn't include the reference background is more effective in the matching and tracking.

Next, we determined what is the most fare value of the distance for discriminating the location matched or not in the keypoint and the distance in here means the pixel value.

Table 2 presents the numbers of the matched and mismatched keypoints according to the distance between the two comparable keypoints in the location matching for the one man case.

Distance	1	2	3	4	5	6	7	8	9
Matched	1	3	16	19	20	21	23	25	25
Mismatched	0	0	0	0	0	0	1	1	1

Table 2: The numbers of the matched and mismatched keypoints according to the distance between the two comparable keypoints in the location matching

Based on the results in table 2, we determined the fare distance range for the location mating lies within [3,6] and in this experiments using 3 or 5.

Lastly, using the proposed multiple objects tracking system the tracking performance is tested for three videos. These videos include various cases that multiple objects close to each other or are overlapped in a variety types of movements and these videos were utilized by the sample videos with the frames more than 300 frames that was provided in the reference [8]. Table 3 shows the tracking results for the three sample videos.

	Success rate	Failure rate
Sample 1	92	8
Sample 2	89	11
Sample 3	87	13

Table 3: The tracking results of the three different videos by using the proposed multiple objects tracking system

In table 3, the tracking rate was about 89.3 on average and especially, in case of sample 3, even if this video has much complicated movements such as closing, overlapping, crossing, and occluding, relatively high tracking rate is generated.

IV. CONCLUSIONS

A robust multiple objects tracking system was proposed that performs well in the complicated movements of multiple objects using the multi-lateral histogram, the SIFT algorithm, the location matching, and the adaptive windowing. And based on the several experimental results the proposed system has a robust characteristic to track multiple objects. In future, it is needed to advance the processing speed and it will be considered to implement the system using the parallel processing technique for the real-time tracking.

V. REFERENCES

[1]. R. Davis, "Lateral histogram for efficient object location: Speed versus ambiguity," Pattern Recognition Letters, Vol. 6, No. 3, 1987, 189-198.

[2]. Q. Jin, X. Tong, P. Ma, S. "Iris Recognition by New Invariant Feature nDescriptor," Journal of Computational Information Systems, Vol. 9, No. 5, 2013, 1943-1948.

[3]. H. Zhang, W. Gao, X. Chen, D. Zhao, "Object Detection using Spatial Histogram features," Image and Vision Computing, Vol. 24, No. 4, 2006, 327-341.

[4]. D. G. Lowe, "Object Recognition form Local Scale-Invariant Features," Proc. of the International C onference on Computer Vision, 1999, 1150-1157.

[5]. K. Mikolajczyk, C. Schmid, "A Performance Evaluation of Local Descriptors," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 10, 2005, 1615-1630.

[6]. H. Bay, T. Tuytelaars, L. V. Gool, "SURF: Speeded Up Robust Features," Proc. of the Ninth European Conference on Computer Vision, 2006, 404-417.

[7]. E. Rafajlowicz, "Improving the Efficiency of Computing Defects by Learning RBF Nets with MAD Loss," International Conference on Artificial Intelligence and Soft Computing, 2008, 146-153.

[8]. Ecole Polytechnique, Computer Vision Laboratory, http://cvlab.epfl.ch/data/pom/.