American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

Unsupervised Grouping of Local Components for Object Segmentation

Mohammad Khairul Islam [a*], Farah Jahan [b], Joong Hwan Baek [c], Seung-Jun Hwang [d]

[a] *Department of Computer Science & Engineering, University of Chittagong, Chittagong-4331, Bangladesh.*
[b] *Department of Information and Telecommunication Engineering, Korea Aerospace University, Goyang-city, Gyeonggi-do, 412-791, South Korea.*
[a] *E-mail: mkislam@cu.ac.bd*
[b] *Email: jhbaek@kau.ac.kr*

Abstract

In this paper, we propose a novel object segmentation method for image understanding. Due to challenges such as variations in object size, orientation, illumination etc. object segmentation is extraordinarily difficult task in the domain of image understanding. It is well-founded concept that a small portion of the pixel set in an image contributes most in image description. Based on this concept, we hypothesize that an image consists of many components or parts each of which represent a small local area in the image and they are very meaningful in visual perception. For object segmentation, we propose spatial segmentation method on such prototypical components of images. Given an image this segmentation method acts as coarse to fine search for object(s) iteratively. The proposed method demonstrate its excellence in localizing objects in various complex backgrounds, multiple objects in a single image even if they have variation in size, orientation, lighting conditions etc. The detection efficiency of our object detector on our self-collected image set which consists of images from six different object categories climbs up to 93% in average.

Keywords: SUFT; object detection; Color histogram.

1. Introduction

An image is usually addressed by the contents such as objects that are present in the image. Knowing the identity of the objects help us to understand the message carried out by the image.

---

*Corresponding author.

However, the object in real world video surveillance system may not occupy the whole image area; rather a cluttered background may occupy a large portion of the image. Therefore, the focus tends to be on object segmentation rather than only classification.

Perhaps the simplest class of approaches to object segmentation is that of sliding window methods. But this method is computationally very expensive and not invariant to window sizes, and rotations. In [1], the authors use exhaustive search by sliding window while authors in [2] treat object segmentation as a set-to-set matching problem between segments. Dalal and his colleagues [3] propose human detector using histograms of oriented gradients (HOG) features in local windows. Viola and Jones [4] build object detectors that incorporated features from large windows surrounding the object, as well as the output of boosting from other objects. Maji and his colleagues [5] propose a max-margin hough voting method with SVM to detect objects. They present a discriminative Hough transform based object detector where each local part casts a weighted vote for the possible locations of the object center. Part based representation forms the basis of some computational theories of object segmentation [6] where object parts are local groups of common primitive features such as edge fragments [7][8]. Due to huge success in part-based concept in object segmentation we adapt this concept in our proposed method. Given a test image, our object segmentation approach aims to determine the presence of any object of interest in the image, locate its area. In order to make the article more self-contained we elaborately discuss our object proposed method in section 2. We present our experimental results in section 3. And finally in section 4, we summerise our work and foresight.

2. Proposed Approach

Our proposed approach consists of several tasks such as Interest point detection which aims to find locally important patches, which go through a clustering technique to form candidate segments. Candidate segments are tested by extracting features from the local patches inside the candidate area. Segmentation and candidature test is iteratively performed to get optimal result. Fig. 1 dipicts our processing steps which are described in detail in the subsequent sections.

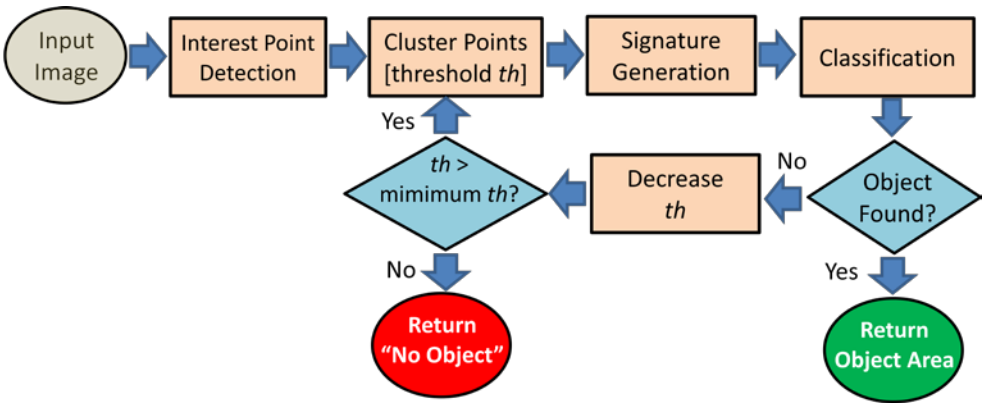


Figure 1: Block diagram of object segmentation algorithm.

2.1. Interest Point Detection

A stable interest point in an image is a point which, in general, has important role in constructing a visual structure of an imagery. So, interest points should be reliably computed with high degree of reproducibility such that the regions centered on those points are stable in the image domain under local and global perturbations including perspective transforms, brightness changes. Considering this situation, we adapt FAST [9] corner detector to locate interest points in object images. In FAST detector, a pixel of corner candidate is tested by considering a circle of 16 pixels around it. In this method, a point p is classified as a corner point if it is brighter or darker than n contiguous pixels around the point. The inventor of FAST chooses n to be 12 because it admits a high-speed test which can be used to exclude a very large number of non-corners. The test examine only the four pixels at 1, 5, 9 and 13 (of the four compass directions). If a point p is a corner, then at least 3 of the 4 must all be brighter or darker than p by a threshold. If neither of these is the case, then p cannot be a corner. Given a pixel p with intensity I_p, each pixel on the circle $C = \{1, \dots, 16\}$ can have one of the three states mentioned in Eq. (1) give $c_j \in C$.

$$S_{c_j} = \begin{cases} d & I_{c_j} \le I_p - t \text{ (darker)} \\ i & I_p - t < I_{c_j} < I_p + t \text{ (identical)} \\ b & I_p + t \le I_{c_j} \text{ (brighter)} \end{cases} \quad (1)$$

Suppose P is the set of pixels in all training images, and L_p is label of a pixel p which is true if p is a corner and false otherwise. The entropy of L for the pixel set P is calculated as Eq. (2).

$$H(P) = (h + \bar{h})\log_2(h + \bar{h}) - h\log_2 h - \bar{h}\log_2 \bar{h} \quad (2)$$

Here, $h = |\{p|L_p \text{ is true}\}|$, and $\bar{h} = |\{p|L_p \text{ is false}\}|$. h, and $\bar{h}$ are respectively the number of corners and non-corners. The choice of C then gives the information gain G calculated using Eq. (3).

$$G = H(p) - H(p_d) - H(p_s) - H(p_b) \quad (3)$$

We select C having most information. The subset C_b is selected to partition p_b into $p_{b,d}$, $p_{b,s}$, $p_{b,b}$, C_s is selected to partition p_s into $p_{s,d}$, $p_{s,s}$ and $p_{s,b}$ and so on. The process ends when this subset has the same value of k_p, i.e. they are either all corners or non-corners. Fig. 2 shows examples of corner detection. A cross-shaped ('x') symbol shows a corner point.

(A) (B)

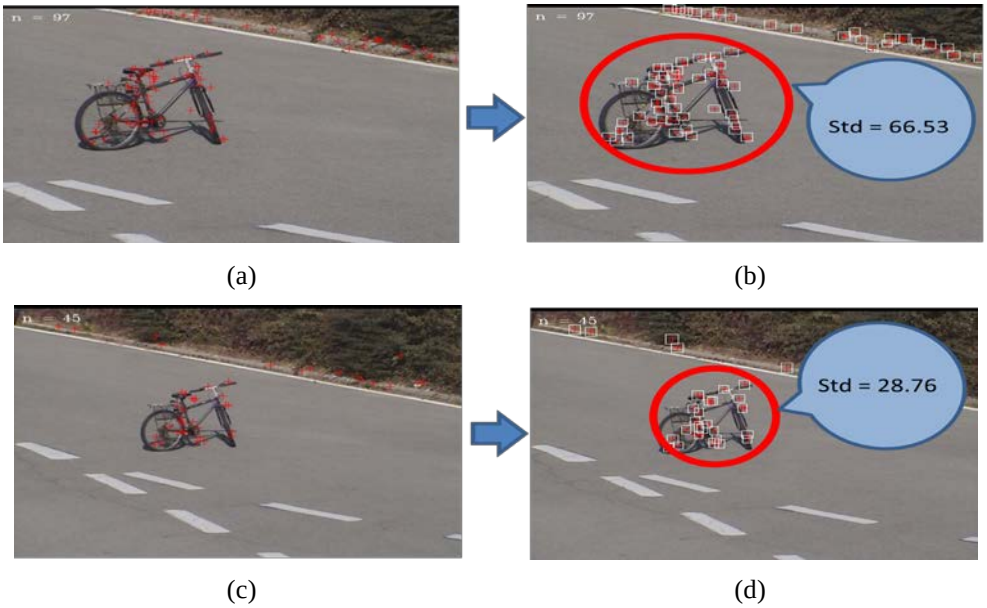
Figure 2: Result of key point detection in images of two different scales using FAST detector

2.2. Spatial Clustering

In our segmentation method, we use the neighborhood attribute of interest points. We use Euclidian distance metric between the points to cluster the points. Let us consider a set of points, $P = \{p_1, p_2, \dots, p_n\}$ where $p_i = (x_i, y_i)$, and $i = \{1,2, \dots, n\}$. Given two points p_i and p_j, the Euclidean distance between them, denoted by d_{ij}, can be defined as Eq. (4).

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad (4)$$

From the set of points P, a subset of them belong to one cluster if each point in the subset finds at least one point, except itself, in the subset that has smaller Euclidian distance than some threshold. Given a pivot point p_v, a subset can be created by finding its neighbor points from the set of points P. Mathematically, neighbors of a point p_v, denoted by N_v, can be defined as $N_v = \{p_k\}$, where $k \in \{\{1,2, \dots, n\} \backslash \{v\}\}$ and Euclidian distance, d_{vk}, between points p_v and p_k holds $d_{vk} < \text{theshold}$. This threshold is dependent on the inter-distance of local patches of object. Larger the distance, larger threshold is need to be set. Fig. 1(b) and 3(d) show standard deviation between local patches of object areas. The pseudo-code of the clustering our clustering process given after Fig. 3.



(a) (b)

(c) (d)

Figure 3: Standard deviation between interest points in object area. Small squares in the right represent local patches, the large circle shows the object area, and the oval callout shows the standard deviation between the patches inside the object area.

ALGORITHM : SPATIAL CLUSTERING

theshold ←75

counter ← 1

FOR i in {1,2, … , n}

LABEL (p_i) ← NULL

FOR v in {1,2, … , n}

IF LABEL (p_v) = NULL

seed ← p_v

LABEL(p_v) ← counter

END

FOR j in {{1,2, … , n}\{v}}

IF DISTANCE(p_v, p_j) < theshold

LABEL(p_j) ← LABEL(p_v)

END

END

counter ← counter+1

END

return LABEL

The very important parameter in this clustering process is the threshold. The number of clusters and points belonging to same cluster are directly dependent on this parameter. Larger the threshold fewer the number of clusters and smaller the sizes of the clusters i.e. smaller the number of points in a cluster. Since each interest point in the image has very important role in describing the appearance of the scene or object, the points belonging to a cluster should be either from the object of interest or the context. If interest points from both context and object of interest groups together, this clustering will increase false alarm. Fig. 4 shows example of spatial grouping results using two different threshold and relevant false alarms. Here we represent a cluster using a minimum rectangle holding the points belonging to the cluster. As we reduce the threshold, clusters become more pure i.e. either dominated by background or by object. Fig. 5 shows examples of segmented area come out from diffent iterations and Fig. 6 represents required number of iterations in order to find object segments.

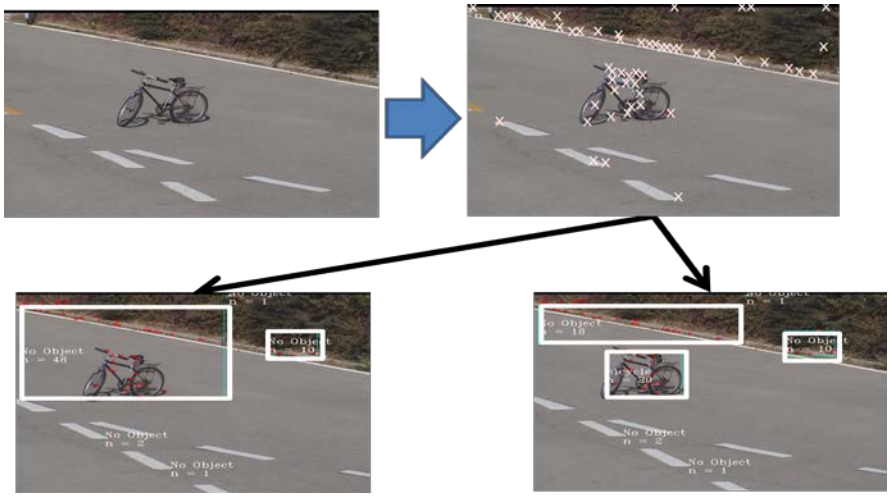


Figure 4: Examples of candidate regions obtained from spatial clustering using different threshold. Top left: input image, top right: detected interest points (small filled circles) in the image, bottom left: spatial grouping using threshold = 75 and bottom right: threshold = 65.

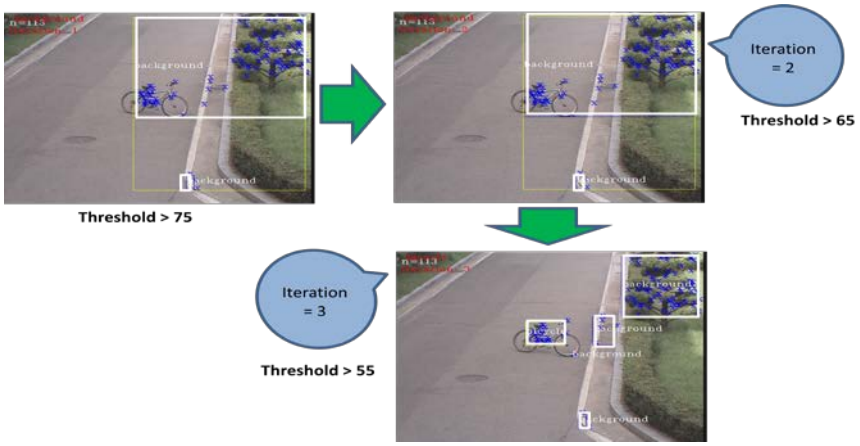


Figure 5: Step-by-step outcomes in object segmentation. Here, threshold means the threshold distance in grouping local patches.

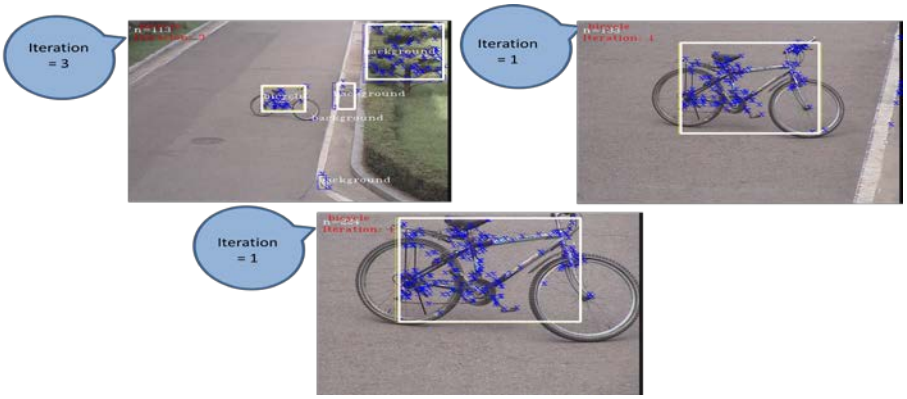


Figure 6: Number of iterations to find out the object area. The number inside oval callout represents the required iteration to local the object.

2.3. Signature Generation

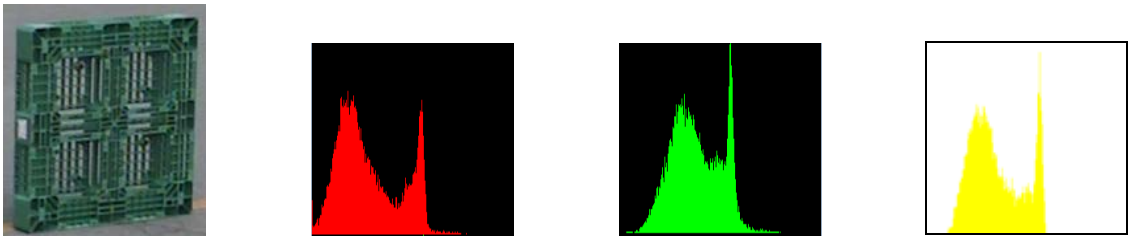
We aim to strengthen the discriminative power of our feature using heterogeneous attributes of local patches. Considering acceptable computational complexity we use sum of wavelet responses and weighted color statistics in our approach.

Speeded Up Robust Feature (SURF): In this method, an integral image is constructed for fast computation [10]. Given an input image $I(p, q)$ of resolution mxn where (p, q) is spatial position of a pixel, an integral image I_Σ is calculated as in Eq. (5).

$$I_\Sigma = \sum_{p=1}^{p\leq m} \sum_{q=1}^{q\leq n} I(p, q) \quad (5)$$

For the extraction of the descriptor, the first step consists of constructing a square region centered around the interest point and oriented along the dominant orientation. An interest region is split into 4x4 square sub-regions with 5x5 regularly spaced sample points inside. Haar wavelet responses d_p and d_q weighted with a Gaussian kernel centered at the interest point are calculated. Sum of responses at P for d_p, $|d_p|$, d_q, and $|d_q|$ creates a feature vector of 16x4 or 64 elements and the sum of responses d_p, and $|d_p|$computed separately for $d_q < 0$ and $q > 0$ and similarly sum of d_q, and $|d_q|$ creates a feature vector of length 128.

Color Feature: Color histograms are widely used for describing the color of objects [11] For digital images, a color histogram represents the number of pixels that have colors in each of a fixed list of color ranges that span the image's color space made up of a set of possible colors. In our experiment, we construct a color descriptor for each key point. In this case, a 16x16 window around a keypoint is considered as a patch. The color values are calculated from the patch and put into a n-bin histogram. Any value in the range 0 to $255/n$ is added to the first bin, $(255/n) + 1$ to $\left(\frac{255}{n}\right) * 2$ is added to the next bin and so on. Fig. 7 displays color histograms obtained from a real image.



(a) (b) (c) (d)

Figure 7: Color Histograms. (a) A color image. (b), (c), and (d) are intensity histograms of R, G, and B planes respectively obtained from the image (a).

Local Descriptor Generation: In our previous research [12] we integrated texture and color-based descriptors by concatenating them together. In such case, each element of the newly generated combined feature has same weight. But, every feature type has different discriminative power as a signature.

Keeping this hypothesis in our mind we analyze our feature set and discover their relative strengths in representing a same problem domain. After discovering their relative strength in a given dataset or problem we weight individual feature type by their strength. In our research, we represent each sample images in a training dataset by using a single feature type, then find correct classification rate as its strength. For instance, suppose we have the above two types of features and their correct classification ratios are a and b respectively. If we consider their performances combinedly then they have strengths $\frac{a}{a+b}$, and $\frac{b}{a+b}$ respectively. We use these strengths as weight of respective descriptor before combining them. Mathematically, we denote the combination for the above descriptors as Eq. (6).

$$D = \left(\frac{a}{a+b} * (t_1, t_2, \dots, t_m), \frac{b}{a+b} * (c_1, c_2, \dots, c_n)\right) \quad (6)$$

Global Signature Generation: We implement Bag of Words model for generating global signature. In this method a codebook or dictionary is generated using k-means clustering [13] over all the local descriptors. Thus, each patch in an image is mapped to its closest codeword to generate a signature. A global signature is a frequency histogram that counts number of patches closest to each codeword into a corresponding bin.

2.4. Classification

Given a signature, we prepare object models using Naïve Bayes algorithm. Given a hypothesis h and data D which bears on the hypothesis. The category or class of D among a set of hypotheses or classes, defined as $H = \{h_i | i = 1,2, \dots, n\}$, is calculated as the maximum of posteriors of all hypotheses as Eq. (7).

$$\text{classify}(D) = \underset{h_i \in H}{\text{argmax}}\, p(h) \prod_{i=1}^{n} p(D_i|h) \quad (7)$$

3. Experimental Results

3.1 System Setup

We capture images from 6 categories such as Bicycle, Chair, Box, Ladder, Luggage, and Pallet using 2 PTZ cameras. For each category, we capture images in 8 different views, and 3 different zoom factors. Thus we have a total of 2×3×6×8 or 288 images each with resolution of 640×480. Fig. 8 shows a few examples of images from the data set. Our application is developed in Microsoft Visual C++ 2005 and uses OpenCV2.0 library. We use a desktop PC containing Intel®Core™2CPU 1.87GHz, 2 GB of RAM.

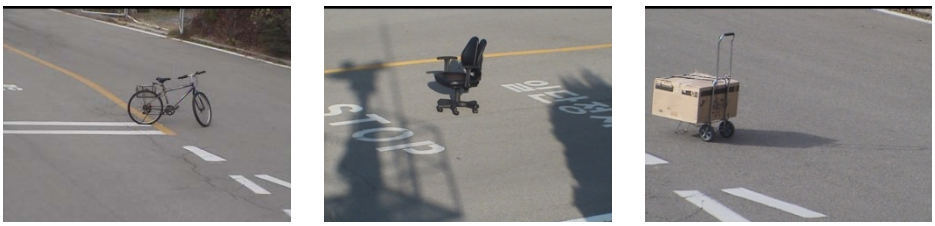


Figure 8: Examples of self-collected images.

3.2 Performance Metrics

Precision-recall is calculated using the following confusion matrix:

	Actual positive	Actual negative
predicted positive	True Positive (TP)	False Positive (FP)
predicted negative	False Negative (FN)	True Negative (TN)

TP is the number of true positives (target objects that were correctly detected) while FN is the number of false negatives (target objects that were not detected).

FP is the number of false positives (detected objects that were not correct), while TN is the number of true negatives (candidate detections that were correctly not detected, because they do not map to a true target object). Rewriting recall and precision in terms of the confusion matrix results as Eq. (8) and Eq. (9).

$$\text{recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{8}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{9}$$

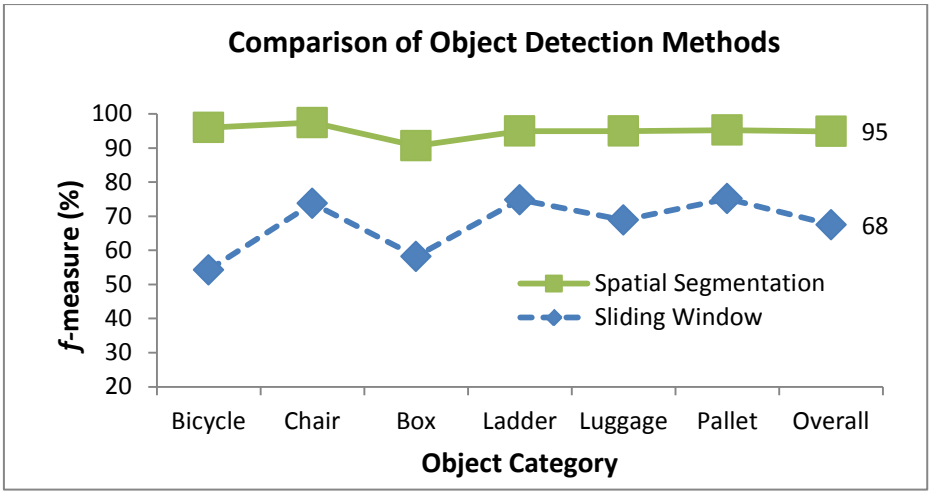
To allow easier comparison among different detection systems, particularly because of this trade-off, it is useful to have a single score summarizing both recall and precision. $\text{f} - \text{measure}$ is sometime used to overcome this trade-off. The $\text{f} - \text{measure}$ of a set of detections is defined as the harmonic mean of the recall and precision as calculated in Eq. (10).

$$\text{f} - \text{measure} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall+precision}} \tag{10}$$

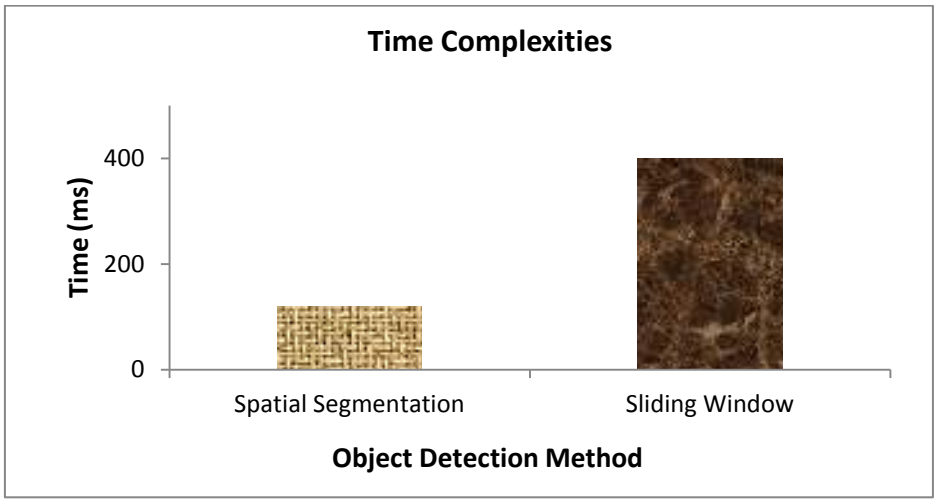
3.3 Experimental Findings

In sliding window method, we slide a window on a given input image and select a window location which best matches an object of interest. For faster computation we center each window at each interest point. The size of the window varies from one object category to other. For example, in our experiment, average resolution of bicycle, chair, box, ladder, luggage, and pallet are 231x213, 132x188, 154x197, 181x260, 98x149, and 182x228 respectively. So, we set size of a sliding window to 163x205 which is the average of these six classes of objects. Fig. 9(a) compares object segmentation efficiency of two object detectors. Fig. 9(b) compares computation time for the two object segmentation methods.

Fig. 10 represents the accuracy of the proposed object detector for bicycle, chair, box, ladder, luggage and pallet images.

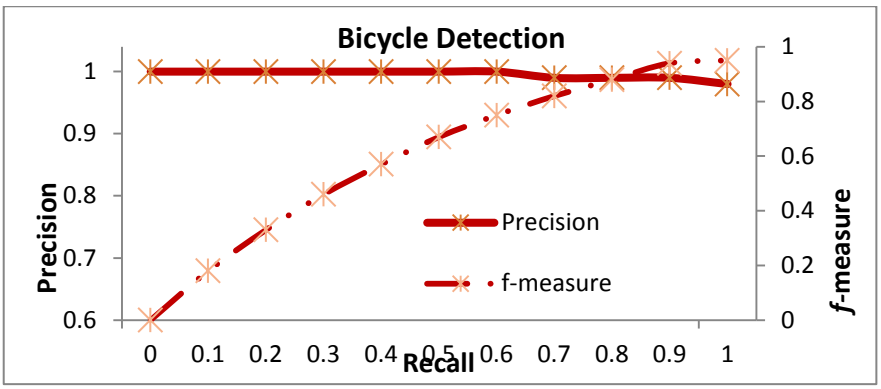


(a)

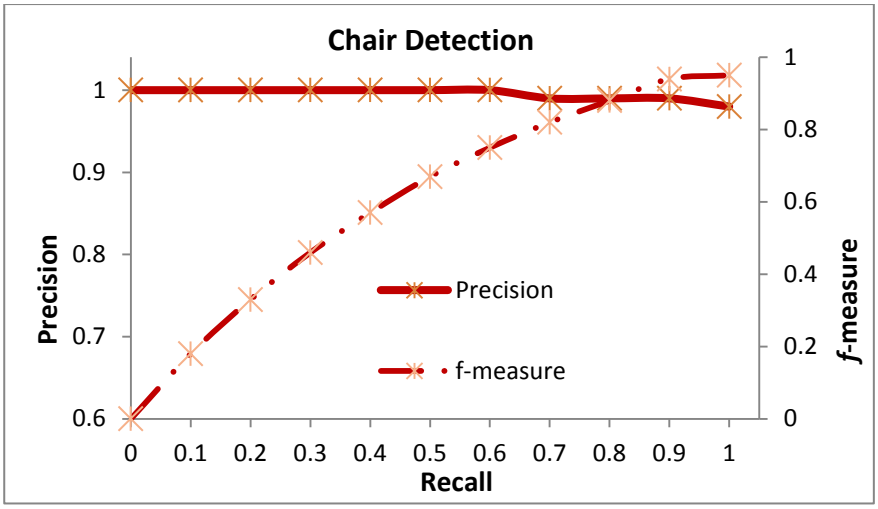


(b)

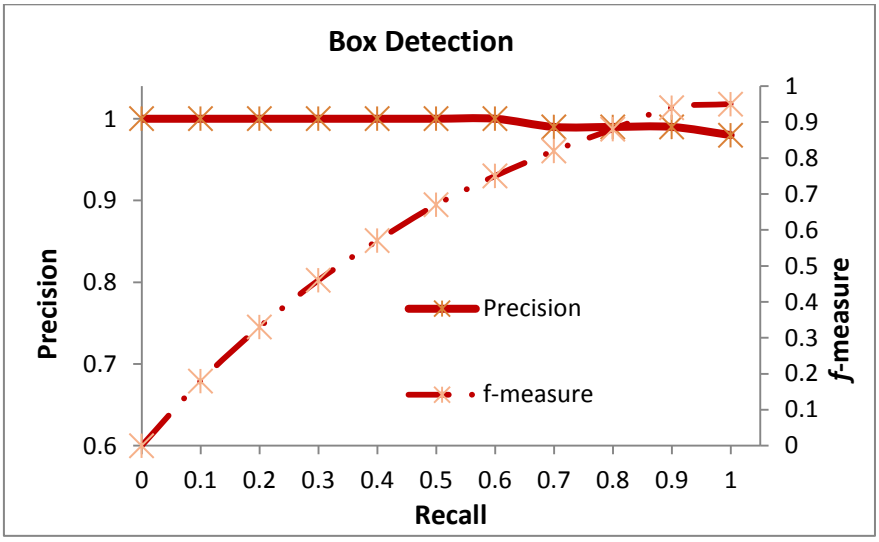
Figure 9: Efficiency comparison between two detectors. (a) Compares f-measure and (b) compares time complexity between sliding window method and proposed method.



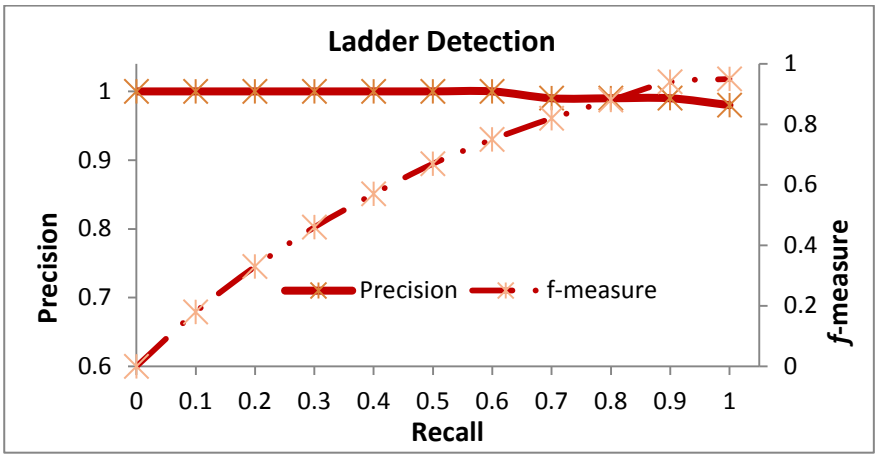
(a)


Chair Detection
Precision
f-measure
Recall
f-measure

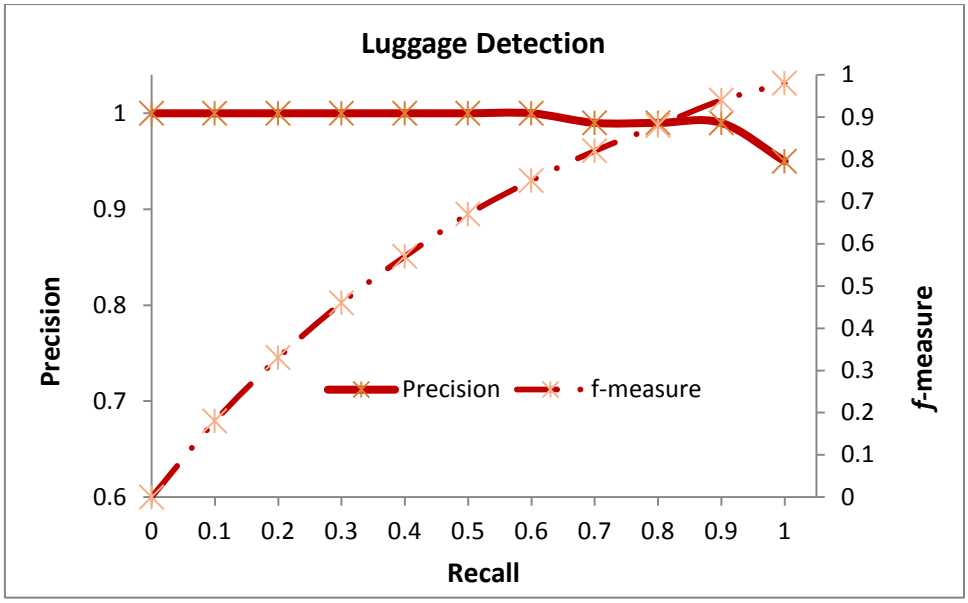
(b)


Box Detection
Precision
f-measure
Recall
f-measure

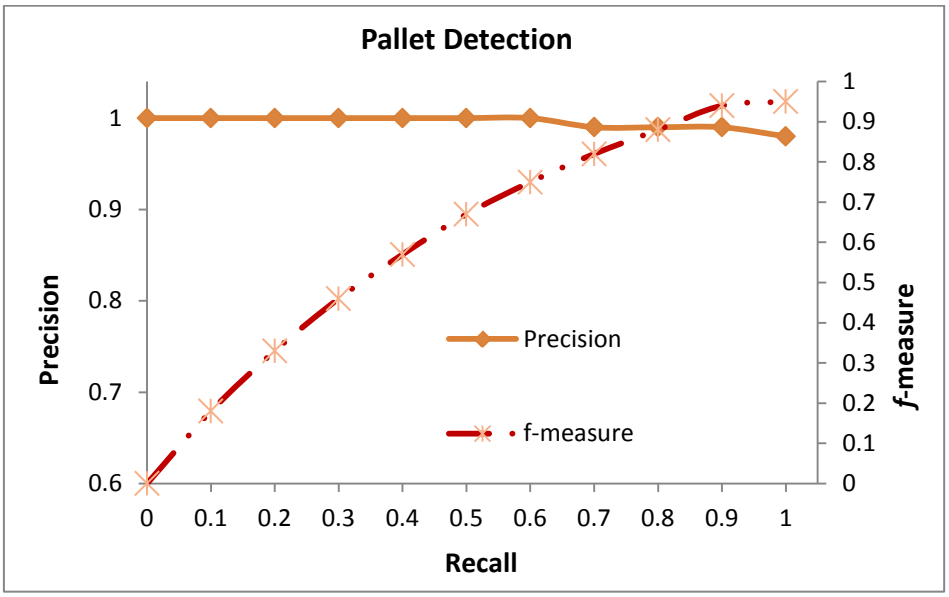
(c)


Ladder Detection
Precision
f-measure
Recall
f-measure

(d)



(e)



(f)

Figure 10: From (a) to (f): recall-precision and corresponding f-measure curves for Bicycle, chair, box, ladder, luggage and pallet detection.

4. Conclusions

Object segmentation is performed to segment object area in order to increase classification rate even when an object is placed in a cluttered background. In order to resolve the difficulties with those technologies we propose new technologies and also extend some cutting edge technologies depending on their outcomes. We propose new local feature which is extracted from local interest regions.

We device a novel method for object localization that shows excellent performance. Object segmentation is based on the concept of object parts and spatial co-existence nature of these parts. We explored their co-existence properties and propose a spatial segmentation based object localization method. For our self collected dataset, 93% of the time, object is accurately located. The experiments in the works in done one specific data set. It might not work so good in natural dataset. In future, we would like to evaluate our approach on more natural datasets, since this is more likely to reveal the advantage of having models with a large number of parts.

Acknowledgments

This study was conducted with the assistance of the Korea Aerospace University Technical Research Center of the next generation broadcast media by the GRRC(Gyeonggi-do Regional Research Center) program.

Refereces

[1] J. Shotton, A. Blake, and R. Cipolla, "Multi-scale categorical object recognition using contour fragments," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 30, no. 7, pp. 1270–1281, 2008.

[2] L. Zhu, Y. Chen, Y. Lin, C. Lin, and A. Yuille, "Recursive segmentation and recognition templates for 2D parsing," *In Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, pp. 1985-1992, 2008.

[3] N. Dalal, and B. Triggs, "Histograms of oriented gradients for human detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, USA, pp. 886-893, 20-26 June, 2005.

[4] P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Hawaii, vol. 1, pp. 511-518, 2001.

[5] S. Maji, and J. Malik, "Object detection using a max-margin hough transform," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* , Miami, FL, USA, pp. 1038, 2009.

[6] I. Biederman, "Recognition-by-components: A theory of human image understanding," Psychological Review, vol. 94, no. 2, pp. 115-147, 1987.

[7] Y. Amit, and D. Geman, "A computational model for visual selection," *Neural Computation*, vol. 11, pp. 1691-1715, 1999.

[8] D. Roth, M. H. Yang, and N. Ahuja, "Learning to recognize 3D objects", *Neural Computation*, vol. 14, no. 5, pp. 1071-1104, 2002.

[9] E. Rosten, and T. Drummond, "Machine learning for high-speed corner detection," *European Conference on Computer Vision (ECCV)*, Graz, Austria, pp. 430-443, 11-14 May, 2006.

[10] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, "SURF: Speeded Up Robust Features," *Journal of Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346-359, 2008.

[11] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking across multiple cameras with disjoint views," *Proceeding of the IEEE International Conference on Computer Vision (ICCV)*, Nice, France, pp. 952-957, 13-16 October, 2003.

[12] M. K. Islam, F. Jahan, J. H. Min, and J. H. Baek, "Object classification based on visual and extended features for video surveillance application," *Proceedings of the 8th Asian Control Conference (ASCC)*, Kaohsiung, Taiwan, pp. 1398 – 140, May 15-18, 2011.

[13] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering objects and their location in images," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, San Diego, USA, pp. 370-377 , 2005.