Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Performance measures for object detection evaluation

Bahadır Özdemir[a], Selim Aksoy[a,*], Sandra Eckert[b], Martino Pesaresi[b], Daniele Ehrlich[b]

[a] Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey
[b] Institute for the Protection and Security of the Citizen, European Commission, Joint Research Centre, 21020 Ispra (VA), Italy

ARTICLE INFO

Article history:
Available online 27 October 2009

Keywords:
Performance evaluation
Object detection
Object matching
Shape modeling
Multi-criteria ranking

ABSTRACT

We propose a new procedure for quantitative evaluation of object detection algorithms. The procedure consists of a matching stage for finding correspondences between reference and output objects, an accuracy score that is sensitive to object shapes as well as boundary and fragmentation errors, and a ranking step for final ordering of the algorithms using multiple performance indicators. The procedure is illustrated on a building detection task where the resulting rankings are consistent with the visual inspection of the detection maps.

1. Introduction

Performance evaluation of pattern recognition and computer vision systems has always received significant attention (Thacker et al., 2008). Studies that characterize the theoretical performance (Haralick, 1996; Liu et al., 2005) as well as empirical comparisons (Phillips and Bowyer, 1999; Flynn et al., 2001; Christensen and Phillips, 2002; Wirth et al., 2006) of different methods can be found in the literature. Some of these studies aim to evaluate the performance of generic classification or clustering techniques on a wide range of ground truth data sets (Asuncion and Newman, 2007), while some concentrate on specific problems with data sets tailored for the corresponding applications. Such efforts have also been coordinated in several performance contests that provide benchmark data sets and quantitative evaluation criteria in the recent years (Aksoy et al., 2000; Smeaton et al., 2006; Alparone et al., 2007; Pacifici et al., 2008,).

This paper is based on our work on developing new performance measures for object detection evaluation and the application of these measures to a building detection task as part of the algorithm performance contest that was organized within the 5th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS 2008, http://www.iapr-tc7.org/prrs08). The contest was organized jointly by the International Association for Pattern Recognition (IAPR) Technical Committee 7 (TC7) on Remote Sensing and the ISFEREA Action of the European Commission, Joint Research Centre, Institute for the Protection and Security of the Citizen.

An important goal of pattern recognition methods developed for the analysis of data collected from satellites or airborne sensors used for Earth observation is to improve human life by providing automatic tools for mapping and monitoring of human settlements for disaster preparedness in terms of vulnerability and risk assessment, and disaster response in terms of impact assessment for relief and reconstruction. In this perspective, optimization of the automatic information extraction about human settlements from new generation satellite data is particularly important. The contest contributed toward this direction by focusing on automatic building detection and building height extraction. A QuickBird data set with a reference map of manually delineated buildings was provided for the evaluation of building detection algorithms. Similarly, a stereo Ikonos data set with a highly accurate reference digital surface model (DSM) was supplied for comparing different DSM extraction algorithms. Aksoy et al. (2008) presented the initial results from nine submissions for the building detection task and three submissions for the DSM extraction task.

In addition to providing challenging data sets from new generation sensors, the contest also aimed to identify useful performance measures for these tasks. In particular, six different measures were used in (Aksoy et al., 2008) to evaluate the building detection performance. An important observation was that no single algorithm stood out as the best performer with respect to all performance measures. Furthermore, different criteria favored different algorithms, and it was not always possible to provide an intuitive explanation of the rankings produced by different measures. Similar observations have been discussed in the literature where the evaluation of building detection algorithms in particular and object detection algorithms in general are still open problems.

This paper presents a new evaluation procedure for characterizing the performance of object detection algorithms where the

* Corresponding author. Tel.: +90 312 2903405; fax: +90 312 2664047.
E-mail addresses: bozdemir@cs.bilkent.edu.tr (B. Özdemir), saksoy@cs.bilkent.edu.tr (S. Aksoy), sandra.eckert@jrc.it (S. Eckert), martino.pesaresi@jrc.it (M. Pesaresi), daniele.ehrlich@jrc.it (D. Ehrlich).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.10.016

objects in the reference map and the algorithm output are represented using masks with arbitrary shapes. We study the evaluation process in three stages. The first stage involves a matching algorithm that finds correspondences between the reference objects in the ground truth and the objects in an algorithm output. An important advantage of the proposed method is that it allows one-to-many and many-to-one correspondences whereas most of the methods in the literature can only handle one-to-one matches between the reference and output objects. The second stage includes performance measures for the quantification of the detection accuracy using the matches found in the previous stage. The proposed measure is sensitive to the shapes of the objects as well as the boundary errors and fragmentation errors as opposed to the common practice of only counting the overlapping pixels for the matching objects. The third stage uses multi-criteria ranking to produce a final ordering of the algorithms using a combination of multiple measures. The proposed evaluation procedure can be used to evaluate the accuracy of any object detection algorithm when the output consists of multiple objects and when the shapes of these objects and the quantification of the geometrical errors in their detection are important.

The rest of the paper is organized as follows: Section 2 summarizes the related work on object detection evaluation, and discusses how the proposed procedure differs from other approaches. Section 3 presents the motivations behind the selection of the particular data set used. Section 4 describes the proposed evaluation procedure in detail, and summarizes two other methods used for comparison. Section 5 introduces the building detection algorithms used in the experiments. Section 6 presents the application of the object detection performance evaluation procedure on the building detection results, and Section 7 provides the conclusions.

2. Related work on object detection evaluation

One way of studying the evaluation of object detection algorithms is to represent the results in a pixel-based classification setting where the detection corresponds to the labeling of image pixels. The most widely adapted strategy for reporting the performance of classification algorithms is to use error rates computed from confusion matrices. Pixel-based evaluation is valuable for applications such as cadastral map updating, change detection, target detection, and defect detection when identifying several pixels on the objects of interest is sufficient so that an expert can manually inspect and correct the algorithm outputs for the final production. However, the confusion matrices computed by pixel-based comparison of reference and output maps cannot effectively characterize the geometric accuracy of the detection when the goal of an algorithm is to produce a full delineation of the objects of interest. Bruzzone and Persello (2008) suggested to compute such rates separately from pixels inside the objects and from pixels on the boundaries of the objects. It is also possible to make a distinction between isolated false alarms, false alarms close to a target, and clusters of false alarms by comparing morphologically dilated versions of the reference maps and the output detection maps (Meur et al., 2008).

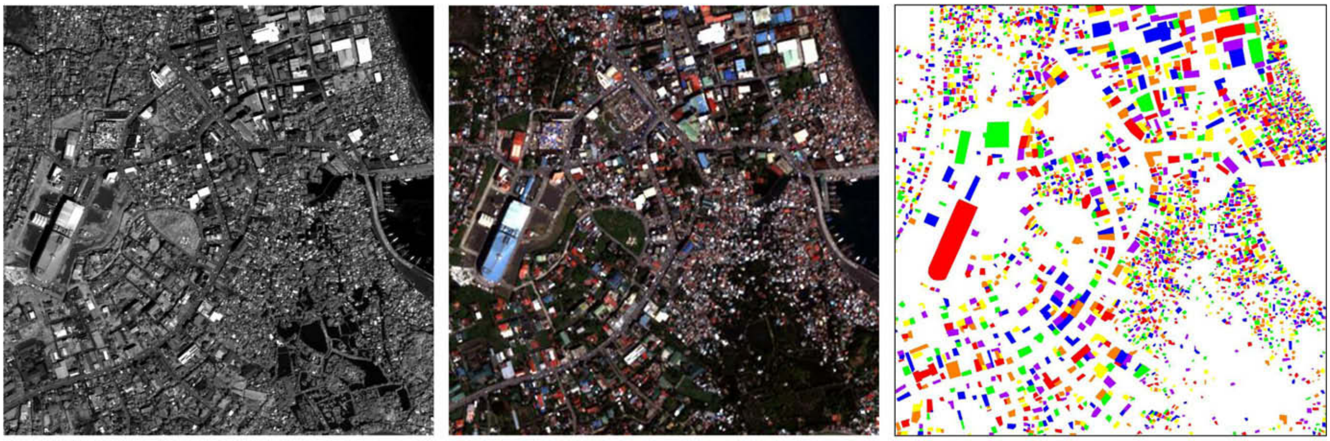
Object-based performance measures try to overcome the limitations of pixel-based evaluation. The evaluation procedure can be studied as a combination of a matching problem for finding correspondences between reference and output objects, and an accuracy assessment problem for quantifying the quality of these matches. The most common method for finding correspondences is to assign an output object to the reference object that has the largest number of overlapping pixels with this object (Huang and Dom, 1995; Bruzzone and Persello, 2008). This method finds one-to-one matches between the reference and output objects. To be able to handle over-detections where more than one output object correspond to a reference object, and under-detections where more than one reference object correspond to an output object, the maximum overlap criterion can be relaxed to allow all overlaps above a certain threshold (Hoover et al., 1996; Mariano et al., 2002; Ortiz and Oliver, 2006). Alternatively, Jiang et al. (2006) used maximum-weight bipartite graph matching to find optimal one-to-one matching between the reference and output objects where the weights correspond to overlaps among the objects. Martin et al. (2004) used a similar minimum-weight bipartite graph matching procedure to find a one-to-one matching between the boundary pixels of two segmentation maps where the weights correspond to pixel distances in the image plane. Liu and Haralick (2002) also used a similar graph matching approach for finding correspondences between pixels in edge maps for edge detection evaluation. The over-detections and under-detections can be important factors in the accuracy assessment process when a very large number of objects are considered (e.g., the ground truth for the test site for the building detection task studied in this paper contains 3064 objects). The evaluation procedure proposed in this paper can handle one-to-one, one-to-many, and many-to-one matches while maximizing the amount of overlap between the matching objects.

After the correspondences are established, the accuracy of the detection can be computed from the resulting matches. This accuracy is typically measured using the percentage of the matching pixels (Huang and Dom, 1995; Hoover et al., 1996; Mariano et al., 2002; Martin et al., 2004; Ortiz and Oliver, 2006; Jiang et al., 2006; Bruzzone and Persello, 2008). Unfortunately, measures that are based on pixel counts cannot be good indicators of the geometric accuracy of the detection, with the exception of (Martin et al., 2004) where the pixels participating in the counts are boundary pixels. To be able to handle fragmentations in the detections, Mariano et al. (2002) and Bruzzone and Persello (2008) proposed measures to penalize higher number of output objects participating in over-detections. Bruzzone and Persello (2008) also proposed a border error measure that counts the number of mismatching pixels between the boundaries of two objects. Furthermore, distance measures based on shape descriptors (e.g., Hausdorff distance, shape signatures, elastic matching) (Zhang and Lu, 2004) can also be used but such measures are often defined only for one-to-one matches. The performance measure defined in this paper is sensitive to the shapes of the objects, and can also quantify boundary and fragmentation errors.

Given all performance measures that can be based on pixel counts or object-based detection rates, a final task of interest is to rank the detection algorithms according to their overall performance. Most of the studies (Huang and Dom, 1995; Hoover et al., 1996; Mariano et al., 2002; Ortiz and Oliver, 2006; Jiang et al., 2006) conclude by providing an exhaustive table of individual scores for all measures and all algorithms. Bruzzone and Persello (2008) proposed to use a genetic algorithm for multi-objective optimization for finding a set of Pareto optimal solutions where such solutions correspond to detection algorithms that dominate each other on some of the criteria. The evaluation procedure proposed in this paper uses Hasse diagrams to produce a final ordering of object detection algorithms using multiple performance indicators (precision, recall, and geometric detection accuracy).

3. Data set

The data set used for evaluation covers the Legaspi City as a very challenging test site for the identification and localization of human settlements. Legaspi City, the capital of the Albay province



(a) Panchromatic band (b) Visible multispectral bands (c) Reference map

© QuickBird © DigitalGlobe 2005, Distributed by Eurimage 2005

Fig. 1. QuickBird image of Legaspi, the Philippines, and the reference map that contains 3064 buildings shown in pseudocolor.

in Bicol, the Philippines, is a multi-hazard hot-spot. Mount Mayon is one of the most active volcanoes in the Philippines with 48 eruptions since its recordings in 1616. Due to its location on the Ring of Fire in the Western Pacific, the Philippines are exposed to earthquakes. A tsunami risk also exists either due to an earthquake from a tectonic structure or because of debris avalanches that could reach the Albay Gulf if the edifice of Mayon would collapse. Besides frequent cyclone impacts, due to the flat and swamp area the city is located in, floods are frequent as a consequence of heavy rainfall. Therefore, the city of Legaspi was selected in the context of a cooperation research project of the World Bank and JRC/ISFEREA to perform a multi-hazard risk analysis based on very high spatial resolution remote sensing data.

A cloud-free QuickBird scene covering the city of Legaspi was acquired on November 7, 2005, and field data such as differential GPS measurements, building structure and infrastructure information were collected. In order to perform a detailed risk analysis based on geospatial data, it is necessary to know the quality of building structure and infrastructure as well as social discrepancies and their geospatial distribution. One of the most required data layers is a building layer preferably available as vector layer. Therefore, all buildings in Legaspi were digitized after a very lengthy manual process.

The data provided to the contest participants consisted of a panchromatic band with 0.6 m spatial resolution and 1668×1668 pixels, and four multispectral bands with 2.4 m spatial resolution and 418×418 pixels. Each submission was expected to be an image where the pixels corresponding to each detected building were labeled with a unique integer value. The raw data and the manually digitized reference map that was used for evaluation are shown in Fig. 1.

4. Evaluation procedure

The proposed evaluation procedure has three stages: finding correspondences between the reference objects in the ground truth and the objects in an algorithm output, measuring the accuracy of detection using these matches, and ordering of the algorithms using a combination of multiple measures. In the formulation below, the ith reference object is denoted as O_i while the jth output object is shown as $\widehat{O}_j$. The set of objects in the reference map are denoted as $\mathcal{O}_r = \{O_0, O_1, \ldots, O_{N_r}\}$ and the output objects are denoted as $\mathcal{O}_o = \{\widehat{O}_0, \widehat{O}_1, \ldots, \widehat{O}_{N_o}\}$. O_0 and $\widehat{O}_0$ correspond to the backgrounds in the reference and the output maps, respectively. N_r and N_o are the number of objects in the reference and the output maps, respectively. $|O|$ represents the size of the object O, and the size of the whole image is shown as $|I|$ (all in number of pixels). Finally, the amount of overlap between the ith reference object and the jth output object is denoted as C_{ij} (also in number of pixels).

4.1. Matching algorithms

This section describes three algorithms for finding matches between the reference and the output objects. The first two algorithms were adapted from different studies on the evaluation of image segmentation algorithms. Adaptation of these measures involved handling of the objects and the background separately. The third algorithm is proposed in this paper.

4.1.1. Bipartite graph matching

Jiang et al. (2006) proposed a bipartite graph matching algorithm for image segmentation evaluation. First, $\mathcal{O}_r$ and $\mathcal{O}_o$ are represented as one common set of nodes $\{O_0, O_1, \ldots, O_{N_r}\} \cup \{\widehat{O}_0, \widehat{O}_1, \ldots, \widehat{O}_{N_o}\}$ of a graph. Then, this graph is set up as a complete bipartite graph by inserting edges between each pair of nodes where the weight of the edge between $(O_i, \widehat{O}_j)$ is equal to C_{ij}. Given this graph, the match between the reference object map and the output object map can be found by determining a maximum-weight bipartite graph matching that is defined by a subset $\{(O_{i_1}, \widehat{O}_{j_1}), \ldots, (O_{i_k}, \widehat{O}_{j_k})\}$ such that each of the nodes O_i and $\widehat{O}_j$ has at most one incident edge, and the sum of the weights is maximized over all possible subsets of edges. The nodes corresponding to the backgrounds O_0 and $\widehat{O}_0$ are removed from the graph before the matching operation so that possible matchings with the backgrounds do not contribute to the sum of the weights.

The problem of computing the maximum-weight bipartite graph matching can be solved using techniques such as the Hungarian algorithm (Munkres, 1957). Given the matching objects, the degree (accuracy) of the match can be computed as

$$BGM(\mathcal{O}_r, \mathcal{O}_o) = \frac{w}{|I| - C_{00}}, \tag{1}$$

where w is the sum of the weights in the result of the matching. In (Jiang et al., 2006), the sum of the weights is divided by the number of pixels in the image since the whole image is used in segmentation evaluation. In this version, w is divided by the size of the union of the objects in the reference and output object maps as the upper bound. Larger values of (1) correspond to a better performance.

This algorithm finds the object pairs that result in the maximum total overlap among all possible object pairs. However, by definition, it can only find one-to-one matches between the reference and the output objects. Fig. 2a shows the matches found by this algorithm in a synthetic example. Six one-to-one matching instances are found with remaining three missed detections and four false alarms.

4.1.2. Hoover index

Hoover et al. (1996) classify every pair of reference O_i and output $\widehat{O}_j$ objects as correct detections, over-detections, under-detections, missed detections or false alarms with respect to a given threshold T, where $0.5 < T \leqslant 1$, as follows:

1. A pair of objects O_i and $\widehat{O}_j$ is classified as an instance of correct detection if
   - $C_{ij} \geqslant T \times |\widehat{O}_j|$ with an overlap score of $s_1 = C_{ij}/|\widehat{O}_j|$, and
   - $C_{ij} \geqslant T \times |O_i|$ with an overlap score of $s_2 = C_{ij}/|O_i|$.

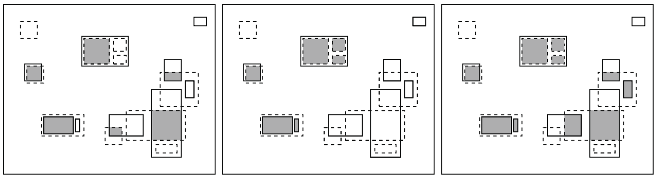
2. An object O_i and a set of objects $\widehat{O}_{j_1}, \ldots, \widehat{O}_{j_k}$, $2 \leqslant k \leqslant N_o$, are classified as an instance of over-detection if
   - $C_{ij_t} \geqslant T \times |\widehat{O}_{j_t}|$, $\forall t \in \{1, \ldots, k\}$ with an overall overlap score of $s_1 = \sum_{t=1}^{k} C_{ij_t} / \sum_{t=1}^{k} |\widehat{O}_{j_t}|$, and
   - $\sum_{t=1}^{k} C_{ij_t} \geqslant T \times |O_i|$ with an overall overlap score of $s_2 = \sum_{t=1}^{k} C_{ij_t} / |O_i|$.

3. A set of objects $O_{i_1}, \ldots, O_{i_k}$, $2 \leqslant k \leqslant N_r$, and an object $\widehat{O}_j$ are classified as an instance of under-detection if
   - $\sum_{t=1}^{k} C_{i_tj} \geqslant T \times |\widehat{O}_j|$ with an overall overlap score of $s_1 = \sum_{t=1}^{k} C_{i_tj} / |\widehat{O}_j|$, and
   - $C_{i_tj} \geqslant T \times |O_{i_t}|$, $\forall t \in \{1, \ldots, k\}$ with an overall overlap score of $s_2 = \sum_{t=1}^{k} C_{i_tj} / \sum_{t=1}^{k} |O_{i_t}|$.

4. A reference object O_i is classified as a missed detection if it does not participate in any instance of correct detection, over-detection or under-detection.
5. An output object $\widehat{O}_j$ is classified as a false alarm if it does not participate in any instance of correct detection, over-detection or under-detection.

Although these definitions result in a classification for every reference and output object, these classifications may not be unique for $T < 1.0$ as discussed in (Hoover et al., 1996). However, for $0.5 < T < 1$, an object can contribute to at most three classifications, namely, one correct detection, one over-detection and one under-detection. When an object participates in two or three classification instances, the instance with the highest overlap score is selected for that object. The score for a match instance is computed using the average of the two overlap scores (s_1 and s_2) in the corresponding definition, and the overall performance score is computed using the average of the scores for all match instances as

$$Hoo ver(\mathcal{O}_r, \mathcal{O}_o) = \frac{1}{H} \sum_{i=1}^{H} \frac{s_{i1} + s_{i2}}{2}, \tag{2}$$



(a) Bipartite graph matching (b) Hoover index (c) Multi-object maximum overlap matching

Fig. 2. Matching examples in a synthetic image. Rectangles with solid and dashed boundaries represent the reference and the output objects, respectively. Shaded areas represent the overlapping portions of the matched objects. The overall match performance scores were computed as 0.3336, 0.8083, and 0.8566 for (a), (b), and (c), using Eqs. (1), (2), and (13), respectively.

where H is the number of match instances. Larger values of (2) correspond to a better performance.

This algorithm can find over-detections (one-to-many matches) and under-detections (many-to-one matches). However, the number of matches may not always change monotonically with increasing or decreasing tolerance threshold T, and a particular choice of T may produce inconsistent results (Jiang et al., 2006). Fig. 2b shows the matches found by this algorithm in a synthetic example using $T = 0.6$. One correct detection, one over-detection, one under-detection, five missed detections, and five false alarm instances are found.

4.1.3. Multi-object maximum overlap matching

We developed a novel matching algorithm that allows one-to-many and many-to-one correspondences between the reference and the output object maps to handle over-detections and under-detections, respectively, without any need for a threshold. The first constraint is that an object can be found in only one matching instance. In other words, if the reference object O_i participates in a match with more than one output object (over-detection) and the output object $\widehat{O}_j$ participates in a match with more than one reference object (under-detection), then these two objects O_i and $\widehat{O}_j$ cannot be in the same matching instance. Another constraint is that the matching objects must have at least one overlapping pixel. The final constraint is that the matching should be optimal in the sense that the total overlapping area between all matching object pairs is maximized.

A matching that satisfies these constraints can be found using nonlinear integer programming. The mathematical model can be given as:

$$\text{Maximize} \quad \sum_{i=1}^{N_r} \sum_{j=1}^{N_o} C_{ij} z_{ij} \tag{3}$$

$$\text{Subject to} \quad 4 - \min\left(\sum_{i=1}^{N_r} z_{ij}, 2\right) - \min\left(\sum_{j=1}^{N_o} z_{ij}, 2\right) \geqslant z_{ij}, \quad 1 \leqslant i \leqslant N_r, \ 1 \leqslant j \leqslant N_o, \tag{4}$$

$$C_{ij} \geqslant z_{ij}, \quad 1 \leqslant i \leqslant N_r, \ 1 \leqslant j \leqslant N_o, \tag{5}$$

$$z_{ij} = 0 \text{ or } 1, \quad 1 \leqslant i \leqslant N_r, \ 1 \leqslant j \leqslant N_o \tag{6}$$

where $z_{ij} = 1$ if the reference object O_i matches with the output object $\widehat{O}_j$, and 0 otherwise. Constraint (4) forces z_{ij} to be 0 if O_i has at least two correspondences in the output map and $\widehat{O}_j$ has at least two correspondences in the reference map in the optimal matching (an object cannot participate in an over-detection and an under-detection instance at the same time). Constraint (5) ensures that C_{ij} is at least 1 for a match to occur ($z_{ij} = 1$). Constraint (6) forces z_{ij} to be either 0 or 1 in the optimal matching.

The optimal matching found using this formulation is not limited to only one-to-one matches as in (Jiang et al., 2006) and is more flexible than (Hoover et al., 1996) in terms of allowing correct, over- and under-detections without any need for a threshold (such a threshold can be handled if needed by modifying the constraint (5)). Fig. 2c shows the matches found by this algorithm in a synthetic example. One one-to-one match, one one-to-many match (over-detection), three many-to-one matches (under-detection), one missed detection, and three false alarm instances are found.

4.2. Performance measures

The accuracy of the detection with respect to the matching by the maximum-weight bipartite graph matching algorithm is computed using Eq. (1) which corresponds to the ratio of the number of overlapping pixels between the matching reference and output objects to the total number of pixels in the union of all objects. The accuracy of the detection with respect to the Hoover matching is computed using Eq. (2) which corresponds to the average of the overlap scores for all matching instances. None of these accuracy measures is sensitive to the shapes of the objects or the boundary and fragmentation errors.

In this section, we propose a performance measure that can distinguish such cases. Let $U = \{(x_1^U, y_1^U), \ldots, (x_m^U, y_m^U)\}$ and $V = \{(x_1^V, y_1^V), \ldots, (x_n^V, y_n^V)\}$ be the set of pixels in the reference and the output objects, respectively, in a particular matching instance. U and V can contain pixels from multiple objects for an under-detection and an over-detection instance, respectively. We model the shape of an object using the distance transform. For each pixel in an object, the distance transform computes its distance to the closest boundary point of that object (i.e., the reference object for the pixels in U and the output object for the pixels in V). Then, U and V are treated as discrete random variables with distributions $P_U = \{p_1^U, \ldots, p_m^U\}$ and $P_V = \{p_1^V, \ldots, p_n^V\}$, respectively, in $\mathbb{Z}^2$ where the probability value at each pixel corresponds to its distance to the object boundary. The distance values are normalized to add up to 1 to have a valid distribution. The values for the pixels that are farther away from the boundary are larger, indicating that they have a higher probability of belonging to that object. Therefore, mismatches between the ground truth pixels and the detected pixels will have a higher cost when these pixels are farther away from the boundaries as described below.

The quality of the match between U and V can be computed using the Mallows distance (Mallows, 1972) between P_U and P_V that is defined as the minimum of the expected difference between U and V, taken over all joint probability distributions F for (U, V), such that the marginal distribution of U is P_U and the marginal distribution of V is P_V. The Mallows distance is computed by solving the following optimization problem:

$$\text{Minimize} \quad E_F[\|U - V\|] = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} \|(x_i^U, y_i^U) - (x_j^V, y_j^V)\| \tag{7}$$

$$\text{Subject to} \quad f_{ij} \geqslant 0, \quad 1 \leqslant i \leqslant m, \ 1 \leqslant j \leqslant n, \tag{8}$$

$$\sum_{j=1}^{n} f_{ij} = p_i^U, \quad 1 \leqslant i \leqslant m, \tag{9}$$

$$\sum_{i=1}^{m} f_{ij} = p_j^V, \quad 1 \leqslant j \leqslant n, \tag{10}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = \sum_{i=1}^{m} p_i^U = \sum_{j=1}^{n} p_j^V = 1. \tag{11}$$

The constraints (8)–(11) ensure that F is indeed a distribution. The minimum in (7) is normalized and used as the match score for the corresponding matching instance as

$$Mallows(U, V) = 1 - \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} \|(x_i^U, y_i^U) - (x_j^V, y_j^V)\|}{\max\limits_{\substack{1 \leqslant i \leqslant m, \\ 1 \leqslant j \leqslant n}} \|(x_i^U, y_i^U) - (x_j^V, y_j^V)\|}. \tag{12}$$

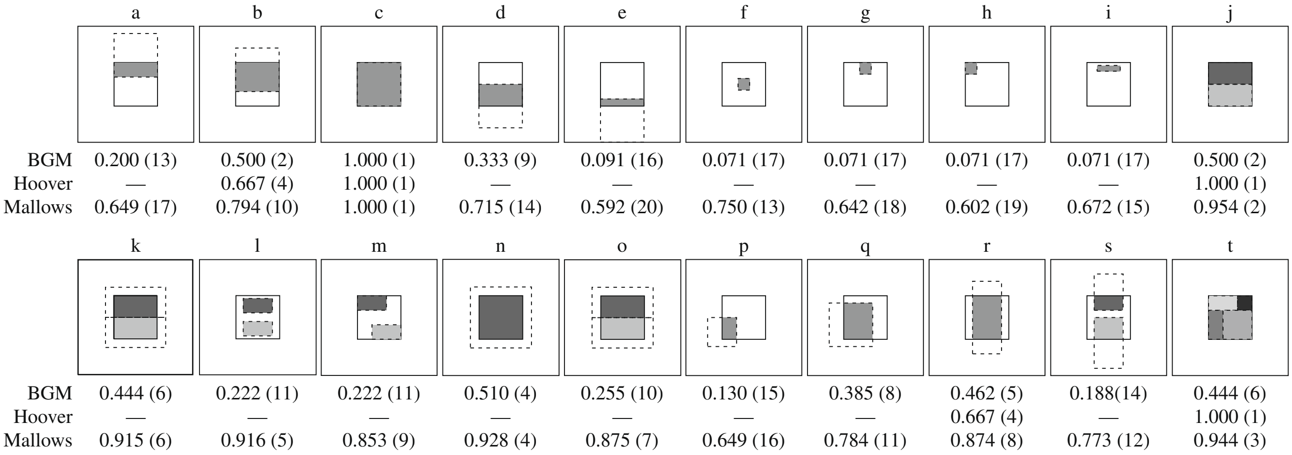
Levina and Bickel (2001) showed that the Mallows distance is equivalent to the Earth Mover's Distance (Rubner et al., 2000) between two signatures when the signatures (in our case U and V) have the same total mass (both probability distributions have a total mass of 1). Given this result, the minimization in (7) can be interpreted as finding the optimal flow $F_{ij} = (f_{ij})$ that minimizes the work required to move earth from one signature to another. In our shape model, the concentration of the earth mass corresponds to the allocation of more mass toward inside of the shape than its boundary, and the quality of the matching corresponds to the amount of work needed for the redistribution of the mass between the shapes. Furthermore, depending on the shape of an object, the corresponding distribution can have a single mode or multiple modes. The proposed measure is sensitive to fragmentation errors because fragmentation of an object in the detection output increases the number of modes further, and the increased number of modes in the probability distribution causes an increase in the amount of work needed for moving the mass from the fewer number of modes in the unfragmented reference object to the fragmented object in the output.

Given all matching instances found using the proposed matching algorithm in Section 4.1.3, the overall matching performance score is computed using the average of the scores for all matching instances as

$$Mallows(\mathcal{O}_r, \mathcal{O}_o) = \frac{1}{|\text{all}(U, V)|} \sum_{\text{all}(U,V)} Mallows(U, V). \tag{13}$$

Larger values of (13) correspond to a better performance.

Fig. 3 shows 20 synthetic examples of matching instances and the corresponding match performance scores (detection accuracy) computed using the BGM (Eq. (1)), the Hoover (Eq. (2)), and the proposed Mallows (Eq. (13)) measures. An overlap threshold of $T = 0.6$ was used for the Hoover index. The examples show that the Hoover algorithm classifies most of the instances as unmatched because of this minimum overlap requirement (T must be greater than 0.5 by definition). Furthermore, it also cannot distinguish fragmentation of the detection, and assigns the same score to such



	a	b	c	d	e	f	g	h	i	j
BGM	0.200 (13)	0.500 (2)	1.000 (1)	0.333 (9)	0.091 (16)	0.071 (17)	0.071 (17)	0.071 (17)	0.071 (17)	0.500 (2)
Hoover	—	0.667 (4)	1.000 (1)	—	—	—	—	—	—	1.000 (1)
Mallows	0.649 (17)	0.794 (10)	1.000 (1)	0.715 (14)	0.592 (20)	0.750 (13)	0.642 (18)	0.602 (19)	0.672 (15)	0.954 (2)

	k	l	m	n	o	p	q	r	s	t
BGM	0.444 (6)	0.222 (11)	0.222 (11)	0.510 (4)	0.255 (10)	0.130 (15)	0.385 (8)	0.462 (5)	0.188(14)	0.444 (6)
Hoover	—	—	—	—	—	—	—	0.667 (4)	—	1.000 (1)
Mallows	0.915 (6)	0.916 (5)	0.853 (9)	0.928 (4)	0.875 (7)	0.649 (16)	0.784 (11)	0.874 (8)	0.773 (12)	0.944 (3)

Fig. 3. Matching performance measure examples using synthetic images. Rectangles with solid and dashed boundaries represent the reference and the output objects, respectively. Shaded areas represent the overlapping portions of the matched objects. The scores computed using the three measures are given below each example. Larger scores correspond to a better performance. The rank for each match instance within the scores for a particular measure is also shown in parenthesis.

cases (c, j, t). The BGM measure can provide a score for each instance but considers only one of the output objects in one-to-many matches (j, l, m, o, s, t). Furthermore, it cannot distinguish the accuracy of the detection according to the location of the overlap when the amount of the overlap is the same (f, g, h, i, and l, m). The proposed Mallows measure produces a more intuitive ranking that is also sensitive to the locations of the detections (f, g, h, i, and l, m) and fragmentations (c, j, t, and n, o).

4.3. Multi-criteria ranking

The last stage of the evaluation procedure is the ranking of the object detection algorithms. The performances of different detection algorithms can be compared using the number of matches between the reference objects and the output objects as well as the quality of these matches that can be computed using Eqs. (1), (2), and (13) as the detection accuracy scores. Precision and recall have been commonly used in the literature to measure how well the detected objects correspond to the reference objects (Akcay and Aksoy, 2008). Recall can be interpreted as the number of true positive objects detected by an algorithm, while precision evaluates the tendency of an algorithm for false positives. Once all reference and output objects are matched using the algorithms described in Section 4.1, precision and recall are computed as

$$\text{precision} = \frac{\#\text{ of correctly detected objects}}{\#\text{ of all detected objects}} = \frac{N_o - FA}{N_o}, \qquad (14)$$

$$\text{recall} = \frac{\#\text{ of correctly detected objects}}{\#\text{ of all objects in the reference map}} = \frac{N_r - MD}{N_r}, \qquad (15)$$

where *FA* and *MD* are the number of false alarms (unmatched objects in the algorithm output) and missed detections (unmatched objects in the reference map), respectively.

Given the precision, recall, and detection accuracy scores as multiple indicators of performance that provide complementary information, a conventional solution for ranking different algorithms is to use a weighted linear combination of these indicators where any choice of the weights involves a judgement about the trade-off among the indicators. Another way of grouping the algorithms based on their indicator values is through multi-criteria optimization that can provide a set of Pareto optimal solutions (Bruzzone and Persello, 2008). A solution (in this case, a detection algorithm) is said to be Pareto optimal if it is not dominated by any other solution. A solution is said to dominate another solution if it is better than the latter in all criteria. The set of Pareto optimal detection algorithms can be considered to be better than others, but this method does not provide an explicit ranking of the algorithms.

Alternatively, Patil and Taillie (2004) proposed a ranking method that uses Hasse diagrams that represent partial orderings in the indicator space. A Hasse diagram is a planar graph used for representing partially ordered sets. Given a set S of items (in this case, a set of detection algorithms) where a suite of p indicator values is available for each member of the set, two items a and a' can be compared based on their indicator values $(I_1, I_2, \ldots, I_p)$ and $(I'_1, I'_2, \ldots, I'_p)$, respectively. If $I_j \leqslant I'_j$ for all j, then a' is considered to be intrinsically "better" than a, and is written as $a \leqslant a'$. $a < a'$ means $a \leqslant a'$ but $a \neq a'$. Furthermore, an item a' is said to cover item a if $a < a'$ and there is no other item b for which $a < b < a'$. When a' covers a, it is shown as $a \prec a'$. In a Hasse diagram, each item is represented as a vertex. Item a' is located higher than item a whenever $a < a'$. Furthermore, a and a' are connected by an edge whenever $a \prec a'$. The Hasse diagram may contain multiple connected components where items that belong to different components are considered to be not comparable.

A consistent ranking of a partially ordered set is an enumeration, $a_1, a_2, \ldots, a_n$, of its elements that satisfies $a_i > a_j \Rightarrow i < j$. A possible ranking of a partially ordered set is called a linear extension of the set. The probability of possible ranks can be used for sorting a partially ordered set. The rank interval of an item can be computed using its upper and lower sets. Given S, the upper set of item $a \in S$ is defined as

$$U_a = \{x \in S : x > a\}. \qquad (16)$$

Similarly, the lower set is defined as

$$L_a = \{x \in S : x < a\}. \qquad (17)$$

The rank interval of item a can be defined as

$$|U_a| + 1 \leqslant r \leqslant |S| - |L_a|, \qquad (18)$$

where there is a ranking that assigns rank r to item a. The collection of all linear extensions of S is denoted as Ω. Members of Ω are denoted by the symbol ω, and the rank that ω assigns to $a \in S$ is written as $\omega(a)$. Then, the rank frequency distribution of item a is given by

$$f_a(r) = \#\{\omega \in \Omega : \omega(a) = r\}, \qquad (19)$$

and the corresponding cumulative rank frequency distribution is obtained as

$$F_a(r) = f_a(1) + f_a(2) + \cdots + f_a(r) = \#\{\omega \in \Omega : \omega(a) \leqslant r\}. \qquad (20)$$

Patil and Taillie (2004) proposed to use the cumulative rank frequency operator for linearizing the partially ordered set represented in the Hasse diagram. The operator uses cumulative rank frequency distributions as new indicator values, and creates a new partially ordered set from the original one. This operation is applied iteratively until the partially ordered set becomes linear. In other words, the final set has only one linear extension that gives the ranking of the items (the object detection algorithms).

We use the precision, recall, and detection accuracy scores as indicator values for ranking object detection algorithms. The cumulative rank frequency operator creates ties if two or more algorithms have exactly the same indicator values. For the cases of ties among some algorithms, those algorithms are ranked among each other according to their detection accuracy scores.

4.4. Computational complexity

Before we present the details of the participating methods and the results, we would like to discuss the computational complexity of different steps in the evaluation procedure. The efficiency of matching algorithms in Section 4.1 can be a concern when the number of candidates significantly increases. The total CPU time for computing the proposed optimal matching depends on the size of the overlap matrix containing C_{ij} and the solver used for nonlinear integer programming. The overlap matrix is generally a sparse matrix for object detection evaluation. For example, given 3064 objects in the reference map and a similar number of objects in the output maps, only 0.05% of the values are greater than 0 on average for the contest submissions. Finding the solutions for sub-components of this matrix, and combining the optimal matches for these sub-components can reduce the amount of computations if needed. As described in (Rubner et al., 2000), the CPU time for computing the Earth Mover's Distance or the Mallows distance depends on the size of the sets U and V (corresponding to the number of pixels in the matching objects) in the formulation in Section 4.2. The computational complexity of the Mallows distance for a matching instance grows exponentially in the number of pixels. For the cases having a very large number of pixels, subsampling of the pixels before the normalization of the probability distributions P_U and P_V or approximation algorithms for the Earth Mover's

Distance can be used as alternative solutions. Finally, the CPU time for ranking the detection algorithms by linearizing the Hasse diagrams as described in Section 4.3 depends on the number of algorithms (i.e., the number of vertices in the diagram). The number of linear extensions of the diagram grows with factorial complexity with respect to the number of vertices. This was not a concern for nine algorithms (vertices) in our case, but Patil and Taillie (2004) suggest using Markov Chain Monte Carlo sampling for very large sets if needed.

5. Participating methods

This section summarizes the methods used for obtaining the nine detection results that were submitted by six groups to the building detection task in the PRRS 2008 algorithm performance contest. More details can be found in (Aksoy et al., 2008).

5.1. Orfeo

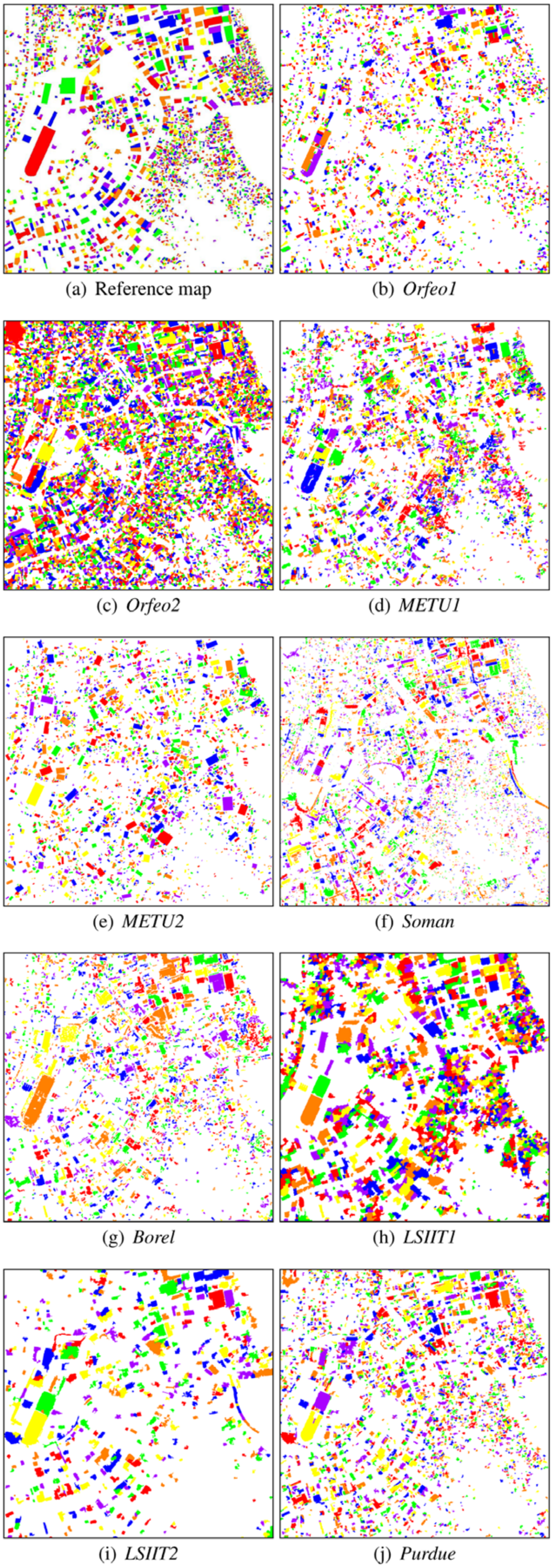
Two submissions were made by Emmanuel Christophe from CRISP in Singapore and Jordi Inglada from CNES in France using the open source Orfeo Toolbox Library. The results were obtained using pan-sharpening of the multispectral data to the pan resolution, supervised SVM-based classification of the four spectral bands, normalized difference vegetation index (NDVI), local variance, and morphological profiles into vegetation, water, road, shadows, and several types of buildings, segmentation of the pan-sharpened image using the mean-shift algorithm, and removal of the non-building segments using the classification mask. The two submissions (namely, *Orfeo1* and *Orfeo2* in the experiments) used the same process but differed in the training samples used for land cover classification, and the parameters of the mean-shift segmentation. The results for *Orfeo1* and *Orfeo2* are shown in Fig. 4b and c, respectively.

5.2. METU

Two submissions were made by seven researchers from the Middle East Technical University (METU) in Turkey. The results were obtained using pan-sharpening, thresholding of the multispectral data to mask out vegetation, water, and shadow areas, segmenting the remaining image using the mean-shift algorithm, and classifying the segments into roads and small and large buildings using their areas and intensities. The results of this step are referred to as *METU1* in the experiments and are shown in Fig. 4d. A final filtering based on the principal axes of inertia was used to eliminate non-building regions such as long, line shaped artifacts. The results of this step are referred to as *METU2* in the experiments and are shown in Fig. 4e.

5.3. Soman

One submission was made by Jyothish Soman from the International Institute of Information Technology in India. The results were obtained using the removal of water bodies, shadows and vegetation using thresholds on multispectral data, finding seed points with neighbors with uniform reflectance, edge-sensitive region growing around the seed points using a variance criterion, and a final thresholding of the regions according to their size. This submission is referred to as *Soman* in the experiments and is shown in Fig. 4f.



(a) Reference map (b) *Orfeo1*

(c) *Orfeo2* (d) *METU1*

(e) *METU2* (f) *Soman*

(g) *Borel* (h) *LSIIT1*

(i) *LSIIT2* (j) *Purdue*

Fig. 4. The building reference map and the detection results by the nine submissions displayed in pseudocolor.

5.4. Borel

One submission was made by Christoph C. Borel from the Ball Aerospace & Technologies Corporation in the USA. The results were obtained using pan-sharpening, thresholding of the original multispectral bands and HSV features for detecting colored building roofs (red, green, blue, and bright roofs), filtering out small regions, and filtering out road-like regions using thresholds on aspect ratio and fill factor. This submission is referred to as *Borel* in the experiments and is shown in Fig. 4g.

5.5. LSIIT

Two submissions were made by Sébastien Lefèvre and Régis Witz from LSIIT, CNRS-University of Strasbourg in France. The results were obtained using a highly supervised procedure by manually placing a 5×5 pixel marker with a manually assigned label (10 classes: six building types with different roofs, water, vegetation, road, boats) on the pan-sharpened data, and using marker-based watershed segmentation for the final regions. The results of this step are referred to as *LSIIT1* in the experiments and are shown in Fig. 4h. A semi-supervised version of this algorithm was also developed where only 14 markers were manually placed and the rest of the markers were found using pixel classification with the 5-nearest neighbors classifier. The results of this version are referred to as *LSIIT2* in the experiments and are shown in Fig. 4i.

5.6. Purdue

One submission was made by Ejaz Hussein and Jie Shan from the Purdue University in the USA. The results were obtained using multi-resolution segmentation of the pan-sharpened image, finding vegetation, water and shadow masks using thresholds on multispectral values, and classifying the rest of the regions using brightness values and object geometry features. This submission is referred to as *Purdue* in the experiments and is shown in Fig. 4j.

6. Results

The building detection results for the nine algorithms described in Section 5 are shown in Fig. 4. The algorithms shared many steps such as pan-sharpening, spectral feature extraction (e.g., NDVI, HSV or other band combinations), mask generation using thresholding or classification, segmentation, and filtering based on shape (e.g., area or aspect ratio). The amount of supervision differed among different methods, ranging from only setting several thresholds to manually placing a marker on every building.

The evaluation procedure was applied to each result. The matching reference and output objects were identified and the detection accuracy scores were computed from these matches using the three algorithms described in Section 4. The precision, recall, and detection accuracy scores computed using each of the evaluation methods are shown in Figs. 5–7. We can observe that, in general, the scores provide complementary information that is also consistent with the visual inspection of the results in Fig. 4. For example, the algorithms that produced too many detections in the output usually resulted in a high recall but had a low precision due to false alarms (e.g., *Orfeo2*). On the other hand, the algorithms that produced fewer detections in the output had higher precision values if these detections were accurate, but could not achieve high recall (e.g., *LSIIT2*). Most of the algorithms were in between these two extreme conditions and produced balanced precision and recall levels. The detection accuracy scores reflected the quality of these detections.

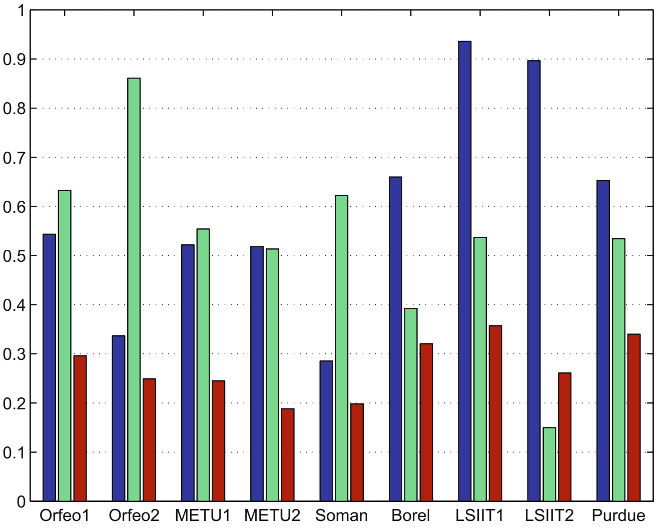


Fig. 5. Precision (blue), recall (green), and detection accuracy (red) scores obtained using the bipartite graph matching algorithm for the results in Fig. 4. (For interpretation of the references in color in this figure legend, the reader is referred to the web version of this article.)

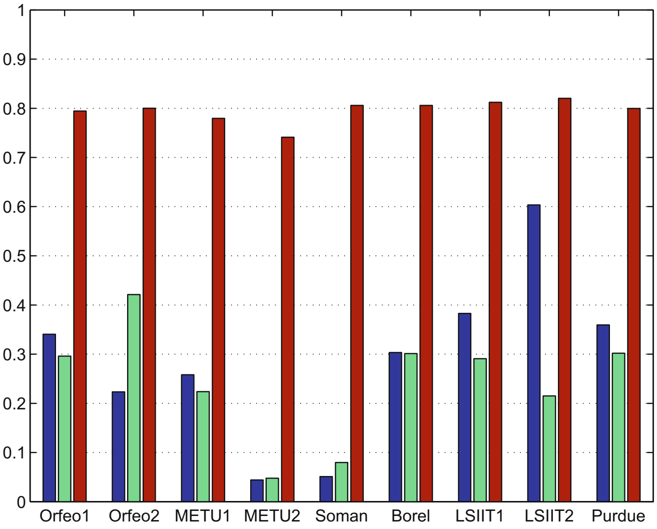


Fig. 6. Precision (blue), recall (green), and detection accuracy (red) scores obtained using the Hoover algorithm for the results in Fig. 4. (For interpretation of the references in color in this figure legend, the reader is referred to the web version of this article.)

The values for the Hoover detection score (Eq. (2)) shown in Fig. 6 were all close to 0.8 due to the overlap threshold requirement during matching. Therefore, we can conclude that the Hoover algorithm may be suitable for computing precision and recall, but may not provide a good indicator of the geometric detection accuracy. The BGM score (Eq. (1)) and the proposed Mallows score (Eq. (13)) shown in Figs. 5 and 7, respectively, also had values in a relatively small range. However, this was due to the normalization with large values in Eqs. (1) and (12). The relative values of these scores are good indicators of the detection accuracy while the Mallows score being the most powerful due to its ability to quantify geometric detection errors as also shown in the synthetic examples in Fig. 3. Furthermore, the BGM score tends to give a higher importance to larger objects to maximize the total overlap using only one-to-one matches, but this is not an issue for the proposed algo-

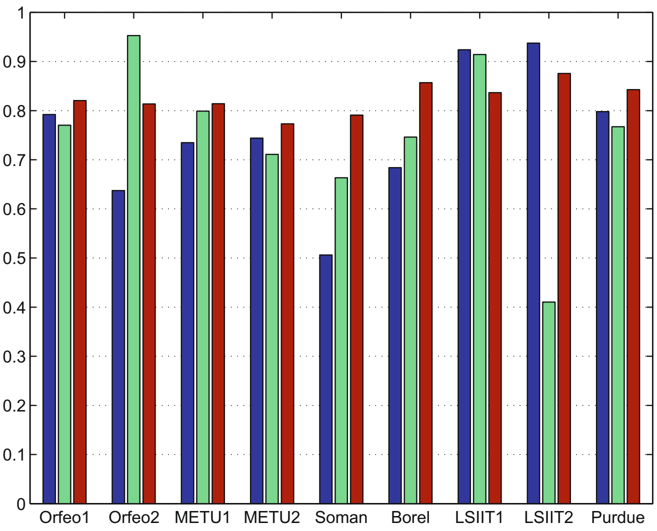
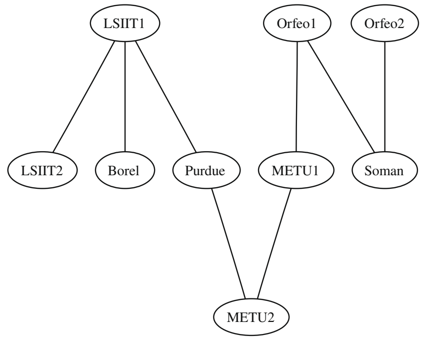
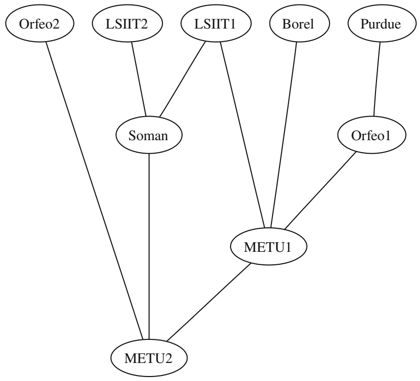


Fig. 7. Precision (blue), recall (green), and detection accuracy (red) scores obtained using the proposed multi-object maximum overlap matching algorithm and the Mallows measure for the results in Fig. 4. (For interpretation of the references in color in this figure legend, the reader is referred to the web version of this article.)



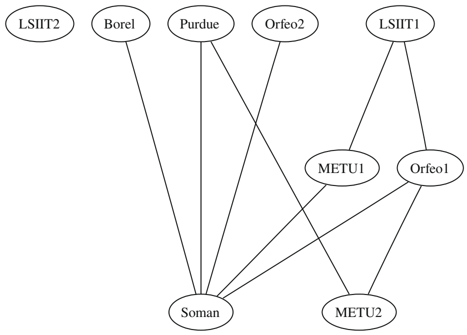
Rank 1: LSIIT1
Rank 2: Orfeo1
Rank 3: Orfeo2
Rank 4: Purdue
Rank 5: Borel
LSIIT2
METU1
Rank 8: Soman
Rank 9: METU2

Fig. 8. The Hasse diagram and the corresponding ranking for the scores in Fig. 5 obtained using the bipartite graph matching algorithm.



Rank 1: Purdue
Rank 2: LSIIT1
Rank 3: LSIIT2
Rank 4: Borel
Rank 5: Orfeo2
Orfeo1
Rank 7: Soman
Rank 8: METU1
Rank 9: METU2

Fig. 9. The Hasse diagram and the corresponding ranking for the scores in Fig. 6 obtained using the Hoover algorithm.



Rank 1: LSIIT1
Rank 2: Purdue
Rank 3: Borel
Orfeo2
Rank 5: Orfeo1
Rank 6: LSIIT2
Rank 7: METU1
Rank 8: METU2
Rank 9: Soman

Fig. 10. The Hasse diagram and the corresponding ranking for the scores in Fig. 7 obtained using the proposed multi-object maximum overlap matching algorithm and the Mallows measure.

rithm as all one-to-one, one-to-many, and many-to-one matches are considered.

Finally, the precision, recall, and detection accuracy scores were used for multi-criteria ranking as described in Section 4.3. The resulting Hasse diagrams and the final rankings are shown in Figs. 8–10. The rankings actually shared some common characteristics. We can observe four groups of detection algorithms. The first group includes *LSIIT1* and *Purdue* algorithms as the most successful. This can be explained by the heavily supervised nature of the *LSIIT1* algorithm that required the manual assignment of a seed point to every building in the image, and the iterative segmentation and classification steps of the *Purdue* algorithm that required detailed parameter tuning for the contribution of different features. The second group includes *Borel* and *LSIIT2* algorithms. This is consistent with the detection maps where these algorithms showed acceptable performance, at least for the larger buildings. The third group consists of *Orfeo1* and *Orfeo2* algorithms. These algorithms resulted in a larger number of buildings in the output map than most of the other methods. These larger number of output objects gave an increased recall, and placed these algorithms in higher ranks. This was particularly apparent in the bipartite graph matching results where the one-to-one matches covered most of the reference objects. Even though they had higher recall, their relatively lower precision due to false alarms placed them in the middle ranks. The last group includes *METU1*, *METU2*, and *Soman* algorithms. These methods were dominated by most of the others with respect to multiple performance indicators. We can conclude that the proposed evaluation procedure provided an effective linearized ranking of the detection algorithms with respect to multiple performance indicators. The rankings were also consistent with the visual inspection of the output detection maps.

7. Conclusions

We described a new evaluation procedure for empirical characterization of the performance of object detection algorithms. Unlike most of the existing methods that perform the evaluation by finding one-to-one matches between reference and output objects and by counting the number of pixels common to the matching object pairs, the proposed procedure involved a multi-object maximum overlap matching algorithm to handle one-to-many and many-to-one matches corresponding to over-detections and under-detections of the reference objects, respectively. Furthermore, a novel measure that modeled object shapes as probability distributions and quantified the detection accuracy by finding the distance between two distributions was shown to be an effective performance criterion that was sensitive to object geometry as well as boundary and fragmentation errors. Finally, a multi-criteria ranking procedure combined the precision, recall, and detection accuracy scores, and produced a final ordering of different detection algorithms.

The evaluation procedure was illustrated on the outputs of nine building detection algorithms for remotely sensed image data. The results showed that the proposed matching algorithm and the performance evaluation criteria provided an intuitive ranking of the object detection algorithms that was also consistent with visual inspection.

Acknowledgement

S. Aksoy and B. Özdemir were supported in part by the TUBITAK CAREER Grant 104E074.

References

Akcay, H.G., Aksoy, S., 2008. Automatic detection of geospatial objects using multiple hierarchical segmentations. IEEE Trans. Geosci. Remote Sens. 46 (7), 2097–2111.

Aksoy, S., Ozdemir, B., Eckert, S., Kayitakire, F., Pesaresi, M., Aytekin, O., Borel, C.C., Cech, J., Christophe, E., Duzgun, S., Erener, A., Ertugay, K., Hussain, E., Inglada, J., Lefevre, S., Ok, O., San, D.K., Sara, R., Shan, J., Soman, J., Ulusoy, I., Witz, R., 2008. Performance evaluation of building detection and digital surface model extraction algorithms: Outcomes of the PRRS 2008 algorithm performance contest. In: Proc. 5th IAPR Workshop on Pattern Recognition in Remote Sensing, Tampa, Florida.

Aksoy, S., Ye, M., Schauf, M.L., Song, M., Wang, Y., Haralick, R.M., Parker, J.R., Pivovarov, J., Royko, D., Sun, C., Farneback, G., 2000. Algorithm performance contest. In: Proc. 15th IAPR Internat. Conf. on Pattern Recognition, vol. IV, Barcelona, Spain, pp. 870–876.

Alparone, L., Wald, L., Chanussot, J., Thomas, C., Gamba, P., Bruce, L.M., 2007. Comparison of pansharpening algorithms: Outcome of the 2006 GRS-S data-fusion contest. IEEE Trans. Geosci. Remote Sens. 45 (10), 3012–3021.

Asuncion, A., Newman, D.J., 2007. UCI machine learning repository. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

Bruzzone, L., Persello, C., 2008. A novel protocol for accuracy assessment in classification of very high resolution multispectral and SAR images. In: Proc. IEEE Internat. Geoscience and Remote Sensing Symposium, Boston, Massachusetts.

Christensen, H.I., Phillips, P.J. (Eds.), 2002. Empirical Evaluation Methods in Computer Vision. World Scientific Press, Singapore.

Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A., 2008. The PASCAL Visual Object Classes Challenge 2008 Results. <http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html>.

Flynn, P.J., Hoover, A., Phillips, P.J., 2001. Special issue on empirical evaluation of computer vision algorithms. Computer Vision and Image Understanding 84 (1), 1–4.

Haralick, R.M., 1996. Propagating covariance in computer vision. Internat. J. Pattern Recognition Artif. Intell. 10 (5), 561–572.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P.J., Bunke, H., Goldgof, D.B., Bowyer, K., Eggert, D.W., Fitzgibbon, A., Fisher, R.B., 1996. An experimental comparison of range image segmentation algorithms. IEEE Trans. Pattern Anal. Machine Intell. 18 (7), 673–689.

Huang, Q., Dom, B., 1995. Quantitative methods of evaluating image segmentation. In: IEEE Internat. Conf. on Image Processing, vol. 3, Washington, DC, pp. 53–56.

Jiang, X., Marti, C., Irniger, C., Bunke, H., 2006. Distance measures for image segmentation evaluation. EURASIP J. Appl. Signal Process., 1–10 (Article ID 35909).

Levina, E., Bickel, P., 2001. The earth mover's distance is the mallows distance: Some insights from statistics. In: Proc. IEEE Internat. Conf. on Computer Vision, vol. 2. Vancouver, British Columbia, Canada, pp. 251–256.

Liu, G., Haralick, R.M., 2002. Optimal matching problem in detection and recognition performance evaluation. Pattern Recognition 35 (10), 2125–2139.

Liu, X., Kanungo, T., Haralick, R.M., 2005. On the use of error propagation for statistical validation of computer vision software. IEEE Trans. Pattern Anal. Machine Intell. 27 (10), 1603–1614.

Mallows, C.L., 1972. A note on asymptotic joint normality. Ann. Math. Statist. 43 (2), 508–515.

Mariano, V. Y., Min, J., Park, J.-H., Kasturi, R., Mihalcik, D., Li, H., Doermann, D., Drayer, T., 2002. Performance evaluation of object detection algorithms. In: Proc. 16th IAPR Internat. Conf. on Pattern Recognition, vol. 3, Quebec, Canada, pp. 965–969.

Martin, D.R., Fowlkes, C.C., Malik, J., 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans. Pattern Anal. Machine Intell. 26 (5), 530–549.

Meur, Y.L., Vignolle, J.-M., Chanussot, J., 2008. Practical use of receiver operating characteristic analysis to assess the performances of defect detection algorithms. J. Electron. Imaging 17 (3).

Munkres, J., 1957. Algorithms for the assignment and transportation problems. J. Soc. Ind. Appl. Math. 5 (1), 32–38.

Ortiz, A., Oliver, G., 2006. On the use of the overlapping area matrix for image segmentation evaluation: A survey and new performance measures. Pattern Recognition Lett. 27 (16), 1916–1926.

Pacifici, F., Frate, F.D., Emery, W.J., Gamba, P., Chanussot, J., 2008. Urban mapping using coarse SAR and optical data: Outcome of the 2007 GRSS data fusion contest. IEEE Geosci. Remote Sens. Lett. 5 (3), 331–335.

Patil, G.P., Taillie, C., 2004. Multiple indicators, partially ordered sets, and linear extensions: Multi-criterion ranking and prioritization. Environ. Ecol. Statist. 11 (2), 199–228.

Phillips, P.J., Bowyer, K.W., 1999. Empirical evaluation of computer vision algorithms. IEEE Trans. Pattern Anal. Machine Intell. 21 (4), 289–290.

Rubner, Y., Tomasi, C., Guibas, L.J., 2000. The earth mover's distance as a metric for image retrieval. Internat. J. Comput. Vision 40 (2), 99–121.

Smeaton, A.F., Over, P., Kraaij, W., 2006. Evaluation campaigns and TRECVid. In: Proc. 8th ACM Internat. Workshop on Multimedia Information Retrieval. Santa Barbara, California, pp. 321–330.

Thacker, N.A., Clark, A.F., Barron, J.L., Beveridge, J.R., Courtney, P., Crum, W.R., Ramesh, V., Clark, C., 2008. Performance characterization in computer vision: A guide to best practices. Computer Vision and Image Understanding 109 (3), 305–334.

Wirth, M., Fraschini, M., Masek, M., Bruynooghe, M., 2006. Performance evaluation in image processing. EURASIP J. Appl. Signal Process., 1–3 (Article ID 45742).

Zhang, D., Lu, G., 2004. Review of shape representation and description techniques. Pattern Recognition 37 (1), 1–19.