Performance Evaluation of Building Detection and Digital Surface Model Extraction Algorithms: Outcomes of the PRRS 2008 Algorithm Performance Contest

Selim Aksoy[1], Bahadır Özdemir[1], Sandra Eckert[2], Francois Kayitakire[2], Martino Pesarasi[2], Orsan Aytekin[6], Christoph C. Borel[9], Jan Čech[7], Emmanuel Christophe[3], Şebnem Düzgün[5], Arzu Erener[5], Kıvanç Ertugay[5], Ejaz Hussain[11], Jordi Inglada[4], Sébastien Lefèvre[10], Özgün Ok[5], Dilek Koç San[5], Radim Šára[7], Jie Shan[11], Jyothish Soman[8], Ilkay Ulusoy[6], Régis Witz[10]

Abstract

This paper presents the initial results of the Algorithm Performance Contest that was organized as part of the 5th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS 2008). The focus of the 2008 contest was automatic building detection and digital surface model (DSM) extraction. A QuickBird data set with manual ground truth was used for building detection evaluation, and a stereo Ikonos data set with a highly accurate reference DSM was used for DSM extraction evaluation. Nine submissions were received for the building detection task, and three submissions were received for the DSM extraction task. We provide an overview of the data sets, the summaries of the methods used for the submissions, the details of the evaluation criteria, and the results of the initial evaluation.

1. Introduction

The goal of the algorithm performance contest that was organized as part of the 5th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS 2008, http://www.iapr-tc7.org/prrs08) was the evaluation of pattern recognition techniques on different remote sensing data sets with known ground truth. The contest was coordinated jointly by the International Association for Pattern Recognition (IAPR) Technical Committee 7 on Remote Sensing (http://www.iapr-tc7.org) and the ISFEREA Action of the European Commission, Joint Research Centre, Institute for the Protection and Security of the Citizen (http://isferea.jrc.ec.europa.eu).

The focus of the 2008 contest was automatic building detection and building height extraction. The precise identification and localization of settlement features is one of the key information sets needed for territorial planning and in any assessment related to human security and safety decision process, from the preparedness to natural hazards and to post-disaster evaluation. Since buildings are one of the most salient settlement features, their detection from satellite imagery has long been an important research topic in remote sensing image analysis.

Despite the fact that current generation Earth Observation (EO) data can provide an updated and detailed source of information related to human settlements, the available geo-information layers derived from these data are often too outdated and/or not enough for the user needs. Furthermore, accurate automatic interpretation using traditional techniques that are based on spectral properties is only possible for low-resolution EO data, while new methods are not stable and mature enough for supporting high- and very high-resolution (VHR) satellite data.

In this perspective, optimization of the automatic information extraction from human settlements using new generation satellite data is particularly important, and

[1] S. Aksoy and B. Özdemir are with Department of Computer Engineering, Bilkent University, Bilkent, 06800, Ankara, Turkey.

[2] S. Eckert, F. Kayitakire and M. Pesaresi are with Institute for the Protection and Security of the Citizen, European Commission, Joint Research Centre, 21020 Ispra (VA), Italy.

[3] E. Christophe is with CRISP, Block SOC-1, Level 2, Lower Kent Ridge Road, Singapore 119260.

[4] J. Inglada is with CNES, DCT/SI/AP, 18, Av. E. Belin, 31401 Toulouse Cedex 9, France.

[5] Ş. Düzgün, D. Koç San, Ö. Ok, A. Erener and K. Ertugay are with Geodetic and Geographic Information Technologies, Middle East Technical University, Ankara, Turkey.

[6] I. Ulusoy and O. Aytekin are with Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey.

[7] J. Čech and R. Šára are with Center for Machine Perception, Department of Cybernetics, Czech Technical University in Prague, Czech Republic.

[8] J. Soman is with International Institute of Information Technology, Gachibowli, Hyderabad, 500019, India.

[9] C. C. Borel is with Ball Aerospace & Technologies Corp., 2875 Presidential Drive, Fairborn, OH 45324, USA.

[10] S. Lefèvre and R. Witz are with LSIIT, CNRS-University of Strasbourg, UMR 7005, Pôle API, Bvd. Brant, 67412 Illkirch, France.

[11] E. Hussain and J. Shan are with Geomatics Engineering, School of Civil Engineering, Purdue University, West Lafayette, IN 47907, USA.

978-1-4244-2653-9/08/$25.00 ©2008 IEEE

the present contest offers an important contribution toward this direction. This paper presents the initial results of the performance evaluation of building detection and digital surface model (DSM) extraction tasks in the PRRS 2008 Algorithm Performance Contest. Section 2 presents the QuickBird data used for building detection, the summaries of nine methods contributed by six groups, the evaluation criteria used, and the results of initial evaluation. Section 3 presents the stereo Ikonos data used for DSM extraction, the summaries of three methods contributed by one group, the evaluation criteria used, and the results of initial evaluation.

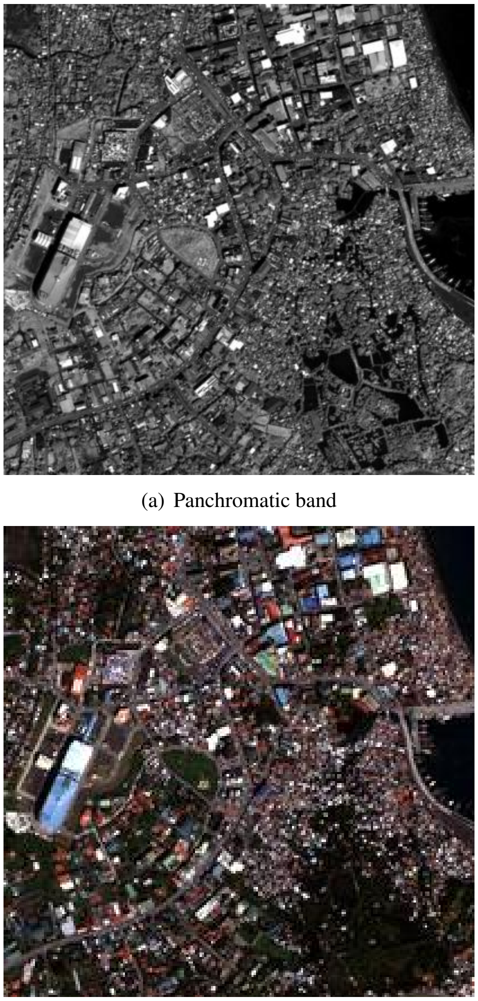
2. Task 1: Building detection from monocular data

2.1. Background and data set

Legaspi City, the capital of the Albay province in Bicol, the Philippines, is a multi-hazard hot-spot with cyclone, volcano eruption, earthquake, tsunami and flood risks. 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 VHR 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 manually; a time demanding and very tedious work.

An automatic or semi-automatic approach to detect and extract buildings would very much simplify the initial step of building information gathering before performing any kind of built-up structure related hazard vulnerability and risk analysis. Consequently, the development of such an algorithm was decided to be a task advertised in this contest. The data provided to the participants consisted of a panchromatic band with 0.6m spatial resolution and 1668×1668 pixels, and four multispectral bands with 2.4m spatial resolution and 418×418 pixels (Figure 1). The manually digitized ground truth was used for evaluation (Figure 2(a)).


(a) Panchromatic band

(b) Visible multispectral bands

Figure 1. QuickBird image of Legaspi, the Philippines. (QuickBird ⓒ DigitalGlobe 2005, Distributed by Eurimage.)

2.2. Participating methods

Nine results were submitted by six groups for the building detection tasks. The methods used for obtaining these results are described below.

Orfeo Two submissions were made by Emmanuel Christophe and Jordi Inglada using the open source Orfeo Toolbox Library [19]. First, pan-sharpening was used to combine the panchromatic and multi-spectral data to get a high-resolution 4-band data set. Usually there is some important contextual information to use to avoid obvious mistakes. It is unlikely to find a house in the middle of the water unless the goal is specifically to count houses flooded during a natural disaster. This basic level information can be exploited by first creating a rough land cover classification. Classes such as water, vegetation, roads, shadows, bare soil and few ad-hoc classes provide a good starting point. To obtain this classification, a Support Vector Machine (SVM) classifier was used on a specific set of features such as the four spectral bands, the NDVI index, a local variance, and morphological profiles. This classification was used as a mask to remove some obvious false alarms in the following steps.

The next step was to segment the pan-sharpened image in order to lower the complexity of the input data. The level of details available in high-resolution images can have a strong negative effect at some stages of the processing: roof superstructures are irrelevant when trying to extract the whole building for example. The mean shift algorithm [6] was used as an efficient way to simplify such images. The segmented image was combined with the classification to remove irrelevant segments. This was the main step where some simple high level information concerning the object was introduced.

Segments were vectorized to enable higher level processing. Finally, some adjustments of the detected objects were made according to the original pan-sharpened data (precise edge adjustment). This steps fitted the obtained polygons to the input data by introducing shifts to the position of the vertices in order to maximize the overlap with respect to the edges of the original image.

The two submissions (namely, *Orfeo1* and *Orfeo2* in the experiments) used the same process but differed in two points:

- The land cover classification used was different. Same classes were used but different samples were given for the learning step.
- Parameter for the mean shift clustering was different, thus, leading to different objects.

The results for *Orfeo1* and *Orfeo2* are shown in Figures 2(b) and 2(c), respectively.

METU Two submissions were made by researchers from Middle East Technical University (METU). First, the multispectral and panchromatic images were fused by using the PANSHARP algorithm of PCI Geomatica. To determine man-made regions, it was needed to mask vegetation, shadow and water regions. The NDVI was calculated by using the NIR and red bands of the pan-sharpened image. A threshold was determined depending on the intensity values to mask the vegetated regions from the pan-sharpened image. The water and shadow areas were masked by applying a suitable threshold to the NIR band. After masking out water, shadow and vegetation regions from the pan-sharpened image, the mean-shift segmentation method [6] was used to obtain man-made regions. To mask the roads, the segmented image was classified by using the maximum likelihood classifier.

The resultant image included only the building patches and some erroneous regions because of the masking processes. To remove these erroneous regions, the data were converted to vector by using the RAS2POLY algorithm of PCI Geomatica. The mean intensity values were assigned to each vector data and some threshold values depending on the intensity values were determined to remove these erroneous regions. The cleaned building patches were converted to raster in the ArcGIS environment. In this way the buildings with unique values were obtained. To merge the over-segmented building patches, hue image, which is invariant to illumination direction and highlights, was generated. The mean hue values were calculated and the hue image was divided into two classes by using the areas of building patches as small and large, where 170m2 of area was considered to be the threshold. The neighboring building patches that had close mean hue values were merged for both small and large building data with different closeness thresholds. Finally, the small and large building data were combined to get the final building patches. The results of this step are referred to as *METU1* in the experiments and are shown in Figure 2(d).

Since some building patches might not have valid shapes such as long, line artifacts, principle component analysis was used to eliminate non-building patches. A high ratio of the eigenvalues of long and line shaped artifacts was used as an evidence of being non-building patches. After eliminating the artifacts, the candidate building patches were obtained. The results of this step are referred to as *METU2* in the experiments and are shown in Figure 2(e).

Soman One submission was made by Jyothish Soman using a fast unsupervised algorithm involving intuitive definitions to find artificial objects in a satellite image. The algorithm used the definition of an isolated artificial object as a section of the image that had a variance lower than its immediate surroundings [17]. Multispectral image was stretched to the size of the pan image by resizing the image using a bi-cubic interpolation. The pre-processing removed water bodies, shadows and vegetation from the image, using derived information from the multispectral data.

The algorithm started by finding points such that the number of its neighboring pixels with relative difference less than the variance of the image was greater than 5, i.e., the point had a nearly uniform surrounding. Thus the most probable seed points were found for region growing. The generated points formed clusters, which were joined to form regions. These regions were then used as starting zones for a variance based region growing. The mask for region growing was kept such that edges were maintained and the regions did not grow into areas containing natural bodies and shadows. Pixels were added to the regions if their values did not exceed the sum of the mean of the current region and the variance of the initial region. A final thresholding was done so that regions with an area within a range was kept. This submission is referred to as *Soman* in the experiments and is shown in Figure 2(f).

Borel One submission was made by Christoph Borel using a series of IDL programs. First, the multispectral data were pan-sharpened to the pan band resolution. Then, a mask generation step was performed to find colored building roofs. The operations in this step included performing a 2% histogram stretch on each band, performing a hue-saturation-value (HSV) transformation on the true color byte image cube, finding the red roofs if the red band's values were greater than a weight multiplied with the sum of green, blue and NIR bands (redroof), finding the green roofs if hue was between two limits and the value above a threshold (greenroof), finding the blue roofs if hue was between two limits and the value above a threshold (blueroof), and finding the bright roofs by thresholding the value (brightroof). The mask generation step was followed by size filtering and shape analysis. The operations in this step included applying a median filter to remove very small regions from all mask images, and labeling all regions and keeping the ones with a size greater than a threshold. Since the brightroof image contained some road features, every region was analyzed for its aspect ratio (length/width) and fill factor (area of minimum enclosing rectangle over actual area). Only regions with an aspect ratio greater than a threshold and a filling factor greater than a threshold were considered buildings. Finally, buildings were found by logical OR operation on the masks redroof, greenroof, blueroof and brightroof. This submission is referred to as *Borel* in the experiments and is shown in Figure 2(g).

LSIIT Two submissions were made by Sébastien Lefèvre and Régis Witz using a recent segmentation method described in [15] that is not specific to the problem under consideration. This method improves the widely used marker-based watershed segmentation by making use of the markers' content (and not only the markers' location) to guide the segmentation process. To do so, this supervised segmentation technique associates each marker to a class (a class may contain several markers). These markers are then considered as a learning set in a fuzzy classification procedure (e.g., 5-nearest neighbours) which returns a membership map per class. These maps are inverted and combined with a multispectral gradient (e.g., the Euclidean norm of a marginal morphological gradient) to produce as many topographic surfaces as classes. Finally, the segmentation is obtained following the flooding procedure which has been adapted to the case of several surfaces: water is flooding simultaneously on the different surfaces, and each pixel is given the label of the marker which reaches it first (i.e., before the other markers).

The direct application of this algorithm required to set a marker per building to be detected. Thus a second algorithm was designed as a semi-supervised solution to the problem of building detection. To limit the user intervention, a marker identification procedure was added as a pre-processing step. It was based on the markers defined by the user and aimed to find new markers. To do so it relied on a pixel classification step using user markers as a learning set. To ensure a minimum robustness to noise, the classification map was filtered with morphological opening (i.e., the minimum size of a building). To avoid border effects between close components, each connected component was also eroded using a small square structuring element. This additional procedure was designed especially for the contest (or for images where it was not relevant to manually mark each object).

The experimental setup for processing the Legaspi image started with a fusion of panchromatic and multispectral bands. Then, the markers were defined manually over the image by a computer scientist (novice in remote sensing), using a web interface such as the one available at http://dpt-info.u-strasbg.fr/∼lefevre/demos/supervisedWatershedApplet. For the first experiment (supervised watershed), the markers were defined using 10 classes (6 for buildings with different roofs, water, vegetation, road, boats). Almost each visible object was marked with the relevant class using a square of 5×5 pixels (smaller if needed). The manual labeling resulted in around 2460 objects identified by the user in 90 minutes. The segmentation procedure was much faster and required between 100 and 180 seconds depending on the optimizations considered. The results of this step are referred to as *LSIIT1* in the experiments and are shown in Figure 2(h).

For the second experiment (semi-supervised watershed), the markers were defined using 2 classes (building and non-building), with a total of markers as small as 14 markers (7 for the buildings, 7 for the other objects). Hence, the goal was to produce some markers required by the semi-supervised method very quickly (setting 14 markers on the contest image was achieved in only a few seconds). The minimum size of objects was assumed to be 11 × 11 pixels (6 × 6 meters) and was used as the structuring element size in the morphological filtering step. The segmentation procedure required between 60 and 150 seconds depending on the optimizations considered. Since the computation time was rather low and the user intervention was rather intuitive, it would be possible to consider an interactive segmentation strategy (e.g., by adding markers where the segmentation fails). The results of this step are referred to as *LSIIT2* in the experiments and are shown in Figure 2(i).

Purdue One submission was made by Ejaz Hussein and Jie Shan using an object-based image classification technique. The method mainly consisted of three steps: pan-sharpening, image segmentation, and object classification. The segmentation and classification were performed in an iterative manner.

In the data pre-processing step, the four-band multispectral image was sharpened with the panchromatic image using the Gram-Schmidt method. The resultant pan-sharpened multispectral image was then segmented to form image objects. Using NDVI, band ratio of IR to green, and brightness as features, the segmented objects were classified to two classes: vegetation and water/shadow. After performing histogram stretching on the panchromatic image, it was segmented with the vegetation and water/shadow classes being the mask. By selecting the brightness, area, and rectangular fit as features, the last segmented results were classified to find bright buildings in the panchromatic image. For other buildings, the pan-sharpened multispectral image was classified with the pre-classified vegetation, water/shadow, and bright building classes being the mask. This was carried out sequentially for green, magenta, dark, and cyan buildings. Once the buildings of one color were classified, they were used as an additional mask for the next classification. When this was completed, all building object classes were combined into one image, which was then segmented to form individual buildings. In this way, a building with several roof colors, which were initially classified as different building classes, could be combined and identified as one building. Finally, building objects of small size were filtered out. ENVI, ArcGIS and Definiens Developer were used in this submission that is referred to as *Purdue* in the experiments and is shown in Figure 2(j).

2.3. Evaluation criteria

In [21], it is stated that "there is no single method which can be considered good for all images, nor are all methods equally good for a particular type of image". Therefore, several error measures were used in this contest for the comparison of the algorithms.

In the building detection task, the outputs of the algorithms are images where the pixels corresponding to each detected building are labeled with a unique integer value. These outputs can be considered as segmentations of the image data. Therefore, all of the measures in this contest 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.

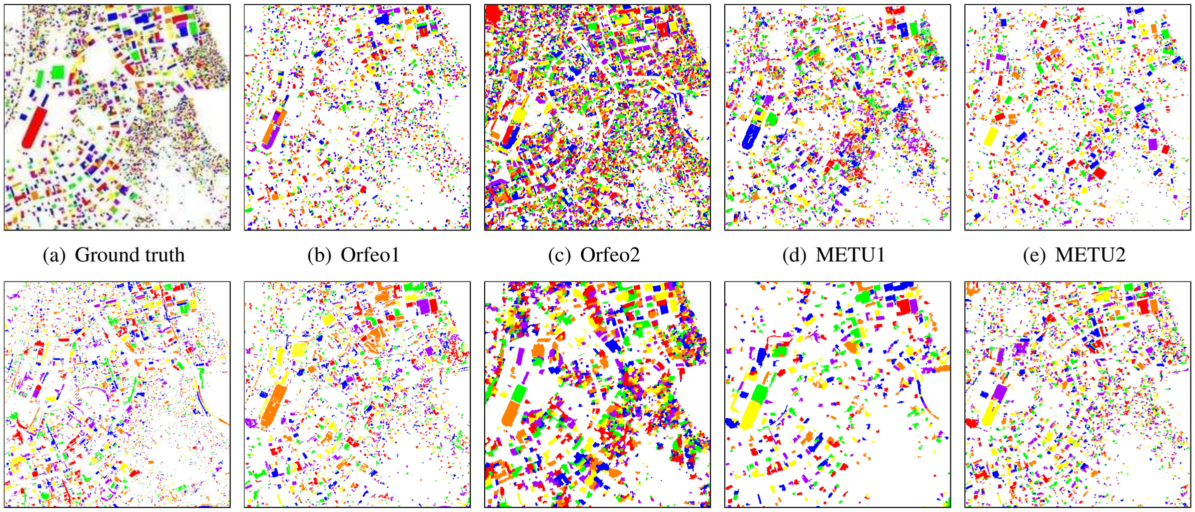
The overlapping area matrix (OAM) introduced in [1] makes computation of performance measures easier. All object-based measures given below can be computed from the OAM. Let C_{ij} be the number of pixels in the i'th object in a reference map that overlap with the j'th object in an output map produced by an algorithm. Ortiz and Oliver [20] formulated some of the performance measures used in the contest using the OAM. A similar notation is used in this paper. The i'th reference object is denoted as O_i while the j'th output object is shown as $\widehat{O}_j$. The objects of interest in the contest include the buildings and the background. 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. The sizes of the objects O_i and $\widehat{O}_j$ and the whole image I can be calculated from the OAM as

$$n(O_i) = \sum_{j=0}^{N_o} C_{ij}, \tag{1}$$

$$n(\widehat{O}_j) = \sum_{i=0}^{N_r} C_{ij}, \tag{2}$$

$$n(I) = \sum_{i=0}^{N_r} n(O_i) = \sum_{j=0}^{N_o} n(\widehat{O}_j). \tag{3}$$

Correct detection, over-detection, under-detection, missed detection, false alarm rates Hoover *et al.* [12] 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 re-



(a) Ground truth (b) Orfeo1 (c) Orfeo2 (d) METU1 (e) METU2

(f) Soman (g) Borel (h) LSIIT1 (i) LSIIT2 (j) Purdue

Figure 2. Ground truth (3065 buildings) and submissions for the building detection task displayed in pseudocolor.

spect to a given threshold T, where $0.5 < T \leq 1$, as follows:

1. A pair of objects O_i and $\widehat{O}_j$ is classified as an instance of correct detection if
   - $C_{ij} \geq T \times n(\widehat{O}_j)$,
   - $C_{ij} \geq T \times n(O_i)$.
2. An object O_i and a set of objects $\widehat{O}_{j_1}, \ldots, \widehat{O}_{j_k}$, $2 \leq k \leq N_o$, are classified as an instance of over-detection if
   - $C_{ij_t} \geq T \times n(\widehat{O}_{j_t}), \forall t \in \{1, \ldots k\}$, and
   - $\sum_{t=1}^{k} C_{ij_t} \geq T \times n(O_i)$.
3. A set of objects $O_{i_1}, \ldots, O_{i_k}$, $2 \leq k \leq N_r$, and an object $\widehat{O}_j$ are classified as an instance of under-detection if
   - $\sum_{t=1}^{k} C_{i_t j} \geq T \times n(\widehat{O}_j)$, and
   - $C_{i_t j} \geq T \times n(O_{i_t}), \forall t \in \{1, \ldots k\}$.
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.

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 [12]. When an object participates in two or three classification instances, the instance with the highest overlap score is selected for that object. For equal scores, we bias toward selecting correct detection, then over-detection, then under-detection to obtain unique classifications.

Maximum-weight bipartite graph matching The next measure is adapted from [14] where a bipartite graph matching algorithm is used for evaluating image segmentation results. 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 total sum of the weights is maximized over all possible subsets of edges.

The problem of computing maximum-weight bipartite graph matching is known as an assignment problem, and one of the solutions for this problem is the Munkres Assignment Algorithm (also known as the Hungarian Algorithm) [18]. In the Munkres algorithm, the minimum cost is aimed instead of the maximum weight. Consequently, by negating the overlapping area matrix, we obtain the cost matrix that can be used for the algorithm. Finally, a modified version of the maximum-weight bipartite graph matching measure is defined as

$$BGM(\mathcal{O}_o, \mathcal{O}_r) = 1 - \frac{w}{n(I) - C_{00}} \tag{4}$$

where w is the sum of the weights. In [14], the sum of the weights is divided by image size. In this version, w is divided by the size of the union of the objects in the reference and output object maps. The measure in (4) represents the error so smaller values correspond to a better performance.

Normalized Hamming distance Huang and Dom [13] proposed a single overall performance measure depending on region matching according to the maximum overlapping area. In the contest, we are interested in how successfully the algorithms can detect the foreground object regions, so we discard the background from the original formula. The directional Hamming distance from the output object map to the reference object map is defined as

$$D_H(\mathcal{O}_o \Rightarrow \mathcal{O}_r) = \sum_{i=1}^{N_r} \sum_{j \neq \underset{k=1,\ldots,N_o}{\arg\max} \{C_{ik}\}} C_{ij}. \tag{5}$$

Similarly, the directional Hamming distance from the reference map to the output map is defined as

$$D_H(\mathcal{O}_r \Rightarrow \mathcal{O}_o) = \sum_{j=1}^{N_o} \sum_{i \neq \underset{k=1,\ldots,N_r}{\arg\max} \{C_{kj}\}} C_{ij}. \tag{6}$$

Finally, these two distances are averaged and normalized in order to obtain a modified version of the normalized Hamming distance

$$D_{NH}(\mathcal{O}_r, \mathcal{O}_o) = \frac{1}{2} \left(\frac{D_H(\mathcal{O}_o \Rightarrow \mathcal{O}_r)}{n(I) - n(O_0)} + \frac{D_H(\mathcal{O}_r \Rightarrow \mathcal{O}_o)}{n(I) - n(\widehat{O}_0)} \right) \tag{7}$$

where $D_{NH}(\mathcal{O}_r, \mathcal{O}_o) \in [0, 1]$. The value of one indicates a total mismatch and zero indicates a perfect match.

Clustering indices Each object map can be considered as a clustering of pixels [14]. As a result, measures that compare two different clustering outputs can be used for object detection evaluation. Object pairing is one of the methods used for cluster comparison. Each pair of pixels (p_a, p_b) in the image is a member of one of the following groups

- p_a and p_b belong to the same object both in the reference map and the output map (N_{11}),
- p_a and p_b belong to the same object in the reference map but belong to different objects in the output map (N_{10}),
- p_a and p_b belong to the same object in the output map but belong to different objects in the reference map (N_{01}),
- p_a and p_b belong to different objects both in the reference map and the output map (N_{00}).

The number of pixel pairs in each group can be computed from the OAM.

The Rand Index given in [23] can be computed as

$$R(\mathcal{O}_r, \mathcal{O}_o) = 1 - \frac{N_{11} + N_{00}}{n(I) \times (n(I) - 1)/2}. \tag{8}$$

Another measure using pixel pairing is introduced by Fowlkes and Mallows in [8], and can be computed as

$$F(\mathcal{O}_r, \mathcal{O}_o) = 1 - \sqrt{W_1(\mathcal{O}_r, \mathcal{O}_o) \times W_2(\mathcal{O}_r, \mathcal{O}_o)} \tag{9}$$

where

$$W_1(\mathcal{O}_r, \mathcal{O}_o) = \frac{N_{11}}{\sum_{i=0}^{N_r} n(O_i)(n(O_i) - 1)/2}, \text{and} \tag{10}$$

$$W_2(\mathcal{O}_r, \mathcal{O}_o) = \frac{N_{11}}{\sum_{j=0}^{N_o} n(\widehat{O}_j)(n(\widehat{O}_j) - 1)/2}. \tag{11}$$

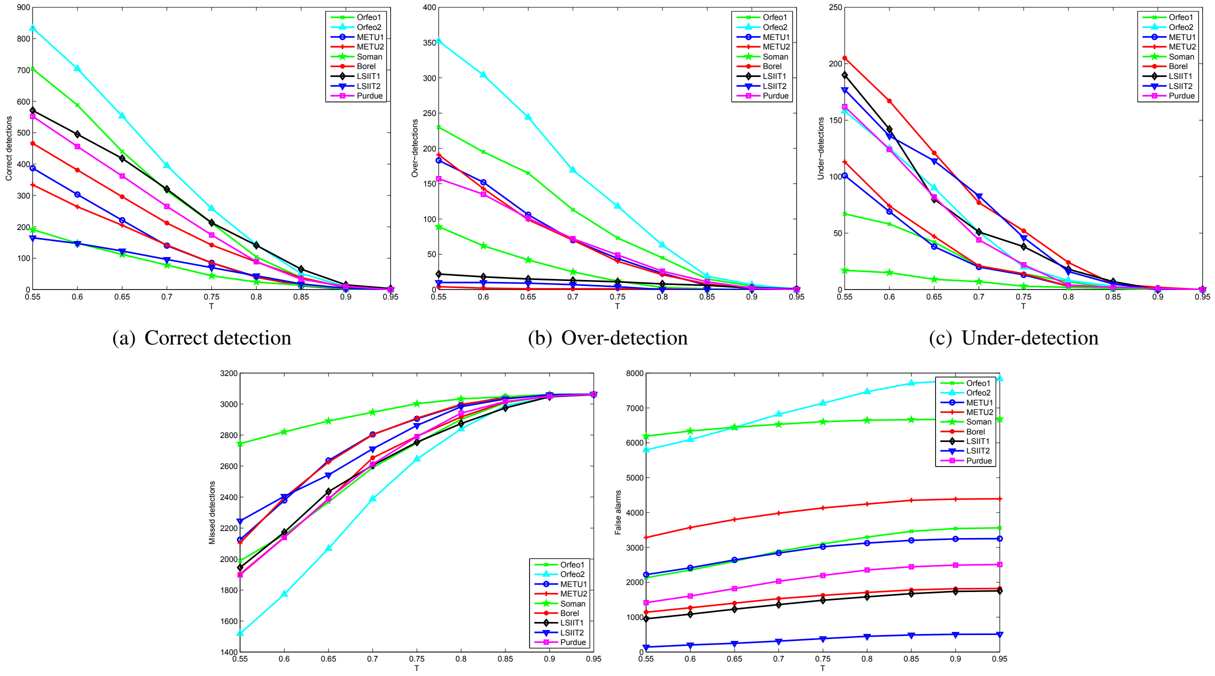
Yet another measure that uses pixel pairings for cluster comparison is the Jaccard index [2], and is defined as

$$J(\mathcal{O}_r, \mathcal{O}_o) = 1 - \frac{N_{11}}{N_{11} + N_{10} + N_{01}}. \tag{12}$$

All three measures are in the $[0, 1]$ range and are modified to represent the error (by subtracting the original index from 1) so smaller values correspond to a better performance.

2.4. Results

The measures described in Section 2.3 were computed for all nine submissions. Figure 3 shows the object-based correct detection, over-detection, under-detection, missed detection, and false alarm rates. Figure 4 shows the graph matching measure, normalized Hamming distance, and clustering indices. Higher values for correct detection, over-detection, and under-detection represent better performance. Lower values indicate better performance for the rest of the measures.



(a) Correct detection

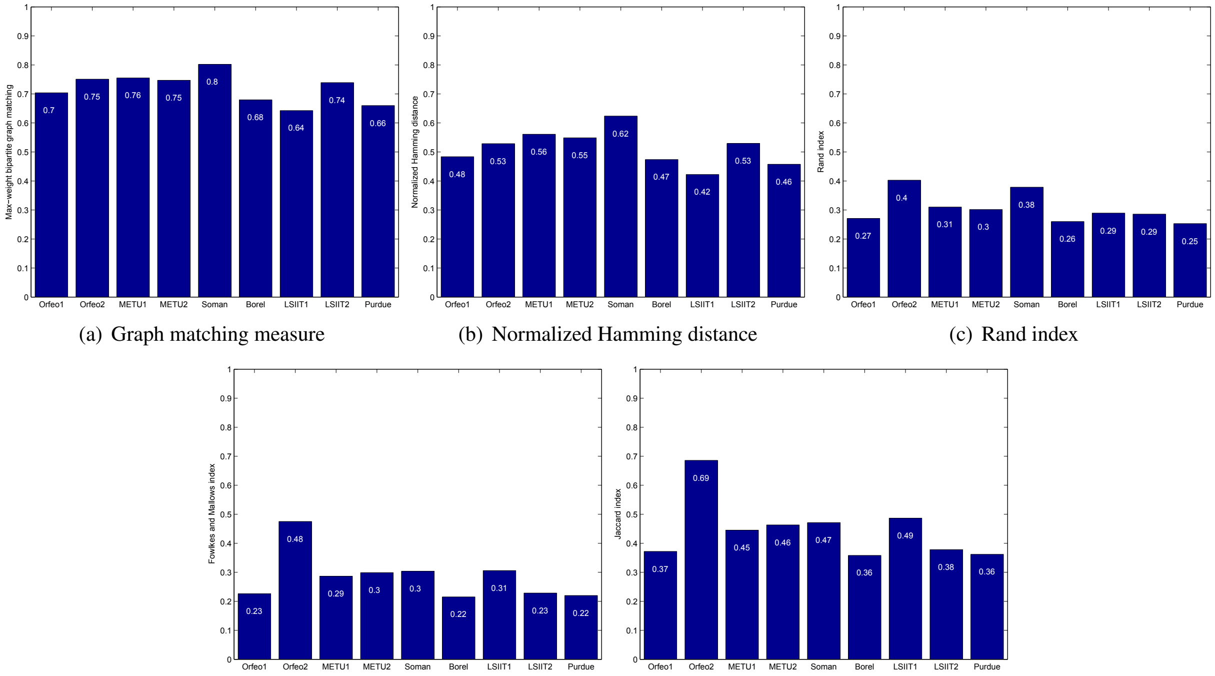
(b) Over-detection

(c) Under-detection

(d) Missed detection

(e) False alarm

Figure 3. Object-based correct detection, over-detection, under-detection, missed detection, and false alarm rates for the nine submission for the building detection task (Task 1).



(a) Graph matching measure

(b) Normalized Hamming distance

(c) Rand index

(d) Fowlkes and Mallows index

(e) Jaccard index

Figure 4. Graph matching measure, normalized Hamming distance, and clustering indices for the nine submissions for the building detection task (Task 1).

The nine submissions shared many steps such as pan-sharpening, spectral feature extraction (e.g., NDVI 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. As can be seen in Figures 3 and 4, no single method stood out as the best performer with respect to all performance measures. Similarly, different criteria favored different methods. New criteria for measuring performance based on boundary errors and fragmentation errors will be added, and all performance measures will be combined to provide a ranking of the submissions using methods such as Hasse diagrams [22] or multi-objective optimization [3] in future work.

3. Task 2: Digital surface model extraction from stereo data

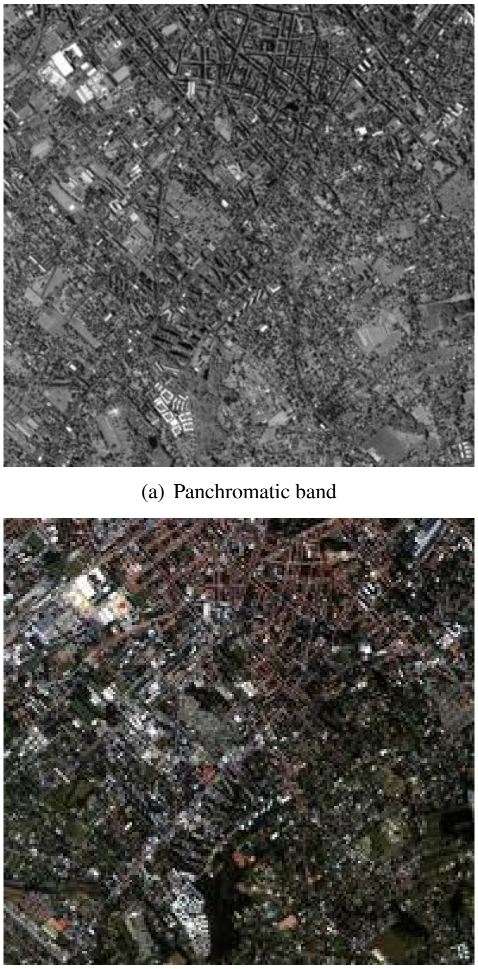
3.1. Background and data set

The objective of this task was to extract a digital surface model (DSM) for buildings from stereo Ikonos data of Graz, Austria. The data provided to the participants consisted of a pair of stereo images where each image had a panchromatic band with 1m spatial resolution and 2974×2918 pixels, and four multispectral bands with 4m spatial resolution and 792×749 pixels (Figure 5). Together with the data the rational polynomials were delivered to orthorectify the stereo images.

A highly accurate reference DSM was made available by the city of Graz (Figure 6). The reference DSM covered an area of 2km by 1km, and represented buildings typically found in European cities such as multistorey buildings with center courtyards, large industrial buildings, residential row houses, and single residential houses. The elevation in the Graz study area ranged from 390m to 480m above sea level, rising from West to East.

3.2. Participating methods

Three submissions were made by Jan Čech and Radim Šára. The first submission used a matching algorithm called Growing Correspondence Seeds (GCS) by Jan Čech. The second submission used a matching algorithm called 3-Label Dynamic Programming (3LDP) by Radim Šára. The submissions differed in putative correspondence pre-selection, and shared the matching procedure and disparity map post-processing. The third submission was a fusion of the GCS and 3LDP algorithms. These submissions are referred to as *GCS*, *3LDP*, and *Fusion*, respectively in the experiments.


(a) Panchromatic band

(b) Visible multispectral bands

Figure 5. One of the Ikonos images of Graz, Austria. (Copyright ©2007 GeoEye)

The putative correspondence stage of the GCS algorithm [5] was based on growing disparity patches (components) from sparse seed correspondences. The seed matches were found automatically. The normalized cross-correlation (MNCC) [16] was used for computing image similarity in a 5×5 neighborhood. This stage was followed by Confidently Stable Matching (CSM) [24] which performed pixel-wise selection from the grown components in a process of their mutual competition. The matching used a modified inhibition zone as described in [4]. Efficiency of this algorithm was achieved by avoiding aggregation over all possible correspondences in the disparity space. Usually, less than 1% of the disparity space was visited. The GCS algorithm

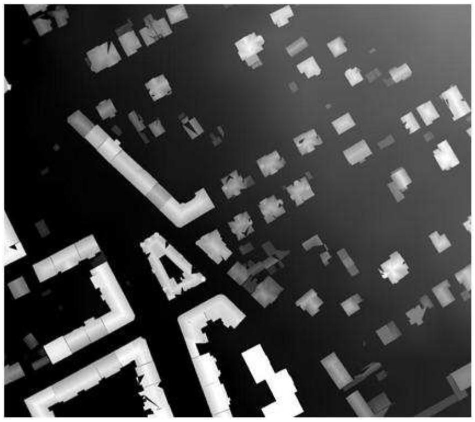


Figure 6. Part of the reference DSM used for task 2. (Kindly made available by Dr. Karlheinz Gutjahr, Joanneum Research, Institute of Digital Image Processing, A-8010 Graz, Austria, Wastiangasse 6, for internal use only.)

produced semi-dense disparity maps with explicitly labeled occlusions and textureless regions. A mixed Matlab/C implementation of the GCS algorithm is available at [4]. This implementation processed the test data in 130 sec using a single core of 2.2GHz Quad-Core AMD Opteron Processor 2354.

The 3LDP algorithm was a previously unpublished three-state dynamic programming stereo. It used all possible correspondences within a disparity search range as putative correspondences. The dynamic programming was used not to obtain a matching but to aggregate support for a subsequent matching procedure.

The algorithm was similar to four-state dynamic stereo programming by Criminisi *et al.* [7] and to an earlier work by Gimelfarb [9]. Unlike in [7], the matched state was modeled by a single label. Unlike in Gimelfarb, MNCC was used for image similarity [16]. The dynamic programming computed the total cost of the optimum path through every possible correspondence. This became the cost of a correspondence. Such aggregation was similar to the work of Gong and Yang [10]. The aggregation process was followed by a robust matching decision based on CSM, in exactly the same way as in GCS. The 3LDP algorithm produced semi-dense disparity maps in the same format as GCS did. The 3LDP algorithm including aggregation processed the test data in 323 sec using the same processor as in GCS but with a C implementation.

A simple fusion of the GCS and 3LDP algorithms was performed by projecting the resulting disparity maps into a common disparity space, computing image similarity anew, and re-running the final CSM procedure, as in GCS. Hereby, better correspondence hypotheses, proposed by either algorithm, were selected. This was an updated version of the disparity map fusion from [25]. The result of fusion was a more dense disparity map.

Since the disparity maps from the above three algorithms were semi-dense (76% density for GCS and 43% for 3LDP), a simple heuristic disparity map densification was included. The densification was designed exclusively for the purpose of evaluation in this contest where a 100% disparity map was required. Densification received a disparity map and the input images, and attempted to fill in the textureless and occluded regions. The result was a fully dense map. A similar procedure was shown effective for aerial imagery in [11].

The first stage of densification worked by proposing new disparities as follows. The reference image was over-segmented by the mean-shift algorithm [6]. Individual segments were processed one by one, and the contents of each segment S_i in the disparity map was subject to the following editing rules (in this order):

1. *Small Component Deletion Rule:* If the disparity map density in S_i fell below threshold T_d, the segment was deleted.

2. *Small Hole Patch Rule:* If the disparity map density in S_i raised above threshold T_p and the standard deviation of disparities in S_i was below threshold T_s, the S_i was replaced by its mean value.

3. *Occlusion Boundary Clip Rule:* If (1) the disparity histogram in S_i was strongly bimodal, and (2) one if its modes m_2 was significantly more prominent, and (3) narrow, the S_i was replaced by the mode value m_2.

4. *Large Hole Patch Rule:* The disparities around the periphery of every contiguous hole in the disparity map were collected. If their standard deviation fell below threshold T_s, the entire component was replaced by the mean value of the periphery. Otherwise, if the lower mode m_1 (corresponding to background disparity) was prominent and narrow, the segment was replaced by m_1.

All parameters were chosen manually to achieve visually acceptable results on a set of outdoor scenes similar to those used in [4]. The procedure removed small errors by Rule 1, patched small holes by Rule 2, removed occlusion artifacts by Rule 3, and patched the majority of large holes in textureless areas by Rule 4. The resulting disparity map was projected to a disparity space, MNCC image similarities were computed anew, and the CSM procedure was re-run once again, as in GCS. The result of this stage was a denser map, albeit not yet 100% dense since occlusions were preserved.

The purpose of the second densification stage was to extrapolate the disparity to occluded areas, most importantly, to mutually occluded regions occurring between tall buildings, where ground disparity should be assigned but the periphery of the occluded region had the building roof disparity. The procedure worked as follows. The image was split to overlapping tiles of 100×100 pixels. Lower quartile of disparity in each tile was computed. This approximated the terrain disparity. All remaining holes in the tile were replaced by this value. Contributions from multiple tiles covering the same pixel were averaged. The output from this procedure was a full-density disparity map.

3.3. Evaluation criteria

The performance of digital surface model extraction was evaluated using the residuals (difference) between the reference DSM and the output DSM. The following statistics were computed from the residuals:

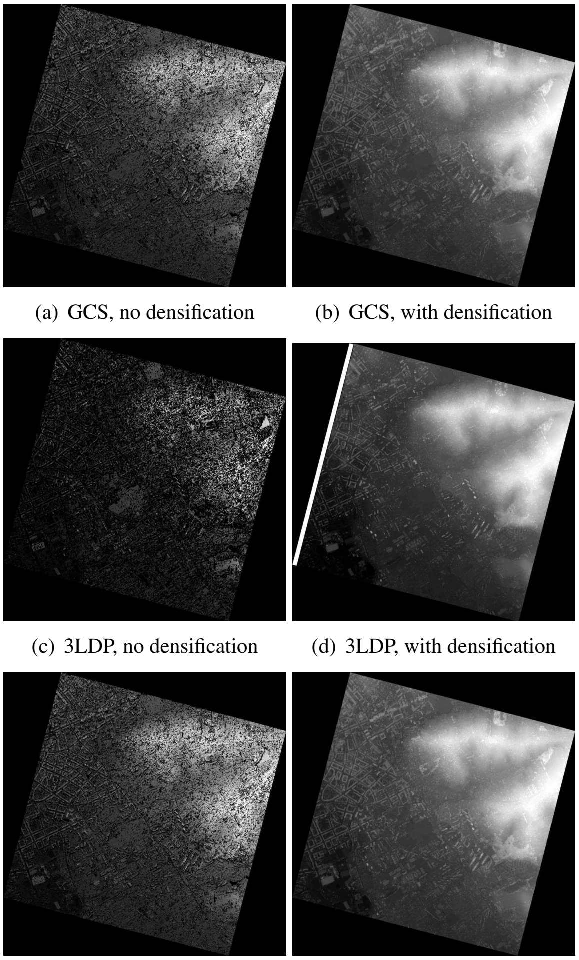
- Bias: mean, standard deviation, and skewness of the residuals.
- Precision: root-mean-squared error (RMSE) and frequency of outliers in the residuals.

3.4. Results

The digital surface models produced by the *GCS*, *3LDP*, and *Fusion* methods without and with densification are shown in Figure 7. The statistics described in Section 3.3 were computed for all three submissions as shown in Table 1. The 3LDP method produced the most accurate result compared to the ground truth. This is also confirmed by visual comparison of the resulting DSMs.

4. Conclusions

This paper presented the results of the PRRS 2008 Algorithm Performance Contest. The contest tasks consisted of automatic building detection from a single QuickBird image, and digital surface model extraction from stereo Ikonos data. Both data sets included ground truth for performance evaluation. We described the data sets, the methods used in the contest submissions, the objective evaluation criteria, and the results of the initial evaluation.



(a) GCS, no densification (b) GCS, with densification

(c) 3LDP, no densification (d) 3LDP, with densification

(e) Fusion, no densification (f) Fusion, with densification

Figure 7. Submissions for the digital surface model extraction task.

The submissions shared some steps such as pansharpening, thresholding, mask generation, segmentation, etc., but different in the ways such steps were combined as well as the amount of supervision used. The evaluation showed that no single method stood out as the best performer with respect to all performance measures. Similarly, different criteria favored different methods. Future work includes combining these performance measures to provide a ranking of the submissions using methods such as Hasse diagrams [22] or multi-objective optimization [3].

References

[1] M. Beauchemin and K. P. B. Thomson. The evaluation of segmentation results and the overlapping area ma-

Table 1. Digital surface model extraction results. The outputs with densification were used.

	Bias			Precision	
	Mean	Std. deviation	Skewness	RMSE	Freq. of outliers
GCS	5.0819	5.9276	0.0041	7.8168	0.1668
3LDP	-0.2210	5.7441	-0.8314	5.7472	0.1620
Fusion	5.1226	5.8414	0.1582	7.7774	0.1655

trix. *International Journal of Remote Sensing*, 18:3895–3899, December 1997.

[2] A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing*, pages 6–17, 2002.

[3] L. Bruzzone and C. Persello. A novel protocol for accuracy assessment in classification of very high resolution multispectral and SAR images. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, Boston, Massachusetts, July 6–11, 2008.

[4] J. Čech. Growing correspondence seeds: A fast stereo matching of large images. [online] http://cmp.felk.cvut.cz/ cechj/GCS/, Last revision: December 2008.

[5] J. Čech and R. Šára. Efficient sampling of disparity space for fast and accurate matching. In *Proc CVPR'2008 BenCOS Workshop*, 2007.

[6] D. Commaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[7] A. Criminisi, A. Blake, C. Rother, J. Shotton, and P. Torr. Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming. *International Journal of Computer Vision*, 71(1):89–110, 2007.

[8] E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

[9] G. L. Gimel'farb, V. B. Marchenko, and V. I. Rybak. Algorithm of automatic matching of identical patches in stereopairs. *Kibernetika*, (2):118–129, 1972. In Russian.

[10] M. Gong and Y.-H. Yang. Fast stereo matching using reliability-based dynamic programming and consistency constraints. In *IEEE International Conference on Computer Vision*, volume 1, pages 610–617, 2003.

[11] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[12] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689, July 1996.

[13] Q. Huang and B. Dom. Quantitative methods of evaluating image segmentation. In *IEEE International Conference on Image Processing*, volume 3, pages 53–56, Washington, DC, October 1995.

[14] X. Jiang, C. Marti, C. Irniger, and H. Bunke. Distance measures for image segmentation evaluation. *EURASIP Journal on Applied Signal Processing*, 2006(Article ID 35909):1–10, 2006.

[15] S. Lefèvre. Knowledge from markers in watershed segmentation. In *IAPR International Conference on Computer Analysis of Images and Patterns (CAIP)*, volume 4673 of *Lecture Notes in Computer Sciences*, pages 579–586, Vienna, Austria, August 2007. Springer-Verlag.

[16] H. P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. IJCAI*, page 584, 1977.

[17] S. Muller and D. W. Zaum. Robust building detection in aerial images. In *ISPRS Workshop CMRT 2005 Object Extraction for 3D City Models, Road Databases and Traffic Monitoring - Concepts, Algorithms and Evaluation*, 2005.

[18] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[19] The ORFEO toolbox software guide. http://www.orfeo-toolbox.org, 2008.

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

[21] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, September 1993.

[22] G. P. Patil and C. Taillie. Multiple indicators, partially ordered sets, and linear extensions: Multi-criterion ranking and prioritization. *Environmental and Ecological Statistics*, 11:199–228, 2004.

[23] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[24] R. Šára. Robust correspondence recognition for computer vision. In *Proc COMPSTAT*, pages 119–131. Physica-Verlag, 2006.

[25] R. Šára, R. Bajcsy, G. Kamberova, and R. A. McKendall. 3-D data acquisition and interpretation for virtual reality and telepresence. In *Proc IEEE/ATR Workshop on Computer Vision for Virtual Reality Based Human Communications*, pages 88–93. IEEE Computer Society Press, January 1998.