Baltic J. Modern Computing, Vol. 7 (2019), No. 2, 255-270
https://doi.org/10.22364/bjmc.2019.7.2.05

Detection of Man-Made Constructions using LiDAR Data and Decision Trees

Sergejs KODORS

Rezekne Academy of Technologies
Atbrivoshanas str.115, Rezekne, LV-4601, Latvia

sergejs.kodors@rta.lv

Abstract. Real estate monitoring is very important aspect of country economics, but old manual methods of land survey are time and resources consuming processes as geodata actualization tasks. Actual, precise, multidimensional and detailed information is the main instrument of geospatial intelligence to understand current economic situation and to make effective decision. Actualization of geoinformation using remote sensing is the modern approach of the computer age to complete Earth observation and human environment monitoring. This article describes multi-stage classification model, which detects man-made constructions in LiDAR point cloud. Proposed classification model applies decision tree and geometrical features of shape to remove noises. The goal of study is to experimentally compare decision trees with crisp and fuzzy logic (ID3 algorithms) to select the more suitable algorithm for noise reduction task. Algorithms are compared using total accuracy and Cohen's Kappa coefficient.

Keywords: classification, decision tree, features, fuzzy, ID3, LiDAR, real estate

Introduction

Land and rural development is important part of human existence, however, natural resources must be efficiently used considering different factors like environment protection, cultural heritage, the potential for development of tourism and manufacturing, legal and economic conditions, etc. The geospatial intelligence can take a correct decision about effective usage of Earth resources, only if they have precise and sufficiently detailed information about actual geospatial situation. Therefore geospatial data actualization must be completed on an on-going basis.

Remote sensing is the modern approach of the computer age to complete Earth observation and monitoring providing relatively fast and cheap solutions to make geospatial data actualization, but the obtained data must be preprocessed to get statistical and semantical information for decision-making. The remote sensing data can be analysed manually, but it is time-consuming process due to massive amount of data, that makes necessary to develop the automatic data actualization systems with the high performance computing (HPC) solution.

This research is a follow-up to HPC system development for real-estate actualization using LiDAR point cloud and computer vision (Kodors et al., 2017). The proposed system consists from three stages:

1st stage: filtering of last return points in LiDAR point cloud to remove vegetation and other noises;

2nd stage: detection and segmentation of surface facilities using min-cut method;

3rd stage: classification of surface facilities to identify buildings among them;

where the goal of each stage is to remove additional noise-objects (see Fig.1).

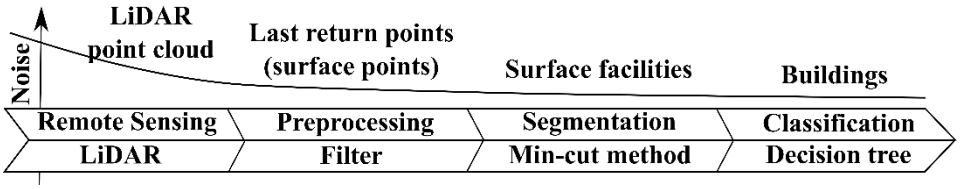

Fig. 1. Classification system with multi-stage noise filtering

The initial classification model has used filter by area to identify buildings among noise-objects (Kodors et al., 2017). The result of LiDAR point cloud processing is vector layer with building shapes prepared for geographical information systems (GIS). The obtained vector layer is compared with a previous layer to detect geospatial changes using an intersection of shapes. When geospatial changes are detected, image analyst must verify them using orthophoto, spectral images or LiDAR before to make data actualization.

Geospatial data belongs to the big data. Therefore, despite the high recognition accuracy of system, even the error smaller than 1% provides too many false objects. To improve classification accuracy, it was decided to replace area filter with decision tree. The previous study was related with geometric feature selection to filter buildings from walls, robust trees, large cars and other surface objects using the random forest of decision trees with crisp logic. 11 geometric features were studied and 5 features were selected as the most effective for classification providing solution with total accuracy 99% and Cohen's Kappa coefficient 0.90 (Kodors, 2017). However, completing classification tasks some authors obtain better accuracy results using fuzzy decision trees comparing with crisp decision trees (Idri and Elyassami, 2011). Therefore, the goal of this study is to compare classification accuracy of two decision tree models: with crisp logic and with fuzzy logic; as a solution for building recognition using the geometric features of shapes. Additional task of study is to measure influence of correct classification probability into recognition accuracy and the loss of data, what can be applied to set verification priority for image analysts.

1. Decision Trees and Remote Sensing

Decision Trees are classification methods and algorithms with relatively long history. The idea of using decision trees to identify and to classify objects is firstly mentioned by Hunt et al. in 1996 (Sharma, 2013). Decision trees successfully find application in tasks related with classification using remote sensing data. For example, decision trees are applied to classify land covers using spectral images due to natural approach, when each pixel is analysed independently (Sharma, 2013), (Kulkarni and Shrestha, 2017), (Pooja

et al., 2011), (Kulkarni and Lowe, 2016). However, pixel-based methods become ineffective with resolution increase (Veljanovski et al., 2011), but it does not reduce the significance of decision trees as classification method, which found renaissance in processing of shape or segment features. For example, LiDAR point cloud can be projected into 2D grid using voxel indices with subsequent classification of each pixels (Nesrine et al., 2009); or shape can be described using mathematical parameters compatible with input of decision tree (Jamil and Bakar, 2006), that was applied in study (Kodors, 2017).

Fuzzy Decision Trees are based on fuzzy logic introduced by Zadeh in 1965 (Idri and Elyassami, 2011). Completing experimental comparison, some authors obtain better accuracy results using fuzzy decision trees in place of decision trees with crisp logic (Idri and Elyassami, 2011). Fuzzy decision trees do not directly work with input data, each value is firstly preprocessed by membership function, which identifies strength of belonging to some subcategory of feature called event. Fuzzy trees have been applied for LiDAR processing before: pixel-based solution for forest boundary detection (Zhang et al., 2017) and object-based – for land cover classification (Syed et al., 2005).

2. Materials and methods

2.1. Dataset

25 samples of LiDAR point cloud have been applied in the experiment. The dataset of LiDAR data was provided by the State Land Service of Latvia for research tasks. The data was collected considering next technical parameters (WEB, a):

- the total minimal point density must be 4 p/m^2, the DEM must have minimum 1.5 p/m^2;
- the vertical precision must be 0.12 m with the level of confidence 95%;
- the horizontal precision must be 0.36 m with the level of confidence 95%.

The collected data was preprocessed, filtered and classified, each sample contained the point cloud with area 1 km^2 and the minimal point density equal to 1 p/m^2.

The provided dataset was processed using next algorithm (Kodors et al., 2017):

1st step: LiDAR point cloud is filtered to retain only surface points (single and last return points).

2nd step: LiDAR point cloud is projected into 2D grid recording the maximally high point in cell of area 1 m^2.

3rd step: the points with strong elevation (1.8 m) are marked as seed points for min-cut segmentation algorithm.

4th step: surface facilities are segmented using Dinic's algorithm with Dijkstra path finding algorithm.

5th step: obtained segments are vectorised using 4-path (rook type) Theo Pavlidis' algorithm to get shapes of object.

Each shape was manually classified into two classes "buildings" and "noise" verifying each object using cadastral map, orthophoto and LiDAR data classified points. Total number of shapes is 844 284 with 99.68% of noise-objects. Total number of unique shapes is 19 999, where 2 428 (12.14%) belong to buildings and 17 825 (89.13%) are noises.

5 geometric features (see Table 1) were calculated for each shape. The features were selected considering the previous study (Kodors, 2017).

Table 1. Geometric features of shapes

Feature	Equation		Variables
Area	$A = \sum p$	(1)	p – geospatial area of pixel; a – major axis (length of minimal bounding rectangle); b – minor axis (width of minimal bounding rectangle); A – area of shape; P – perimeter of shape.
Rectangularity	$R = \frac{A}{a \cdot b}$	(2)	
Form factor	$F = A / a^2$	(3)	
Compactness	$C_1 = \frac{P}{2\sqrt{\pi A}}$	(4)	
	$C_2 = P / A$	(5)	

2.2. Features of Dataset

The traditional classification methods with crisp logic try to find hyperplanes, which separate one class from other; however, classical logical reasoning is not effective due to intersection of feature values (see Fig.2-3).

Unique samples of dataset (19 999) have been analysed with a goal to identify how many common samples belong to classes “buildings” and “noise” depending on the number of features (see Table 2).

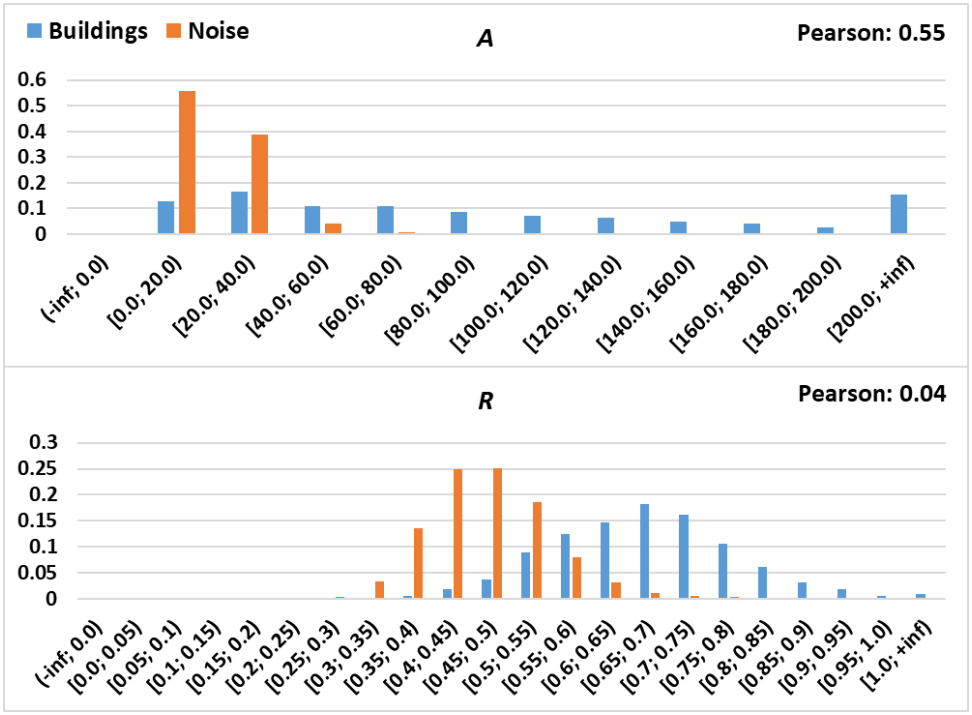

Fig. 2. Distribution of feature values for buildings and noise

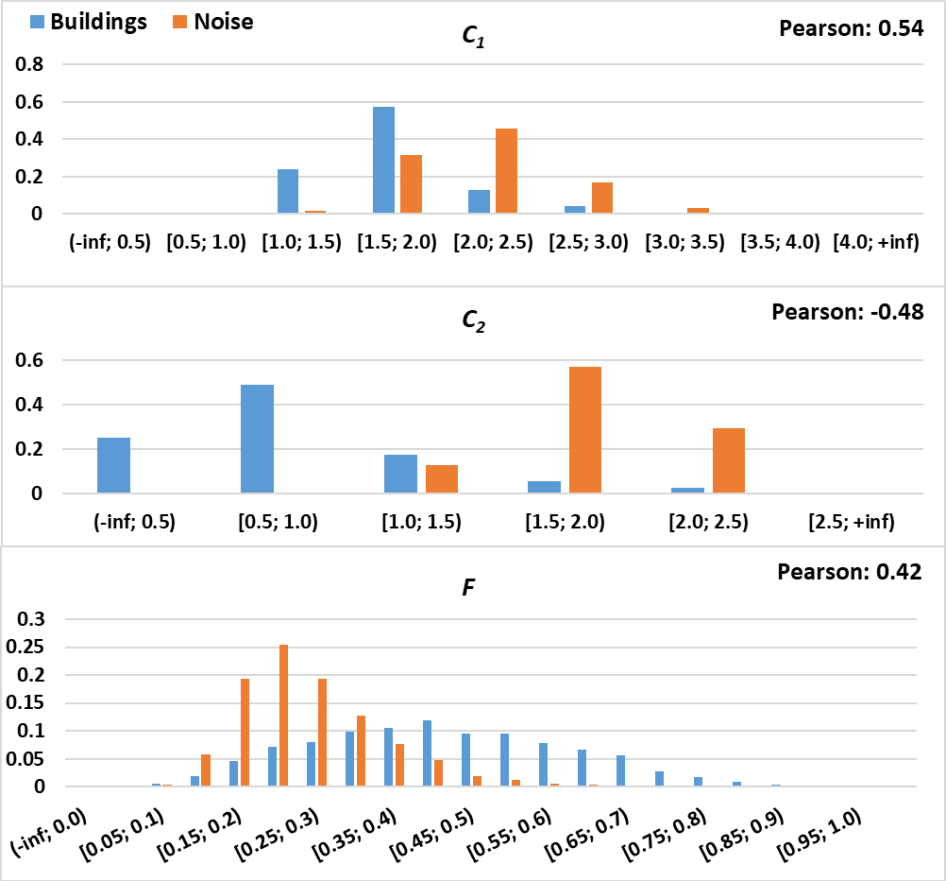

Fig. 3. Distribution of feature values for buildings and noise

Table 2. Decrease of common unique samples

Set of Features	Common samples	From buildings	Decrease Δ
{ C_2 }	735	30.27%	69.73%
{ C_2, R }	316	13.01%	57.01%
{ C_2, R, F }	257	10.58%	18.67%
{ C_2, R, F, A }	254	10.48%	1.18%
{ C_2, R, F, A, C_1 }	254	10.48%	0.00%
Unique samples = 19 999, Buildings = 2 428, Noise = 17 825			

The analysis of common sample decrease depending on the set of features (see Table 2) has showed, that feature C_1 does not minimize the number of common samples. The Spearman correlation between C_1 and A is 0.626 for class "noise" and 0.423 for class "buildings", according to source (Kodors, 2017). If C_1 is compared with C_2, the equations have similar form. The correlation analysis (Spearman) for 254 common samples has showed, that C_1 has very weak correlation with A (0.039),

moderate – with C_2 (0.456) and strong – with{ F (-0.707), R (-0.693) }. The feature A has very strong correlation with C_2 (–0.838) and weak – with { F (0.223), R (0.183), C_1 (0.039) }.

Completing the analysis of entropy (see Eq.1-2 (Sharma, 2013), (Pooja et al., 2011), (Kulkarni and Lowe, 2016) and see Table 3), the higher information gain is provided by features { C_2, A, R }, that complies with features' importance; but { C_1, F } are weak features for cluster split. The conclusion is "A and C_1 features do not replace each other, simply C_1 is too weak in this case to decrease number of common samples".

$$E(D) = -\sum_{k=1}^{n} \rho(c_k) \log_2 \rho(c_k), \quad (1)$$

where E – the entropy of dataset D,
n – the number of classes;
c_k – a class;
D – a dataset;
$\rho(c_k)$ – probability of class c_k.

$$G(a_i) = E(D) - \sum_{j=1}^{v} \rho(a_{ij}) E(D \mid a_{ij}), \quad (2)$$

where $G(a_i)$ – an information gain of feature a_i;
a_i – a feature;
j – a band of feature (subgroup of value range);
$\rho(a_{ij})$ – probability, that a sample of feature a_i belongs to a band j (see Eq.3);
$E(D \mid a_{ij})$ – an entropy of subdataset ($D \mid a_{ij}$) (see Eq.1);
$E(D)$ – the entropy of all dataset (see Eq.1).

$$\rho(a_{ij}) = |(D \mid a_{ij})| / N, \quad (3)$$

where $\rho(a_{ij})$ – probability, that a sample of feature a_i belongs to a band j;
$N = |D|$ – size of all dataset;
$(D \mid a_{ij})$ – samples, which belong to a band j of feature a_i.

Table 3. Information gain

Feature	Information gain	Feature	Information gain
C_2	0.344	F	0.132
A	0.236	C_1	0.101
R	0.233	Entropy of dataset = 0.583	

So, there is not possibility to uniquely classify 254 shapes using the set of features { $C2$, R, F, A, C_1 }, however, each sample of shape has different probability to belong to each class (see Fig.4), which can be applied by classification system, when users get classified layer with probability coefficient for each shape. The frequency analysis is completed for each feature (see Fig.5).

Frequency and distribution analysis is applied to define membership functions for fuzzy decision tree and to better understand each feature.

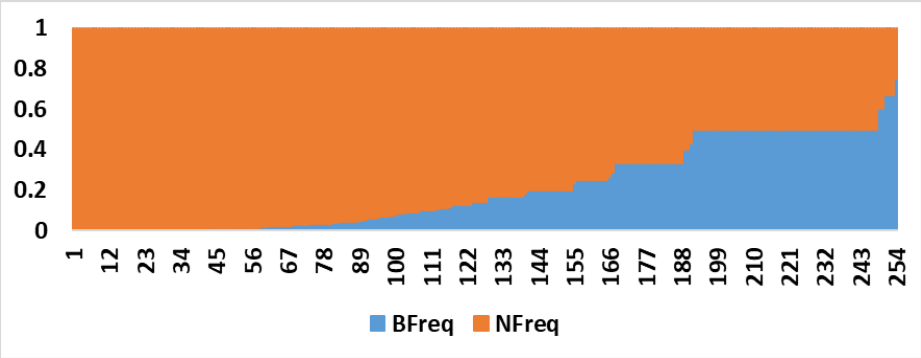

Fig. 4. Frequency separation for class "buildings" (*BFreq*) and "noise" (*NFreq*) in case of 254 common unique samples, where *X* – *ID* of sample, *Y* – frequency

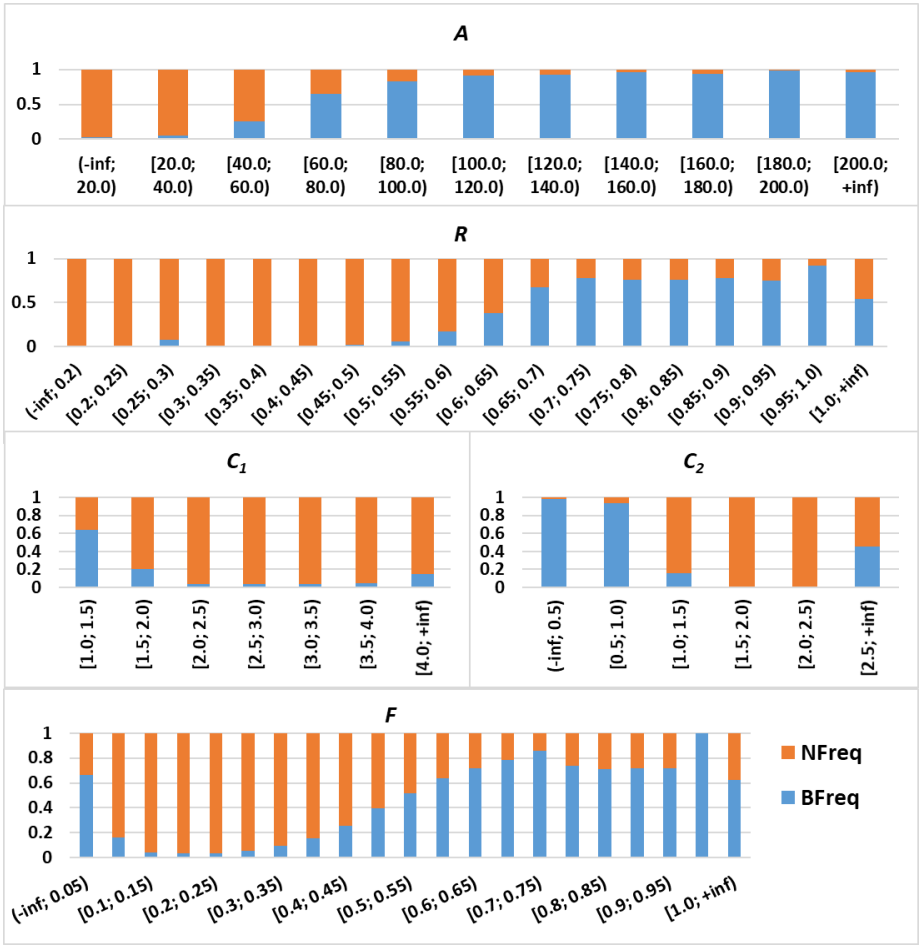

Fig. 5. Frequency distribution in feature bands, where *BFreq* – frequency of buildings and *NFreq* – frequency of noise

2.3. Compared algorithms

Interactive Dichotomizer 3 (ID3)

Function: ***Create node***
Goal: to generate decision tree.
Input: D – a training dataset, *A* – the set of features, *C* – the set of classes.
Output: node of decision tree.
Start
If ($|D| = 0$)
Return empty node (∅).

Else-if ($\exists! \, |(D|c_k)| > 0$)
Return the leaf node of class $c_k \in C$ with probability 1.0.

Else-if ($|A| = 1$)
Return the leaf node of feature $a \in A$ with probability $\rho(c)$ of each class.

Else
Calculate the entropy *E* of dataset *D* using Eq.1;
Calculate the information gain $G(a_i)$ using Eq.2 for each attribute $a_i \in A$;
Select the attribute with the maximal information gain a_{max};
Construct new node with the feature a_{max};
For each band *b* of the feature a_{max} obtain subdataset D'_b;
Remove the feature a_{max} from the set: $A' = A - a_{max}$;
For each band of new node call the function *"Create node"* with the parameters (D'_b, A', *C*);
Return the node of decision tree with probability $\rho(c)$ of each class.
End

Function: ***Classify sample***
Goal: to classify sample.
Input: n – a node of decision tree, *s* – a sample.
Output: class and its probability.
Start
If (*n* is leaf node)
Return class with the maximal probability in the current node *n*.

Else
Select next node n' by the band of node feature;

If ($n' = \emptyset$)
Return class with the maximal probability in the current node *n*.

Else
Return output of the function *"Classify sample"* using parameters (n', *s*).
End

Fuzzy Interactive Dichotomizer 3 (FID3)

Introduction: FID3 is based on ID3 algorithm, the difference is the star entropy calculated using membership functions. A linguistic group of feature is called event and its probability is defined by the membership function (see example in Fig.6).

Function: ***Create fuzzy node***
Goal: to generate fuzzy decision tree.
Input: D – a training dataset, A – the set of attributes, C – the set of classes, M – membership functions.
Output: fuzzy node.
Start
If ($|D| = 0$)
Return empty node ($\emptyset$).
Else-if ($\exists! \, |(D|c_k)| > 0$)
Return the leaf node of class $c_k \in C$ with probability 1.0.
Else-if ($|A| = 1$)
Return the leaf node of feature $a \in A$ with probability $\rho(c)$ of each class.
Else
For each class $c \in C$ calculate the star probability $\rho^*(c_k|a_{ij})$ using Eq.6.
For each feature $a_i \in A$ calculate the star entropy E^* using Eq.4;
Select feature $a_{min} \in A$ with the minimal star entropy E^*;
Construct new fuzzy node with the feature a_{min};
For each event g of feature a_{min} obtain subdataset $D'_g \subset D$;
Remove the feature a_{min} from set: $A' = A - a_{min}$;
For each event of new fuzzy node call the function *"Create fuzzy node"* with the parameters (D'_g, A', C, M);
Return the fuzzy node with probability $\rho(c)$ of each class.
End

Function: ***Classify sample***
Goal: to classify sample.
Input: n – a node of decision tree, s – a sample, M – membership functions.
Output: class and its probability.
Start
If (n is leaf node)
Return class with the maximal probability in the current node n.

Else
Select next node n' by the membership function $m \in M$ with the maximal output;

If ($n' = \emptyset$)
Return class with the maximal probability in the current node n.

Else
Return output of the function *"Classify sample"* using parameters (n', s).
End

Fuzzy Equations:

$$E^*(a_i) = -\sum_{j=1}^{m_i} \rho^*(a_{ij}) \sum_{k=1}^{n} \rho^*(c_k|a_{ij}) \log_2 \rho^*(c_k|a_{ij}), \quad (4)$$

where $E^*(a_i)$ – the star entropy of feature a_i;
n – the number of classes;
c_k – a class;
m_i – the number of membership functions (events) of feature $a_i \in A$;
$\rho^*(a_{ij})$ – the star probability of event a_{ij} (see Eq.5);
$\rho^*(c_k|a_{ij})$ – the star probability of event a_{ij} for class c_k (see Eq.6).

$$\rho^*(a_{ij}) = \sum \mu_{ij}(d) \,/\, N, \quad (5)$$

where $\rho^*(a_{ij})$ – mean star probability of event a_{ij};
$N = |(D \mid a_{ij})|$ – the number of samples, which belong to event a_{ij};
$\mu_{ij}(d)$ – a membership function j of attribute a_i;
d – samples, which belong to event a_{ij} Sample belongs to event with the maximal output of a membership function in other words $\{d \in D \mid a_{ij} \leftarrow \max \mu\}$.

$$\rho^*(c_k|a_{ij}) = \sum \mu_{ij}(d') \,/N', \quad (6)$$

where $\rho^*(c_k|a_{ij})$ – mean star probability of class c_k in event a_{ij};
$N' = |(D \mid c_k)|$ – size of subdataset D, where all samples belong to class c_k;
$\mu_{ij}(d')$ – a membership function j of attribute a_i;
d' – samples, which belong to class c_k and event a_{ij}.

3. Results and discussions

The obtained dataset was processed using ID3 and FID3 algorithms. Each algorithm was analysed using two training approaches:

a) the algorithms are trained using dataset with the unique samples (see Tables 4a and 5a);
b) the algorithms are trained using the full dataset with repeating samples (with probability of each sample), (see Tables 4b and 5b).

Validation was completed using the full dataset with repeating samples, where the area of objects is greater than 10 m^2 (this condition is defined in the previous study (Kodors, 2017)).

The histogram bands of Fig.2-3 and Fig.5 were used by ID3 algorithm to construct the decision tree with the crisp logic. FID3 algorithm has used the membership functions manually defined using the distribution and frequency analysis of data (see Fig.6).

The accuracy of each algorithm is evaluated using the confusion matrix, the total accuracy (*A*) and Cohen's Kappa coefficient (*K*) (see Table 4 and 5). Additionally the results of experiments are compared with the random forest algorithm applied in the previous study (Kodors, 2017).

The result of experiment has showed, that a fuzzy decision tree can process unique samples, when a frequency about each sample is unknown (see Table 5a). However, ID3 algorithm is more precise (see Table 4b), if there is a sufficiently large dataset to obtain probability of unique samples. This comparison of the algorithms identifies the importance of sample probability for ID3 algorithm. In contrast, a fuzzy decision tree applies knowledge about a sample probability hidden in membership functions. To verify the possibility of FID3 to identify unknown samples, the dataset of unique samples was split into the training dataset (20%) and the validation dataset (80%). The measurements were completed 1000 times and they provided next results:

- A_{min} = 0.90687, A_{mean} = 0.90988, A_{max} = 0.91292;
- K_{min} = 0.54366, K_{mean} = 0.55919, K_{max} = 0.57290.

Therefore, fuzzy decision trees are preferable, when there is not a dataset with the probability of samples, but experts can identify linguistic groups, which generalize biases and probability.

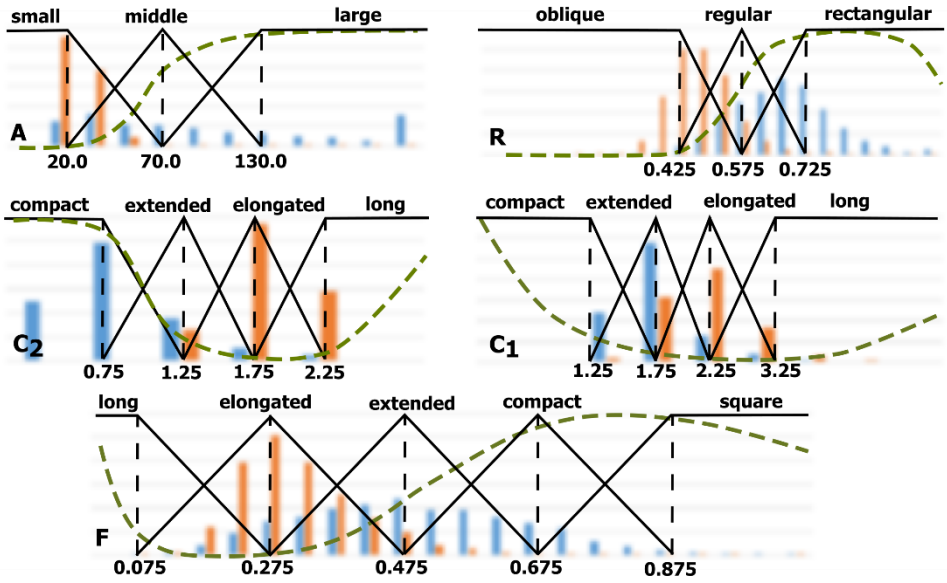

Fig. 6. The membership functions approximated using the distribution of features (the histograms) and the frequency distribution (the green dot-lines)

Table 4. Experiment results of ID3 using different training datasets

a) Only unique samples			b) All samples	
	B	**N**	**B**	**N**
B	0.00720	0.75063	0.04471	0.00259
N	0.04932	0.19285	0.01181	0.94089
A = 0.20005, K = –0.09779			A = 0.98560, K = 0.85375	

Table 5. Experiment results of FID3 using different training datasets

a) Only unique samples			b) All samples	
	B	**N**	**B**	**N**
B	0.00361	0.04154	0.04314	0.00287
N	0.05291	0.90195	0.01338	0.94061
A = 0.90555, K = 0.02196			A = 0.98375, K = 0.83299	

Comparing with a random forest algorithm applied in the previous study (Kodors, 2017), both algorithms ID3 and FID3 have smaller precision, their errors are 1.4% and 1.6% versus 1.1% of the random forest algorithm.

Selecting an answer, a decision tree verifies probability of each class in the node. Considering that, probability can be applied to filter incorrect answers; however, it provides the loss of data (see Fig.7). According to Fig.7, ID3 is more robust – 60% of buildings are classified with probability 99% unlike 40% of FID3. The parameter "probability" can be provided together with a shape, it will be useful to accelerate manual data verification using the filter of GIS.

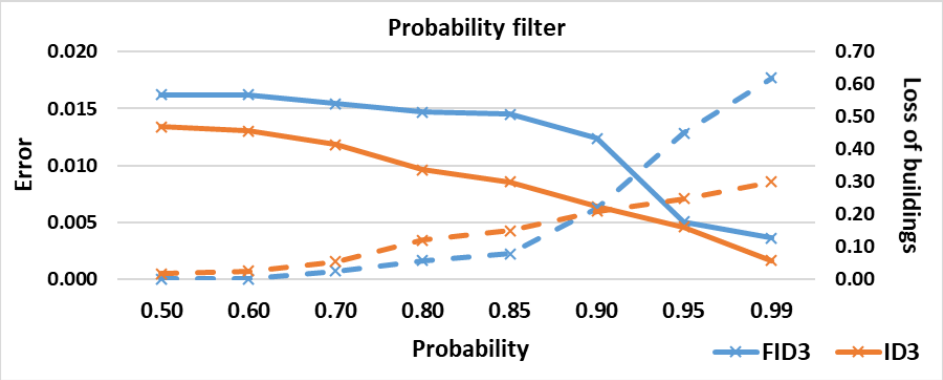

Fig. 7. Error decrease and data loss increase depending on probability of correct answer

Analysing the constructed fuzzy decision tree, 15 rules with 99% probability of the category "Buildings" were obtained (see Table 6). All rules excluding the 9th row contain compactness (C_2) equal to value "compact". It identifies relatively strong "linear dividing" of classes using this feature that correlates with the result of feature analysis in the previous research (Kodors, 2017).

Looking globally into the building detection and classification problem using remote sensing data with the high resolution, ISPRS provides benchmark test (WEB, b) with spectral and DSM data, which has the ground sampling distance equal to 9 cm. The Vashington 2D labelling challenge provides building classification precision F_1 from 0.82 to 0.96 with mean value 0.93, where the unit of calculation is a pixel. The proposed method with filter 16 m^2 (Kodors et al., 2015) had the F_1 score 0.95, While the proposed method with improved filter by the random forest algorithm provides F_1 equal to 0.985. However, the different resolution and landscape of ISPRS and experiment datasets must be considered. Therefore, the precise comparison, of course, must be completed using ISPRS benchmark dataset.

Table 6. Classification rules of buildings

Rules with probability 99%
1) *Area*=”small” *AND R*=”regular” *AND C2*=”compact” *AND F*=”long”
2) *Area*=”small” *AND R*=”regular” *AND C2*=”compact” *AND F*=”compact”
3) *Area*=”small” *AND R*=”rectangular” *AND C2*=”compact” *AND C1*=”elongated”
4) *Area*=”small” *AND R*=”rectangular” *AND C2*=”compact” *AND C1*=”rectangular”
5) *Area*=”middle” *AND C2*=”compact” *AND R*=”oblique” *AND F*=”extended”
6) *Area*=”middle” *AND C2*=”compact” *AND R*=”regular” *AND C1*=”compact”
7) *Area*=”middle” *AND C2*=”compact” *AND R*=”rectangular” *AND C1*=”compact”
8) *Area*=”middle” *AND C2*=”compact” *AND R*=”rectangular” *AND C1*=”extended”
9) *Area*=”middle” *AND C2*=”extended” *AND R*=”rectangular”
10) *Area*=”large” *AND C2*=”compact” *AND R*=”oblique” *AND C1*=”extended”
11) *Area*=”large” *AND C2*=”compact” *AND R*=”regular” *AND C1*=”compact”
12) *Area*=”large” *AND C2*=”compact” *AND R*=”rectangular” *AND F*=”long”
13) *Area*=”large” *AND C2*=”compact” *AND R*=”rectangular” *AND F*=”elongated”
14) *Area*=”large” *AND C2*=”compact” *AND R*=”rectangular” *AND F*=”compact”
15) *Area*=”large” *AND C2*=”compact” *AND R*=”rectangular” *AND F*=”square”

The well-developed libraries like *TensorFlow*, *Keras*, *Caffe*, etc., supporting GPU calculations increased the number of machine learning engineers. And the understandable supervised solution, expected high precision, available open data, plenty of training courses and simple tuning model only increases the number of deep learning scholars. Therefore, nowadays, the deep learning is massively used for image classification including LiDAR data processing (Yang et al., 2017; Rizaldy et al., 2018; Sun et al., 2018a, 2018b). The deep learning is based on the application of the artificial neural networks and it is intuitive to use 2D projection of LiDAR data as input that is actually applied in practise. As result, the deep learning deserves attention, because the proposed semantic segmentation algorithm based on the energy minimization approach methodology processes 2D projection of LiDAR data too.

The deep learning scholars propose next results: overall kappa = 0.89 (Sun et al., 2018a), F_1 of roofs = 0.93 and F_1 of impervious surfaces = 0.90 (Yang et al., 2017), F_1 score of buildings = 0.95 (Sun et al., 2018b); that is close to the mean F_1 score for ISPRS data (WEB, b). Therefore, it can be concluded, that the proposed method has potential, which is comparable to deep learning methods, but it does not require training and the high-performance computing as deep learning solutions. However, the deep learning is applicable to process orthoimages and shows good results for building detection providing F_1 equal to 0.95 (Liu et al., 2018), that is important considering the fact, that airborne and satellite imaging are the more cost-effective services neither airborne laser scanning.

But regardless of deep learning and proposed segmentation classification algorithm precisions, the ID3, FID3 and the random forests filters, which analyse the geometric features of shapes, can extend all classification algorithms providing different precision improvement for each method. Considering to this experiment, the improvement is equal to $\Delta F_1 = 0.035$, that replaces the method from category “middle” ($0.82 < x < 0.96$) to “high” ($x > 0.96$). Of course, it must be considered, that proposed method only detects buildings.

Other application of geometric feature filters is quality control and tuning. It is time-consuming to analyse the visual features of the correctly and incorrectly classified objects, but the conversion of adjectives like “compact”, “large”, “long”, etc. into digital/ mathematical form provides possibility to apply computers for big data analysis.

Conclusions

The analysis of common sample decrease (see Table 2) has showed, that 5 features { C_2, R, F, A, C_1 } do not separate all samples. Considering only classification accuracy, the correct solution is to add features for stronger division of classes, what can be obtained, for example, using features of spectral images. Firstly, increase of feature number requires additional performance, secondly, these additional features can be restricted, for example, if end user has only LiDAR data, therefore the increase of classification accuracy using a more powerful algorithm remains important task.

The comparison of the algorithms identifies, that the random forest algorithm provides better classification accuracy than ID3 and FID3. The errors among 3 algorithms are not very different, however, the difference is palpable in the case of the big data. The extension with the random forest algorithm increased the precision of previously published method (Kodors et al., 2015) from 0.95 to 0.985 (F_1 score) showing the high potential of method comparing with the modern solutions, which have the average precision equal to 0.93 (the deep learning solutions – approximately 0.95). Close classification results show, that intelligent system must be extended with additional services like land cover classification, 3D model generation etc., or 1st and 2nd classification stages must be improved, that is confirmed by 254 common shapes of both classes, which provides constantly incorrect classification.

However, tuning of current system is possible. Decision trees are working using linear separators, which are provided by bands defined by logical expressions “less than” and “greater than”. Fuzzy trees have overlays of membership functions, but the rule “event with maximal probability” identifies biases too in the intersection points between two events. Samples are distributed in multidimensional space and clusters can have custom forms. Linear separators can draw custom forms out, only if pixilation is sufficiently small. Therefore, the usage of PCA transformation can provide the better space for linear split of classes, that can improve classification accuracy and simplify decision tree structure; but the bi-plots – additional information about relations among the features.

The significance of feature “compactness” (C_2) for classification task was proved by the entropy and fuzzy decision tree structure analysis, that correlates with the results of the previous study (Kodors, 2017).

Speaking about the type of logic, the crisp logic was more effective in the present case, when frequency distribution between classes is known for each sample.

The error decrease and data loss increase depending on probability of correct answer identify, that data loss increase is too strong to apply this filter for automatic classification. However, it is useful for the semi-automatic solution, when the part of data is accepted without manual verification, but other data are verified considering decrease of building probability according to classifier.

Acknowledgement

Author expresses his gratitude to the State Land Service of Latvia and to Latvian Geospatial Information Agency for providing remote sensing data for the research purposes.

References

Idri, A., Elyassami, S. (2011). Applying Fuzzy ID3 Decision Tree for Software Effort Estimation. *IJCSI International Journal of Computer Science Issues*, Vol. 8, Issue 4, No. 1, 131-138.

Jamil, N., Bakar, Z.A. (2006). *Shape-Based Image Retrieval of Songket Motifs*. Proceedings of the 19th Annual Conference of the National Advisory Committee on Computing Qualifications (7-10 Jul., 2006, Wellington, New Zealand), Hillcrest, Hamiliton, New Zealand, 213-219, ISSN 1176-8053.

Kodors, S., Ratkevics, A., Rausis, A., Buls, J. (2015). Building Recognition Using LiDAR and Energy Minimization Approach, *Procedia Computer Science*, Vol. 43, 109–117, https://doi.org/10.1016/j.procs.2014.12.015

Kodors, S. (2017). *Geometric Feature Selection of Building Shape for Urban Classification*. Proceedings of the 11th International Scientific and Practical Conference, Environment. Technology. Resources (15-17 Jun. 2017, Rezekne, Latvia), Vol. II, Rezekne Academy of Technologies, Rezekne, 78-83, http://dx.doi.org/ 10.17770/etr2017vol2.2613

Kodors, S., Rausis, A., Ratkevics, A., Zvirgzds, J., Teilans, A., Ansone, I. (2017). Real Estate Monitoring System Based on Remote Sensing and Image Recognition Technologies. *Procedia Computer Science*, Vol. 104, 460-467, https://doi.org/10.1016/j.procs.2017.01.160

Kulkarni, A., Shrestha, A. (2017). Multispectral Image Analysis using Decision Trees. (*IJACSA*) *International Journal of Advanced Computer Science and Applications*,Vol. 8, No. 6, 11-18.

Kulkarni, A.D., Lowe, B. (2016). Random Forest Algorithm for Land Cover Classification. *International Journal on Recent and Innovation Trends in Computing and Communication*, Vol. 4, Issue 3, 58 – 63.

Liu, S., Ding, W., Liu, C., Liu, Y., Wang, Y., Li, H. (2018). ERN: Edge Loss Reinforced Semantic Segmentation Network for Remote Sensing Images. *Remote Sensing*, Vol. 10, 1339. doi:10.3390/rs10091339

Nesrine, C., Li, G., Mallet, C. (2009). *Airborne LiDAR Features Selection for Urban Classification Using Random Forests.* Laser scanning 2009, IAPRS (1-2 Sep., 2009, Paris, France), Vol. XXXVIII, International Society for Photogrammetry and Remote Sensing, Lemmer, 207-212, available at www.isprs.org/proceedings/XXXVIII/3-W8/papers/p207.pdf

Pooja, A.P., Jayanth, J., Shivaprakash, K. (2011). Classification of RS data using decision three approach. *International Journal of Computer Applications*, Vol. 23, No. 3, 7-11.

Rizaldy, A., Persello, C., Gevaert, C.M., Oude Elberink, S.J. (2018). *Fully Convolutional Networks for Ground Classification From Lidar Point Clouds*, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences (4–7 June, 2018, Riva del Garda, Italy), Vol. IV-2, International Society for Photogrammetry and Remote Sensing, Riva Del Garda, 231-238, available at www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/IV-2/231/2018/isprs-annals-IV-2-231-2018.pdf

Sharma, R., Ghosh, A., Joshi, P.K. (2013). Decision tree approach for classification of remotely sensed satellite data using open source support. *J. Earth Syst. Sci.,* Vol. 122, No. 5, 1237–1247.

Sun, Z., Zhao, X., Wu, M., Wang, C. (2018a). Extracting Urban Impervious Surface from WorldView-2 and Airborne LiDAR Data Using 3D Convolutional Neural Networks, Journal of the Indian Society of Remote Sensing, 1–12, 2018. https://doi.org/10.1007/s12524-018-0917-5

Sun, Y., Zhang, X., Zhao, X., Xin, Q. (2018b). Extracting Building Boundaries from High Resolution Optical Images and LiDAR Data by Integrating the Convolutional Neural Network and the Active Contour Model. *Remote Sensing*, Vol. 10, 1459. doi:10.3390/rs10091459

Syed, S., Dare, P., Jones, S. (2005). *Automatic Classification of Land Cover Features with High Resolution Imagery and Lidar Data: An Object-Oriented Approach*. Proceedings of SSC2005 Spatial Intelligence, Innovation and Praxis: The national biennial Conference of the Spatial Sciences Institute (Sep., 2005), Spatial Sciences Institute, Melbourne, 512-22, ISBN 0-9581366-2-9, available at https://pdfs.semanticscholar.org/47a3/6f82098005121b419e6b24ee6de18bb90702.pdf

Veljanovski, T., Kanjir, U., Ostir, K. (2011). Object-based image analysis of remote sensing data. *Geodetski Vestnik*, Vol. 55, No. 4, 678-688.

Yang, Z., Jiang, W., Xu, B., Zhu, Q., Jiang, S., Huang, W. (2017). A Convolutional Neural Network-Based 3D Semantic Labeling Method for ALS Point Clouds. *Remote Sensing*, Vol. 9, 936. doi:10.3390/rs9090936

Zhang, Z., Liu, X., McDougall, K., Wright. W. (2017). *Fuzzy Analysis of Airborne LiDAR Data for Rainforest Boundary Determination*. ICTRS'17, Proceedings of the 6th International Conference on Telecommunications and Remote Sensing (6–7 Nov., 2017, Delft, Netherlands), ACM, NY, 48-53, https://doi.org/10.1145/3152808.3152816

WEB (a). *Airborne Laser Scanning: Main Technical Parameters*, http://map.lgia.gov.lv/index.php?lang=0&cPath=4_5&txt_id=126

WEB (b). *ISPRS Test Project on Urban Classification and 3D Building Reconstruction: Results* http://www2.isprs.org/commissions/comm2/wg4/vaihingen-2d-semantic-labeling-contest.html

Received July 16, 2018, revised May 5, 2019, accepted May 6, 2019